Trio Unified API User Documentation for C# Language

Generated by Doxygen 1.10.0

1 Namespace Index 1
1.1 Namespace List . . . 1

2 Hierarchical Index 3
2.1 Class Hierarchy . . . 3

3 Class Index 5
3.1 Class List . . . 5

4 File Index 7
4.1 File List . . . 7

5 Namespace Documentation 9
5.1 Trio Namespace Reference . . . 9
5.2 Trio.UnifiedApi Namespace Reference . . . 9
5.2.1 Enumeration Type Documentation . . . 17
5.2.1.1 AxisMode . . . 17
5.2.1.2 AxisStatus . . . 17
5.2.1.3 AxisType . . . 18
5.2.1.4 CamBoxLinkOptions . . . 21
5.2.1.5 Co_ObjectType . . . 22
5.2.1.6 ConnectionType . . . 22
5.2.1.7 ConnectMode . . . 23
5.2.1.8 CornerMode . . . 23
5.2.1.9 CornerState . . . 23
5.2.1.10 ErrorCode . . . 24
5.2.1.11 EthercatStartOption . . . 33
5.2.1.12 EthercatState . . . 33
5.2.1.13 EventType . . . 33
5.2.1.14 FileFlags . . . 34
5.2.1.15 FileTransferOption . . . 34
5.2.1.16 FileType . . . 34
5.2.1.17 MotionType . . . 35
5.2.1.18 ParamBlockType . . . 36
5.2.1.19 ProcessStatus . . . 38
5.2.1.20 ProgressEvent . . . 38
5.2.1.21 PSwitchMode . . . 38
5.2.1.22 PSwitchMultiMode . . . 39
5.2.1.23 RobotStatus . . . 39
5.2.1.24 SyncControl . . . 40
5.2.2 Function Documentation . . . 40
5.2.2.1 ConnectionEventHandler() . . . 40
5.2.2.2 FileCallback() . . . 40
5.2.2.3 ProgressHandlerCallback() . . . 40

5.2.2.4 StreamHandlerCallback() . . . 41

6 Class Documentation **43**

6.1 Trio.UnifiedApi.DataStreamPacket Struct Reference . . . 43
6.1.1 Detailed Description . . . 43
6.1.2 Member Data Documentation . . . 43
6.1.2.1 data_length . . . 43
6.1.2.2 packet_count . . . 44
6.1.2.3 packet_number . . . 44
6.1.2.4 total_length . . . 44
6.2 Trio.UnifiedApi.FileInfo Struct Reference . . . 44
6.2.1 Detailed Description . . . 45
6.2.2 Member Data Documentation . . . 45
6.2.2.1 autorun_process . . . 45
6.2.2.2 compiled_size . . . 45
6.2.2.3 flags . . . 45
6.2.2.4 size . . . 45
6.2.2.5 type . . . 45
6.3 Trio.UnifiedApi.ITrioConnection Interface Reference . . . 46
6.3.1 Detailed Description . . . 75
6.3.2 Member Function Documentation . . . 75
6.3.2.1 AddAx() [1/2] . . . 75
6.3.2.2 AddAx() [2/2] . . . 75
6.3.2.3 AddDac() [1/2] . . . 75
6.3.2.4 AddDac() [2/2] . . . 76
6.3.2.5 Backlash() . . . 76
6.3.2.6 BacklashDisable() . . . 77
6.3.2.7 Cam() . . . 77
6.3.2.8 CamBox() . . . 78
6.3.2.9 Cancel() . . . 78
6.3.2.10 ClearVariables() . . . 79
6.3.2.11 CloseConnection() . . . 79
6.3.2.12 Connect() . . . 79
6.3.2.13 ConnPath() . . . 80
6.3.2.14 Datum() . . . 80
6.3.2.15 DefPos() . . . 80
6.3.2.16 DefPosMulti() . . . 81
6.3.2.17 Delete() . . . 81
6.3.2.18 DeleteFrameGroup() . . . 82
6.3.2.19 Dir() . . . 82
6.3.2.20 DownloadFile() . . . 82
6.3.2.21 EncoderRatio() . . . 83

6.3.2.22 Ethercat_CheckNumberOfSlaves() . . . 83
6.3.2.23 Ethercat_CheckNumberOfSlaves_VR() . . . 84
6.3.2.24 Ethercat_CheckSlaveOnline() . . . 84
6.3.2.25 Ethercat_CheckSlaveOnline_VR() . . . 84
6.3.2.26 Ethercat_CoRead() . . . 85
6.3.2.27 Ethercat_CoReadAxis() . . . 86
6.3.2.28 Ethercat_CoWrite_Value() [1/2] . . . 86
6.3.2.29 Ethercat_CoWrite_Value() [2/2] . . . 87
6.3.2.30 Ethercat_CoWrite_VR() . . . 88
6.3.2.31 Ethercat_CoWriteAxis_Value() [1/2] . . . 88
6.3.2.32 Ethercat_CoWriteAxis_Value() [2/2] . . . 89
6.3.2.33 Ethercat_CoWriteAxis_VR() . . . 90
6.3.2.34 Ethercat_DisplayNetworkConfig() . . . 90
6.3.2.35 Ethercat_GetEmergencyMsgControl() . . . 91
6.3.2.36 Ethercat_GetEmergencyMsgControl_VR() . . . 91
6.3.2.37 Ethercat_GetSlave_ESC_Register() . . . 92
6.3.2.38 Ethercat_GetSlaveAddress() . . . 92
6.3.2.39 Ethercat_GetSlaveAddress_VR() . . . 93
6.3.2.40 Ethercat_GetSlaveAxis() . . . 93
6.3.2.41 Ethercat_GetSlaveAxis_VR() . . . 93
6.3.2.42 Ethercat_GetState() . . . 95
6.3.2.43 Ethercat_GetState_VR() . . . 95
6.3.2.44 Ethercat_GetStatusMaskAndTriggerPattern() . . . 96
6.3.2.45 Ethercat_GetStatusMaskAndTriggerPattern_VR() . . . 96
6.3.2.46 Ethercat_SendResetSequence() [1/3] . . . 97
6.3.2.47 Ethercat_SendResetSequence() [2/3] . . . 97
6.3.2.48 Ethercat_SendResetSequence() [3/3] . . . 98
6.3.2.49 Ethercat_SetDistrClockMaster() . . . 98
6.3.2.50 Ethercat_SetDistrClockMaster_VR() . . . 98
6.3.2.51 Ethercat_SetEmergencyMsgControl() . . . 99
6.3.2.52 Ethercat_SetSlave_ESC_Register() . . . 99
6.3.2.53 Ethercat_SetState() . . . 101
6.3.2.54 Ethercat_SetStatusMaskAndTriggerPattern() . . . 101
6.3.2.55 Ethercat_SetStatusMaskAndTriggerPattern_VR() . . . 102
6.3.2.56 Ethercat_SetSync0StartTime() . . . 103
6.3.2.57 Ethercat_SetSync0StartTime_VR() . . . 103
6.3.2.58 Ethercat_StartNetwork() [1/3] . . . 103
6.3.2.59 Ethercat_StartNetwork() [2/3] . . . 104
6.3.2.60 Ethercat_StartNetwork() [3/3] . . . 104
6.3.2.61 Ethercat_StopNetwork() . . . 105
6.3.2.62 Ex() . . . 105
6.3.2.63 Execute() . . . 105

6.3.2.64 ExecuteWithResponse() [1/3] . . . 106
6.3.2.65 ExecuteWithResponse() [2/3] . . . 106
6.3.2.66 ExecuteWithResponse() [3/3] . . . 106
6.3.2.67 Forward() . . . 107
6.3.2.68 GetAIn() . . . 107
6.3.2.69 GetAOut() . . . 108
6.3.2.70 GetApiVersion() . . . 108
6.3.2.71 GetApiVersionStr() . . . 108
6.3.2.72 GetAxisParameter_ACCEL() . . . 109
6.3.2.73 GetAxisParameter_ADDAX_AXIS() . . . 109
6.3.2.74 GetAxisParameter_AFF_GAIN() . . . 109
6.3.2.75 GetAxisParameter_ATYPE() . . . 110
6.3.2.76 GetAxisParameter_AXIS_ACCEL() . . . 110
6.3.2.77 GetAxisParameter_AXIS_ADDRESS() . . . 111
6.3.2.78 GetAxisParameter_AXIS_DECEL() . . . 111
6.3.2.79 GetAxisParameter_AXIS_DISPLAY() . . . 111
6.3.2.80 GetAxisParameter_AXIS_DPOS() . . . 112
6.3.2.81 GetAxisParameter_AXIS_ENABLE() . . . 112
6.3.2.82 GetAxisParameter_AXIS_ENABLE_OVERRIDE() . . . 113
6.3.2.83 GetAxisParameter_AXIS_ERROR_COUNT() . . . 113
6.3.2.84 GetAxisParameter_AXIS_FS_LIMIT() . . . 114
6.3.2.85 GetAxisParameter_AXIS_MAX_JOG_SPEED() . . . 114
6.3.2.86 GetAxisParameter_AXIS_MODE() . . . 114
6.3.2.87 GetAxisParameter_AXIS_RS_LIMIT() . . . 115
6.3.2.88 GetAxisParameter_AXIS_SPEED() . . . 115
6.3.2.89 GetAxisParameter_AXIS_UNITS() . . . 116
6.3.2.90 GetAxisParameter_AXIS_Z_OUTPUT() . . . 116
6.3.2.91 GetAxisParameter_AXISSTATUS() . . . 116
6.3.2.92 GetAxisParameter_BACKLASH_DIST() . . . 117
6.3.2.93 GetAxisParameter_CHANGE_DIR_LAST() . . . 117
6.3.2.94 GetAxisParameter_CLOSE_WIN() . . . 118
6.3.2.95 GetAxisParameter_CLUTCH_RATE() . . . 118
6.3.2.96 GetAxisParameter_CORNER_MODE() . . . 119
6.3.2.97 GetAxisParameter_CORNER_STATE() . . . 119
6.3.2.98 GetAxisParameter_CREEP() . . . 119
6.3.2.99 GetAxisParameter_D_GAIN() . . . 120
6.3.2.100 GetAxisParameter_DAC() . . . 120
6.3.2.101 GetAxisParameter_DAC_OUT() . . . 121
6.3.2.102 GetAxisParameter_DAC_SCALE() . . . 121
6.3.2.103 GetAxisParameter_DATUM_IN() . . . 121
6.3.2.104 GetAxisParameter_DECEL() . . . 122
6.3.2.105 GetAxisParameter_DECEL_ANGLE() . . . 122

6.3.2.106 GetAxisParameter_DEMAND_EDGES() . . . 123
6.3.2.107 GetAxisParameter_DEMAND_SPEED() . . . 123
6.3.2.108 GetAxisParameter_DPOS() . . . 124
6.3.2.109 GetAxisParameter_DRIVE_CONTROL() . . . 124
6.3.2.110 GetAxisParameter_DRIVE_CONTROLWORD() . . . 124
6.3.2.111 GetAxisParameter_DRIVE_CURRENT() . . . 125
6.3.2.112 GetAxisParameter_DRIVE_CW_MODE() . . . 125
6.3.2.113 GetAxisParameter_DRIVE_ENABLE() . . . 126
6.3.2.114 GetAxisParameter_DRIVE_FE() . . . 126
6.3.2.115 GetAxisParameter_DRIVE_FE_LIMIT() . . . 126
6.3.2.116 GetAxisParameter_DRIVE_INDEX() . . . 127
6.3.2.117 GetAxisParameter_DRIVE_INPUTS() . . . 127
6.3.2.118 GetAxisParameter_DRIVE_MODE() . . . 128
6.3.2.119 GetAxisParameter_DRIVE_MONITOR() . . . 128
6.3.2.120 GetAxisParameter_DRIVE_NEG_TORQUE() . . . 129
6.3.2.121 GetAxisParameter_DRIVE_PARAMETER() . . . 129
6.3.2.122 GetAxisParameter_DRIVE_POS_TORQUE() . . . 129
6.3.2.123 GetAxisParameter_DRIVE_PROFILE() . . . 130
6.3.2.124 GetAxisParameter_DRIVE_SET_VAL() . . . 130
6.3.2.125 GetAxisParameter_DRIVE_STATUS() . . . 131
6.3.2.126 GetAxisParameter_DRIVE_TORQUE() . . . 131
6.3.2.127 GetAxisParameter_DRIVE_TYPE() . . . 131
6.3.2.128 GetAxisParameter_DRIVE_VALUE() . . . 132
6.3.2.129 GetAxisParameter_DRIVE_VELOCITY() . . . 132
6.3.2.130 GetAxisParameter_ENCODER() . . . 133
6.3.2.131 GetAxisParameter_ENCODER_BITS() . . . 133
6.3.2.132 GetAxisParameter_ENCODER_CONTROL() . . . 134
6.3.2.133 GetAxisParameter_ENCODER_FILTER() . . . 134
6.3.2.134 GetAxisParameter_ENCODER_ID() . . . 134
6.3.2.135 GetAxisParameter_ENCODER_STATUS() . . . 135
6.3.2.136 GetAxisParameter_ENCODER_TURNS() . . . 135
6.3.2.137 GetAxisParameter_END_DIR_LAST() . . . 136
6.3.2.138 GetAxisParameter_ENDMOVE() . . . 136
6.3.2.139 GetAxisParameter_ENDMOVE_BUFFER() . . . 136
6.3.2.140 GetAxisParameter_ENDMOVE_SPEED() . . . 137
6.3.2.141 GetAxisParameter_ERRORMASK() . . . 137
6.3.2.142 GetAxisParameter_FAST_JOG() . . . 138
6.3.2.143 GetAxisParameter_FASTDEC() . . . 138
6.3.2.144 GetAxisParameter_FE() . . . 139
6.3.2.145 GetAxisParameter_FE_LATCH() . . . 139
6.3.2.146 GetAxisParameter_FE_LIMIT() . . . 139
6.3.2.147 GetAxisParameter_FE_LIMIT_MODE() . . . 140

6.3.2.148 GetAxisParameter_FE_RANGE() . . . 140
6.3.2.149 GetAxisParameter_FEED_OVERRIDE() . . . 141
6.3.2.150 GetAxisParameter_FHOLD_IN() . . . 141
6.3.2.151 GetAxisParameter_FHSPEED() . . . 141
6.3.2.152 GetAxisParameter_FORCE_ACCEL() . . . 142
6.3.2.153 GetAxisParameter_FORCE_DECEL() . . . 142
6.3.2.154 GetAxisParameter_FORCE_DWELL() . . . 143
6.3.2.155 GetAxisParameter_FORCE_JERK() . . . 143
6.3.2.156 GetAxisParameter_FORCE_RAMP() . . . 144
6.3.2.157 GetAxisParameter_FORCE_SPEED() . . . 144
6.3.2.158 GetAxisParameter_FRAME() . . . 144
6.3.2.159 GetAxisParameter_FRAME_REP_DIST() . . . 145
6.3.2.160 GetAxisParameter_FRAME_SCALE() . . . 145
6.3.2.161 GetAxisParameter_FS_LIMIT() . . . 146
6.3.2.162 GetAxisParameter_FULL_SP_RADIUS() . . . 146
6.3.2.163 GetAxisParameter_FWD_IN() . . . 146
6.3.2.164 GetAxisParameter_FWD_JOG() . . . 147
6.3.2.165 GetAxisParameter_FWD_START() . . . 147
6.3.2.166 GetAxisParameter_I_GAIN() . . . 148
6.3.2.167 GetAxisParameter_IDLE() . . . 148
6.3.2.168 GetAxisParameter_IN_POS() . . . 149
6.3.2.169 GetAxisParameter_IN_POS_DIST() . . . 149
6.3.2.170 GetAxisParameter_IN_POS_SPEED() . . . 149
6.3.2.171 GetAxisParameter_INTERP_FACTOR() . . . 150
6.3.2.172 GetAxisParameter_INVERT_STEP() . . . 150
6.3.2.173 GetAxisParameter_JERK() . . . 151
6.3.2.174 GetAxisParameter_JOGSPEED() . . . 151
6.3.2.175 GetAxisParameter_LINK_AXIS() . . . 151
6.3.2.176 GetAxisParameter_LOADED() . . . 152
6.3.2.177 GetAxisParameter_LOOKAHEAD_FACTOR() . . . 152
6.3.2.178 GetAxisParameter_MARK() . . . 153
6.3.2.179 GetAxisParameter_MARKB() . . . 153
6.3.2.180 GetAxisParameter_MERGE() . . . 154
6.3.2.181 GetAxisParameter_MOVE_COUNT() . . . 154
6.3.2.182 GetAxisParameter_MOVE_PA() . . . 154
6.3.2.183 GetAxisParameter_MOVE_PA_CONT() . . . 155
6.3.2.184 GetAxisParameter_MOVE_PA_IDLE() . . . 155
6.3.2.185 GetAxisParameter_MOVE_PB() . . . 156
6.3.2.186 GetAxisParameter_MOVE_PB_CONT() . . . 156
6.3.2.187 GetAxisParameter_MOVE_PB_IDLE() . . . 156
6.3.2.188 GetAxisParameter_MOVELINK_MODIFY() . . . 157
6.3.2.189 GetAxisParameter_MOVES_BUFFERED() . . . 157

6.3.2.190 GetAxisParameter_MPOS() . . . 158
6.3.2.191 GetAxisParameter_MSPEED() . . . 158
6.3.2.192 GetAxisParameter_MSPEED_FILTER() . . . 159
6.3.2.193 GetAxisParameter_MSPEEDF() . . . 159
6.3.2.194 GetAxisParameter_MTYPE() . . . 159
6.3.2.195 GetAxisParameter_NTYPE() . . . 160
6.3.2.196 GetAxisParameter_OFFPOS() . . . 160
6.3.2.197 GetAxisParameter_OPEN_WIN() . . . 161
6.3.2.198 GetAxisParameter_OUTLIMIT() . . . 161
6.3.2.199 GetAxisParameter_OV_GAIN() . . . 161
6.3.2.200 GetAxisParameter_P_GAIN() . . . 162
6.3.2.201 GetAxisParameter_POS_DELAY() . . . 162
6.3.2.202 GetAxisParameter_POSI_SEQ_DELAY() . . . 163
6.3.2.203 GetAxisParameter_POSI_SEQ_MODE() . . . 163
6.3.2.204 GetAxisParameter_PP_STEP() . . . 164
6.3.2.205 GetAxisParameter_PS_ENCODER() . . . 164
6.3.2.206 GetAxisParameter_PWM_CYCLE() . . . 164
6.3.2.207 GetAxisParameter_PWM_MARK() . . . 165
6.3.2.208 GetAxisParameter_RAISE_ANGLE() . . . 165
6.3.2.209 GetAxisParameter_REG_INPUTS() . . . 166
6.3.2.210 GetAxisParameter_REG_POS() . . . 166
6.3.2.211 GetAxisParameter_REG_POSB() . . . 166
6.3.2.212 GetAxisParameter_REGIST_CONTROL() . . . 167
6.3.2.213 GetAxisParameter_REGIST_DELAY() . . . 167
6.3.2.214 GetAxisParameter_REGIST_SPEED() . . . 168
6.3.2.215 GetAxisParameter_REGIST_SPEEDB() . . . 168
6.3.2.216 GetAxisParameter_REMAIN() . . . 169
6.3.2.217 GetAxisParameter_REP_DIST() . . . 169
6.3.2.218 GetAxisParameter_REP_OPTION() . . . 169
6.3.2.219 GetAxisParameter_REV_IN() . . . 170
6.3.2.220 GetAxisParameter_REV_JOG() . . . 170
6.3.2.221 GetAxisParameter_REV_START() . . . 171
6.3.2.222 GetAxisParameter_ROBOT_DPOS() . . . 171
6.3.2.223 GetAxisParameter_ROBOT_FE() . . . 171
6.3.2.224 GetAxisParameter_ROBOT_FE_LATCH() . . . 172
6.3.2.225 GetAxisParameter_ROBOT_FE_RANGE() . . . 172
6.3.2.226 GetAxisParameter_ROBOT_FS_LIMIT() . . . 173
6.3.2.227 GetAxisParameter_ROBOT_RS_LIMIT() . . . 173
6.3.2.228 GetAxisParameter_ROBOT_SP_MODE() . . . 174
6.3.2.229 GetAxisParameter_ROBOT_UNITS() . . . 174
6.3.2.230 GetAxisParameter_ROBOTSTATUS() . . . 174
6.3.2.231 GetAxisParameter_RS_LIMIT() . . . 175

6.3.2.232 GetAxisParameter_S_REF() . . . 175
6.3.2.233 GetAxisParameter_S_REF_OUT() . . . 176
6.3.2.234 GetAxisParameter_SERVO() . . . 176
6.3.2.235 GetAxisParameter_SLOT_NUMBER() . . . 176
6.3.2.236 GetAxisParameter_SPEED() . . . 177
6.3.2.237 GetAxisParameter_SRAMP() . . . 177
6.3.2.238 GetAxisParameter_START_DIR_LAST() . . . 178
6.3.2.239 GetAxisParameter_STARTMOVE_SPEED() . . . 178
6.3.2.240 GetAxisParameter_STOP_ANGLE() . . . 179
6.3.2.241 GetAxisParameter_SYNC_AXIS() . . . 179
6.3.2.242 GetAxisParameter_SYNC_CONTROL() . . . 179
6.3.2.243 GetAxisParameter_SYNC_DPOS() . . . 180
6.3.2.244 GetAxisParameter_SYNC_DWELL() . . . 180
6.3.2.245 GetAxisParameter_SYNC_FLIGHT() . . . 181
6.3.2.246 GetAxisParameter_SYNC_PAUSE() . . . 181
6.3.2.247 GetAxisParameter_SYNC_TIME() . . . 181
6.3.2.248 GetAxisParameter_SYNC_TIMER() . . . 182
6.3.2.249 GetAxisParameter_SYNC_WITHDRAW() . . . 182
6.3.2.250 GetAxisParameter_T_REF() . . . 183
6.3.2.251 GetAxisParameter_T_REF_OUT() . . . 183
6.3.2.252 GetAxisParameter_TABLE_POINTER() . . . 184
6.3.2.253 GetAxisParameter_TANG_DIRECTION() . . . 184
6.3.2.254 GetAxisParameter_TCP_DPOS() . . . 184
6.3.2.255 GetAxisParameter_TCP_FS_LIMIT() . . . 185
6.3.2.256 GetAxisParameter_TCP_RS_LIMIT() . . . 185
6.3.2.257 GetAxisParameter_TCP_UNITS() . . . 186
6.3.2.258 GetAxisParameter_UNITS() . . . 186
6.3.2.259 GetAxisParameter_VECTOR_BUFFERED() . . . 186
6.3.2.260 GetAxisParameter_VERIFY() . . . 187
6.3.2.261 GetAxisParameter_VFF_GAIN() . . . 187
6.3.2.262 GetAxisParameter_VP_ACCEL() . . . 188
6.3.2.263 GetAxisParameter_VP_JERK() . . . 188
6.3.2.264 GetAxisParameter_VP_MODE() . . . 189
6.3.2.265 GetAxisParameter_VP_POSITION() . . . 189
6.3.2.266 GetAxisParameter_VP_SPEED() . . . 189
6.3.2.267 GetAxisParameter_WORLD_ACCEL() . . . 190
6.3.2.268 GetAxisParameter_WORLD_DECEL() . . . 190
6.3.2.269 GetAxisParameter_WORLD_DPOS() . . . 191
6.3.2.270 GetAxisParameter_WORLD_FS_LIMIT() . . . 191
6.3.2.271 GetAxisParameter_WORLD_JOGSPEED() . . . 191
6.3.2.272 GetAxisParameter_WORLD_RS_LIMIT() . . . 192
6.3.2.273 GetAxisParameter_WORLD_SPEED() . . . 192

6.3.2.274 GetAxisParameter_WORLD_UNITS() . . . 193
6.3.2.275 GetDIn() [1/2] . . . 193
6.3.2.276 GetDIn() [2/2] . . . 194
6.3.2.277 GetDIn32() . . . 194
6.3.2.278 GetDOut() [1/2] . . . 194
6.3.2.279 GetDOut() [2/2] . . . 195
6.3.2.280 GetDOut32() . . . 195
6.3.2.281 GetFirmwareVersionNumber() . . . 196
6.3.2.282 GetFrameGroup() . . . 196
6.3.2.283 GetInvertIN() . . . 196
6.3.2.284 GetMultiTableValues() . . . 197
6.3.2.285 GetMultiVrValues() . . . 197
6.3.2.286 GetProcessParameter_ERROR_LINE() . . . 198
6.3.2.287 GetProcessParameter_INDEVICE() . . . 198
6.3.2.288 GetProcessParameter_OUTDEVICE() . . . 198
6.3.2.289 GetProcessParameter_PMOVE() . . . 199
6.3.2.290 GetProcessParameter_PROC_LINE() . . . 199
6.3.2.291 GetProcessParameter_PROC_STATUS() . . . 200
6.3.2.292 GetProcessParameter_RUN_ERROR() . . . 200
6.3.2.293 GetProcessParameter_TICKS() . . . 201
6.3.2.294 GetSlotParameter_SERIAL_NUMBER() . . . 201
6.3.2.295 GetSystemParameter_ADDRESS() . . . 201
6.3.2.296 GetSystemParameter_AIN0() . . . 202
6.3.2.297 GetSystemParameter_AIN1() . . . 202
6.3.2.298 GetSystemParameter_AIN2() . . . 202
6.3.2.299 GetSystemParameter_AIN3() . . . 203
6.3.2.300 GetSystemParameter_AINBI0() . . . 203
6.3.2.301 GetSystemParameter_AINBI1() . . . 203
6.3.2.302 GetSystemParameter_AINBI2() . . . 203
6.3.2.303 GetSystemParameter_AINBI3() . . . 204
6.3.2.304 GetSystemParameter_AOUT0() . . . 204
6.3.2.305 GetSystemParameter_AOUT1() . . . 204
6.3.2.306 GetSystemParameter_AOUT2() . . . 205
6.3.2.307 GetSystemParameter_AOUT3() . . . 205
6.3.2.308 GetSystemParameter_AUTO_ETHERCAT() . . . 205
6.3.2.309 GetSystemParameter_AVERAGE_JITTER() . . . 206
6.3.2.310 GetSystemParameter_CONTROL() . . . 206
6.3.2.311 GetSystemParameter_CPU_EXCEPTIONS() . . . 206
6.3.2.312 GetSystemParameter_CURRENT_JITTER() . . . 206
6.3.2.313 GetSystemParameter_DISPLAY() . . . 207
6.3.2.314 GetSystemParameter_ERROR_AXIS() . . . 207
6.3.2.315 GetSystemParameter_FLASH_STATUS() . . . 207

6.3.2.316 GetSystemParameter_FPU_EXCEPTIONS() . . . 208
6.3.2.317 GetSystemParameter_HMI_PROC() . . . 208
6.3.2.318 GetSystemParameter_INPUTS0() . . . 208
6.3.2.319 GetSystemParameter_INPUTS1() . . . 209
6.3.2.320 GetSystemParameter_IP_ADDRESS() . . . 209
6.3.2.321 GetSystemParameter_IP_GATEWAY() . . . 209
6.3.2.322 GetSystemParameter_IP_MAC() . . . 209
6.3.2.323 GetSystemParameter_IP_MEMORY_CONFIG() . . . 210
6.3.2.324 GetSystemParameter_IP_NETMASK() . . . 210
6.3.2.325 GetSystemParameter_IP_PROTOCOL_CONFIG() . . . 210
6.3.2.326 GetSystemParameter_IP_PROTOCOL_CTRL() . . . 211
6.3.2.327 GetSystemParameter_IP_TCP_TIMEOUT() . . . 211
6.3.2.328 GetSystemParameter_IP_TCP_TX_THRESHOLD() . . . 211
6.3.2.329 GetSystemParameter_IP_TCP_TX_TIMEOUT() . . . 212
6.3.2.330 GetSystemParameter_LAST_AXIS() . . . 212
6.3.2.331 GetSystemParameter_LIMIT_BUFFERED() . . . 212
6.3.2.332 GetSystemParameter_MAX_JITTER() . . . 212
6.3.2.333 GetSystemParameter_MISSED_INTERRUPTS() . . . 213
6.3.2.334 GetSystemParameter_MISSED_RECV_FRAMES() . . . 213
6.3.2.335 GetSystemParameter_MISSED_SEND_FRAMES() . . . 213
6.3.2.336 GetSystemParameter_MODBUS_TX_DELAY() . . . 214
6.3.2.337 GetSystemParameter_MOTION_ERROR() . . . 214
6.3.2.338 GetSystemParameter_MPE_MODE() . . . 214
6.3.2.339 GetSystemParameter_N_ANA_IN() . . . 215
6.3.2.340 GetSystemParameter_N_ANA_OUT() . . . 215
6.3.2.341 GetSystemParameter_NIN() . . . 215
6.3.2.342 GetSystemParameter_NIO() . . . 215
6.3.2.343 GetSystemParameter_NOP() . . . 216
6.3.2.344 GetSystemParameter_PCMCAT_PROC() . . . 216
6.3.2.345 GetSystemParameter_PLC_CONFIG() . . . 216
6.3.2.346 GetSystemParameter_PLC_ERROR() . . . 217
6.3.2.347 GetSystemParameter_PLC_OVERFLOW() . . . 217
6.3.2.348 GetSystemParameter_PLC_RUN() . . . 217
6.3.2.349 GetSystemParameter_PROCNUMBER() . . . 218
6.3.2.350 GetSystemParameter_REMOTE_PROC() . . . 218
6.3.2.351 GetSystemParameter_SCHEDULE_TYPE() . . . 218
6.3.2.352 GetSystemParameter_SCOPE_CYCLE_COUNT() . . . 218
6.3.2.353 GetSystemParameter_SCOPE_DELAY() . . . 219
6.3.2.354 GetSystemParameter_SCOPE_POS() . . . 219
6.3.2.355 GetSystemParameter_SCOPE_TRIGGER_POS() . . . 219
6.3.2.356 GetSystemParameter_SERIAL_NUMBER() . . . 220
6.3.2.357 GetSystemParameter_SERVO_OFFSET() . . . 220

6.3.2.358 GetSystemParameter_SERVO_PERIOD() . . . 220
6.3.2.359 GetSystemParameter_SLOT0_TIME() . . . 221
6.3.2.360 GetSystemParameter_SLOT1_TIME() . . . 221
6.3.2.361 GetSystemParameter_SLOT2_TIME() . . . 221
6.3.2.362 GetSystemParameter_SLOT3_TIME() . . . 221
6.3.2.363 GetSystemParameter_SYSTEM_ERROR() . . . 222
6.3.2.364 GetSystemParameter_SYSTEM_ERROR_MASK() . . . 222
6.3.2.365 GetSystemParameter_SYSTEM_LOAD() . . . 222
6.3.2.366 GetSystemParameter_SYSTEM_LOAD_MAX() . . . 223
6.3.2.367 GetSystemParameter_TEXT_FILE_LOADER_PROC() . . . 223
6.3.2.368 GetSystemParameter_TSIZE() . . . 223
6.3.2.369 GetSystemParameter_UNIT_ERROR() . . . 224
6.3.2.370 GetSystemParameter_VERSION() . . . 224
6.3.2.371 GetSystemParameter_VRSIZE() . . . 224
6.3.2.372 GetSystemParameter_WDOG() . . . 224
6.3.2.373 GetTableValue() . . . 225
6.3.2.374 GetTcpHostName() . . . 225
6.3.2.375 GetTcpPort() . . . 226
6.3.2.376 GetTcpTimeout() . . . 226
6.3.2.377 GetVrString() [1/3] . . . 226
6.3.2.378 GetVrString() [2/3] . . . 227
6.3.2.379 GetVrString() [3/3] . . . 227
6.3.2.380 GetVrStringLen() . . . 227
6.3.2.381 GetVrValue() . . . 228
6.3.2.382 MoveAbs() . . . 228
6.3.2.383 MoveAbsMulti() . . . 229
6.3.2.384 MoveAbsSeq() . . . 229
6.3.2.385 MoveAbsSp() . . . 230
6.3.2.386 MoveAbsSpMulti() . . . 230
6.3.2.387 MoveCirc() . . . 230
6.3.2.388 MoveCircSp() . . . 231
6.3.2.389 MoveHelical() . . . 232
6.3.2.390 MoveHelicalSp() . . . 232
6.3.2.391 MoveLink() . . . 233
6.3.2.392 MoveModify() . . . 235
6.3.2.393 MovePick() . . . 235
6.3.2.394 MoveRel() . . . 236
6.3.2.395 MoveRelMulti() . . . 236
6.3.2.396 MoveRelSeq() . . . 237
6.3.2.397 MoveRelSp() . . . 237
6.3.2.398 MoveRelSpMulti() . . . 238
6.3.2.399 MoveSpherical() . . . 238

6.3.2.400 MoveSphericalSp() . . . 239
6.3.2.401 New() . . . 240
6.3.2.402 OpenConnection() . . . 240
6.3.2.403 PSwitch() . . . 241
6.3.2.404 PSwitchMulti() . . . 241
6.3.2.405 PSwitchOff() . . . 242
6.3.2.406 RapidStop() . . . 242
6.3.2.407 Reverse() . . . 242
6.3.2.408 Select() . . . 244
6.3.2.409 SetAOut() . . . 244
6.3.2.410 SetAxisBase() . . . 245
6.3.2.411 SetAxisParameter_ACCEL() . . . 245
6.3.2.412 SetAxisParameter_AFF_GAIN() . . . 245
6.3.2.413 SetAxisParameter_ATYPE() . . . 246
6.3.2.414 SetAxisParameter_AXIS_ACCEL() . . . 246
6.3.2.415 SetAxisParameter_AXIS_ADDRESS() . . . 246
6.3.2.416 SetAxisParameter_AXIS_DECEL() . . . 247
6.3.2.417 SetAxisParameter_AXIS_DISPLAY() . . . 247
6.3.2.418 SetAxisParameter_AXIS_ENABLE() . . . 248
6.3.2.419 SetAxisParameter_AXIS_ENABLE_OVERRIDE() . . . 248
6.3.2.420 SetAxisParameter_AXIS_ERROR_COUNT() . . . 248
6.3.2.421 SetAxisParameter_AXIS_FS_LIMIT() . . . 249
6.3.2.422 SetAxisParameter_AXIS_MAX_JOG_SPEED() . . . 249
6.3.2.423 SetAxisParameter_AXIS_MODE() . . . 249
6.3.2.424 SetAxisParameter_AXIS_RS_LIMIT() . . . 250
6.3.2.425 SetAxisParameter_AXIS_SPEED() . . . 250
6.3.2.426 SetAxisParameter_AXIS_UNITS() . . . 251
6.3.2.427 SetAxisParameter_AXIS_Z_OUTPUT() . . . 251
6.3.2.428 SetAxisParameter_BACKLASH_DIST() . . . 251
6.3.2.429 SetAxisParameter_CHANGE_DIR_LAST() . . . 252
6.3.2.430 SetAxisParameter_CLOSE_WIN() . . . 252
6.3.2.431 SetAxisParameter_CLUTCH_RATE() . . . 252
6.3.2.432 SetAxisParameter_CORNER_MODE() . . . 253
6.3.2.433 SetAxisParameter_CORNER_STATE() . . . 253
6.3.2.434 SetAxisParameter_CREEP() . . . 254
6.3.2.435 SetAxisParameter_D_GAIN() . . . 254
6.3.2.436 SetAxisParameter_DAC() . . . 254
6.3.2.437 SetAxisParameter_DAC_SCALE() . . . 255
6.3.2.438 SetAxisParameter_DATUM_IN() . . . 255
6.3.2.439 SetAxisParameter_DECEL() . . . 255
6.3.2.440 SetAxisParameter_DECEL_ANGLE() . . . 256
6.3.2.441 SetAxisParameter_DRIVE_CONTROL() . . . 256

6.3.2.442 SetAxisParameter_DRIVE_CONTROLWORD() . . . 257
6.3.2.443 SetAxisParameter_DRIVE_CW_MODE() . . . 257
6.3.2.444 SetAxisParameter_DRIVE_ENABLE() . . . 257
6.3.2.445 SetAxisParameter_DRIVE_FE_LIMIT() . . . 258
6.3.2.446 SetAxisParameter_DRIVE_INDEX() . . . 258
6.3.2.447 SetAxisParameter_DRIVE_MODE() . . . 258
6.3.2.448 SetAxisParameter_DRIVE_NEG_TORQUE() . . . 259
6.3.2.449 SetAxisParameter_DRIVE_PARAMETER() . . . 259
6.3.2.450 SetAxisParameter_DRIVE_POS_TORQUE() . . . 260
6.3.2.451 SetAxisParameter_DRIVE_PROFILE() . . . 260
6.3.2.452 SetAxisParameter_DRIVE_SET_VAL() . . . 260
6.3.2.453 SetAxisParameter_DRIVE_TYPE() . . . 261
6.3.2.454 SetAxisParameter_DRIVE_VALUE() . . . 261
6.3.2.455 SetAxisParameter_ENCODER_BITS() . . . 261
6.3.2.456 SetAxisParameter_ENCODER_CONTROL() . . . 262
6.3.2.457 SetAxisParameter_ENCODER_FILTER() . . . 262
6.3.2.458 SetAxisParameter_ENCODER_ID() . . . 263
6.3.2.459 SetAxisParameter_ENCODER_STATUS() . . . 263
6.3.2.460 SetAxisParameter_ENCODER_TURNS() . . . 263
6.3.2.461 SetAxisParameter_END_DIR_LAST() . . . 264
6.3.2.462 SetAxisParameter_ENDMOVE() . . . 264
6.3.2.463 SetAxisParameter_ENDMOVE_BUFFER() . . . 264
6.3.2.464 SetAxisParameter_ENDMOVE_SPEED() . . . 265
6.3.2.465 SetAxisParameter_ERRORMASK() . . . 265
6.3.2.466 SetAxisParameter_FAST_JOG() . . . 266
6.3.2.467 SetAxisParameter_FASTDEC() . . . 266
6.3.2.468 SetAxisParameter_FE_LIMIT() . . . 266
6.3.2.469 SetAxisParameter_FE_LIMIT_MODE() . . . 267
6.3.2.470 SetAxisParameter_FE_RANGE() . . . 267
6.3.2.471 SetAxisParameter_FEED_OVERRIDE() . . . 267
6.3.2.472 SetAxisParameter_FHOLD_IN() . . . 268
6.3.2.473 SetAxisParameter_FHSPEED() . . . 268
6.3.2.474 SetAxisParameter_FORCE_ACCEL() . . . 269
6.3.2.475 SetAxisParameter_FORCE_DECEL() . . . 269
6.3.2.476 SetAxisParameter_FORCE_DWELL() . . . 269
6.3.2.477 SetAxisParameter_FORCE_JERK() . . . 270
6.3.2.478 SetAxisParameter_FORCE_RAMP() . . . 270
6.3.2.479 SetAxisParameter_FORCE_SPEED() . . . 270
6.3.2.480 SetAxisParameter_FRAME() . . . 271
6.3.2.481 SetAxisParameter_FRAME_REP_DIST() . . . 271
6.3.2.482 SetAxisParameter_FRAME_SCALE() . . . 272
6.3.2.483 SetAxisParameter_FS_LIMIT() . . . 272

6.3.2.484 SetAxisParameter_FULL_SP_RADIUS() . . . 272
6.3.2.485 SetAxisParameter_FWD_IN() . . . 273
6.3.2.486 SetAxisParameter_FWD_JOG() . . . 273
6.3.2.487 SetAxisParameter_FWD_START() . . . 273
6.3.2.488 SetAxisParameter_I_GAIN() . . . 274
6.3.2.489 SetAxisParameter_IN_POS() . . . 274
6.3.2.490 SetAxisParameter_IN_POS_DIST() . . . 275
6.3.2.491 SetAxisParameter_IN_POS_SPEED() . . . 275
6.3.2.492 SetAxisParameter_INTERP_FACTOR() . . . 275
6.3.2.493 SetAxisParameter_INVERT_STEP() . . . 276
6.3.2.494 SetAxisParameter_JERK() . . . 276
6.3.2.495 SetAxisParameter_JOGSPEED() . . . 276
6.3.2.496 SetAxisParameter_LOOKAHEAD_FACTOR() . . . 277
6.3.2.497 SetAxisParameter_MERGE() . . . 277
6.3.2.498 SetAxisParameter_MOVE_COUNT() . . . 278
6.3.2.499 SetAxisParameter_MOVE_PA() . . . 278
6.3.2.500 SetAxisParameter_MOVE_PA_CONT() . . . 278
6.3.2.501 SetAxisParameter_MOVE_PA_IDLE() . . . 279
6.3.2.502 SetAxisParameter_MOVE_PB() . . . 279
6.3.2.503 SetAxisParameter_MOVE_PB_CONT() . . . 279
6.3.2.504 SetAxisParameter_MOVE_PB_IDLE() . . . 280
6.3.2.505 SetAxisParameter_MOVELINK_MODIFY() . . . 280
6.3.2.506 SetAxisParameter_MSPEED_FILTER() . . . 281
6.3.2.507 SetAxisParameter_OFFPOS() . . . 281
6.3.2.508 SetAxisParameter_OPEN_WIN() . . . 281
6.3.2.509 SetAxisParameter_OUTLIMIT() . . . 282
6.3.2.510 SetAxisParameter_OV_GAIN() . . . 282
6.3.2.511 SetAxisParameter_P_GAIN() . . . 282
6.3.2.512 SetAxisParameter_POS_DELAY() . . . 283
6.3.2.513 SetAxisParameter_POSI_SEQ_DELAY() . . . 283
6.3.2.514 SetAxisParameter_POSI_SEQ_MODE() . . . 284
6.3.2.515 SetAxisParameter_PP_STEP() . . . 284
6.3.2.516 SetAxisParameter_PWM_CYCLE() . . . 284
6.3.2.517 SetAxisParameter_PWM_MARK() . . . 285
6.3.2.518 SetAxisParameter_RAISE_ANGLE() . . . 285
6.3.2.519 SetAxisParameter_REG_INPUTS() . . . 285
6.3.2.520 SetAxisParameter_REGIST_CONTROL() . . . 286
6.3.2.521 SetAxisParameter_REGIST_DELAY() . . . 286
6.3.2.522 SetAxisParameter_REP_DIST() . . . 287
6.3.2.523 SetAxisParameter_REP_OPTION() . . . 287
6.3.2.524 SetAxisParameter_REV_IN() . . . 287
6.3.2.525 SetAxisParameter_REV_JOG() . . . 288

6.3.2.526 SetAxisParameter_REV_START() . . . 288
6.3.2.527 SetAxisParameter_ROBOT_FE_RANGE() . . . 288
6.3.2.528 SetAxisParameter_ROBOT_FS_LIMIT() . . . 289
6.3.2.529 SetAxisParameter_ROBOT_RS_LIMIT() . . . 289
6.3.2.530 SetAxisParameter_ROBOT_SP_MODE() . . . 290
6.3.2.531 SetAxisParameter_ROBOT_UNITS() . . . 290
6.3.2.532 SetAxisParameter_RS_LIMIT() . . . 290
6.3.2.533 SetAxisParameter_S_REF() . . . 291
6.3.2.534 SetAxisParameter_S_REF_OUT() . . . 291
6.3.2.535 SetAxisParameter_SERVO() . . . 291
6.3.2.536 SetAxisParameter_SLOT_NUMBER() . . . 292
6.3.2.537 SetAxisParameter_SPEED() . . . 292
6.3.2.538 SetAxisParameter_SRAMP() . . . 293
6.3.2.539 SetAxisParameter_START_DIR_LAST() . . . 293
6.3.2.540 SetAxisParameter_STARTMOVE_SPEED() . . . 293
6.3.2.541 SetAxisParameter_STOP_ANGLE() . . . 294
6.3.2.542 SetAxisParameter_SYNC_AXIS() . . . 294
6.3.2.543 SetAxisParameter_SYNC_CONTROL() . . . 294
6.3.2.544 SetAxisParameter_SYNC_DPOS() . . . 295
6.3.2.545 SetAxisParameter_SYNC_DWELL() . . . 295
6.3.2.546 SetAxisParameter_SYNC_FLIGHT() . . . 296
6.3.2.547 SetAxisParameter_SYNC_PAUSE() . . . 296
6.3.2.548 SetAxisParameter_SYNC_TIME() . . . 296
6.3.2.549 SetAxisParameter_SYNC_WITHDRAW() . . . 297
6.3.2.550 SetAxisParameter_T_REF() . . . 297
6.3.2.551 SetAxisParameter_T_REF_OUT() . . . 297
6.3.2.552 SetAxisParameter_TABLE_POINTER() . . . 298
6.3.2.553 SetAxisParameter_TANG_DIRECTION() . . . 298
6.3.2.554 SetAxisParameter_TCP_FS_LIMIT() . . . 299
6.3.2.555 SetAxisParameter_TCP_RS_LIMIT() . . . 299
6.3.2.556 SetAxisParameter_TCP_UNITS() . . . 299
6.3.2.557 SetAxisParameter_UNITS() . . . 300
6.3.2.558 SetAxisParameter_VECTOR_BUFFERED() . . . 300
6.3.2.559 SetAxisParameter_VERIFY() . . . 300
6.3.2.560 SetAxisParameter_VFF_GAIN() . . . 301
6.3.2.561 SetAxisParameter_VP_MODE() . . . 301
6.3.2.562 SetAxisParameter_WORLD_ACCEL() . . . 302
6.3.2.563 SetAxisParameter_WORLD_DECEL() . . . 302
6.3.2.564 SetAxisParameter_WORLD_FS_LIMIT() . . . 302
6.3.2.565 SetAxisParameter_WORLD_JOGSPEED() . . . 303
6.3.2.566 SetAxisParameter_WORLD_RS_LIMIT() . . . 303
6.3.2.567 SetAxisParameter_WORLD_SPEED() . . . 303

6.3.2.568 SetAxisParameter_WORLD_UNITS() . . . 304
6.3.2.569 SetDOut() [1/2] . . . 304
6.3.2.570 SetDOut() [2/2] . . . 305
6.3.2.571 SetDOut32() . . . 305
6.3.2.572 SetFrameGroup() . . . 305
6.3.2.573 SetInvertIN() . . . 306
6.3.2.574 SetMotion() [1/4] . . . 306
6.3.2.575 SetMotion() [2/4] . . . 306
6.3.2.576 SetMotion() [3/4] . . . 307
6.3.2.577 SetMotion() [4/4] . . . 307
6.3.2.578 SetMotionSp() [1/8] . . . 308
6.3.2.579 SetMotionSp() [2/8] . . . 308
6.3.2.580 SetMotionSp() [3/8] . . . 308
6.3.2.581 SetMotionSp() [4/8] . . . 309
6.3.2.582 SetMotionSp() [5/8] . . . 309
6.3.2.583 SetMotionSp() [6/8] . . . 310
6.3.2.584 SetMotionSp() [7/8] . . . 311
6.3.2.585 SetMotionSp() [8/8] . . . 311
6.3.2.586 SetMultiAxisBase() . . . 312
6.3.2.587 SetMultiTableValues() . . . 312
6.3.2.588 SetMultiVrValues() . . . 313
6.3.2.589 SetProcessParameter_INDEVICE() . . . 313
6.3.2.590 SetProcessParameter_OUTDEVICE() . . . 313
6.3.2.591 SetProcessParameter_TICKS() . . . 314
6.3.2.592 SetSlotParameter_SERIAL_NUMBER() . . . 314
6.3.2.593 SetSystemParameter_ADDRESS() . . . 314
6.3.2.594 SetSystemParameter_AIN0() . . . 315
6.3.2.595 SetSystemParameter_AIN1() . . . 315
6.3.2.596 SetSystemParameter_AIN2() . . . 315
6.3.2.597 SetSystemParameter_AIN3() . . . 316
6.3.2.598 SetSystemParameter_AINBI0() . . . 316
6.3.2.599 SetSystemParameter_AINBI1() . . . 316
6.3.2.600 SetSystemParameter_AINBI2() . . . 317
6.3.2.601 SetSystemParameter_AINBI3() . . . 317
6.3.2.602 SetSystemParameter_AOUT0() . . . 317
6.3.2.603 SetSystemParameter_AOUT1() . . . 318
6.3.2.604 SetSystemParameter_AOUT2() . . . 318
6.3.2.605 SetSystemParameter_AOUT3() . . . 318
6.3.2.606 SetSystemParameter_AUTO_ETHERCAT() . . . 319
6.3.2.607 SetSystemParameter_AVERAGE_JITTER() . . . 319
6.3.2.608 SetSystemParameter_CPU_EXCEPTIONS() . . . 319
6.3.2.609 SetSystemParameter_CURRENT_JITTER() . . . 320

6.3.2.610 SetSystemParameter_DISPLAY() . . . 320
6.3.2.611 SetSystemParameter_FPU_EXCEPTIONS() . . . 320
6.3.2.612 SetSystemParameter_HMI_PROC() . . . 321
6.3.2.613 SetSystemParameter_IP_ADDRESS() . . . 321
6.3.2.614 SetSystemParameter_IP_GATEWAY() . . . 321
6.3.2.615 SetSystemParameter_IP_MAC() . . . 322
6.3.2.616 SetSystemParameter_IP_MEMORY_CONFIG() . . . 322
6.3.2.617 SetSystemParameter_IP_NETMASK() . . . 322
6.3.2.618 SetSystemParameter_IP_PROTOCOL_CONFIG() . . . 323
6.3.2.619 SetSystemParameter_IP_PROTOCOL_CTRL() . . . 323
6.3.2.620 SetSystemParameter_IP_TCP_TIMEOUT() . . . 323
6.3.2.621 SetSystemParameter_IP_TCP_TX_THRESHOLD() . . . 324
6.3.2.622 SetSystemParameter_IP_TCP_TX_TIMEOUT() . . . 324
6.3.2.623 SetSystemParameter_LAST_AXIS() . . . 324
6.3.2.624 SetSystemParameter_LIMIT_BUFFERED() . . . 325
6.3.2.625 SetSystemParameter_MAX_JITTER() . . . 325
6.3.2.626 SetSystemParameter_MISSED_INTERRUPTS() . . . 325
6.3.2.627 SetSystemParameter_MISSED_RECV_FRAMES() . . . 326
6.3.2.628 SetSystemParameter_MISSED_SEND_FRAMES() . . . 326
6.3.2.629 SetSystemParameter_MODBUS_TX_DELAY() . . . 326
6.3.2.630 SetSystemParameter_NIO() . . . 327
6.3.2.631 SetSystemParameter_PCMCAT_PROC() . . . 327
6.3.2.632 SetSystemParameter_PLC_CONFIG() . . . 327
6.3.2.633 SetSystemParameter_REMOTE_PROC() . . . 328
6.3.2.634 SetSystemParameter_SCHEDULE_TYPE() . . . 328
6.3.2.635 SetSystemParameter_SCOPE_CYCLE_COUNT() . . . 328
6.3.2.636 SetSystemParameter_SCOPE_DELAY() . . . 329
6.3.2.637 SetSystemParameter_SCOPE_POS() . . . 329
6.3.2.638 SetSystemParameter_SERIAL_NUMBER() . . . 329
6.3.2.639 SetSystemParameter_SERVO_OFFSET() . . . 330
6.3.2.640 SetSystemParameter_SERVO_PERIOD() . . . 330
6.3.2.641 SetSystemParameter_SLOT0_TIME() . . . 330
6.3.2.642 SetSystemParameter_SLOT1_TIME() . . . 331
6.3.2.643 SetSystemParameter_SLOT2_TIME() . . . 331
6.3.2.644 SetSystemParameter_SLOT3_TIME() . . . 331
6.3.2.645 SetSystemParameter_SYSTEM_ERROR_MASK() . . . 332
6.3.2.646 SetSystemParameter_SYSTEM_LOAD_MAX() . . . 332
6.3.2.647 SetSystemParameter_TEXT_FILE_LOADER_PROC() . . . 332
6.3.2.648 SetSystemParameter_WDOG() . . . 333
6.3.2.649 SetTableValue() . . . 333
6.3.2.650 SetTcpHostName() . . . 333
6.3.2.651 SetTcpPort() . . . 334

6.3.2.652 SetTcpTimeout() . . . 334
6.3.2.653 SetVrString() . . . 334
6.3.2.654 SetVrValue() . . . 335
6.3.2.655 StepRatio() . . . 335
6.3.2.656 StopProgram() [1/2] . . . 336
6.3.2.657 StopProgram() [2/2] . . . 336
6.3.2.658 UploadFile() . . . 336
6.3.3 Property Documentation . . . 337
6.3.3.1 ConnectionType . . . 337
6.3.3.2 IsConnected . . . 337
6.3.4 Event Documentation . . . 337
6.3.4.1 ConnectionEvent . . . 337
6.4 Trio.UnifiedApi.ProgressInfo Struct Reference . . . 338
6.4.1 Detailed Description . . . 338
6.4.2 Member Data Documentation . . . 338
6.4.2.1 current_pos . . . 338
6.4.2.2 evt . . . 338
6.4.2.3 total_length . . . 338
6.5 Trio.UnifiedApi.TrioConnectionException Class Reference . . . 339
6.5.1 Detailed Description . . . 339
6.5.2 Constructor & Destructor Documentation . . . 339
6.5.2.1 TrioConnectionException() . . . 339
6.5.3 Property Documentation . . . 340
6.5.3.1 ErrorCode . . . 340
6.5.3.2 Message . . . 340
6.6 Trio.UnifiedApi.TrioConnectionFactory Class Reference . . . 340
6.6.1 Detailed Description . . . 340
6.6.2 Member Function Documentation . . . 341
6.6.2.1 CreateConnectionFlex7() . . . 341
6.6.2.2 CreateConnectionPCMCAT() . . . 341
6.6.2.3 CreateConnectionTCP() . . . 341
6.6.3 Property Documentation . . . 342
6.6.3.1 ErrorStringGetter . . . 342

7 File Documentation **343**

7.1 ITrioConnection.cs File Reference . . . 343
7.1.1 Typedef Documentation . . . 353
7.1.1.1 size_t . . . 353
7.2 ITrioConnection.cs . . . 353
7.3 TrioConnectionFactory.cs File Reference . . . 457
7.4 TrioConnectionFactory.cs . . . 457

Index **459**

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Trio 9
Trio.UnifiedApi 9

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Trio.UnifiedApi.DataStreamPacket 43
Exception
    Trio.UnifiedApi.TrioConnectionException 339
Trio.UnifiedApi.FileInfo 44
IDisposable
    Trio.UnifiedApi.ITrioConnection 46
Trio.UnifiedApi.ProgressInfo 338
Trio.UnifiedApi.TrioConnectionFactory 340

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Trio.UnifiedApi.DataStreamPacket
: Data used with the delegate for streaming back the response of commands that return string data 43

Trio.UnifiedApi.FileInfo
: Data used with the delegate for processing the response of Dir command 44

Trio.UnifiedApi.ITrioConnection
: Connection interface 46

Trio.UnifiedApi.ProgressInfo
: Data used with the progress report 338

Trio.UnifiedApi.TrioConnectionException
: Exception raised when error occurs in communication library 339

Trio.UnifiedApi.TrioConnectionFactory
: Factory for creating communication instances 340

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

ITrioConnection.cs 343
TrioConnectionFactory.cs 457

Chapter 5

Namespace Documentation

5.1 Trio Namespace Reference

Namespaces

- namespace UnifiedApi

5.2 Trio.UnifiedApi Namespace Reference

Classes

- struct DataStreamPacket

  Data used with the delegate for streaming back the response of commands that return string data.
- struct FileInfo

  Data used with the delegate for processing the response of Dir command.
- interface ITrioConnection

  Connection interface.
- struct ProgressInfo

  Data used with the progress report.
- class TrioConnectionException

  Exception raised when error occurs in communication library.
- class TrioConnectionFactory

  Factory for creating communication instances.

Enumerations

- enum ErrorCode : UInt64 {
  NoError = 0 , _Unknown = 0xFFFFFFFF , _ApiError = 0x100000000 , Api_Unknown = (_ApiError + 0) ,
  Fail , NotSupported , ConnectionModule , ConnectionContext ,
  TokenTable , Timeout , UnknownFloatingPointFormat , Remote_TokenBuffer ,
  Remote_UnknownToken , UnsupportedController , BadResponse , BadArgument ,
  Reopen , BufferTooSmall , SocketRead , SocketWrite ,
  LowLevelError , VersionMismatch , Compression , Abort ,
  Disconnected , _ApiLast , _SystemError = 0x200000000 , Sys_Unknown = (_SystemError | _Unknown) ,
  _ControllerError = 0x400000000 , ControllerError_NONE = 0 | _ControllerError , ControllerError_COMMAND_NOT_RECOGNIS

= 1 | _ControllerError , ControllerError_INVALID_TRANSFER_TYPE = 2 | _ControllerError , ControllerError_FLASH_PROGRAMMING_ERROR = 3 | _ControllerError , ControllerError_OPERAND_EXPECTED = 4 | _ControllerError , ControllerError_ASSIGNMENT_EXPECTED = 5 | _ControllerError , ControllerError_QUOTES_EXPECTI = 6 | _ControllerError , ControllerError_STACK_OVERFLOW = 7 | _ControllerError , ControllerError_TOO_MANY_VARIABLES = 8 | _ControllerError , ControllerError_DIVISION_BY_ZERO = 9 | _ControllerError , ControllerError_EXTRA_CHARS_AT_EOLN = 10 | _ControllerError , ControllerError_CLOSE_SQUARE_BRACKET_EXPECTED_IN_PRINT = 11 | _ControllerError , ControllerError_CANNOT_MO = 12 | _ControllerError , ControllerError_THEN_EXPECTED_IN_IF_ELSEIF = 13 | _ControllerError , ControllerError_FLASH_ERASE_ERROR = 14 | _ControllerError , ControllerError_START_OF_EXPRESSION_EXPECTED = 15 | _ControllerError , ControllerError_CLOSE_ROUND_BRACKET = 16 | _ControllerError , ControllerError_COMMA_EXPECTED = 17 | _ControllerError , ControllerError_COMMAND_LINE_BRC = 18 | _ControllerError , ControllerError_PARAMETER_OUT_OF_RANGE = 19 | _ControllerError , ControllerError_NO_PROCESS_AVAILABLE = 20 | _ControllerError , ControllerError_VALUE_IS_READ_ONLY = 21 | _ControllerError , ControllerError_MODIFIER_NOT_AL = 22 | _ControllerError , ControllerError_REMOTE_AXIS_IS_IN_USE = 23 | _ControllerError , ControllerError_COMMAND_IS_COMMAND_LINE_ONL = 24 | _ControllerError , ControllerError_COMMAND_IS_RUNTIME_ONLY = 25 | _ControllerError , ControllerError_LABEL_EXPECTED = 26 | _ControllerError , ControllerError_PROGRAM_NOT_FOUND = 27 | _ControllerError , ControllerError_DUPLICATE_IDENTIFIER = 28 | _ControllerError , ControllerError_PROGRAM_IS_LOCKED = 29 | _ControllerError , ControllerError_PROGRAMS_RUNN = 30 | _ControllerError , ControllerError_PROGRAM_IS_STOPPED = 31 | _ControllerError , ControllerError_CANNOT_SELECT_PROGRAM = 32 | _ControllerError , ControllerError_NO_PROGRAM_SELECTED = 33 | _ControllerError , ControllerError_NO_MORE_PROGRAMS_AVAILABLE = 34 | _ControllerError , ControllerError_OUT_OF_MEMORY = 35 | _ControllerError , ControllerError_NO_CODE_AVAILABLE_FOR_PROGRAM = 36 | _ControllerError , ControllerError_COMMAND_OUT_OF_CONTEXT = 37 | _ControllerError , ControllerError_TOO_MANY_NESTED_STRUCTURES = 38 | _ControllerError , ControllerError_STRUCTURE_NESTING_ERROR = 39 | _ControllerError , ControllerError_ELSE_ELSEIF_ENDIF_WITHOUT_ = 40 | _ControllerError , ControllerError_WEND_WITHOUT_PREVIOUS_WHILE = 41 | _ControllerError , ControllerError_UNTIL_WITHOUT_PREVIOUS_REPEAT = 42 | _ControllerError , ControllerError_IDENTIFIER_EXPECTED = 43 | _ControllerError , ControllerError_TO_EXPECTED_AFTER_FOR = 44 | _ControllerError , ControllerError_TOO_MANY_NESTED_FOR_NEXT = 45 | _ControllerError , ControllerError_NEXT_WITHOUT_FOR = 46 | _ControllerError , ControllerError_UNTIL_IDLE_LOADED_EXPECTED_AFTER_WAIT = 47 | _ControllerError , ControllerError_GOTO_GOSUB_ = 48 | _ControllerError , ControllerError_TOO_MANY_NESTED_GOSUB = 49 | _ControllerError , ControllerError_RETURN_WITHOUT_GOSUB = 50 | _ControllerError , ControllerError_LABEL_MUST_BE_AT_START_OF_LINE = 51 | _ControllerError , ControllerError_CANNOT_NEST_ONE_LIN = 52 | _ControllerError , ControllerError_LABEL_NOT_FOUND = 53 | _ControllerError , ControllerError_LINE_NUMBER_CANN = 54 | _ControllerError , ControllerError_MAXIMUM_NUMBER_OF_PROGRAM_INSTANCES_ACTIVE = 55 | _ControllerError , ControllerError_INVALID_USE_OF_DOLAR_SIGN = 56 | _ControllerError , ControllerError_VR_X_OR_STRING_VARIABLE_E = 57 | _ControllerError , ControllerError_PROGRAM_ALREADY_EXISTS = 58 | _ControllerError , ControllerError_PROCESS_ALREADY_SELECTED = 59 | _ControllerError , ControllerError_DUPLICATE_AXES_NOT_PERMI = 60 | _ControllerError , ControllerError_PLC_TYPE_IS_INVALID = 61 | _ControllerError , ControllerError_EVALUATION_ERRO = 62 | _ControllerError , ControllerError_RESERVED_KEYWORD_NOT_AVAILABLE_ON_THIS_CONTROLLER = 63 | _Controller↩ Error , ControllerError_VARIABLE_NOT_FOUND = 64 | _ControllerError , ControllerError_TABLE_INDEX_RANGE_ERROR = 65 | _ControllerError , ControllerError_FEATURES_ENABLED_DO_NOT_ALLOW_ATYPE_CHANGE = 66 | _ControllerError , ControllerError_INVALID_LINE_NUMBER = 67 | _ControllerError , ControllerError_STRING_EXCEEDS_PERMITTED_LENGT = 68 | _ControllerError , ControllerError_SCOPE_PERIOD_SHOULD_EXCEED_NUMBER_OF_AIN_PARAMS = 69 | _ControllerError , ControllerError_VALUE_IS_INCORRECT = 70 | _ControllerError , ControllerError_INVALID_IO_CHANNEL = 71 | _ControllerError , ControllerError_FLASH_VALUE_CANNOT_BE_SET = 72 | _ControllerError , ControllerError_DIRECTORY_NOT_LOCKED = 73 | _ControllerError , ControllerError_DIRECTORY_ALREADY_LOCKED = 74 | _ControllerError , ControllerError_PROGRAM_NOT_RUNNING_ON_THIS_PROCESS = 75 | _ControllerError , ControllerError_PROGRAM_NOT = 76 | _ControllerError , ControllerError_PROGRAM_NOT_PAUSED_ON_THIS_PROCESS = 77 | _↩

ControllerError , ControllerError_PROGRAM_NOT_PAUSED = 78 | _ControllerError , ControllerError_COMMAND_NOT_ALLOWED_WHEN_RUNNING_MOTION_PERFECT = 79 | _ControllerError , ControllerError_DIRECTORY_STRUCTURE_INVALID = 80 | _ControllerError , ControllerError_DIRECTORY_IS_LOCKE = 81 | _ControllerError , ControllerError_CANNOT_EDIT_PROGRAM = 82 | _ControllerError , ControllerError_TOO_MANY_NESTED_OPERANDS = 83 | _ControllerError , ControllerError_CANNOT_RESET_WHEN_DRIV = 84 | _ControllerError , ControllerError_FLASH_STICK_BLANK = 85 | _ControllerError , ControllerError_FLASH_STICK_NOT_ = 86 | _ControllerError , ControllerError_SLAVE_ERROR = 87 | _ControllerError , ControllerError_MASTER_ERROR = 88 | _ControllerError , ControllerError_NETWORK_TIMEOUT = 89 | _ControllerError , ControllerError_NETWORK_PROTOCOL_ERR = 90 | _ControllerError , ControllerError_GLOBAL_DEFINITION_IS_DIFFERENT = 91 | _ControllerError , ControllerError_INVALID_PROGRAM_NAME = 92 | _ControllerError , ControllerError_PROGRAM_CORRUPT = 93 | _ControllerError , ControllerError_MORE_THAN_ONE_ = 94 | _ControllerError , ControllerError_PROGRAM_ENCRYPTED = 95 | _ControllerError , ControllerError_BASIC_TOKEN_DEFINITION_INCORREC = 96 | _ControllerError , ControllerError_OPEN_BRACKET_EXPECTED = 97 | _ControllerError , ControllerError_NUMBER_EXPECTED = 98 | _ControllerError , ControllerError_AS_EXPECTED = 99 | _ControllerError , ControllerError_DATA_TYPE_EXPECTED = 100 | _ControllerError , ControllerError_STRING_EXPECTED = 101 | _ControllerError , ControllerError_DOWNLOAD_ABORT_OR_ = 102 | _ControllerError , ControllerError_CANNOT_SPECIFY_PROGRAM_TYPE_FOR_AN_EXISTING_PROGRAM = 103 | _ControllerError , ControllerError_INVALID_COFF_IMAGE_FILE = 104 | _ControllerError , ControllerError_VARIABLE_DEFINED = 105 | _ControllerError , ControllerError_COMMAND_NOT_ALLOWED_WITHIN_INCLUDE_FILE = 106 | _ControllerError , ControllerError_SERIAL_NUMBER_MUST_BE_MINUS_1 = 107 | _ControllerError , ControllerError_APPEND_BLOCK_INCON = 108 | _ControllerError , ControllerError_INVALID_RANGE_SPECIFIED = 109 | _ControllerError , ControllerError_TOO_MANY_ITEMS_DEFINED_FOR_BLOCK = 110 | _ControllerError , ControllerError_INVALID_MSPHERICAL_INPUT = 111 | _ControllerError , ControllerError_TOO_MANY_LABELS = 112 | _ControllerError , ControllerError_SYMBOL_TABLE_LOCKED = 113 | _ControllerError , ControllerError_INCORRECT_SYMBOL_TYPE = 114 | _ControllerError , ControllerError_VARIABLES_NOT_PERMITTED_ON_COMMAND_LINE = 115 | _ControllerError , ControllerError_INVALID_PROGRAM_TYPE = 116 | _ControllerError , ControllerError_PARAMETER_EXPECTED = 117 | _ControllerError , ControllerError_FIRMWARE_ERROR__DEVICE_IN_USE = 118 | _ControllerError , ControllerError_DEVICE_ERROR__TIMEOUT_WAITING_FOR_DEVICE = 119 | _ControllerError , ControllerError_DEVICE_ERROR__COMMAND_NOT_SUPPORTED_BY_DEVICE = 120 | _ControllerError , ControllerError_DEVICE_ERROR__CRC_ERROR = 121 | _ControllerError , ControllerError_DEVICE_ERROR__ERROR_WR = 122 | _ControllerError , ControllerError_DEVICE_ERROR__INVALID_RESPONSE_FROM_DEVICE = 123 | _ControllerError , ControllerError_FIRMWARE_ERROR__CANNOT_REFERENCE_DATA_OUTSIDE_CURRENT_BLOCK = 124 | _ControllerError , ControllerError_DISK_ERROR__INVALID_MBR = 125 | _ControllerError , ControllerError_DISK_ERROR__INVALID_BOOT_SECTOR = 126 | _ControllerError , ControllerError_DISK_ERROR__INVALID_SECTOR_CLUSTER_REFERENCE = 127 | _ControllerError , ControllerError_FILE_ERROR__DISK_FULL = 128 | _ControllerError , ControllerError_FILE_ERROR__FILE_NOT_FOUND = 129 | _ControllerError , ControllerError_FILE_ERROR__FILENAME_ALREADY_EXISTS = 130 | _ControllerError , ControllerError_FILE_ERROR__INVALID_FILENAME = 131 | _ControllerError , ControllerError_FILE_ERROR__DIRECTORY_ = 132 | _ControllerError , ControllerError_COMMAND_ONLY_ALLOWED_WHEN_RUNNING_MOTION_PERFECT = 133 | _ControllerError , ControllerError_HASH_EXPECTED = 134 | _ControllerError , ControllerError_FOR_EXPECTED = 135 | _ControllerError , ControllerError_INPUT__OUTPUT__APPEND__FIFO_READ__FI = 136 | _ControllerError , ControllerError_FILE_NOT_OPEN = 137 | _ControllerError , ControllerError_END_OF_FILE = 138 | _ControllerError , ControllerError_FILE_ALREADY_OPEN = 139 | _ControllerError , ControllerError_INVALID_STORAGE_AREA = 140 | _ControllerError , ControllerError_INVALID_FLOATING_POINT_OPERATION = 141 | _ControllerError , ControllerError_INVALID_SYSTEM_CODE_WRONG_CONTROLLER = 142 | _ControllerError , ControllerError_IEC_ERROR__INVALID_VARIABLE_ACCESS = 143 | _ControllerError , ControllerError_NUMERICAL_ERROR = 144 | _ControllerError , ControllerError_NUMERICAL_ERROR__INF_USED = 145 | _ControllerError , ControllerError_NUMERICAL_ERROR__SUBNORMAL_VALUE_USED = 146 | _ControllerError , ControllerError_MAC_EEPROM_IS_LOCKED = 147 | _ControllerError , ControllerError_INVALID_MIX_OF_DATA_TYPES

= 148 | _ControllerError , ControllerError_INVALID_STARTUP_CONFIGURATION_COMMAND = 149 | _↩
ControllerError , ControllerError_SYMBOL_IS_NOT_A_VARIABLE = 150 | _ControllerError ,
ControllerError_KINEMATICS_RUNTIME_LIMITED_TO_1_HOUR_FEC_22 = 151 | _ControllerError ,
ControllerError_IEC_RUNTIME_LIMITED_TO_1_HOUR_FEC_21 = 152 | _ControllerError , ControllerError_COMMAND_NOT_
= 153 | _ControllerError , ControllerError_WILDCARD_LENGTH_MUST_BE_1 = 154 | _ControllerError ,
ControllerError_INCOMPATIBLE_ARRAY_DIMENSIONS = 155 | _ControllerError , ControllerError_MATRIX_IS_SINGULAR
= 156 | _ControllerError , ControllerError_PROGRAM_IS_NOT_EXECUTABLE_TYPE = 157 | _Controller↩
Error , ControllerError_DISK_ERROR__FORMAT_MUST_BE_FAT32 = 158 | _ControllerError ,
ControllerError_PROGRAM_IS_STOPPED_HALT_FORCED = 159 | _ControllerError , ControllerError_VARIABLE_NOT_DEFIN
= 160 | _ControllerError , ControllerError_VARIABLE_NOT_DEFINED_USING_DIM_WITHIN_INCLUDE_FILE
= 161 | _ControllerError , ControllerError_PARAMETER_CAN_ONLY_BE_CHANGED_USING_MC_CONFIG
= 162 | _ControllerError ,
ControllerError_IEC_TASK_MEMORY_ALLOCATION_ERROR = 163 | _ControllerError , ControllerError_IEC_IO_BINDING_IN
= 164 | _ControllerError , ControllerError_IEC_VR_BINDING_INDEX_OUT_OF_RANGE = 165 | _↩
ControllerError , ControllerError_IEC_TABLE_BINDING_INDEX_OUT_OF_RANGE = 166 | _ControllerError ,
ControllerError_FAILED_OPEN_CONNECTION = 167 | _ControllerError , ControllerError_MCCONFIG_CHANGED_EX_REQU
= 168 | _ControllerError , ControllerError_NO_RESPONSE_FROM_CAN_MODULE = 169 | _ControllerError
, ControllerError_INVALID_FPGA_VERSION = 170 | _ControllerError ,
ControllerError_VARIABLE_NOT_CREATED_USING_GLOBAL_CONSTANT = 171 | _ControllerError ,
ControllerError_PARAMETER_CAN_ONLY_BE_CHANGED_USING_MC_CONFIG_OR_COMMAND_LINE
= 172 | _ControllerError , ControllerError_ETHERCAT_AXES_CANNOT_BE_SET_USING_ATYPE_PLEASE_USE_DRIVE_MC
= 173 | _ControllerError , ControllerError_BIT_NUMBER_OUT_OF_RANGE = 174 | _ControllerError ,
ControllerError_FUNCTION_NAME_EXPECTED = 175 | _ControllerError , ControllerError_FUNCTION_MUST_BE_DEFINED_
= 176 | _ControllerError , ControllerError_TOO_MANY_FUNCTIONS_DEFINED = 177 | _ControllerError ,
ControllerError_MISSING_ENDFUNC = 178 | _ControllerError ,
ControllerError_ENDFUNC_WITHOUT_PREVIOUS_FUNCTION = 179 | _ControllerError , ControllerError_PARAMETER_NAM
= 180 | _ControllerError , ControllerError_DEVICE_ERROR_NO_SDCARD_INSERTED = 181 | _Controller↩
Error , ControllerError_DEVICE_ERROR_SDCARD_WRITE_PROTECTED = 182 | _ControllerError ,
ControllerError_FAILED_ATTEMPT_1 = 183 | _ControllerError , ControllerError_FAILED_ATTEMPT_2 = 184
| _ControllerError , ControllerError_FAILED_TOO_MANY_TIMES_PLEASE_RESET = 185 | _ControllerError
, ControllerError_LINE_MUST_BE_INSIDE_FUNCTION = 186 | _ControllerError ,
ControllerError_DUPLICATE_NAME_USED_IN_LABEL_AND_VARIABLE = 187 | _ControllerError ,
ControllerError_STRUCTURE_FIELD_NAME_NOT_RECOGNISED = 188 | _ControllerError , ControllerError_ID_NOT_DEFINE
= 189 | _ControllerError , ControllerError_ROBOT_CONFIGURATION_CANNOT_BE_CHANGED_WHILE_MOVEL
= 190 | _ControllerError ,
ControllerError_TOOL_OFFSET_ID_NOT_DEFINED = 191 | _ControllerError , ControllerError_OBJECT_FRAME_ID_NOT_DE
= 192 | _ControllerError , ControllerError_ROBOT_FRAME_ID_NOT_DEFINED = 193 | _ControllerError ,
ControllerError_COMMAND_NOT_PERMITTED_IN_FUNCTION = 194 | _ControllerError ,
ControllerError_TOO_MANY_FUNCTION_PARAMETERS = 195 | _ControllerError , ControllerError_OPERATION_CANNOT_B
= 196 | _ControllerError , ControllerError_SINGULARITY = 197 | _ControllerError , ControllerError_TOOL_CANNOT_BE_MODI
= 198 | _ControllerError ,
ControllerError_OBJECT_FRAME_CANNOT_BE_MODIFIED_OR_DELETED_WHILE_SELECTED = 199 |
_ControllerError , ControllerError_ROBOT_FRAME_CANNOT_BE_MODIFIED_OR_DELETED_WHILE_SELECTED
= 200 | _ControllerError , ControllerError_ROBOT_MOVES_CANNOT_BE_PERFORMED_UNTIL_JOG_OPERATION_IS_DISA
= 201 | _ControllerError , ControllerError_CYCLIC_INCLUDE_DEPENDENCY = 202 | _ControllerError ,
ControllerError_OBJECT_FRAME_ALREADY_ATTACHED = 203 | _ControllerError , ControllerError_OBJECT_FRAME_HAS_N
= 204 | _ControllerError , ControllerError_OBJECT_FRAME_ALREADY_SYNCED = 205 | _ControllerError ,
ControllerError_BASE_AXIS_DOES_NOT_BELONG_TO_ANY_KINEMATIC_GROUP = 206 | _Controller↩
Error ,
ControllerError_SYNC_IN_PROCESS = 207 | _ControllerError , ControllerError_ROBOT_PROGRAM_STOPPED
= 208 | _ControllerError , ControllerError_TARGET_POINT_NOT_ACTIVATED = 209 | _ControllerError ,
ControllerError_PROGRAM_FAILED_TO_RUN = 210 | _ControllerError ,
ControllerError_RETURN_STATEMENT_EXPECTED_BEFORE_ENDFUNC = 211 | _ControllerError ,
ControllerError_CONTINUE_WITHOUT_PREVIOUS_FOR_WHILE_OR_REPEAT = 212 | _Controller↩
Error , ControllerError_EXIT_WITHOUT_PREVIOUS_FOR_WHILE_OR_REPEAT = 213 | _ControllerError ,
ControllerError_TOO_MANY_EXIT_STATEMENTS_WITHIN_SINGLE_BLOCK = 214 | _ControllerError ,
ControllerError_CASE_OR_END_CASE_WITHOUT_PREVIOUS_SELECT_CASE = 215 | _ControllerError ,
ControllerError_FIXED_INTEGER_OR_ELSE_EXPECTED = 216 | _ControllerError , ControllerError_TOO_MANY_CASE_OPT

= 217 | _ControllerError , ControllerError_ROBOTICS_RUNTIME_LIMITED_TO_1_HOUR = 218 | _↩
ControllerError ,
ControllerError_COLLISION_OBJECTS_CANNOT_BE_MODIFIED_OR_DELETED_WHILE_SELECTED =
219 | _ControllerError , ControllerError_TOO_MANY_COLLISION_OBJECTS_ACTIVE = 220 | _Controller↩
Error , ControllerError_ATYPE_MUST_BE_CONFIGURED_USING_MC_CONFIG_PARAMETER_DRIVE_MODE
= 221 | _ControllerError , ControllerError_COMMAND_NOT_ALLOWED_IN_FRAME = 222 | _ControllerError
,
ControllerError_COMMAND_LINE_TOO_LONG = 223 | _ControllerError , ControllerError_TABLE_LOCATION_WRITE_PROTE
= 224 | _ControllerError , ControllerError_ENCRYPTION_AUTHORITY_NOT_FOUND = 225 | _Controller↩
Error , ControllerError_NO_FREE_ENCRYPTION_AUTHORITY = 226 | _ControllerError ,
ControllerError_LOCAL_TARGET_VALUE_HAS_NOT_BEEN_DEFINED = 227 | _ControllerError ,
ControllerError_COMMAND_NOT_PERMITTED_IN_BASIC_LIBRARY = 228 | _ControllerError , ControllerError_INVALID_MO
= 229 | _ControllerError , ControllerError_FUNCTION_EXPECTED = 230 | _ControllerError ,
ControllerError_CAM_TABLE_ID_NOT_DEFINED = 231 | _ControllerError , ControllerError_RUNNING_OF_RPS_PROGRAMS
= 232 | _ControllerError , ControllerError_U_AND_V_ORIENTATION_INHIBITED_WHILE_WRIST_SINGULARITY
= 233 | _ControllerError , ControllerError_TARGET_POINT_OUT_OF_ROBOT_WORKING_AREA = 234 |
_ControllerError ,
ControllerError_AXIS_LOCKED_TO_A_ROBOT = 235 | _ControllerError , ControllerError_ROBOT_CONTROLLED_BY_ANOT
= 236 | _ControllerError , ControllerError_ROBOT_NOT_DEFINED_USING_KINEMATIC_GROUP = 237 | ↩
_ControllerError , ControllerError_COMMAND_INHIBITED_ROBOT_MUST_BE_USED_TO_ACTIVATE_A_KINEMATIC_GROU
= 238 | _ControllerError ,
ControllerError_UNDEFINED_VARIABLE_IN_DIM_INITIALISER = 239 | _ControllerError , ControllerError_THE_SPECIFIED_F
= 240 | _ControllerError , ControllerError_TOO_MANY_AXES_ACTIVATED = 241 | _ControllerError ,
ControllerError_MULTIPLE_MOVE_TRIGGERS_SPECIFIED = 242 | _ControllerError ,
ControllerError_MOVE_TRIGGER_RESPONSE_NOT_SPECIFIED = 243 | _ControllerError , ControllerError_MOVE_TRIGGER
= 244 | _ControllerError , ControllerError_INVALID_ARRAY_DIMENSION = 245 | _ControllerError ,
ControllerError_DUPLICATE_VR_INDEX = 246 | _ControllerError ,
ControllerError_CONSTANT_VALUE_EXPECTED = 247 | _ControllerError , ControllerError_RPS_KERNEL_V2_NOT_ENABLE
= 248 | _ControllerError , ControllerError_INCLUDE_FILE_MUST_CONTAIN_DEFCONST_DIM_STATEMENTS_ONLY
= 249 | _ControllerError , ControllerError_TARGET_TAUGHT_WITH_ANOTHER_ROBOT = 250 | _↩
ControllerError ,
ControllerError_TARGET_ROBOT_CONFIG_MISMATCH = 251 | _ControllerError , ControllerError_TARGET_TOOL_OFFSET_
= 252 | _ControllerError , ControllerError_TARGET_OBJECT_FRAME_MISMATCH = 253 | _ControllerError
, ControllerError_TARGET_ROBOT_FRAME_MISMATCH = 254 | _ControllerError ,
ControllerError_INVALID_FRAME_NUMBER = 255 | _ControllerError , ControllerError_FRAME_NOT_SUPPORTED_BY_RPS
= 256 | _ControllerError , ControllerError_FORWARD_TRANSFORM_FUNCTION_NOT_DEFINED = 257
| _ControllerError , ControllerError_INVERSE_TRANSFORM_FUNCTION_NOT_DEFINED = 258 | _↩
ControllerError ,
ControllerError_INVALID_MOVEOPTION = 259 | _ControllerError , ControllerError_MAX_OPEN_KEYWORDS_REACHED
= 260 | _ControllerError , ControllerError_KEYWORD_INACTIVE = 261 | _ControllerError , ControllerError_KEYWORD_FUNCT
= 262 | _ControllerError ,
ControllerError_KEYWORD_SAME_AS_SYSTEM_WORD = 263 | _ControllerError , ControllerError_UNEXPECTED_CHARAC
= 264 | _ControllerError , ControllerError_THIS_COMMAND_CAN_BE_EXECUTED_ONLY_ONCE = 265 |
_ControllerError , ControllerError_TABLE_MEMORY_CONFLICT = 266 | _ControllerError ,
ControllerError_MAX_OPEN_MOVES_REACHED = 267 | _ControllerError , ControllerError_USER_DEFINED_ERROR
= 268 | _ControllerError , ControllerError_INVALID_FORMAT_SPECIFIER = 269 | _ControllerError ,
ControllerError_OPEN_MOVE_FUNCTIONS_NOT_DEFINED = 270 | _ControllerError ,
ControllerError_TOO_MANY_CHARS_IN_STR = 271 | _ControllerError , ControllerError_INVALID_OPEN_ARCH_FUNC_DEF
= 272 | _ControllerError , ControllerError_PROCESS_IN_USE = 273 | _ControllerError , ControllerError_VR_LOCATION_WRITE
= 274 | _ControllerError }

Enumeration of error codes used in Unified API.

- enum EventType { Error , Warning , Message }

  Events that may occur on a connection.

- enum ConnectionType : UInt32 { Unknown = 0x00 , TCP = 0x01 , PCMCAT = 0x02 , FLEX7 = 0x04 }

  Connection types.

- enum ProgressEvent { Start , Progress , Finish }

  Progress event.

- enum ParamBlockType {
  AxisParam = 0 , SystemParameters = 1 , VR = 2 , TABLE = 3 ,
  ProgramVariable = 4 }

  Types of connections available.
- enum Co_ObjectType {
  Boolean = 1 , Integer8 = 2 , Integer16 = 3 , Integer32 = 4 ,
  Unsigned8 = 5 , Unsigned16 = 6 , Unsigned32 = 7 , Real32 = 8 ,
  Real64 = 17 , Integer64 = 21 , Unsigned64 = 27 }

  Co Read/Write operation object types.
- enum EthercatState { Initial = 0 , PreOprational = 1 , SafeOperational = 2 , Operational = 3 }

  EtherCAT state.
- enum EthercatStartOption : UInt32 { None = 0x0000 , PrintOutProgress = 0x0001 , ForceRescan = 0x0002 , ExpressStartup = 0x0004 }

  EtherCAT start option.
- enum FileTransferOption : UInt32 { None = 0x0000 , CompressOnly = 0x0001 , CompressDecompress = 0x0002 }

  File transfer option.
- enum FileType : Int32 {
  Undefined = -1 , BASIC = 0 , EncryptedBASIC = 1 , GCode = 2 ,
  TextFile = 3 , IecProgram = 4 , FifoFile = 5 , TempFile = 6 ,
  Project = 7 , Binary = 8 , MCConfig = 9 , IecLibrary = 10 ,
  HmiDesign = 11 , BASICLibrary = 12 , EncryptedBASICibrary = 13 , HmiLibrary = 14 ,
  TempVolatileFile = 15 , Robot = 16 , RobotBasic = 17 , TempTextFile = 18 ,
  RobotLibrary = 19 , RobotBasicLibrary = 20 , Recipe = 21 , Alarms = 22 ,
  RealtimeBASICLibrary = 23 , EncryptedRealtimeBasicLibrary = 24 , EncryptedIecProgram = 25 ,
  EncryptedIecLibrary = 26 ,
  __Size }

  File types on the Motion Coordinator.
- enum FileFlags : UInt32 { None = 0x00 , Editable = 0x01 , Autorun = 0x02 }

  Flags related to files and programs in internal storage on the Motion Coordinator.
- enum ConnectMode : UInt32 { MPOS = 0 , DPOS = 1 }

  Mode for CONNECT command.
- enum PSwitchMode : UInt32 { None = 0x00 , Invert = 0x02 , DPOS = 0x04 , Mask = Invert | DPOS }

  Flags PSWITCH command in single/cyclic mode.
- enum PSwitchMultiMode : UInt32 {
  None = 0x00 , DPOS = 0x04 , TimerMode = 0x10 , NegativeDirection = 0x20 ,
  ZOut = 0x40 , Mask = DPOS | TimerMode | NegativeDirection | ZOut }

  Flags PSWITCH command in multi/sequence mode.
- enum CamBoxLinkOptions : UInt32 {
  None = 0 , LinkMark = 1 , LinkAbsolute = 2 , LinkRepeat = 4 ,
  LinkPattern = 8 , LinkPositiveMove = 32 , LinkForceStart = 128 , LinkMarkB = 256 ,
  LinkR_Mark = 512 , LinkThreshold = 16384 }

  Link options in CamBox command.
- enum AxisType : UInt16 {
  None = 0 , Stepper = 1 , Servo = 2 , Encoder = 3 ,
  StepperEncoderInt = 4 , Resolver = 5 , VOut = 6 , SSI = 7 ,
  CAN = 8 , CAN_Remote = 9 , PSwitch = 10 , DLink_Remote = 11 ,
  Servo2 = 12 , Yaskawa = 13 , EncoderOut = 14 , HighestMotionType = EncoderOut ,
  Dummy15 = 15 , Dummy16 = 16 , Dummy17 = 17 , DS402PosType1 = 18 ,
  DS402VelType1 = 19 , First_Comms = 20 , CAN_Comms = First_Comms , USB_Comms = 21 ,
  DLink_Comms = 22 , Profibus_Comms = 23 , Dummy24 = 24 , Dummy25 = 25 ,
  DS402PosType2 = 26 , DS402VelType2 = 27 , Dummy28 = 28 , Dummy29 = 29 ,
  DACout_ANALin = 30 , ANALout_ANALin = 31 , Dummy32 = 32 , Dummy33 = 33 ,
  Dummy34 = 34 , SDout_PWMout = 35 , SDout_PWMout2 = 36 , PWMout_SDFREQin = 37 ,
  PWMout_SDin = 37 , Dummy38 = 38 , Dummy39 = 39 , MechatroPos = 40 ,

MechatroSpeed = 41 , MechatroTorque = 42 , SDout_Zout = 43 , DACout_QUADin = 44 ,
QUADout_Zout = 45 , DACout_TAMAin = 46 , DACout_ENDATin = 47 , DACout_SSIin = 48 ,
MechatroInv = 49 , PanasonicPos = 50 , PanasonicSpeed = 51 , PanasonicTorque = 52 ,
SercosSpeed = 53 , SercosPos = 54 , SercosTorque = 55 , SercosOpen = 56 ,
SercosSpeedReg = 57 , SercosPosReg = 58 , SercosEncoder = 59 , DACout_SDin = 60 ,
SLM32 = 61 , PLM32 = 62 , SDout_Zin = 63 , QUADout_Zin = 64 ,
EthercatPos = 65 , EthercatSpeed = 66 , EthercatTorque = 67 , EthercatOpenSpeed = 68 ,
EthercatEncoderIn = 69 , EthercatInAnalOut = 70 , EthercatPosTrio = 71 , EthercatSpeedTrio = 72 ,
EthercatDummy5 = 73 , EthercatDummy6 = 74 , DACout_SSISLAVEin = 75 , QUADin = 76 ,
DACout_QUADin_Zout = 77 , SDout_VFFDACout = 78 , SDin = 79 , G5Pos = 80 ,
G5Speed = 81 , G5Torque = 82 , Jacquard = 83 , QUADout_VFFDACout = 84 ,
AxesDiff = 85 , TAMAin = 86 , ENDATin = 87 , SSIin = 88 ,
SDFREQout_ANALin = 89 , DACout_SDFREQin = 90 , AFFDACout = 91 , DACout_BISSin = 92 ,
BISSin = 93 , DACPWMout_QUADin = 94 , SDout_TAMAin = 95 , Dummy96 = 96 ,
QUADin_Zout = 97 , Dummy98 = 98 , Dummy99 = 99 , SDout_HD = 100 ,
QUADout_HD = 101 , DACout_QUADin_HD = 102 , DACout_SDin_HD = 103 , QUADin_HD = 104 ,
SSIin_HD = 105 , TAMAin_HD = 106 , Dummy107 = 107 , Dummy108 = 108 ,
Dummy109 = 109 , SDout_SDin_HD = 110 , SDout_QUADin_HD = 111 , QUADout_SDin_HD = 112 ,
QUADout_QUADin_HD = 113 , SDout_TAMAin_HD = 114 , SDout_ENDATin_HD = 115 , SDout_SSIin_HD
= 116 ,
SDout_BISSin_HD = 117 , QUADout_TAMAin_HD = 118 , QUADout_ENDATin_HD = 119 , QUADout_SSIin_HD
= 120 ,
QUADout_BISSin_HD = 121 , Dummy122 = 122 , Dummy123 = 123 , Dummy124 = 124 ,
Dummy125 = 125 , Dummy126 = 126 , Dummy127 = 127 , Dummy128 = 128 ,
Dummy129 = 129 , SLMvirtual = 200 , PLMvirtual = 201 }

Enumeration of known axis types used with ATYPE axis parameter.

- enum AxisMode : UInt32 {
NOT_USED_1 = ((UInt32)1 << 0) , ALLOW_CONNECT_MOVES_IN_LIMIT = ((UInt32)1 << 1) ,
DIRECTION_MODE_3D = ((UInt32)1 << 2) , CANCEL_ADDAX_ON_LIMIT = ((UInt32)1 << 3) ,
DISABLE_LOOKAHEAD = ((UInt32)1 << 4) , NOT_USED_2 = ((UInt32)1 << 5) , USE_NON_SIGNEXTENDED_ANALOGUE
= ((UInt32)1 << 6) , NOT_USED_3 = ((UInt32)1 << 7) ,
SOFTWARE_LIMIT_USES_DPOS = ((UInt32)1 << 8) , DISABLE_FE_ACCUMULATION_DURING_MOVE
= ((UInt32)1 << 9) , USE_FASTDEC_WHEN_CANCELLING_RPS_JOG = ((UInt32)1 << 10) ,
DELAY_POSITION_EVALUATION_TO_LOAD_TIME = ((UInt32)1 << 11) ,
DIRECTION_SHORTEST = ((UInt32)1 << 12) , MONITOR_FE = ((UInt32)1 << 13) , OVERRIDE_FE_LIMITS
= ((UInt32)1 << 14) , DUAL_AXIS_AVERAGE_POS = ((UInt32)1 << 15) ,
INVERT_AXIS_DIRECTION = ((UInt32)1 << 16) , CANCEL_MTYPE_ON_MASTER_IF_ADDAX_HITS_LIM
= ((UInt32)1 << 17) }

  Enumeration of axis mode flags used with AXIS_MODE axis parameter.

- enum CornerMode : UInt32 {
RESERVED = ((UInt32)1 << 0) , AUTO_SPEED_CONTROL = ((UInt32)1 << 1) , CORNER_STATE_MACHINE
= ((UInt32)1 << 2) , AUTO_RADIUS_SPEED_CONTROL = ((UInt32)1 << 3) ,
AUTO_RADIUS_SPEED_CONTROL_CENTRIPETAL_MODE = ((UInt32)1 << 4) , AUTO_RADIUS_SPEED_CONTROL_CEN
= ((UInt32)1 << 5) , AUTO_BLEND_MODE_0_USING_CORNER_RADIUS = ((UInt32)1 << 6) ,
AUTO_BLEND_MODE_1_USING_CORNER_RADIUS = ((UInt32)1 << 7) }

  Enumeration of corner mode flags used with CORNER_MODE axis parameter.

- enum CornerState : UInt32 { HELD = ((UInt32)1 << 0) , LOAD = ((UInt32)1 << 1) , LOAD_ZERO_SPEED
= ((UInt32)1 << 2) }

  Enumeration of corner state flags used with CORNER_STATE axis parameter.

- enum MotionType : UInt32 {
IDLE = 0 , MOVE = 1 , MOVEABS = 2 , MHELICAL = 3 ,
MOVECIRC = 4 , MOVEMODIFY = 5 , MOVESP = 6 , MOVEABSSP = 7 ,
MOVECIRCSP = 8 , MHELICALSP = 9 , FORWARD = 10 , REVERSE = 11 ,
DATUM = 12 , CAM = 13 , FORWARD_JOG = 14 , REVERSE_JOG = 15 ,
WETCAM = 16 , CAMBOX = 20 , CONNECT = 21 , MOVELINK = 22 ,
CONNPATH = 23 , FLEXLINK = 24 , CAMLINK = 25 , COMPENSATE_XY = 26 ,
PICKLINK = 27 , CAMIN = 28 , MOVETANG = 30 , MSPHERICAL = 31 ,

MTRANSCIRC = 32 , MOVESEQ = 33 , MOVEABSSEQ = 34 , MOVETURN = 35 ,
MOVEPICK = 36 , MOVECONTOUR = 37 , MOVEDISPENSE = 38 , MSPHERICALSP = 40 ,
MTRANSCIRCSP = 41 , MOVESEQSP = 42 , MOVEABSSEQSP = 43 , MOVEJ = 44 ,
MOVEL = 45 , MOVEC = 46 , MOVEBLEND = 47 , MOVEJREL = 48 ,
MOVELREL = 49 , MOVETEST = 50 , MOVEP = 51 , MOVEU = 52 ,
MOVEB = 53 , MOVESPCONT = 60 , MOVEABSSPCONT = 61 , GEARIN = 62 ,
ACCPROFILE = 70 , VELPROFILE = 71 , POSPROFILE = 72 , COMBINEAXES = 73 ,
OPENMOVE_BASE = 100 , OPENMOVE_1 = OPENMOVE_BASE , OPENMOVE_2 = OPENMOVE_BASE + 1 , OPENMOVE_3 = OPENMOVE_BASE + 2 ,
OPENMOVE_4 = OPENMOVE_BASE + 3 , OPENMOVE_5 = OPENMOVE_BASE + 4 }

Motion types returned by MTYPE and NTYPE axis parameters.

- enum AxisStatus : UInt64 {
AS_LOOKAHEAD_OVERRIDE = ((UInt64)1 << 0) , AS_FE_WARNING = ((UInt64)1 << 1) ,
AS_COMMS_ERROR = ((UInt64)1 << 2) , AS_REMOTE_DRIVE_ERROR = ((UInt64)1 << 3) ,
AS_IN_FORWARD_LIMIT = ((UInt64)1 << 4) , AS_IN_REVERSE_LIMIT = ((UInt64)1 << 5) ,
AS_DATUMING = ((UInt64)1 << 6) , AS_FEEDHOLD = ((UInt64)1 << 7) ,
AS_FE_EXCEEDS_LIMIT = ((UInt64)1 << 8) , AS_FORWARD_SOFTWARE_LIMIT = ((UInt64)1 << 9) ,
AS_REVERSE_SOFTWARE_LIMIT = ((UInt64)1 << 10) , AS_CANCEL_CURRENT = ((UInt64)1 << 11) ,
AS_PULSE_OVER_SPEED = ((UInt64)1 << 12) , AS_OVERRIDE_TO_ZERO = ((UInt64)1 << 13) ,
AS_MOVE_CANCEL_DONE = ((UInt64)1 << 14) , AS_VOLUME_LIMIT = ((UInt64)1 << 15) ,
AS_AXIS_FS_LIMIT = ((UInt64)1 << 16) , AS_AXIS_RS_LIMIT = ((UInt64)1 << 17) , AS_ENCODER_OVER_I = ((UInt64)1 << 18) , AS_PSWITCH_FIFO_NOT_EMPTY = ((UInt64)1 << 19) ,
AS_PSWITCH_FIFO_FULL = ((UInt64)1 << 20) , AS_KINEMATICS_1HOUR = ((UInt64)1 << 21) ,
AS_BISS_CLEAN_WARNING = ((UInt64)1 << 22) , AS_WORLD_FS_LIMIT = ((UInt64)1 << 23) ,
AS_WORLD_RS_LIMIT = ((UInt64)1 << 24) , AS_CANCEL_CURRENT_AT_SPEED = ((UInt64)1 << 25) , AS_REMOTE_NODE_STATUS_ERROR = ((UInt64)1 << 26) , AS_PSWITCH_FIFO_HALF_FULL = ((UInt64)1 << 27) ,
AS_MOVEADD = ((UInt64)1 << 28) , AS_CANCEL_MOVEADD = ((UInt64)1 << 29) , AS_REMOTE_NODE_ERROR2 = ((UInt64)1 << 30) , AS_SINGULARITY = ((UInt64)1 << 31) ,
AS_OPEN_FRAME_ERROR = ((UInt64)1 << 32) , AS_VP_ERROR = ((UInt64)1 << 33) , AS_CANCEL_BUFFERED = ((UInt64)1 << 34) , AS_CANCEL_BUFFERED_DECEL_INTO_BUFFERED_MOVES = ((UInt64)1 << 35) ,
AS_CANCEL_BUFFERED_DECEL_WITHIN_CURRENT_MOVE = ((UInt64)1 << 36) , AS_CANCEL_LAST_BUFFERED = ((UInt64)1 << 37) }
- enum RobotStatus : UInt64 {
RS_WORLD_FS_LIMIT = ((UInt64)1 << 0) , RS_WORLD_RS_LIMIT = ((UInt64)1 << 1) , RS_ROBOT_FS_LIMIT = ((UInt64)1 << 2) , RS_ROBOT_RS_LIMIT = ((UInt64)1 << 3) ,
RS_TCP_FS_LIMIT = ((UInt64)1 << 4) , RS_TCP_RS_LIMIT = ((UInt64)1 << 5) , RS_FE_WARNING = ((UInt64)1 << 6) , RS_FE_EXCEEDS_LIMIT = ((UInt64)1 << 7) ,
RS_WRIST_SINGULARITY = ((UInt64)1 << 8) , RS_ALIGNMENT_SINGULARITY = ((UInt64)1 << 9) ,
RS_ELBOW_SINGULARITY = ((UInt64)1 << 10) , RS_MAX_SPEED_LIMIT = ((UInt64)1 << 11) ,
RS_COLLIDED = ((UInt64)1 << 12) , RS_NOT_IN_POS = ((UInt64)1 << 13) , RS_ELBOW_CLOSED_SINGULARITY = ((UInt64)1 << 14) , RS_DISALLOW_JOG = ((UInt64)1 << 15) }
- enum ProcessStatus : UInt32 {
STOPPED , READY , STEPPING , PAUSED ,
PAUSING , ABORTED , SUSPEND , SUSPENDED ,
SLEEPING }
- enum SyncControl : Int32 {
SC_WAITING_TO_START = -1 , SC_NO_SYNC = 0 , SC_STARTING_SYNC = 1 , SC_PERFORMING_SYNC = 2 ,
SC_SYNCHRONISED = 3 , SC_STOPPING_SYNC = 4 , SC_STARTING_INTERP_MOVE = 5 ,
SC_PERFORMING_INTERP_MOVE = 6 ,
SC_STARTING_RESYNC = 10 , SC_PERFORMING_RESYNC = 11 , SC_STARTING_RESYNC_TO_USERFRAME = 20 , SC_PERFORMING_RESYNC_TO_USERFRAME = 21 }

Functions

- delegate void ConnectionEventHandler (EventType eventType, long integerData, string stringData)

Delegate for connection related events.

- delegate void FileCallback (ref FileInfo file, string fileName)

  Delegate for processing the response of Dir command.

- unsafe delegate void StreamHandlerCallback (ref DataStreamPacket packet, byte *buff)

  Delegate for streaming back the response of commands that return string data.

- delegate void ProgressHandlerCallback (ref ProgressInfo progress)

  Delegate for streaming back the response of commands that return string data.

5.2.1 Enumeration Type Documentation

5.2.1.1 AxisMode

```
enum Trio.UnifiedApi.AxisMode : UInt32
```

Enumeration of axis mode flags used with AXIS_MODE axis parameter.

Enumerator

NOT_USED_1	
ALLOW_CONNECT_MOVES_IN_LIMIT	
DIRECTION_MODE_3D	
CANCEL_ADDAX_ON_LIMIT	
DISABLE_LOOKAHEAD	
NOT_USED_2	
USE_NON_SIGNEXTENDED_ANALOGUE_FEEDBACK	
NOT_USED_3	
SOFTWARE_LIMIT_USES_DPOS	
DISABLE_FE_ACCUMULATION_DURING_MOVE	
USE_FASTDEC_WHEN_CANCELLING_RPS_JOG	
DELAY_POSITION_EVALUATION_TO_LOAD_TIME	
DIRECTION_SHORTEST	
MONITOR_FE	
OVERRIDE_FE_LIMITS	
DUAL_AXIS_AVERAGE_POS	
INVERT_AXIS_DIRECTION	
CANCEL_MTYPE_ON_MASTER_IF_ADDAX_HITS_LIM	

Definition at line 1569 of file ITrioConnection.cs.

5.2.1.2 AxisStatus

```
enum Trio.UnifiedApi.AxisStatus : UInt64
```

Enumerator

AS_LOOKAHEAD_OVERRIDE	
AS_FE_WARNING	
AS_COMMS_ERROR	
AS_REMOTE_DRIVE_ERROR	

Enumerator

AS_IN_FORWARD_LIMIT	
AS_IN_REVERSE_LIMIT	
AS_DATUMING	
AS_FEEDHOLD	
AS_FE_EXCEEDS_LIMIT	
AS_FORWARD_SOFTWARE_LIMIT	
AS_REVERSE_SOFTWARE_LIMIT	
AS_CANCEL_CURRENT	
AS_PULSE_OVER_SPEED	
AS_OVERRIDE_TO_ZERO	
AS_MOVE_CANCEL_DONE	
AS_VOLUME_LIMIT	
AS_AXIS_FS_LIMIT	
AS_AXIS_RS_LIMIT	
AS_ENCODER_OVER_I	
AS_PSWITCH_FIFO_NOT_EMPTY	
AS_PSWITCH_FIFO_FULL	
AS_KINEMATICS_1HOUR	
AS_BISS_CLEAN_WARNING	
AS_WORLD_FS_LIMIT	
AS_WORLD_RS_LIMIT	
AS_CANCEL_CURRENT_AT_SPEED	
AS_REMOTE_NODE_STATUS_ERROR	
AS_PSWITCH_FIFO_HALF_FULL	
AS_MOVEADD	
AS_CANCEL_MOVEADD	
AS_REMOTE_NODE_ERROR2	
AS_SINGULARITY	
AS_OPEN_FRAME_ERROR	
AS_VP_ERROR	
AS_CANCEL_BUFFERED	
AS_CANCEL_BUFFERED_DECEL_INTO_BUFFERED_MOVES	
AS_CANCEL_BUFFERED_DECEL_WITHIN_CURRENT_MOVE	
AS_CANCEL_LAST_BUFFERED	

Definition at line 1775 of file ITrioConnection.cs.

5.2.1.3 AxisType

```
enum Trio.UnifiedApi.AxisType : UInt16
```

Enumeration of known axis types used with ATYPE axis parameter.

Enumerator

None	
Stepper	
Servo	

Enumerator

Encoder	
StepperEncoderInt	
Resolver	
VOut	
SSI	
CAN	
CAN_Remote	
PSwitch	
DLink_Remote	
Servo2	
Yaskawa	
EncoderOut	
HighestMotionType	
Dummy15	
Dummy16	
Dummy17	
DS402PosType1	
DS402VelType1	
First_Comms	
CAN_Comms	
USB_Comms	
DLink_Comms	
Profibus_Comms	
Dummy24	
Dummy25	
DS402PosType2	
DS402VelType2	
Dummy28	
Dummy29	
DACout_ANALin	
ANALout_ANALin	
Dummy32	
Dummy33	
Dummy34	
SDout_PWMout	
SDout_PWMout2	PWM variant that uses path distance to generate output.
PWMout_SDFREQin	
PWMout_SDin	
Dummy38	
Dummy39	
MechatroPos	This is the first MC464 usable type.
MechatroSpeed	
MechatroTorque	
SDout_Zout	Standard stepper axis.
DACout_QUADin	Standard analogue servo axis with encoder feedback.
QUADout_Zout	
DACout_TAMAin	
DACout_ENDATin	

Enumerator

DACout_SSIin	
MechatroInv	
PanasonicPos	
PanasonicSpeed	
PanasonicTorque	
SercosSpeed	
SercosPos	
SercosTorque	
SercosOpen	
SercosSpeedReg	
SercosPosReg	
SercosEncoder	
DACout_SDin	
SLM32	
PLM32	
SDout_Zin	
QUADout_Zin	
EthercatPos	
EthercatSpeed	
EthercatTorque	
EthercatOpenSpeed	
EthercatEncoderIn	
EthercatInAnalOut	
EthercatPosTrio	
EthercatSpeedTrio	
EthercatDummy5	
EthercatDummy6	
DACout_SSISLAVEin	
QUADin	Standard encoder only axis.
DACout_QUADin_Zout	
SDout_VFFDACout	
SDin	
G5Pos	
G5Speed	
G5Torque	
Jacquard	
QUADout_VFFDACout	
AxesDiff	
TAMAin	
ENDATin	
SSIin	
SDFREQout_ANALin	
DACout_SDFREQin	
AFFDACout	
DACout_BISSin	
BISSin	
DACPWMout_QUADin	
SDout_TAMAin	
Dummy96	

Enumerator

QUADin_Zout	
Dummy98	
Dummy99	
SDout_HD	
QUADout_HD	
DACout_QUADin_HD	
DACout_SDin_HD	
QUADin_HD	
SSIin_HD	
TAMAin_HD	
Dummy107	
Dummy108	
Dummy109	
SDout_SDin_HD	
SDout_QUADin_HD	
QUADout_SDin_HD	
QUADout_QUADin_HD	
SDout_TAMAin_HD	
SDout_ENDATin_HD	
SDout_SSIin_HD	
SDout_BISSin_HD	
QUADout_TAMAin_HD	
QUADout_ENDATin_HD	
QUADout_SSIin_HD	
QUADout_BISSin_HD	
Dummy122	
Dummy123	
Dummy124	
Dummy125	
Dummy126	
Dummy127	
Dummy128	
Dummy129	
SLMvirtual	
PLMvirtual	

Definition at line 1277 of file ITrioConnection.cs.

5.2.1.4 CamBoxLinkOptions

```
enum Trio.UnifiedApi.CamBoxLinkOptions : UInt32
```

Link options in CamBox command.

Enumerator

None	No flags are set.
LinkMark	link commences exactly when registration event MARK occurs on link axis

Enumerator

LinkAbsolute	link commences at an absolute position on link axis
LinkRepeat	CAMBOX repeats automatically and bi-directionally when this bit is set.
LinkPattern	PATTERN mode. Advanced use of CAMBOX: allows multiple scale values to be used.
LinkPositiveMove	Link is only active during a positive move on the link axis.
LinkForceStart	Forces the profile to start at a defined point in the link_dist.
LinkMarkB	link commences exactly when registration event MARKB occurs on link axis
LinkR_Mark	link commences exactly when registration event R_MARK occurs on link axis.
LinkThreshold	Link is only active during a negative move on the link axisLink is active during positive movement from a threshold.

Definition at line 1221 of file ITrioConnection.cs.

5.2.1.5 Co_ObjectType

```
enum Trio.UnifiedApi.Co_ObjectType
```

Co Read/Write operation object types.

Enumerator

Boolean	Boolean type.
Integer8	8 bit integer
Integer16	16 bit integer
Integer32	32 bit integer
Unsigned8	8 bit unsigned integer
Unsigned16	16 bit unsigned integer
Unsigned32	32 bit unsigned integer
Real32	32 bit floating point
Real64	64 bit floating point
Integer64	64 bit integer
Unsigned64	64 bit unsigned integer

Definition at line 819 of file ITrioConnection.cs.

5.2.1.6 ConnectionType

```
enum Trio.UnifiedApi.ConnectionType : UInt32
```

Connection types.

Enumerator

Unknown	Unknown connection type. Returned when there is no connection context.
TCP	TCP connection via Remote protocol.
PCMCAT	Shared memory connection for PC-MCAT devices.
FLEX7	Shared memory connection for Flex7 devices.

Definition at line 741 of file ITrioConnection.cs.

5.2.1.7 ConnectMode

```
enum Trio.UnifiedApi.ConnectMode : UInt32
```

Mode for CONNECT command.

Enumerator

MPOS	Measured position (MPOS) is followed by connected axis.
DPOS	Demand position (DPOS) is followed by connected axis.

Definition at line 1134 of file ITrioConnection.cs.

5.2.1.8 CornerMode

```
enum Trio.UnifiedApi.CornerMode : UInt32
```

Enumeration of corner mode flags used with CORNER_MODE axis parameter.

Enumerator

RESERVED	
AUTO_SPEED_CONTROL	
CORNER_STATE_MACHINE	
AUTO_RADIUS_SPEED_CONTROL	
AUTO_RADIUS_SPEED_CONTROL_CENTRIPETAL_MODE	
AUTO_RADIUS_SPEED_CONTROL_CENTRIPETAL_MODE_USE_ACCEL	
AUTO_BLEND_MODE_0_USING_CORNER_RADIUS	
AUTO_BLEND_MODE_1_USING_CORNER_RADIUS	

Definition at line 1612 of file ITrioConnection.cs.

5.2.1.9 CornerState

```
enum Trio.UnifiedApi.CornerState : UInt32
```

Enumeration of corner state flags used with CORNER_STATE axis parameter.

Enumerator

HELD	
LOAD	
LOAD_ZERO_SPEED	

Definition at line 1635 of file ITrioConnection.cs.

5.2.1.10 ErrorCode

```
enum Trio.UnifiedApi.ErrorCode : UInt64
```

Enumeration of error codes used in Unified API.

Enumerator

NoError	No error.
_Unknown	An error code that cannot be determined.
_ApiError	UnifiedAPI error flag. The actual error code is in the lowest 32 bits.
Api_Unknown	An UnifiedAPI error that cannot be determined.
Fail	General error.
NotSupported	Error indicating that operation is not supported.
ConnectionModule	Requested connection type (module) cannot be load/initialized or is missing.
ConnectionContext	A bad connection context is used.
TokenTable	TOKENTABLE reading failed.
Timeout	The operation timed out.
UnknownFloatingPointFormat	The data is of unknown floating point format.
Remote_TokenBuffer	Error in the token buffer for the Remote protocol.
Remote_UnknownToken	No such token (Remote protocol)
UnsupportedController	Unsupported controller.
BadResponse	Bad response/unrecognized received.
BadArgument	A bad argument has been passed.
Reopen	The connection or object is already opened.
BufferTooSmall	The passed buffer is too small.
SocketRead	The read to the socket failed.
SocketWrite	The write to the socket failed.
LowLevelError	Low level API call failed.
VersionMismatch	Versions of modules do not match.
Compression	Compress/Decompress data error.
Abort	The operation got aborted.
Disconnected	No connection.
_ApiLast	The last API error.
_SystemError	A system error flag. The actual system error code is in the lowest 32 bits.
Sys_Unknown	Unknown system error.
_ControllerError	A controller error flag. The actual controller error code is in the lowest 32 bits.
ControllerError_NONE	Controller Error codes.
ControllerError_COMMAND_NOT_RECOGNISED	
ControllerError_INVALID_TRANSFER_TYPE	
ControllerError_FLASH_PROGRAMMING_ERROR	
ControllerError_OPERAND_EXPECTED	
ControllerError_ASSIGNMENT_EXPECTED	
ControllerError_QUOTES_EXPECTED	
ControllerError_STACK_OVERFLOW	
ControllerError_TOO_MANY_VARIABLES	
ControllerError_DIVISION_BY_ZERO	
ControllerError_EXTRA_CHARS_AT_EOLN	

Enumerator

ControllerError_CLOSE_SQUARE_BRACKET_↩ EXPECTED_IN_PRINT	
ControllerError_CANNOT_MODIOFY_SPECIAL_↩ PROGRAM	
ControllerError_THEN_EXPECTED_IN_IF_ELSEIF	
ControllerError_FLASH_ERASE_ERROR	
ControllerError_START_OF_EXPRESSION_↩ EXPECTED	
ControllerError_CLOSE_ROUND_BRACKET_↩ EXPECTED	
ControllerError_COMMA_EXPECTED	
ControllerError_COMMAND_LINE_BROKEN_BY_↩ ESC	
ControllerError_PARAMETER_OUT_OF_RANGE	
ControllerError_NO_PROCESS_AVAILABLE	
ControllerError_VALUE_IS_READ_ONLY	
ControllerError_MODIFIER_NOT_ALLOWED	
ControllerError_REMOTE_AXIS_IS_IN_USE	
ControllerError_COMMAND_IS_COMMAND_LINE↩ _ONLY	
ControllerError_COMMAND_IS_RUNTIME_ONLY	
ControllerError_LABEL_EXPECTED	
ControllerError_PROGRAM_NOT_FOUND	
ControllerError_DUPLICATE_IDENTIFIER	
ControllerError_PROGRAM_IS_LOCKED	
ControllerError_PROGRAMS_RUNNING	
ControllerError_PROGRAM_IS_STOPPED	
ControllerError_CANNOT_SELECT_PROGRAM	
ControllerError_NO_PROGRAM_SELECTED	
ControllerError_NO_MORE_PROGRAMS_↩ AVAILABLE	
ControllerError_OUT_OF_MEMORY	
ControllerError_NO_CODE_AVAILABLE_FOR_↩ PROGRAM	
ControllerError_COMMAND_OUT_OF_CONTEXT	
ControllerError_TOO_MANY_NESTED_↩ STRUCTURES	
ControllerError_STRUCTURE_NESTING_ERROR	
ControllerError_ELSE_ELSEIF_ENDIF_WITHOUT↩ _PREVIOUS_IF	
ControllerError_WEND_WITHOUT_PREVIOUS_↩ WHILE	
ControllerError_UNTIL_WITHOUT_PREVIOUS_↩ REPEAT	
ControllerError_IDENTIFIER_EXPECTED	
ControllerError_TO_EXPECTED_AFTER_FOR	
ControllerError_TOO_MANY_NESTED_FOR_NEXT	
ControllerError_NEXT_WITHOUT_FOR	
ControllerError_UNTIL_IDLE_LOADED_↩ EXPECTED_AFTER_WAIT	
ControllerError_GOTO_GOSUB_EXEC_EXPECTED	

Enumerator

ControllerError_TOO_MANY_NESTED_GOSUB	
ControllerError_RETURN_WITHOUT_GOSUB	
ControllerError_LABEL_MUST_BE_AT_START_↩ OF_LINE	
ControllerError_CANNOT_NEST_ONE_LINE_IF	
ControllerError_LABEL_NOT_FOUND	
ControllerError_LINE_NUMBER_CANNOT_HAVE↩ _DECIMAL_POINT	
ControllerError_MAXIMUM_NUMBER_OF_↩ PROGRAM_INSTANCES_ACTIVE	
ControllerError_INVALID_USE_OF_DOLAR_SIGN	
ControllerError_VR_X_OR_STRING_VARIABLE_↩ EXPECTED	
ControllerError_PROGRAM_ALREADY_EXISTS	
ControllerError_PROCESS_ALREADY_SELECTED	
ControllerError_DUPLICATE_AXES_NOT_↩ PERMITTED	
ControllerError_PLC_TYPE_IS_INVALID	
ControllerError_EVALUATION_ERROR	
ControllerError_RESERVED_KEYWORD_NOT_↩ AVAILABLE_ON_THIS_CONTROLLER	
ControllerError_VARIABLE_NOT_FOUND	
ControllerError_TABLE_INDEX_RANGE_ERROR	
ControllerError_FEATURES_ENABLED_DO_NOT↩ _ALLOW_ATYPE_CHANGE	
ControllerError_INVALID_LINE_NUMBER	
ControllerError_STRING_EXCEEDS_PERMITTED↩ _LENGTH	
ControllerError_SCOPE_PERIOD_SHOULD_↩ EXCEED_NUMBER_OF_AIN_PARAMS	
ControllerError_VALUE_IS_INCORRECT	
ControllerError_INVALID_IO_CHANNEL	
ControllerError_FLASH_VALUE_CANNOT_BE_SET	
ControllerError_DIRECTORY_NOT_LOCKED	
ControllerError_DIRECTORY_ALREADY_LOCKED	
ControllerError_PROGRAM_NOT_RUNNING_ON↩ _THIS_PROCESS	
ControllerError_PROGRAM_NOT_RUNNING	
ControllerError_PROGRAM_NOT_PAUSED_ON_↩ THIS_PROCESS	
ControllerError_PROGRAM_NOT_PAUSED	
ControllerError_COMMAND_NOT_ALLOWED_↩ WHEN_RUNNING_MOTION_PERFECT	
ControllerError_DIRECTORY_STRUCTURE_↩ INVALID	
ControllerError_DIRECTORY_IS_LOCKED	
ControllerError_CANNOT_EDIT_PROGRAM	
ControllerError_TOO_MANY_NESTED_OPERANDS	
ControllerError_CANNOT_RESET_WHEN_DRIVE↩ _SERVO_ON	
ControllerError_FLASH_STICK_BLANK	

Enumerator

ControllerError_FLASH_STICK_NOT_AVAILABLE↩ _ON_THIS_CONTROLLER	
ControllerError_SLAVE_ERROR	
ControllerError_MASTER_ERROR	
ControllerError_NETWORK_TIMEOUT	
ControllerError_NETWORK_PROTOCOL_ERROR	
ControllerError_GLOBAL_DEFINITION_IS_↩ DIFFERENT	
ControllerError_INVALID_PROGRAM_NAME	
ControllerError_PROGRAM_CORRUPT	
ControllerError_MORE_THAN_ONE_PROGRAM_↩ RUNNING_WHEN_TRYING_TO_SET_GLOBAL_↩ OR_CONSTANT	
ControllerError_PROGRAM_ENCRYPTED	
ControllerError_BASIC_TOKEN_DEFINITION_↩ INCORRECT	
ControllerError_OPEN_BRACKET_EXPECTED	
ControllerError_NUMBER_EXPECTED	
ControllerError_AS_EXPECTED	
ControllerError_DATA_TYPE_EXPECTED	
ControllerError_STRING_EXPECTED	
ControllerError_DOWNLOAD_ABORT_OR_↩ TIMEOUT	
ControllerError_CANNOT_SPECIFY_PROGRAM_↩ TYPE_FOR_AN_EXISTING_PROGRAM	
ControllerError_INVALID_COFF_IMAGE_FILE	
ControllerError_VARIABLE_DEFINED_OUTSIDE_↩ INCLUDE_FILE	
ControllerError_COMMAND_NOT_ALLOWED_↩ WITHIN_INCLUDE_FILE	
ControllerError_SERIAL_NUMBER_MUST_BE_↩ MINUS_1	
ControllerError_APPEND_BLOCK_INCONSISTENT	
ControllerError_INVALID_RANGE_SPECIFIED	
ControllerError_TOO_MANY_ITEMS_DEFINED_↩ FOR_BLOCK	
ControllerError_INVALID_MSPHERICAL_INPUT	
ControllerError_TOO_MANY_LABELS	
ControllerError_SYMBOL_TABLE_LOCKED	
ControllerError_INCORRECT_SYMBOL_TYPE	
ControllerError_VARIABLES_NOT_PERMITTED_↩ ON_COMMAND_LINE	
ControllerError_INVALID_PROGRAM_TYPE	
ControllerError_PARAMETER_EXPECTED	
ControllerError_FIRMWARE_ERROR__DEVICE_↩ IN_USE	
ControllerError_DEVICE_ERROR__TIMEOUT_↩ WAITING_FOR_DEVICE	
ControllerError_DEVICE_ERROR__COMMAND_↩ NOT_SUPPORTED_BY_DEVICE	
ControllerError_DEVICE_ERROR__CRC_ERROR	

Enumerator

ControllerError_DEVICE_ERROR__ERROR_↩ WRITING_TO_DEVICE	
ControllerError_DEVICE_ERROR__INVALID_↩ RESPONSE_FROM_DEVICE	
ControllerError_FIRMWARE_ERROR__CANNOT_↩ REFERENCE_DATA_OUTSIDE_CURRENT_BLOCK	
ControllerError_DISK_ERROR__INVALID_MBR	
ControllerError_DISK_ERROR__INVALID_BOOT_↩ SECTOR	
ControllerError_DISK_ERROR__INVALID_↩ SECTOR_CLUSTER_REFERENCE	
ControllerError_FILE_ERROR__DISK_FULL	
ControllerError_FILE_ERROR__FILE_NOT_FOUND	
ControllerError_FILE_ERROR__FILENAME_↩ ALREADY_EXISTS	
ControllerError_FILE_ERROR__INVALID_FILENAME	
ControllerError_FILE_ERROR__DIRECTORY_FULL	
ControllerError_COMMAND_ONLY_ALLOWED_↩ WHEN_RUNNING_MOTION_PERFECT	
ControllerError_HASH_EXPECTED	
ControllerError_FOR_EXPECTED	
ControllerError_INPUT__OUTPUT__APPEND__↩ FIFO_READ__FIFO_WRITE_EXPECTED	
ControllerError_FILE_NOT_OPEN	
ControllerError_END_OF_FILE	
ControllerError_FILE_ALREADY_OPEN	
ControllerError_INVALID_STORAGE_AREA	
ControllerError_INVALID_FLOATING_POINT_↩ OPERATION	
ControllerError_INVALID_SYSTEM_CODE_↩ WRONG_CONTROLLER	
ControllerError_IEC_ERROR__INVALID_↩ VARIABLE_ACCESS	
ControllerError_NUMERICAL_ERROR__NAN_USED	
ControllerError_NUMERICAL_ERROR__INF_USED	
ControllerError_NUMERICAL_ERROR__↩ SUBNORMAL_VALUE_USED	
ControllerError_MAC_EEPROM_IS_LOCKED	
ControllerError_INVALID_MIX_OF_DATA_TYPES	
ControllerError_INVALID_STARTUP_↩ CONFIGURATION_COMMAND	
ControllerError_SYMBOL_IS_NOT_A_VARIABLE	
ControllerError_KINEMATICS_RUNTIME_↩ LIMITED_TO_1_HOUR_FEC_22	
ControllerError_IEC_RUNTIME_LIMITED_TO_1_↩ HOUR_FEC_21	
ControllerError_COMMAND_NOT_ALLOWED_↩ WITH_CURRENT_ATYPE	
ControllerError_WILDCARD_LENGTH_MUST_BE_1	

Enumerator

ControllerError_INCOMPATIBLE_ARRAY_↩ DIMENSIONS	
ControllerError_MATRIX_IS_SINGULAR	
ControllerError_PROGRAM_IS_NOT_↩ EXECUTABLE_TYPE	
ControllerError_DISK_ERROR__FORMAT_MUST↩ _BE_FAT32	
ControllerError_PROGRAM_IS_STOPPED_HALT_↩ FORCED	
ControllerError_VARIABLE_NOT_DEFINED_↩ USING_DIM_OR_GLOBAL_CONSTANT	
ControllerError_VARIABLE_NOT_DEFINED_↩ USING_DIM_WITHIN_INCLUDE_FILE	
ControllerError_PARAMETER_CAN_ONLY_BE_↩ CHANGED_USING_MC_CONFIG	
ControllerError_IEC_TASK_MEMORY_↩ ALLOCATION_ERROR	
ControllerError_IEC_IO_BINDING_INDEX_OUT_↩ OF_RANGE	
ControllerError_IEC_VR_BINDING_INDEX_OUT_↩ OF_RANGE	
ControllerError_IEC_TABLE_BINDING_INDEX_↩ OUT_OF_RANGE	
ControllerError_FAILED_OPEN_CONNECTION	
ControllerError_MCCONFIG_CHANGED_EX_↩ REQUIRED	
ControllerError_NO_RESPONSE_FROM_CAN_↩ MODULE	
ControllerError_INVALID_FPGA_VERSION	
ControllerError_VARIABLE_NOT_CREATED_↩ USING_GLOBAL_CONSTANT	
ControllerError_PARAMETER_CAN_ONLY_BE_↩ CHANGED_USING_MC_CONFIG_OR_↩ COMMAND_LINE	
ControllerError_ETHERCAT_AXES_CANNOT_BE_↩ SET_USING_ATYPE_PLEASE_USE_DRIVE_MODE	
ControllerError_BIT_NUMBER_OUT_OF_RANGE	
ControllerError_FUNCTION_NAME_EXPECTED	
ControllerError_FUNCTION_MUST_BE_DEFINED↩ _WITHIN_A_BASIC_LIBRARY	
ControllerError_TOO_MANY_FUNCTIONS_↩ DEFINED	
ControllerError_MISSING_ENDFUNC	
ControllerError_ENDFUNC_WITHOUT_↩ PREVIOUS_FUNCTION	
ControllerError_PARAMETER_NAME_EXPECTED	
ControllerError_DEVICE_ERROR_NO_SDCARD_↩ INSERTED	
ControllerError_DEVICE_ERROR_SDCARD_↩ WRITE_PROTECTED	
ControllerError_FAILED_ATTEMPT_1	
ControllerError_FAILED_ATTEMPT_2	

Enumerator

ControllerError_FAILED_TOO_MANY_TIMES_↩ PLEASE_RESET	
ControllerError_LINE_MUST_BE_INSIDE_↩ FUNCTION	
ControllerError_DUPLICATE_NAME_USED_IN_↩ LABEL_AND_VARIABLE	
ControllerError_STRUCTURE_FIELD_NAME_↩ NOT_RECOGNISED	
ControllerError_ID_NOT_DEFINED	
ControllerError_ROBOT_CONFIGURATION_↩ CANNOT_BE_CHANGED_WHILE_MOVEL	
ControllerError_TOOL_OFFSET_ID_NOT_DEFINED	
ControllerError_OBJECT_FRAME_ID_NOT_↩ DEFINED	
ControllerError_ROBOT_FRAME_ID_NOT_DEFINED	
ControllerError_COMMAND_NOT_PERMITTED_↩ IN_FUNCTION	
ControllerError_TOO_MANY_FUNCTION_↩ PARAMETERS	
ControllerError_OPERATION_CANNOT_BE_↩ PERFORMED_UNTIL_IDLE	
ControllerError_SINGULARITY	
ControllerError_TOOL_CANNOT_BE_MODIFIED_↩ OR_DELETED_WHILE_SELECTED	
ControllerError_OBJECT_FRAME_CANNOT_BE_↩ MODIFIED_OR_DELETED_WHILE_SELECTED	
ControllerError_ROBOT_FRAME_CANNOT_BE_↩ MODIFIED_OR_DELETED_WHILE_SELECTED	
ControllerError_ROBOT_MOVES_CANNOT_BE_↩ PERFORMED_UNTIL_JOG_OPERATION_IS_↩ DISABLED	
ControllerError_CYCLIC_INCLUDE_DEPENDENCY	
ControllerError_OBJECT_FRAME_ALREADY_↩ ATTACHED	
ControllerError_OBJECT_FRAME_HAS_NOT_↩ BEEN_ATTACHED	
ControllerError_OBJECT_FRAME_ALREADY_↩ SYNCED	
ControllerError_BASE_AXIS_DOES_NOT_↩ BELONG_TO_ANY_KINEMATIC_GROUP	
ControllerError_SYNC_IN_PROCESS	
ControllerError_ROBOT_PROGRAM_STOPPED	
ControllerError_TARGET_POINT_NOT_ACTIVATED	
ControllerError_PROGRAM_FAILED_TO_RUN	
ControllerError_RETURN_STATEMENT_↩ EXPECTED_BEFORE_ENDFUNC	
ControllerError_CONTINUE_WITHOUT_↩ PREVIOUS_FOR_WHILE_OR_REPEAT	
ControllerError_EXIT_WITHOUT_PREVIOUS_↩ FOR_WHILE_OR_REPEAT	
ControllerError_TOO_MANY_EXIT_STATEMENTS↩ _WITHIN_SINGLE_BLOCK	

Enumerator

ControllerError_CASE_OR_END_CASE_↩ WITHOUT_PREVIOUS_SELECT_CASE	
ControllerError_FIXED_INTEGER_OR_ELSE_↩ EXPECTED	
ControllerError_TOO_MANY_CASE_OPTIONS	
ControllerError_ROBOTICS_RUNTIME_LIMITED_↩ TO_1_HOUR	
ControllerError_COLLISION_OBJECTS_CANNOT↩ _BE_MODIFIED_OR_DELETED_WHILE_↩ SELECTED	
ControllerError_TOO_MANY_COLLISION_↩ OBJECTS_ACTIVE	
ControllerError_ATYPE_MUST_BE_↩ CONFIGURED_USING_MC_CONFIG_↩ PARAMETER_DRIVE_MODE	
ControllerError_COMMAND_NOT_ALLOWED_IN_↩ FRAME	
ControllerError_COMMAND_LINE_TOO_LONG	
ControllerError_TABLE_LOCATION_WRITE_↩ PROTECTED	
ControllerError_ENCRYPTION_AUTHORITY_NOT↩ _FOUND	
ControllerError_NO_FREE_ENCRYPTION_↩ AUTHORITY	
ControllerError_LOCAL_TARGET_VALUE_HAS_↩ NOT_BEEN_DEFINED	
ControllerError_COMMAND_NOT_PERMITTED_↩ IN_BASIC_LIBRARY	
ControllerError_INVALID_MOVEC_INPUT	
ControllerError_FUNCTION_EXPECTED	
ControllerError_CAM_TABLE_ID_NOT_DEFINED	
ControllerError_RUNNING_OF_RPS_↩ PROGRAMS_INHIBITED_USE_DISABLE_ROBOT	
ControllerError_U_AND_V_ORIENTATION_↩ INHIBITED_WHILE_WRIST_SINGULARITY	
ControllerError_TARGET_POINT_OUT_OF_↩ ROBOT_WORKING_AREA	
ControllerError_AXIS_LOCKED_TO_A_ROBOT	
ControllerError_ROBOT_CONTROLLED_BY_↩ ANOTHER_PROCESS	
ControllerError_ROBOT_NOT_DEFINED_USING_↩ KINEMATIC_GROUP	
ControllerError_COMMAND_INHIBITED_ROBOT_↩ MUST_BE_USED_TO_ACTIVATE_A_KINEMATIC_↩ GROUP	
ControllerError_UNDEFINED_VARIABLE_IN_DIM↩ _INITIALISER	
ControllerError_THE_SPECIFIED_ROBOT_↩ CONFIGURATION_IS_NOT_ALLOWED	
ControllerError_TOO_MANY_AXES_ACTIVATED	
ControllerError_MULTIPLE_MOVE_TRIGGERS_↩ SPECIFIED	

Enumerator

ControllerError_MOVE_TRIGGER_RESPONSE_↩ NOT_SPECIFIED	
ControllerError_MOVE_TRIGGER_NOT_SPECIFIED	
ControllerError_INVALID_ARRAY_DIMENSION	
ControllerError_DUPLICATE_VR_INDEX	
ControllerError_CONSTANT_VALUE_EXPECTED	
ControllerError_RPS_KERNEL_V2_NOT_ENABLED	
ControllerError_INCLUDE_FILE_MUST_CONTAIN↩ _DEFCONST_DIM_STATEMENTS_ONLY	
ControllerError_TARGET_TAUGHT_WITH_↩ ANOTHER_ROBOT	
ControllerError_TARGET_ROBOT_CONFIG_↩ MISMATCH	
ControllerError_TARGET_TOOL_OFFSET_↩ MISMATCH	
ControllerError_TARGET_OBJECT_FRAME_↩ MISMATCH	
ControllerError_TARGET_ROBOT_FRAME_↩ MISMATCH	
ControllerError_INVALID_FRAME_NUMBER	
ControllerError_FRAME_NOT_SUPPORTED_BY_↩ RPS	
ControllerError_FORWARD_TRANSFORM_↩ FUNCTION_NOT_DEFINED	
ControllerError_INVERSE_TRANSFORM_↩ FUNCTION_NOT_DEFINED	
ControllerError_INVALID_MOVEOPTION	
ControllerError_MAX_OPEN_KEYWORDS_↩ REACHED	
ControllerError_KEYWORD_INACTIVE	
ControllerError_KEYWORD_FUNCTION_NOT_↩ DEFINED	
ControllerError_KEYWORD_SAME_AS_SYSTEM↩ _WORD	
ControllerError_UNEXPECTED_CHARACTERS_↩ AFTER_LINE_CONTINUATION	
ControllerError_THIS_COMMAND_CAN_BE_↩ EXECUTED_ONLY_ONCE	
ControllerError_TABLE_MEMORY_CONFLICT	
ControllerError_MAX_OPEN_MOVES_REACHED	
ControllerError_USER_DEFINED_ERROR	
ControllerError_INVALID_FORMAT_SPECIFIER	
ControllerError_OPEN_MOVE_FUNCTIONS_NOT↩ _DEFINED	
ControllerError_TOO_MANY_CHARS_IN_STR	
ControllerError_INVALID_OPEN_ARCH_FUNC_↩ DEFINITION	
ControllerError_PROCESS_IN_USE	
ControllerError_VR_LOCATION_WRITE_↩ PROTECTED	

Definition at line 15 of file ITrioConnection.cs.

5.2.1.11 EthercatStartOption

```
enum Trio.UnifiedApi.EthercatStartOption : UInt32
```

EtherCAT start option.

Enumerator

None	No option is selected.
PrintOutProgress	Print out progress information on - Print out progress information (default) off - Do not print progress information.
ForceRescan	Network re-scan off - Normal state change (default) on - Force a re-scan of the network.
ExpressStartup	Start-up type on - Express start-up; do not wait for PLL to lock between state changes off - Standard start-up.

Definition at line 907 of file ITrioConnection.cs.

5.2.1.12 EthercatState

```
enum Trio.UnifiedApi.EthercatState
```

EtherCAT state.

Enumerator

Initial	Initial.
PreOprational	Pre operational.
SafeOperational	Safe operational.
Operational	Operational.

Definition at line 880 of file ITrioConnection.cs.

5.2.1.13 EventType

```
enum Trio.UnifiedApi.EventType
```

Events that may occur on a connection.

Enumerator

Error	Error processing a command.
Warning	Warning for non critical problem.
Message	Information about the command being processed.

Definition at line 719 of file ITrioConnection.cs.

5.2.1.14 FileFlags

```
enum Trio.UnifiedApi.FileFlags : UInt32
```

Flags related to files and programs in internal storage on the Motion Coordinator.

Enumerator

None	No flags are set.
Editable	If set file is editable on the controller.
Autorun	If set program can be auto-run.

Definition at line 1113 of file ITrioConnection.cs.

5.2.1.15 FileTransferOption

```
enum Trio.UnifiedApi.FileTransferOption : UInt32
```

File transfer option.

Enumerator

None	No option is selected.
CompressOnly	The data gets compressed and transferred. The other side receives and stores the compressed data.
CompressDecompress	The data gets compressed, transferred and decompressed. This saves bandwidth.

Definition at line 940 of file ITrioConnection.cs.

5.2.1.16 FileType

```
enum Trio.UnifiedApi.FileType : Int32
```

File types on the Motion Coordinator.

Enumerator

Undefined	The file type is not recognized.
BASIC	TrioBASIC program.
EncryptedBASIC	Encrypted TrioBASIC program.
GCode	GCode program.
TextFile	Unknown contents.
IecProgram	IEC program.
FifoFile	FIFO (not valid in project)
TempFile	Temporary File (not valid in project)
Project	Project file.
Binary	Container for anything that the controller does not understand.
MCConfig	Special for the MC_CONFIG file.

Enumerator

IecLibrary	IEC library.
HmiDesign	HMI.
BASICLibrary	Trio BASIC library.
EncryptedBASICibrary	Encrypted Trio BASIC library.
HmiLibrary	HMI Library.
TempVolatileFile	Volatile temporary controller file stored in RAM (not valid in project)
Robot	Robot program.
RobotBasic	Robot basic program.
TempTextFile	Temporary Text File (not valid in project)
RobotLibrary	Robot library.
RobotBasicLibrary	Robot basic library.
Recipe	Recipe data.
Alarms	Alarm data.
RealtimeBASICLibrary	Real-time basic library.
EncryptedRealtimeBasicLibrary	Encrypted real-time basic library.
EncryptedIecProgram	Encrypted IEC program.
EncryptedIecLibrary	Encrypted IEC library.
__Size	Not a file type. Used to size the FileType enumeration.

Definition at line 961 of file ITrioConnection.cs.

5.2.1.17 MotionType

```
enum Trio.UnifiedApi.MotionType : UInt32
```

Motion types returned by MTYPE and NTYPE axis parameters.

Enumerator

IDLE	
MOVE	
MOVEABS	
MHELICAL	
MOVECIRC	
MOVEMODIFY	
MOVESP	
MOVEABSSP	
MOVECIRCSP	
MHELICALSP	
FORWARD	
REVERSE	
DATUM	
CAM	
FORWARD_JOG	
REVERSE_JOG	
WETCAM	
CAMBOX	
CONNECT	

Enumerator

MOVELINK	
CONNPATH	
FLEXLINK	
CAMLINK	
COMPENSATE_XY	
PICKLINK	
CAMIN	
MOVETANG	
MSPHERICAL	
MTRANSCIRC	
MOVESEQ	
MOVEABSSEQ	
MOVETURN	
MOVEPICK	
MOVECONTOUR	
MOVEDISPENSE	
MSPHERICALSP	
MTRANSCIRCSP	
MOVESEQSP	
MOVEABSSEQSP	
MOVEJ	
MOVEL	
MOVEC	
MOVEBLEND	
MOVEJREL	
MOVELREL	
MOVETEST	
MOVEP	
MOVEU	
MOVEB	
MOVESPCONT	
MOVEABSSPCONT	
GEARIN	
ACCPROFILE	
VELPROFILE	
POSPROFILE	
COMBINEAXES	
OPENMOVE_BASE	
OPENMOVE_1	
OPENMOVE_2	
OPENMOVE_3	
OPENMOVE_4	
OPENMOVE_5	

Definition at line 1647 of file ITrioConnection.cs.

5.2.1.18 ParamBlockType

```
enum Trio.UnifiedApi.ParamBlockType
```

Types of connections available.

Enumerator

AxisParam	Axis parameters.
SystemParameters	System parameters.
VR	VR values.
TABLE	TABLE values.
ProgramVariable	Program variables.

Definition at line 788 of file ITrioConnection.cs.

5.2.1.19 ProcessStatus

`enum Trio.UnifiedApi.ProcessStatus : UInt32`

Enumerator

STOPPED	
READY	
STEPPING	
PAUSED	
PAUSING	
ABORTED	
SUSPEND	
SUSPENDED	
SLEEPING	

Definition at line 1890 of file ITrioConnection.cs.

5.2.1.20 ProgressEvent

`enum Trio.UnifiedApi.ProgressEvent`

Progress event.

Enumerator

Start	Progress started.
Progress	Progress.
Finish	Progress finished.

Definition at line 767 of file ITrioConnection.cs.

5.2.1.21 PSwitchMode

`enum Trio.UnifiedApi.PSwitchMode : UInt32`

Flags PSWITCH command in single/cyclic mode.

Enumerator

None	No flags are set.
Invert	If set - set position should be bigger than reset position If not set - set position should be less than reset position.
DPOS	If set DPOS is used If not set MPOS is used.
Mask	Mask of allowed bits.

Definition at line 1151 of file ITrioConnection.cs.

5.2.1.22 PSwitchMultiMode

`enum Trio.UnifiedApi.PSwitchMultiMode : UInt32`

Flags PSWITCH command in multi/sequence mode.

Enumerator

None	No flags are set.
DPOS	If set DPOS is used If not set MPOS is used.
TimerMode	If set timer mode If not set set and reset positions mode.
NegativeDirection	If set negative direction If not set positive direction.
ZOut	If set Z output mode If not set digital output mode.
Mask	Mask of allowed bits.

Definition at line 1180 of file ITrioConnection.cs.

5.2.1.23 RobotStatus

`enum Trio.UnifiedApi.RobotStatus : UInt64`

Enumerator

RS_WORLD_FS_LIMIT	
RS_WORLD_RS_LIMIT	
RS_ROBOT_FS_LIMIT	
RS_ROBOT_RS_LIMIT	
RS_TCP_FS_LIMIT	
RS_TCP_RS_LIMIT	
RS_FE_WARNING	
RS_FE_EXCEEDS_LIMIT	
RS_WRIST_SINGULARITY	
RS_ALIGNMENT_SINGULARITY	
RS_ELBOW_SINGULARITY	
RS_MAX_SPEED_LIMIT	
RS_COLLIDED	
RS_NOT_IN_POS	
RS_ELBOW_CLOSED_SINGULARITY	
RS_DISALLOW_JOG	

Definition at line 1855 of file ITrioConnection.cs.

5.2.1.24 SyncControl

```
enum Trio.UnifiedApi.SyncControl :  Int32
```

Enumerator

SC_WAITING_TO_START	
SC_NO_SYNC	
SC_STARTING_SYNC	
SC_PERFORMING_SYNC	
SC_SYNCHRONISED	
SC_STOPPING_SYNC	
SC_STARTING_INTERP_MOVE	
SC_PERFORMING_INTERP_MOVE	
SC_STARTING_RESYNC	
SC_PERFORMING_RESYNC	
SC_STARTING_RESYNC_TO_USERFRAME	
SC_PERFORMING_RESYNC_TO_USERFRAME	

Definition at line 1911 of file ITrioConnection.cs.

5.2.2 Function Documentation

5.2.2.1 ConnectionEventHandler()

```
delegate void Trio.UnifiedApi.ConnectionEventHandler (
            EventType eventType,
            long integerData,
            string stringData )
```

Delegate for connection related events.

5.2.2.2 FileCallback()

```
delegate void Trio.UnifiedApi.FileCallback (
            ref FileInfo file,
            string fileName )
```

Delegate for processing the response of Dir command.

5.2.2.3 ProgressHandlerCallback()

```
delegate void Trio.UnifiedApi.ProgressHandlerCallback (
            ref ProgressInfo progress )
```

Delegate for streaming back the response of commands that return string data.

5.2.2.4 StreamHandlerCallback()

```
unsafe delegate void Trio.UnifiedApi.StreamHandlerCallback (
            ref DataStreamPacket packet,
            byte * buff )
```

Delegate for streaming back the response of commands that return string data.

Chapter 6

Class Documentation

6.1 Trio.UnifiedApi.DataStreamPacket Struct Reference

Data used with the delegate for streaming back the response of commands that return string data.

Public Attributes

- uint total_length

  The length of the stream, if available. 0 otherwise.
- uint packet_number

  The packet number.
- uint packet_count

  The total count of the packets of the stream, if available. 0 otherwise.
- uint data_length

  The number of bytes.

6.1.1 Detailed Description

Data used with the delegate for streaming back the response of commands that return string data.

Definition at line 1986 of file ITrioConnection.cs.

6.1.2 Member Data Documentation

6.1.2.1 data_length

```
uint Trio.UnifiedApi.DataStreamPacket.data_length
```

The number of bytes.

Definition at line 2003 of file ITrioConnection.cs.

6.1.2.2 packet_count

```
uint Trio.UnifiedApi.DataStreamPacket.packet_count
```

The total count of the packets of the stream, if available. 0 otherwise.

Definition at line 1999 of file ITrioConnection.cs.

6.1.2.3 packet_number

```
uint Trio.UnifiedApi.DataStreamPacket.packet_number
```

The packet number.

Definition at line 1995 of file ITrioConnection.cs.

6.1.2.4 total_length

```
uint Trio.UnifiedApi.DataStreamPacket.total_length
```

The length of the stream, if available. 0 otherwise.

Definition at line 1991 of file ITrioConnection.cs.

The documentation for this struct was generated from the following file:

- ITrioConnection.cs

6.2 Trio.UnifiedApi.FileInfo Struct Reference

Data used with the delegate for processing the response of Dir command.

Public Attributes

- FileType type
  Type of the file on the controller.
- uint size
  Text code size.
- uint compiled_size
  Compiled binary size.
- int autorun_process
  Autorun process number.
- FileFlags flags
  Flags related to files and programs on the controller.

6.2.1 Detailed Description

Data used with the delegate for processing the response of Dir command.

Definition at line 1953 of file ITrioConnection.cs.

6.2.2 Member Data Documentation

6.2.2.1 autorun_process

```
int Trio.UnifiedApi.FileInfo.autorun_process
```

Autorun process number.

Definition at line 1970 of file ITrioConnection.cs.

6.2.2.2 compiled_size

```
uint Trio.UnifiedApi.FileInfo.compiled_size
```

Compiled binary size.

Definition at line 1966 of file ITrioConnection.cs.

6.2.2.3 flags

```
FileFlags Trio.UnifiedApi.FileInfo.flags
```

Flags related to files and programs on the controller.

Definition at line 1974 of file ITrioConnection.cs.

6.2.2.4 size

```
uint Trio.UnifiedApi.FileInfo.size
```

Text code size.

Definition at line 1962 of file ITrioConnection.cs.

6.2.2.5 type

```
FileType Trio.UnifiedApi.FileInfo.type
```

Type of the file on the controller.

Definition at line 1958 of file ITrioConnection.cs.

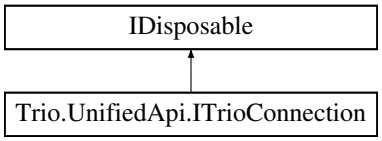
The documentation for this struct was generated from the following file:

- ITrioConnection.cs

6.3 Trio.UnifiedApi.ITrioConnection Interface Reference

Connection interface.

Inheritance diagram for Trio.UnifiedApi.ITrioConnection:



Public Member Functions

- void SetTcpHostName (String host)

  Set host name or IP to connection object. Valid only for TCP connection.
- String GetTcpHostName ()

  Get current host name or IP of connection object. Valid only for TCP connection.
- void SetTcpPort (UInt16 port)

  Set TCP port to connection object. Valid only for TCP connection.
- UInt16 GetTcpPort ()

  Get current TCP port from connection object. Valid only for TCP connection.
- void SetTcpTimeout (UInt32 timeout_ms)

  Set TCP connection timeout. Valid only for TCP connection.
- UInt32 GetTcpTimeout ()

  Get current TCP connection timeout. Valid only for TCP connection.
- Version GetApiVersion ()

  Get Unified API version info.
- String GetApiVersionStr ()

  Get Unified API version string.
- double GetFirmwareVersionNumber ()

  Get the firmware version of the Motion Coordinator.
- void OpenConnection ()

  Open a connection to the Motion Coordinator.
- void CloseConnection ()

  Closes previously open connection to the Motion Coordinator.
- void Ex (Int32 mode=0)

  Perform software reset on Motion Coordinator.
- void Cancel (Int32 mode, Int32 axis=-1)

  Used to cancel current or buffered axis commands on an axis or an interpolating axis group.
- void Connect (double ratio, Int32 linkAxis, ConnectMode mode=ConnectMode.MPOS, Int32 axis=-1)

  Links the demand position of the base axis to the measured movements of the driving axes to produce an electronic gearbox.
- void ConnPath (double ratio, Int32 linkAxis, Int32 axis=-1)

  Enables you to link to the path of an interpolated movement by linking the demand position of the base axis, to the interpolated path distance of the driving axis.
- void PSwitch (int switchNumber, PSwitchMode mode, Int32 axis, UInt32 outputNumber, bool outputState, double setPosition, double resetPosition)

  The PSWITCH command allows an output to be set when a predefined position is reached, and to be reset when a second position is reached.
- void PSwitchOff (int switchNumber, bool hold=false)

The PSWITCHOFF command is used to disable.

- void PSwitchMulti (int switchNumber, PSwitchMultiMode mode, Int32 axis, UInt32 outputNumber, bool outputState, UInt32 tableStart, UInt32 tableEnd, UInt32 pulseTime=0)

  The PSWITCHMULTI command allows an output to be set that turns the output on and off at selected positions loaded from the TABLE memory.
- void Execute (String command)

  Execute arbitrary command.
- unsafe void ExecuteWithResponse (String command, Byte *buff, ref UInt32 buff_size)

  Execute arbitrary command with response.
- void ExecuteWithResponse (String command, StreamHandlerCallback handler)

  Execute arbitrary command with response.
- String ExecuteWithResponse (String command)

  Execute arbitrary command with response.
- void SetAxisBase (Int32 axis)

  The BASE command is used to direct all subsequent motion commands and axis parameter read/writes to a particular axis, or group of axes. The default setting is a sequence: 0, 1, 2, 3...
- void SetMultiAxisBase (Int32[ ] axes)

  Set multiple base axes.
- void Datum (UInt32 sequence, Int32 axis=-1)

  Performs one of 6 homing sequences to locate an axis to an absolute position.
- void DefPos (double pos, Int32 axis=-1)

  Defines the current axis position as a new absolute value.
- void DefPosMulti (double[ ] pos, Int32 axis=-1)

  Defines the current axis position as a new absolute value to multiple axes.
- void Forward (Int32 axis=-1)

  Sets continuous forward movement.
- void Reverse (Int32 axis=-1)

  Sets continuous reverse movement.
- void MoveRel (double pos, Int32 axis=-1)

  Incremental move of single axis.
- void MoveRelMulti (double[ ] dist, Int32 axis=-1)

  Incremental move of multiple axes.
- void MoveRelSp (double pos, Int32 axis=-1)

  Incremental move of single axis with speed changed between moves.
- void MoveRelSpMulti (double[ ] dist, Int32 axis=-1)

  Incremental move of multiple axes with speed changed between moves.
- void MoveAbs (double pos, Int32 axis=-1)

  Absolute positional move of single axis.
- void MoveAbsMulti (double[ ] pos, Int32 axis=-1)

  Absolute positional move of multiple axes.
- void MoveAbsSp (double pos, Int32 axis=-1)

  Absolute positional move of single axis with speed changed between moves.
- void MoveAbsSpMulti (double[ ] pos, Int32 axis=-1)

  Absolute positional move of multiple axes with speed changed between moves.
- void MoveCirc (Int32 axis, double finishBase, double finishNext, double centreBase, double centreNext, UInt32 direction, double transitionAngle=0, UInt32 tableOutput=unchecked((UInt32)(-1)))

  Moves 2 orthogonal axes in such a way as to produce a circular arc at the tool point. The length and radius of the arc are defined by the five parameters in the command line. The move parameters are always relative to the end of the last specified move. This is the start position on the circle circumference. Axis 1 is the current BASE axis. Axis 2 is the next axis in the BASE array. The first 4 distance parameters are scaled according to the current unit conversion factor for the BASE axis.
- void MoveCircSp (Int32 axis, double finishBase, double finishNext, double centreBase, double centreNext, UInt32 direction, double transitionAngle=0, UInt32 tableOutput=0)

Works as MoveCirc and additionally allows vector speed to be changed when using multiple moves in the look ahead buffer when MERGE = ON , using additional parameters FORCE_SPEED and ENDMOVE_SPEED. Optional parameters ta and output allow the use of transition curves.

- void MoveHelical (lnt32 axis, double finishBase, double finishNext, double centreBase, double centreNext, UInt32 direction, double linearDistance, UInt32 mode, UInt32 transitionAngle=0, UInt32 tableOutput=0)

  Performs a helical move. Moves 2 orthogonal axes in such a way as to produce a circular arc at the tool point with a simultaneous linear move on a third axis. The first 5 parameters are similar to those of an MOVECIRC command.The sixth parameter defines the simultaneous linear move.
- void MoveHelicalSp (lnt32 axis, double finishBase, double finishNext, double centreBase, double centreNext, UInt32 direction, double linearDistance, UInt32 mode, UInt32 transitionAngle=0, UInt32 tableOutput=0)

  Performs a helical move the same as MHELICAL and additionally allows vector speed to be changed when using multiple moves in the buffer. Uses additional axis parameters FORCE_SPEED, ENDMOVE_SPEED and STARTMOVE↩ _SPEED.
- void MoveSpherical (lnt32 axis, double endMidX, double endMidY, double endMidZ, double centreX, double centreY, double centreZ, UInt32 mode, UInt32 gtpi=0, double rotau=0, double rotav=0, double rotaw=0, double rotap=0)

  Moves the three axis group defined in BASE along a spherical path with a vector speed determined by the SPEED set in the first axis of the BASE array. There are 4 modes of operation with the option of finishing the move at an endpoint different to the start, or returning to the start point to complete a circle. The path of the movement in 3D space can be defined either by specifying a point somewhere along the path, or by specifying the centre of the sphere.
- void MoveSphericalSp (lnt32 axis, double endMidX, double endMidY, double endMidZ, double centreX, double centreY, double centreZ, UInt32 mode, UInt32 gtpi=0, double rotau=0, double rotav=0, double rotaw=0, double rotap=0)

  Performs a spherical move the same as MSPHERICAL and additionally allows vector speed to be changed when using multiple moves in the look ahead buffer when MERGE = ON, using additional parameters FORCE_SPEED, ENDMOVE_SPEED and STARTMOVE_SPEED.
- void MoveRelSeq (lnt32 axis, UInt32 tablePointer, UInt32 nAxes, UInt32 nPoints, UInt32 options, double radius, UInt32 tableOutput=0, double transitionAngle=0)

  Function allows a sequence of relative movements to be loaded via TABLE values. The moves can be automatically merged together using a circular or spherical arc. Movement on 2 to 6 axes may be controlled in this way.
- void MoveAbsSeq (lnt32 axis, UInt32 tablePointer, UInt32 nAxes, UInt32 nPoints, UInt32 options, double radius, UInt32 tableOutput=0, double transitionAngle=0)

  Function allows a sequence of absolute movements to be loaded via TABLE values. The moves can be automatically merged together using a circular or spherical arc. Movement on 2 to 6 axes may be controlled in this way.
- void MoveModify (double position, lnt32 axis=-1)

  MOVEMODIFY will change the absolute end position of a single axis MOVE , MOVEABS , MOVESP , MOVEABSSP or MOVEMODIFY that is in the last position in the movement buffer. If there is no motion command in the movement buffers or the last movement is not a single axis linear move, then MOVEMODIFY is loaded.
- void MovePick (lnt32 axis, UInt32 mode, double x, double y, double z, double withdraw, double overlap↩ ControlA=0.5, double overlapControlB=0.5, double u=0, double v=0, double w=0, double withdrawControl=1, double xyControl=1, double approachControl=1)

  MOVEPICK is a new move type aimed at making it much simpler to program fast pick and place operations.
- void MoveLink (lnt32 axis, double distance, double link_dist, double link_acc, double link_dec, int link_axis=-1, UInt32 link_options=0, double link_pos=0, double base_dist=0, double y_dist=0, double z_dist=0, double u_dist=0, double v_dist=0, double w_dist=0)

  The linked move command is designed for controlling movements such as:
- void SetMotion (lnt32 axis, double speed, double accel, double decel, double jerk)

  Function used to pre-set the parameter list for a subsequent move using just one command.
- void SetMotion (lnt32 axis, double speed, double accel, double decel)

  Function used to pre-set the parameter list for a subsequent move using just one command.
- void SetMotion (lnt32 axis, double speed, double accel)

  Function used to pre-set the parameter list for a subsequent move using just one command.
- void SetMotion (lnt32 axis, double speed)

  Function used to pre-set the parameter list for a subsequent move using just one command.
- void SetMotionSp (lnt32 axis, double forceSpeed, double startMoveSpeed, double endMoveSpeed, double forceAccel, double forceDecel, double forceJerk, double forceRamp, double forceDwell)

SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP, MOVECIRCSP etc. using just one command.

- void SetMotionSp (Int32 axis, double forceSpeed, double startMoveSpeed, double endMoveSpeed, double forceAccel, double forceDecel, double forceJerk, double forceRamp)

  SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP, MOVECIRCSP etc. using just one command.
- void SetMotionSp (Int32 axis, double forceSpeed, double startMoveSpeed, double endMoveSpeed, double forceAccel, double forceDecel, double forceJerk)

  SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP, MOVECIRCSP etc. using just one command.
- void SetMotionSp (Int32 axis, double forceSpeed, double startMoveSpeed, double endMoveSpeed, double forceAccel, double forceDecel)

  SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP, MOVECIRCSP etc. using just one command.
- void SetMotionSp (Int32 axis, double forceSpeed, double startMoveSpeed, double endMoveSpeed, double forceAccel)

  SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP, MOVECIRCSP etc. using just one command.
- void SetMotionSp (Int32 axis, double forceSpeed, double startMoveSpeed, double endMoveSpeed)

  SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP, MOVECIRCSP etc. using just one command.
- void SetMotionSp (Int32 axis, double forceSpeed, double startMoveSpeed)

  SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP, MOVECIRCSP etc. using just one command.
- void SetMotionSp (Int32 axis, double forceSpeed)

  SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP, MOVECIRCSP etc. using just one command.
- void SetTableValue (UInt32 index, double value)

  Set single table value.
- double GetTableValue (UInt32 index)

  Get single table value.
- void SetMultiTableValues (UInt32 start_index, UInt32 count, double[ ] values)

  Set multiple table entries.
- double[ ] GetMultiTableValues (UInt32 start_index, UInt32 count)

  Get multiple table entries.
- void SetVrValue (UInt32 index, double value)

  Set single VR value.
- double GetVrValue (UInt32 index)

  Get single VR value.
- void SetMultiVrValues (UInt32 start_index, UInt32 count, double[ ] values)

  Set multiple VR entries.
- double[ ] GetMultiVrValues (UInt32 start_index, UInt32 count)

  Get multiple VR entries.
- void SetInvertIN (UInt32 input, bool invert)

  The INVERT_IN command allows the input channels to be individually inverted in software.
- bool GetInvertIN (UInt32 input)

  Gets the invert status set by INVERT_IN command.
- void RapidStop (Int32 mode=0)

  The RAPIDSTOP command operates like the CANCEL command but does so for ALL axes using a subset of cancel modes.
- void Dir (FileCallback handler)

  Function returns the list of programs and files in internal memory of the Motion Coordinator.
- void SetVrString (UInt32 vr, String str)

Set a null-terminated string into consecutive VRs.

- UInt32 GetVrStringLen (UInt32 vr)

  Get the length of a null-terminated string from consecutive VRs.
- unsafe void GetVrString (UInt32 vr, Byte *buff, ref UInt32 buff_size)

  Get a null-terminated string from consecutive VRs.
- void GetVrString (UInt32 vr, StreamHandlerCallback handler)

  Get a null-terminated string from consecutive VRs using callback function to re-construct the string from received chunks.
- String GetVrString (UInt32 vr)

  Get a null-terminated string from consecutive VRs.
- Int32 GetAIn (UInt32 channel)

  Get a value from an Analogue Input.
- Int32 GetAOut (UInt32 channel)

  Get the state of the Analogue Output.
- void SetAOut (UInt32 channel, Int32 value)

  Set the state of the Analogue Output.
- UInt32 GetDIn (UInt32 firstChannel, UInt32 lastChannel)

  Get the state of the Digital Inputs.
- bool GetDIn (UInt32 channel)

  Get the state of the Digital Input.
- UInt32 GetDIn32 ()

  Get the state of the first 32 Digital Inputs.
- UInt32 GetDOut (UInt32 firstChannel, UInt32 lastChannel)

  Get the state of the Digital Outputs.
- bool GetDOut (UInt32 channel)

  Get the state of the Digital Output.
- UInt32 GetDOut32 ()

  Get the state of the first 32 Digital Outputs.
- void SetDOut (UInt32 firstChannel, UInt32 lastChannel, UInt32 states)

  Set the state of the Digital Outputs.
- void SetDOut (UInt32 channel, bool state)

  Set the state of the Digital Output.
- void SetDOut32 (UInt32 states32)

  Set the state of the first 32 Digital Outputs.
- void StopProgram (String name)

  Stops a program at its current line.
- void StopProgram (String name, Int32 process_number)

  Stops a program at its current line.
- void AddDac (Int32 axis, Int32 secondAxis)

  Adds the output from the servo control block of a secondary axis to the output of the base axis. The resulting DAC↩ _OUT of the base axis is then the sum of the two control loop outputs.
- void AddDac (Int32 secondAxis)

  Adds the output from the servo control block of a secondary axis to the output of the base axis. The resulting DAC↩ _OUT of the base axis is then the sum of the two control loop outputs. Axis previously set by SetAxisBase() is subject to superimposition of DAC_OUT value.
- void AddAx (Int32 axis, Int32 linkAxis)

  Function is used to superimpose 2 or more movements to build up a more complex movement profile.
- void AddAx (Int32 linkAxis)

  Function is used to superimpose 2 or more movements to build up a more complex movement profile. Axis previously set by SetAxisBase() is subject to superimposition.
- void ClearVariables ()

Sets all global (numbered) variables and VR values to 0 and sets local variables on the process on which command is run to 0.

- void DeleteFrameGroup (UInt32 group)

  Delete frame group. When the group is deleted FRAME is set to 0. See the TrioBASIC FRAME_GROUP documentation for more information.
- void SetFrameGroup (UInt32 group, UInt32 tableOffset, Int32[ ] axes)

  Define new frame group.
- Tuple< UInt32, Int32[ ]> GetFrameGroup (UInt32 group)

  Get frame group info.
- void Ethercat_StartNetwork (Int32 slot, EthercatStartOption options, UInt32 MAC_retries)

  Initialise EtherCAT network, and put it onto operational mode. See Trio BASIC help for more details.
- void Ethercat_StartNetwork (Int32 slot, EthercatStartOption options)

  Initialise EtherCAT network, and put it onto operational mode. See Trio BASIC help for more details.
- void Ethercat_StartNetwork (Int32 slot)

  Initialise EtherCAT network, and put it onto operational mode. See Trio BASIC help for more details.
- void Ethercat_StopNetwork (Int32 slot)

  Closedown the EtherCAT network.
- void Ethercat_CheckSlaveOnline_VR (Int32 slot, Int32 slave, UInt32 out_vr_index)

  Check whether given slave is online. See Trio BASIC help for more details.
- bool Ethercat_CheckSlaveOnline (Int32 slot, Int32 slave)

  Check whether given slave is online. See Trio BASIC help for more details.
- void Ethercat_CheckNumberOfSlaves_VR (Int32 slot, UInt32 out_vr_index)

  Determine the number of slaves on the bus. See Trio BASIC help for more details.
- UInt32 Ethercat_CheckNumberOfSlaves (Int32 slot)

  Determine the number of slaves on the bus. See Trio BASIC help for more details.
- void Ethercat_GetSlaveAddress_VR (Int32 slot, Int32 slave, UInt32 out_vr_index)

  Determine the configured address of the slave identified by its position on the network. See Trio BASIC help for more details.
- Int32 Ethercat_GetSlaveAddress (Int32 slot, Int32 slave)

  Determine the configured address of the slave identified by its position on the network. See Trio BASIC help for more details.
- void Ethercat_GetSlaveAxis_VR (Int32 slot, Int32 slave, UInt32 out_vr_index)

  Determine the axis of the slave identified by its position on the network. See Trio BASIC help for more details.
- Int32 Ethercat_GetSlaveAxis (Int32 slot, Int32 slave)

  Determine the axis of the slave identified by its position on the network. See Trio BASIC help for more details.
- void Ethercat_SetState (Int32 slot, EthercatState state, UInt32 display)

  This function controls the EtherCAT State Machine (ESM). It requests the master change to given EtherCAT 'state', and hence changes all slaves to the same state. When a change to a higher state is made, the EtherCAT network will progress to the new state through the in-between states to allow correct starting of the network. See Trio BASIC help for more details.
- void Ethercat_GetState_VR (Int32 slot, UInt32 out_vr_index)

  Gets the present state of the EtherCAT running on the defined slot. See Trio BASIC help for more details.
- EthercatState Ethercat_GetState (Int32 slot)

  Gets the present state of the EtherCAT running on the defined slot. See Trio BASIC help for more details.
- unsafe void Ethercat_SetSlave_ESC_Register (Int32 slot, Int32 slave, UInt32 esc_offset, UInt32 data_length, Byte *data)

  Low-level write to the slave's ESC registers, addressing method used is auto-increment addressing. See Trio BASIC help for more details.
- void Ethercat_GetSlave_ESC_Register (Int32 slot, Int32 slave, UInt32 esc_offset, UInt32 data_length, UInt32 out_vr_index)

  Low-level read from the slave's ESC registers, addressing method used is auto-increment addressing. See Trio BASIC help for more details.
- void Ethercat_SendResetSequence (Int32 axis, UInt32 mode, UInt32 timeout)

Reset a slave error. This function runs the error reset sequence on the drive control word. DRIVE_CONTROLWORD bit 8 is toggled high then low. This will instruct the drive to reset any errors in the drive where the cause of the error has been removed. See Trio BASIC help for more details.

- void Ethercat_SendResetSequence (Int32 axis, UInt32 mode)

  Reset a slave error. This function runs the error reset sequence on the drive control word. DRIVE_CONTROLWORD bit 8 is toggled high then low. This will instruct the drive to reset any errors in the drive where the cause of the error has been removed. See Trio BASIC help for more details.

- void Ethercat_SendResetSequence (Int32 axis)

  Reset a slave error. This function runs the error reset sequence on the drive control word. DRIVE_CONTROLWORD bit 8 is toggled high then low. This will instruct the drive to reset any errors in the drive where the cause of the error has been removed. See Trio BASIC help for more details.

- void Ethercat_SetStatusMaskAndTriggerPattern_VR (Int32 slot, Int32 addr, UInt32 vr_index)

  This command enables the user to define a CoE Statusword (0x6041:00) mask and trigger pattern for raising the controller AXISSTATUS 'Remote Drive Status word Error' flag (bit 26.) The mask defines which bits within the CoE status word are relevant, and the trigger pattern defines the state of these bits(0 or 1) which will trigger the controller AXISSTATUS flag. If it is necessary to reset the watchdog(master enable) when this AXISISTATUS flag is set then the axis ERRORMASK must be set accordingly. The slave mask and pattern can be set and modified from preop to operational state of the network(after the configured addresses have been assigned.) It must be reset if the network is restarted. The slave must support the CoE statusword(0x6041:00) within the PDO cyclic data. If the Mask is zero the controller will not execute any runtime check of the CoE statusword. If the watchdog must be cleared to OFF when the 'remote drive statusword error' flag is raised then bit 26 of the axis ERRORMASK parameter must be set. See Trio BASIC help for more details.

- void Ethercat_SetStatusMaskAndTriggerPattern (Int32 slot, Int32 addr, UInt32 mask, UInt32 trigger_pattern)

  This command enables the user to define a CoE Statusword (0x6041:00) mask and trigger pattern for raising the controller AXISSTATUS 'Remote Drive Status word Error' flag (bit 26.) The mask defines which bits within the CoE status word are relevant, and the trigger pattern defines the state of these bits(0 or 1) which will trigger the controller AXISSTATUS flag. If it is necessary to reset the watchdog(master enable) when this AXISISTATUS flag is set then the axis ERRORMASK must be set accordingly. The slave mask and pattern can be set and modified from preop to operational state of the network(after the configured addresses have been assigned.) It must be reset if the network is restarted. The slave must support the CoE statusword(0x6041:00) within the PDO cyclic data. If the Mask is zero the controller will not execute any runtime check of the CoE statusword. If the watchdog must be cleared to OFF when the 'remote drive statusword error' flag is raised then bit 26 of the axis ERRORMASK parameter must be set. See Trio BASIC help for more details.

- void Ethercat_GetStatusMaskAndTriggerPattern_VR (Int32 slot, Int32 addr, UInt32 out_vr_index)

  This command enables the user to display and check the mask and trigger pattern values set (if any) for a given slave. See Trio BASIC help for more details.

- Tuple< UInt32, UInt32 > Ethercat_GetStatusMaskAndTriggerPattern (Int32 slot, Int32 addr)

  This command enables the user to display and check the mask and trigger pattern values set (if any) for a given slave. See Trio BASIC help for more details.

- void Ethercat_DisplayNetworkConfig (Int32 slot)

  Displays the network configuration to the command line terminal in Motion Perfect. See Trio BASIC help for more details.

- void Ethercat_SetSync0StartTime_VR (UInt32 vr_index)

  Set the network default Sync0 start time offset. The offset value is from the given VR. The default start time is 1 second(1000000 microseconds).Add or subtract the required offset in microseconds.For example 1000150 is an offset of 150 microseconds and 999930 is an offset of -70 microseconds.The offset must not exceed ±50 % of the servo period. The function returns TRUE / FALSE depending upon whether the operation was successful. This command must be run before the EtherCAT network is started. See Trio BASIC help for more details.

- void Ethercat_SetSync0StartTime (Int32 offset)

  Set the network default Sync0 start time offset. If offset value is directly provided as a parameter. The default start time is 1 second(1000000 microseconds).Add or subtract the required offset in microseconds.For example 1000150 is an offset of 150 microseconds and 999930 is an offset of -70 microseconds.The offset must not exceed ±50 % of the servo period. The function returns TRUE / FALSE depending upon whether the operation was successful. This command must be run before the EtherCAT network is started. See Trio BASIC help for more details.

- void Ethercat_SetDistrClockMaster_VR (Int32 slot, UInt32 function, UInt32 vr_index)

  Set the EtherCAT master to enable the first DC enabled node to be the Distributed Clocks master. As this requires the Trio Motion Coordinator to synchronise to the external DC master, it does not function on P876, MC4N-ECAT or MC664(X) as these have FPGA based fixed timing hardware for the EtherCAT port. The function returns TRUE

/ FALSE depending upon whether the operation was successful. This command must be run before the EtherCAT network is started. See Trio BASIC help for more details.

- void Ethercat_SetDistrClockMaster (Int32 slot, UInt32 function, UInt32 value)

  Set the EtherCAT master to enable the first DC enabled node to be the Distributed Clocks master. As this requires the Trio Motion Coordinator to synchronise to the external DC master, it does not function on P876, MC4N-ECAT or MC664(X) as these have FPGA based fixed timing hardware for the EtherCAT port. The function returns TRUE / FALSE depending upon whether the operation was successful. This command must be run before the EtherCAT network is started. See Trio BASIC help for more details.

- void Ethercat_SetEmergencyMsgControl (Int32 slot, UInt32 mode)

  Set the response of the master to an emergency telegram transmitted by a slave node. This function enables the user to determine whether emergency messages should trigger a system error alarm and cause the WDOG to be dropped. See Trio BASIC help for more details.

- void Ethercat_GetEmergencyMsgControl_VR (Int32 slot, UInt32 out_vr_index)

  Get the current value; indicating the response of the master to an emergency telegram transmitted by a slave node. See Trio BASIC help for more details.

- UInt32 Ethercat_GetEmergencyMsgControl (Int32 slot)

  Get the current value; indicating the response of the master to an emergency telegram transmitted by a slave node. See Trio BASIC help for more details.

- void Ethercat_CoRead (Int32 slot, Int32 address, UInt16 object_index, Byte object_subindex, Co_ObjectType type, Int32 out_vr_index)

  This function gets a CANopen-over-EtherCAT (CoE) object from the remote drive or I/O device. The object's index and sub-index are used to request a value and that value is either placed in the VR or is displayed in the Motion Perfect terminal if the VR number is set to -1. See Trio BASIC help for more details.

- void Ethercat_CoWrite_VR (Int32 slot, Int32 address, UInt16 object_index, Byte object_subindex, Co_ObjectType type, UInt32 vr_index)

  This function sets a CANopen-over-EtherCAT (CoE) object in the remote drive or I/O device. The object's index and sub-index are used to write a value to that object. The value comes from a VR. Use this function for integer values. Refer to the remote device's manual for a list of available objects. See Trio BASIC help for more details.

- void Ethercat_CoWrite_Value (Int32 slot, Int32 address, UInt16 object_index, Byte object_subindex, Co_ObjectType type, Int64 value)

  This function sets a CANopen-over-EtherCAT (CoE) object in the remote drive or I/O device. The object's index and sub-index are used to write a value to that object. Use this function for integer values. Refer to the remote device's manual for a list of available objects. See Trio BASIC help for more details.

- void Ethercat_CoWrite_Value (Int32 slot, Int32 address, UInt16 object_index, Byte object_subindex, Co_ObjectType type, double value)

  This function sets a CANopen-over-EtherCAT (CoE) object in the remote drive or I/O device. The object's index and sub-index are used to write a value to that object. Use this function for floating-point values. Refer to the remote device's manual for a list of available objects. See Trio BASIC help for more details.

- void Ethercat_CoReadAxis (Int32 axis_number, UInt16 object_index, Byte object_subindex, Co_ObjectType type, Int32 out_vr_index)

  This function gets a CANopen-over-EtherCAT (CoE) object from the remote drive or I/O device. The object's index and sub-index are used to request a value and that value is either placed in the VR or is displayed in the Motion Perfect terminal if the VR number is set to -1. Refer to the remote device's manual for a list of available objects. See Trio BASIC help for more details.

- void Ethercat_CoWriteAxis_VR (Int32 axis_number, UInt16 object_index, Byte object_subindex, Co_ObjectType type, UInt32 vr_index)

  This function sets a CANopen-over-EtherCAT (CoE) object in the remote drive or I/O device. The object's index and sub-index are used to write a value to that object. The value comes from a VR. Refer to the remote device's manual for a list of available objects. See Trio BASIC help for more details.

- void Ethercat_CoWriteAxis_Value (Int32 axis_number, UInt16 object_index, Byte object_subindex, Co_ObjectType type, Int64 value)

  This function sets a CANopen-over-EtherCAT (CoE) object in the remote drive or I/O device. The object's index and sub-index are used to write a value to that object. Use this function for integer values. Refer to the remote device's manual for a list of available objects. See Trio BASIC help for more details.

- void Ethercat_CoWriteAxis_Value (Int32 axis_number, UInt16 object_index, Byte object_subindex, Co_ObjectType type, double value)

This function sets a CANopen-over-EtherCAT (CoE) object in the remote drive or I/O device. The object's index and sub-index are used to write a value to that object. Use this function for floating-point values. Refer to the remote device's manual for a list of available objects. See Trio BASIC help for more details.

- void UploadFile (String local_file, String remote_file, FileType remote_file_type, FileTransferOption opt, ProgressHandlerCallback handler)

  Uploads a file to the Motion Coordinator.
- void DownloadFile (String local_file, String remote_file, ProgressHandlerCallback handler)

  Downloads a file from the Motion Coordinator.
- void Delete (String remote_file, bool compiled_code_only=false)

  Deletes a file on the Motion Coordinator.
- void New (String target)

  Deletes a file, all files or the TABLE data the Motion Coordinator.
- void Select (String remote_file, FileType file_type)

  Makes the named program the currently selected program on the Motion Coordinator, if the named program does not exist then it makes a program of that name. The selected program cannot be changed when programs are running. When a program is selected any previously selected program is compiled. When program_type is specified is to make a program of that name and type, it cannot be possible to change the type of an existing program.
- void BacklashDisable (Int32 axis=-1)

  This axis function disables backlash compensation.
- void Backlash (double distance, double speed, double acceleration, Int32 axis=-1)

  This axis function allows backlash compensation to be loaded. This is achieved by applying an offset move when the motor demand is in one direction, then reversing the offset move when the motor demand is in the opposite direction. These moves are superimposed on the commanded axis movements. The backlash compensation is applied after a reversal of the direction of change of the DPOS parameter. The backlash compensation can be seen in the AXIS_DPOS axis parameter.This is effectively DPOS + backlash compensation.
- void EncoderRatio (Int64 mpos_count, Int64 input_count, Int32 axis=-1)

  This command allows the incoming encoder count to be scaled by a non integer ratio : MPOS = (mpos_count / input_count) x encoder_edges_input When using the servo loop you will need to adjust the gains to maintain performance and stability. Unlike the UNITS parameter, which only affects the scaling seen by the user programs, ENCODER_RATIO affects all motion commands. ENCODER_RATIO does not replace UNITS.Only use ENCODER_RATIO where absolutely necessary.PP_STEP and ENCODER_RATIO cannot be used at the same time on the same axis.
- void StepRatio (Int64 output_count, Int64 dpos_count, Int32 axis=-1)

  This command sets up an integer ratio for the axis' demanded position. Usually this is used on a stepper (pulse+direction) output. It can also be applied to a networked axis in positon mode, such as EtherCAT and RTEX position servo. Every servo-period the change in position is passed through the STEP_RATIO function before it goes to the step pulse or network drive output. The STEP_RATIO function operates before the divide by 16 factor(divide by 256 in Flexslice P375) in the stepper axis.This maintains the good timing resolution of the stepper output circuit. STEP_RATIO does not replace UNITS.
- void Cam (int tableStart, int tableStop, double tableMultiplier, double linkDistance, Int32 axis=-1)

  This command is used to generate movement of an axis according to a table of positions which define a movement profile.
- void CamBox (int tableStart, int tableStop, double tableMultiplier, double linkDistance, int linkAxis, CamBoxLinkOptions linkOptions, double linkPos=0, double offsetStart=0, Int32 axis=-1)

  This command is used to generate movement of an axis according to a table of POSITIONS which define the movement profile.
- uint GetSystemParameter_AIN0 ()

  Getter for AIN0 system parameter.
- void SetSystemParameter_AIN0 (uint value)

  Setter for AIN0 system parameter.
- uint GetSystemParameter_AIN1 ()

  Getter for AIN1 system parameter.
- void SetSystemParameter_AIN1 (uint value)

  Setter for AIN1 system parameter.
- uint GetSystemParameter_AIN2 ()

  Getter for AIN2 system parameter.

- void SetSystemParameter_AIN2 (uint value)

  Setter for AIN2 system parameter.
- uint GetSystemParameter_AIN3 ()

  Getter for AIN3 system parameter.
- void SetSystemParameter_AIN3 (uint value)

  Setter for AIN3 system parameter.
- uint GetSystemParameter_AOUT0 ()

  Getter for AOUT0 system parameter.
- void SetSystemParameter_AOUT0 (uint value)

  Setter for AOUT0 system parameter.
- uint GetSystemParameter_AOUT1 ()

  Getter for AOUT1 system parameter.
- void SetSystemParameter_AOUT1 (uint value)

  Setter for AOUT1 system parameter.
- uint GetSystemParameter_AOUT2 ()

  Getter for AOUT2 system parameter.
- void SetSystemParameter_AOUT2 (uint value)

  Setter for AOUT2 system parameter.
- uint GetSystemParameter_AOUT3 ()

  Getter for AOUT3 system parameter.
- void SetSystemParameter_AOUT3 (uint value)

  Setter for AOUT3 system parameter.
- uint GetSystemParameter_AINBI0 ()

  Getter for AINBI0 system parameter.
- void SetSystemParameter_AINBI0 (uint value)

  Setter for AINBI0 system parameter.
- uint GetSystemParameter_AINBI1 ()

  Getter for AINBI1 system parameter.
- void SetSystemParameter_AINBI1 (uint value)

  Setter for AINBI1 system parameter.
- uint GetSystemParameter_AINBI2 ()

  Getter for AINBI2 system parameter.
- void SetSystemParameter_AINBI2 (uint value)

  Setter for AINBI2 system parameter.
- uint GetSystemParameter_AINBI3 ()

  Getter for AINBI3 system parameter.
- void SetSystemParameter_AINBI3 (uint value)

  Setter for AINBI3 system parameter.
- uint GetSystemParameter_INPUTS0 ()

  Getter for INPUTS0 system parameter.
- uint GetSystemParameter_INPUTS1 ()

  Getter for INPUTS1 system parameter.
- uint GetSystemParameter_SLOT0_TIME ()

  Getter for SLOT0_TIME system parameter.
- void SetSystemParameter_SLOT0_TIME (uint value)

  Setter for SLOT0_TIME system parameter.
- uint GetSystemParameter_SLOT1_TIME ()

  Getter for SLOT1_TIME system parameter.
- void SetSystemParameter_SLOT1_TIME (uint value)

  Setter for SLOT1_TIME system parameter.
- uint GetSystemParameter_SLOT2_TIME ()

Getter for SLOT2_TIME system parameter.

- void SetSystemParameter_SLOT2_TIME (uint value)

  Setter for SLOT2_TIME system parameter.
- uint GetSystemParameter_SLOT3_TIME ()

  Getter for SLOT3_TIME system parameter.
- void SetSystemParameter_SLOT3_TIME (uint value)

  Setter for SLOT3_TIME system parameter.
- bool GetSystemParameter_WDOG ()

  Getter for WDOG system parameter.
- void SetSystemParameter_WDOG (bool value)

  Setter for WDOG system parameter.
- uint GetSystemParameter_SYSTEM_ERROR ()

  Getter for SYSTEM_ERROR system parameter.
- long GetSystemParameter_CURRENT_JITTER ()

  Getter for CURRENT_JITTER system parameter.
- void SetSystemParameter_CURRENT_JITTER (long value)

  Setter for CURRENT_JITTER system parameter.
- long GetSystemParameter_AVERAGE_JITTER ()

  Getter for AVERAGE_JITTER system parameter.
- void SetSystemParameter_AVERAGE_JITTER (long value)

  Setter for AVERAGE_JITTER system parameter.
- long GetSystemParameter_MAX_JITTER ()

  Getter for MAX_JITTER system parameter.
- void SetSystemParameter_MAX_JITTER (long value)

  Setter for MAX_JITTER system parameter.
- long GetSystemParameter_MISSED_INTERRUPTS ()

  Getter for MISSED_INTERRUPTS system parameter.
- void SetSystemParameter_MISSED_INTERRUPTS (long value)

  Setter for MISSED_INTERRUPTS system parameter.
- long GetSystemParameter_MISSED_SEND_FRAMES ()

  Getter for MISSED_SEND_FRAMES system parameter.
- void SetSystemParameter_MISSED_SEND_FRAMES (long value)

  Setter for MISSED_SEND_FRAMES system parameter.
- long GetSystemParameter_MISSED_RECV_FRAMES ()

  Getter for MISSED_RECV_FRAMES system parameter.
- void SetSystemParameter_MISSED_RECV_FRAMES (long value)

  Setter for MISSED_RECV_FRAMES system parameter.
- short GetSystemParameter_LIMIT_BUFFERED ()

  Getter for LIMIT_BUFFERED system parameter.
- void SetSystemParameter_LIMIT_BUFFERED (short value)

  Setter for LIMIT_BUFFERED system parameter.
- uint GetSystemParameter_ADDRESS ()

  Getter for ADDRESS system parameter.
- void SetSystemParameter_ADDRESS (uint value)

  Setter for ADDRESS system parameter.
- int GetSystemParameter_AUTO_ETHERCAT ()

  Getter for AUTO_ETHERCAT system parameter.
- void SetSystemParameter_AUTO_ETHERCAT (int value)

  Setter for AUTO_ETHERCAT system parameter.
- uint GetSystemParameter_CONTROL ()

  Getter for CONTROL system parameter.

- long GetSystemParameter_CPU_EXCEPTIONS ()

  Getter for CPU_EXCEPTIONS system parameter.
- void SetSystemParameter_CPU_EXCEPTIONS (long value)

  Setter for CPU_EXCEPTIONS system parameter.
- int GetSystemParameter_DISPLAY ()

  Getter for DISPLAY system parameter.
- void SetSystemParameter_DISPLAY (int value)

  Setter for DISPLAY system parameter.
- uint GetSystemParameter_FLASH_STATUS ()

  Getter for FLASH_STATUS system parameter.
- int GetSystemParameter_ERROR_AXIS ()

  Getter for ERROR_AXIS system parameter.
- long GetSystemParameter_FPU_EXCEPTIONS ()

  Getter for FPU_EXCEPTIONS system parameter.
- void SetSystemParameter_FPU_EXCEPTIONS (long value)

  Setter for FPU_EXCEPTIONS system parameter.
- int GetSystemParameter_HMI_PROC ()

  Getter for HMI_PROC system parameter.
- void SetSystemParameter_HMI_PROC (int value)

  Setter for HMI_PROC system parameter.
- uint GetSystemParameter_IP_ADDRESS ()

  Getter for IP_ADDRESS system parameter.
- void SetSystemParameter_IP_ADDRESS (uint value)

  Setter for IP_ADDRESS system parameter.
- uint GetSystemParameter_IP_GATEWAY ()

  Getter for IP_GATEWAY system parameter.
- void SetSystemParameter_IP_GATEWAY (uint value)

  Setter for IP_GATEWAY system parameter.
- ulong GetSystemParameter_IP_MAC ()

  Getter for IP_MAC system parameter.
- void SetSystemParameter_IP_MAC (ulong value)

  Setter for IP_MAC system parameter.
- uint GetSystemParameter_IP_MEMORY_CONFIG ()

  Getter for IP_MEMORY_CONFIG system parameter.
- void SetSystemParameter_IP_MEMORY_CONFIG (uint value)

  Setter for IP_MEMORY_CONFIG system parameter.
- uint GetSystemParameter_IP_NETMASK ()

  Getter for IP_NETMASK system parameter.
- void SetSystemParameter_IP_NETMASK (uint value)

  Setter for IP_NETMASK system parameter.
- uint GetSystemParameter_IP_PROTOCOL_CONFIG ()

  Getter for IP_PROTOCOL_CONFIG system parameter.
- void SetSystemParameter_IP_PROTOCOL_CONFIG (uint value)

  Setter for IP_PROTOCOL_CONFIG system parameter.
- uint GetSystemParameter_IP_PROTOCOL_CTRL ()

  Getter for IP_PROTOCOL_CTRL system parameter.
- void SetSystemParameter_IP_PROTOCOL_CTRL (uint value)

  Setter for IP_PROTOCOL_CTRL system parameter.
- ulong GetSystemParameter_IP_TCP_TIMEOUT ()

  Getter for IP_TCP_TIMEOUT system parameter.
- void SetSystemParameter_IP_TCP_TIMEOUT (ulong value)

Setter for IP_TCP_TIMEOUT system parameter.

- uint GetSystemParameter_IP_TCP_TX_THRESHOLD ()

  Getter for IP_TCP_TX_THRESHOLD system parameter.
- void SetSystemParameter_IP_TCP_TX_THRESHOLD (uint value)

  Setter for IP_TCP_TX_THRESHOLD system parameter.
- uint GetSystemParameter_IP_TCP_TX_TIMEOUT ()

  Getter for IP_TCP_TX_TIMEOUT system parameter.
- void SetSystemParameter_IP_TCP_TX_TIMEOUT (uint value)

  Setter for IP_TCP_TX_TIMEOUT system parameter.
- int GetSystemParameter_LAST_AXIS ()

  Getter for LAST_AXIS system parameter.
- void SetSystemParameter_LAST_AXIS (int value)

  Setter for LAST_AXIS system parameter.
- ushort GetSystemParameter_MODBUS_TX_DELAY ()

  Getter for MODBUS_TX_DELAY system parameter.
- void SetSystemParameter_MODBUS_TX_DELAY (ushort value)

  Setter for MODBUS_TX_DELAY system parameter.
- ulong GetSystemParameter_MOTION_ERROR ()

  Getter for MOTION_ERROR system parameter.
- short GetSystemParameter_MPE_MODE ()

  Getter for MPE_MODE system parameter.
- uint GetSystemParameter_NOP ()

  Getter for NOP system parameter.
- uint GetSystemParameter_NIN ()

  Getter for NIN system parameter.
- uint GetSystemParameter_NIO ()

  Getter for NIO system parameter.
- void SetSystemParameter_NIO (uint value)

  Setter for NIO system parameter.
- uint GetSystemParameter_N_ANA_IN ()

  Getter for N_ANA_IN system parameter.
- uint GetSystemParameter_N_ANA_OUT ()

  Getter for N_ANA_OUT system parameter.
- int GetSystemParameter_PCMCAT_PROC ()

  Getter for PCMCAT_PROC system parameter.
- void SetSystemParameter_PCMCAT_PROC (int value)

  Setter for PCMCAT_PROC system parameter.
- int GetSystemParameter_PLC_CONFIG ()

  Getter for PLC_CONFIG system parameter.
- void SetSystemParameter_PLC_CONFIG (int value)

  Setter for PLC_CONFIG system parameter.
- long GetSystemParameter_PLC_ERROR ()

  Getter for PLC_ERROR system parameter.
- long GetSystemParameter_PLC_OVERFLOW ()

  Getter for PLC_OVERFLOW system parameter.
- long GetSystemParameter_PLC_RUN ()

  Getter for PLC_RUN system parameter.
- int GetSystemParameter_REMOTE_PROC ()

  Getter for REMOTE_PROC system parameter.
- void SetSystemParameter_REMOTE_PROC (int value)

  Setter for REMOTE_PROC system parameter.

- uint GetSystemParameter_SCHEDULE_TYPE ()
  Getter for SCHEDULE_TYPE system parameter.
- void SetSystemParameter_SCHEDULE_TYPE (uint value)
  Setter for SCHEDULE_TYPE system parameter.
- int GetSystemParameter_SCOPE_CYCLE_COUNT ()
  Getter for SCOPE_CYCLE_COUNT system parameter.
- void SetSystemParameter_SCOPE_CYCLE_COUNT (int value)
  Setter for SCOPE_CYCLE_COUNT system parameter.
- long GetSystemParameter_SCOPE_DELAY ()
  Getter for SCOPE_DELAY system parameter.
- void SetSystemParameter_SCOPE_DELAY (long value)
  Setter for SCOPE_DELAY system parameter.
- int GetSystemParameter_SCOPE_POS ()
  Getter for SCOPE_POS system parameter.
- void SetSystemParameter_SCOPE_POS (int value)
  Setter for SCOPE_POS system parameter.
- int GetSystemParameter_SCOPE_TRIGGER_POS ()
  Getter for SCOPE_TRIGGER_POS system parameter.
- int GetSystemParameter_SERIAL_NUMBER ()
  Getter for SERIAL_NUMBER system parameter.
- void SetSystemParameter_SERIAL_NUMBER (int value)
  Setter for SERIAL_NUMBER system parameter.
- int GetSystemParameter_SERVO_OFFSET ()
  Getter for SERVO_OFFSET system parameter.
- void SetSystemParameter_SERVO_OFFSET (int value)
  Setter for SERVO_OFFSET system parameter.
- int GetSystemParameter_SERVO_PERIOD ()
  Getter for SERVO_PERIOD system parameter.
- void SetSystemParameter_SERVO_PERIOD (int value)
  Setter for SERVO_PERIOD system parameter.
- uint GetSystemParameter_SYSTEM_ERROR_MASK ()
  Getter for SYSTEM_ERROR_MASK system parameter.
- void SetSystemParameter_SYSTEM_ERROR_MASK (uint value)
  Setter for SYSTEM_ERROR_MASK system parameter.
- uint GetSystemParameter_SYSTEM_LOAD ()
  Getter for SYSTEM_LOAD system parameter.
- uint GetSystemParameter_SYSTEM_LOAD_MAX ()
  Getter for SYSTEM_LOAD_MAX system parameter.
- void SetSystemParameter_SYSTEM_LOAD_MAX (uint value)
  Setter for SYSTEM_LOAD_MAX system parameter.
- int GetSystemParameter_TEXT_FILE_LOADER_PROC ()
  Getter for TEXT_FILE_LOADER_PROC system parameter.
- void SetSystemParameter_TEXT_FILE_LOADER_PROC (int value)
  Setter for TEXT_FILE_LOADER_PROC system parameter.
- int GetSystemParameter_TSIZE ()
  Getter for TSIZE system parameter.
- uint GetSystemParameter_UNIT_ERROR ()
  Getter for UNIT_ERROR system parameter.
- double GetSystemParameter_VERSION ()
  Getter for VERSION system parameter.
- int GetSystemParameter_VRSIZE ()

Getter for VRSIZE system parameter.

- short GetSystemParameter_PROCNUMBER ()

  Getter for PROCNUMBER system parameter.
- double GetAxisParameter_ACCEL (int axis)

  Getter for ACCEL axis parameter.
- void SetAxisParameter_ACCEL (int axis, double value)

  Setter for ACCEL axis parameter.
- int GetAxisParameter_ADDAX_AXIS (int axis)

  Getter for ADDAX_AXIS axis parameter.
- double GetAxisParameter_AFF_GAIN (int axis)

  Getter for AFF_GAIN axis parameter.
- void SetAxisParameter_AFF_GAIN (int axis, double value)

  Setter for AFF_GAIN axis parameter.
- AxisType GetAxisParameter_ATYPE (int axis)

  Getter for ATYPE axis parameter.
- void SetAxisParameter_ATYPE (int axis, AxisType value)

  Setter for ATYPE axis parameter.
- double GetAxisParameter_AXIS_ACCEL (int axis)

  Getter for AXIS_ACCEL axis parameter.
- void SetAxisParameter_AXIS_ACCEL (int axis, double value)

  Setter for AXIS_ACCEL axis parameter.
- int GetAxisParameter_AXIS_ADDRESS (int axis)

  Getter for AXIS_ADDRESS axis parameter.
- void SetAxisParameter_AXIS_ADDRESS (int axis, int value)

  Setter for AXIS_ADDRESS axis parameter.
- double GetAxisParameter_AXIS_DECEL (int axis)

  Getter for AXIS_DECEL axis parameter.
- void SetAxisParameter_AXIS_DECEL (int axis, double value)

  Setter for AXIS_DECEL axis parameter.
- double GetAxisParameter_AXIS_DPOS (int axis)

  Getter for AXIS_DPOS axis parameter.
- bool GetAxisParameter_AXIS_ENABLE (int axis)

  Getter for AXIS_ENABLE axis parameter.
- void SetAxisParameter_AXIS_ENABLE (int axis, bool value)

  Setter for AXIS_ENABLE axis parameter.
- int GetAxisParameter_AXIS_ERROR_COUNT (int axis)

  Getter for AXIS_ERROR_COUNT axis parameter.
- void SetAxisParameter_AXIS_ERROR_COUNT (int axis, int value)

  Setter for AXIS_ERROR_COUNT axis parameter.
- double GetAxisParameter_AXIS_FS_LIMIT (int axis)

  Getter for AXIS_FS_LIMIT axis parameter.
- void SetAxisParameter_AXIS_FS_LIMIT (int axis, double value)

  Setter for AXIS_FS_LIMIT axis parameter.
- double GetAxisParameter_AXIS_MAX_JOG_SPEED (int axis)

  Getter for AXIS_MAX_JOG_SPEED axis parameter.
- void SetAxisParameter_AXIS_MAX_JOG_SPEED (int axis, double value)

  Setter for AXIS_MAX_JOG_SPEED axis parameter.
- AxisMode GetAxisParameter_AXIS_MODE (int axis)

  Getter for AXIS_MODE axis parameter.
- void SetAxisParameter_AXIS_MODE (int axis, AxisMode value)

  Setter for AXIS_MODE axis parameter.

- double GetAxisParameter_AXIS_RS_LIMIT (int axis)

  Getter for AXIS_RS_LIMIT axis parameter.
- void SetAxisParameter_AXIS_RS_LIMIT (int axis, double value)

  Setter for AXIS_RS_LIMIT axis parameter.
- double GetAxisParameter_AXIS_SPEED (int axis)

  Getter for AXIS_SPEED axis parameter.
- void SetAxisParameter_AXIS_SPEED (int axis, double value)

  Setter for AXIS_SPEED axis parameter.
- double GetAxisParameter_AXIS_UNITS (int axis)

  Getter for AXIS_UNITS axis parameter.
- void SetAxisParameter_AXIS_UNITS (int axis, double value)

  Setter for AXIS_UNITS axis parameter.
- AxisStatus GetAxisParameter_AXISSTATUS (int axis)

  Getter for AXISSTATUS axis parameter.
- long GetAxisParameter_BACKLASH_DIST (int axis)

  Getter for BACKLASH_DIST axis parameter.
- void SetAxisParameter_BACKLASH_DIST (int axis, long value)

  Setter for BACKLASH_DIST axis parameter.
- double GetAxisParameter_CHANGE_DIR_LAST (int axis)

  Getter for CHANGE_DIR_LAST axis parameter.
- void SetAxisParameter_CHANGE_DIR_LAST (int axis, double value)

  Setter for CHANGE_DIR_LAST axis parameter.
- double GetAxisParameter_CLOSE_WIN (int axis)

  Getter for CLOSE_WIN axis parameter.
- void SetAxisParameter_CLOSE_WIN (int axis, double value)

  Setter for CLOSE_WIN axis parameter.
- double GetAxisParameter_CLUTCH_RATE (int axis)

  Getter for CLUTCH_RATE axis parameter.
- void SetAxisParameter_CLUTCH_RATE (int axis, double value)

  Setter for CLUTCH_RATE axis parameter.
- CornerMode GetAxisParameter_CORNER_MODE (int axis)

  Getter for CORNER_MODE axis parameter.
- void SetAxisParameter_CORNER_MODE (int axis, CornerMode value)

  Setter for CORNER_MODE axis parameter.
- CornerState GetAxisParameter_CORNER_STATE (int axis)

  Getter for CORNER_STATE axis parameter.
- void SetAxisParameter_CORNER_STATE (int axis, CornerState value)

  Setter for CORNER_STATE axis parameter.
- double GetAxisParameter_CREEP (int axis)

  Getter for CREEP axis parameter.
- void SetAxisParameter_CREEP (int axis, double value)

  Setter for CREEP axis parameter.
- double GetAxisParameter_D_GAIN (int axis)

  Getter for D_GAIN axis parameter.
- void SetAxisParameter_D_GAIN (int axis, double value)

  Setter for D_GAIN axis parameter.
- int GetAxisParameter_DAC (int axis)

  Getter for DAC axis parameter.
- void SetAxisParameter_DAC (int axis, int value)

  Setter for DAC axis parameter.
- long GetAxisParameter_DAC_SCALE (int axis)

Getter for DAC_SCALE axis parameter.

- void SetAxisParameter_DAC_SCALE (int axis, long value)

  Setter for DAC_SCALE axis parameter.
- int GetAxisParameter_DAC_OUT (int axis)

  Getter for DAC_OUT axis parameter.
- short GetAxisParameter_DATUM_IN (int axis)

  Getter for DATUM_IN axis parameter.
- void SetAxisParameter_DATUM_IN (int axis, short value)

  Setter for DATUM_IN axis parameter.
- double GetAxisParameter_DECEL (int axis)

  Getter for DECEL axis parameter.
- void SetAxisParameter_DECEL (int axis, double value)

  Setter for DECEL axis parameter.
- double GetAxisParameter_DECEL_ANGLE (int axis)

  Getter for DECEL_ANGLE axis parameter.
- void SetAxisParameter_DECEL_ANGLE (int axis, double value)

  Setter for DECEL_ANGLE axis parameter.
- double GetAxisParameter_DEMAND_SPEED (int axis)

  Getter for DEMAND_SPEED axis parameter.
- double GetAxisParameter_DPOS (int axis)

  Getter for DPOS axis parameter.
- int GetAxisParameter_DRIVE_CONTROL (int axis)

  Getter for DRIVE_CONTROL axis parameter.
- void SetAxisParameter_DRIVE_CONTROL (int axis, int value)

  Setter for DRIVE_CONTROL axis parameter.
- int GetAxisParameter_DRIVE_CONTROLWORD (int axis)

  Getter for DRIVE_CONTROLWORD axis parameter.
- void SetAxisParameter_DRIVE_CONTROLWORD (int axis, int value)

  Setter for DRIVE_CONTROLWORD axis parameter.
- int GetAxisParameter_DRIVE_CW_MODE (int axis)

  Getter for DRIVE_CW_MODE axis parameter.
- void SetAxisParameter_DRIVE_CW_MODE (int axis, int value)

  Setter for DRIVE_CW_MODE axis parameter.
- short GetAxisParameter_DRIVE_CURRENT (int axis)

  Getter for DRIVE_CURRENT axis parameter.
- long GetAxisParameter_DRIVE_ENABLE (int axis)

  Getter for DRIVE_ENABLE axis parameter.
- void SetAxisParameter_DRIVE_ENABLE (int axis, long value)

  Setter for DRIVE_ENABLE axis parameter.
- long GetAxisParameter_DRIVE_FE (int axis)

  Getter for DRIVE_FE axis parameter.
- long GetAxisParameter_DRIVE_FE_LIMIT (int axis)

  Getter for DRIVE_FE_LIMIT axis parameter.
- void SetAxisParameter_DRIVE_FE_LIMIT (int axis, long value)

  Setter for DRIVE_FE_LIMIT axis parameter.
- uint GetAxisParameter_DRIVE_INPUTS (int axis)

  Getter for DRIVE_INPUTS axis parameter.
- int GetAxisParameter_DRIVE_MODE (int axis)

  Getter for DRIVE_MODE axis parameter.
- void SetAxisParameter_DRIVE_MODE (int axis, int value)

  Setter for DRIVE_MODE axis parameter.

- int GetAxisParameter_DRIVE_MONITOR (int axis)

  Getter for DRIVE_MONITOR axis parameter.
- int GetAxisParameter_DRIVE_PROFILE (int axis)

  Getter for DRIVE_PROFILE axis parameter.
- void SetAxisParameter_DRIVE_PROFILE (int axis, int value)

  Setter for DRIVE_PROFILE axis parameter.
- int GetAxisParameter_DRIVE_STATUS (int axis)

  Getter for DRIVE_STATUS axis parameter.
- int GetAxisParameter_DRIVE_TORQUE (int axis)

  Getter for DRIVE_TORQUE axis parameter.
- int GetAxisParameter_DRIVE_TYPE (int axis)

  Getter for DRIVE_TYPE axis parameter.
- void SetAxisParameter_DRIVE_TYPE (int axis, int value)

  Setter for DRIVE_TYPE axis parameter.
- int GetAxisParameter_DRIVE_VELOCITY (int axis)

  Getter for DRIVE_VELOCITY axis parameter.
- int GetAxisParameter_ENCODER (int axis)

  Getter for ENCODER axis parameter.
- ushort GetAxisParameter_ENCODER_BITS (int axis)

  Getter for ENCODER_BITS axis parameter.
- void SetAxisParameter_ENCODER_BITS (int axis, ushort value)

  Setter for ENCODER_BITS axis parameter.
- int GetAxisParameter_ENCODER_CONTROL (int axis)

  Getter for ENCODER_CONTROL axis parameter.
- void SetAxisParameter_ENCODER_CONTROL (int axis, int value)

  Setter for ENCODER_CONTROL axis parameter.
- double GetAxisParameter_ENCODER_FILTER (int axis)

  Getter for ENCODER_FILTER axis parameter.
- void SetAxisParameter_ENCODER_FILTER (int axis, double value)

  Setter for ENCODER_FILTER axis parameter.
- int GetAxisParameter_ENCODER_ID (int axis)

  Getter for ENCODER_ID axis parameter.
- void SetAxisParameter_ENCODER_ID (int axis, int value)

  Setter for ENCODER_ID axis parameter.
- int GetAxisParameter_ENCODER_STATUS (int axis)

  Getter for ENCODER_STATUS axis parameter.
- void SetAxisParameter_ENCODER_STATUS (int axis, int value)

  Setter for ENCODER_STATUS axis parameter.
- int GetAxisParameter_ENCODER_TURNS (int axis)

  Getter for ENCODER_TURNS axis parameter.
- void SetAxisParameter_ENCODER_TURNS (int axis, int value)

  Setter for ENCODER_TURNS axis parameter.
- double GetAxisParameter_END_DIR_LAST (int axis)

  Getter for END_DIR_LAST axis parameter.
- void SetAxisParameter_END_DIR_LAST (int axis, double value)

  Setter for END_DIR_LAST axis parameter.
- long GetAxisParameter_ENDMOVE (int axis)

  Getter for ENDMOVE axis parameter.
- void SetAxisParameter_ENDMOVE (int axis, long value)

  Setter for ENDMOVE axis parameter.
- double GetAxisParameter_ENDMOVE_BUFFER (int axis)

Getter for ENDMOVE_BUFFER axis parameter.

- void SetAxisParameter_ENDMOVE_BUFFER (int axis, double value)

  Setter for ENDMOVE_BUFFER axis parameter.
- double GetAxisParameter_ENDMOVE_SPEED (int axis)

  Getter for ENDMOVE_SPEED axis parameter.
- void SetAxisParameter_ENDMOVE_SPEED (int axis, double value)

  Setter for ENDMOVE_SPEED axis parameter.
- AxisStatus GetAxisParameter_ERRORMASK (int axis)

  Getter for ERRORMASK axis parameter.
- void SetAxisParameter_ERRORMASK (int axis, AxisStatus value)

  Setter for ERRORMASK axis parameter.
- short GetAxisParameter_FAST_JOG (int axis)

  Getter for FAST_JOG axis parameter.
- void SetAxisParameter_FAST_JOG (int axis, short value)

  Setter for FAST_JOG axis parameter.
- double GetAxisParameter_FASTDEC (int axis)

  Getter for FASTDEC axis parameter.
- void SetAxisParameter_FASTDEC (int axis, double value)

  Setter for FASTDEC axis parameter.
- double GetAxisParameter_FE (int axis)

  Getter for FE axis parameter.
- double GetAxisParameter_FE_LIMIT (int axis)

  Getter for FE_LIMIT axis parameter.
- void SetAxisParameter_FE_LIMIT (int axis, double value)

  Setter for FE_LIMIT axis parameter.
- short GetAxisParameter_FE_LIMIT_MODE (int axis)

  Getter for FE_LIMIT_MODE axis parameter.
- void SetAxisParameter_FE_LIMIT_MODE (int axis, short value)

  Setter for FE_LIMIT_MODE axis parameter.
- double GetAxisParameter_FE_RANGE (int axis)

  Getter for FE_RANGE axis parameter.
- void SetAxisParameter_FE_RANGE (int axis, double value)

  Setter for FE_RANGE axis parameter.
- double GetAxisParameter_FE_LATCH (int axis)

  Getter for FE_LATCH axis parameter.
- short GetAxisParameter_FHOLD_IN (int axis)

  Getter for FHOLD_IN axis parameter.
- void SetAxisParameter_FHOLD_IN (int axis, short value)

  Setter for FHOLD_IN axis parameter.
- double GetAxisParameter_FHSPEED (int axis)

  Getter for FHSPEED axis parameter.
- void SetAxisParameter_FHSPEED (int axis, double value)

  Setter for FHSPEED axis parameter.
- int GetAxisParameter_FORCE_DWELL (int axis)

  Getter for FORCE_DWELL axis parameter.
- void SetAxisParameter_FORCE_DWELL (int axis, int value)

  Setter for FORCE_DWELL axis parameter.
- double GetAxisParameter_FORCE_RAMP (int axis)

  Getter for FORCE_RAMP axis parameter.
- void SetAxisParameter_FORCE_RAMP (int axis, double value)

  Setter for FORCE_RAMP axis parameter.

- double GetAxisParameter_FORCE_SPEED (int axis)

  Getter for FORCE_SPEED axis parameter.
- void SetAxisParameter_FORCE_SPEED (int axis, double value)

  Setter for FORCE_SPEED axis parameter.
- double GetAxisParameter_FRAME_REP_DIST (int axis)

  Getter for FRAME_REP_DIST axis parameter.
- void SetAxisParameter_FRAME_REP_DIST (int axis, double value)

  Setter for FRAME_REP_DIST axis parameter.
- double GetAxisParameter_FRAME_SCALE (int axis)

  Getter for FRAME_SCALE axis parameter.
- void SetAxisParameter_FRAME_SCALE (int axis, double value)

  Setter for FRAME_SCALE axis parameter.
- double GetAxisParameter_FS_LIMIT (int axis)

  Getter for FS_LIMIT axis parameter.
- void SetAxisParameter_FS_LIMIT (int axis, double value)

  Setter for FS_LIMIT axis parameter.
- double GetAxisParameter_FULL_SP_RADIUS (int axis)

  Getter for FULL_SP_RADIUS axis parameter.
- void SetAxisParameter_FULL_SP_RADIUS (int axis, double value)

  Setter for FULL_SP_RADIUS axis parameter.
- short GetAxisParameter_FWD_IN (int axis)

  Getter for FWD_IN axis parameter.
- void SetAxisParameter_FWD_IN (int axis, short value)

  Setter for FWD_IN axis parameter.
- short GetAxisParameter_FWD_JOG (int axis)

  Getter for FWD_JOG axis parameter.
- void SetAxisParameter_FWD_JOG (int axis, short value)

  Setter for FWD_JOG axis parameter.
- double GetAxisParameter_I_GAIN (int axis)

  Getter for I_GAIN axis parameter.
- void SetAxisParameter_I_GAIN (int axis, double value)

  Setter for I_GAIN axis parameter.
- bool GetAxisParameter_IDLE (int axis)

  Getter for IDLE axis parameter.
- uint GetAxisParameter_IN_POS (int axis)

  Getter for IN_POS axis parameter.
- void SetAxisParameter_IN_POS (int axis, uint value)

  Setter for IN_POS axis parameter.
- long GetAxisParameter_IN_POS_DIST (int axis)

  Getter for IN_POS_DIST axis parameter.
- void SetAxisParameter_IN_POS_DIST (int axis, long value)

  Setter for IN_POS_DIST axis parameter.
- double GetAxisParameter_IN_POS_SPEED (int axis)

  Getter for IN_POS_SPEED axis parameter.
- void SetAxisParameter_IN_POS_SPEED (int axis, double value)

  Setter for IN_POS_SPEED axis parameter.
- double GetAxisParameter_INTERP_FACTOR (int axis)

  Getter for INTERP_FACTOR axis parameter.
- void SetAxisParameter_INTERP_FACTOR (int axis, double value)

  Setter for INTERP_FACTOR axis parameter.
- int GetAxisParameter_INVERT_STEP (int axis)

Getter for INVERT_STEP axis parameter.

- void SetAxisParameter_INVERT_STEP (int axis, int value)

  Setter for INVERT_STEP axis parameter.
- double GetAxisParameter_JERK (int axis)

  Getter for JERK axis parameter.
- void SetAxisParameter_JERK (int axis, double value)

  Setter for JERK axis parameter.
- double GetAxisParameter_JOGSPEED (int axis)

  Getter for JOGSPEED axis parameter.
- void SetAxisParameter_JOGSPEED (int axis, double value)

  Setter for JOGSPEED axis parameter.
- short GetAxisParameter_LINK_AXIS (int axis)

  Getter for LINK_AXIS axis parameter.
- uint GetAxisParameter_LOADED (int axis)

  Getter for LOADED axis parameter.
- double GetAxisParameter_LOOKAHEAD_FACTOR (int axis)

  Getter for LOOKAHEAD_FACTOR axis parameter.
- void SetAxisParameter_LOOKAHEAD_FACTOR (int axis, double value)

  Setter for LOOKAHEAD_FACTOR axis parameter.
- int GetAxisParameter_MARK (int axis)

  Getter for MARK axis parameter.
- int GetAxisParameter_MARKB (int axis)

  Getter for MARKB axis parameter.
- bool GetAxisParameter_MERGE (int axis)

  Getter for MERGE axis parameter.
- void SetAxisParameter_MERGE (int axis, bool value)

  Setter for MERGE axis parameter.
- int GetAxisParameter_MOVE_COUNT (int axis)

  Getter for MOVE_COUNT axis parameter.
- void SetAxisParameter_MOVE_COUNT (int axis, int value)

  Setter for MOVE_COUNT axis parameter.
- double GetAxisParameter_MOVE_PA (int axis)

  Getter for MOVE_PA axis parameter.
- void SetAxisParameter_MOVE_PA (int axis, double value)

  Setter for MOVE_PA axis parameter.
- int GetAxisParameter_MOVE_PA_CONT (int axis)

  Getter for MOVE_PA_CONT axis parameter.
- void SetAxisParameter_MOVE_PA_CONT (int axis, int value)

  Setter for MOVE_PA_CONT axis parameter.
- double GetAxisParameter_MOVE_PA_IDLE (int axis)

  Getter for MOVE_PA_IDLE axis parameter.
- void SetAxisParameter_MOVE_PA_IDLE (int axis, double value)

  Setter for MOVE_PA_IDLE axis parameter.
- double GetAxisParameter_MOVE_PB (int axis)

  Getter for MOVE_PB axis parameter.
- void SetAxisParameter_MOVE_PB (int axis, double value)

  Setter for MOVE_PB axis parameter.
- int GetAxisParameter_MOVE_PB_CONT (int axis)

  Getter for MOVE_PB_CONT axis parameter.
- void SetAxisParameter_MOVE_PB_CONT (int axis, int value)

  Setter for MOVE_PB_CONT axis parameter.

- double GetAxisParameter_MOVE_PB_IDLE (int axis)

  Getter for MOVE_PB_IDLE axis parameter.
- void SetAxisParameter_MOVE_PB_IDLE (int axis, double value)

  Setter for MOVE_PB_IDLE axis parameter.
- double GetAxisParameter_MOVELINK_MODIFY (int axis)

  Getter for MOVELINK_MODIFY axis parameter.
- void SetAxisParameter_MOVELINK_MODIFY (int axis, double value)

  Setter for MOVELINK_MODIFY axis parameter.
- short GetAxisParameter_MOVES_BUFFERED (int axis)

  Getter for MOVES_BUFFERED axis parameter.
- double GetAxisParameter_MPOS (int axis)

  Getter for MPOS axis parameter.
- double GetAxisParameter_MSPEED (int axis)

  Getter for MSPEED axis parameter.
- double GetAxisParameter_MSPEED_FILTER (int axis)

  Getter for MSPEED_FILTER axis parameter.
- void SetAxisParameter_MSPEED_FILTER (int axis, double value)

  Setter for MSPEED_FILTER axis parameter.
- double GetAxisParameter_MSPEEDF (int axis)

  Getter for MSPEEDF axis parameter.
- MotionType GetAxisParameter_MTYPE (int axis)

  Getter for MTYPE axis parameter.
- MotionType GetAxisParameter_NTYPE (int axis)

  Getter for NTYPE axis parameter.
- double GetAxisParameter_OFFPOS (int axis)

  Getter for OFFPOS axis parameter.
- void SetAxisParameter_OFFPOS (int axis, double value)

  Setter for OFFPOS axis parameter.
- double GetAxisParameter_OPEN_WIN (int axis)

  Getter for OPEN_WIN axis parameter.
- void SetAxisParameter_OPEN_WIN (int axis, double value)

  Setter for OPEN_WIN axis parameter.
- int GetAxisParameter_OUTLIMIT (int axis)

  Getter for OUTLIMIT axis parameter.
- void SetAxisParameter_OUTLIMIT (int axis, int value)

  Setter for OUTLIMIT axis parameter.
- double GetAxisParameter_OV_GAIN (int axis)

  Getter for OV_GAIN axis parameter.
- void SetAxisParameter_OV_GAIN (int axis, double value)

  Setter for OV_GAIN axis parameter.
- double GetAxisParameter_P_GAIN (int axis)

  Getter for P_GAIN axis parameter.
- void SetAxisParameter_P_GAIN (int axis, double value)

  Setter for P_GAIN axis parameter.
- double GetAxisParameter_POS_DELAY (int axis)

  Getter for POS_DELAY axis parameter.
- void SetAxisParameter_POS_DELAY (int axis, double value)

  Setter for POS_DELAY axis parameter.
- int GetAxisParameter_PP_STEP (int axis)

  Getter for PP_STEP axis parameter.
- void SetAxisParameter_PP_STEP (int axis, int value)

Setter for PP_STEP axis parameter.

- int GetAxisParameter_PS_ENCODER (int axis)

  Getter for PS_ENCODER axis parameter.
- double GetAxisParameter_RAISE_ANGLE (int axis)

  Getter for RAISE_ANGLE axis parameter.
- void SetAxisParameter_RAISE_ANGLE (int axis, double value)

  Setter for RAISE_ANGLE axis parameter.
- double GetAxisParameter_REG_POS (int axis)

  Getter for REG_POS axis parameter.
- double GetAxisParameter_REG_POSB (int axis)

  Getter for REG_POSB axis parameter.
- double GetAxisParameter_REMAIN (int axis)

  Getter for REMAIN axis parameter.
- double GetAxisParameter_REP_DIST (int axis)

  Getter for REP_DIST axis parameter.
- void SetAxisParameter_REP_DIST (int axis, double value)

  Setter for REP_DIST axis parameter.
- short GetAxisParameter_REP_OPTION (int axis)

  Getter for REP_OPTION axis parameter.
- void SetAxisParameter_REP_OPTION (int axis, short value)

  Setter for REP_OPTION axis parameter.
- short GetAxisParameter_REV_IN (int axis)

  Getter for REV_IN axis parameter.
- void SetAxisParameter_REV_IN (int axis, short value)

  Setter for REV_IN axis parameter.
- short GetAxisParameter_REV_JOG (int axis)

  Getter for REV_JOG axis parameter.
- void SetAxisParameter_REV_JOG (int axis, short value)

  Setter for REV_JOG axis parameter.
- double GetAxisParameter_ROBOT_DPOS (int axis)

  Getter for ROBOT_DPOS axis parameter.
- double GetAxisParameter_ROBOT_FS_LIMIT (int axis)

  Getter for ROBOT_FS_LIMIT axis parameter.
- void SetAxisParameter_ROBOT_FS_LIMIT (int axis, double value)

  Setter for ROBOT_FS_LIMIT axis parameter.
- double GetAxisParameter_ROBOT_RS_LIMIT (int axis)

  Getter for ROBOT_RS_LIMIT axis parameter.
- void SetAxisParameter_ROBOT_RS_LIMIT (int axis, double value)

  Setter for ROBOT_RS_LIMIT axis parameter.
- short GetAxisParameter_ROBOT_SP_MODE (int axis)

  Getter for ROBOT_SP_MODE axis parameter.
- void SetAxisParameter_ROBOT_SP_MODE (int axis, short value)

  Setter for ROBOT_SP_MODE axis parameter.
- RobotStatus GetAxisParameter_ROBOTSTATUS (int axis)

  Getter for ROBOTSTATUS axis parameter.
- double GetAxisParameter_ROBOT_UNITS (int axis)

  Getter for ROBOT_UNITS axis parameter.
- void SetAxisParameter_ROBOT_UNITS (int axis, double value)

  Setter for ROBOT_UNITS axis parameter.
- double GetAxisParameter_RS_LIMIT (int axis)

  Getter for RS_LIMIT axis parameter.

- void SetAxisParameter_RS_LIMIT (int axis, double value)

  Setter for RS_LIMIT axis parameter.
- bool GetAxisParameter_SERVO (int axis)

  Getter for SERVO axis parameter.
- void SetAxisParameter_SERVO (int axis, bool value)

  Setter for SERVO axis parameter.
- short GetAxisParameter_SLOT_NUMBER (int axis)

  Getter for SLOT_NUMBER axis parameter.
- void SetAxisParameter_SLOT_NUMBER (int axis, short value)

  Setter for SLOT_NUMBER axis parameter.
- double GetAxisParameter_SPEED (int axis)

  Getter for SPEED axis parameter.
- void SetAxisParameter_SPEED (int axis, double value)

  Setter for SPEED axis parameter.
- int GetAxisParameter_SRAMP (int axis)

  Getter for SRAMP axis parameter.
- void SetAxisParameter_SRAMP (int axis, int value)

  Setter for SRAMP axis parameter.
- int GetAxisParameter_S_REF (int axis)

  Getter for S_REF axis parameter.
- void SetAxisParameter_S_REF (int axis, int value)

  Setter for S_REF axis parameter.
- int GetAxisParameter_S_REF_OUT (int axis)

  Getter for S_REF_OUT axis parameter.
- void SetAxisParameter_S_REF_OUT (int axis, int value)

  Setter for S_REF_OUT axis parameter.
- double GetAxisParameter_START_DIR_LAST (int axis)

  Getter for START_DIR_LAST axis parameter.
- void SetAxisParameter_START_DIR_LAST (int axis, double value)

  Setter for START_DIR_LAST axis parameter.
- double GetAxisParameter_STARTMOVE_SPEED (int axis)

  Getter for STARTMOVE_SPEED axis parameter.
- void SetAxisParameter_STARTMOVE_SPEED (int axis, double value)

  Setter for STARTMOVE_SPEED axis parameter.
- double GetAxisParameter_STOP_ANGLE (int axis)

  Getter for STOP_ANGLE axis parameter.
- void SetAxisParameter_STOP_ANGLE (int axis, double value)

  Setter for STOP_ANGLE axis parameter.
- int GetAxisParameter_SYNC_AXIS (int axis)

  Getter for SYNC_AXIS axis parameter.
- void SetAxisParameter_SYNC_AXIS (int axis, int value)

  Setter for SYNC_AXIS axis parameter.
- SyncControl GetAxisParameter_SYNC_CONTROL (int axis)

  Getter for SYNC_CONTROL axis parameter.
- void SetAxisParameter_SYNC_CONTROL (int axis, SyncControl value)

  Setter for SYNC_CONTROL axis parameter.
- double GetAxisParameter_SYNC_DPOS (int axis)

  Getter for SYNC_DPOS axis parameter.
- void SetAxisParameter_SYNC_DPOS (int axis, double value)

  Setter for SYNC_DPOS axis parameter.
- int GetAxisParameter_SYNC_DWELL (int axis)

Getter for SYNC_DWELL axis parameter.

- void SetAxisParameter_SYNC_DWELL (int axis, int value)

  Setter for SYNC_DWELL axis parameter.
- int GetAxisParameter_SYNC_FLIGHT (int axis)

  Getter for SYNC_FLIGHT axis parameter.
- void SetAxisParameter_SYNC_FLIGHT (int axis, int value)

  Setter for SYNC_FLIGHT axis parameter.
- int GetAxisParameter_SYNC_PAUSE (int axis)

  Getter for SYNC_PAUSE axis parameter.
- void SetAxisParameter_SYNC_PAUSE (int axis, int value)

  Setter for SYNC_PAUSE axis parameter.
- int GetAxisParameter_SYNC_TIME (int axis)

  Getter for SYNC_TIME axis parameter.
- void SetAxisParameter_SYNC_TIME (int axis, int value)

  Setter for SYNC_TIME axis parameter.
- int GetAxisParameter_SYNC_TIMER (int axis)

  Getter for SYNC_TIMER axis parameter.
- double GetAxisParameter_SYNC_WITHDRAW (int axis)

  Getter for SYNC_WITHDRAW axis parameter.
- void SetAxisParameter_SYNC_WITHDRAW (int axis, double value)

  Setter for SYNC_WITHDRAW axis parameter.
- double GetAxisParameter_TABLE_POINTER (int axis)

  Getter for TABLE_POINTER axis parameter.
- void SetAxisParameter_TABLE_POINTER (int axis, double value)

  Setter for TABLE_POINTER axis parameter.
- double GetAxisParameter_TANG_DIRECTION (int axis)

  Getter for TANG_DIRECTION axis parameter.
- void SetAxisParameter_TANG_DIRECTION (int axis, double value)

  Setter for TANG_DIRECTION axis parameter.
- int GetAxisParameter_T_REF (int axis)

  Getter for T_REF axis parameter.
- void SetAxisParameter_T_REF (int axis, int value)

  Setter for T_REF axis parameter.
- int GetAxisParameter_T_REF_OUT (int axis)

  Getter for T_REF_OUT axis parameter.
- void SetAxisParameter_T_REF_OUT (int axis, int value)

  Setter for T_REF_OUT axis parameter.
- double GetAxisParameter_UNITS (int axis)

  Getter for UNITS axis parameter.
- void SetAxisParameter_UNITS (int axis, double value)

  Setter for UNITS axis parameter.
- double GetAxisParameter_VECTOR_BUFFERED (int axis)

  Getter for VECTOR_BUFFERED axis parameter.
- void SetAxisParameter_VECTOR_BUFFERED (int axis, double value)

  Setter for VECTOR_BUFFERED axis parameter.
- short GetAxisParameter_VERIFY (int axis)

  Getter for VERIFY axis parameter.
- void SetAxisParameter_VERIFY (int axis, short value)

  Setter for VERIFY axis parameter.
- double GetAxisParameter_VFF_GAIN (int axis)

  Getter for VFF_GAIN axis parameter.

- void SetAxisParameter_VFF_GAIN (int axis, double value)

  Setter for VFF_GAIN axis parameter.
- double GetAxisParameter_VP_ACCEL (int axis)

  Getter for VP_ACCEL axis parameter.
- double GetAxisParameter_VP_JERK (int axis)

  Getter for VP_JERK axis parameter.
- int GetAxisParameter_VP_MODE (int axis)

  Getter for VP_MODE axis parameter.
- void SetAxisParameter_VP_MODE (int axis, int value)

  Setter for VP_MODE axis parameter.
- double GetAxisParameter_VP_POSITION (int axis)

  Getter for VP_POSITION axis parameter.
- double GetAxisParameter_VP_SPEED (int axis)

  Getter for VP_SPEED axis parameter.
- double GetAxisParameter_WORLD_ACCEL (int axis)

  Getter for WORLD_ACCEL axis parameter.
- void SetAxisParameter_WORLD_ACCEL (int axis, double value)

  Setter for WORLD_ACCEL axis parameter.
- double GetAxisParameter_WORLD_DECEL (int axis)

  Getter for WORLD_DECEL axis parameter.
- void SetAxisParameter_WORLD_DECEL (int axis, double value)

  Setter for WORLD_DECEL axis parameter.
- double GetAxisParameter_WORLD_DPOS (int axis)

  Getter for WORLD_DPOS axis parameter.
- double GetAxisParameter_WORLD_FS_LIMIT (int axis)

  Getter for WORLD_FS_LIMIT axis parameter.
- void SetAxisParameter_WORLD_FS_LIMIT (int axis, double value)

  Setter for WORLD_FS_LIMIT axis parameter.
- double GetAxisParameter_WORLD_JOGSPEED (int axis)

  Getter for WORLD_JOGSPEED axis parameter.
- void SetAxisParameter_WORLD_JOGSPEED (int axis, double value)

  Setter for WORLD_JOGSPEED axis parameter.
- double GetAxisParameter_WORLD_RS_LIMIT (int axis)

  Getter for WORLD_RS_LIMIT axis parameter.
- void SetAxisParameter_WORLD_RS_LIMIT (int axis, double value)

  Setter for WORLD_RS_LIMIT axis parameter.
- double GetAxisParameter_WORLD_SPEED (int axis)

  Getter for WORLD_SPEED axis parameter.
- void SetAxisParameter_WORLD_SPEED (int axis, double value)

  Setter for WORLD_SPEED axis parameter.
- double GetAxisParameter_WORLD_UNITS (int axis)

  Getter for WORLD_UNITS axis parameter.
- void SetAxisParameter_WORLD_UNITS (int axis, double value)

  Setter for WORLD_UNITS axis parameter.
- bool GetAxisParameter_AXIS_ENABLE_OVERRIDE (int axis)

  Getter for AXIS_ENABLE_OVERRIDE axis parameter.
- void SetAxisParameter_AXIS_ENABLE_OVERRIDE (int axis, bool value)

  Setter for AXIS_ENABLE_OVERRIDE axis parameter.
- short GetAxisParameter_AXIS_DISPLAY (int axis)

  Getter for AXIS_DISPLAY axis parameter.
- void SetAxisParameter_AXIS_DISPLAY (int axis, short value)

Setter for AXIS_DISPLAY axis parameter.

- int GetAxisParameter_AXIS_Z_OUTPUT (int axis)

  Getter for AXIS_Z_OUTPUT axis parameter.
- void SetAxisParameter_AXIS_Z_OUTPUT (int axis, int value)

  Setter for AXIS_Z_OUTPUT axis parameter.
- long GetAxisParameter_DEMAND_EDGES (int axis)

  Getter for DEMAND_EDGES axis parameter.
- int GetAxisParameter_DRIVE_INDEX (int axis)

  Getter for DRIVE_INDEX axis parameter.
- void SetAxisParameter_DRIVE_INDEX (int axis, int value)

  Setter for DRIVE_INDEX axis parameter.
- int GetAxisParameter_DRIVE_NEG_TORQUE (int axis)

  Getter for DRIVE_NEG_TORQUE axis parameter.
- void SetAxisParameter_DRIVE_NEG_TORQUE (int axis, int value)

  Setter for DRIVE_NEG_TORQUE axis parameter.
- int GetAxisParameter_DRIVE_PARAMETER (int axis)

  Getter for DRIVE_PARAMETER axis parameter.
- void SetAxisParameter_DRIVE_PARAMETER (int axis, int value)

  Setter for DRIVE_PARAMETER axis parameter.
- int GetAxisParameter_DRIVE_POS_TORQUE (int axis)

  Getter for DRIVE_POS_TORQUE axis parameter.
- void SetAxisParameter_DRIVE_POS_TORQUE (int axis, int value)

  Setter for DRIVE_POS_TORQUE axis parameter.
- int GetAxisParameter_DRIVE_SET_VAL (int axis)

  Getter for DRIVE_SET_VAL axis parameter.
- void SetAxisParameter_DRIVE_SET_VAL (int axis, int value)

  Setter for DRIVE_SET_VAL axis parameter.
- int GetAxisParameter_DRIVE_VALUE (int axis)

  Getter for DRIVE_VALUE axis parameter.
- void SetAxisParameter_DRIVE_VALUE (int axis, int value)

  Setter for DRIVE_VALUE axis parameter.
- double GetAxisParameter_FEED_OVERRIDE (int axis)

  Getter for FEED_OVERRIDE axis parameter.
- void SetAxisParameter_FEED_OVERRIDE (int axis, double value)

  Setter for FEED_OVERRIDE axis parameter.
- double GetAxisParameter_FORCE_ACCEL (int axis)

  Getter for FORCE_ACCEL axis parameter.
- void SetAxisParameter_FORCE_ACCEL (int axis, double value)

  Setter for FORCE_ACCEL axis parameter.
- double GetAxisParameter_FORCE_DECEL (int axis)

  Getter for FORCE_DECEL axis parameter.
- void SetAxisParameter_FORCE_DECEL (int axis, double value)

  Setter for FORCE_DECEL axis parameter.
- double GetAxisParameter_FORCE_JERK (int axis)

  Getter for FORCE_JERK axis parameter.
- void SetAxisParameter_FORCE_JERK (int axis, double value)

  Setter for FORCE_JERK axis parameter.
- int GetAxisParameter_FRAME (int axis)

  Getter for FRAME axis parameter.
- void SetAxisParameter_FRAME (int axis, int value)

  Setter for FRAME axis parameter.

- short GetAxisParameter_FWD_START (int axis)

  Getter for FWD_START axis parameter.
- void SetAxisParameter_FWD_START (int axis, short value)

  Setter for FWD_START axis parameter.
- int GetAxisParameter_POSI_SEQ_DELAY (int axis)

  Getter for POSI_SEQ_DELAY axis parameter.
- void SetAxisParameter_POSI_SEQ_DELAY (int axis, int value)

  Setter for POSI_SEQ_DELAY axis parameter.
- int GetAxisParameter_POSI_SEQ_MODE (int axis)

  Getter for POSI_SEQ_MODE axis parameter.
- void SetAxisParameter_POSI_SEQ_MODE (int axis, int value)

  Setter for POSI_SEQ_MODE axis parameter.
- uint GetAxisParameter_PWM_CYCLE (int axis)

  Getter for PWM_CYCLE axis parameter.
- void SetAxisParameter_PWM_CYCLE (int axis, uint value)

  Setter for PWM_CYCLE axis parameter.
- uint GetAxisParameter_PWM_MARK (int axis)

  Getter for PWM_MARK axis parameter.
- void SetAxisParameter_PWM_MARK (int axis, uint value)

  Setter for PWM_MARK axis parameter.
- int GetAxisParameter_REG_INPUTS (int axis)

  Getter for REG_INPUTS axis parameter.
- void SetAxisParameter_REG_INPUTS (int axis, int value)

  Setter for REG_INPUTS axis parameter.
- int GetAxisParameter_REGIST_CONTROL (int axis)

  Getter for REGIST_CONTROL axis parameter.
- void SetAxisParameter_REGIST_CONTROL (int axis, int value)

  Setter for REGIST_CONTROL axis parameter.
- double GetAxisParameter_REGIST_DELAY (int axis)

  Getter for REGIST_DELAY axis parameter.
- void SetAxisParameter_REGIST_DELAY (int axis, double value)

  Setter for REGIST_DELAY axis parameter.
- double GetAxisParameter_REGIST_SPEED (int axis)

  Getter for REGIST_SPEED axis parameter.
- double GetAxisParameter_REGIST_SPEEDB (int axis)

  Getter for REGIST_SPEEDB axis parameter.
- short GetAxisParameter_REV_START (int axis)

  Getter for REV_START axis parameter.
- void SetAxisParameter_REV_START (int axis, short value)

  Setter for REV_START axis parameter.
- double GetAxisParameter_TCP_DPOS (int axis)

  Getter for TCP_DPOS axis parameter.
- double GetAxisParameter_TCP_FS_LIMIT (int axis)

  Getter for TCP_FS_LIMIT axis parameter.
- void SetAxisParameter_TCP_FS_LIMIT (int axis, double value)

  Setter for TCP_FS_LIMIT axis parameter.
- double GetAxisParameter_TCP_RS_LIMIT (int axis)

  Getter for TCP_RS_LIMIT axis parameter.
- void SetAxisParameter_TCP_RS_LIMIT (int axis, double value)

  Setter for TCP_RS_LIMIT axis parameter.
- double GetAxisParameter_TCP_UNITS (int axis)

Getter for TCP_UNITS axis parameter.

- void SetAxisParameter_TCP_UNITS (int axis, double value)

  Setter for TCP_UNITS axis parameter.
- double GetAxisParameter_ROBOT_FE (int axis)

  Getter for ROBOT_FE axis parameter.
- double GetAxisParameter_ROBOT_FE_LATCH (int axis)

  Getter for ROBOT_FE_LATCH axis parameter.
- double GetAxisParameter_ROBOT_FE_RANGE (int axis)

  Getter for ROBOT_FE_RANGE axis parameter.
- void SetAxisParameter_ROBOT_FE_RANGE (int axis, double value)

  Setter for ROBOT_FE_RANGE axis parameter.
- ProcessStatus GetProcessParameter_PROC_STATUS (int process)

  Getter for PROC_STATUS process parameter.
- int GetProcessParameter_ERROR_LINE (int process)

  Getter for ERROR_LINE process parameter.
- int GetProcessParameter_INDEVICE (int process)

  Getter for INDEVICE process parameter.
- void SetProcessParameter_INDEVICE (int process, int value)

  Setter for INDEVICE process parameter.
- int GetProcessParameter_OUTDEVICE (int process)

  Getter for OUTDEVICE process parameter.
- void SetProcessParameter_OUTDEVICE (int process, int value)

  Setter for OUTDEVICE process parameter.
- bool GetProcessParameter_PMOVE (int process)

  Getter for PMOVE process parameter.
- int GetProcessParameter_PROC_LINE (int process)

  Getter for PROC_LINE process parameter.
- int GetProcessParameter_RUN_ERROR (int process)

  Getter for RUN_ERROR process parameter.
- long GetProcessParameter_TICKS (int process)

  Getter for TICKS process parameter.
- void SetProcessParameter_TICKS (int process, long value)

  Setter for TICKS process parameter.
- int GetSlotParameter_SERIAL_NUMBER (int slot)

  Getter for SERIAL_NUMBER slot parameter.
- void SetSlotParameter_SERIAL_NUMBER (int slot, int value)

  Setter for SERIAL_NUMBER slot parameter.

Properties

- bool IsConnected `[get]`

  True if connection to Motion Coordinator was opened.
- ConnectionType ConnectionType `[get]`

  Connection type or current connection instance or Unknown if connection is disposed.

Events

- ConnectionEventHandler ConnectionEvent

  Connection related event.

6.3.1 Detailed Description

Connection interface.

Definition at line 2072 of file ITrioConnection.cs.

6.3.2 Member Function Documentation

6.3.2.1 AddAx() [1/2]

```
void Trio.UnifiedApi.ITrioConnection.AddAx (
            Int32 axis,
            Int32 linkAxis )
```

Function is used to superimpose 2 or more movements to build up a more complex movement profile.

Parameters

axis	Target axis for to be subject to superimposition
linkAxis	Axis to superimpose. -1 breaks the link with the other axis.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.2 AddAx() [2/2]

```
void Trio.UnifiedApi.ITrioConnection.AddAx (
            Int32 linkAxis )
```

Function is used to superimpose 2 or more movements to build up a more complex movement profile. Axis previously set by SetAxisBase() is subject to superimposition.

Parameters

linkAxis	Axis to superimpose. -1 breaks the link with the other axis.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.3 AddDac() [1/2]

```
void Trio.UnifiedApi.ITrioConnection.AddDac (
            Int32 axis,
            Int32 secondAxis )
```

Adds the output from the servo control block of a secondary axis to the output of the base axis. The resulting DAC_OUT of the base axis is then the sum of the two control loop outputs.

Parameters

axis	Base axis for to be subject to DAC_OUT superimposition
secondAxis	Axis which DAC_OUT value to superimpose. -1 breaks the link with the other axis.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.4 AddDac() [2/2]

```
void Trio.UnifiedApi.ITrioConnection.AddDac (
            Int32 secondAxis )
```

Adds the output from the servo control block of a secondary axis to the output of the base axis. The resulting DAC_OUT of the base axis is then the sum of the two control loop outputs. Axis previously set by SetAxisBase() is subject to superimposition of DAC_OUT value.

Parameters

secondAxis	Axis which DAC_OUT value to superimpose. -1 breaks the link with the other axis.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.5 Backlash()

```
void Trio.UnifiedApi.ITrioConnection.Backlash (
            double distance,
            double speed,
            double acceleration,
            Int32 axis = -1 )
```

This axis function allows backlash compensation to be loaded. This is achieved by applying an offset move when the motor demand is in one direction, then reversing the offset move when the motor demand is in the opposite direction. These moves are superimposed on the commanded axis movements. The backlash compensation is applied after a reversal of the direction of change of the DPOS parameter. The backlash compensation can be seen in the AXIS_DPOS axis parameter.This is effectively DPOS + backlash compensation.

Parameters

distance	The distance to be offset in user units
speed	The ACCEL / DECEL rate at which is compensation move is applied in user units
acceleration	Enables/disables the backlash compensation
axis	Base axis for this command. If value is -1 then the default base axis is used

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.6 BacklashDisable()

```
void Trio.UnifiedApi.ITrioConnection.BacklashDisable (
            Int32 axis = -1 )
```

This axis function disables backlash compensation.

Parameters

axis	Base axis for this command. If value is -1 then the default base axis is used

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.7 Cam()

```
void Trio.UnifiedApi.ITrioConnection.Cam (
            int tableStart,
            int tableStop,
            double tableMultiplier,
            double linkDistance,
            Int32 axis = -1 )
```

This command is used to generate movement of an axis according to a table of positions which define a movement profile.

Parameters

tableStart	The start position of the cam profile in the TABLE.
tableStop	The end position of the cam profile in the TABLE.
tableMultiplier	The table values are multiplied by this value to generate the positions.
linkDistance	The distance parameter relates the speed of the axis to the time taken to complete the cam profile.
axis	Base axis for this command. If value is -1 then the default base axis is used

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.8 CamBox()

```
void Trio.UnifiedApi.ITrioConnection.CamBox (
            int tableStart,
            int tableStop,
            double tableMultiplier,
            double linkDistance,
            int linkAxis,
            CamBoxLinkOptions linkOptions,
            double linkPos = 0,
            double offsetStart = 0,
            Int32 axis = -1 )
```

This command is used to generate movement of an axis according to a table of POSITIONS which define the movement profile.

Parameters

tableStart	The start position of the cam profile in the TABLE.
tableStop	The end position of the cam profile in the TABLE.
tableMultiplier	The table values are multiplied by this value to generate the positions.
linkDistance	The distance the link axis must move to complete CAMBOX profile.
linkAxis	The axis to link to.
linkOptions	Bit value options to customise how your CAMBOX operates.
linkPos	linkOptions bit 1 - the absolute position on the link axis in user UNITS where the CAMBOX is to be start. linkOptions bit 9 – the registration channel to start the movement on
offsetStart	The position defined on the link_dist where the profile will start.
axis	Base axis for this command. If value is -1 then the default base axis is used

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.9 Cancel()

```
void Trio.UnifiedApi.ITrioConnection.Cancel (
            Int32 mode,
            Int32 axis = -1 )
```

Used to cancel current or buffered axis commands on an axis or an interpolating axis group.

Parameters

mode	Cancel mode
axis	Index of axis subject to cancel operation or -1 to cancel default base axis

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.10 ClearVariables()

```
void Trio.UnifiedApi.ITrioConnection.ClearVariables ( )
```

Sets all global (numbered) variables and VR values to 0 and sets local variables on the process on which command is run to 0.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.11 CloseConnection()

```
void Trio.UnifiedApi.ITrioConnection.CloseConnection ( )
```

Closes previously open connection to the Motion Coordinator.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.12 Connect()

```
void Trio.UnifiedApi.ITrioConnection.Connect (
            double ratio,
            Int32 linkAxis,
            ConnectMode mode = ConnectMode.MPOS,
            Int32 axis = -1 )
```

Links the demand position of the base axis to the measured movements of the driving axes to produce an electronic gearbox.

Parameters

ratio	This parameter holds the number of edges the base axis is required to move per increment of the driving axis. The ratio value can be either positive or negative. The ratio is always specified as an encoder edge ratio
linkAxis	This parameter specifies the axis to link to.
mode	0 The axis follows the measured position(MPOS) of the driving axis. (default) 1 The axis follows the demand position (DPOS) of the driving axis.
axis	Index of axis subject to cancel operation or -1 to cancel default base axis

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.13 ConnPath()

```
void Trio.UnifiedApi.ITrioConnection.ConnPath (
            double ratio,
            Int32 linkAxis,
            Int32 axis = -1 )
```

Enables you to link to the path of an interpolated movement by linking the demand position of the base axis, to the interpolated path distance of the driving axis.

Parameters

ratio	This is the ratio between the interpolated distance moved on the driving axis to the distance moved on the base axis.
linkAxis	This parameter specifies the axis to link to.
axis	Index of axis subject to cancel operation or -1 to cancel default base axis

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.14 Datum()

```
void Trio.UnifiedApi.ITrioConnection.Datum (
            UInt32 sequence,
            Int32 axis = -1 )
```

Performs one of 6 homing sequences to locate an axis to an absolute position.

Parameters

sequence	Bit-mask for homing sequence to be executed
axis	Index of the axis to be subject of datum sequence or -1 to datum default base axis

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.15 DefPos()

```
void Trio.UnifiedApi.ITrioConnection.DefPos (
            double pos,
            Int32 axis = -1 )
```

Defines the current axis position as a new absolute value.

Parameters

pos	Absolute position to set on axis in user units
axis	Index of the axis for which the default position to be set or -1 to set position to default base axis

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.16 DefPosMulti()

```
void Trio.UnifiedApi.ITrioConnection.DefPosMulti (
            double[] pos,
            Int32 axis = -1 )
```

Defines the current axis position as a new absolute value to multiple axes.

Parameters

pos	Array of absolute positions in user units to set on axes defined by SetMultiAxisBase
axis	Index of the first axis for which the default position to be set or -1 to set position to default base axis

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.17 Delete()

```
void Trio.UnifiedApi.ITrioConnection.Delete (
            String remote_file,
            bool compiled_code_only = false )
```

Deletes a file on the Motion Coordinator.

Parameters

remote_file	The name of the file
compiled_code_only	In case of a program: when true only the compiled code of the program will be deleted, the source code will remain. When false all data related to the program will be deleted

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.18 DeleteFrameGroup()

```
void Trio.UnifiedApi.ITrioConnection.DeleteFrameGroup (
            UInt32 group )
```

Delete frame group. When the group is deleted FRAME is set to 0. See the TrioBASIC FRAME_GROUP documentation for more information.

Parameters

group	Frame group number to delete

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.19 Dir()

```
void Trio.UnifiedApi.ITrioConnection.Dir (
            FileCallback handler )
```

Function returns the list of programs and files in internal memory of the Motion Coordinator.

Parameters

handler	Callback handler function executed for each item in internal memory

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.20 DownloadFile()

```
void Trio.UnifiedApi.ITrioConnection.DownloadFile (
            String local_file,
            String remote_file,
            ProgressHandlerCallback handler )
```

Downloads a file from the Motion Coordinator.

Parameters

local_file	Local source file
remote_file	Remote file name. Can be prefixed by: "RAM:" - the location of the file is in the controller RAM. "SD:" - the location of the file is on the SD card. "FIFO:" - The target file is a FIFO file on the controller. If no prefix "RAM:" is assumed.
handler	Callback handler with the progress advance

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.21 EncoderRatio()

```
void Trio.UnifiedApi.ITrioConnection.EncoderRatio (
            Int64 mpos_count,
            Int64 input_count,
            Int32 axis = -1 )
```

This command allows the incoming encoder count to be scaled by a non integer ratio : MPOS = (mpos_count / input_count) x encoder_edges_input When using the servo loop you will need to adjust the gains to maintain performance and stability. Unlike the UNITS parameter, which only affects the scaling seen by the user programs, ENCODER_RATIO affects all motion commands. ENCODER_RATIO does not replace UNITS.Only use ENCODER_RATIO where absolutely necessary.PP_STEP and ENCODER_RATIO cannot be used at the same time on the same axis.

Parameters

mpos_count	An integer number which defines the numerator
input_count	An integer number which defines the denominator
axis	Base axis for this command. If value is -1 then the default base axis is used

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.22 Ethercat_CheckNumberOfSlaves()

```
UInt32 Trio.UnifiedApi.ITrioConnection.Ethercat_CheckNumberOfSlaves (
            Int32 slot )
```

Determine the number of slaves on the bus. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.

Returns

On success receives the number of slaves

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.23 Ethercat_CheckNumberOfSlaves_VR()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_CheckNumberOfSlaves_VR (
            Int32 slot,
            UInt32 out_vr_index )
```

Determine the number of slaves on the bus. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
out_vr_index	Result is placed in the selected VR variable

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.24 Ethercat_CheckSlaveOnline()

```
bool Trio.UnifiedApi.ITrioConnection.Ethercat_CheckSlaveOnline (
            Int32 slot,
            Int32 slave )
```

Check whether given slave is online. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
slave	Position on the network of the slave

Returns

On success true if slave is online or false otherwise

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.25 Ethercat_CheckSlaveOnline_VR()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_CheckSlaveOnline_VR (
            Int32 slot,
            Int32 slave,
            UInt32 out_vr_index )
```

Check whether given slave is online. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
slave	Position on the network of the slave
out_vr_index	Result is placed in the selected VR variable

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.26 Ethercat_CoRead()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_CoRead (
            Int32 slot,
            Int32 address,
            UInt16 object_index,
            Byte object_subindex,
            Co_ObjectType type,
            Int32 out_vr_index )
```

This function gets a CANopen-over-EtherCAT (CoE) object from the remote drive or I/O device. The object's index and sub-index are used to request a value and that value is either placed in the VR or is displayed in the Motion Perfect terminal if the VR number is set to -1. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
address	Node address of the remote device on the network
object_index	CANopen Object index
object_subindex	CANopen Object sub-index
type	1 : Boolean 2 : Integer 8 3 : Integer 16 4 : Integer 32 5 : Unsigned 8 6 : Unsigned 16 7 : Unsigned 32 8 : Real 32 17 : Real 64 21 : Integer 64 27 : Unsigned 64
out_vr_index	Result is placed in the selected VR variable

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.27 Ethercat_CoReadAxis()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_CoReadAxis (
            Int32 axis_number,
            UInt16 object_index,
            Byte object_subindex,
            Co_ObjectType type,
            Int32 out_vr_index )
```

This function gets a CANopen-over-EtherCAT (CoE) object from the remote drive or I/O device. The object's index and sub-index are used to request a value and that value is either placed in the VR or is displayed in the Motion Perfect terminal if the VR number is set to -1. Refer to the remote device's manual for a list of available objects. See Trio BASIC help for more details.

Parameters

axis_number	Axis number of the EtherCAT drive.
object_index	CANopen Object index
object_subindex	CANopen Object sub-index
type	1 : Boolean 2 : Integer 8 3 : Integer 16 4 : Integer 32 5 : Unsigned 8 6 : Unsigned 16 7 : Unsigned 32 8 : Real 32 17 : Real 64 21 : Integer 64 27 : Unsigned 64
out_vr_index	Result is placed in the selected VR variable

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.28 Ethercat_CoWrite_Value() [1/2]

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_CoWrite_Value (
            Int32 slot,
            Int32 address,
            UInt16 object_index,
            Byte object_subindex,
            Co_ObjectType type,
            double value )
```

This function sets a CANopen-over-EtherCAT (CoE) object in the remote drive or I/O device. The object's index and sub-index are used to write a value to that object. Use this function for floating-point values. Refer to the remote device's manual for a list of available objects. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.

Parameters

address	Node address of the remote device on the network
object_index	CANopen Object index
object_subindex	CANopen Object sub-index
type	1 : Boolean 2 : Integer 8 3 : Integer 16 4 : Integer 32 5 : Unsigned 8 6 : Unsigned 16 7 : Unsigned 32 8 : Real 32 17 : Real 64 21 : Integer 64 27 : Unsigned 64
value	The value

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.29 Ethercat_CoWrite_Value() [2/2]

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_CoWrite_Value (
            Int32 slot,
            Int32 address,
            UInt16 object_index,
            Byte object_subindex,
            Co_ObjectType type,
            Int64 value )
```

This function sets a CANopen-over-EtherCAT (CoE) object in the remote drive or I/O device. The object's index and sub-index are used to write a value to that object. Use this function for integer values. Refer to the remote device's manual for a list of available objects. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
address	Node address of the remote device on the network
object_index	CANopen Object index
object_subindex	CANopen Object sub-index
type	1 : Boolean 2 : Integer 8 3 : Integer 16 4 : Integer 32 5 : Unsigned 8 6 : Unsigned 16 7 : Unsigned 32 8 : Real 32 17 : Real 64 21 : Integer 64 27 : Unsigned 64
value	The value

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.30 Ethercat_CoWrite_VR()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_CoWrite_VR (
            Int32 slot,
            Int32 address,
            UInt16 object_index,
            Byte object_subindex,
            Co_ObjectType type,
            UInt32 vr_index )
```

This function sets a CANopen-over-EtherCAT (CoE) object in the remote drive or I/O device. The object's index and sub-index are used to write a value to that object. The value comes from a VR. Use this function for integer values. Refer to the remote device's manual for a list of available objects. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
address	Node address of the remote device on the network
object_index	CANopen Object index
object_subindex	CANopen Object sub-index
type	1 : Boolean 2 : Integer 8 3 : Integer 16 4 : Integer 32 5 : Unsigned 8 6 : Unsigned 16 7 : Unsigned 32 8 : Real 32 17 : Real 64 21 : Integer 64 27 : Unsigned 64
vr_index	VR number between 0 and max VR where the result will be stored. (-1 means the value will be printed to the terminal)

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.31 Ethercat_CoWriteAxis_Value() [1/2]

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_CoWriteAxis_Value (
            Int32 axis_number,
            UInt16 object_index,
            Byte object_subindex,
            Co_ObjectType type,
            double value )
```

This function sets a CANopen-over-EtherCAT (CoE) object in the remote drive or I/O device. The object's index and sub-index are used to write a value to that object. Use this function for floating-point values. Refer to the remote device's manual for a list of available objects. See Trio BASIC help for more details.

Parameters

axis_number	Axis number of the EtherCAT drive.
object_index	CANopen Object index
object_subindex	CANopen Object sub-index
type	1 : Boolean 2 : Integer 8 3 : Integer 16 4 : Integer 32 5 : Unsigned 8 6 : Unsigned 16 7 : Unsigned 32 8 : Real 32 17 : Real 64 21 : Integer 64 27 : Unsigned 64
value	The value.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.32 Ethercat_CoWriteAxis_Value() [2/2]

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_CoWriteAxis_Value (
            Int32 axis_number,
            UInt16 object_index,
            Byte object_subindex,
            Co_ObjectType type,
            Int64 value )
```

This function sets a CANopen-over-EtherCAT (CoE) object in the remote drive or I/O device. The object's index and sub-index are used to write a value to that object. Use this function for integer values. Refer to the remote device's manual for a list of available objects. See Trio BASIC help for more details.

Parameters

axis_number	Axis number of the EtherCAT drive.
object_index	CANopen Object index
object_subindex	CANopen Object sub-index
type	1 : Boolean 2 : Integer 8 3 : Integer 16 4 : Integer 32 5 : Unsigned 8 6 : Unsigned 16 7 : Unsigned 32 8 : Real 32 17 : Real 64 21 : Integer 64 27 : Unsigned 64

Parameters

value	The value.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.33 Ethercat_CoWriteAxis_VR()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_CoWriteAxis_VR (
            Int32 axis_number,
            UInt16 object_index,
            Byte object_subindex,
            Co_ObjectType type,
            UInt32 vr_index )
```

This function sets a CANopen-over-EtherCAT (CoE) object in the remote drive or I/O device. The object's index and sub-index are used to write a value to that object. The value comes from a VR. Refer to the remote device's manual for a list of available objects. See Trio BASIC help for more details.

Parameters

axis_number	Axis number of the EtherCAT drive.
object_index	CANopen Object index
object_subindex	CANopen Object sub-index
type	1 : Boolean 2 : Integer 8 3 : Integer 16 4 : Integer 32 5 : Unsigned 8 6 : Unsigned 16 7 : Unsigned 32 8 : Real 32 17 : Real 64 21 : Integer 64 27 : Unsigned 64
vr_index	VR number between 0 and max VR where the value is stored.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.34 Ethercat_DisplayNetworkConfig()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_DisplayNetworkConfig (
            Int32 slot )
```

Displays the network configuration to the command line terminal in Motion Perfect. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.35 Ethercat_GetEmergencyMsgControl()

```
UInt32 Trio.UnifiedApi.ITrioConnection.Ethercat_GetEmergencyMsgControl (
            Int32 slot )
```

Get the current value; indicating the response of the master to an emergency telegram transmitted by a slave node. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.

Returns

On success the current value

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.36 Ethercat_GetEmergencyMsgControl_VR()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_GetEmergencyMsgControl_VR (
            Int32 slot,
            UInt32 out_vr_index )
```

Get the current value; indicating the response of the master to an emergency telegram transmitted by a slave node. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
out_vr_index	Result is placed in the selected VR variable

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.37 Ethercat_GetSlave_ESC_Register()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_GetSlave_ESC_Register (
            Int32 slot,
            Int32 slave,
            UInt32 esc_offset,
            UInt32 data_length,
            UInt32 out_vr_index )
```

Low-level read from the slave's ESC registers, addressing method used is auto-increment addressing. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
slave	Auto-increment address
esc_offset	Register address
data_length	Number of bytes in the register size
out_vr_index	Result is placed in the selected VR variable

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.38 Ethercat_GetSlaveAddress()

```
Int32 Trio.UnifiedApi.ITrioConnection.Ethercat_GetSlaveAddress (
            Int32 slot,
            Int32 slave )
```

Determine the configured address of the slave identified by its position on the network. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
slave	Position on the network of the slave

Returns

On success receives the configured slave address

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.39 Ethercat_GetSlaveAddress_VR()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_GetSlaveAddress_VR (
            Int32 slot,
            Int32 slave,
            UInt32 out_vr_index )
```

Determine the configured address of the slave identified by its position on the network. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
slave	Position on the network of the slave
out_vr_index	Result is placed in the selected VR variable

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.40 Ethercat_GetSlaveAxis()

```
Int32 Trio.UnifiedApi.ITrioConnection.Ethercat_GetSlaveAxis (
            Int32 slot,
            Int32 slave )
```

Determine the axis of the slave identified by its position on the network. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
slave	Position on the network of the slave

Returns

On success receives the axis assigned to the slave

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.41 Ethercat_GetSlaveAxis_VR()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_GetSlaveAxis_VR (
            Int32 slot,
            Int32 slave,
            UInt32 out_vr_index )
```

Determine the axis of the slave identified by its position on the network. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
slave	Position on the network of the slave
out_vr_index	Result is placed in the selected VR variable

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.42 Ethercat_GetState()

```
EthercatState Trio.UnifiedApi.ITrioConnection.Ethercat_GetState (
            Int32 slot )
```

Gets the present state of the EtherCAT running on the defined slot. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.

Returns

On success redeives the state. The returned value means:
0 : Initial (EtherCAT ESC value 0x01)
1 : Pre-Operational (0x02)
2 : Safe-Operational (0x04)
3 : Operational (0x08)

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.43 Ethercat_GetState_VR()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_GetState_VR (
            Int32 slot,
            UInt32 out_vr_index )
```

Gets the present state of the EtherCAT running on the defined slot. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
out_vr_index	Result is placed in the selected VR variable. The returned value means: 0 : Initial (EtherCAT ESC value 0x01) 1 : Pre-Operational (0x02) 2 : Safe-Operational (0x04) 3 : Operational (0x08)

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.44 Ethercat_GetStatusMaskAndTriggerPattern()

```
Tuple< UInt32, UInt32 > Trio.UnifiedApi.ITrioConnection.Ethercat_GetStatusMaskAndTrigger↩
Pattern (
            Int32 slot,
            Int32 addr )
```

This command enables the user to display and check the mask and trigger pattern values set (if any) for a given slave. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port
addr	Configured address of the slave

Returns

On success a tuple is returned containing following:

- Current statusword mask
- Current trigger pattern

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.45 Ethercat_GetStatusMaskAndTriggerPattern_VR()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_GetStatusMaskAndTriggerPattern_VR (
            Int32 slot,
            Int32 addr,
            UInt32 out_vr_index )
```

This command enables the user to display and check the mask and trigger pattern values set (if any) for a given slave. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
addr	Configured address of the slave
out_vr_index	VR index in range [0..max_vr_index-2]. Index of the array of 2 VRs where the mask and the trigger_pattern will be stored.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.46 Ethercat_SendResetSequence() [1/3]

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_SendResetSequence (
            Int32 axis )
```

Reset a slave error. This function runs the error reset sequence on the drive control word. DRIVE_CONTROLWORD bit 8 is toggled high then low. This will instruct the drive to reset any errors in the drive where the cause of the error has been removed. See Trio BASIC help for more details.

Parameters

axis	The axis number of the drive to be reset

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

Default timeout is 100 ms

6.3.2.47 Ethercat_SendResetSequence() [2/3]

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_SendResetSequence (
            Int32 axis,
            UInt32 mode )
```

Reset a slave error. This function runs the error reset sequence on the drive control word. DRIVE_CONTROLWORD bit 8 is toggled high then low. This will instruct the drive to reset any errors in the drive where the cause of the error has been removed. See Trio BASIC help for more details.

Parameters

axis	The axis number of the drive to be reset
mode	0 : The 'Fault Reset' (bit 7) of DS402 control word is set high and then set low again after a hard coded timeout. (default) 1 : Bit 7 is set high until the 'Fault Flag' (bit 3) of the status word goes low, or a timeout occurs.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

Default timeout is 100 ms

6.3.2.48 Ethercat_SendResetSequence() [3/3]

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_SendResetSequence (
            Int32 axis,
            UInt32 mode,
            UInt32 timeout )
```

Reset a slave error. This function runs the error reset sequence on the drive control word. DRIVE_CONTROLWORD bit 8 is toggled high then low. This will instruct the drive to reset any errors in the drive where the cause of the error has been removed. See Trio BASIC help for more details.

Parameters

axis	The axis number of the drive to be reset
mode	0 : The 'Fault Reset' (bit 7) of DS402 control word is set high and then set low again after a hard coded timeout. (default) 1 : Bit 7 is set high until the 'Fault Flag' (bit 3) of the status word goes low, or a timeout occurs.
timeout	Timeout in ms used during mode 1 operation. Default is 100 ms. Range is 1 to 10000 ms.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.49 Ethercat_SetDistrClockMaster()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_SetDistrClockMaster (
            Int32 slot,
            UInt32 function,
            UInt32 value )
```

Set the EtherCAT master to enable the first DC enabled node to be the Distributed Clocks master. As this requires the Trio Motion Coordinator to synchronise to the external DC master, it does not function on P876, MC4N-ECAT or MC664(X) as these have FPGA based fixed timing hardware for the EtherCAT port. The function returns TRUE / FALSE depending upon whether the operation was successful. This command must be run before the EtherCAT network is started. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
function	Distributed Clocks function. 2 = use first DC enabled node as clocks master.
value	The value. 0 = use first DC enabled node as the clocks master.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.50 Ethercat_SetDistrClockMaster_VR()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_SetDistrClockMaster_VR (
```

```
            Int32 slot,
            UInt32 function,
            UInt32 vr_index )
```

Set the EtherCAT master to enable the first DC enabled node to be the Distributed Clocks master. As this requires the Trio Motion Coordinator to synchronise to the external DC master, it does not function on P876, MC4N-ECAT or MC664(X) as these have FPGA based fixed timing hardware for the EtherCAT port. The function returns TRUE / FALSE depending upon whether the operation was successful. This command must be run before the EtherCAT network is started. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
function	Distributed Clocks function. 2 = use first DC enabled node as clocks master.
vr_index	VR index in range [0..max_vr_index-1]. Use the value in the VR number specified.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.51 Ethercat_SetEmergencyMsgControl()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_SetEmergencyMsgControl (
            Int32 slot,
            UInt32 mode )
```

Set the response of the master to an emergency telegram transmitted by a slave node. This function enables the user to determine whether emergency messages should trigger a system error alarm and cause the WDOG to be dropped. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
mode	0 : Log error and raise a System Error and drop WDOG off. (Default) 1 : Log error only.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.52 Ethercat_SetSlave_ESC_Register()

```
unsafe void Trio.UnifiedApi.ITrioConnection.Ethercat_SetSlave_ESC_Register (
            Int32 slot,
            Int32 slave,
            UInt32 esc_offset,
            UInt32 data_length,
            Byte * data )
```

Low-level write to the slave's ESC registers, addressing method used is auto-increment addressing. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
slave	Auto-increment address
esc_offset	Register address
data_length	Number of data bytes to write
data	Data values

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.53 Ethercat_SetState()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_SetState (
            Int32 slot,
            EthercatState state,
            UInt32 display )
```

This function controls the EtherCAT State Machine (ESM). It requests the master change to given EtherCAT 'state', and hence changes all slaves to the same state. When a change to a higher state is made, the EtherCAT network will progress to the new state through the in-between states to allow correct starting of the network. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
state	-1 : Reserved 0 : Initial (EtherCAT ESC value 0x01) 1 : Pre-Operational (0x02) 2 : Safe-Operational (0x04) 3 : Operational (0x08)
display	0 : Writes state change information to the standard output stream. 1 : Do not write out state change information.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.54 Ethercat_SetStatusMaskAndTriggerPattern()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_SetStatusMaskAndTriggerPattern (
            Int32 slot,
            Int32 addr,
            UInt32 mask,
            UInt32 trigger_pattern )
```

This command enables the user to define a CoE Statusword (0x6041:00) mask and trigger pattern for raising the controller AXISSTATUS 'Remote Drive Status word Error' flag (bit 26.) The mask defines which bits within the CoE

status word are relevant, and the trigger pattern defines the state of these bits(0 or 1) which will trigger the controller AXISSTATUS flag. If it is necessary to reset the watchdog(master enable) when this AXISISTATUS flag is set then the axis ERRORMASK must be set accordingly. The slave mask and pattern can be set and modified from preop to operational state of the network(after the configured addresses have been assigned.) It must be reset if the network is restarted. The slave must support the CoE statusword(0x6041:00) within the PDO cyclic data. If the Mask is zero the controller will not execute any runtime check of the CoE statusword. If the watchdog must be cleared to OFF when the 'remote drive statusword error' flag is raised then bit 26 of the axis ERRORMASK parameter must be set. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
addr	Configured address of the slave
mask	CoE statusword mask.
trigger_pattern	Defines the state of the relevant bits which will trigger the controller AXISSTATUS 'remote drive statusword error' flag (bit 26.)

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.55 Ethercat_SetStatusMaskAndTriggerPattern_VR()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_SetStatusMaskAndTriggerPattern_VR (
            Int32 slot,
            Int32 addr,
            UInt32 vr_index )
```

This command enables the user to define a CoE Statusword (0x6041:00) mask and trigger pattern for raising the controller AXISSTATUS 'Remote Drive Status word Error' flag (bit 26.) The mask defines which bits within the CoE status word are relevant, and the trigger pattern defines the state of these bits(0 or 1) which will trigger the controller AXISSTATUS flag. If it is necessary to reset the watchdog(master enable) when this AXISISTATUS flag is set then the axis ERRORMASK must be set accordingly. The slave mask and pattern can be set and modified from preop to operational state of the network(after the configured addresses have been assigned.) It must be reset if the network is restarted. The slave must support the CoE statusword(0x6041:00) within the PDO cyclic data. If the Mask is zero the controller will not execute any runtime check of the CoE statusword. If the watchdog must be cleared to OFF when the 'remote drive statusword error' flag is raised then bit 26 of the axis ERRORMASK parameter must be set. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
addr	Configured address of the slave
vr_index	VR index in range [0..max_vr_index-1]. The index of the VR which contains the mask and the next VR (vr_index+1) contains the trigger_pattern.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.56 Ethercat_SetSync0StartTime()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_SetSync0StartTime (
            Int32 offset )
```

Set the network default Sync0 start time offset. If offset value is directly provided as a parameter. The default start time is 1 second(1000000 microseconds).Add or subtract the required offset in microseconds.For example 1000150 is an offset of 150 microseconds and 999930 is an offset of -70 microseconds.The offset must not exceed ±50 % of the servo period. The function returns TRUE / FALSE depending upon whether the operation was successful. This command must be run before the EtherCAT network is started. See Trio BASIC help for more details.

Parameters

offset	Direct value for the start time

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.57 Ethercat_SetSync0StartTime_VR()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_SetSync0StartTime_VR (
            UInt32 vr_index )
```

Set the network default Sync0 start time offset. The offset value is from the given VR. The default start time is 1 second(1000000 microseconds).Add or subtract the required offset in microseconds.For example 1000150 is an offset of 150 microseconds and 999930 is an offset of -70 microseconds.The offset must not exceed ±50 % of the servo period. The function returns TRUE / FALSE depending upon whether the operation was successful. This command must be run before the EtherCAT network is started. See Trio BASIC help for more details.

Parameters

vr_index	VR index in range [0..max_vr_index-1]. Use the value in the VR number specified.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.58 Ethercat_StartNetwork() [1/3]

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_StartNetwork (
            Int32 slot )
```

Initialise EtherCAT network, and put it onto operational mode. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.59 Ethercat_StartNetwork() [2/3]

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_StartNetwork (
            Int32 slot,
            EthercatStartOption options )
```

Initialise EtherCAT network, and put it onto operational mode. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
options	bit 0: 0-Do not print progress information; 1-Print out progress information (default) bit 1: 0-Normal state change (default); 1-Force a re-scan of the network bit 2: 0-Standard startup; 1-Express startup; do not wait for PLL to lock between state changes

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.60 Ethercat_StartNetwork() [3/3]

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_StartNetwork (
            Int32 slot,
            EthercatStartOption options,
            UInt32 MAC_retries )
```

Initialise EtherCAT network, and put it onto operational mode. See Trio BASIC help for more details.

Parameters

slot	Slot number of EtherCAT port.
options	bit 0: 0-Do not print progress information; 1-Print out progress information (default) bit 1: 0-Normal state change (default); 1-Force a re-scan of the network bit 2: 0-Standard startup; 1-Express startup; do not wait for PLL to lock between state changes
MAC_retries	Sets the number of times the master attempts to restart the Ethernet auto-negotiation. Default = 2

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.61 Ethercat_StopNetwork()

```
void Trio.UnifiedApi.ITrioConnection.Ethercat_StopNetwork (
            Int32 slot )
```

Closedown the EtherCAT network.

Parameters

slot	Slot number of EtherCAT port.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.62 Ex()

```
void Trio.UnifiedApi.ITrioConnection.Ex (
            Int32 mode = 0 )
```

Perform software reset on Motion Coordinator.

Parameters

mode	Reset mode. See the TrioBASIC EX documentation for more details.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.63 Execute()

```
void Trio.UnifiedApi.ITrioConnection.Execute (
            String command )
```

Execute arbitrary command.

Parameters

command	Command to be executed

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.64 ExecuteWithResponse() [1/3]

```
String Trio.UnifiedApi.ITrioConnection.ExecuteWithResponse (
            String command )
```

Execute arbitrary command with response.

Parameters

command	Command to be executed

Returns

Command response

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.65 ExecuteWithResponse() [2/3]

```
unsafe void Trio.UnifiedApi.ITrioConnection.ExecuteWithResponse (
            String command,
            Byte * buff,
            ref UInt32 buff_size )
```

Execute arbitrary command with response.

Parameters

command	Command to be executed
buff	Buffer to receive the response string characters
buff_size	Size of the buffer on input and on exit required buffer size including null terminator

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.66 ExecuteWithResponse() [3/3]

```
void Trio.UnifiedApi.ITrioConnection.ExecuteWithResponse (
            String command,
            StreamHandlerCallback handler )
```

Execute arbitrary command with response.

Parameters

command	Command to be executed
handler	Callback function to receive response string chunks

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.67 Forward()

```
void Trio.UnifiedApi.ITrioConnection.Forward (
            Int32 axis = -1 )
```

Sets continuous forward movement.

Parameters

axis	Axis index to set forward movement to or -1 to apply to default base axis.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.68 GetAIn()

```
Int32 Trio.UnifiedApi.ITrioConnection.GetAIn (
            UInt32 channel )
```

Get a value from an Analogue Input.

Parameters

channel	Analogue input channel number

Returns

Current value of the analog input channel

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.69 GetAOut()

```
Int32 Trio.UnifiedApi.ITrioConnection.GetAOut (
            UInt32 channel )
```

Get the state of the Analogue Output.

Parameters

channel	Analogue output channel number

Returns

Current value of the analogue output channel

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.70 GetApiVersion()

```
Version Trio.UnifiedApi.ITrioConnection.GetApiVersion ( )
```

Get Unified API version info.

Returns

On success version info of the Unified API

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.71 GetApiVersionStr()

```
String Trio.UnifiedApi.ITrioConnection.GetApiVersionStr ( )
```

Get Unified API version string.

Returns

On success string representation of the Unified API version

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.72 GetAxisParameter_ACCEL()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ACCEL (
            int axis )
```

Getter for ACCEL axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.73 GetAxisParameter_ADDAX_AXIS()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ADDAX_AXIS (
            int axis )
```

Getter for ADDAX_AXIS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.74 GetAxisParameter_AFF_GAIN()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_AFF_GAIN (
            int axis )
```

Getter for AFF_GAIN axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.75 GetAxisParameter_ATYPE()

```
AxisType Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ATYPE (
            int axis )
```

Getter for ATYPE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.76 GetAxisParameter_AXIS_ACCEL()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_AXIS_ACCEL (
            int axis )
```

Getter for AXIS_ACCEL axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.77 GetAxisParameter_AXIS_ADDRESS()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_AXIS_ADDRESS (
            int axis )
```

Getter for AXIS_ADDRESS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.78 GetAxisParameter_AXIS_DECEL()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_AXIS_DECEL (
            int axis )
```

Getter for AXIS_DECEL axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.79 GetAxisParameter_AXIS_DISPLAY()

```
short Trio.UnifiedApi.ITrioConnection.GetAxisParameter_AXIS_DISPLAY (
```

```
            int axis )
```

Getter for AXIS_DISPLAY axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.80 GetAxisParameter_AXIS_DPOS()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_AXIS_DPOS (
            int axis )
```

Getter for AXIS_DPOS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.81 GetAxisParameter_AXIS_ENABLE()

```
bool Trio.UnifiedApi.ITrioConnection.GetAxisParameter_AXIS_ENABLE (
            int axis )
```

Getter for AXIS_ENABLE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.82 GetAxisParameter_AXIS_ENABLE_OVERRIDE()

```
bool Trio.UnifiedApi.ITrioConnection.GetAxisParameter_AXIS_ENABLE_OVERRIDE (
            int axis )
```

Getter for AXIS_ENABLE_OVERRIDE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.83 GetAxisParameter_AXIS_ERROR_COUNT()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_AXIS_ERROR_COUNT (
            int axis )
```

Getter for AXIS_ERROR_COUNT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.84 GetAxisParameter_AXIS_FS_LIMIT()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_AXIS_FS_LIMIT (
            int axis )
```

Getter for AXIS_FS_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.85 GetAxisParameter_AXIS_MAX_JOG_SPEED()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_AXIS_MAX_JOG_SPEED (
            int axis )
```

Getter for AXIS_MAX_JOG_SPEED axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.86 GetAxisParameter_AXIS_MODE()

```
AxisMode Trio.UnifiedApi.ITrioConnection.GetAxisParameter_AXIS_MODE (
            int axis )
```

Getter for AXIS_MODE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.87 GetAxisParameter_AXIS_RS_LIMIT()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_AXIS_RS_LIMIT (
            int axis )
```

Getter for AXIS_RS_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.88 GetAxisParameter_AXIS_SPEED()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_AXIS_SPEED (
            int axis )
```

Getter for AXIS_SPEED axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.89 GetAxisParameter_AXIS_UNITS()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_AXIS_UNITS (
            int axis )
```

Getter for AXIS_UNITS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.90 GetAxisParameter_AXIS_Z_OUTPUT()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_AXIS_Z_OUTPUT (
            int axis )
```

Getter for AXIS_Z_OUTPUT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.91 GetAxisParameter_AXISSTATUS()

```
AxisStatus Trio.UnifiedApi.ITrioConnection.GetAxisParameter_AXISSTATUS (
```

```
            int axis )
```

Getter for AXISSTATUS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.92 GetAxisParameter_BACKLASH_DIST()

```
long Trio.UnifiedApi.ITrioConnection.GetAxisParameter_BACKLASH_DIST (
            int axis )
```

Getter for BACKLASH_DIST axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.93 GetAxisParameter_CHANGE_DIR_LAST()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_CHANGE_DIR_LAST (
            int axis )
```

Getter for CHANGE_DIR_LAST axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.94 GetAxisParameter_CLOSE_WIN()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_CLOSE_WIN (
            int axis )
```

Getter for CLOSE_WIN axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.95 GetAxisParameter_CLUTCH_RATE()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_CLUTCH_RATE (
            int axis )
```

Getter for CLUTCH_RATE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.96 GetAxisParameter_CORNER_MODE()

```
CornerMode Trio.UnifiedApi.ITrioConnection.GetAxisParameter_CORNER_MODE (
            int axis )
```

Getter for CORNER_MODE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.97 GetAxisParameter_CORNER_STATE()

```
CornerState Trio.UnifiedApi.ITrioConnection.GetAxisParameter_CORNER_STATE (
            int axis )
```

Getter for CORNER_STATE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.98 GetAxisParameter_CREEP()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_CREEP (
            int axis )
```

Getter for CREEP axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.99 GetAxisParameter_D_GAIN()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_D_GAIN (
            int axis )
```

Getter for D_GAIN axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.100 GetAxisParameter_DAC()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DAC (
            int axis )
```

Getter for DAC axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.101 GetAxisParameter_DAC_OUT()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DAC_OUT (
            int axis )
```

Getter for DAC_OUT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.102 GetAxisParameter_DAC_SCALE()

```
long Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DAC_SCALE (
            int axis )
```

Getter for DAC_SCALE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.103 GetAxisParameter_DATUM_IN()

```
short Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DATUM_IN (
```

```
            int axis )
```

Getter for DATUM_IN axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.104 GetAxisParameter_DECEL()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DECEL (
            int axis )
```

Getter for DECEL axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.105 GetAxisParameter_DECEL_ANGLE()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DECEL_ANGLE (
            int axis )
```

Getter for DECEL_ANGLE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.106 GetAxisParameter_DEMAND_EDGES()

```
long Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DEMAND_EDGES (
            int axis )
```

Getter for DEMAND_EDGES axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.107 GetAxisParameter_DEMAND_SPEED()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DEMAND_SPEED (
            int axis )
```

Getter for DEMAND_SPEED axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.108 GetAxisParameter_DPOS()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DPOS (
            int axis )
```

Getter for DPOS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.109 GetAxisParameter_DRIVE_CONTROL()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_CONTROL (
            int axis )
```

Getter for DRIVE_CONTROL axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.110 GetAxisParameter_DRIVE_CONTROLWORD()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_CONTROLWORD (
            int axis )
```

Getter for DRIVE_CONTROLWORD axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.111 GetAxisParameter_DRIVE_CURRENT()

```
short Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_CURRENT (
            int axis )
```

Getter for DRIVE_CURRENT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.112 GetAxisParameter_DRIVE_CW_MODE()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_CW_MODE (
            int axis )
```

Getter for DRIVE_CW_MODE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.113 GetAxisParameter_DRIVE_ENABLE()

```
long Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_ENABLE (
            int axis )
```

Getter for DRIVE_ENABLE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.114 GetAxisParameter_DRIVE_FE()

```
long Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_FE (
            int axis )
```

Getter for DRIVE_FE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.115 GetAxisParameter_DRIVE_FE_LIMIT()

```
long Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_FE_LIMIT (
```

```
            int axis )
```

Getter for DRIVE_FE_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.116 GetAxisParameter_DRIVE_INDEX()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_INDEX (
            int axis )
```

Getter for DRIVE_INDEX axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.117 GetAxisParameter_DRIVE_INPUTS()

```
uint Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_INPUTS (
            int axis )
```

Getter for DRIVE_INPUTS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.118 GetAxisParameter_DRIVE_MODE()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_MODE (
            int axis )
```

Getter for DRIVE_MODE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.119 GetAxisParameter_DRIVE_MONITOR()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_MONITOR (
            int axis )
```

Getter for DRIVE_MONITOR axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.120 GetAxisParameter_DRIVE_NEG_TORQUE()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_NEG_TORQUE (
            int axis )
```

Getter for DRIVE_NEG_TORQUE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.121 GetAxisParameter_DRIVE_PARAMETER()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_PARAMETER (
            int axis )
```

Getter for DRIVE_PARAMETER axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.122 GetAxisParameter_DRIVE_POS_TORQUE()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_POS_TORQUE (
            int axis )
```

Getter for DRIVE_POS_TORQUE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.123 GetAxisParameter_DRIVE_PROFILE()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_PROFILE (
            int axis )
```

Getter for DRIVE_PROFILE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.124 GetAxisParameter_DRIVE_SET_VAL()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_SET_VAL (
            int axis )
```

Getter for DRIVE_SET_VAL axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.125 GetAxisParameter_DRIVE_STATUS()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_STATUS (
            int axis )
```

Getter for DRIVE_STATUS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.126 GetAxisParameter_DRIVE_TORQUE()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_TORQUE (
            int axis )
```

Getter for DRIVE_TORQUE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.127 GetAxisParameter_DRIVE_TYPE()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_TYPE (
```

```
            int axis )
```

Getter for DRIVE_TYPE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.128 GetAxisParameter_DRIVE_VALUE()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_VALUE (
            int axis )
```

Getter for DRIVE_VALUE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.129 GetAxisParameter_DRIVE_VELOCITY()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_DRIVE_VELOCITY (
            int axis )
```

Getter for DRIVE_VELOCITY axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.130 GetAxisParameter_ENCODER()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ENCODER (
            int axis )
```

Getter for ENCODER axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.131 GetAxisParameter_ENCODER_BITS()

```
ushort Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ENCODER_BITS (
            int axis )
```

Getter for ENCODER_BITS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.132 GetAxisParameter_ENCODER_CONTROL()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ENCODER_CONTROL (
            int axis )
```

Getter for ENCODER_CONTROL axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.133 GetAxisParameter_ENCODER_FILTER()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ENCODER_FILTER (
            int axis )
```

Getter for ENCODER_FILTER axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.134 GetAxisParameter_ENCODER_ID()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ENCODER_ID (
            int axis )
```

Getter for ENCODER_ID axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.135 GetAxisParameter_ENCODER_STATUS()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ENCODER_STATUS (
            int axis )
```

Getter for ENCODER_STATUS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.136 GetAxisParameter_ENCODER_TURNS()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ENCODER_TURNS (
            int axis )
```

Getter for ENCODER_TURNS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.137 GetAxisParameter_END_DIR_LAST()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_END_DIR_LAST (
            int axis )
```

Getter for END_DIR_LAST axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.138 GetAxisParameter_ENDMOVE()

```
long Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ENDMOVE (
            int axis )
```

Getter for ENDMOVE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.139 GetAxisParameter_ENDMOVE_BUFFER()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ENDMOVE_BUFFER (
```

```
            int axis )
```

Getter for ENDMOVE_BUFFER axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.140 GetAxisParameter_ENDMOVE_SPEED()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ENDMOVE_SPEED (
            int axis )
```

Getter for ENDMOVE_SPEED axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.141 GetAxisParameter_ERRORMASK()

```
AxisStatus Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ERRORMASK (
            int axis )
```

Getter for ERRORMASK axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.142 GetAxisParameter_FAST_JOG()

```
short Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FAST_JOG (
            int axis )
```

Getter for FAST_JOG axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.143 GetAxisParameter_FASTDEC()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FASTDEC (
            int axis )
```

Getter for FASTDEC axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.144 GetAxisParameter_FE()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FE (
            int axis )
```

Getter for FE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.145 GetAxisParameter_FE_LATCH()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FE_LATCH (
            int axis )
```

Getter for FE_LATCH axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.146 GetAxisParameter_FE_LIMIT()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FE_LIMIT (
            int axis )
```

Getter for FE_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.147 GetAxisParameter_FE_LIMIT_MODE()

```
short Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FE_LIMIT_MODE (
            int axis )
```

Getter for FE_LIMIT_MODE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.148 GetAxisParameter_FE_RANGE()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FE_RANGE (
            int axis )
```

Getter for FE_RANGE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.149 GetAxisParameter_FEED_OVERRIDE()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FEED_OVERRIDE (
            int axis )
```

Getter for FEED_OVERRIDE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.150 GetAxisParameter_FHOLD_IN()

```
short Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FHOLD_IN (
            int axis )
```

Getter for FHOLD_IN axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.151 GetAxisParameter_FHSPEED()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FHSPEED (
```

```
            int axis )
```

Getter for FHSPEED axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.152 GetAxisParameter_FORCE_ACCEL()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FORCE_ACCEL (
            int axis )
```

Getter for FORCE_ACCEL axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.153 GetAxisParameter_FORCE_DECEL()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FORCE_DECEL (
            int axis )
```

Getter for FORCE_DECEL axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.154 GetAxisParameter_FORCE_DWELL()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FORCE_DWELL (
            int axis )
```

Getter for FORCE_DWELL axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.155 GetAxisParameter_FORCE_JERK()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FORCE_JERK (
            int axis )
```

Getter for FORCE_JERK axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.156 GetAxisParameter_FORCE_RAMP()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FORCE_RAMP (
            int axis )
```

Getter for FORCE_RAMP axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.157 GetAxisParameter_FORCE_SPEED()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FORCE_SPEED (
            int axis )
```

Getter for FORCE_SPEED axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.158 GetAxisParameter_FRAME()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FRAME (
            int axis )
```

Getter for FRAME axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.159 GetAxisParameter_FRAME_REP_DIST()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FRAME_REP_DIST (
            int axis )
```

Getter for FRAME_REP_DIST axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.160 GetAxisParameter_FRAME_SCALE()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FRAME_SCALE (
            int axis )
```

Getter for FRAME_SCALE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.161 GetAxisParameter_FS_LIMIT()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FS_LIMIT (
            int axis )
```

Getter for FS_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.162 GetAxisParameter_FULL_SP_RADIUS()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FULL_SP_RADIUS (
            int axis )
```

Getter for FULL_SP_RADIUS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.163 GetAxisParameter_FWD_IN()

```
short Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FWD_IN (
```

```
            int axis )
```

Getter for FWD_IN axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.164 GetAxisParameter_FWD_JOG()

```
short Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FWD_JOG (
            int axis )
```

Getter for FWD_JOG axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.165 GetAxisParameter_FWD_START()

```
short Trio.UnifiedApi.ITrioConnection.GetAxisParameter_FWD_START (
            int axis )
```

Getter for FWD_START axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.166 GetAxisParameter_I_GAIN()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_I_GAIN (
            int axis )
```

Getter for I_GAIN axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.167 GetAxisParameter_IDLE()

```
bool Trio.UnifiedApi.ITrioConnection.GetAxisParameter_IDLE (
            int axis )
```

Getter for IDLE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.168 GetAxisParameter_IN_POS()

```
uint Trio.UnifiedApi.ITrioConnection.GetAxisParameter_IN_POS (
            int axis )
```

Getter for IN_POS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.169 GetAxisParameter_IN_POS_DIST()

```
long Trio.UnifiedApi.ITrioConnection.GetAxisParameter_IN_POS_DIST (
            int axis )
```

Getter for IN_POS_DIST axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.170 GetAxisParameter_IN_POS_SPEED()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_IN_POS_SPEED (
            int axis )
```

Getter for IN_POS_SPEED axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.171 GetAxisParameter_INTERP_FACTOR()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_INTERP_FACTOR (
            int axis )
```

Getter for INTERP_FACTOR axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.172 GetAxisParameter_INVERT_STEP()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_INVERT_STEP (
            int axis )
```

Getter for INVERT_STEP axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.173 GetAxisParameter_JERK()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_JERK (
            int axis )
```

Getter for JERK axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.174 GetAxisParameter_JOGSPEED()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_JOGSPEED (
            int axis )
```

Getter for JOGSPEED axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.175 GetAxisParameter_LINK_AXIS()

```
short Trio.UnifiedApi.ITrioConnection.GetAxisParameter_LINK_AXIS (
```

```
            int axis )
```

Getter for LINK_AXIS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.176 GetAxisParameter_LOADED()

```
uint Trio.UnifiedApi.ITrioConnection.GetAxisParameter_LOADED (
            int axis )
```

Getter for LOADED axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.177 GetAxisParameter_LOOKAHEAD_FACTOR()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_LOOKAHEAD_FACTOR (
            int axis )
```

Getter for LOOKAHEAD_FACTOR axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.178 GetAxisParameter_MARK()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_MARK (
            int axis )
```

Getter for MARK axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.179 GetAxisParameter_MARKB()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_MARKB (
            int axis )
```

Getter for MARKB axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.180 GetAxisParameter_MERGE()

```
bool Trio.UnifiedApi.ITrioConnection.GetAxisParameter_MERGE (
            int axis )
```

Getter for MERGE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.181 GetAxisParameter_MOVE_COUNT()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_MOVE_COUNT (
            int axis )
```

Getter for MOVE_COUNT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.182 GetAxisParameter_MOVE_PA()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_MOVE_PA (
            int axis )
```

Getter for MOVE_PA axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.183 GetAxisParameter_MOVE_PA_CONT()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_MOVE_PA_CONT (
            int axis )
```

Getter for MOVE_PA_CONT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.184 GetAxisParameter_MOVE_PA_IDLE()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_MOVE_PA_IDLE (
            int axis )
```

Getter for MOVE_PA_IDLE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.185 GetAxisParameter_MOVE_PB()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_MOVE_PB (
            int axis )
```

Getter for MOVE_PB axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.186 GetAxisParameter_MOVE_PB_CONT()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_MOVE_PB_CONT (
            int axis )
```

Getter for MOVE_PB_CONT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.187 GetAxisParameter_MOVE_PB_IDLE()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_MOVE_PB_IDLE (
```

```
            int axis )
```

Getter for MOVE_PB_IDLE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.188 GetAxisParameter_MOVELINK_MODIFY()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_MOVELINK_MODIFY (
            int axis )
```

Getter for MOVELINK_MODIFY axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.189 GetAxisParameter_MOVES_BUFFERED()

```
short Trio.UnifiedApi.ITrioConnection.GetAxisParameter_MOVES_BUFFERED (
            int axis )
```

Getter for MOVES_BUFFERED axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.190 GetAxisParameter_MPOS()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_MPOS (
            int axis )
```

Getter for MPOS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.191 GetAxisParameter_MSPEED()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_MSPEED (
            int axis )
```

Getter for MSPEED axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.192 GetAxisParameter_MSPEED_FILTER()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_MSPEED_FILTER (
            int axis )
```

Getter for MSPEED_FILTER axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.193 GetAxisParameter_MSPEEDF()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_MSPEEDF (
            int axis )
```

Getter for MSPEEDF axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.194 GetAxisParameter_MTYPE()

```
MotionType Trio.UnifiedApi.ITrioConnection.GetAxisParameter_MTYPE (
            int axis )
```

Getter for MTYPE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.195 GetAxisParameter_NTYPE()

```
MotionType Trio.UnifiedApi.ITrioConnection.GetAxisParameter_NTYPE (
            int axis )
```

Getter for NTYPE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.196 GetAxisParameter_OFFPOS()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_OFFPOS (
            int axis )
```

Getter for OFFPOS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.197 GetAxisParameter_OPEN_WIN()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_OPEN_WIN (
            int axis )
```

Getter for OPEN_WIN axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.198 GetAxisParameter_OUTLIMIT()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_OUTLIMIT (
            int axis )
```

Getter for OUTLIMIT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.199 GetAxisParameter_OV_GAIN()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_OV_GAIN (
```

```
            int axis )
```

Getter for OV_GAIN axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.200 GetAxisParameter_P_GAIN()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_P_GAIN (
            int axis )
```

Getter for P_GAIN axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.201 GetAxisParameter_POS_DELAY()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_POS_DELAY (
            int axis )
```

Getter for POS_DELAY axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.202 GetAxisParameter_POSI_SEQ_DELAY()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_POSI_SEQ_DELAY (
            int axis )
```

Getter for POSI_SEQ_DELAY axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.203 GetAxisParameter_POSI_SEQ_MODE()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_POSI_SEQ_MODE (
            int axis )
```

Getter for POSI_SEQ_MODE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.204 GetAxisParameter_PP_STEP()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_PP_STEP (
            int axis )
```

Getter for PP_STEP axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.205 GetAxisParameter_PS_ENCODER()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_PS_ENCODER (
            int axis )
```

Getter for PS_ENCODER axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.206 GetAxisParameter_PWM_CYCLE()

```
uint Trio.UnifiedApi.ITrioConnection.GetAxisParameter_PWM_CYCLE (
            int axis )
```

Getter for PWM_CYCLE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.207 GetAxisParameter_PWM_MARK()

```
uint Trio.UnifiedApi.ITrioConnection.GetAxisParameter_PWM_MARK (
            int axis )
```

Getter for PWM_MARK axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.208 GetAxisParameter_RAISE_ANGLE()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_RAISE_ANGLE (
            int axis )
```

Getter for RAISE_ANGLE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.209 GetAxisParameter_REG_INPUTS()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_REG_INPUTS (
            int axis )
```

Getter for REG_INPUTS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.210 GetAxisParameter_REG_POS()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_REG_POS (
            int axis )
```

Getter for REG_POS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.211 GetAxisParameter_REG_POSB()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_REG_POSB (
```

```
            int axis )
```

Getter for REG_POSB axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.212 GetAxisParameter_REGIST_CONTROL()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_REGIST_CONTROL (
            int axis )
```

Getter for REGIST_CONTROL axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.213 GetAxisParameter_REGIST_DELAY()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_REGIST_DELAY (
            int axis )
```

Getter for REGIST_DELAY axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.214 GetAxisParameter_REGIST_SPEED()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_REGIST_SPEED (
            int axis )
```

Getter for REGIST_SPEED axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.215 GetAxisParameter_REGIST_SPEEDB()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_REGIST_SPEEDB (
            int axis )
```

Getter for REGIST_SPEEDB axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.216 GetAxisParameter_REMAIN()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_REMAIN (
            int axis )
```

Getter for REMAIN axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.217 GetAxisParameter_REP_DIST()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_REP_DIST (
            int axis )
```

Getter for REP_DIST axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.218 GetAxisParameter_REP_OPTION()

```
short Trio.UnifiedApi.ITrioConnection.GetAxisParameter_REP_OPTION (
            int axis )
```

Getter for REP_OPTION axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.219 GetAxisParameter_REV_IN()

```
short Trio.UnifiedApi.ITrioConnection.GetAxisParameter_REV_IN (
            int axis )
```

Getter for REV_IN axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.220 GetAxisParameter_REV_JOG()

```
short Trio.UnifiedApi.ITrioConnection.GetAxisParameter_REV_JOG (
            int axis )
```

Getter for REV_JOG axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.221 GetAxisParameter_REV_START()

```
short Trio.UnifiedApi.ITrioConnection.GetAxisParameter_REV_START (
            int axis )
```

Getter for REV_START axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.222 GetAxisParameter_ROBOT_DPOS()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ROBOT_DPOS (
            int axis )
```

Getter for ROBOT_DPOS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.223 GetAxisParameter_ROBOT_FE()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ROBOT_FE (
```

```
            int axis )
```

Getter for ROBOT_FE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.224 GetAxisParameter_ROBOT_FE_LATCH()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ROBOT_FE_LATCH (
            int axis )
```

Getter for ROBOT_FE_LATCH axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.225 GetAxisParameter_ROBOT_FE_RANGE()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ROBOT_FE_RANGE (
            int axis )
```

Getter for ROBOT_FE_RANGE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.226 GetAxisParameter_ROBOT_FS_LIMIT()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ROBOT_FS_LIMIT (
            int axis )
```

Getter for ROBOT_FS_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.227 GetAxisParameter_ROBOT_RS_LIMIT()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ROBOT_RS_LIMIT (
            int axis )
```

Getter for ROBOT_RS_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.228 GetAxisParameter_ROBOT_SP_MODE()

```
short Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ROBOT_SP_MODE (
            int axis )
```

Getter for ROBOT_SP_MODE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.229 GetAxisParameter_ROBOT_UNITS()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ROBOT_UNITS (
            int axis )
```

Getter for ROBOT_UNITS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.230 GetAxisParameter_ROBOTSTATUS()

```
RobotStatus Trio.UnifiedApi.ITrioConnection.GetAxisParameter_ROBOTSTATUS (
            int axis )
```

Getter for ROBOTSTATUS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.231 GetAxisParameter_RS_LIMIT()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_RS_LIMIT (
            int axis )
```

Getter for RS_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.232 GetAxisParameter_S_REF()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_S_REF (
            int axis )
```

Getter for S_REF axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.233 GetAxisParameter_S_REF_OUT()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_S_REF_OUT (
            int axis )
```

Getter for S_REF_OUT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.234 GetAxisParameter_SERVO()

```
bool Trio.UnifiedApi.ITrioConnection.GetAxisParameter_SERVO (
            int axis )
```

Getter for SERVO axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.235 GetAxisParameter_SLOT_NUMBER()

```
short Trio.UnifiedApi.ITrioConnection.GetAxisParameter_SLOT_NUMBER (
```

```
            int axis )
```

Getter for SLOT_NUMBER axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.236 GetAxisParameter_SPEED()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_SPEED (
            int axis )
```

Getter for SPEED axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.237 GetAxisParameter_SRAMP()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_SRAMP (
            int axis )
```

Getter for SRAMP axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.238 GetAxisParameter_START_DIR_LAST()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_START_DIR_LAST (
            int axis )
```

Getter for START_DIR_LAST axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.239 GetAxisParameter_STARTMOVE_SPEED()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_STARTMOVE_SPEED (
            int axis )
```

Getter for STARTMOVE_SPEED axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.240 GetAxisParameter_STOP_ANGLE()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_STOP_ANGLE (
            int axis )
```

Getter for STOP_ANGLE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.241 GetAxisParameter_SYNC_AXIS()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_SYNC_AXIS (
            int axis )
```

Getter for SYNC_AXIS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.242 GetAxisParameter_SYNC_CONTROL()

```
SyncControl Trio.UnifiedApi.ITrioConnection.GetAxisParameter_SYNC_CONTROL (
            int axis )
```

Getter for SYNC_CONTROL axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.243 GetAxisParameter_SYNC_DPOS()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_SYNC_DPOS (
            int axis )
```

Getter for SYNC_DPOS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.244 GetAxisParameter_SYNC_DWELL()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_SYNC_DWELL (
            int axis )
```

Getter for SYNC_DWELL axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.245 GetAxisParameter_SYNC_FLIGHT()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_SYNC_FLIGHT (
            int axis )
```

Getter for SYNC_FLIGHT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.246 GetAxisParameter_SYNC_PAUSE()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_SYNC_PAUSE (
            int axis )
```

Getter for SYNC_PAUSE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.247 GetAxisParameter_SYNC_TIME()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_SYNC_TIME (
```

```
            int axis )
```

Getter for SYNC_TIME axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.248 GetAxisParameter_SYNC_TIMER()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_SYNC_TIMER (
            int axis )
```

Getter for SYNC_TIMER axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.249 GetAxisParameter_SYNC_WITHDRAW()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_SYNC_WITHDRAW (
            int axis )
```

Getter for SYNC_WITHDRAW axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.250 GetAxisParameter_T_REF()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_T_REF (
            int axis )
```

Getter for T_REF axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.251 GetAxisParameter_T_REF_OUT()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_T_REF_OUT (
            int axis )
```

Getter for T_REF_OUT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.252 GetAxisParameter_TABLE_POINTER()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_TABLE_POINTER (
            int axis )
```

Getter for TABLE_POINTER axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.253 GetAxisParameter_TANG_DIRECTION()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_TANG_DIRECTION (
            int axis )
```

Getter for TANG_DIRECTION axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.254 GetAxisParameter_TCP_DPOS()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_TCP_DPOS (
            int axis )
```

Getter for TCP_DPOS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.255 GetAxisParameter_TCP_FS_LIMIT()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_TCP_FS_LIMIT (
            int axis )
```

Getter for TCP_FS_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.256 GetAxisParameter_TCP_RS_LIMIT()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_TCP_RS_LIMIT (
            int axis )
```

Getter for TCP_RS_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.257 GetAxisParameter_TCP_UNITS()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_TCP_UNITS (
            int axis )
```

Getter for TCP_UNITS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.258 GetAxisParameter_UNITS()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_UNITS (
            int axis )
```

Getter for UNITS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.259 GetAxisParameter_VECTOR_BUFFERED()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_VECTOR_BUFFERED (
```

```
            int axis )
```

Getter for VECTOR_BUFFERED axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.260 GetAxisParameter_VERIFY()

```
short Trio.UnifiedApi.ITrioConnection.GetAxisParameter_VERIFY (
            int axis )
```

Getter for VERIFY axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.261 GetAxisParameter_VFF_GAIN()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_VFF_GAIN (
            int axis )
```

Getter for VFF_GAIN axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.262 GetAxisParameter_VP_ACCEL()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_VP_ACCEL (
            int axis )
```

Getter for VP_ACCEL axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.263 GetAxisParameter_VP_JERK()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_VP_JERK (
            int axis )
```

Getter for VP_JERK axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.264 GetAxisParameter_VP_MODE()

```
int Trio.UnifiedApi.ITrioConnection.GetAxisParameter_VP_MODE (
            int axis )
```

Getter for VP_MODE axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.265 GetAxisParameter_VP_POSITION()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_VP_POSITION (
            int axis )
```

Getter for VP_POSITION axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.266 GetAxisParameter_VP_SPEED()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_VP_SPEED (
            int axis )
```

Getter for VP_SPEED axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.267 GetAxisParameter_WORLD_ACCEL()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_WORLD_ACCEL (
            int axis )
```

Getter for WORLD_ACCEL axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.268 GetAxisParameter_WORLD_DECEL()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_WORLD_DECEL (
            int axis )
```

Getter for WORLD_DECEL axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.269 GetAxisParameter_WORLD_DPOS()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_WORLD_DPOS (
            int axis )
```

Getter for WORLD_DPOS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.270 GetAxisParameter_WORLD_FS_LIMIT()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_WORLD_FS_LIMIT (
            int axis )
```

Getter for WORLD_FS_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.271 GetAxisParameter_WORLD_JOGSPEED()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_WORLD_JOGSPEED (
```

```
            int axis )
```

Getter for WORLD_JOGSPEED axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.272 GetAxisParameter_WORLD_RS_LIMIT()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_WORLD_RS_LIMIT (
            int axis )
```

Getter for WORLD_RS_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.273 GetAxisParameter_WORLD_SPEED()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_WORLD_SPEED (
            int axis )
```

Getter for WORLD_SPEED axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.274 GetAxisParameter_WORLD_UNITS()

```
double Trio.UnifiedApi.ITrioConnection.GetAxisParameter_WORLD_UNITS (
            int axis )
```

Getter for WORLD_UNITS axis parameter.

Parameters

axis	Target axis whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.275 GetDIn() [1/2]

```
bool Trio.UnifiedApi.ITrioConnection.GetDIn (
            UInt32 channel )
```

Get the state of the Digital Input.

Parameters

channel	Digital input channel number

Returns

Current state of the digital input channel

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.276 GetDIn() [2/2]

```
UInt32 Trio.UnifiedApi.ITrioConnection.GetDIn (
            UInt32 firstChannel,
            UInt32 lastChannel )
```

Get the state of the Digital Inputs.

Parameters

firstChannel	First digital input channel number
lastChannel	Last digital input channel number. The difference between last and first channel should be at most 32

Returns

Binary sum of digital input states

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.277 GetDIn32()

```
UInt32 Trio.UnifiedApi.ITrioConnection.GetDIn32 ( )
```

Get the state of the first 32 Digital Inputs.

Returns

Bit sum of the states of the first 32 digital input channels

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.278 GetDOut() [1/2]

```
bool Trio.UnifiedApi.ITrioConnection.GetDOut (
            UInt32 channel )
```

Get the state of the Digital Output.

Parameters

channel	Digital output channel number

Returns

Current state of the digital output channel

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.279 GetDOut() [2/2]

```
UInt32 Trio.UnifiedApi.ITrioConnection.GetDOut (
            UInt32 firstChannel,
            UInt32 lastChannel )
```

Get the state of the Digital Outputs.

Parameters

firstChannel	First digital output channel number
lastChannel	Last digital output channel number. The difference between last and first channel should be at most 32

Returns

Binary sum of digital output states

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.280 GetDOut32()

```
UInt32 Trio.UnifiedApi.ITrioConnection.GetDOut32 ( )
```

Get the state of the first 32 Digital Outputs.

Returns

Bit sum of the states of the first 32 digital output channels

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.281 GetFirmwareVersionNumber()

```
double Trio.UnifiedApi.ITrioConnection.GetFirmwareVersionNumber ( )
```

Get the firmware version of the Motion Coordinator.

Returns

On success Motion Coordinator version as a floating point number

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.282 GetFrameGroup()

```
Tuple< UInt32, Int32[]> Trio.UnifiedApi.ITrioConnection.GetFrameGroup (
            UInt32 group )
```

Get frame group info.

Parameters

group	Frame group number to get info

Returns

On success a tuple is returned containing following:

- Current start offset of TABLE range used by the frame group
- Array of axis indices currently belonging to the frame group

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.283 GetInvertIN()

```
bool Trio.UnifiedApi.ITrioConnection.GetInvertIN (
            UInt32 input )
```

Gets the invert status set by INVERT_IN command.

Parameters

input	The input to invert

Returns

True if inverted, false if not inverted

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.284 GetMultiTableValues()

```
double[] Trio.UnifiedApi.ITrioConnection.GetMultiTableValues (
            UInt32 start_index,
            UInt32 count )
```

Get multiple table entries.

Parameters

start_index	Start index of the table range to be read
count	Number of table entries to be read

Returns

Placeholder array for the values to be read

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.285 GetMultiVrValues()

```
double[] Trio.UnifiedApi.ITrioConnection.GetMultiVrValues (
            UInt32 start_index,
            UInt32 count )
```

Get multiple VR entries.

Parameters

start_index	Start index of the VR range to be read
count	Number of VR entries to be read

Returns

Placeholder array for the values to be read

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.286 GetProcessParameter_ERROR_LINE()

```
int Trio.UnifiedApi.ITrioConnection.GetProcessParameter_ERROR_LINE (
            int process )
```

Getter for ERROR_LINE process parameter.

Parameters

process	Target process whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.287 GetProcessParameter_INDEVICE()

```
int Trio.UnifiedApi.ITrioConnection.GetProcessParameter_INDEVICE (
            int process )
```

Getter for INDEVICE process parameter.

Parameters

process	Target process whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.288 GetProcessParameter_OUTDEVICE()

```
int Trio.UnifiedApi.ITrioConnection.GetProcessParameter_OUTDEVICE (
```

```
            int process )
```

Getter for OUTDEVICE process parameter.

Parameters

process	Target process whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.289 GetProcessParameter_PMOVE()

```
bool Trio.UnifiedApi.ITrioConnection.GetProcessParameter_PMOVE (
            int process )
```

Getter for PMOVE process parameter.

Parameters

process	Target process whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.290 GetProcessParameter_PROC_LINE()

```
int Trio.UnifiedApi.ITrioConnection.GetProcessParameter_PROC_LINE (
            int process )
```

Getter for PROC_LINE process parameter.

Parameters

process	Target process whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.291 GetProcessParameter_PROC_STATUS()

```
ProcessStatus Trio.UnifiedApi.ITrioConnection.GetProcessParameter_PROC_STATUS (
            int process )
```

Getter for PROC_STATUS process parameter.

Parameters

process	Target process whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.292 GetProcessParameter_RUN_ERROR()

```
int Trio.UnifiedApi.ITrioConnection.GetProcessParameter_RUN_ERROR (
            int process )
```

Getter for RUN_ERROR process parameter.

Parameters

process	Target process whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.293 GetProcessParameter_TICKS()

```
long Trio.UnifiedApi.ITrioConnection.GetProcessParameter_TICKS (
            int process )
```

Getter for TICKS process parameter.

Parameters

process	Target process whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.294 GetSlotParameter_SERIAL_NUMBER()

```
int Trio.UnifiedApi.ITrioConnection.GetSlotParameter_SERIAL_NUMBER (
            int slot )
```

Getter for SERIAL_NUMBER slot parameter.

Parameters

slot	Target slot whose parameter is retrieved

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.295 GetSystemParameter_ADDRESS()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_ADDRESS ( )
```

Getter for ADDRESS system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.296 GetSystemParameter_AIN0()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_AIN0 ( )
```

Getter for AIN0 system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.297 GetSystemParameter_AIN1()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_AIN1 ( )
```

Getter for AIN1 system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.298 GetSystemParameter_AIN2()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_AIN2 ( )
```

Getter for AIN2 system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.299 GetSystemParameter_AIN3()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_AIN3 ( )
```

Getter for AIN3 system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.300 GetSystemParameter_AINBI0()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_AINBI0 ( )
```

Getter for AINBI0 system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.301 GetSystemParameter_AINBI1()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_AINBI1 ( )
```

Getter for AINBI1 system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.302 GetSystemParameter_AINBI2()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_AINBI2 ( )
```

Getter for AINBI2 system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.303 GetSystemParameter_AINBI3()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_AINBI3 ( )
```

Getter for AINBI3 system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.304 GetSystemParameter_AOUT0()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_AOUT0 ( )
```

Getter for AOUT0 system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.305 GetSystemParameter_AOUT1()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_AOUT1 ( )
```

Getter for AOUT1 system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.306 GetSystemParameter_AOUT2()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_AOUT2 ( )
```

Getter for AOUT2 system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.307 GetSystemParameter_AOUT3()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_AOUT3 ( )
```

Getter for AOUT3 system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.308 GetSystemParameter_AUTO_ETHERCAT()

```
int Trio.UnifiedApi.ITrioConnection.GetSystemParameter_AUTO_ETHERCAT ( )
```

Getter for AUTO_ETHERCAT system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.309 GetSystemParameter_AVERAGE_JITTER()

```
long Trio.UnifiedApi.ITrioConnection.GetSystemParameter_AVERAGE_JITTER ( )
```

Getter for AVERAGE_JITTER system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.310 GetSystemParameter_CONTROL()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_CONTROL ( )
```

Getter for CONTROL system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.311 GetSystemParameter_CPU_EXCEPTIONS()

```
long Trio.UnifiedApi.ITrioConnection.GetSystemParameter_CPU_EXCEPTIONS ( )
```

Getter for CPU_EXCEPTIONS system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.312 GetSystemParameter_CURRENT_JITTER()

```
long Trio.UnifiedApi.ITrioConnection.GetSystemParameter_CURRENT_JITTER ( )
```

Getter for CURRENT_JITTER system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.313 GetSystemParameter_DISPLAY()

```
int Trio.UnifiedApi.ITrioConnection.GetSystemParameter_DISPLAY ( )
```

Getter for DISPLAY system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.314 GetSystemParameter_ERROR_AXIS()

```
int Trio.UnifiedApi.ITrioConnection.GetSystemParameter_ERROR_AXIS ( )
```

Getter for ERROR_AXIS system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.315 GetSystemParameter_FLASH_STATUS()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_FLASH_STATUS ( )
```

Getter for FLASH_STATUS system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.316 GetSystemParameter_FPU_EXCEPTIONS()

```
long Trio.UnifiedApi.ITrioConnection.GetSystemParameter_FPU_EXCEPTIONS ( )
```

Getter for FPU_EXCEPTIONS system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.317 GetSystemParameter_HMI_PROC()

```
int Trio.UnifiedApi.ITrioConnection.GetSystemParameter_HMI_PROC ( )
```

Getter for HMI_PROC system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.318 GetSystemParameter_INPUTS0()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_INPUTS0 ( )
```

Getter for INPUTS0 system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.319 GetSystemParameter_INPUTS1()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_INPUTS1 ( )
```

Getter for INPUTS1 system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.320 GetSystemParameter_IP_ADDRESS()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_IP_ADDRESS ( )
```

Getter for IP_ADDRESS system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.321 GetSystemParameter_IP_GATEWAY()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_IP_GATEWAY ( )
```

Getter for IP_GATEWAY system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.322 GetSystemParameter_IP_MAC()

```
ulong Trio.UnifiedApi.ITrioConnection.GetSystemParameter_IP_MAC ( )
```

Getter for IP_MAC system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.323 GetSystemParameter_IP_MEMORY_CONFIG()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_IP_MEMORY_CONFIG ( )
```

Getter for IP_MEMORY_CONFIG system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.324 GetSystemParameter_IP_NETMASK()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_IP_NETMASK ( )
```

Getter for IP_NETMASK system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.325 GetSystemParameter_IP_PROTOCOL_CONFIG()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_IP_PROTOCOL_CONFIG ( )
```

Getter for IP_PROTOCOL_CONFIG system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.326 GetSystemParameter_IP_PROTOCOL_CTRL()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_IP_PROTOCOL_CTRL ( )
```

Getter for IP_PROTOCOL_CTRL system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.327 GetSystemParameter_IP_TCP_TIMEOUT()

```
ulong Trio.UnifiedApi.ITrioConnection.GetSystemParameter_IP_TCP_TIMEOUT ( )
```

Getter for IP_TCP_TIMEOUT system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.328 GetSystemParameter_IP_TCP_TX_THRESHOLD()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_IP_TCP_TX_THRESHOLD ( )
```

Getter for IP_TCP_TX_THRESHOLD system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.329 GetSystemParameter_IP_TCP_TX_TIMEOUT()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_IP_TCP_TX_TIMEOUT ( )
```

Getter for IP_TCP_TX_TIMEOUT system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.330 GetSystemParameter_LAST_AXIS()

```
int Trio.UnifiedApi.ITrioConnection.GetSystemParameter_LAST_AXIS ( )
```

Getter for LAST_AXIS system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.331 GetSystemParameter_LIMIT_BUFFERED()

```
short Trio.UnifiedApi.ITrioConnection.GetSystemParameter_LIMIT_BUFFERED ( )
```

Getter for LIMIT_BUFFERED system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.332 GetSystemParameter_MAX_JITTER()

```
long Trio.UnifiedApi.ITrioConnection.GetSystemParameter_MAX_JITTER ( )
```

Getter for MAX_JITTER system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.333 GetSystemParameter_MISSED_INTERRUPTS()

```
long Trio.UnifiedApi.ITrioConnection.GetSystemParameter_MISSED_INTERRUPTS ( )
```

Getter for MISSED_INTERRUPTS system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.334 GetSystemParameter_MISSED_RECV_FRAMES()

```
long Trio.UnifiedApi.ITrioConnection.GetSystemParameter_MISSED_RECV_FRAMES ( )
```

Getter for MISSED_RECV_FRAMES system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.335 GetSystemParameter_MISSED_SEND_FRAMES()

```
long Trio.UnifiedApi.ITrioConnection.GetSystemParameter_MISSED_SEND_FRAMES ( )
```

Getter for MISSED_SEND_FRAMES system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.336 GetSystemParameter_MODBUS_TX_DELAY()

```
ushort Trio.UnifiedApi.ITrioConnection.GetSystemParameter_MODBUS_TX_DELAY ( )
```

Getter for MODBUS_TX_DELAY system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.337 GetSystemParameter_MOTION_ERROR()

```
ulong Trio.UnifiedApi.ITrioConnection.GetSystemParameter_MOTION_ERROR ( )
```

Getter for MOTION_ERROR system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.338 GetSystemParameter_MPE_MODE()

```
short Trio.UnifiedApi.ITrioConnection.GetSystemParameter_MPE_MODE ( )
```

Getter for MPE_MODE system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.339 GetSystemParameter_N_ANA_IN()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_N_ANA_IN ( )
```

Getter for N_ANA_IN system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.340 GetSystemParameter_N_ANA_OUT()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_N_ANA_OUT ( )
```

Getter for N_ANA_OUT system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.341 GetSystemParameter_NIN()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_NIN ( )
```

Getter for NIN system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.342 GetSystemParameter_NIO()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_NIO ( )
```

Getter for NIO system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.343 GetSystemParameter_NOP()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_NOP ( )
```

Getter for NOP system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.344 GetSystemParameter_PCMCAT_PROC()

```
int Trio.UnifiedApi.ITrioConnection.GetSystemParameter_PCMCAT_PROC ( )
```

Getter for PCMCAT_PROC system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.345 GetSystemParameter_PLC_CONFIG()

```
int Trio.UnifiedApi.ITrioConnection.GetSystemParameter_PLC_CONFIG ( )
```

Getter for PLC_CONFIG system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.346 GetSystemParameter_PLC_ERROR()

```
long Trio.UnifiedApi.ITrioConnection.GetSystemParameter_PLC_ERROR ( )
```

Getter for PLC_ERROR system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.347 GetSystemParameter_PLC_OVERFLOW()

```
long Trio.UnifiedApi.ITrioConnection.GetSystemParameter_PLC_OVERFLOW ( )
```

Getter for PLC_OVERFLOW system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.348 GetSystemParameter_PLC_RUN()

```
long Trio.UnifiedApi.ITrioConnection.GetSystemParameter_PLC_RUN ( )
```

Getter for PLC_RUN system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.349 GetSystemParameter_PROCNUMBER()

```
short Trio.UnifiedApi.ITrioConnection.GetSystemParameter_PROCNUMBER ( )
```

Getter for PROCNUMBER system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.350 GetSystemParameter_REMOTE_PROC()

```
int Trio.UnifiedApi.ITrioConnection.GetSystemParameter_REMOTE_PROC ( )
```

Getter for REMOTE_PROC system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.351 GetSystemParameter_SCHEDULE_TYPE()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_SCHEDULE_TYPE ( )
```

Getter for SCHEDULE_TYPE system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.352 GetSystemParameter_SCOPE_CYCLE_COUNT()

```
int Trio.UnifiedApi.ITrioConnection.GetSystemParameter_SCOPE_CYCLE_COUNT ( )
```

Getter for SCOPE_CYCLE_COUNT system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.353 GetSystemParameter_SCOPE_DELAY()

```
long Trio.UnifiedApi.ITrioConnection.GetSystemParameter_SCOPE_DELAY ( )
```

Getter for SCOPE_DELAY system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.354 GetSystemParameter_SCOPE_POS()

```
int Trio.UnifiedApi.ITrioConnection.GetSystemParameter_SCOPE_POS ( )
```

Getter for SCOPE_POS system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.355 GetSystemParameter_SCOPE_TRIGGER_POS()

```
int Trio.UnifiedApi.ITrioConnection.GetSystemParameter_SCOPE_TRIGGER_POS ( )
```

Getter for SCOPE_TRIGGER_POS system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.356 GetSystemParameter_SERIAL_NUMBER()

```
int Trio.UnifiedApi.ITrioConnection.GetSystemParameter_SERIAL_NUMBER ( )
```

Getter for SERIAL_NUMBER system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.357 GetSystemParameter_SERVO_OFFSET()

```
int Trio.UnifiedApi.ITrioConnection.GetSystemParameter_SERVO_OFFSET ( )
```

Getter for SERVO_OFFSET system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.358 GetSystemParameter_SERVO_PERIOD()

```
int Trio.UnifiedApi.ITrioConnection.GetSystemParameter_SERVO_PERIOD ( )
```

Getter for SERVO_PERIOD system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.359 GetSystemParameter_SLOT0_TIME()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_SLOT0_TIME ( )
```

Getter for SLOT0_TIME system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.360 GetSystemParameter_SLOT1_TIME()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_SLOT1_TIME ( )
```

Getter for SLOT1_TIME system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.361 GetSystemParameter_SLOT2_TIME()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_SLOT2_TIME ( )
```

Getter for SLOT2_TIME system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.362 GetSystemParameter_SLOT3_TIME()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_SLOT3_TIME ( )
```

Getter for SLOT3_TIME system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.363 GetSystemParameter_SYSTEM_ERROR()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_SYSTEM_ERROR ( )
```

Getter for SYSTEM_ERROR system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.364 GetSystemParameter_SYSTEM_ERROR_MASK()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_SYSTEM_ERROR_MASK ( )
```

Getter for SYSTEM_ERROR_MASK system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.365 GetSystemParameter_SYSTEM_LOAD()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_SYSTEM_LOAD ( )
```

Getter for SYSTEM_LOAD system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.366 GetSystemParameter_SYSTEM_LOAD_MAX()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_SYSTEM_LOAD_MAX ( )
```

Getter for SYSTEM_LOAD_MAX system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.367 GetSystemParameter_TEXT_FILE_LOADER_PROC()

```
int Trio.UnifiedApi.ITrioConnection.GetSystemParameter_TEXT_FILE_LOADER_PROC ( )
```

Getter for TEXT_FILE_LOADER_PROC system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.368 GetSystemParameter_TSIZE()

```
int Trio.UnifiedApi.ITrioConnection.GetSystemParameter_TSIZE ( )
```

Getter for TSIZE system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.369 GetSystemParameter_UNIT_ERROR()

```
uint Trio.UnifiedApi.ITrioConnection.GetSystemParameter_UNIT_ERROR ( )
```

Getter for UNIT_ERROR system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.370 GetSystemParameter_VERSION()

```
double Trio.UnifiedApi.ITrioConnection.GetSystemParameter_VERSION ( )
```

Getter for VERSION system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.371 GetSystemParameter_VRSIZE()

```
int Trio.UnifiedApi.ITrioConnection.GetSystemParameter_VRSIZE ( )
```

Getter for VRSIZE system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.372 GetSystemParameter_WDOG()

```
bool Trio.UnifiedApi.ITrioConnection.GetSystemParameter_WDOG ( )
```

Getter for WDOG system parameter.

Returns

Current value of the parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.373 GetTableValue()

```
double Trio.UnifiedApi.ITrioConnection.GetTableValue (
            UInt32 index )
```

Get single table value.

Parameters

index	Index of the table entry to be read

Returns

On success the value of table entry returned

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.374 GetTcpHostName()

```
String Trio.UnifiedApi.ITrioConnection.GetTcpHostName ( )
```

Get current host name or IP of connection object. Valid only for TCP connection.

Returns

Current value of the host name

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.375 GetTcpPort()

```
UInt16 Trio.UnifiedApi.ITrioConnection.GetTcpPort ( )
```

Get current TCP port from connection object. Valid only for TCP connection.

Returns

Current value of the TCP port

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.376 GetTcpTimeout()

```
UInt32 Trio.UnifiedApi.ITrioConnection.GetTcpTimeout ( )
```

Get current TCP connection timeout. Valid only for TCP connection.

Returns

Current value of the TCP connection timeout in milliseconds

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.377 GetVrString() [1/3]

```
String Trio.UnifiedApi.ITrioConnection.GetVrString (
            UInt32 vr )
```

Get a null-terminated string from consecutive VRs.

Parameters

vr	Starting VR index

Returns

On success the string stored in consecutive VR entries

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.378 GetVrString() [2/3]

```
unsafe void Trio.UnifiedApi.ITrioConnection.GetVrString (
            UInt32 vr,
            Byte * buff,
            ref UInt32 buff_size )
```

Get a null-terminated string from consecutive VRs.

Parameters

vr	Starting VR index
buff	Buffer to receive the string characters. If NULL function succeeds and returns expected buffers size in buff_size parameter including null terminator character
buff_size	Size of the buffer on input and on exit required buffer size including null terminator

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.379 GetVrString() [3/3]

```
void Trio.UnifiedApi.ITrioConnection.GetVrString (
            UInt32 vr,
            StreamHandlerCallback handler )
```

Get a null-terminated string from consecutive VRs using callback function to re-construct the string from received chunks.

Parameters

vr	Starting VR index
handler	Callback function to receive string chunks

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.380 GetVrStringLen()

```
UInt32 Trio.UnifiedApi.ITrioConnection.GetVrStringLen (
            UInt32 vr )
```

Get the length of a null-terminated string from consecutive VRs.

Parameters

vr	Starting VR index

Returns

Length of the string

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.381 GetVrValue()

```
double Trio.UnifiedApi.ITrioConnection.GetVrValue (
            UInt32 index )
```

Get single VR value.

Parameters

index	Index of the VR entry to be read

Returns

On success the value VR entry returned

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.382 MoveAbs()

```
void Trio.UnifiedApi.ITrioConnection.MoveAbs (
            double pos,
            Int32 axis = -1 )
```

Absolute positional move of single axis.

Parameters

pos	Absolute position to move axis to in user units
axis	Index of the axis for which to be subject of move operation or -1 to use default base axis

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.383 MoveAbsMulti()

```
void Trio.UnifiedApi.ITrioConnection.MoveAbsMulti (
            double[] pos,
            Int32 axis = -1 )
```

Absolute positional move of multiple axes.

Parameters

pos	Array of absolute positions in user units to move axes defined by SetMultiAxisBase
axis	Base axis for this command. If value is -1 then the default base axis is used.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.384 MoveAbsSeq()

```
void Trio.UnifiedApi.ITrioConnection.MoveAbsSeq (
            Int32 axis,
            UInt32 tablePointer,
            UInt32 nAxes,
            UInt32 nPoints,
            UInt32 options,
            double radius,
            UInt32 tableOutput = 0,
            double transitionAngle = 0 )
```

Function allows a sequence of absolute movements to be loaded via TABLE values. The moves can be automatically merged together using a circular or spherical arc. Movement on 2 to 6 axes may be controlled in this way.

Parameters

axis	Base axis for this command. If value is -1 then the default base axis is used
tablePointer	Start of the move sequence in the TABLE data.
nAxes	Number of axes 2 to 6
nPoints	The number of points, each point requires 2 or 3 table values
options	Modifiers for the movements. See the TrioBASIC MOVEABSSEQ documentation for more details.
radius	The merging/filleting radius to be applied. 0 for no filleting
tableOutput	Output data to TABLE. 0 means do not output data to TABLE, otherwise is the TABLE start index for the data.
transitionAngle	Transition angle (in Radians) for any filleted arcs that are generated. Set to 0 if no transition curve is to be applied.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.385 MoveAbsSp()

```
void Trio.UnifiedApi.ITrioConnection.MoveAbsSp (
            double pos,
            Int32 axis = -1 )
```

Absolute positional move of single axis with speed changed between moves.

Parameters

pos	Absolute position to move axis to in user units
axis	Index of the axis for which to be subject of move operation or -1 to use default base axis

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.386 MoveAbsSpMulti()

```
void Trio.UnifiedApi.ITrioConnection.MoveAbsSpMulti (
            double[] pos,
            Int32 axis = -1 )
```

Absolute positional move of multiple axes with speed changed between moves.

Parameters

pos	Array of absolute positions in user units to move axes defined by SetMultiAxisBase
axis	Index of the first axis for which to be subject of move operation or -1 to use default base axis

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.387 MoveCirc()

```
void Trio.UnifiedApi.ITrioConnection.MoveCirc (
            Int32 axis,
            double finishBase,
            double finishNext,
            double centreBase,
            double centreNext,
            UInt32 direction,
            double transitionAngle = 0,
            UInt32 tableOutput = unchecked((UInt32)(-1)) )
```

Moves 2 orthogonal axes in such a way as to produce a circular arc at the tool point. The length and radius of the arc are defined by the five parameters in the command line. The move parameters are always relative to the end

of the last specified move. This is the start position on the circle circumference. Axis 1 is the current BASE axis. Axis 2 is the next axis in the BASE array. The first 4 distance parameters are scaled according to the current unit conversion factor for the BASE axis.

Parameters

axis	Base axis for this command. If value is -1 then the default base axis is used.
finishBase	Position on BASE axis to finish at.
finishNext	Position on next axis in BASE array to finish at.
centreBase	Position on BASE about which to move.
centreNext	Position on next axis in BASE array about which to move.
direction	0 Arc is interpolated in an anti - clockwise direction 1 Arc is interpolated in a clockwise direction 2 Arc is interpolated using the shortest path to endpoint 3 Arc is interpolated using the longest path to endpoint
transitionAngle	Transition angle (in Radians) for any filleted arcs that are generated. Set to 0 if no transition curve is to be applied.
tableOutput	Output data to TABLE. 0 means do not output data to TABLE, otherwise is the TABLE start index for the data.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.388 MoveCircSp()

```
void Trio.UnifiedApi.ITrioConnection.MoveCircSp (
            Int32 axis,
            double finishBase,
            double finishNext,
            double centreBase,
            double centreNext,
            UInt32 direction,
            double transitionAngle = 0,
            UInt32 tableOutput = 0 )
```

Works as MoveCirc and additionally allows vector speed to be changed when using multiple moves in the look ahead buffer when MERGE = ON , using additional parameters FORCE_SPEED and ENDMOVE_SPEED. Optional parameters ta and output allow the use of transition curves.

Parameters

axis	Base axis for this command. If value is -1 then the default base axis is used.
finishBase	Position on BASE axis to finish at.
finishNext	Position on next axis in BASE array to finish at.
centreBase	Position on BASE about which to move.
centreNext	Position on next axis in BASE array about which to move.
direction	0 Arc is interpolated in an anti - clockwise direction 1 Arc is interpolated in a clockwise direction 2 Arc is interpolated using the shortest path to endpoint 3 Arc is interpolated using the longest path to endpoint
transitionAngle	Transition angle (in Radians) for any filleted arcs that are generated. Set to 0 if no transition curve is to be applied.

Parameters

tableOutput	Output data to TABLE. 0 means do not output data to TABLE, otherwise is the TABLE start index for the data.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.389 MoveHelical()

```
void Trio.UnifiedApi.ITrioConnection.MoveHelical (
            Int32 axis,
            double finishBase,
            double finishNext,
            double centreBase,
            double centreNext,
            UInt32 direction,
            double linearDistance,
            UInt32 mode,
            UInt32 transitionAngle = 0,
            UInt32 tableOutput = 0 )
```

Performs a helical move. Moves 2 orthogonal axes in such a way as to produce a circular arc at the tool point with a simultaneous linear move on a third axis. The first 5 parameters are similar to those of an MOVECIRC command.The sixth parameter defines the simultaneous linear move.

Parameters

axis	Base axis for this command. If value is -1 then the default base axis is used.
finishBase	Position on the base axis to finish at.
finishNext	Position on the next axis in the base vector to finish at.
centreBase	Position on the base axis about which to move.
centreNext	Position on the next axis in the base vector about which to move.
direction	Direction in which to move around the arc.
linearDistance	Distance to move on the third axis in the base vector.
mode	Mode of operation. See the MHELICAL command documentation for more details.
transitionAngle	Transition angle (in Radians) for any filleted arcs that are generated. Set to 0 if no transition curve is to be applied.
tableOutput	Output data to TABLE. 0 means do not output data to TABLE, otherwise is the TABLE start index for the data.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.390 MoveHelicalSp()

```
void Trio.UnifiedApi.ITrioConnection.MoveHelicalSp (
            Int32 axis,
```

```
            double finishBase,
            double finishNext,
            double centreBase,
            double centreNext,
            UInt32 direction,
            double linearDistance,
            UInt32 mode,
            UInt32 transitionAngle = 0,
            UInt32 tableOutput = 0 )
```

Performs a helical move the same as MHELICAL and additionally allows vector speed to be changed when using multiple moves in the buffer. Uses additional axis parameters FORCE_SPEED, ENDMOVE_SPEED and STARTMOVE_SPEED.

Parameters

axis	Base axis for this command. If value is -1 then the default base axis is used.
finishBase	Position on the base axis to finish at.
finishNext	Position on the next axis in the base vector to finish at.
centreBase	Position on the base axis about which to move.
centreNext	Position on the next axis in the base vector about which to move.
direction	Direction in which to move around the arc.
linearDistance	Distance to move on the third axis in the base vector.
mode	Mode of operation. See the MHELICAL command documentation for more details.
transitionAngle	Transition angle (in Radians) for any filleted arcs that are generated. Set to 0 if no transition curve is to be applied.
tableOutput	Output data to TABLE. 0 means do not output data to TABLE, otherwise is the TABLE start index for the data.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.391 MoveLink()

```
void Trio.UnifiedApi.ITrioConnection.MoveLink (
            Int32 axis,
            double distance,
            double link_dist,
            double link_acc,
            double link_dec,
            int link_axis = -1,
            UInt32 link_options = 0,
            double link_pos = 0,
            double base_dist = 0,
            double y_dist = 0,
            double z_dist = 0,
            double u_dist = 0,
            double v_dist = 0,
            double w_dist = 0 )
```

The linked move command is designed for controlling movements such as:

- Synchronisation to conveyors
- Flying shears
- Thread chasing, tapping etc.
- Coil winding The motion consists of a linear movement with separately variable acceleration and deceleration phases linked via a software gearbox to the MEASURED position(MPOS) of another axis.The command uses the BASE and AXIS, and unit conversion factors in a similar way to other move commands. The “link” axis may move in either direction to drive the output motion.The link distances specified are always positive.

Parameters

axis	Base axis for this command. If value is -1 then the default base axis is used
distance	Incremental distance in user units to be moved on the current base axis, as a result of the measured movement on the “input” axis which drives the move.
link_dist	Positive incremental distance in user units which is required to be measured on the “link” axis to result in the motion on the base axis.
link_acc	Positive incremental distance in user units on the input axis over which the base axis accelerates.
link_dec	Positive incremental distance in user units on the input axis over which the base axis decelerates.
link_axis	Specifies the axis to “link” to. It should be set to a value between 0 and the number of available axes.
link_options	Bit value options to customise how your MOVELINK operates: Bit 0 (1) - link commences exactly when registration event MARK occurs on link axis Bit 1 (2) - link commences at an absolute position on link axis(see link_pos for start position) Bit 2 (4) - MOVELINK repeats automatically and bi - directionally when this bit is set. (This mode can be cleared by setting bit 1 of the REP_OPTION axis parameter) Bit 4 (16)- If this bit is set the MOVELINK acceleration and deceleration phases are constructed using an “S” speed profile not a trapezoidal speed profile Bit 5 (32)- Link is only active during a positive move on the link axis Bit 8 (256) - link commences exactly when registration event MARKB occurs on link axis Bit 9 (512) - link commences exactly when registration event R_MARK occurs on link axis. (see link_pos for channel number) Bit 10 and 11 - S-Ramp mode selection for use when bit 4 is set.Figure in() shows peak acceleration compared to linear ramp. 0 - SIN speed Profile(1.55) 1 - Power 9 polynomial speed(2.42) 2 - Power 7 polynomial speed(2.16) 3 - Power 5 polynomial speed(1.86) Bit 13 - Slave axis follows master axis DPOS Bit 14 - Link is active during positive movement from a threshold.The threshold is updated to the highest positive value.
link_pos	Link_option bit 1 - the absolute position on the link axis in user UNITS where the MOVELINK is to be start. link_option bit 9 – the registration channel to start the movement on.
base_dist	Firmware V2.0253 onwards. The “base distance” is the distance that is to be travelled at the base speed and is part of the total first “distance” parameter. See Examples 4 and 5 below. MOVELINK_MODIFY cannot be used when the base distance is in use.
y_dist	Same as base_dist, for 2nd base axis
z_dist	Same as base_dist, for 3rd base axis
u_dist	Same as base_dist, for 4th base axis
v_dist	Same as base_dist, for 5th base axis
w_dist	Same as base_dist, for 6th base axis

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.392 MoveModify()

```
void Trio.UnifiedApi.ITrioConnection.MoveModify (
            double position,
            Int32 axis = -1 )
```

MOVEMODIFY will change the absolute end position of a single axis MOVE , MOVEABS , MOVESP , MOVEABSSP or MOVEMODIFY that is in the last position in the movement buffer. If there is no motion command in the movement buffers or the last movement is not a single axis linear move, then MOVEMODIFY is loaded.

Parameters

position	Absolute position for the current move to complete at
axis	Base axis for this command. If value is -1 then the default base axis is used

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.393 MovePick()

```
void Trio.UnifiedApi.ITrioConnection.MovePick (
            Int32 axis,
            UInt32 mode,
            double x,
            double y,
            double z,
            double withdraw,
            double overlapControlA = 0.5,
            double overlapControlB = 0.5,
            double u = 0,
            double v = 0,
            double w = 0,
            double withdrawControl = 1,
            double xyControl = 1,
            double approachControl = 1 )
```

MOVEPICK is a new move type aimed at making it much simpler to program fast pick and place operations.

Parameters

axis	Base axis for this command. If value is -1 then the default base axis is used
mode	Specify the number of axes 0 = xyz, 1 = xyzu, 2 = xyzuv, 3 = xyzuvw
x	Target x absolute position
y	Target y absolute position
z	Target z absolute position

Parameters

withdraw	Incremental withdraw distance from the start position on z axis
overlapControlA	0..0.9 Specify rounding on corner A (optional)
overlapControlB	0..0.9 Specify rounding on corner B (optional)
u	Target u absolute position (optional)
v	Target v absolute position (optional)
w	Target w absolute position (optional)
withdrawControl	0..1 Adjust the speed for withdraw section (optional)
xyControl	0..1 Adjust the speed for xy section (optional)
approachControl	0..1 Adjust the speed for approach section (optional)

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.394 MoveRel()

```
void Trio.UnifiedApi.ITrioConnection.MoveRel (
            double pos,
            Int32 axis = -1 )
```

Incremental move of single axis.

Parameters

pos	Distance to move axis in user units
axis	Index of the axis for which to be subject of move operation or -1 to use default base axis

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.395 MoveRelMulti()

```
void Trio.UnifiedApi.ITrioConnection.MoveRelMulti (
            double[] dist,
            Int32 axis = -1 )
```

Incremental move of multiple axes.

Parameters

dist	Array of distances in user units to move axes defined by SetMultiAxisBase
axis	Index of the first axis for which to be subject of move operation or -1 to use default base axis

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.396 MoveRelSeq()

```
void Trio.UnifiedApi.ITrioConnection.MoveRelSeq (
            Int32 axis,
            UInt32 tablePointer,
            UInt32 nAxes,
            UInt32 nPoints,
            UInt32 options,
            double radius,
            UInt32 tableOutput = 0,
            double transitionAngle = 0 )
```

Function allows a sequence of relative movements to be loaded via TABLE values. The moves can be automatically merged together using a circular or spherical arc. Movement on 2 to 6 axes may be controlled in this way.

Parameters

axis	Base axis for this command. If value is -1 then the default base axis is used
tablePointer	Start of the move sequence in the TABLE data.
nAxes	Number of axes 2 to 6
nPoints	The number of points, each point requires 2 or 3 table values
options	Modifiers for the movements. See the TrioBASIC MOVEABSSEQ documentation for more details.
radius	The merging/filleting radius to be applied. 0 for no filleting
tableOutput	Output data to TABLE. 0 means do not output data to TABLE, otherwise is the TABLE start index for the data.
transitionAngle	Transition angle (in Radians) for any filleted arcs that are generated. Set to 0 if no transition curve is to be applied.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.397 MoveRelSp()

```
void Trio.UnifiedApi.ITrioConnection.MoveRelSp (
            double pos,
            Int32 axis = -1 )
```

Incremental move of single axis with speed changed between moves.

Parameters

pos	Distance to move axis in user units
axis	Index of the axis for which to be subject of move operation or -1 to use default base axis

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.398 MoveRelSpMulti()

```
void Trio.UnifiedApi.ITrioConnection.MoveRelSpMulti (
            double[] dist,
            Int32 axis = -1 )
```

Incremental move of multiple axes with speed changed between moves.

Parameters

dist	Array of distances in user units to move axes defined by SetMultiAxisBase
axis	Index of the first axis for which to be subject of move operation or -1 to use default base axis

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.399 MoveSpherical()

```
void Trio.UnifiedApi.ITrioConnection.MoveSpherical (
            Int32 axis,
            double endMidX,
            double endMidY,
            double endMidZ,
            double centreX,
            double centreY,
            double centreZ,
            UInt32 mode,
            UInt32 gtpi = 0,
            double rotau = 0,
            double rotav = 0,
            double rotaw = 0,
            double rotap = 0 )
```

Moves the three axis group defined in BASE along a spherical path with a vector speed determined by the SPEED set in the first axis of the BASE array. There are 4 modes of operation with the option of finishing the move at an endpoint different to the start, or returning to the start point to complete a circle. The path of the movement in 3D space can be defined either by specifying a point somewhere along the path, or by specifying the centre of the sphere.

Parameters

axis	Base axis for this command. If value is -1 then the default base axis is used.
endMidX	End position of the first axis
endMidY	End position of the second axis
endMidZ	End position of the third axis

Parameters

centreX	Centre position of the first axis
centreY	Centre position of the second axis
centreZ	Centre position of the third axis
mode	Mode of operation. See the MSPHERICAL command documentation for more details.
gtpi	If this optional parameter is non zero, modes 0 and 1 will perform a move taking the opposite way around a 360 degree circle to the same endpoint
rotau	If this optional parameter is non zero, a fourth axis will perform linear interpolation at the same time as the spherical move. The axis is the next in the BASE sequence
rotav	If this optional parameter is non zero, a fifth axis will perform linear interpolation at the same time as the spherical move
rotaw	If this optional parameter is non zero, a sixth axis will perform linear interpolation at the same time as the spherical move
rotap	If this optional parameter is non zero, a seventh axis will perform linear interpolation at the same time as the spherical move

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.400 MoveSphericalSp()

```
void Trio.UnifiedApi.ITrioConnection.MoveSphericalSp (
            Int32 axis,
            double endMidX,
            double endMidY,
            double endMidZ,
            double centreX,
            double centreY,
            double centreZ,
            UInt32 mode,
            UInt32 gtpi = 0,
            double rotau = 0,
            double rotav = 0,
            double rotaw = 0,
            double rotap = 0 )
```

Performs a spherical move the same as MSPHERICAL and additionally allows vector speed to be changed when using multiple moves in the look ahead buffer when MERGE = ON, using additional parameters FORCE_SPEED, ENDMOVE_SPEED and STARTMOVE_SPEED.

Parameters

axis	Base axis for this command. If value is -1 then the default base axis is used.
endMidX	End position of the first axis
endMidY	End position of the second axis
endMidZ	End position of the third axis
centreX	Centre position of the first axis
centreY	Centre position of the second axis
centreZ	Centre position of the third axis

Parameters

mode	Mode of operation. See the MSPHERICAL command documentation for more details.
gtpi	If this optional parameter is non zero, modes 0 and 1 will perform a move taking the opposite way around a 360 degree circle to the same endpoint
rotau	If this optional parameter is non zero, a fourth axis will perform linear interpolation at the same time as the spherical move. The axis is the next in the BASE sequence
rotav	If this optional parameter is non zero, a fifth axis will perform linear interpolation at the same time as the spherical move
rotaw	If this optional parameter is non zero, a sixth axis will perform linear interpolation at the same time as the spherical move
rotap	If this optional parameter is non zero, a seventh axis will perform linear interpolation at the same time as the spherical move

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.401 New()

```
void Trio.UnifiedApi.ITrioConnection.New (
            String target )
```

Deletes a file, all files or the TABLE data the Motion Coordinator.

Parameters

target	The target to be cleared (deleted) "file_name" - deletes the file with given name from the Motion Coordinator "ALL" - deletes all user files from the Motion Coordinator. System files are not deleted. "TABLE" - sets all TABLE values to 0 .

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.402 OpenConnection()

```
void Trio.UnifiedApi.ITrioConnection.OpenConnection ( )
```

Open a connection to the Motion Coordinator.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.403 PSwitch()

```
void Trio.UnifiedApi.ITrioConnection.PSwitch (
            int switchNumber,
            PSwitchMode mode,
            Int32 axis,
            UInt32 outputNumber,
            bool outputState,
            double setPosition,
            double resetPosition )
```

The PSWITCH command allows an output to be set when a predefined position is reached, and to be reset when a second position is reached.

Parameters

switchNumber	The switch number in the range 0..63
mode	Mode of operation. See PSwitchMode enumeration for more information
axis	Axis to link the PSWITCH to, may be any real or virtual axis
outputNumber	Selects the output to set, can be any real or virtual output
outputState	False - turn the output OFF at setpos, or true - turn the output ON at setpos
setPosition	The position at which output is set or reset, in user units
resetPosition	The position at which output is set or reset, in user units

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.404 PSwitchMulti()

```
void Trio.UnifiedApi.ITrioConnection.PSwitchMulti (
            int switchNumber,
            PSwitchMultiMode mode,
            Int32 axis,
            UInt32 outputNumber,
            bool outputState,
            UInt32 tableStart,
            UInt32 tableEnd,
            UInt32 pulseTime = 0 )
```

The PSWITCHMULTI command allows an output to be set that turns the output on and off at selected positions loaded from the TABLE memory.

Parameters

switchNumber	The switch number in the range 0..63
mode	Mode of operation. See PSwitchMultiMode enumeration for more information
axis	Axis to link the PSWITCH to, may be any real or virtual axis
outputNumber	Selects the output to set, can be any real or virtual output
outputState	False - turn the output OFF at setpos, or true - turn the output ON at setpos
tableStart	TABLE index for start of PSWITCH data
tableEnd	TABLE index for end of PSWITCH data
pulseTime	Time in servo-cycles when pulse output mode bit 4 is set

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.405 PSwitchOff()

```
void Trio.UnifiedApi.ITrioConnection.PSwitchOff (
            int switchNumber,
            bool hold = false )
```

The PSWITCHOFF command is used to disable.

Parameters

switchNumber	The switch number in the range 0..63
hold	false = The PSWITCH output will hold in the same state it was when the PSWITCH is set to OFF . (Default) true = The PSWITCH output is forced OFF even if it was ON when the PSWTICH is set to OFF .

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.406 RapidStop()

```
void Trio.UnifiedApi.ITrioConnection.RapidStop (
            Int32 mode = 0 )
```

The RAPIDSTOP command operates like the CANCEL command but does so for ALL axes using a subset of cancel modes.

Parameters

mode	0 Cancels axis commands from the MTYPE buffers(default) 1 Cancels all buffered moves on all axis(excluding the PMOVE) 2 Cancels all active and buffered moves including the PMOVE

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.407 Reverse()

```
void Trio.UnifiedApi.ITrioConnection.Reverse (
            Int32 axis = -1 )
```

Sets continuous reverse movement.

Parameters

axis	Axis index to set reverse movement to or -1 to apply to default base axis.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.408 Select()

```
void Trio.UnifiedApi.ITrioConnection.Select (
            String remote_file,
            FileType file_type )
```

Makes the named program the currently selected program on the Motion Coordinator, if the named program does not exist then it makes a program of that name. The selected program cannot be changed when programs are running. When a program is selected any previously selected program is compiled. When program_type is specified is to make a program of that name and type, it cannot be possible to change the type of an existing program.

Parameters

remote_file	The file (program) name
file_type	The type of the file (program) to be created

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.409 SetAOut()

```
void Trio.UnifiedApi.ITrioConnection.SetAOut (
            UInt32 channel,
            Int32 value )
```

Set the state of the Analogue Output.

Parameters

channel	Analogue output channel number
value	New value for the analogue output channel

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.410 SetAxisBase()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisBase (
            Int32 axis )
```

The BASE command is used to direct all subsequent motion commands and axis parameter read/writes to a particular axis, or group of axes. The default setting is a sequence: 0, 1, 2, 3...

Parameters

axis	The axes to send the motion commands to

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.411 SetAxisParameter_ACCEL()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_ACCEL (
            int axis,
            double value )
```

Setter for ACCEL axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.412 SetAxisParameter_AFF_GAIN()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_AFF_GAIN (
            int axis,
            double value )
```

Setter for AFF_GAIN axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.413 SetAxisParameter_ATYPE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_ATYPE (
            int axis,
            AxisType value )
```

Setter for ATYPE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.414 SetAxisParameter_AXIS_ACCEL()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_AXIS_ACCEL (
            int axis,
            double value )
```

Setter for AXIS_ACCEL axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.415 SetAxisParameter_AXIS_ADDRESS()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_AXIS_ADDRESS (
            int axis,
            int value )
```

Setter for AXIS_ADDRESS axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.416 SetAxisParameter_AXIS_DECEL()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_AXIS_DECEL (
            int axis,
            double value )
```

Setter for AXIS_DECEL axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.417 SetAxisParameter_AXIS_DISPLAY()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_AXIS_DISPLAY (
            int axis,
            short value )
```

Setter for AXIS_DISPLAY axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.418 SetAxisParameter_AXIS_ENABLE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_AXIS_ENABLE (
            int axis,
            bool value )
```

Setter for AXIS_ENABLE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.419 SetAxisParameter_AXIS_ENABLE_OVERRIDE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_AXIS_ENABLE_OVERRIDE (
            int axis,
            bool value )
```

Setter for AXIS_ENABLE_OVERRIDE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.420 SetAxisParameter_AXIS_ERROR_COUNT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_AXIS_ERROR_COUNT (
            int axis,
            int value )
```

Setter for AXIS_ERROR_COUNT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.421 SetAxisParameter_AXIS_FS_LIMIT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_AXIS_FS_LIMIT (
            int axis,
            double value )
```

Setter for AXIS_FS_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.422 SetAxisParameter_AXIS_MAX_JOG_SPEED()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_AXIS_MAX_JOG_SPEED (
            int axis,
            double value )
```

Setter for AXIS_MAX_JOG_SPEED axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.423 SetAxisParameter_AXIS_MODE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_AXIS_MODE (
            int axis,
            AxisMode value )
```

Setter for AXIS_MODE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.424 SetAxisParameter_AXIS_RS_LIMIT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_AXIS_RS_LIMIT (
            int axis,
            double value )
```

Setter for AXIS_RS_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.425 SetAxisParameter_AXIS_SPEED()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_AXIS_SPEED (
            int axis,
            double value )
```

Setter for AXIS_SPEED axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.426 SetAxisParameter_AXIS_UNITS()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_AXIS_UNITS (
            int axis,
            double value )
```

Setter for AXIS_UNITS axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.427 SetAxisParameter_AXIS_Z_OUTPUT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_AXIS_Z_OUTPUT (
            int axis,
            int value )
```

Setter for AXIS_Z_OUTPUT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.428 SetAxisParameter_BACKLASH_DIST()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_BACKLASH_DIST (
            int axis,
            long value )
```

Setter for BACKLASH_DIST axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.429 SetAxisParameter_CHANGE_DIR_LAST()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_CHANGE_DIR_LAST (
            int axis,
            double value )
```

Setter for CHANGE_DIR_LAST axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.430 SetAxisParameter_CLOSE_WIN()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_CLOSE_WIN (
            int axis,
            double value )
```

Setter for CLOSE_WIN axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.431 SetAxisParameter_CLUTCH_RATE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_CLUTCH_RATE (
            int axis,
            double value )
```

Setter for CLUTCH_RATE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.432 SetAxisParameter_CORNER_MODE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_CORNER_MODE (
            int axis,
            CornerMode value )
```

Setter for CORNER_MODE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.433 SetAxisParameter_CORNER_STATE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_CORNER_STATE (
            int axis,
            CornerState value )
```

Setter for CORNER_STATE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.434 SetAxisParameter_CREEP()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_CREEP (
            int axis,
            double value )
```

Setter for CREEP axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.435 SetAxisParameter_D_GAIN()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_D_GAIN (
            int axis,
            double value )
```

Setter for D_GAIN axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.436 SetAxisParameter_DAC()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_DAC (
            int axis,
            int value )
```

Setter for DAC axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.437 SetAxisParameter_DAC_SCALE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_DAC_SCALE (
            int axis,
            long value )
```

Setter for DAC_SCALE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.438 SetAxisParameter_DATUM_IN()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_DATUM_IN (
            int axis,
            short value )
```

Setter for DATUM_IN axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.439 SetAxisParameter_DECEL()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_DECEL (
            int axis,
            double value )
```

Setter for DECEL axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.440 SetAxisParameter_DECEL_ANGLE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_DECEL_ANGLE (
            int axis,
            double value )
```

Setter for DECEL_ANGLE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.441 SetAxisParameter_DRIVE_CONTROL()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_DRIVE_CONTROL (
            int axis,
            int value )
```

Setter for DRIVE_CONTROL axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.442 SetAxisParameter_DRIVE_CONTROLWORD()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_DRIVE_CONTROLWORD (
            int axis,
            int value )
```

Setter for DRIVE_CONTROLWORD axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.443 SetAxisParameter_DRIVE_CW_MODE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_DRIVE_CW_MODE (
            int axis,
            int value )
```

Setter for DRIVE_CW_MODE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.444 SetAxisParameter_DRIVE_ENABLE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_DRIVE_ENABLE (
            int axis,
            long value )
```

Setter for DRIVE_ENABLE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.445 SetAxisParameter_DRIVE_FE_LIMIT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_DRIVE_FE_LIMIT (
            int axis,
            long value )
```

Setter for DRIVE_FE_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.446 SetAxisParameter_DRIVE_INDEX()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_DRIVE_INDEX (
            int axis,
            int value )
```

Setter for DRIVE_INDEX axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.447 SetAxisParameter_DRIVE_MODE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_DRIVE_MODE (
            int axis,
            int value )
```

Setter for DRIVE_MODE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.448 SetAxisParameter_DRIVE_NEG_TORQUE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_DRIVE_NEG_TORQUE (
            int axis,
            int value )
```

Setter for DRIVE_NEG_TORQUE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.449 SetAxisParameter_DRIVE_PARAMETER()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_DRIVE_PARAMETER (
            int axis,
            int value )
```

Setter for DRIVE_PARAMETER axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.450 SetAxisParameter_DRIVE_POS_TORQUE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_DRIVE_POS_TORQUE (
            int axis,
            int value )
```

Setter for DRIVE_POS_TORQUE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.451 SetAxisParameter_DRIVE_PROFILE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_DRIVE_PROFILE (
            int axis,
            int value )
```

Setter for DRIVE_PROFILE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.452 SetAxisParameter_DRIVE_SET_VAL()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_DRIVE_SET_VAL (
            int axis,
            int value )
```

Setter for DRIVE_SET_VAL axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.453 SetAxisParameter_DRIVE_TYPE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_DRIVE_TYPE (
            int axis,
            int value )
```

Setter for DRIVE_TYPE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.454 SetAxisParameter_DRIVE_VALUE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_DRIVE_VALUE (
            int axis,
            int value )
```

Setter for DRIVE_VALUE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.455 SetAxisParameter_ENCODER_BITS()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_ENCODER_BITS (
            int axis,
            ushort value )
```

Setter for ENCODER_BITS axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.456 SetAxisParameter_ENCODER_CONTROL()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_ENCODER_CONTROL (
            int axis,
            int value )
```

Setter for ENCODER_CONTROL axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.457 SetAxisParameter_ENCODER_FILTER()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_ENCODER_FILTER (
            int axis,
            double value )
```

Setter for ENCODER_FILTER axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.458 SetAxisParameter_ENCODER_ID()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_ENCODER_ID (
            int axis,
            int value )
```

Setter for ENCODER_ID axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.459 SetAxisParameter_ENCODER_STATUS()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_ENCODER_STATUS (
            int axis,
            int value )
```

Setter for ENCODER_STATUS axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.460 SetAxisParameter_ENCODER_TURNS()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_ENCODER_TURNS (
            int axis,
            int value )
```

Setter for ENCODER_TURNS axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.461 SetAxisParameter_END_DIR_LAST()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_END_DIR_LAST (
            int axis,
            double value )
```

Setter for END_DIR_LAST axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.462 SetAxisParameter_ENDMOVE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_ENDMOVE (
            int axis,
            long value )
```

Setter for ENDMOVE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.463 SetAxisParameter_ENDMOVE_BUFFER()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_ENDMOVE_BUFFER (
            int axis,
            double value )
```

Setter for ENDMOVE_BUFFER axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.464 SetAxisParameter_ENDMOVE_SPEED()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_ENDMOVE_SPEED (
            int axis,
            double value )
```

Setter for ENDMOVE_SPEED axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.465 SetAxisParameter_ERRORMASK()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_ERRORMASK (
            int axis,
            AxisStatus value )
```

Setter for ERRORMASK axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.466 SetAxisParameter_FAST_JOG()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FAST_JOG (
            int axis,
            short value )
```

Setter for FAST_JOG axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.467 SetAxisParameter_FASTDEC()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FASTDEC (
            int axis,
            double value )
```

Setter for FASTDEC axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.468 SetAxisParameter_FE_LIMIT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FE_LIMIT (
            int axis,
            double value )
```

Setter for FE_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.469 SetAxisParameter_FE_LIMIT_MODE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FE_LIMIT_MODE (
            int axis,
            short value )
```

Setter for FE_LIMIT_MODE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.470 SetAxisParameter_FE_RANGE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FE_RANGE (
            int axis,
            double value )
```

Setter for FE_RANGE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.471 SetAxisParameter_FEED_OVERRIDE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FEED_OVERRIDE (
            int axis,
            double value )
```

Setter for FEED_OVERRIDE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.472 SetAxisParameter_FHOLD_IN()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FHOLD_IN (
            int axis,
            short value )
```

Setter for FHOLD_IN axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.473 SetAxisParameter_FHSPEED()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FHSPEED (
            int axis,
            double value )
```

Setter for FHSPEED axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.474 SetAxisParameter_FORCE_ACCEL()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FORCE_ACCEL (
            int axis,
            double value )
```

Setter for FORCE_ACCEL axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.475 SetAxisParameter_FORCE_DECEL()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FORCE_DECEL (
            int axis,
            double value )
```

Setter for FORCE_DECEL axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.476 SetAxisParameter_FORCE_DWELL()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FORCE_DWELL (
            int axis,
            int value )
```

Setter for FORCE_DWELL axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.477 SetAxisParameter_FORCE_JERK()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FORCE_JERK (
            int axis,
            double value )
```

Setter for FORCE_JERK axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.478 SetAxisParameter_FORCE_RAMP()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FORCE_RAMP (
            int axis,
            double value )
```

Setter for FORCE_RAMP axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.479 SetAxisParameter_FORCE_SPEED()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FORCE_SPEED (
            int axis,
            double value )
```

Setter for FORCE_SPEED axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.480 SetAxisParameter_FRAME()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FRAME (
            int axis,
            int value )
```

Setter for FRAME axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.481 SetAxisParameter_FRAME_REP_DIST()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FRAME_REP_DIST (
            int axis,
            double value )
```

Setter for FRAME_REP_DIST axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.482 SetAxisParameter_FRAME_SCALE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FRAME_SCALE (
            int axis,
            double value )
```

Setter for FRAME_SCALE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.483 SetAxisParameter_FS_LIMIT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FS_LIMIT (
            int axis,
            double value )
```

Setter for FS_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.484 SetAxisParameter_FULL_SP_RADIUS()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FULL_SP_RADIUS (
            int axis,
            double value )
```

Setter for FULL_SP_RADIUS axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.485 SetAxisParameter_FWD_IN()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FWD_IN (
            int axis,
            short value )
```

Setter for FWD_IN axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.486 SetAxisParameter_FWD_JOG()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FWD_JOG (
            int axis,
            short value )
```

Setter for FWD_JOG axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.487 SetAxisParameter_FWD_START()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_FWD_START (
            int axis,
            short value )
```

Setter for FWD_START axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.488 SetAxisParameter_I_GAIN()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_I_GAIN (
            int axis,
            double value )
```

Setter for I_GAIN axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.489 SetAxisParameter_IN_POS()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_IN_POS (
            int axis,
            uint value )
```

Setter for IN_POS axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.490 SetAxisParameter_IN_POS_DIST()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_IN_POS_DIST (
            int axis,
            long value )
```

Setter for IN_POS_DIST axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.491 SetAxisParameter_IN_POS_SPEED()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_IN_POS_SPEED (
            int axis,
            double value )
```

Setter for IN_POS_SPEED axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.492 SetAxisParameter_INTERP_FACTOR()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_INTERP_FACTOR (
            int axis,
            double value )
```

Setter for INTERP_FACTOR axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.493 SetAxisParameter_INVERT_STEP()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_INVERT_STEP (
            int axis,
            int value )
```

Setter for INVERT_STEP axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.494 SetAxisParameter_JERK()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_JERK (
            int axis,
            double value )
```

Setter for JERK axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.495 SetAxisParameter_JOGSPEED()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_JOGSPEED (
            int axis,
            double value )
```

Setter for JOGSPEED axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.496 SetAxisParameter_LOOKAHEAD_FACTOR()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_LOOKAHEAD_FACTOR (
            int axis,
            double value )
```

Setter for LOOKAHEAD_FACTOR axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.497 SetAxisParameter_MERGE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_MERGE (
            int axis,
            bool value )
```

Setter for MERGE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.498 SetAxisParameter_MOVE_COUNT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_MOVE_COUNT (
            int axis,
            int value )
```

Setter for MOVE_COUNT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.499 SetAxisParameter_MOVE_PA()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_MOVE_PA (
            int axis,
            double value )
```

Setter for MOVE_PA axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.500 SetAxisParameter_MOVE_PA_CONT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_MOVE_PA_CONT (
            int axis,
            int value )
```

Setter for MOVE_PA_CONT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.501 SetAxisParameter_MOVE_PA_IDLE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_MOVE_PA_IDLE (
            int axis,
            double value )
```

Setter for MOVE_PA_IDLE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.502 SetAxisParameter_MOVE_PB()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_MOVE_PB (
            int axis,
            double value )
```

Setter for MOVE_PB axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.503 SetAxisParameter_MOVE_PB_CONT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_MOVE_PB_CONT (
            int axis,
            int value )
```

Setter for MOVE_PB_CONT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.504 SetAxisParameter_MOVE_PB_IDLE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_MOVE_PB_IDLE (
            int axis,
            double value )
```

Setter for MOVE_PB_IDLE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.505 SetAxisParameter_MOVELINK_MODIFY()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_MOVELINK_MODIFY (
            int axis,
            double value )
```

Setter for MOVELINK_MODIFY axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.506 SetAxisParameter_MSPEED_FILTER()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_MSPEED_FILTER (
            int axis,
            double value )
```

Setter for MSPEED_FILTER axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.507 SetAxisParameter_OFFPOS()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_OFFPOS (
            int axis,
            double value )
```

Setter for OFFPOS axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.508 SetAxisParameter_OPEN_WIN()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_OPEN_WIN (
            int axis,
            double value )
```

Setter for OPEN_WIN axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.509 SetAxisParameter_OUTLIMIT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_OUTLIMIT (
            int axis,
            int value )
```

Setter for OUTLIMIT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.510 SetAxisParameter_OV_GAIN()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_OV_GAIN (
            int axis,
            double value )
```

Setter for OV_GAIN axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.511 SetAxisParameter_P_GAIN()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_P_GAIN (
            int axis,
            double value )
```

Setter for P_GAIN axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.512 SetAxisParameter_POS_DELAY()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_POS_DELAY (
            int axis,
            double value )
```

Setter for POS_DELAY axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.513 SetAxisParameter_POSI_SEQ_DELAY()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_POSI_SEQ_DELAY (
            int axis,
            int value )
```

Setter for POSI_SEQ_DELAY axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.514 SetAxisParameter_POSI_SEQ_MODE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_POSI_SEQ_MODE (
            int axis,
            int value )
```

Setter for POSI_SEQ_MODE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.515 SetAxisParameter_PP_STEP()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_PP_STEP (
            int axis,
            int value )
```

Setter for PP_STEP axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.516 SetAxisParameter_PWM_CYCLE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_PWM_CYCLE (
            int axis,
            uint value )
```

Setter for PWM_CYCLE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.517 SetAxisParameter_PWM_MARK()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_PWM_MARK (
            int axis,
            uint value )
```

Setter for PWM_MARK axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.518 SetAxisParameter_RAISE_ANGLE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_RAISE_ANGLE (
            int axis,
            double value )
```

Setter for RAISE_ANGLE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.519 SetAxisParameter_REG_INPUTS()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_REG_INPUTS (
            int axis,
            int value )
```

Setter for REG_INPUTS axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.520 SetAxisParameter_REGIST_CONTROL()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_REGIST_CONTROL (
            int axis,
            int value )
```

Setter for REGIST_CONTROL axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.521 SetAxisParameter_REGIST_DELAY()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_REGIST_DELAY (
            int axis,
            double value )
```

Setter for REGIST_DELAY axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.522 SetAxisParameter_REP_DIST()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_REP_DIST (
            int axis,
            double value )
```

Setter for REP_DIST axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.523 SetAxisParameter_REP_OPTION()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_REP_OPTION (
            int axis,
            short value )
```

Setter for REP_OPTION axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.524 SetAxisParameter_REV_IN()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_REV_IN (
            int axis,
            short value )
```

Setter for REV_IN axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.525 SetAxisParameter_REV_JOG()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_REV_JOG (
            int axis,
            short value )
```

Setter for REV_JOG axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.526 SetAxisParameter_REV_START()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_REV_START (
            int axis,
            short value )
```

Setter for REV_START axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.527 SetAxisParameter_ROBOT_FE_RANGE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_ROBOT_FE_RANGE (
            int axis,
            double value )
```

Setter for ROBOT_FE_RANGE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.528 SetAxisParameter_ROBOT_FS_LIMIT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_ROBOT_FS_LIMIT (
            int axis,
            double value )
```

Setter for ROBOT_FS_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.529 SetAxisParameter_ROBOT_RS_LIMIT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_ROBOT_RS_LIMIT (
            int axis,
            double value )
```

Setter for ROBOT_RS_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.530 SetAxisParameter_ROBOT_SP_MODE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_ROBOT_SP_MODE (
            int axis,
            short value )
```

Setter for ROBOT_SP_MODE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.531 SetAxisParameter_ROBOT_UNITS()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_ROBOT_UNITS (
            int axis,
            double value )
```

Setter for ROBOT_UNITS axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.532 SetAxisParameter_RS_LIMIT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_RS_LIMIT (
            int axis,
            double value )
```

Setter for RS_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.533 SetAxisParameter_S_REF()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_S_REF (
            int axis,
            int value )
```

Setter for S_REF axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.534 SetAxisParameter_S_REF_OUT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_S_REF_OUT (
            int axis,
            int value )
```

Setter for S_REF_OUT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.535 SetAxisParameter_SERVO()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_SERVO (
            int axis,
            bool value )
```

Setter for SERVO axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.536 SetAxisParameter_SLOT_NUMBER()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_SLOT_NUMBER (
            int axis,
            short value )
```

Setter for SLOT_NUMBER axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.537 SetAxisParameter_SPEED()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_SPEED (
            int axis,
            double value )
```

Setter for SPEED axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.538 SetAxisParameter_SRAMP()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_SRAMP (
            int axis,
            int value )
```

Setter for SRAMP axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.539 SetAxisParameter_START_DIR_LAST()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_START_DIR_LAST (
            int axis,
            double value )
```

Setter for START_DIR_LAST axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.540 SetAxisParameter_STARTMOVE_SPEED()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_STARTMOVE_SPEED (
            int axis,
            double value )
```

Setter for STARTMOVE_SPEED axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.541 SetAxisParameter_STOP_ANGLE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_STOP_ANGLE (
            int axis,
            double value )
```

Setter for STOP_ANGLE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.542 SetAxisParameter_SYNC_AXIS()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_SYNC_AXIS (
            int axis,
            int value )
```

Setter for SYNC_AXIS axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.543 SetAxisParameter_SYNC_CONTROL()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_SYNC_CONTROL (
            int axis,
            SyncControl value )
```

Setter for SYNC_CONTROL axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.544 SetAxisParameter_SYNC_DPOS()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_SYNC_DPOS (
            int axis,
            double value )
```

Setter for SYNC_DPOS axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.545 SetAxisParameter_SYNC_DWELL()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_SYNC_DWELL (
            int axis,
            int value )
```

Setter for SYNC_DWELL axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.546 SetAxisParameter_SYNC_FLIGHT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_SYNC_FLIGHT (
            int axis,
            int value )
```

Setter for SYNC_FLIGHT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.547 SetAxisParameter_SYNC_PAUSE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_SYNC_PAUSE (
            int axis,
            int value )
```

Setter for SYNC_PAUSE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.548 SetAxisParameter_SYNC_TIME()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_SYNC_TIME (
            int axis,
            int value )
```

Setter for SYNC_TIME axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.549 SetAxisParameter_SYNC_WITHDRAW()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_SYNC_WITHDRAW (
            int axis,
            double value )
```

Setter for SYNC_WITHDRAW axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.550 SetAxisParameter_T_REF()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_T_REF (
            int axis,
            int value )
```

Setter for T_REF axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.551 SetAxisParameter_T_REF_OUT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_T_REF_OUT (
            int axis,
            int value )
```

Setter for T_REF_OUT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.552 SetAxisParameter_TABLE_POINTER()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_TABLE_POINTER (
            int axis,
            double value )
```

Setter for TABLE_POINTER axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.553 SetAxisParameter_TANG_DIRECTION()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_TANG_DIRECTION (
            int axis,
            double value )
```

Setter for TANG_DIRECTION axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.554 SetAxisParameter_TCP_FS_LIMIT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_TCP_FS_LIMIT (
            int axis,
            double value )
```

Setter for TCP_FS_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.555 SetAxisParameter_TCP_RS_LIMIT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_TCP_RS_LIMIT (
            int axis,
            double value )
```

Setter for TCP_RS_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.556 SetAxisParameter_TCP_UNITS()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_TCP_UNITS (
            int axis,
            double value )
```

Setter for TCP_UNITS axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.557 SetAxisParameter_UNITS()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_UNITS (
            int axis,
            double value )
```

Setter for UNITS axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.558 SetAxisParameter_VECTOR_BUFFERED()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_VECTOR_BUFFERED (
            int axis,
            double value )
```

Setter for VECTOR_BUFFERED axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.559 SetAxisParameter_VERIFY()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_VERIFY (
            int axis,
            short value )
```

Setter for VERIFY axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.560 SetAxisParameter_VFF_GAIN()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_VFF_GAIN (
            int axis,
            double value )
```

Setter for VFF_GAIN axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.561 SetAxisParameter_VP_MODE()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_VP_MODE (
            int axis,
            int value )
```

Setter for VP_MODE axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.562 SetAxisParameter_WORLD_ACCEL()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_WORLD_ACCEL (
            int axis,
            double value )
```

Setter for WORLD_ACCEL axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.563 SetAxisParameter_WORLD_DECEL()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_WORLD_DECEL (
            int axis,
            double value )
```

Setter for WORLD_DECEL axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.564 SetAxisParameter_WORLD_FS_LIMIT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_WORLD_FS_LIMIT (
            int axis,
            double value )
```

Setter for WORLD_FS_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.565 SetAxisParameter_WORLD_JOGSPEED()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_WORLD_JOGSPEED (
            int axis,
            double value )
```

Setter for WORLD_JOGSPEED axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.566 SetAxisParameter_WORLD_RS_LIMIT()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_WORLD_RS_LIMIT (
            int axis,
            double value )
```

Setter for WORLD_RS_LIMIT axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.567 SetAxisParameter_WORLD_SPEED()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_WORLD_SPEED (
            int axis,
            double value )
```

Setter for WORLD_SPEED axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.568 SetAxisParameter_WORLD_UNITS()

```
void Trio.UnifiedApi.ITrioConnection.SetAxisParameter_WORLD_UNITS (
            int axis,
            double value )
```

Setter for WORLD_UNITS axis parameter.

Parameters

axis	Target axis whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.569 SetDOut() [1/2]

```
void Trio.UnifiedApi.ITrioConnection.SetDOut (
            UInt32 channel,
            bool state )
```

Set the state of the Digital Output.

Parameters

channel	Digital output channel number
state	New state for the digital output channel

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.570 SetDOut() [2/2]

```
void Trio.UnifiedApi.ITrioConnection.SetDOut (
            UInt32 firstChannel,
            UInt32 lastChannel,
            UInt32 states )
```

Set the state of the Digital Outputs.

Parameters

firstChannel	First digital output channel number
lastChannel	Last digital output channel number
states	Binary sum of digital output states

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.571 SetDOut32()

```
void Trio.UnifiedApi.ITrioConnection.SetDOut32 (
            UInt32 states32 )
```

Set the state of the first 32 Digital Outputs.

Parameters

states32	Bit sum of new the states for the first 32 digital output channels

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.572 SetFrameGroup()

```
void Trio.UnifiedApi.ITrioConnection.SetFrameGroup (
            UInt32 group,
            UInt32 tableOffset,
            Int32[] axes )
```

Define new frame group.

Parameters

group	Frame group number to define
tableOffset	Start offset of TABLE range used by the frame group
axes	Array of axes to be added to the frame group

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.573 SetInvertIN()

```
void Trio.UnifiedApi.ITrioConnection.SetInvertIN (
            UInt32 input,
            bool invert )
```

The INVERT_IN command allows the input channels to be individually inverted in software.

Parameters

input	The input to invert
invert	True to invert, false not to invert

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.574 SetMotion() [1/4]

```
void Trio.UnifiedApi.ITrioConnection.SetMotion (
            Int32 axis,
            double speed )
```

Function used to pre-set the parameter list for a subsequent move using just one command.

Parameters

axis	Target axis for whose parameters are to be modified
speed	The value for SPEED to be set

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.575 SetMotion() [2/4]

```
void Trio.UnifiedApi.ITrioConnection.SetMotion (
            Int32 axis,
            double speed,
            double accel )
```

Function used to pre-set the parameter list for a subsequent move using just one command.

Parameters

axis	Target axis for whose parameters are to be modified
speed	The value for SPEED to be set
accel	The value for ACCEL to be set

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.576 SetMotion() [3/4]

```
void Trio.UnifiedApi.ITrioConnection.SetMotion (
            Int32 axis,
            double speed,
            double accel,
            double decel )
```

Function used to pre-set the parameter list for a subsequent move using just one command.

Parameters

axis	Target axis for whose parameters are to be modified
speed	The value for SPEED to be set
accel	The value for ACCEL to be set
decel	The value for DECEL to be set

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.577 SetMotion() [4/4]

```
void Trio.UnifiedApi.ITrioConnection.SetMotion (
            Int32 axis,
            double speed,
            double accel,
            double decel,
            double jerk )
```

Function used to pre-set the parameter list for a subsequent move using just one command.

Parameters

axis	Target axis for whose parameters are to be modified
speed	The value for SPEED to be set
accel	The value for ACCEL to be set
decel	The value for DECEL to be set
jerk	The value for JERK to be set

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.578 SetMotionSp() [1/8]

```
void Trio.UnifiedApi.ITrioConnection.SetMotionSp (
            Int32 axis,
            double forceSpeed )
```

SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP, MOVECIRCSP etc. using just one command.

Parameters

axis	Target axis for whose parameters are to be modified
forceSpeed	The value for FORCE_SPEED to be set

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.579 SetMotionSp() [2/8]

```
void Trio.UnifiedApi.ITrioConnection.SetMotionSp (
            Int32 axis,
            double forceSpeed,
            double startMoveSpeed )
```

SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP, MOVECIRCSP etc. using just one command.

Parameters

axis	Target axis for whose parameters are to be modified
forceSpeed	The value for FORCE_SPEED to be set
startMoveSpeed	The value for STARTMOVE_SPEED to be set

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.580 SetMotionSp() [3/8]

```
void Trio.UnifiedApi.ITrioConnection.SetMotionSp (
            Int32 axis,
```

```
            double forceSpeed,
            double startMoveSpeed,
            double endMoveSpeed )
```

SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP, MOVECIRCSP etc. using just one command.

Parameters

axis	Target axis for whose parameters are to be modified
forceSpeed	The value for FORCE_SPEED to be set
startMoveSpeed	The value for STARTMOVE_SPEED to be set
endMoveSpeed	The value for ENDMOVE_SPEED to be set

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.581 SetMotionSp() [4/8]

```
void Trio.UnifiedApi.ITrioConnection.SetMotionSp (
            Int32 axis,
            double forceSpeed,
            double startMoveSpeed,
            double endMoveSpeed,
            double forceAccel )
```

SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP, MOVECIRCSP etc. using just one command.

Parameters

axis	Target axis for whose parameters are to be modified
forceSpeed	The value for FORCE_SPEED to be set
startMoveSpeed	The value for STARTMOVE_SPEED to be set
endMoveSpeed	The value for ENDMOVE_SPEED to be set
forceAccel	The value for FORCE_ACCEL to be set

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.582 SetMotionSp() [5/8]

```
void Trio.UnifiedApi.ITrioConnection.SetMotionSp (
            Int32 axis,
            double forceSpeed,
            double startMoveSpeed,
```

```
            double endMoveSpeed,
            double forceAccel,
            double forceDecel )
```

SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP, MOVECIRCSP etc. using just one command.

Parameters

axis	Target axis for whose parameters are to be modified
forceSpeed	The value for FORCE_SPEED to be set
startMoveSpeed	The value for STARTMOVE_SPEED to be set
endMoveSpeed	The value for ENDMOVE_SPEED to be set
forceAccel	The value for FORCE_ACCEL to be set
forceDecel	The value for FORCE_DECEL to be set

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.583 SetMotionSp() [6/8]

```
void Trio.UnifiedApi.ITrioConnection.SetMotionSp (
            Int32 axis,
            double forceSpeed,
            double startMoveSpeed,
            double endMoveSpeed,
            double forceAccel,
            double forceDecel,
            double forceJerk )
```

SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP, MOVECIRCSP etc. using just one command.

Parameters

axis	Target axis for whose parameters are to be modified
forceSpeed	The value for FORCE_SPEED to be set
startMoveSpeed	The value for STARTMOVE_SPEED to be set
endMoveSpeed	The value for ENDMOVE_SPEED to be set
forceAccel	The value for FORCE_ACCEL to be set
forceDecel	The value for FORCE_DECEL to be set
forceJerk	The value for FORCE_JERK to be set

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.584 SetMotionSp() [7/8]

```
void Trio.UnifiedApi.ITrioConnection.SetMotionSp (
            Int32 axis,
            double forceSpeed,
            double startMoveSpeed,
            double endMoveSpeed,
            double forceAccel,
            double forceDecel,
            double forceJerk,
            double forceRamp )
```

SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP, MOVECIRCSP etc. using just one command.

Parameters

axis	Target axis for whose parameters are to be modified
forceSpeed	The value for FORCE_SPEED to be set
startMoveSpeed	The value for STARTMOVE_SPEED to be set
endMoveSpeed	The value for ENDMOVE_SPEED to be set
forceAccel	The value for FORCE_ACCEL to be set
forceDecel	The value for FORCE_DECEL to be set
forceJerk	The value for FORCE_JERK to be set
forceRamp	The value for FORCE_RAMP to be set

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.585 SetMotionSp() [8/8]

```
void Trio.UnifiedApi.ITrioConnection.SetMotionSp (
            Int32 axis,
            double forceSpeed,
            double startMoveSpeed,
            double endMoveSpeed,
            double forceAccel,
            double forceDecel,
            double forceJerk,
            double forceRamp,
            double forceDwell )
```

SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP, MOVECIRCSP etc. using just one command.

Parameters

axis	Target axis for whose parameters are to be modified
forceSpeed	The value for FORCE_SPEED to be set
startMoveSpeed	The value for STARTMOVE_SPEED to be set

Parameters

endMoveSpeed	The value for ENDMOVE_SPEED to be set
forceAccel	The value for FORCE_ACCEL to be set
forceDecel	The value for FORCE_DECEL to be set
forceJerk	The value for FORCE_JERK to be set
forceRamp	The value for FORCE_RAMP to be set
forceDwell	The value for FORCE_DWELL to be set

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.586 SetMultiAxisBase()

```
void Trio.UnifiedApi.ITrioConnection.SetMultiAxisBase (
            Int32[] axes )
```

Set multiple base axes.

Parameters

axes	Array of axis indexes to set as base

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.587 SetMultiTableValues()

```
void Trio.UnifiedApi.ITrioConnection.SetMultiTableValues (
            UInt32 start_index,
            UInt32 count,
            double[] values )
```

Set multiple table entries.

Parameters

start_index	Start index of the table range to be written
count	Number of table entries to be written
values	Array of values to be written

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.588 SetMultiVrValues()

```
void Trio.UnifiedApi.ITrioConnection.SetMultiVrValues (
            UInt32 start_index,
            UInt32 count,
            double[] values )
```

Set multiple VR entries.

Parameters

start_index	Start index of the VR range to be written
count	Number of VR entries to be written
values	Array of VR values to be written

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.589 SetProcessParameter_INDEVICE()

```
void Trio.UnifiedApi.ITrioConnection.SetProcessParameter_INDEVICE (
            int process,
            int value )
```

Setter for INDEVICE process parameter.

Parameters

process	Target process whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.590 SetProcessParameter_OUTDEVICE()

```
void Trio.UnifiedApi.ITrioConnection.SetProcessParameter_OUTDEVICE (
            int process,
            int value )
```

Setter for OUTDEVICE process parameter.

Parameters

process	Target process whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.591 SetProcessParameter_TICKS()

```
void Trio.UnifiedApi.ITrioConnection.SetProcessParameter_TICKS (
            int process,
            long value )
```

Setter for TICKS process parameter.

Parameters

process	Target process whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.592 SetSlotParameter_SERIAL_NUMBER()

```
void Trio.UnifiedApi.ITrioConnection.SetSlotParameter_SERIAL_NUMBER (
            int slot,
            int value )
```

Setter for SERIAL_NUMBER slot parameter.

Parameters

slot	Target slot whose parameter is modified
value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.593 SetSystemParameter_ADDRESS()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_ADDRESS (
            uint value )
```

Setter for ADDRESS system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.594 SetSystemParameter_AIN0()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_AIN0 (
            uint value )
```

Setter for AIN0 system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.595 SetSystemParameter_AIN1()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_AIN1 (
            uint value )
```

Setter for AIN1 system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.596 SetSystemParameter_AIN2()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_AIN2 (
            uint value )
```

Setter for AIN2 system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.597 SetSystemParameter_AIN3()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_AIN3 (
            uint value )
```

Setter for AIN3 system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.598 SetSystemParameter_AINBI0()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_AINBI0 (
            uint value )
```

Setter for AINBI0 system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.599 SetSystemParameter_AINBI1()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_AINBI1 (
            uint value )
```

Setter for AINBI1 system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.600 SetSystemParameter_AINBI2()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_AINBI2 (
            uint value )
```

Setter for AINBI2 system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.601 SetSystemParameter_AINBI3()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_AINBI3 (
            uint value )
```

Setter for AINBI3 system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.602 SetSystemParameter_AOUT0()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_AOUT0 (
            uint value )
```

Setter for AOUT0 system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.603 SetSystemParameter_AOUT1()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_AOUT1 (
            uint value )
```

Setter for AOUT1 system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.604 SetSystemParameter_AOUT2()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_AOUT2 (
            uint value )
```

Setter for AOUT2 system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.605 SetSystemParameter_AOUT3()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_AOUT3 (
            uint value )
```

Setter for AOUT3 system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.606 SetSystemParameter_AUTO_ETHERCAT()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_AUTO_ETHERCAT (
            int value )
```

Setter for AUTO_ETHERCAT system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.607 SetSystemParameter_AVERAGE_JITTER()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_AVERAGE_JITTER (
            long value )
```

Setter for AVERAGE_JITTER system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.608 SetSystemParameter_CPU_EXCEPTIONS()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_CPU_EXCEPTIONS (
            long value )
```

Setter for CPU_EXCEPTIONS system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.609 SetSystemParameter_CURRENT_JITTER()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_CURRENT_JITTER (
            long value )
```

Setter for CURRENT_JITTER system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.610 SetSystemParameter_DISPLAY()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_DISPLAY (
            int value )
```

Setter for DISPLAY system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.611 SetSystemParameter_FPU_EXCEPTIONS()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_FPU_EXCEPTIONS (
            long value )
```

Setter for FPU_EXCEPTIONS system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.612 SetSystemParameter_HMI_PROC()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_HMI_PROC (
            int value )
```

Setter for HMI_PROC system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.613 SetSystemParameter_IP_ADDRESS()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_IP_ADDRESS (
            uint value )
```

Setter for IP_ADDRESS system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.614 SetSystemParameter_IP_GATEWAY()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_IP_GATEWAY (
            uint value )
```

Setter for IP_GATEWAY system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.615 SetSystemParameter_IP_MAC()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_IP_MAC (
            ulong value )
```

Setter for IP_MAC system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.616 SetSystemParameter_IP_MEMORY_CONFIG()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_IP_MEMORY_CONFIG (
            uint value )
```

Setter for IP_MEMORY_CONFIG system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.617 SetSystemParameter_IP_NETMASK()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_IP_NETMASK (
            uint value )
```

Setter for IP_NETMASK system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.618 SetSystemParameter_IP_PROTOCOL_CONFIG()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_IP_PROTOCOL_CONFIG (
            uint value )
```

Setter for IP_PROTOCOL_CONFIG system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.619 SetSystemParameter_IP_PROTOCOL_CTRL()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_IP_PROTOCOL_CTRL (
            uint value )
```

Setter for IP_PROTOCOL_CTRL system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.620 SetSystemParameter_IP_TCP_TIMEOUT()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_IP_TCP_TIMEOUT (
            ulong value )
```

Setter for IP_TCP_TIMEOUT system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.621 SetSystemParameter_IP_TCP_TX_THRESHOLD()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_IP_TCP_TX_THRESHOLD (
            uint value )
```

Setter for IP_TCP_TX_THRESHOLD system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.622 SetSystemParameter_IP_TCP_TX_TIMEOUT()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_IP_TCP_TX_TIMEOUT (
            uint value )
```

Setter for IP_TCP_TX_TIMEOUT system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.623 SetSystemParameter_LAST_AXIS()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_LAST_AXIS (
            int value )
```

Setter for LAST_AXIS system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.624 SetSystemParameter_LIMIT_BUFFERED()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_LIMIT_BUFFERED (
            short value )
```

Setter for LIMIT_BUFFERED system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.625 SetSystemParameter_MAX_JITTER()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_MAX_JITTER (
            long value )
```

Setter for MAX_JITTER system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.626 SetSystemParameter_MISSED_INTERRUPTS()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_MISSED_INTERRUPTS (
            long value )
```

Setter for MISSED_INTERRUPTS system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.627 SetSystemParameter_MISSED_RECV_FRAMES()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_MISSED_RECV_FRAMES (
            long value )
```

Setter for MISSED_RECV_FRAMES system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.628 SetSystemParameter_MISSED_SEND_FRAMES()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_MISSED_SEND_FRAMES (
            long value )
```

Setter for MISSED_SEND_FRAMES system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.629 SetSystemParameter_MODBUS_TX_DELAY()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_MODBUS_TX_DELAY (
            ushort value )
```

Setter for MODBUS_TX_DELAY system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.630 SetSystemParameter_NIO()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_NIO (
            uint value )
```

Setter for NIO system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.631 SetSystemParameter_PCMCAT_PROC()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_PCMCAT_PROC (
            int value )
```

Setter for PCMCAT_PROC system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.632 SetSystemParameter_PLC_CONFIG()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_PLC_CONFIG (
            int value )
```

Setter for PLC_CONFIG system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.633 SetSystemParameter_REMOTE_PROC()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_REMOTE_PROC (
            int value )
```

Setter for REMOTE_PROC system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.634 SetSystemParameter_SCHEDULE_TYPE()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_SCHEDULE_TYPE (
            uint value )
```

Setter for SCHEDULE_TYPE system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.635 SetSystemParameter_SCOPE_CYCLE_COUNT()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_SCOPE_CYCLE_COUNT (
            int value )
```

Setter for SCOPE_CYCLE_COUNT system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.636 SetSystemParameter_SCOPE_DELAY()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_SCOPE_DELAY (
            long value )
```

Setter for SCOPE_DELAY system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.637 SetSystemParameter_SCOPE_POS()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_SCOPE_POS (
            int value )
```

Setter for SCOPE_POS system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.638 SetSystemParameter_SERIAL_NUMBER()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_SERIAL_NUMBER (
            int value )
```

Setter for SERIAL_NUMBER system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.639 SetSystemParameter_SERVO_OFFSET()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_SERVO_OFFSET (
            int value )
```

Setter for SERVO_OFFSET system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.640 SetSystemParameter_SERVO_PERIOD()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_SERVO_PERIOD (
            int value )
```

Setter for SERVO_PERIOD system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.641 SetSystemParameter_SLOT0_TIME()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_SLOT0_TIME (
            uint value )
```

Setter for SLOT0_TIME system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.642 SetSystemParameter_SLOT1_TIME()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_SLOT1_TIME (
            uint value )
```

Setter for SLOT1_TIME system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.643 SetSystemParameter_SLOT2_TIME()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_SLOT2_TIME (
            uint value )
```

Setter for SLOT2_TIME system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.644 SetSystemParameter_SLOT3_TIME()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_SLOT3_TIME (
            uint value )
```

Setter for SLOT3_TIME system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.645 SetSystemParameter_SYSTEM_ERROR_MASK()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_SYSTEM_ERROR_MASK (
            uint value )
```

Setter for SYSTEM_ERROR_MASK system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.646 SetSystemParameter_SYSTEM_LOAD_MAX()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_SYSTEM_LOAD_MAX (
            uint value )
```

Setter for SYSTEM_LOAD_MAX system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.647 SetSystemParameter_TEXT_FILE_LOADER_PROC()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_TEXT_FILE_LOADER_PROC (
            int value )
```

Setter for TEXT_FILE_LOADER_PROC system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.648 SetSystemParameter_WDOG()

```
void Trio.UnifiedApi.ITrioConnection.SetSystemParameter_WDOG (
            bool value )
```

Setter for WDOG system parameter.

Parameters

value	New value to set to parameter

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.649 SetTableValue()

```
void Trio.UnifiedApi.ITrioConnection.SetTableValue (
            UInt32 index,
            double value )
```

Set single table value.

Parameters

index	Index of the table entry to be written
value	New value to be set

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.650 SetTcpHostName()

```
void Trio.UnifiedApi.ITrioConnection.SetTcpHostName (
            String host )
```

Set host name or IP to connection object. Valid only for TCP connection.

Parameters

host	New value for the host name

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.651 SetTcpPort()

```
void Trio.UnifiedApi.ITrioConnection.SetTcpPort (
            UInt16 port )
```

Set TCP port to connection object. Valid only for TCP connection.

Parameters

port	New value for the TCP port

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.652 SetTcpTimeout()

```
void Trio.UnifiedApi.ITrioConnection.SetTcpTimeout (
            UInt32 timeout_ms )
```

Set TCP connection timeout. Valid only for TCP connection.

Parameters

timeout_ms	New value for the TCP connection timeout in milliseconds

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.653 SetVrString()

```
void Trio.UnifiedApi.ITrioConnection.SetVrString (
            UInt32 vr,
            String str )
```

Set a null-terminated string into consecutive VRs.

Parameters

vr	Starting VR index
str	String to be set

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.654 SetVrValue()

```
void Trio.UnifiedApi.ITrioConnection.SetVrValue (
            UInt32 index,
            double value )
```

Set single VR value.

Parameters

index	Index of the VR entry to be written
value	New value to be written

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.655 StepRatio()

```
void Trio.UnifiedApi.ITrioConnection.StepRatio (
            Int64 output_count,
            Int64 dpos_count,
            Int32 axis = -1 )
```

This command sets up an integer ratio for the axis' demanded position. Usually this is used on a stepper (pulse+direction) output. It can also be applied to a networked axis in positon mode, such as EtherCAT and RTEX position servo. Every servo-period the change in position is passed through the STEP_RATIO function before it goes to the step pulse or network drive output. The STEP_RATIO function operates before the divide by 16 factor(divide by 256 in Flexslice P375) in the stepper axis.This maintains the good timing resolution of the stepper output circuit. STEP_RATIO does not replace UNITS.

Parameters

output_count	Number of counts to output for the given dpos_count value. Range: 0 to 16777215
dpos_count	Change in DPOS value for corresponding output count. Range: 0 to 16777215.
axis	Base axis for this command. If value is -1 then the default base axis is used

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.656 StopProgram() [1/2]

```
void Trio.UnifiedApi.ITrioConnection.StopProgram (
            String name )
```

Stops a program at its current line.

Parameters

name	Name of the program to be stopped.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.657 StopProgram() [2/2]

```
void Trio.UnifiedApi.ITrioConnection.StopProgram (
            String name,
            Int32 process_number )
```

Stops a program at its current line.

Parameters

name	Name of the program to be stopped.
process_number	Optional process number to be used when multiple instances of the program are running and only one is to be stopped.

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.2.658 UploadFile()

```
void Trio.UnifiedApi.ITrioConnection.UploadFile (
            String local_file,
            String remote_file,
            FileType remote_file_type,
            FileTransferOption opt,
            ProgressHandlerCallback handler )
```

Uploads a file to the Motion Coordinator.

Parameters

local_file	Local file name
remote_file	Remote file name. Can be prefixed by: "RAM:" - the location of the file is in the controller RAM. "SD:" - the location of the file is on the SD card. "FIFO:" - The target file is a FIFO file on the controller. It is expected that the controller (a program) will consume the data. If no prefix "RAM:" is assumed.
remote_file_type	The type of the file. Ignored when uploading to the SD card or with for FIFO transfers.
opt	File transfer option
handler	Callback handler with the progress advance

Exceptions

TrioConnectionException	Thrown if error occurs in communication module.

6.3.3 Property Documentation

6.3.3.1 ConnectionType

```
ConnectionType Trio.UnifiedApi.ITrioConnection.ConnectionType  [get]
```

Connection type or current connection instance or Unknown if connection is disposed.

Definition at line 2087 of file ITrioConnection.cs.

6.3.3.2 IsConnected

```
bool Trio.UnifiedApi.ITrioConnection.IsConnected  [get]
```

True if connection to Motion Coordinator was opened.

Definition at line 2082 of file ITrioConnection.cs.

6.3.4 Event Documentation

6.3.4.1 ConnectionEvent

```
ConnectionEventHandler Trio.UnifiedApi.ITrioConnection.ConnectionEvent
```

Connection related event.

Definition at line 2077 of file ITrioConnection.cs.

The documentation for this interface was generated from the following file:

- ITrioConnection.cs

6.4 Trio.UnifiedApi.ProgressInfo Struct Reference

Data used with the progress report.

Public Attributes

- uint total_length

  The length of the process. If 0 the length is unknown.
- uint current_pos

  The current progress position.
- ProgressEvent evt

  Progress event.

6.4.1 Detailed Description

Data used with the progress report.

Definition at line 2015 of file ITrioConnection.cs.

6.4.2 Member Data Documentation

6.4.2.1 current_pos

```
uint Trio.UnifiedApi.ProgressInfo.current_pos
```

The current progress position.

Definition at line 2024 of file ITrioConnection.cs.

6.4.2.2 evt

```
ProgressEvent Trio.UnifiedApi.ProgressInfo.evt
```

Progress event.

Definition at line 2028 of file ITrioConnection.cs.

6.4.2.3 total_length

```
uint Trio.UnifiedApi.ProgressInfo.total_length
```

The length of the process. If 0 the length is unknown.

Definition at line 2020 of file ITrioConnection.cs.

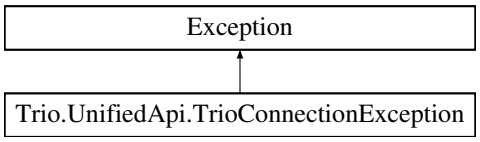
The documentation for this struct was generated from the following file:

- ITrioConnection.cs

6.5 Trio.UnifiedApi.TrioConnectionException Class Reference

Exception raised when error occurs in communication library.

Inheritance diagram for Trio.UnifiedApi.TrioConnectionException:



Public Member Functions

- TrioConnectionException (ErrorCode errorCode)
  Initializing constructor.

Properties

- ErrorCode ErrorCode `[get]`
  Error code associated causing the exception.
- override string Message `[get]`
  Text representation of the error.

6.5.1 Detailed Description

Exception raised when error occurs in communication library.

Definition at line 2043 of file ITrioConnection.cs.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 TrioConnectionException()

```
Trio.UnifiedApi.TrioConnectionException.TrioConnectionException (
            ErrorCode errorCode )  [inline]
```

Initializing constructor.

Parameters

errorCode	

Definition at line 2059 of file ITrioConnection.cs.

6.5.3 Property Documentation

6.5.3.1 ErrorCode

```
ErrorCode Trio.UnifiedApi.TrioConnectionException.ErrorCode  [get]
```

Error code associated causing the exception.

Definition at line 2048 of file ITrioConnection.cs.

6.5.3.2 Message

```
override string Trio.UnifiedApi.TrioConnectionException.Message  [get]
```

Text representation of the error.

Definition at line 2053 of file ITrioConnection.cs.

The documentation for this class was generated from the following file:

- ITrioConnection.cs

6.6 Trio.UnifiedApi.TrioConnectionFactory Class Reference

Factory for creating communication instances.

Static Public Member Functions

- static ITrioConnection CreateConnectionTCP (string hostName=null, ushort port=0)
  Create TCP based connection to Motion Coordinator.
- static ITrioConnection CreateConnectionPCMCAT ()
  Create PC-MCAT based connection to Motion Coordinator.
- static ITrioConnection CreateConnectionFlex7 ()
  Create Flex7 based connection to Motion Coordinator.

Properties

- static Func< ErrorCode, string > ErrorStringGetter [get]
  Delegate for transforming error code to message.

6.6.1 Detailed Description

Factory for creating communication instances.

Definition at line 13 of file TrioConnectionFactory.cs.

6.6.2 Member Function Documentation

6.6.2.1 CreateConnectionFlex7()

```
static ITrioConnection Trio.UnifiedApi.TrioConnectionFactory.CreateConnectionFlex7 ( )  [inline],
[static]
```

Create Flex7 based connection to Motion Coordinator.

Returns

On success an instance of connection interface is returned

Definition at line 44 of file TrioConnectionFactory.cs.

6.6.2.2 CreateConnectionPCMCAT()

```
static ITrioConnection Trio.UnifiedApi.TrioConnectionFactory.CreateConnectionPCMCAT ( )  [inline],
[static]
```

Create PC-MCAT based connection to Motion Coordinator.

Returns

On success an instance of connection interface is returned

Definition at line 35 of file TrioConnectionFactory.cs.

6.6.2.3 CreateConnectionTCP()

```
static ITrioConnection Trio.UnifiedApi.TrioConnectionFactory.CreateConnectionTCP (
            string hostName = null,
            ushort port = 0 )  [inline], [static]
```

Create TCP based connection to Motion Coordinator.

Parameters

hostName	Host name or IP of the Motion Coordinator to connect via TCP protocol
port	Port on Motion Coordinator open for TCP connections. It is normally 3240 unless Motion Coorinator has custom configuration

Returns

On success an instance of connection interface is returned

Definition at line 26 of file TrioConnectionFactory.cs.

6.6.3 Property Documentation

6.6.3.1 ErrorStringGetter

```
Func<ErrorCode, string> Trio.UnifiedApi.TrioConnectionFactory.ErrorStringGetter  [static],
[get]
```

Delegate for transforming error code to message.

Definition at line 18 of file TrioConnectionFactory.cs.

The documentation for this class was generated from the following file:

- TrioConnectionFactory.cs

Chapter 7

File Documentation

7.1 ITrioConnection.cs File Reference

Classes

- struct Trio.UnifiedApi.FileInfo

  Data used with the delegate for processing the response of Dir command.
- struct Trio.UnifiedApi.DataStreamPacket

  Data used with the delegate for streaming back the response of commands that return string data.
- struct Trio.UnifiedApi.ProgressInfo

  Data used with the progress report.
- class Trio.UnifiedApi.TrioConnectionException

  Exception raised when error occurs in communication library.
- interface Trio.UnifiedApi.ITrioConnection

  Connection interface.

Namespaces

- namespace Trio
- namespace Trio.UnifiedApi

Typedefs

- using size_t = System.UInt64

Enumerations

- enum Trio.UnifiedApi.ErrorCode : UInt64 {
  Trio.UnifiedApi.NoError = 0 , Trio.UnifiedApi._Unknown = 0xFFFFFFFF , Trio.UnifiedApi._ApiError = 0x100000000 , Trio.UnifiedApi.Api_Unknown = (_ApiError + 0) ,
  Trio.UnifiedApi.Fail , Trio.UnifiedApi.NotSupported , Trio.UnifiedApi.ConnectionModule , Trio.UnifiedApi.ConnectionContext ,
  Trio.UnifiedApi.TokenTable , Trio.UnifiedApi.Timeout , Trio.UnifiedApi.UnknownFloatingPointFormat , Trio.UnifiedApi.Remote_TokenBuffer ,
  Trio.UnifiedApi.Remote_UnknownToken , Trio.UnifiedApi.UnsupportedController , Trio.UnifiedApi.BadResponse , Trio.UnifiedApi.BadArgument ,
  Trio.UnifiedApi.Reopen , Trio.UnifiedApi.BufferTooSmall , Trio.UnifiedApi.SocketRead , Trio.UnifiedApi.SocketWrite ,
  Trio.UnifiedApi.LowLevelError , Trio.UnifiedApi.VersionMismatch , Trio.UnifiedApi.Compression , Trio.UnifiedApi.Abort ,
  Trio.UnifiedApi.Disconnected , Trio.UnifiedApi._ApiLast , Trio.UnifiedApi._SystemError = 0x200000000 , Trio.UnifiedApi.Sys_Unknown = (_SystemError | _Unknown) ,
  Trio.UnifiedApi._ControllerError = 0x400000000 , Trio.UnifiedApi.ControllerError_NONE = 0 | _ControllerError , Trio.UnifiedApi.ControllerError_COMMAND_NOT_RECOGNISED = 1 | _ControllerError , Trio.UnifiedApi.ControllerError_INVALID_TRANSFER_TYPE = 2 | _ControllerError ,
  Trio.UnifiedApi.ControllerError_FLASH_PROGRAMMING_ERROR = 3 | _ControllerError , Trio.UnifiedApi.ControllerError_OPE = 4 | _ControllerError , Trio.UnifiedApi.ControllerError_ASSIGNMENT_EXPECTED = 5 | _ControllerError ,
  Trio.UnifiedApi.ControllerError_QUOTES_EXPECTED = 6 | _ControllerError ,
  Trio.UnifiedApi.ControllerError_STACK_OVERFLOW = 7 | _ControllerError , Trio.UnifiedApi.ControllerError_TOO_MANY_VARIA = 8 | _ControllerError , Trio.UnifiedApi.ControllerError_DIVISION_BY_ZERO = 9 | _ControllerError ,
  Trio.UnifiedApi.ControllerError_EXTRA_CHARS_AT_EOLN = 10 | _ControllerError ,
  Trio.UnifiedApi.ControllerError_CLOSE_SQUARE_BRACKET_EXPECTED_IN_PRINT = 11 | _ControllerError , Trio.UnifiedApi.ControllerError_CANNOT_MODIOFY_SPECIAL_PROGRAM = 12 | _ControllerError , Trio.UnifiedApi.ControllerError_THEN_EXPECTED_IN_IF_ELSEIF = 13 | _ControllerError ,
  Trio.UnifiedApi.ControllerError_FLASH_ERASE_ERROR = 14 | _ControllerError ,
  Trio.UnifiedApi.ControllerError_START_OF_EXPRESSION_EXPECTED = 15 | _ControllerError , Trio.UnifiedApi.ControllerError = 16 | _ControllerError , Trio.UnifiedApi.ControllerError_COMMA_EXPECTED = 17 | _ControllerError ,
  Trio.UnifiedApi.ControllerError_COMMAND_LINE_BROKEN_BY_ESC = 18 | _ControllerError ,
  Trio.UnifiedApi.ControllerError_PARAMETER_OUT_OF_RANGE = 19 | _ControllerError , Trio.UnifiedApi.ControllerError_NO_P = 20 | _ControllerError , Trio.UnifiedApi.ControllerError_VALUE_IS_READ_ONLY = 21 | _ControllerError ,
  Trio.UnifiedApi.ControllerError_MODIFIER_NOT_ALLOWED = 22 | _ControllerError ,
  Trio.UnifiedApi.ControllerError_REMOTE_AXIS_IS_IN_USE = 23 | _ControllerError , Trio.UnifiedApi.ControllerError_COMMAN = 24 | _ControllerError , Trio.UnifiedApi.ControllerError_COMMAND_IS_RUNTIME_ONLY = 25 | _ControllerError , Trio.UnifiedApi.ControllerError_LABEL_EXPECTED = 26 | _ControllerError ,
  Trio.UnifiedApi.ControllerError_PROGRAM_NOT_FOUND = 27 | _ControllerError , Trio.UnifiedApi.ControllerError_DUPLICATE = 28 | _ControllerError , Trio.UnifiedApi.ControllerError_PROGRAM_IS_LOCKED = 29 | _ControllerError ,
  Trio.UnifiedApi.ControllerError_PROGRAMS_RUNNING = 30 | _ControllerError ,
  Trio.UnifiedApi.ControllerError_PROGRAM_IS_STOPPED = 31 | _ControllerError , Trio.UnifiedApi.ControllerError_CANNOT_S = 32 | _ControllerError , Trio.UnifiedApi.ControllerError_NO_PROGRAM_SELECTED = 33 | _ControllerError , Trio.UnifiedApi.ControllerError_NO_MORE_PROGRAMS_AVAILABLE = 34 | _ControllerError ,
  Trio.UnifiedApi.ControllerError_OUT_OF_MEMORY = 35 | _ControllerError , Trio.UnifiedApi.ControllerError_NO_CODE_AVAIL = 36 | _ControllerError , Trio.UnifiedApi.ControllerError_COMMAND_OUT_OF_CONTEXT = 37 | _ControllerError , Trio.UnifiedApi.ControllerError_TOO_MANY_NESTED_STRUCTURES = 38 | _ControllerError ,
  Trio.UnifiedApi.ControllerError_STRUCTURE_NESTING_ERROR = 39 | _ControllerError , Trio.UnifiedApi.ControllerError_ELSE = 40 | _ControllerError , Trio.UnifiedApi.ControllerError_WEND_WITHOUT_PREVIOUS_WHILE = 41 | _ControllerError , Trio.UnifiedApi.ControllerError_UNTIL_WITHOUT_PREVIOUS_REPEAT = 42 | _ControllerError ,
  Trio.UnifiedApi.ControllerError_IDENTIFIER_EXPECTED = 43 | _ControllerError , Trio.UnifiedApi.ControllerError_TO_EXPECT = 44 | _ControllerError , Trio.UnifiedApi.ControllerError_TOO_MANY_NESTED_FOR_NEXT = 45 | _ControllerError , Trio.UnifiedApi.ControllerError_NEXT_WITHOUT_FOR = 46 | _ControllerError ,
  Trio.UnifiedApi.ControllerError_UNTIL_IDLE_LOADED_EXPECTED_AFTER_WAIT = 47 | _ControllerError , Trio.UnifiedApi.ControllerError_GOTO_GOSUB_EXEC_EXPECTED = 48 | _ControllerError ,
  Trio.UnifiedApi.ControllerError_TOO_MANY_NESTED_GOSUB = 49 | _ControllerError , Trio.UnifiedApi.ControllerError_RETUR

= 50 | _ControllerError ,
Trio.UnifiedApi.ControllerError_LABEL_MUST_BE_AT_START_OF_LINE = 51 | _ControllerError ,
Trio.UnifiedApi.ControllerError_CANNOT_NEST_ONE_LINE_IF = 52 | _ControllerError , Trio.UnifiedApi.ControllerError_LABEL
= 53 | _ControllerError , Trio.UnifiedApi.ControllerError_LINE_NUMBER_CANNOT_HAVE_DECIMAL_POINT
= 54 | _ControllerError ,
Trio.UnifiedApi.ControllerError_MAXIMUM_NUMBER_OF_PROGRAM_INSTANCES_ACTIVE = 55 | _↩
ControllerError , Trio.UnifiedApi.ControllerError_INVALID_USE_OF_DOLAR_SIGN = 56 | _ControllerError
, Trio.UnifiedApi.ControllerError_VR_X_OR_STRING_VARIABLE_EXPECTED = 57 | _ControllerError ,
Trio.UnifiedApi.ControllerError_PROGRAM_ALREADY_EXISTS = 58 | _ControllerError ,
Trio.UnifiedApi.ControllerError_PROCESS_ALREADY_SELECTED = 59 | _ControllerError , Trio.UnifiedApi.ControllerError_DU
= 60 | _ControllerError , Trio.UnifiedApi.ControllerError_PLC_TYPE_IS_INVALID = 61 | _ControllerError ,
Trio.UnifiedApi.ControllerError_EVALUATION_ERROR = 62 | _ControllerError ,
Trio.UnifiedApi.ControllerError_RESERVED_KEYWORD_NOT_AVAILABLE_ON_THIS_CONTROLLER =
63 | _ControllerError , Trio.UnifiedApi.ControllerError_VARIABLE_NOT_FOUND = 64 | _Controller↩
Error , Trio.UnifiedApi.ControllerError_TABLE_INDEX_RANGE_ERROR = 65 | _ControllerError ,
Trio.UnifiedApi.ControllerError_FEATURES_ENABLED_DO_NOT_ALLOW_ATYPE_CHANGE = 66 | _↩
ControllerError ,
Trio.UnifiedApi.ControllerError_INVALID_LINE_NUMBER = 67 | _ControllerError , Trio.UnifiedApi.ControllerError_STRING_EX
= 68 | _ControllerError , Trio.UnifiedApi.ControllerError_SCOPE_PERIOD_SHOULD_EXCEED_NUMBER_OF_AIN_PARAMS
= 69 | _ControllerError , Trio.UnifiedApi.ControllerError_VALUE_IS_INCORRECT = 70 | _ControllerError ,
Trio.UnifiedApi.ControllerError_INVALID_IO_CHANNEL = 71 | _ControllerError , Trio.UnifiedApi.ControllerError_FLASH_VALUE
= 72 | _ControllerError , Trio.UnifiedApi.ControllerError_DIRECTORY_NOT_LOCKED = 73 | _ControllerError
, Trio.UnifiedApi.ControllerError_DIRECTORY_ALREADY_LOCKED = 74 | _ControllerError ,
Trio.UnifiedApi.ControllerError_PROGRAM_NOT_RUNNING_ON_THIS_PROCESS = 75 | _ControllerError ,
Trio.UnifiedApi.ControllerError_PROGRAM_NOT_RUNNING = 76 | _ControllerError , Trio.UnifiedApi.ControllerError_PROGRA
= 77 | _ControllerError , Trio.UnifiedApi.ControllerError_PROGRAM_NOT_PAUSED = 78 | _ControllerError ,
Trio.UnifiedApi.ControllerError_COMMAND_NOT_ALLOWED_WHEN_RUNNING_MOTION_PERFECT =
79 | _ControllerError , Trio.UnifiedApi.ControllerError_DIRECTORY_STRUCTURE_INVALID = 80 | ↩
_ControllerError , Trio.UnifiedApi.ControllerError_DIRECTORY_IS_LOCKED = 81 | _ControllerError ,
Trio.UnifiedApi.ControllerError_CANNOT_EDIT_PROGRAM = 82 | _ControllerError ,
Trio.UnifiedApi.ControllerError_TOO_MANY_NESTED_OPERANDS = 83 | _ControllerError , Trio.UnifiedApi.ControllerError_CA
= 84 | _ControllerError , Trio.UnifiedApi.ControllerError_FLASH_STICK_BLANK = 85 | _ControllerError
, Trio.UnifiedApi.ControllerError_FLASH_STICK_NOT_AVAILABLE_ON_THIS_CONTROLLER = 86 | _↩
ControllerError ,
Trio.UnifiedApi.ControllerError_SLAVE_ERROR = 87 | _ControllerError , Trio.UnifiedApi.ControllerError_MASTER_ERROR
= 88 | _ControllerError , Trio.UnifiedApi.ControllerError_NETWORK_TIMEOUT = 89 | _ControllerError ,
Trio.UnifiedApi.ControllerError_NETWORK_PROTOCOL_ERROR = 90 | _ControllerError ,
Trio.UnifiedApi.ControllerError_GLOBAL_DEFINITION_IS_DIFFERENT = 91 | _ControllerError , Trio.UnifiedApi.ControllerError_
= 92 | _ControllerError , Trio.UnifiedApi.ControllerError_PROGRAM_CORRUPT = 93 | _ControllerError ,
Trio.UnifiedApi.ControllerError_MORE_THAN_ONE_PROGRAM_RUNNING_WHEN_TRYING_TO_SET_GLOBAL_OR_CONS
= 94 | _ControllerError ,
Trio.UnifiedApi.ControllerError_PROGRAM_ENCRYPTED = 95 | _ControllerError , Trio.UnifiedApi.ControllerError_BASIC_TOK
= 96 | _ControllerError , Trio.UnifiedApi.ControllerError_OPEN_BRACKET_EXPECTED = 97 | _Controller↩
Error , Trio.UnifiedApi.ControllerError_NUMBER_EXPECTED = 98 | _ControllerError ,
Trio.UnifiedApi.ControllerError_AS_EXPECTED = 99 | _ControllerError , Trio.UnifiedApi.ControllerError_DATA_TYPE_EXPECT
= 100 | _ControllerError , Trio.UnifiedApi.ControllerError_STRING_EXPECTED = 101 | _ControllerError ,
Trio.UnifiedApi.ControllerError_DOWNLOAD_ABORT_OR_TIMEOUT = 102 | _ControllerError ,
Trio.UnifiedApi.ControllerError_CANNOT_SPECIFY_PROGRAM_TYPE_FOR_AN_EXISTING_PROGRAM
= 103 | _ControllerError , Trio.UnifiedApi.ControllerError_INVALID_COFF_IMAGE_FILE = 104 | _↩
ControllerError , Trio.UnifiedApi.ControllerError_VARIABLE_DEFINED_OUTSIDE_INCLUDE_FILE = 105
| _ControllerError , Trio.UnifiedApi.ControllerError_COMMAND_NOT_ALLOWED_WITHIN_INCLUDE_FILE
= 106 | _ControllerError ,
Trio.UnifiedApi.ControllerError_SERIAL_NUMBER_MUST_BE_MINUS_1 = 107 | _ControllerError ,
Trio.UnifiedApi.ControllerError_APPEND_BLOCK_INCONSISTENT = 108 | _ControllerError , Trio.UnifiedApi.ControllerError_IN
= 109 | _ControllerError , Trio.UnifiedApi.ControllerError_TOO_MANY_ITEMS_DEFINED_FOR_BLOCK =
110 | _ControllerError ,
Trio.UnifiedApi.ControllerError_INVALID_MSPHERICAL_INPUT = 111 | _ControllerError , Trio.UnifiedApi.ControllerError_TOO_
= 112 | _ControllerError , Trio.UnifiedApi.ControllerError_SYMBOL_TABLE_LOCKED = 113 | _Controller↩

Error , Trio.UnifiedApi.ControllerError_INCORRECT_SYMBOL_TYPE = 114 | _ControllerError , Trio.UnifiedApi.ControllerError_VARIABLES_NOT_PERMITTED_ON_COMMAND_LINE = 115 | _ControllerError , Trio.UnifiedApi.ControllerError_INVALID_PROGRAM_TYPE = 116 | _ControllerError , Trio.UnifiedApi.ControllerError_PARAMETER_EXPECTED = 117 | _ControllerError , Trio.UnifiedApi.ControllerError_FIRMWAR = 118 | _ControllerError , Trio.UnifiedApi.ControllerError_DEVICE_ERROR__TIMEOUT_WAITING_FOR_DEVICE = 119 | _ControllerError , Trio.UnifiedApi.ControllerError_DEVICE_ERROR__COMMAND_NOT_SUPPORTED_BY_DEVICE = 120 | _ControllerError , Trio.UnifiedApi.ControllerError_DEVICE_ERROR__CRC_ERROR = 121 | _ControllerError , Trio.UnifiedApi.ControllerError_DEVICE_ERROR__ERROR_WRITING_TO_DEVICE = 122 | _ControllerError , Trio.UnifiedApi.ControllerError_DEVICE_ERROR__INVALID_RESPONSE_FROM_DEVICE = 123 | _ControllerError , Trio.UnifiedApi.ControllerError_FIRMWARE_ERROR__CANNOT_REFERENCE_DATA_OUTSIDE_CURRENT = 124 | _ControllerError , Trio.UnifiedApi.ControllerError_DISK_ERROR__INVALID_MBR = 125 | _ControllerError , Trio.UnifiedApi.ControllerError_DISK_ERROR__INVALID_BOOT_SECTOR = 126 | _ControllerError , Trio.UnifiedApi.ControllerError_DISK_ERROR__INVALID_SECTOR_CLUSTER_REFERENCE = 127 | _ControllerError , Trio.UnifiedApi.ControllerError_FILE_ERROR__DISK_FULL = 128 | _ControllerError , Trio.UnifiedApi.ControllerError_FILE_ERROR__FILE_NOT_FOUND = 129 | _ControllerError , Trio.UnifiedApi.ControllerError_FILE_ERROR__FILENAME_ALREADY_EXISTS = 130 | _ControllerError , Trio.UnifiedApi.ControllerError_FILE_ERROR__INVALID_FILENAME = 131 | _ControllerError , Trio.UnifiedApi.ControllerError_ = 132 | _ControllerError , Trio.UnifiedApi.ControllerError_COMMAND_ONLY_ALLOWED_WHEN_RUNNING_MOTION_PERFE = 133 | _ControllerError , Trio.UnifiedApi.ControllerError_HASH_EXPECTED = 134 | _ControllerError , Trio.UnifiedApi.ControllerError_FOR_EXPECTED = 135 | _ControllerError , Trio.UnifiedApi.ControllerError_INPUT__OUTPUT_ = 136 | _ControllerError , Trio.UnifiedApi.ControllerError_FILE_NOT_OPEN = 137 | _ControllerError , Trio.UnifiedApi.ControllerError_END_OF_FILE = 138 | _ControllerError , Trio.UnifiedApi.ControllerError_FILE_ALREADY_OPEN = 139 | _ControllerError , Trio.UnifiedApi.ControllerError_INVALID_STO = 140 | _ControllerError , Trio.UnifiedApi.ControllerError_INVALID_FLOATING_POINT_OPERATION = 141 | _ControllerError , Trio.UnifiedApi.ControllerError_INVALID_SYSTEM_CODE_WRONG_CONTROLLER = 142 | _ControllerError , Trio.UnifiedApi.ControllerError_IEC_ERROR__INVALID_VARIABLE_ACCESS = 143 | _ControllerError , Trio.UnifiedApi.ControllerError_NUMERICAL_ERROR__NAN_USED = 144 | _ControllerError , Trio.UnifiedApi.ControllerError_NUMERICAL_ERROR__INF_USED = 145 | _ControllerError , Trio.UnifiedApi.ControllerError_NUMERICAL_ERROR__SUBNORMAL_VALUE_USED = 146 | _ControllerError , Trio.UnifiedApi.ControllerError_MAC_EEPROM_IS_LOCKED = 147 | _ControllerError , Trio.UnifiedApi.ControllerError_INVALID = 148 | _ControllerError , Trio.UnifiedApi.ControllerError_INVALID_STARTUP_CONFIGURATION_COMMAND = 149 | _ControllerError , Trio.UnifiedApi.ControllerError_SYMBOL_IS_NOT_A_VARIABLE = 150 | _ControllerError , Trio.UnifiedApi.ControllerError_KINEMATICS_RUNTIME_LIMITED_TO_1_HOUR_FEC_22 = 151 | _ControllerError , Trio.UnifiedApi.ControllerError_IEC_RUNTIME_LIMITED_TO_1_HOUR_FEC_21 = 152 | _ControllerError , Trio.UnifiedApi.ControllerError_COMMAND_NOT_ALLOWED_WITH_CURRENT_ATYPE = 153 | _ControllerError , Trio.UnifiedApi.ControllerError_WILDCARD_LENGTH_MUST_BE_1 = 154 | _ControllerError , Trio.UnifiedApi.ControllerError_INCOMPATIBLE_ARRAY_DIMENSIONS = 155 | _ControllerError , Trio.UnifiedApi.ControllerError_MATRIX_IS_SINGULAR = 156 | _ControllerError , Trio.UnifiedApi.ControllerError_PROGRAM_I = 157 | _ControllerError , Trio.UnifiedApi.ControllerError_DISK_ERROR__FORMAT_MUST_BE_FAT32 = 158 | _ControllerError , Trio.UnifiedApi.ControllerError_PROGRAM_IS_STOPPED_HALT_FORCED = 159 | _ControllerError , Trio.UnifiedApi.ControllerError_VARIABLE_NOT_DEFINED_USING_DIM_OR_GLOBAL_CONSTANT = 160 | _ControllerError , Trio.UnifiedApi.ControllerError_VARIABLE_NOT_DEFINED_USING_DIM_WITHIN_INCLUDE_FILE = 161 | _ControllerError , Trio.UnifiedApi.ControllerError_PARAMETER_CAN_ONLY_BE_CHANGED_USING_MC_CONFIG = 162 | _ControllerError , Trio.UnifiedApi.ControllerError_IEC_TASK_MEMORY_ALLOCATION_ERROR = 163 | _ControllerError , Trio.UnifiedApi.ControllerError_IEC_IO_BINDING_INDEX_OUT_OF_RANGE = 164 | _ControllerError , Trio.UnifiedApi.ControllerError_IEC_VR_BINDING_INDEX_OUT_OF_RANGE = 165 | _ControllerError , Trio.UnifiedApi.ControllerError_IEC_TABLE_BINDING_INDEX_OUT_OF_RANGE = 166 | _ControllerError , Trio.UnifiedApi.ControllerError_FAILED_OPEN_CONNECTION = 167 | _ControllerError , Trio.UnifiedApi.ControllerError_MCC

= 168 | _ControllerError , Trio.UnifiedApi.ControllerError_NO_RESPONSE_FROM_CAN_MODULE = 169 | _ControllerError , Trio.UnifiedApi.ControllerError_INVALID_FPGA_VERSION = 170 | _ControllerError , Trio.UnifiedApi.ControllerError_VARIABLE_NOT_CREATED_USING_GLOBAL_CONSTANT = 171 | _ControllerError , Trio.UnifiedApi.ControllerError_PARAMETER_CAN_ONLY_BE_CHANGED_USING_MC_CONFIG_OR_COM = 172 | _ControllerError , Trio.UnifiedApi.ControllerError_ETHERCAT_AXES_CANNOT_BE_SET_USING_ATYPE_PLEASE_U = 173 | _ControllerError , Trio.UnifiedApi.ControllerError_BIT_NUMBER_OUT_OF_RANGE = 174 | _ControllerError , Trio.UnifiedApi.ControllerError_FUNCTION_NAME_EXPECTED = 175 | _ControllerError , Trio.UnifiedApi.ControllerError_FUN = 176 | _ControllerError , Trio.UnifiedApi.ControllerError_TOO_MANY_FUNCTIONS_DEFINED = 177 | _ControllerError , Trio.UnifiedApi.ControllerError_MISSING_ENDFUNC = 178 | _ControllerError , Trio.UnifiedApi.ControllerError_ENDFUNC_WITHOUT_PREVIOUS_FUNCTION = 179 | _ControllerError , Trio.UnifiedApi.ControllerError_PARAMETER_NAME_EXPECTED = 180 | _ControllerError , Trio.UnifiedApi.ControllerError_DEVICE_ERROR_NO_SDCARD_INSERTED = 181 | _ControllerError , Trio.UnifiedApi.ControllerError_DEVICE_ERROR_SDCARD_WRITE_PROTECTED = 182 | _ControllerError , Trio.UnifiedApi.ControllerError_FAILED_ATTEMPT_1 = 183 | _ControllerError , Trio.UnifiedApi.ControllerError_FAILED_ATTEM = 184 | _ControllerError , Trio.UnifiedApi.ControllerError_FAILED_TOO_MANY_TIMES_PLEASE_RESET = 185 | _ControllerError , Trio.UnifiedApi.ControllerError_LINE_MUST_BE_INSIDE_FUNCTION = 186 | _ControllerError , Trio.UnifiedApi.ControllerError_DUPLICATE_NAME_USED_IN_LABEL_AND_VARIABLE = 187 | _ControllerError , Trio.UnifiedApi.ControllerError_STRUCTURE_FIELD_NAME_NOT_RECOGNISED = 188 | _ControllerError , Trio.UnifiedApi.ControllerError_ID_NOT_DEFINED = 189 | _ControllerError , Trio.UnifiedApi.ControllerError_ROBOT_CONFIGURATION_CANNOT_BE_CHANGED_WHILE_MOVEL = 190 | _ControllerError , Trio.UnifiedApi.ControllerError_TOOL_OFFSET_ID_NOT_DEFINED = 191 | _ControllerError , Trio.UnifiedApi.ControllerError_C = 192 | _ControllerError , Trio.UnifiedApi.ControllerError_ROBOT_FRAME_ID_NOT_DEFINED = 193 | _ControllerError , Trio.UnifiedApi.ControllerError_COMMAND_NOT_PERMITTED_IN_FUNCTION = 194 | _ControllerError , Trio.UnifiedApi.ControllerError_TOO_MANY_FUNCTION_PARAMETERS = 195 | _ControllerError , Trio.UnifiedApi.ControllerError_OPERATION_CANNOT_BE_PERFORMED_UNTIL_IDLE = 196 | _ControllerError , Trio.UnifiedApi.ControllerError_SINGULARITY = 197 | _ControllerError , Trio.UnifiedApi.ControllerError_TOOL_ = 198 | _ControllerError , Trio.UnifiedApi.ControllerError_OBJECT_FRAME_CANNOT_BE_MODIFIED_OR_DELETED_WHILE_SELECTED = 199 | _ControllerError , Trio.UnifiedApi.ControllerError_ROBOT_FRAME_CANNOT_BE_MODIFIED_OR_DELETED_WHILE_ = 200 | _ControllerError , Trio.UnifiedApi.ControllerError_ROBOT_MOVES_CANNOT_BE_PERFORMED_UNTIL_JOG_OPERA = 201 | _ControllerError , Trio.UnifiedApi.ControllerError_CYCLIC_INCLUDE_DEPENDENCY = 202 | _ControllerError , Trio.UnifiedApi.ControllerError_OBJECT_FRAME_ALREADY_ATTACHED = 203 | _ControllerError , Trio.UnifiedApi.ControllerError_OBJECT_FRAME_HAS_NOT_BEEN_ATTACHED = 204 | _ControllerError , Trio.UnifiedApi.ControllerError_OBJECT_FRAME_ALREADY_SYNCED = 205 | _ControllerError , Trio.UnifiedApi.ControllerError_BASE_AXIS_DOES_NOT_BELONG_TO_ANY_KINEMATIC_GROUP = 206 | _ControllerError , Trio.UnifiedApi.ControllerError_SYNC_IN_PROCESS = 207 | _ControllerError , Trio.UnifiedApi.ControllerError_ROBOT_PROG = 208 | _ControllerError , Trio.UnifiedApi.ControllerError_TARGET_POINT_NOT_ACTIVATED = 209 | _ControllerError , Trio.UnifiedApi.ControllerError_PROGRAM_FAILED_TO_RUN = 210 | _ControllerError , Trio.UnifiedApi.ControllerError_RETURN_STATEMENT_EXPECTED_BEFORE_ENDFUNC = 211 | _ControllerError , Trio.UnifiedApi.ControllerError_CONTINUE_WITHOUT_PREVIOUS_FOR_WHILE_OR_REPEAT = 212 | _ControllerError , Trio.UnifiedApi.ControllerError_EXIT_WITHOUT_PREVIOUS_FOR_WHILE_OR_REPEAT = 213 | _ControllerError , Trio.UnifiedApi.ControllerError_TOO_MANY_EXIT_STATEMENTS_WITHIN_SINGLE_BLOCK = 214 | _ControllerError , Trio.UnifiedApi.ControllerError_CASE_OR_END_CASE_WITHOUT_PREVIOUS_SELECT_CASE = 215 | _ControllerError , Trio.UnifiedApi.ControllerError_FIXED_INTEGER_OR_ELSE_EXPECTED = 216 | _ControllerError , Trio.UnifiedApi.ControllerError_TOO_MANY_CASE_OPTIONS = 217 | _ControllerError , Trio.UnifiedApi.ControllerError_ROBOTICS_RUNTIME_LIMITED_TO_1_HOUR = 218 | _ControllerError , Trio.UnifiedApi.ControllerError_COLLISION_OBJECTS_CANNOT_BE_MODIFIED_OR_DELETED_WHILE_SELECTED = 219 | _ControllerError , Trio.UnifiedApi.ControllerError_TOO_MANY_COLLISION_OBJECTS_ACTIVE = 220 | _ControllerError , Trio.UnifiedApi.ControllerError_ATYPE_MUST_BE_CONFIGURED_USING_MC_CONFIG_PARAMETE = 221 | _ControllerError , Trio.UnifiedApi.ControllerError_COMMAND_NOT_ALLOWED_IN_FRAME = 222 |

_ControllerError ,
Trio.UnifiedApi.ControllerError_COMMAND_LINE_TOO_LONG = 223 | _ControllerError , Trio.UnifiedApi.ControllerError_TABLE
= 224 | _ControllerError , Trio.UnifiedApi.ControllerError_ENCRYPTION_AUTHORITY_NOT_FOUND = 225
| _ControllerError , Trio.UnifiedApi.ControllerError_NO_FREE_ENCRYPTION_AUTHORITY = 226 | _↩
ControllerError ,
Trio.UnifiedApi.ControllerError_LOCAL_TARGET_VALUE_HAS_NOT_BEEN_DEFINED = 227 | _↩
ControllerError , Trio.UnifiedApi.ControllerError_COMMAND_NOT_PERMITTED_IN_BASIC_LIBRARY =
228 | _ControllerError , Trio.UnifiedApi.ControllerError_INVALID_MOVEC_INPUT = 229 | _ControllerError ,
Trio.UnifiedApi.ControllerError_FUNCTION_EXPECTED = 230 | _ControllerError ,
Trio.UnifiedApi.ControllerError_CAM_TABLE_ID_NOT_DEFINED = 231 | _ControllerError , Trio.UnifiedApi.ControllerError_RUN
= 232 | _ControllerError , Trio.UnifiedApi.ControllerError_U_AND_V_ORIENTATION_INHIBITED_WHILE_WRIST_SINGULARI
= 233 | _ControllerError , Trio.UnifiedApi.ControllerError_TARGET_POINT_OUT_OF_ROBOT_WORKING_AREA
= 234 | _ControllerError ,
Trio.UnifiedApi.ControllerError_AXIS_LOCKED_TO_A_ROBOT = 235 | _ControllerError , Trio.UnifiedApi.ControllerError_ROBO
= 236 | _ControllerError , Trio.UnifiedApi.ControllerError_ROBOT_NOT_DEFINED_USING_KINEMATIC_GROUP
= 237 | _ControllerError , Trio.UnifiedApi.ControllerError_COMMAND_INHIBITED_ROBOT_MUST_BE_USED_TO_ACTIVATE_
= 238 | _ControllerError ,
Trio.UnifiedApi.ControllerError_UNDEFINED_VARIABLE_IN_DIM_INITIALISER = 239 | _ControllerError ,
Trio.UnifiedApi.ControllerError_THE_SPECIFIED_ROBOT_CONFIGURATION_IS_NOT_ALLOWED = 240 |
_ControllerError , Trio.UnifiedApi.ControllerError_TOO_MANY_AXES_ACTIVATED = 241 | _ControllerError ,
Trio.UnifiedApi.ControllerError_MULTIPLE_MOVE_TRIGGERS_SPECIFIED = 242 | _ControllerError ,
Trio.UnifiedApi.ControllerError_MOVE_TRIGGER_RESPONSE_NOT_SPECIFIED = 243 | _Controller↩
Error , Trio.UnifiedApi.ControllerError_MOVE_TRIGGER_NOT_SPECIFIED = 244 | _ControllerError ,
Trio.UnifiedApi.ControllerError_INVALID_ARRAY_DIMENSION = 245 | _ControllerError , Trio.UnifiedApi.ControllerError_DUPLI
= 246 | _ControllerError ,
Trio.UnifiedApi.ControllerError_CONSTANT_VALUE_EXPECTED = 247 | _ControllerError , Trio.UnifiedApi.ControllerError_RPS
= 248 | _ControllerError , Trio.UnifiedApi.ControllerError_INCLUDE_FILE_MUST_CONTAIN_DEFCONST_DIM_STATEMENTS_
= 249 | _ControllerError , Trio.UnifiedApi.ControllerError_TARGET_TAUGHT_WITH_ANOTHER_ROBOT =
250 | _ControllerError ,
Trio.UnifiedApi.ControllerError_TARGET_ROBOT_CONFIG_MISMATCH = 251 | _ControllerError ,
Trio.UnifiedApi.ControllerError_TARGET_TOOL_OFFSET_MISMATCH = 252 | _ControllerError , Trio.UnifiedApi.ControllerError
= 253 | _ControllerError , Trio.UnifiedApi.ControllerError_TARGET_ROBOT_FRAME_MISMATCH = 254 |
_ControllerError ,
Trio.UnifiedApi.ControllerError_INVALID_FRAME_NUMBER = 255 | _ControllerError , Trio.UnifiedApi.ControllerError_FRAME_
= 256 | _ControllerError , Trio.UnifiedApi.ControllerError_FORWARD_TRANSFORM_FUNCTION_NOT_DEFINED
= 257 | _ControllerError , Trio.UnifiedApi.ControllerError_INVERSE_TRANSFORM_FUNCTION_NOT_DEFINED
= 258 | _ControllerError ,
Trio.UnifiedApi.ControllerError_INVALID_MOVEOPTION = 259 | _ControllerError , Trio.UnifiedApi.ControllerError_MAX_OPEN_
= 260 | _ControllerError , Trio.UnifiedApi.ControllerError_KEYWORD_INACTIVE = 261 | _ControllerError ,
Trio.UnifiedApi.ControllerError_KEYWORD_FUNCTION_NOT_DEFINED = 262 | _ControllerError ,
Trio.UnifiedApi.ControllerError_KEYWORD_SAME_AS_SYSTEM_WORD = 263 | _ControllerError ,
Trio.UnifiedApi.ControllerError_UNEXPECTED_CHARACTERS_AFTER_LINE_CONTINUATION = 264 |
_ControllerError , Trio.UnifiedApi.ControllerError_THIS_COMMAND_CAN_BE_EXECUTED_ONLY_ONCE =
265 | _ControllerError , Trio.UnifiedApi.ControllerError_TABLE_MEMORY_CONFLICT = 266 | _Controller↩
Error ,
Trio.UnifiedApi.ControllerError_MAX_OPEN_MOVES_REACHED = 267 | _ControllerError , Trio.UnifiedApi.ControllerError_USE
= 268 | _ControllerError , Trio.UnifiedApi.ControllerError_INVALID_FORMAT_SPECIFIER = 269 | _↩
ControllerError , Trio.UnifiedApi.ControllerError_OPEN_MOVE_FUNCTIONS_NOT_DEFINED = 270 | _↩
ControllerError ,
Trio.UnifiedApi.ControllerError_TOO_MANY_CHARS_IN_STR = 271 | _ControllerError , Trio.UnifiedApi.ControllerError_INVALI
= 272 | _ControllerError , Trio.UnifiedApi.ControllerError_PROCESS_IN_USE = 273 | _ControllerError ,
Trio.UnifiedApi.ControllerError_VR_LOCATION_WRITE_PROTECTED = 274 | _ControllerError }

Enumeration of error codes used in Unified API.

- enum Trio.UnifiedApi.EventType { Trio.UnifiedApi.Error , Trio.UnifiedApi.Warning , Trio.UnifiedApi.Message }

  Events that may occur on a connection.

- enum Trio.UnifiedApi.ConnectionType : UInt32 { Trio.UnifiedApi.Unknown = 0x00 , Trio.UnifiedApi.TCP = 0x01
, Trio.UnifiedApi.PCMCAT = 0x02 , Trio.UnifiedApi.FLEX7 = 0x04 }

  Connection types.

- enum Trio.UnifiedApi.ProgressEvent { Trio.UnifiedApi.Start , Trio.UnifiedApi.Progress , Trio.UnifiedApi.Finish }

  Progress event.
- enum Trio.UnifiedApi.ParamBlockType { Trio.UnifiedApi.AxisParam = 0 , Trio.UnifiedApi.SystemParameters = 1 , Trio.UnifiedApi.VR = 2 , Trio.UnifiedApi.TABLE = 3 , Trio.UnifiedApi.ProgramVariable = 4 }

  Types of connections available.
- enum Trio.UnifiedApi.Co_ObjectType { Trio.UnifiedApi.Boolean = 1 , Trio.UnifiedApi.Integer8 = 2 , Trio.UnifiedApi.Integer16 = 3 , Trio.UnifiedApi.Integer32 = 4 , Trio.UnifiedApi.Unsigned8 = 5 , Trio.UnifiedApi.Unsigned16 = 6 , Trio.UnifiedApi.Unsigned32 = 7 , Trio.UnifiedApi.Real32 = 8 , Trio.UnifiedApi.Real64 = 17 , Trio.UnifiedApi.Integer64 = 21 , Trio.UnifiedApi.Unsigned64 = 27 }

  Co Read/Write operation object types.
- enum Trio.UnifiedApi.EthercatState { Trio.UnifiedApi.Initial = 0 , Trio.UnifiedApi.PreOprational = 1 , Trio.UnifiedApi.SafeOperational = 2 , Trio.UnifiedApi.Operational = 3 }

  EtherCAT state.
- enum Trio.UnifiedApi.EthercatStartOption : UInt32 { Trio.UnifiedApi.None = 0x0000 , Trio.UnifiedApi.PrintOutProgress = 0x0001 , Trio.UnifiedApi.ForceRescan = 0x0002 , Trio.UnifiedApi.ExpressStartup = 0x0004 }

  EtherCAT start option.
- enum Trio.UnifiedApi.FileTransferOption : UInt32 { Trio.UnifiedApi.None = 0x0000 , Trio.UnifiedApi.CompressOnly = 0x0001 , Trio.UnifiedApi.CompressDecompress = 0x0002 }

  File transfer option.
- enum Trio.UnifiedApi.FileType : Int32 { Trio.UnifiedApi.Undefined = -1 , Trio.UnifiedApi.BASIC = 0 , Trio.UnifiedApi.EncryptedBASIC = 1 , Trio.UnifiedApi.GCode = 2 , Trio.UnifiedApi.TextFile = 3 , Trio.UnifiedApi.IecProgram = 4 , Trio.UnifiedApi.FifoFile = 5 , Trio.UnifiedApi.TempFile = 6 , Trio.UnifiedApi.Project = 7 , Trio.UnifiedApi.Binary = 8 , Trio.UnifiedApi.MCConfig = 9 , Trio.UnifiedApi.IecLibrary = 10 , Trio.UnifiedApi.HmiDesign = 11 , Trio.UnifiedApi.BASICLibrary = 12 , Trio.UnifiedApi.EncryptedBASICibrary = 13 , Trio.UnifiedApi.HmiLibrary = 14 , Trio.UnifiedApi.TempVolatileFile = 15 , Trio.UnifiedApi.Robot = 16 , Trio.UnifiedApi.RobotBasic = 17 , Trio.UnifiedApi.TempTextFile = 18 , Trio.UnifiedApi.RobotLibrary = 19 , Trio.UnifiedApi.RobotBasicLibrary = 20 , Trio.UnifiedApi.Recipe = 21 , Trio.UnifiedApi.Alarms = 22 , Trio.UnifiedApi.RealtimeBASICLibrary = 23 , Trio.UnifiedApi.EncryptedRealtimeBasicLibrary = 24 , Trio.UnifiedApi.EncryptedIecProgram = 25 , Trio.UnifiedApi.EncryptedIecLibrary = 26 , Trio.UnifiedApi.__Size }

  File types on the Motion Coordinator.
- enum Trio.UnifiedApi.FileFlags : UInt32 { Trio.UnifiedApi.None = 0x00 , Trio.UnifiedApi.Editable = 0x01 , Trio.UnifiedApi.Autorun = 0x02 }

  Flags related to files and programs in internal storage on the Motion Coordinator.
- enum Trio.UnifiedApi.ConnectMode : UInt32 { Trio.UnifiedApi.MPOS = 0 , Trio.UnifiedApi.DPOS = 1 }

  Mode for CONNECT command.
- enum Trio.UnifiedApi.PSwitchMode : UInt32 { Trio.UnifiedApi.None = 0x00 , Trio.UnifiedApi.Invert = 0x02 , Trio.UnifiedApi.DPOS = 0x04 , Trio.UnifiedApi.Mask = Invert | DPOS }

  Flags PSWITCH command in single/cyclic mode.
- enum Trio.UnifiedApi.PSwitchMultiMode : UInt32 { Trio.UnifiedApi.None = 0x00 , Trio.UnifiedApi.DPOS = 0x04 , Trio.UnifiedApi.TimerMode = 0x10 , Trio.UnifiedApi.NegativeDirection = 0x20 , Trio.UnifiedApi.ZOut = 0x40 , Trio.UnifiedApi.Mask = DPOS | TimerMode | NegativeDirection | ZOut }

  Flags PSWITCH command in multi/sequence mode.

- enum Trio.UnifiedApi.CamBoxLinkOptions : UInt32 {
  Trio.UnifiedApi.None = 0 , Trio.UnifiedApi.LinkMark = 1 , Trio.UnifiedApi.LinkAbsolute = 2 , Trio.UnifiedApi.LinkRepeat = 4 ,
  Trio.UnifiedApi.LinkPattern = 8 , Trio.UnifiedApi.LinkPositiveMove = 32 , Trio.UnifiedApi.LinkForceStart = 128 , Trio.UnifiedApi.LinkMarkB = 256 ,
  Trio.UnifiedApi.LinkR_Mark = 512 , Trio.UnifiedApi.LinkThreshold = 16384 }

  Link options in CamBox command.
- enum Trio.UnifiedApi.AxisType : UInt16 {
  Trio.UnifiedApi.None = 0 , Trio.UnifiedApi.Stepper = 1 , Trio.UnifiedApi.Servo = 2 , Trio.UnifiedApi.Encoder = 3 ,
  Trio.UnifiedApi.StepperEncoderInt = 4 , Trio.UnifiedApi.Resolver = 5 , Trio.UnifiedApi.VOut = 6 , Trio.UnifiedApi.SSI = 7 ,
  Trio.UnifiedApi.CAN = 8 , Trio.UnifiedApi.CAN_Remote = 9 , Trio.UnifiedApi.PSwitch = 10 , Trio.UnifiedApi.DLink_Remote = 11 ,
  Trio.UnifiedApi.Servo2 = 12 , Trio.UnifiedApi.Yaskawa = 13 , Trio.UnifiedApi.EncoderOut = 14 , Trio.UnifiedApi.HighestMotionType = EncoderOut ,
  Trio.UnifiedApi.Dummy15 = 15 , Trio.UnifiedApi.Dummy16 = 16 , Trio.UnifiedApi.Dummy17 = 17 , Trio.UnifiedApi.DS402PosType1 = 18 ,
  Trio.UnifiedApi.DS402VelType1 = 19 , Trio.UnifiedApi.First_Comms = 20 , Trio.UnifiedApi.CAN_Comms = First_Comms , Trio.UnifiedApi.USB_Comms = 21 ,
  Trio.UnifiedApi.DLink_Comms = 22 , Trio.UnifiedApi.Profibus_Comms = 23 , Trio.UnifiedApi.Dummy24 = 24 , Trio.UnifiedApi.Dummy25 = 25 ,
  Trio.UnifiedApi.DS402PosType2 = 26 , Trio.UnifiedApi.DS402VelType2 = 27 , Trio.UnifiedApi.Dummy28 = 28 , Trio.UnifiedApi.Dummy29 = 29 ,
  Trio.UnifiedApi.DACout_ANALin = 30 , Trio.UnifiedApi.ANALout_ANALin = 31 , Trio.UnifiedApi.Dummy32 = 32 , Trio.UnifiedApi.Dummy33 = 33 ,
  Trio.UnifiedApi.Dummy34 = 34 , Trio.UnifiedApi.SDout_PWMout = 35 , Trio.UnifiedApi.SDout_PWMout2 = 36 , Trio.UnifiedApi.PWMout_SDFREQin = 37 ,
  Trio.UnifiedApi.PWMout_SDin = 37 , Trio.UnifiedApi.Dummy38 = 38 , Trio.UnifiedApi.Dummy39 = 39 , Trio.UnifiedApi.MechatroPos = 40 ,
  Trio.UnifiedApi.MechatroSpeed = 41 , Trio.UnifiedApi.MechatroTorque = 42 , Trio.UnifiedApi.SDout_Zout = 43 , Trio.UnifiedApi.DACout_QUADin = 44 ,
  Trio.UnifiedApi.QUADout_Zout = 45 , Trio.UnifiedApi.DACout_TAMAin = 46 , Trio.UnifiedApi.DACout_ENDATin = 47 , Trio.UnifiedApi.DACout_SSIin = 48 ,
  Trio.UnifiedApi.MechatroInv = 49 , Trio.UnifiedApi.PanasonicPos = 50 , Trio.UnifiedApi.PanasonicSpeed = 51 , Trio.UnifiedApi.PanasonicTorque = 52 ,
  Trio.UnifiedApi.SercosSpeed = 53 , Trio.UnifiedApi.SercosPos = 54 , Trio.UnifiedApi.SercosTorque = 55 , Trio.UnifiedApi.SercosOpen = 56 ,
  Trio.UnifiedApi.SercosSpeedReg = 57 , Trio.UnifiedApi.SercosPosReg = 58 , Trio.UnifiedApi.SercosEncoder = 59 , Trio.UnifiedApi.DACout_SDin = 60 ,
  Trio.UnifiedApi.SLM32 = 61 , Trio.UnifiedApi.PLM32 = 62 , Trio.UnifiedApi.SDout_Zin = 63 , Trio.UnifiedApi.QUADout_Zin = 64 ,
  Trio.UnifiedApi.EthercatPos = 65 , Trio.UnifiedApi.EthercatSpeed = 66 , Trio.UnifiedApi.EthercatTorque = 67 , Trio.UnifiedApi.EthercatOpenSpeed = 68 ,
  Trio.UnifiedApi.EthercatEncoderIn = 69 , Trio.UnifiedApi.EthercatInAnalOut = 70 , Trio.UnifiedApi.EthercatPosTrio = 71 , Trio.UnifiedApi.EthercatSpeedTrio = 72 ,
  Trio.UnifiedApi.EthercatDummy5 = 73 , Trio.UnifiedApi.EthercatDummy6 = 74 , Trio.UnifiedApi.DACout_SSISLAVEin = 75 , Trio.UnifiedApi.QUADin = 76 ,
  Trio.UnifiedApi.DACout_QUADin_Zout = 77 , Trio.UnifiedApi.SDout_VFFDACout = 78 , Trio.UnifiedApi.SDin = 79 , Trio.UnifiedApi.G5Pos = 80 ,
  Trio.UnifiedApi.G5Speed = 81 , Trio.UnifiedApi.G5Torque = 82 , Trio.UnifiedApi.Jacquard = 83 , Trio.UnifiedApi.QUADout_VFFDACout = 84 ,
  Trio.UnifiedApi.AxesDiff = 85 , Trio.UnifiedApi.TAMAin = 86 , Trio.UnifiedApi.ENDATin = 87 , Trio.UnifiedApi.SSIin = 88 ,
  Trio.UnifiedApi.SDFREQout_ANALin = 89 , Trio.UnifiedApi.DACout_SDFREQin = 90 , Trio.UnifiedApi.AFFDACout = 91 , Trio.UnifiedApi.DACout_BISSin = 92 ,
  Trio.UnifiedApi.BISSin = 93 , Trio.UnifiedApi.DACPWMout_QUADin = 94 , Trio.UnifiedApi.SDout_TAMAin = 95 , Trio.UnifiedApi.Dummy96 = 96 ,

Trio.UnifiedApi.QUADin_Zout = 97 , Trio.UnifiedApi.Dummy98 = 98 , Trio.UnifiedApi.Dummy99 = 99 , Trio.UnifiedApi.SDout_HD = 100 , Trio.UnifiedApi.QUADout_HD = 101 , Trio.UnifiedApi.DACout_QUADin_HD = 102 , Trio.UnifiedApi.DACout_SDin_HD = 103 , Trio.UnifiedApi.QUADin_HD = 104 , Trio.UnifiedApi.SSIin_HD = 105 , Trio.UnifiedApi.TAMAin_HD = 106 , Trio.UnifiedApi.Dummy107 = 107 , Trio.UnifiedApi.Dummy108 = 108 , Trio.UnifiedApi.Dummy109 = 109 , Trio.UnifiedApi.SDout_SDin_HD = 110 , Trio.UnifiedApi.SDout_QUADin_HD = 111 , Trio.UnifiedApi.QUADout_SDin_HD = 112 , Trio.UnifiedApi.QUADout_QUADin_HD = 113 , Trio.UnifiedApi.SDout_TAMAin_HD = 114 , Trio.UnifiedApi.SDout_ENDATin_HD = 115 , Trio.UnifiedApi.SDout_SSIin_HD = 116 , Trio.UnifiedApi.SDout_BISSin_HD = 117 , Trio.UnifiedApi.QUADout_TAMAin_HD = 118 , Trio.UnifiedApi.QUADout_ENDATin_HD = 119 , Trio.UnifiedApi.QUADout_SSIin_HD = 120 , Trio.UnifiedApi.QUADout_BISSin_HD = 121 , Trio.UnifiedApi.Dummy122 = 122 , Trio.UnifiedApi.Dummy123 = 123 , Trio.UnifiedApi.Dummy124 = 124 , Trio.UnifiedApi.Dummy125 = 125 , Trio.UnifiedApi.Dummy126 = 126 , Trio.UnifiedApi.Dummy127 = 127 , Trio.UnifiedApi.Dummy128 = 128 , Trio.UnifiedApi.Dummy129 = 129 , Trio.UnifiedApi.SLMvirtual = 200 , Trio.UnifiedApi.PLMvirtual = 201 }

Enumeration of known axis types used with ATYPE axis parameter.

- enum Trio.UnifiedApi.AxisMode : UInt32 { Trio.UnifiedApi.NOT_USED_1 = ((UInt32)1 << 0) , Trio.UnifiedApi.ALLOW_CONNECT_MOVES_IN_LIMIT = ((UInt32)1 << 1) , Trio.UnifiedApi.DIRECTION_MODE_3D = ((UInt32)1 << 2) , Trio.UnifiedApi.CANCEL_ADDAX_ON_LIMIT = ((UInt32)1 << 3) , Trio.UnifiedApi.DISABLE_LOOKAHEAD = ((UInt32)1 << 4) , Trio.UnifiedApi.NOT_USED_2 = ((UInt32)1 << 5) , Trio.UnifiedApi.USE_NON_SIGNEXTENDED_ANALOGUE_FEEDBACK = ((UInt32)1 << 6) , Trio.UnifiedApi.NOT_USED_3 = ((UInt32)1 << 7) , Trio.UnifiedApi.SOFTWARE_LIMIT_USES_DPOS = ((UInt32)1 << 8) , Trio.UnifiedApi.DISABLE_FE_ACCUMULATION_DURI = ((UInt32)1 << 9) , Trio.UnifiedApi.USE_FASTDEC_WHEN_CANCELLING_RPS_JOG = ((UInt32)1 << 10) , Trio.UnifiedApi.DELAY_POSITION_EVALUATION_TO_LOAD_TIME = ((UInt32)1 << 11) , Trio.UnifiedApi.DIRECTION_SHORTEST = ((UInt32)1 << 12) , Trio.UnifiedApi.MONITOR_FE = ((UInt32)1 << 13) , Trio.UnifiedApi.OVERRIDE_FE_LIMITS = ((UInt32)1 << 14) , Trio.UnifiedApi.DUAL_AXIS_AVERAGE_POS = ((UInt32)1 << 15) , Trio.UnifiedApi.INVERT_AXIS_DIRECTION = ((UInt32)1 << 16) , Trio.UnifiedApi.CANCEL_MTYPE_ON_MASTER_IF_ADDAX = ((UInt32)1 << 17) }

  Enumeration of axis mode flags used with AXIS_MODE axis parameter.
- enum Trio.UnifiedApi.CornerMode : UInt32 { Trio.UnifiedApi.RESERVED = ((UInt32)1 << 0) , Trio.UnifiedApi.AUTO_SPEED_CONTROL = ((UInt32)1 << 1) , Trio.UnifiedApi.CORNER_STATE_MACHINE = ((UInt32)1 << 2) , Trio.UnifiedApi.AUTO_RADIUS_SPEED_CONTROL = ((UInt32)1 << 3) , Trio.UnifiedApi.AUTO_RADIUS_SPEED_CONTROL_CENTRIPETAL_MODE = ((UInt32)1 << 4) , Trio.UnifiedApi.AUTO_RADIUS_SPEED_CONTROL_CENTRIPETAL_MODE_USE_ACCEL = ((UInt32)1 << 5) , Trio.UnifiedApi.AUTO_BLEND_MODE_0_USING_CORNER_RADIUS = ((UInt32)1 << 6) , Trio.UnifiedApi.AUTO_BLEND_MODE_1_USING_CORNER_RADIUS = ((UInt32)1 << 7) }

  Enumeration of corner mode flags used with CORNER_MODE axis parameter.
- enum Trio.UnifiedApi.CornerState : UInt32 { Trio.UnifiedApi.HELD = ((UInt32)1 << 0) , Trio.UnifiedApi.LOAD = ((UInt32)1 << 1) , Trio.UnifiedApi.LOAD_ZERO_SPEED = ((UInt32)1 << 2) }

  Enumeration of corner state flags used with CORNER_STATE axis parameter.
- enum Trio.UnifiedApi.MotionType : UInt32 { Trio.UnifiedApi.IDLE = 0 , Trio.UnifiedApi.MOVE = 1 , Trio.UnifiedApi.MOVEABS = 2 , Trio.UnifiedApi.MHELICAL = 3 , Trio.UnifiedApi.MOVECIRC = 4 , Trio.UnifiedApi.MOVEMODIFY = 5 , Trio.UnifiedApi.MOVESP = 6 , Trio.UnifiedApi.MOVEABSSP = 7 , Trio.UnifiedApi.MOVECIRCSP = 8 , Trio.UnifiedApi.MHELICALSP = 9 , Trio.UnifiedApi.FORWARD = 10 , Trio.UnifiedApi.REVERSE = 11 , Trio.UnifiedApi.DATUM = 12 , Trio.UnifiedApi.CAM = 13 , Trio.UnifiedApi.FORWARD_JOG = 14 , Trio.UnifiedApi.REVERSE_JOG = 15 , Trio.UnifiedApi.WETCAM = 16 , Trio.UnifiedApi.CAMBOX = 20 , Trio.UnifiedApi.CONNECT = 21 , Trio.UnifiedApi.MOVELINK = 22 ,

Trio.UnifiedApi.CONNPATH = 23 , Trio.UnifiedApi.FLEXLINK = 24 , Trio.UnifiedApi.CAMLINK = 25 , Trio.UnifiedApi.COMPENSATE_XY = 26 ,
Trio.UnifiedApi.PICKLINK = 27 , Trio.UnifiedApi.CAMIN = 28 , Trio.UnifiedApi.MOVETANG = 30 , Trio.UnifiedApi.MSPHERICAL = 31 ,
Trio.UnifiedApi.MTRANSCIRC = 32 , Trio.UnifiedApi.MOVESEQ = 33 , Trio.UnifiedApi.MOVEABSSEQ = 34 , Trio.UnifiedApi.MOVETURN = 35 ,
Trio.UnifiedApi.MOVEPICK = 36 , Trio.UnifiedApi.MOVECONTOUR = 37 , Trio.UnifiedApi.MOVEDISPENSE = 38 , Trio.UnifiedApi.MSPHERICALSP = 40 ,
Trio.UnifiedApi.MTRANSCIRCSP = 41 , Trio.UnifiedApi.MOVESEQSP = 42 , Trio.UnifiedApi.MOVEABSSEQSP = 43 , Trio.UnifiedApi.MOVEJ = 44 ,
Trio.UnifiedApi.MOVEL = 45 , Trio.UnifiedApi.MOVEC = 46 , Trio.UnifiedApi.MOVEBLEND = 47 , Trio.UnifiedApi.MOVEJREL = 48 ,
Trio.UnifiedApi.MOVELREL = 49 , Trio.UnifiedApi.MOVETEST = 50 , Trio.UnifiedApi.MOVEP = 51 , Trio.UnifiedApi.MOVEU = 52 ,
Trio.UnifiedApi.MOVEB = 53 , Trio.UnifiedApi.MOVESPCONT = 60 , Trio.UnifiedApi.MOVEABSSPCONT = 61 , Trio.UnifiedApi.GEARIN = 62 ,
Trio.UnifiedApi.ACCPROFILE = 70 , Trio.UnifiedApi.VELPROFILE = 71 , Trio.UnifiedApi.POSPROFILE = 72 , Trio.UnifiedApi.COMBINEAXES = 73 ,
Trio.UnifiedApi.OPENMOVE_BASE = 100 , Trio.UnifiedApi.OPENMOVE_1 = OPENMOVE_BASE , Trio.UnifiedApi.OPENMOVE_2 = OPENMOVE_BASE + 1 , Trio.UnifiedApi.OPENMOVE_3 = OPENMOVE↩
_BASE + 2 ,
Trio.UnifiedApi.OPENMOVE_4 = OPENMOVE_BASE + 3 , Trio.UnifiedApi.OPENMOVE_5 = OPENMOVE↩
_BASE + 4 }

Motion types returned by MTYPE and NTYPE axis parameters.

- enum Trio.UnifiedApi.AxisStatus : UInt64 {
Trio.UnifiedApi.AS_LOOKAHEAD_OVERRIDE = ((UInt64)1 << 0) , Trio.UnifiedApi.AS_FE_WARNING = ((UInt64)1 << 1) , Trio.UnifiedApi.AS_COMMS_ERROR = ((UInt64)1 << 2) , Trio.UnifiedApi.AS_REMOTE_DRIVE_ERROR = ((UInt64)1 << 3) ,
Trio.UnifiedApi.AS_IN_FORWARD_LIMIT = ((UInt64)1 << 4) , Trio.UnifiedApi.AS_IN_REVERSE_LIMIT = ((UInt64)1 << 5) , Trio.UnifiedApi.AS_DATUMING = ((UInt64)1 << 6) , Trio.UnifiedApi.AS_FEEDHOLD = ((UInt64)1 << 7) ,
Trio.UnifiedApi.AS_FE_EXCEEDS_LIMIT = ((UInt64)1 << 8) , Trio.UnifiedApi.AS_FORWARD_SOFTWARE_LIMIT = ((UInt64)1 << 9) , Trio.UnifiedApi.AS_REVERSE_SOFTWARE_LIMIT = ((UInt64)1 << 10) ,
Trio.UnifiedApi.AS_CANCEL_CURRENT = ((UInt64)1 << 11) ,
Trio.UnifiedApi.AS_PULSE_OVER_SPEED = ((UInt64)1 << 12) , Trio.UnifiedApi.AS_OVERRIDE_TO_ZERO = ((UInt64)1 << 13) , Trio.UnifiedApi.AS_MOVE_CANCEL_DONE = ((UInt64)1 << 14) , Trio.UnifiedApi.AS_VOLUME_LIMIT = ((UInt64)1 << 15) ,
Trio.UnifiedApi.AS_AXIS_FS_LIMIT = ((UInt64)1 << 16) , Trio.UnifiedApi.AS_AXIS_RS_LIMIT = ((UInt64)1 << 17) , Trio.UnifiedApi.AS_ENCODER_OVER_I = ((UInt64)1 << 18) , Trio.UnifiedApi.AS_PSWITCH_FIFO_NOT_EMPTY = ((UInt64)1 << 19) ,
Trio.UnifiedApi.AS_PSWITCH_FIFO_FULL = ((UInt64)1 << 20) , Trio.UnifiedApi.AS_KINEMATICS_1HOUR = ((UInt64)1 << 21) , Trio.UnifiedApi.AS_BISS_CLEAN_WARNING = ((UInt64)1 << 22) , Trio.UnifiedApi.AS_WORLD_FS_LIM = ((UInt64)1 << 23) ,
Trio.UnifiedApi.AS_WORLD_RS_LIMIT = ((UInt64)1 << 24) , Trio.UnifiedApi.AS_CANCEL_CURRENT_AT_SPEED = ((UInt64)1 << 25) , Trio.UnifiedApi.AS_REMOTE_NODE_STATUS_ERROR = ((UInt64)1 << 26) ,
Trio.UnifiedApi.AS_PSWITCH_FIFO_HALF_FULL = ((UInt64)1 << 27) ,
Trio.UnifiedApi.AS_MOVEADD = ((UInt64)1 << 28) , Trio.UnifiedApi.AS_CANCEL_MOVEADD = ((UInt64)1 << 29) , Trio.UnifiedApi.AS_REMOTE_NODE_ERROR2 = ((UInt64)1 << 30) , Trio.UnifiedApi.AS_SINGULARITY = ((UInt64)1 << 31) ,
Trio.UnifiedApi.AS_OPEN_FRAME_ERROR = ((UInt64)1 << 32) , Trio.UnifiedApi.AS_VP_ERROR = ((UInt64)1 << 33) , Trio.UnifiedApi.AS_CANCEL_BUFFERED = ((UInt64)1 << 34) , Trio.UnifiedApi.AS_CANCEL_BUFFERED = ((UInt64)1 << 35) ,
Trio.UnifiedApi.AS_CANCEL_BUFFERED_DECEL_WITHIN_CURRENT_MOVE = ((UInt64)1 << 36) ,
Trio.UnifiedApi.AS_CANCEL_LAST_BUFFERED = ((UInt64)1 << 37) }
- enum Trio.UnifiedApi.RobotStatus : UInt64 {
Trio.UnifiedApi.RS_WORLD_FS_LIMIT = ((UInt64)1 << 0) , Trio.UnifiedApi.RS_WORLD_RS_LIMIT = ((UInt64)1 << 1) , Trio.UnifiedApi.RS_ROBOT_FS_LIMIT = ((UInt64)1 << 2) , Trio.UnifiedApi.RS_ROBOT_RS_LIMIT = ((UInt64)1 << 3) ,

Trio.UnifiedApi.RS_TCP_FS_LIMIT = ((UInt64)1 << 4) , Trio.UnifiedApi.RS_TCP_RS_LIMIT = ((UInt64)1 << 5) , Trio.UnifiedApi.RS_FE_WARNING = ((UInt64)1 << 6) , Trio.UnifiedApi.RS_FE_EXCEEDS_LIMIT = ((UInt64)1 << 7) ,
Trio.UnifiedApi.RS_WRIST_SINGULARITY = ((UInt64)1 << 8) , Trio.UnifiedApi.RS_ALIGNMENT_SINGULARITY = ((UInt64)1 << 9) , Trio.UnifiedApi.RS_ELBOW_SINGULARITY = ((UInt64)1 << 10) , Trio.UnifiedApi.RS_MAX_SPEED_LIMI = ((UInt64)1 << 11) ,
Trio.UnifiedApi.RS_COLLIDED = ((UInt64)1 << 12) , Trio.UnifiedApi.RS_NOT_IN_POS = ((UInt64)1 << 13) , Trio.UnifiedApi.RS_ELBOW_CLOSED_SINGULARITY = ((UInt64)1 << 14) , Trio.UnifiedApi.RS_DISALLOW_JOG = ((UInt64)1 << 15) }

- enum Trio.UnifiedApi.ProcessStatus : UInt32 {
  Trio.UnifiedApi.STOPPED , Trio.UnifiedApi.READY , Trio.UnifiedApi.STEPPING , Trio.UnifiedApi.PAUSED , Trio.UnifiedApi.PAUSING , Trio.UnifiedApi.ABORTED , Trio.UnifiedApi.SUSPEND , Trio.UnifiedApi.SUSPENDED ,
  Trio.UnifiedApi.SLEEPING }
- enum Trio.UnifiedApi.SyncControl : Int32 {
  Trio.UnifiedApi.SC_WAITING_TO_START = -1 , Trio.UnifiedApi.SC_NO_SYNC = 0 , Trio.UnifiedApi.SC_STARTING_SYNC = 1 , Trio.UnifiedApi.SC_PERFORMING_SYNC = 2 ,
  Trio.UnifiedApi.SC_SYNCHRONISED = 3 , Trio.UnifiedApi.SC_STOPPING_SYNC = 4 , Trio.UnifiedApi.SC_STARTING_INTERP = 5 , Trio.UnifiedApi.SC_PERFORMING_INTERP_MOVE = 6 ,
  Trio.UnifiedApi.SC_STARTING_RESYNC = 10 , Trio.UnifiedApi.SC_PERFORMING_RESYNC = 11 ,
  Trio.UnifiedApi.SC_STARTING_RESYNC_TO_USERFRAME = 20 , Trio.UnifiedApi.SC_PERFORMING_RESYNC_TO_USERF = 21 }

Functions

- delegate void Trio.UnifiedApi.ConnectionEventHandler (EventType eventType, long integerData, string stringData)

  Delegate for connection related events.
- delegate void Trio.UnifiedApi.FileCallback (ref FileInfo file, string fileName)

  Delegate for processing the response of Dir command.
- unsafe delegate void Trio.UnifiedApi.StreamHandlerCallback (ref DataStreamPacket packet, byte *buff)

  Delegate for streaming back the response of commands that return string data.
- delegate void Trio.UnifiedApi.ProgressHandlerCallback (ref ProgressInfo progress)

  Delegate for streaming back the response of commands that return string data.

7.1.1 Typedef Documentation

7.1.1.1 size_t

```
using size_t = System.UInt64
```

Definition at line 4 of file ITrioConnection.cs.

7.2 ITrioConnection.cs

Go to the documentation of this file.

```
00001 using System;
00002 using System.Runtime.InteropServices;
00003 using System.Threading.Tasks;
00004 using size_t = System.UInt64;
00005
00006 namespace Trio.UnifiedApi
00007 {
```

```
00008     #region Enumerations
00009
00010
      ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
00011
00012     /// <summary>
00013     /// Enumeration of error codes used in Unified API
00014     /// </summary>
00015     public enum ErrorCode : UInt64
00016     {
00017         /// <summary>
00018         /// No error
00019         /// </summary>
00020         NoError = 0,
00021
00022         /// <summary>
00023         /// An error code that cannot be determined
00024         /// </summary>
00025         _Unknown = 0xFFFFFFFF,
00026
00027         /// <summary>
00028         /// UnifiedAPI error flag. The actual error code is in the lowest 32 bits.
00029         /// </summary>
00030         _ApiError = 0x100000000,
00031
00032         /// <summary>
00033         /// An UnifiedAPI error that cannot be determined
00034         /// </summary>
00035         Api_Unknown = (_ApiError + 0),
00036
00037         /// <summary>
00038         /// General error
00039         /// </summary>
00040         Fail,
00041
00042         /// <summary>
00043         /// Error indicating that operation is not supported
00044         /// </summary>
00045         NotSupported,
00046
00047         /// <summary>
00048         /// Requested connection type (module) cannot be load/initialized or is missing
00049         /// </summary>
00050         ConnectionModule,
00051
00052         /// <summary>
00053         /// A bad connection context is used
00054         /// </summary>
00055         ConnectionContext,
00056
00057         /// <summary>
00058         /// TOKENTABLE reading failed
00059         /// </summary>
00060         TokenTable,
00061
00062         /// <summary>
00063         /// The operation timed out
00064         /// </summary>
00065         Timeout,
00066
00067         /// <summary>
00068         /// The data is of unknown floating point format
00069         /// </summary>
00070         UnknownFloatingPointFormat,
00071
00072         /// <summary>
00073         /// Error in the token buffer for the Remote protocol
00074         /// </summary>
00075         Remote_TokenBuffer,
00076
00077         /// <summary>
00078         /// No such token (Remote protocol)
00079         /// </summary>
00080         Remote_UnknownToken,
00081
00082         /// <summary>
00083         /// Unsupported controller
00084         /// </summary>
00085         UnsupportedController,
00086
00087         /// <summary>
00088         /// Bad response/unrecognized received
00089         /// </summary>
00090         BadResponse,
00091
00092         /// <summary>
00093         /// A bad argument has been passed
```

```
00094         /// </summary>
00095         BadArgument,
00096
00097         /// <summary>
00098         /// The connection or object is already opened
00099         /// </summary>
00100         Reopen,
00101
00102         /// <summary>
00103         /// The passed buffer is too small
00104         /// </summary>
00105         BufferTooSmall,
00106
00107         /// <summary>
00108         /// The read to the socket failed
00109         /// </summary>
00110         SocketRead,
00111
00112         /// <summary>
00113         /// The write to the socket failed
00114         /// </summary>
00115         SocketWrite,
00116
00117         /// <summary>
00118         /// Low level API call failed
00119         /// </summary>
00120         LowLevelError,
00121
00122         /// <summary>
00123         /// Versions of modules do not match
00124         /// </summary>
00125         VersionMismatch,
00126
00127         /// <summary>
00128         /// Compress/Decompress data error
00129         /// </summary>
00130         Compression,
00131
00132         /// <summary>
00133         /// The operation got aborted.
00134         /// </summary>
00135         Abort,
00136
00137         /// <summary>
00138         /// No connection
00139         /// </summary>
00140         Disconnected,
00141
00142         /// <summary>
00143         /// The last API error
00144         /// </summary>
00145         _ApiLast,
00146
00147         /// <summary>
00148         /// A system error flag. The actual system error code is in the lowest 32 bits.
00149         /// </summary>
00150         _SystemError = 0x200000000,
00151
00152         /// <summary>
00153         /// Unknown system error.
00154         /// </summary>
00155         Sys_Unknown = (_SystemError | _Unknown),
00156
00157         /// <summary>
00158         /// A controller error flag. The actual controller error code is in the lowest 32 bits.
00159         /// </summary>
00160         _ControllerError = 0x400000000,
00161
00162         /// <summary>
00163         /// Controller Error codes
00164         /// </summary>
00165         ControllerError_NONE = 0 | _ControllerError,
00166
00167         ControllerError_COMMAND_NOT_RECOGNISED = 1 | _ControllerError,
00168
00169         ControllerError_INVALID_TRANSFER_TYPE = 2 | _ControllerError,
00170
00171         ControllerError_FLASH_PROGRAMMING_ERROR = 3 | _ControllerError,
00172
00173         ControllerError_OPERAND_EXPECTED = 4 | _ControllerError,
00174
00175         ControllerError_ASSIGNMENT_EXPECTED = 5 | _ControllerError,
00176
00177         ControllerError_QUOTES_EXPECTED = 6 | _ControllerError,
00178
00179         ControllerError_STACK_OVERFLOW = 7 | _ControllerError,
00180
```

```
00181         ControllerError_TOO_MANY_VARIABLES = 8 | _ControllerError,
00182
00183         ControllerError_DIVISION_BY_ZERO = 9 | _ControllerError,
00184
00185         ControllerError_EXTRA_CHARS_AT_EOLN = 10 | _ControllerError,
00186
00187         ControllerError_CLOSE_SQUARE_BRACKET_EXPECTED_IN_PRINT = 11 | _ControllerError,
00188
00189         ControllerError_CANNOT_MODIOFY_SPECIAL_PROGRAM = 12 | _ControllerError,
00190
00191         ControllerError_THEN_EXPECTED_IN_IF_ELSEIF = 13 | _ControllerError,
00192
00193         ControllerError_FLASH_ERASE_ERROR = 14 | _ControllerError,
00194
00195         ControllerError_START_OF_EXPRESSION_EXPECTED = 15 | _ControllerError,
00196
00197         ControllerError_CLOSE_ROUND_BRACKET_EXPECTED = 16 | _ControllerError,
00198
00199         ControllerError_COMMA_EXPECTED = 17 | _ControllerError,
00200
00201         ControllerError_COMMAND_LINE_BROKEN_BY_ESC = 18 | _ControllerError,
00202
00203         ControllerError_PARAMETER_OUT_OF_RANGE = 19 | _ControllerError,
00204
00205         ControllerError_NO_PROCESS_AVAILABLE = 20 | _ControllerError,
00206
00207         ControllerError_VALUE_IS_READ_ONLY = 21 | _ControllerError,
00208
00209         ControllerError_MODIFIER_NOT_ALLOWED = 22 | _ControllerError,
00210
00211         ControllerError_REMOTE_AXIS_IS_IN_USE = 23 | _ControllerError,
00212
00213         ControllerError_COMMAND_IS_COMMAND_LINE_ONLY = 24 | _ControllerError,
00214
00215         ControllerError_COMMAND_IS_RUNTIME_ONLY = 25 | _ControllerError,
00216
00217         ControllerError_LABEL_EXPECTED = 26 | _ControllerError,
00218
00219         ControllerError_PROGRAM_NOT_FOUND = 27 | _ControllerError,
00220
00221         ControllerError_DUPLICATE_IDENTIFIER = 28 | _ControllerError,
00222
00223         ControllerError_PROGRAM_IS_LOCKED = 29 | _ControllerError,
00224
00225         ControllerError_PROGRAMS_RUNNING = 30 | _ControllerError,
00226
00227         ControllerError_PROGRAM_IS_STOPPED = 31 | _ControllerError,
00228
00229         ControllerError_CANNOT_SELECT_PROGRAM = 32 | _ControllerError,
00230
00231         ControllerError_NO_PROGRAM_SELECTED = 33 | _ControllerError,
00232
00233         ControllerError_NO_MORE_PROGRAMS_AVAILABLE = 34 | _ControllerError,
00234
00235         ControllerError_OUT_OF_MEMORY = 35 | _ControllerError,
00236
00237         ControllerError_NO_CODE_AVAILABLE_FOR_PROGRAM = 36 | _ControllerError,
00238
00239         ControllerError_COMMAND_OUT_OF_CONTEXT = 37 | _ControllerError,
00240
00241         ControllerError_TOO_MANY_NESTED_STRUCTURES = 38 | _ControllerError,
00242
00243         ControllerError_STRUCTURE_NESTING_ERROR = 39 | _ControllerError,
00244
00245         ControllerError_ELSE_ELSEIF_ENDIF_WITHOUT_PREVIOUS_IF = 40 | _ControllerError,
00246
00247         ControllerError_WEND_WITHOUT_PREVIOUS_WHILE = 41 | _ControllerError,
00248
00249         ControllerError_UNTIL_WITHOUT_PREVIOUS_REPEAT = 42 | _ControllerError,
00250
00251         ControllerError_IDENTIFIER_EXPECTED = 43 | _ControllerError,
00252
00253         ControllerError_TO_EXPECTED_AFTER_FOR = 44 | _ControllerError,
00254
00255         ControllerError_TOO_MANY_NESTED_FOR_NEXT = 45 | _ControllerError,
00256
00257         ControllerError_NEXT_WITHOUT_FOR = 46 | _ControllerError,
00258
00259         ControllerError_UNTIL_IDLE_LOADED_EXPECTED_AFTER_WAIT = 47 | _ControllerError,
00260
00261         ControllerError_GOTO_GOSUB_EXEC_EXPECTED = 48 | _ControllerError,
00262
00263         ControllerError_TOO_MANY_NESTED_GOSUB = 49 | _ControllerError,
00264
00265         ControllerError_RETURN_WITHOUT_GOSUB = 50 | _ControllerError,
00266
00267         ControllerError_LABEL_MUST_BE_AT_START_OF_LINE = 51 | _ControllerError,
```

```
00268
00269         ControllerError_CANNOT_NEST_ONE_LINE_IF = 52 | _ControllerError,
00270
00271         ControllerError_LABEL_NOT_FOUND = 53 | _ControllerError,
00272
00273         ControllerError_LINE_NUMBER_CANNOT_HAVE_DECIMAL_POINT = 54 | _ControllerError,
00274
00275         ControllerError_MAXIMUM_NUMBER_OF_PROGRAM_INSTANCES_ACTIVE = 55 | _ControllerError,
00276
00277         ControllerError_INVALID_USE_OF_DOLAR_SIGN = 56 | _ControllerError,
00278
00279         ControllerError_VR_X_OR_STRING_VARIABLE_EXPECTED = 57 | _ControllerError,
00280
00281         ControllerError_PROGRAM_ALREADY_EXISTS = 58 | _ControllerError,
00282
00283         ControllerError_PROCESS_ALREADY_SELECTED = 59 | _ControllerError,
00284
00285         ControllerError_DUPLICATE_AXES_NOT_PERMITTED = 60 | _ControllerError,
00286
00287         ControllerError_PLC_TYPE_IS_INVALID = 61 | _ControllerError,
00288
00289         ControllerError_EVALUATION_ERROR = 62 | _ControllerError,
00290
00291         ControllerError_RESERVED_KEYWORD_NOT_AVAILABLE_ON_THIS_CONTROLLER = 63 | _ControllerError,
00292
00293         ControllerError_VARIABLE_NOT_FOUND = 64 | _ControllerError,
00294
00295         ControllerError_TABLE_INDEX_RANGE_ERROR = 65 | _ControllerError,
00296
00297         ControllerError_FEATURES_ENABLED_DO_NOT_ALLOW_ATYPE_CHANGE = 66 | _ControllerError,
00298
00299         ControllerError_INVALID_LINE_NUMBER = 67 | _ControllerError,
00300
00301         ControllerError_STRING_EXCEEDS_PERMITTED_LENGTH = 68 | _ControllerError,
00302
00303         ControllerError_SCOPE_PERIOD_SHOULD_EXCEED_NUMBER_OF_AIN_PARAMS = 69 | _ControllerError,
00304
00305         ControllerError_VALUE_IS_INCORRECT = 70 | _ControllerError,
00306
00307         ControllerError_INVALID_IO_CHANNEL = 71 | _ControllerError,
00308
00309         ControllerError_FLASH_VALUE_CANNOT_BE_SET = 72 | _ControllerError,
00310
00311         ControllerError_DIRECTORY_NOT_LOCKED = 73 | _ControllerError,
00312
00313         ControllerError_DIRECTORY_ALREADY_LOCKED = 74 | _ControllerError,
00314
00315         ControllerError_PROGRAM_NOT_RUNNING_ON_THIS_PROCESS = 75 | _ControllerError,
00316
00317         ControllerError_PROGRAM_NOT_RUNNING = 76 | _ControllerError,
00318
00319         ControllerError_PROGRAM_NOT_PAUSED_ON_THIS_PROCESS = 77 | _ControllerError,
00320
00321         ControllerError_PROGRAM_NOT_PAUSED = 78 | _ControllerError,
00322
00323         ControllerError_COMMAND_NOT_ALLOWED_WHEN_RUNNING_MOTION_PERFECT = 79 | _ControllerError,
00324
00325         ControllerError_DIRECTORY_STRUCTURE_INVALID = 80 | _ControllerError,
00326
00327         ControllerError_DIRECTORY_IS_LOCKED = 81 | _ControllerError,
00328
00329         ControllerError_CANNOT_EDIT_PROGRAM = 82 | _ControllerError,
00330
00331         ControllerError_TOO_MANY_NESTED_OPERANDS = 83 | _ControllerError,
00332
00333         ControllerError_CANNOT_RESET_WHEN_DRIVE_SERVO_ON = 84 | _ControllerError,
00334
00335         ControllerError_FLASH_STICK_BLANK = 85 | _ControllerError,
00336
00337         ControllerError_FLASH_STICK_NOT_AVAILABLE_ON_THIS_CONTROLLER = 86 | _ControllerError,
00338
00339         ControllerError_SLAVE_ERROR = 87 | _ControllerError,
00340
00341         ControllerError_MASTER_ERROR = 88 | _ControllerError,
00342
00343         ControllerError_NETWORK_TIMEOUT = 89 | _ControllerError,
00344
00345         ControllerError_NETWORK_PROTOCOL_ERROR = 90 | _ControllerError,
00346
00347         ControllerError_GLOBAL_DEFINITION_IS_DIFFERENT = 91 | _ControllerError,
00348
00349         ControllerError_INVALID_PROGRAM_NAME = 92 | _ControllerError,
00350
00351         ControllerError_PROGRAM_CORRUPT = 93 | _ControllerError,
00352
00353         ControllerError_MORE_THAN_ONE_PROGRAM_RUNNING_WHEN_TRYING_TO_SET_GLOBAL_OR_CONSTANT = 94 |
      _ControllerError,
```

```
00354
00355         ControllerError_PROGRAM_ENCRYPTED = 95 | _ControllerError,
00356
00357         ControllerError_BASIC_TOKEN_DEFINITION_INCORRECT = 96 | _ControllerError,
00358
00359         ControllerError_OPEN_BRACKET_EXPECTED = 97 | _ControllerError,
00360
00361         ControllerError_NUMBER_EXPECTED = 98 | _ControllerError,
00362
00363         ControllerError_AS_EXPECTED = 99 | _ControllerError,
00364
00365         ControllerError_DATA_TYPE_EXPECTED = 100 | _ControllerError,
00366
00367         ControllerError_STRING_EXPECTED = 101 | _ControllerError,
00368
00369         ControllerError_DOWNLOAD_ABORT_OR_TIMEOUT = 102 | _ControllerError,
00370
00371         ControllerError_CANNOT_SPECIFY_PROGRAM_TYPE_FOR_AN_EXISTING_PROGRAM = 103 | _ControllerError,
00372
00373         ControllerError_INVALID_COFF_IMAGE_FILE = 104 | _ControllerError,
00374
00375         ControllerError_VARIABLE_DEFINED_OUTSIDE_INCLUDE_FILE = 105 | _ControllerError,
00376
00377         ControllerError_COMMAND_NOT_ALLOWED_WITHIN_INCLUDE_FILE = 106 | _ControllerError,
00378
00379         ControllerError_SERIAL_NUMBER_MUST_BE_MINUS_1 = 107 | _ControllerError,
00380
00381         ControllerError_APPEND_BLOCK_INCONSISTENT = 108 | _ControllerError,
00382
00383         ControllerError_INVALID_RANGE_SPECIFIED = 109 | _ControllerError,
00384
00385         ControllerError_TOO_MANY_ITEMS_DEFINED_FOR_BLOCK = 110 | _ControllerError,
00386
00387         ControllerError_INVALID_MSPHERICAL_INPUT = 111 | _ControllerError,
00388
00389         ControllerError_TOO_MANY_LABELS = 112 | _ControllerError,
00390
00391         ControllerError_SYMBOL_TABLE_LOCKED = 113 | _ControllerError,
00392
00393         ControllerError_INCORRECT_SYMBOL_TYPE = 114 | _ControllerError,
00394
00395         ControllerError_VARIABLES_NOT_PERMITTED_ON_COMMAND_LINE = 115 | _ControllerError,
00396
00397         ControllerError_INVALID_PROGRAM_TYPE = 116 | _ControllerError,
00398
00399         ControllerError_PARAMETER_EXPECTED = 117 | _ControllerError,
00400
00401         ControllerError_FIRMWARE_ERROR__DEVICE_IN_USE = 118 | _ControllerError,
00402
00403         ControllerError_DEVICE_ERROR__TIMEOUT_WAITING_FOR_DEVICE = 119 | _ControllerError,
00404
00405         ControllerError_DEVICE_ERROR__COMMAND_NOT_SUPPORTED_BY_DEVICE = 120 | _ControllerError,
00406
00407         ControllerError_DEVICE_ERROR__CRC_ERROR = 121 | _ControllerError,
00408
00409         ControllerError_DEVICE_ERROR__ERROR_WRITING_TO_DEVICE = 122 | _ControllerError,
00410
00411         ControllerError_DEVICE_ERROR__INVALID_RESPONSE_FROM_DEVICE = 123 | _ControllerError,
00412
00413         ControllerError_FIRMWARE_ERROR__CANNOT_REFERENCE_DATA_OUTSIDE_CURRENT_BLOCK = 124 |
    _ControllerError,
00414
00415         ControllerError_DISK_ERROR__INVALID_MBR = 125 | _ControllerError,
00416
00417         ControllerError_DISK_ERROR__INVALID_BOOT_SECTOR = 126 | _ControllerError,
00418
00419         ControllerError_DISK_ERROR__INVALID_SECTOR_CLUSTER_REFERENCE = 127 | _ControllerError,
00420
00421         ControllerError_FILE_ERROR__DISK_FULL = 128 | _ControllerError,
00422
00423         ControllerError_FILE_ERROR__FILE_NOT_FOUND = 129 | _ControllerError,
00424
00425         ControllerError_FILE_ERROR__FILENAME_ALREADY_EXISTS = 130 | _ControllerError,
00426
00427         ControllerError_FILE_ERROR__INVALID_FILENAME = 131 | _ControllerError,
00428
00429         ControllerError_FILE_ERROR__DIRECTORY_FULL = 132 | _ControllerError,
00430
00431         ControllerError_COMMAND_ONLY_ALLOWED_WHEN_RUNNING_MOTION_PERFECT = 133 | _ControllerError,
00432
00433         ControllerError_HASH_EXPECTED = 134 | _ControllerError,
00434
00435         ControllerError_FOR_EXPECTED = 135 | _ControllerError,
00436
00437         ControllerError_INPUT__OUTPUT__APPEND__FIFO_READ__FIFO_WRITE_EXPECTED = 136 |
    _ControllerError,
00438
```

```
00439         ControllerError_FILE_NOT_OPEN = 137 | _ControllerError,
00440
00441         ControllerError_END_OF_FILE = 138 | _ControllerError,
00442
00443         ControllerError_FILE_ALREADY_OPEN = 139 | _ControllerError,
00444
00445         ControllerError_INVALID_STORAGE_AREA = 140 | _ControllerError,
00446
00447         ControllerError_INVALID_FLOATING_POINT_OPERATION = 141 | _ControllerError,
00448
00449         ControllerError_INVALID_SYSTEM_CODE_WRONG_CONTROLLER = 142 | _ControllerError,
00450
00451         ControllerError_IEC_ERROR__INVALID_VARIABLE_ACCESS = 143 | _ControllerError,
00452
00453         ControllerError_NUMERICAL_ERROR__NAN_USED = 144 | _ControllerError,
00454
00455         ControllerError_NUMERICAL_ERROR__INF_USED = 145 | _ControllerError,
00456
00457         ControllerError_NUMERICAL_ERROR__SUBNORMAL_VALUE_USED = 146 | _ControllerError,
00458
00459         ControllerError_MAC_EEPROM_IS_LOCKED = 147 | _ControllerError,
00460
00461         ControllerError_INVALID_MIX_OF_DATA_TYPES = 148 | _ControllerError,
00462
00463         ControllerError_INVALID_STARTUP_CONFIGURATION_COMMAND = 149 | _ControllerError,
00464
00465         ControllerError_SYMBOL_IS_NOT_A_VARIABLE = 150 | _ControllerError,
00466
00467         ControllerError_KINEMATICS_RUNTIME_LIMITED_TO_1_HOUR_FEC_22 = 151 | _ControllerError,
00468
00469         ControllerError_IEC_RUNTIME_LIMITED_TO_1_HOUR_FEC_21 = 152 | _ControllerError,
00470
00471         ControllerError_COMMAND_NOT_ALLOWED_WITH_CURRENT_ATYPE = 153 | _ControllerError,
00472
00473         ControllerError_WILDCARD_LENGTH_MUST_BE_1 = 154 | _ControllerError,
00474
00475         ControllerError_INCOMPATIBLE_ARRAY_DIMENSIONS = 155 | _ControllerError,
00476
00477         ControllerError_MATRIX_IS_SINGULAR = 156 | _ControllerError,
00478
00479         ControllerError_PROGRAM_IS_NOT_EXECUTABLE_TYPE = 157 | _ControllerError,
00480
00481         ControllerError_DISK_ERROR__FORMAT_MUST_BE_FAT32 = 158 | _ControllerError,
00482
00483         ControllerError_PROGRAM_IS_STOPPED_HALT_FORCED = 159 | _ControllerError,
00484
00485         ControllerError_VARIABLE_NOT_DEFINED_USING_DIM_OR_GLOBAL_CONSTANT = 160 | _ControllerError,
00486
00487         ControllerError_VARIABLE_NOT_DEFINED_USING_DIM_WITHIN_INCLUDE_FILE = 161 | _ControllerError,
00488
00489         ControllerError_PARAMETER_CAN_ONLY_BE_CHANGED_USING_MC_CONFIG = 162 | _ControllerError,
00490
00491         ControllerError_IEC_TASK_MEMORY_ALLOCATION_ERROR = 163 | _ControllerError,
00492
00493         ControllerError_IEC_IO_BINDING_INDEX_OUT_OF_RANGE = 164 | _ControllerError,
00494
00495         ControllerError_IEC_VR_BINDING_INDEX_OUT_OF_RANGE = 165 | _ControllerError,
00496
00497         ControllerError_IEC_TABLE_BINDING_INDEX_OUT_OF_RANGE = 166 | _ControllerError,
00498
00499         ControllerError_FAILED_OPEN_CONNECTION = 167 | _ControllerError,
00500
00501         ControllerError_MCCONFIG_CHANGED_EX_REQUIRED = 168 | _ControllerError,
00502
00503         ControllerError_NO_RESPONSE_FROM_CAN_MODULE = 169 | _ControllerError,
00504
00505         ControllerError_INVALID_FPGA_VERSION = 170 | _ControllerError,
00506
00507         ControllerError_VARIABLE_NOT_CREATED_USING_GLOBAL_CONSTANT = 171 | _ControllerError,
00508
00509         ControllerError_PARAMETER_CAN_ONLY_BE_CHANGED_USING_MC_CONFIG_OR_COMMAND_LINE = 172 |
      _ControllerError,
00510
00511         ControllerError_ETHERCAT_AXES_CANNOT_BE_SET_USING_ATYPE_PLEASE_USE_DRIVE_MODE = 173 |
      _ControllerError,
00512
00513         ControllerError_BIT_NUMBER_OUT_OF_RANGE = 174 | _ControllerError,
00514
00515         ControllerError_FUNCTION_NAME_EXPECTED = 175 | _ControllerError,
00516
00517         ControllerError_FUNCTION_MUST_BE_DEFINED_WITHIN_A_BASIC_LIBRARY = 176 | _ControllerError,
00518
00519         ControllerError_TOO_MANY_FUNCTIONS_DEFINED = 177 | _ControllerError,
00520
00521         ControllerError_MISSING_ENDFUNC = 178 | _ControllerError,
00522
00523         ControllerError_ENDFUNC_WITHOUT_PREVIOUS_FUNCTION = 179 | _ControllerError,
```

```
00524
00525         ControllerError_PARAMETER_NAME_EXPECTED = 180 | _ControllerError,
00526
00527         ControllerError_DEVICE_ERROR_NO_SDCARD_INSERTED = 181 | _ControllerError,
00528
00529         ControllerError_DEVICE_ERROR_SDCARD_WRITE_PROTECTED = 182 | _ControllerError,
00530
00531         ControllerError_FAILED_ATTEMPT_1 = 183 | _ControllerError,
00532
00533         ControllerError_FAILED_ATTEMPT_2 = 184 | _ControllerError,
00534
00535         ControllerError_FAILED_TOO_MANY_TIMES_PLEASE_RESET = 185 | _ControllerError,
00536
00537         ControllerError_LINE_MUST_BE_INSIDE_FUNCTION = 186 | _ControllerError,
00538
00539         ControllerError_DUPLICATE_NAME_USED_IN_LABEL_AND_VARIABLE = 187 | _ControllerError,
00540
00541         ControllerError_STRUCTURE_FIELD_NAME_NOT_RECOGNISED = 188 | _ControllerError,
00542
00543         ControllerError_ID_NOT_DEFINED = 189 | _ControllerError,
00544
00545         ControllerError_ROBOT_CONFIGURATION_CANNOT_BE_CHANGED_WHILE_MOVEL = 190 | _ControllerError,
00546
00547         ControllerError_TOOL_OFFSET_ID_NOT_DEFINED = 191 | _ControllerError,
00548
00549         ControllerError_OBJECT_FRAME_ID_NOT_DEFINED = 192 | _ControllerError,
00550
00551         ControllerError_ROBOT_FRAME_ID_NOT_DEFINED = 193 | _ControllerError,
00552
00553         ControllerError_COMMAND_NOT_PERMITTED_IN_FUNCTION = 194 | _ControllerError,
00554
00555         ControllerError_TOO_MANY_FUNCTION_PARAMETERS = 195 | _ControllerError,
00556
00557         ControllerError_OPERATION_CANNOT_BE_PERFORMED_UNTIL_IDLE = 196 | _ControllerError,
00558
00559         ControllerError_SINGULARITY = 197 | _ControllerError,
00560
00561         ControllerError_TOOL_CANNOT_BE_MODIFIED_OR_DELETED_WHILE_SELECTED = 198 | _ControllerError,
00562
00563         ControllerError_OBJECT_FRAME_CANNOT_BE_MODIFIED_OR_DELETED_WHILE_SELECTED = 199 |
      _ControllerError,
00564
00565         ControllerError_ROBOT_FRAME_CANNOT_BE_MODIFIED_OR_DELETED_WHILE_SELECTED = 200 |
      _ControllerError,
00566
00567         ControllerError_ROBOT_MOVES_CANNOT_BE_PERFORMED_UNTIL_JOG_OPERATION_IS_DISABLED = 201 |
      _ControllerError,
00568
00569         ControllerError_CYCLIC_INCLUDE_DEPENDENCY = 202 | _ControllerError,
00570
00571         ControllerError_OBJECT_FRAME_ALREADY_ATTACHED = 203 | _ControllerError,
00572
00573         ControllerError_OBJECT_FRAME_HAS_NOT_BEEN_ATTACHED = 204 | _ControllerError,
00574
00575         ControllerError_OBJECT_FRAME_ALREADY_SYNCED = 205 | _ControllerError,
00576
00577         ControllerError_BASE_AXIS_DOES_NOT_BELONG_TO_ANY_KINEMATIC_GROUP = 206 | _ControllerError,
00578
00579         ControllerError_SYNC_IN_PROCESS = 207 | _ControllerError,
00580
00581         ControllerError_ROBOT_PROGRAM_STOPPED = 208 | _ControllerError,
00582
00583         ControllerError_TARGET_POINT_NOT_ACTIVATED = 209 | _ControllerError,
00584
00585         ControllerError_PROGRAM_FAILED_TO_RUN = 210 | _ControllerError,
00586
00587         ControllerError_RETURN_STATEMENT_EXPECTED_BEFORE_ENDFUNC = 211 | _ControllerError,
00588
00589         ControllerError_CONTINUE_WITHOUT_PREVIOUS_FOR_WHILE_OR_REPEAT = 212 | _ControllerError,
00590
00591         ControllerError_EXIT_WITHOUT_PREVIOUS_FOR_WHILE_OR_REPEAT = 213 | _ControllerError,
00592
00593         ControllerError_TOO_MANY_EXIT_STATEMENTS_WITHIN_SINGLE_BLOCK = 214 | _ControllerError,
00594
00595         ControllerError_CASE_OR_END_CASE_WITHOUT_PREVIOUS_SELECT_CASE = 215 | _ControllerError,
00596
00597         ControllerError_FIXED_INTEGER_OR_ELSE_EXPECTED = 216 | _ControllerError,
00598
00599         ControllerError_TOO_MANY_CASE_OPTIONS = 217 | _ControllerError,
00600
00601         ControllerError_ROBOTICS_RUNTIME_LIMITED_TO_1_HOUR = 218 | _ControllerError,
00602
00603         ControllerError_COLLISION_OBJECTS_CANNOT_BE_MODIFIED_OR_DELETED_WHILE_SELECTED = 219 |
      _ControllerError,
00604
00605         ControllerError_TOO_MANY_COLLISION_OBJECTS_ACTIVE = 220 | _ControllerError,
00606
```

```
00607         ControllerError_ATYPE_MUST_BE_CONFIGURED_USING_MC_CONFIG_PARAMETER_DRIVE_MODE = 221 |
      _ControllerError,
00608
00609         ControllerError_COMMAND_NOT_ALLOWED_IN_FRAME = 222 | _ControllerError,
00610
00611         ControllerError_COMMAND_LINE_TOO_LONG = 223 | _ControllerError,
00612
00613         ControllerError_TABLE_LOCATION_WRITE_PROTECTED = 224 | _ControllerError,
00614
00615         ControllerError_ENCRYPTION_AUTHORITY_NOT_FOUND = 225 | _ControllerError,
00616
00617         ControllerError_NO_FREE_ENCRYPTION_AUTHORITY = 226 | _ControllerError,
00618
00619         ControllerError_LOCAL_TARGET_VALUE_HAS_NOT_BEEN_DEFINED = 227 | _ControllerError,
00620
00621         ControllerError_COMMAND_NOT_PERMITTED_IN_BASIC_LIBRARY = 228 | _ControllerError,
00622
00623         ControllerError_INVALID_MOVEC_INPUT = 229 | _ControllerError,
00624
00625         ControllerError_FUNCTION_EXPECTED = 230 | _ControllerError,
00626
00627         ControllerError_CAM_TABLE_ID_NOT_DEFINED = 231 | _ControllerError,
00628
00629         ControllerError_RUNNING_OF_RPS_PROGRAMS_INHIBITED_USE_DISABLE_ROBOT = 232 | _ControllerError,
00630
00631         ControllerError_U_AND_V_ORIENTATION_INHIBITED_WHILE_WRIST_SINGULARITY = 233 |
      _ControllerError,
00632
00633         ControllerError_TARGET_POINT_OUT_OF_ROBOT_WORKING_AREA = 234 | _ControllerError,
00634
00635         ControllerError_AXIS_LOCKED_TO_A_ROBOT = 235 | _ControllerError,
00636
00637         ControllerError_ROBOT_CONTROLLED_BY_ANOTHER_PROCESS = 236 | _ControllerError,
00638
00639         ControllerError_ROBOT_NOT_DEFINED_USING_KINEMATIC_GROUP = 237 | _ControllerError,
00640
00641         ControllerError_COMMAND_INHIBITED_ROBOT_MUST_BE_USED_TO_ACTIVATE_A_KINEMATIC_GROUP = 238 |
      _ControllerError,
00642
00643         ControllerError_UNDEFINED_VARIABLE_IN_DIM_INITIALISER = 239 | _ControllerError,
00644
00645         ControllerError_THE_SPECIFIED_ROBOT_CONFIGURATION_IS_NOT_ALLOWED = 240 | _ControllerError,
00646
00647         ControllerError_TOO_MANY_AXES_ACTIVATED = 241 | _ControllerError,
00648
00649         ControllerError_MULTIPLE_MOVE_TRIGGERS_SPECIFIED = 242 | _ControllerError,
00650
00651         ControllerError_MOVE_TRIGGER_RESPONSE_NOT_SPECIFIED = 243 | _ControllerError,
00652
00653         ControllerError_MOVE_TRIGGER_NOT_SPECIFIED = 244 | _ControllerError,
00654
00655         ControllerError_INVALID_ARRAY_DIMENSION = 245 | _ControllerError,
00656
00657         ControllerError_DUPLICATE_VR_INDEX = 246 | _ControllerError,
00658
00659         ControllerError_CONSTANT_VALUE_EXPECTED = 247 | _ControllerError,
00660
00661         ControllerError_RPS_KERNEL_V2_NOT_ENABLED = 248 | _ControllerError,
00662
00663         ControllerError_INCLUDE_FILE_MUST_CONTAIN_DEFCONST_DIM_STATEMENTS_ONLY = 249 |
      _ControllerError,
00664
00665         ControllerError_TARGET_TAUGHT_WITH_ANOTHER_ROBOT = 250 | _ControllerError,
00666
00667         ControllerError_TARGET_ROBOT_CONFIG_MISMATCH = 251 | _ControllerError,
00668
00669         ControllerError_TARGET_TOOL_OFFSET_MISMATCH = 252 | _ControllerError,
00670
00671         ControllerError_TARGET_OBJECT_FRAME_MISMATCH = 253 | _ControllerError,
00672
00673         ControllerError_TARGET_ROBOT_FRAME_MISMATCH = 254 | _ControllerError,
00674
00675         ControllerError_INVALID_FRAME_NUMBER = 255 | _ControllerError,
00676
00677         ControllerError_FRAME_NOT_SUPPORTED_BY_RPS = 256 | _ControllerError,
00678
00679         ControllerError_FORWARD_TRANSFORM_FUNCTION_NOT_DEFINED = 257 | _ControllerError,
00680
00681         ControllerError_INVERSE_TRANSFORM_FUNCTION_NOT_DEFINED = 258 | _ControllerError,
00682
00683         ControllerError_INVALID_MOVEOPTION = 259 | _ControllerError,
00684
00685         ControllerError_MAX_OPEN_KEYWORDS_REACHED = 260 | _ControllerError,
00686
00687         ControllerError_KEYWORD_INACTIVE = 261 | _ControllerError,
00688
00689         ControllerError_KEYWORD_FUNCTION_NOT_DEFINED = 262 | _ControllerError,
```

```
00690
00691         ControllerError_KEYWORD_SAME_AS_SYSTEM_WORD = 263 | _ControllerError,
00692
00693         ControllerError_UNEXPECTED_CHARACTERS_AFTER_LINE_CONTINUATION = 264 | _ControllerError,
00694
00695         ControllerError_THIS_COMMAND_CAN_BE_EXECUTED_ONLY_ONCE = 265 | _ControllerError,
00696
00697         ControllerError_TABLE_MEMORY_CONFLICT = 266 | _ControllerError,
00698
00699         ControllerError_MAX_OPEN_MOVES_REACHED = 267 | _ControllerError,
00700
00701         ControllerError_USER_DEFINED_ERROR = 268 | _ControllerError,
00702
00703         ControllerError_INVALID_FORMAT_SPECIFIER = 269 | _ControllerError,
00704
00705         ControllerError_OPEN_MOVE_FUNCTIONS_NOT_DEFINED = 270 | _ControllerError,
00706
00707         ControllerError_TOO_MANY_CHARS_IN_STR = 271 | _ControllerError,
00708
00709         ControllerError_INVALID_OPEN_ARCH_FUNC_DEFINITION = 272 | _ControllerError,
00710
00711         ControllerError_PROCESS_IN_USE = 273 | _ControllerError,
00712
00713         ControllerError_VR_LOCATION_WRITE_PROTECTED = 274 | _ControllerError,
00714     };
00715
00716     /// <summary>
00717     /// Events that may occur on a connection.
00718     /// </summary>
00719     public enum EventType
00720     {
00721         /// <summary>
00722         /// Error processing a command.
00723         /// </summary>
00724         Error,
00725
00726         /// <summary>
00727         /// Warning for non critical problem
00728         /// </summary>
00729         Warning,
00730
00731         /// <summary>
00732         /// Information about the command being processed.
00733         /// </summary>
00734         Message,
00735     };
00736
00737     /// <summary>
00738     /// Connection types.
00739     /// </summary>
00740     [Flags]
00741     public enum ConnectionType : UInt32
00742     {
00743         /// <summary>
00744         /// Unknown connection type. Returned when there is no connection context
00745         /// </summary>
00746         Unknown = 0x00,
00747
00748         /// <summary>
00749         /// TCP connection via Remote protocol
00750         /// </summary>
00751         TCP = 0x01,
00752
00753         /// <summary>
00754         /// Shared memory connection for PC-MCAT devices
00755         /// </summary>
00756         PCMCAT = 0x02,
00757
00758         /// <summary>
00759         /// Shared memory connection for Flex7 devices
00760         /// </summary>
00761         FLEX7 = 0x04,
00762     };
00763
00764     /// <summary>
00765     /// Progress event
00766     /// </summary>
00767     public enum ProgressEvent
00768     {
00769         /// <summary>
00770         /// Progress started.
00771         /// </summary>
00772         Start,
00773
00774         /// <summary>
00775         /// Progress.
00776         /// </summary>
```

```
00777         Progress,
00778
00779         /// <summary>
00780         /// Progress finished.
00781         /// </summary>
00782         Finish,
00783     };
00784
00785     /// <summary>
00786     /// Types of connections available.
00787     /// </summary>
00788     public enum ParamBlockType
00789     {
00790         /// <summary>
00791         /// Axis parameters.
00792         /// </summary>
00793         AxisParam = 0,
00794
00795         /// <summary>
00796         /// System parameters.
00797         /// </summary>
00798         SystemParameters = 1,
00799
00800         /// <summary>
00801         /// VR values.
00802         /// </summary>
00803         VR = 2,
00804
00805         /// <summary>
00806         /// TABLE values.
00807         /// </summary>
00808         TABLE = 3,
00809
00810         /// <summary>
00811         /// Program variables.
00812         /// </summary>
00813         ProgramVariable = 4,
00814     };
00815
00816     /// <summary>
00817     /// Co Read/Write operation object types
00818     /// </summary>
00819     public enum Co_ObjectType
00820     {
00821         /// <summary>
00822         /// Boolean type
00823         /// </summary>
00824         Boolean = 1,
00825
00826         /// <summary>
00827         /// 8 bit integer
00828         /// </summary>
00829         Integer8 = 2,
00830
00831         /// <summary>
00832         /// 16 bit integer
00833         /// </summary>
00834         Integer16 = 3,
00835
00836         /// <summary>
00837         /// 32 bit integer
00838         /// </summary>
00839         Integer32 = 4,
00840
00841         /// <summary>
00842         /// 8 bit unsigned integer
00843         /// </summary>
00844         Unsigned8 = 5,
00845
00846         /// <summary>
00847         /// 16 bit unsigned integer
00848         /// </summary>
00849         Unsigned16 = 6,
00850
00851         /// <summary>
00852         /// 32 bit unsigned integer
00853         /// </summary>
00854         Unsigned32 = 7,
00855
00856         /// <summary>
00857         /// 32 bit floating point
00858         /// </summary>
00859         Real32 = 8,
00860
00861         /// <summary>
00862         /// 64 bit floating point
00863         /// </summary>
```

```
00864         Real64 = 17,
00865
00866         /// <summary>
00867         /// 64 bit integer
00868         /// </summary>
00869         Integer64 = 21,
00870
00871         /// <summary>
00872         /// 64 bit unsigned integer
00873         /// </summary>
00874         Unsigned64 = 27,
00875     };
00876
00877     /// <summary>
00878     /// EtherCAT state
00879     /// </summary>
00880     public enum EthercatState
00881     {
00882         /// <summary>
00883         /// Initial
00884         /// </summary>
00885         Initial = 0,
00886
00887         /// <summary>
00888         /// Pre operational
00889         /// </summary>
00890         PreOprational = 1,
00891
00892         /// <summary>
00893         /// Safe operational
00894         /// </summary>
00895         SafeOperational = 2,
00896
00897         /// <summary>
00898         /// Operational
00899         /// </summary>
00900         Operational = 3,
00901     };
00902
00903     /// <summary>
00904     /// EtherCAT start option
00905     /// </summary>
00906     [Flags]
00907     public enum EthercatStartOption : UInt32
00908     {
00909         /// <summary>
00910         /// No option is selected
00911         /// </summary>
00912         None = 0x0000,
00913
00914         /// <summary>
00915         /// Print out progress information
00916         ///  on - Print out progress information (default)
00917         ///  off - Do not print progress information
00918         /// </summary>
00919         PrintOutProgress = 0x0001,
00920
00921         /// <summary>
00922         /// Network re-scan
00923         ///  off - Normal state change (default)
00924         ///  on - Force a re-scan of the network
00925         /// </summary>
00926         ForceRescan = 0x0002,
00927
00928         /// <summary>
00929         /// Start-up type
00930         ///  on - Express start-up; do not wait for PLL to lock between state changes
00931         ///  off - Standard start-up
00932         /// </summary>
00933         ExpressStartup = 0x0004,
00934     };
00935
00936     /// <summary>
00937     /// File transfer option
00938     /// </summary>
00939     [Flags]
00940     public enum FileTransferOption : UInt32
00941     {
00942         /// <summary>
00943         /// No option is selected
00944         /// </summary>
00945         None = 0x0000,
00946
00947         /// <summary>
00948         /// The data gets compressed and transferred. The other side receives and stores the
      compressed data.
00949         /// </summary>
```

```
00950         CompressOnly = 0x0001,
00951
00952         /// <summary>
00953         /// The data gets compressed, transferred and decompressed. This saves bandwidth.
00954         /// </summary>
00955         CompressDecompress = 0x0002,
00956     };
00957
00958     /// <summary>
00959     /// File types on the Motion Coordinator
00960     /// </summary>
00961     public enum FileType : Int32
00962     {
00963         /// <summary>
00964         /// The file type is not recognized
00965         /// </summary>
00966         Undefined = -1,
00967
00968         /// <summary>
00969         /// TrioBASIC program
00970         /// </summary>
00971         BASIC = 0,
00972
00973         /// <summary>
00974         /// Encrypted TrioBASIC program.
00975         /// </summary>
00976         EncryptedBASIC = 1,
00977
00978         /// <summary>
00979         /// GCode program.
00980         /// </summary>
00981         GCode = 2,
00982
00983         /// <summary>
00984         /// Unknown contents.
00985         /// </summary>
00986         TextFile = 3,
00987
00988         /// <summary>
00989         /// IEC program
00990         /// </summary>
00991         IecProgram = 4,
00992
00993         /// <summary>
00994         /// FIFO (not valid in project)
00995         /// </summary>
00996         FifoFile = 5,
00997
00998         /// <summary>
00999         /// Temporary File (not valid in project)
01000         /// </summary>
01001         TempFile = 6,
01002
01003         /// <summary>
01004         /// Project file.
01005         /// </summary>
01006         Project = 7,
01007
01008         /// <summary>
01009         /// Container for anything that the controller does not understand
01010         /// </summary>
01011         Binary = 8,
01012
01013         /// <summary>
01014         /// Special for the MC_CONFIG file
01015         /// </summary>
01016         MCConfig = 9,
01017
01018         /// <summary>
01019         /// IEC library
01020         /// </summary>
01021         IecLibrary = 10,
01022
01023         /// <summary>
01024         /// HMI
01025         /// </summary>
01026         HmiDesign = 11,
01027
01028         /// <summary>
01029         /// Trio BASIC library
01030         /// </summary>
01031         BASICLibrary = 12,
01032
01033         /// <summary>
01034         /// Encrypted Trio BASIC library
01035         /// </summary>
01036         EncryptedBASICibrary = 13,
```

```
01037
01038         /// <summary>
01039         /// HMI Library
01040         /// </summary>
01041         HmiLibrary = 14,
01042
01043         /// <summary>
01044         /// Volatile temporary controller file stored in RAM (not valid in project)
01045         /// </summary>
01046         TempVolatileFile = 15,
01047
01048         /// <summary>
01049         /// Robot program
01050         /// </summary>
01051         Robot = 16,
01052
01053         /// <summary>
01054         /// Robot basic program
01055         /// </summary>
01056         RobotBasic = 17,
01057
01058         /// <summary>
01059         /// Temporary Text File (not valid in project)
01060         /// </summary>
01061         TempTextFile = 18,
01062
01063         /// <summary>
01064         /// Robot library
01065         /// </summary>
01066         RobotLibrary = 19,
01067
01068         /// <summary>
01069         /// Robot basic library
01070         /// </summary>
01071         RobotBasicLibrary = 20,
01072
01073         /// <summary>
01074         /// Recipe data
01075         /// </summary>
01076         Recipe = 21,
01077
01078         /// <summary>
01079         /// Alarm data
01080         /// </summary>
01081         Alarms = 22,
01082
01083         /// <summary>
01084         /// Real-time basic library
01085         /// </summary>
01086         RealtimeBASICLibrary = 23,
01087
01088         /// <summary>
01089         /// Encrypted real-time basic library
01090         /// </summary>
01091         EncryptedRealtimeBasicLibrary = 24,
01092
01093         /// <summary>
01094         /// Encrypted IEC program
01095         /// </summary>
01096         EncryptedIecProgram = 25,
01097
01098         /// <summary>
01099         /// Encrypted IEC library
01100         /// </summary>
01101         EncryptedIecLibrary = 26,
01102
01103         /// <summary>
01104         /// Not a file type. Used to size the FileType enumeration.
01105         /// </summary>
01106         __Size,
01107     };
01108
01109     /// <summary>
01110     /// Flags related to files and programs in internal storage on the Motion Coordinator
01111     /// </summary>
01112     [Flags]
01113     public enum FileFlags : UInt32
01114     {
01115         /// <summary>
01116         /// No flags are set
01117         /// </summary>
01118         None = 0x00,
01119
01120         /// <summary>
01121         /// If set file is editable on the controller
01122         /// </summary>
01123         Editable = 0x01,
```

```
01124
01125         /// <summary>
01126         /// If set program can be auto-run
01127         /// </summary>
01128         Autorun = 0x02,
01129     };
01130
01131     /// <summary>
01132     /// Mode for CONNECT command
01133     /// </summary>
01134     public enum ConnectMode : UInt32
01135     {
01136         /// <summary>
01137         /// Measured position (MPOS) is followed by connected axis
01138         /// </summary>
01139         MPOS = 0,
01140
01141         /// <summary>
01142         /// Demand position (DPOS) is followed by connected axis
01143         /// </summary>
01144         DPOS = 1,
01145     };
01146
01147     /// <summary>
01148     /// Flags PSWITCH command in single/cyclic mode
01149     /// </summary>
01150     [Flags]
01151     public enum PSwitchMode : UInt32
01152     {
01153         /// <summary>
01154         /// No flags are set
01155         /// </summary>
01156         None = 0x00,
01157
01158         /// <summary>
01159         /// If set - set position should be bigger than reset position
01160         ///  If not set - set position should be less than reset position
01161         /// </summary>
01162         Invert = 0x02,
01163
01164         /// <summary>
01165         /// If set DPOS is used
01166         ///  If not set MPOS is used
01167         /// </summary>
01168         DPOS = 0x04,
01169
01170         /// <summary>
01171         /// Mask of allowed bits
01172         /// </summary>
01173         Mask = Invert | DPOS,
01174     };
01175
01176     /// <summary>
01177     /// Flags PSWITCH command in multi/sequence mode
01178     /// </summary>
01179     [Flags]
01180     public enum PSwitchMultiMode : UInt32
01181     {
01182         /// <summary>
01183         /// No flags are set
01184         /// </summary>
01185         None = 0x00,
01186
01187         /// <summary>
01188         /// If set DPOS is used
01189         ///  If not set MPOS is used
01190         /// </summary>
01191         DPOS = 0x04,
01192
01193         /// <summary>
01194         /// If set timer mode
01195         ///  If not set set and reset positions mode
01196         /// </summary>
01197         TimerMode = 0x10,
01198
01199         /// <summary>
01200         /// If set negative direction
01201         ///  If not set positive direction
01202         /// </summary>
01203         NegativeDirection = 0x20,
01204
01205         /// <summary>
01206         /// If set Z output mode
01207         ///  If not set digital output mode
01208         /// </summary>
01209         ZOut = 0x40,
01210
```

```
01211         /// <summary>
01212         /// Mask of allowed bits
01213         /// </summary>
01214         Mask = DPOS | TimerMode | NegativeDirection | ZOut,
01215     };
01216 
01217     /// <summary>
01218     /// Link options in CamBox command
01219     /// </summary>
01220     [Flags]
01221     public enum CamBoxLinkOptions : UInt32
01222     {
01223         /// <summary>
01224         /// No flags are set
01225         /// </summary>
01226         None = 0,
01227 
01228         /// <summary>
01229         /// link commences exactly when registration event MARK occurs on link axis
01230         /// </summary>
01231         LinkMark = 1,
01232 
01233         /// <summary>
01234         /// link commences at an absolute position on link axis
01235         /// </summary>
01236         LinkAbsolute = 2,
01237 
01238         /// <summary>
01239         /// CAMBOX repeats automatically and bi-directionally when this bit is set.
01240         /// </summary>
01241         LinkRepeat = 4,
01242 
01243         /// <summary>
01244         /// PATTERN mode. Advanced use of CAMBOX: allows multiple scale values to be used
01245         /// </summary>
01246         LinkPattern = 8,
01247 
01248         /// <summary>
01249         /// Link is only active during a positive move on the link axis
01250         /// </summary>
01251         LinkPositiveMove = 32,
01252 
01253         /// <summary>
01254         /// Forces the profile to start at a defined point in the link_dist
01255         /// </summary>
01256         LinkForceStart = 128,
01257 
01258         /// <summary>
01259         /// link commences exactly when registration event MARKB occurs on link axis
01260         /// </summary>
01261         LinkMarkB = 256,
01262 
01263         /// <summary>
01264         /// link commences exactly when registration event R_MARK occurs on link axis.
01265         /// </summary>
01266         LinkR_Mark = 512,
01267 
01268         /// <summary>
01269         /// Link is only active during a negative move on the link axisLink is active during positive
      movement from a threshold.
01270         /// </summary>
01271         LinkThreshold = 16384,
01272     };
01273 
01274     /// <summary>
01275     /// Enumeration of known axis types used with ATYPE axis parameter
01276     /// </summary>
01277     public enum AxisType : UInt16
01278     {
01279         None = 0,
01280 
01281         Stepper = 1,
01282 
01283         Servo = 2,
01284 
01285         Encoder = 3,
01286 
01287         StepperEncoderInt = 4,
01288 
01289         Resolver = 5,
01290 
01291         VOut = 6,
01292 
01293         SSI = 7,
01294 
01295         CAN = 8,
01296 
```

```
01297         CAN_Remote = 9,
01298
01299         PSwitch = 10,
01300
01301         DLink_Remote = 11,
01302
01303         Servo2 = 12,
01304
01305         Yaskawa = 13,
01306
01307         EncoderOut = 14,
01308
01309         HighestMotionType = EncoderOut,
01310
01311         Dummy15 = 15,
01312
01313         Dummy16 = 16,
01314
01315         Dummy17 = 17,
01316
01317         DS402PosType1 = 18,
01318
01319         DS402VelType1 = 19,
01320
01321         First_Comms = 20,
01322
01323         CAN_Comms = First_Comms,
01324
01325         USB_Comms = 21,
01326
01327         DLink_Comms = 22,
01328
01329         Profibus_Comms = 23,
01330
01331         Dummy24 = 24,
01332
01333         Dummy25 = 25,
01334
01335         DS402PosType2 = 26,
01336
01337         DS402VelType2 = 27,
01338
01339         Dummy28 = 28,
01340
01341         Dummy29 = 29,
01342
01343         DACout_ANALin = 30,
01344
01345         ANALout_ANALin = 31,
01346
01347         Dummy32 = 32,
01348
01349         Dummy33 = 33,
01350
01351         Dummy34 = 34,
01352
01353         SDout_PWMout = 35,
01354
01355         /// <summary>
01356         /// PWM variant that uses path distance to generate output
01357         /// </summary>
01358         SDout_PWMout2 = 36,
01359
01360         PWMout_SDFREQin = 37,
01361
01362         PWMout_SDin = 37,
01363
01364         Dummy38 = 38,
01365
01366         Dummy39 = 39,
01367
01368         /// <summary>
01369         /// This is the first MC464 usable type
01370         /// </summary>
01371         MechatroPos = 40,
01372
01373         MechatroSpeed = 41,
01374
01375         MechatroTorque = 42,
01376
01377         /// <summary>
01378         /// Standard stepper axis
01379         /// </summary>
01380         SDout_Zout = 43,
01381
01382         /// <summary>
01383         /// Standard analogue servo axis with encoder feedback
```

```
01384         /// </summary>
01385         DACout_QUADin = 44,
01386
01387         QUADout_Zout = 45,
01388
01389         DACout_TAMAin = 46,
01390
01391         DACout_ENDATin = 47,
01392
01393         DACout_SSIin = 48,
01394
01395         MechatroInv = 49,
01396
01397         PanasonicPos = 50,
01398
01399         PanasonicSpeed = 51,
01400
01401         PanasonicTorque = 52,
01402
01403         SercosSpeed = 53,
01404
01405         SercosPos = 54,
01406
01407         SercosTorque = 55,
01408
01409         SercosOpen = 56,
01410
01411         SercosSpeedReg = 57,
01412
01413         SercosPosReg = 58,
01414
01415         SercosEncoder = 59,
01416
01417         DACout_SDin = 60,
01418
01419         SLM32 = 61,
01420
01421         PLM32 = 62,
01422
01423         SDout_Zin = 63,
01424
01425         QUADout_Zin = 64,
01426
01427         EthercatPos = 65,
01428
01429         EthercatSpeed = 66,
01430
01431         EthercatTorque = 67,
01432
01433         EthercatOpenSpeed = 68,
01434
01435         EthercatEncoderIn = 69,
01436
01437         EthercatInAnalOut = 70,
01438
01439         EthercatPosTrio = 71,
01440
01441         EthercatSpeedTrio = 72,
01442
01443         EthercatDummy5 = 73,
01444
01445         EthercatDummy6 = 74,
01446
01447         DACout_SSISLAVEin = 75,
01448
01449         /// <summary>
01450         /// Standard encoder only axis
01451         /// </summary>
01452         QUADin = 76,
01453
01454         DACout_QUADin_Zout = 77,
01455
01456         SDout_VFFDACout = 78,
01457
01458         SDin = 79,
01459
01460         G5Pos = 80,
01461
01462         G5Speed = 81,
01463
01464         G5Torque = 82,
01465
01466         Jacquard = 83,
01467
01468         QUADout_VFFDACout = 84,
01469
01470         AxesDiff = 85,
```

```
01471
01472         TAMAin = 86,
01473
01474         ENDATin = 87,
01475
01476         SSIin = 88,
01477
01478         SDFREQout_ANALin = 89,
01479
01480         DACout_SDFREQin = 90,
01481
01482         AFFDACout = 91,
01483
01484         DACout_BISSin = 92,
01485
01486         BISSin = 93,
01487
01488         DACPWMout_QUADin = 94,
01489
01490         SDout_TAMAin = 95,
01491
01492         Dummy96 = 96,
01493
01494         QUADin_Zout = 97,
01495
01496         Dummy98 = 98,
01497
01498         Dummy99 = 99,
01499
01500         SDout_HD = 100,
01501
01502         QUADout_HD = 101,
01503
01504         DACout_QUADin_HD = 102,
01505
01506         DACout_SDin_HD = 103,
01507
01508         QUADin_HD = 104,
01509
01510         SSIin_HD = 105,
01511
01512         TAMAin_HD = 106,
01513
01514         Dummy107 = 107,
01515
01516         Dummy108 = 108,
01517
01518         Dummy109 = 109,
01519
01520         SDout_SDin_HD = 110,
01521
01522         SDout_QUADin_HD = 111,
01523
01524         QUADout_SDin_HD = 112,
01525
01526         QUADout_QUADin_HD = 113,
01527
01528         SDout_TAMAin_HD = 114,
01529
01530         SDout_ENDATin_HD = 115,
01531
01532         SDout_SSIin_HD = 116,
01533
01534         SDout_BISSin_HD = 117,
01535
01536         QUADout_TAMAin_HD = 118,
01537
01538         QUADout_ENDATin_HD = 119,
01539
01540         QUADout_SSIin_HD = 120,
01541
01542         QUADout_BISSin_HD = 121,
01543
01544         Dummy122 = 122,
01545
01546         Dummy123 = 123,
01547
01548         Dummy124 = 124,
01549
01550         Dummy125 = 125,
01551
01552         Dummy126 = 126,
01553
01554         Dummy127 = 127,
01555
01556         Dummy128 = 128,
01557
```

```
01558         Dummy129 = 129,
01559
01560         SLMvirtual = 200,
01561
01562         PLMvirtual = 201,
01563     };
01564
01565     /// <summary>
01566     /// Enumeration of axis mode flags used with AXIS_MODE axis parameter
01567     /// </summary>
01568     [Flags]
01569     public enum AxisMode : UInt32
01570     {
01571         NOT_USED_1 = ((UInt32)1 << 0 ),
01572
01573         ALLOW_CONNECT_MOVES_IN_LIMIT = ((UInt32)1 << 1 ),
01574
01575         DIRECTION_MODE_3D = ((UInt32)1 << 2 ),
01576
01577         CANCEL_ADDAX_ON_LIMIT = ((UInt32)1 << 3 ),
01578
01579         DISABLE_LOOKAHEAD = ((UInt32)1 << 4 ),
01580
01581         NOT_USED_2 = ((UInt32)1 << 5 ),
01582
01583         USE_NON_SIGNEXTENDED_ANALOGUE_FEEDBACK = ((UInt32)1 << 6 ),
01584
01585         NOT_USED_3 = ((UInt32)1 << 7 ),
01586
01587         SOFTWARE_LIMIT_USES_DPOS = ((UInt32)1 << 8 ),
01588
01589         DISABLE_FE_ACCUMULATION_DURING_MOVE = ((UInt32)1 << 9 ),
01590
01591         USE_FASTDEC_WHEN_CANCELLING_RPS_JOG = ((UInt32)1 << 10),
01592
01593         DELAY_POSITION_EVALUATION_TO_LOAD_TIME = ((UInt32)1 << 11),
01594
01595         DIRECTION_SHORTEST = ((UInt32)1 << 12),
01596
01597         MONITOR_FE = ((UInt32)1 << 13),
01598
01599         OVERRIDE_FE_LIMITS = ((UInt32)1 << 14),
01600
01601         DUAL_AXIS_AVERAGE_POS = ((UInt32)1 << 15),
01602
01603         INVERT_AXIS_DIRECTION = ((UInt32)1 << 16),
01604
01605         CANCEL_MTYPE_ON_MASTER_IF_ADDAX_HITS_LIM = ((UInt32)1 << 17),
01606     };
01607
01608     /// <summary>
01609     /// Enumeration of corner mode flags used with CORNER_MODE axis parameter
01610     /// </summary>
01611     [Flags]
01612     public enum CornerMode : UInt32
01613     {
01614         RESERVED = ((UInt32)1 << 0),
01615
01616         AUTO_SPEED_CONTROL = ((UInt32)1 << 1),
01617
01618         CORNER_STATE_MACHINE = ((UInt32)1 << 2),
01619
01620         AUTO_RADIUS_SPEED_CONTROL = ((UInt32)1 << 3),
01621
01622         AUTO_RADIUS_SPEED_CONTROL_CENTRIPETAL_MODE = ((UInt32)1 << 4),
01623
01624         AUTO_RADIUS_SPEED_CONTROL_CENTRIPETAL_MODE_USE_ACCEL = ((UInt32)1 << 5),
01625
01626         AUTO_BLEND_MODE_0_USING_CORNER_RADIUS = ((UInt32)1 << 6),
01627
01628         AUTO_BLEND_MODE_1_USING_CORNER_RADIUS = ((UInt32)1 << 7),
01629     };
01630
01631     /// <summary>
01632     /// Enumeration of corner state flags used with CORNER_STATE axis parameter
01633     /// </summary>
01634     [Flags]
01635     public enum CornerState : UInt32
01636     {
01637         HELD = ((UInt32)1 << 0),
01638
01639         LOAD = ((UInt32)1 << 1),
01640
01641         LOAD_ZERO_SPEED = ((UInt32)1 << 2),
01642     };
01643
01644     /// <summary>
```

```
01645     /// Motion types returned by MTYPE and NTYPE axis parameters
01646     /// </summary>
01647     public enum MotionType : UInt32
01648     {
01649         IDLE = 0,
01650 
01651         MOVE = 1,
01652 
01653         MOVEABS = 2,
01654 
01655         MHELICAL = 3,
01656 
01657         MOVECIRC = 4,
01658 
01659         MOVEMODIFY = 5,
01660 
01661         MOVESP = 6,
01662 
01663         MOVEABSSP = 7,
01664 
01665         MOVECIRCSP = 8,
01666 
01667         MHELICALSP = 9,
01668 
01669         FORWARD = 10,
01670 
01671         REVERSE = 11,
01672 
01673         DATUM = 12,
01674 
01675         CAM = 13,
01676 
01677         FORWARD_JOG = 14,
01678 
01679         REVERSE_JOG = 15,
01680 
01681         WETCAM = 16,
01682 
01683         CAMBOX = 20,
01684 
01685         CONNECT = 21,
01686 
01687         MOVELINK = 22,
01688 
01689         CONNPATH = 23,
01690 
01691         FLEXLINK = 24,
01692 
01693         CAMLINK = 25,
01694 
01695         COMPENSATE_XY = 26,
01696 
01697         PICKLINK = 27,
01698 
01699         CAMIN = 28,
01700 
01701         MOVETANG = 30,
01702 
01703         MSPHERICAL = 31,
01704 
01705         MTRANSCIRC = 32,
01706 
01707         MOVESEQ = 33,
01708 
01709         MOVEABSSEQ = 34,
01710 
01711         MOVETURN = 35,
01712 
01713         MOVEPICK = 36,
01714 
01715         MOVECONTOUR = 37,
01716 
01717         MOVEDISPENSE = 38,
01718 
01719         MSPHERICALSP = 40,
01720 
01721         MTRANSCIRCSP = 41,
01722 
01723         MOVESEQSP = 42,
01724 
01725         MOVEABSSEQSP = 43,
01726 
01727         MOVEJ = 44,
01728 
01729         MOVEL = 45,
01730 
01731         MOVEC = 46,
```

```
01732
01733         MOVEBLEND = 47,
01734
01735         MOVEJREL = 48,
01736
01737         MOVELREL = 49,
01738
01739         MOVETEST = 50,
01740
01741         MOVEP = 51,
01742
01743         MOVEU = 52,
01744
01745         MOVEB = 53,
01746
01747         MOVESPCONT = 60,
01748
01749         MOVEABSSPCONT = 61,
01750
01751         GEARIN = 62,
01752
01753         ACCPROFILE = 70,
01754
01755         VELPROFILE = 71,
01756
01757         POSPROFILE = 72,
01758
01759         COMBINEAXES = 73,
01760
01761         OPENMOVE_BASE = 100,
01762
01763         OPENMOVE_1 = OPENMOVE_BASE,
01764
01765         OPENMOVE_2 = OPENMOVE_BASE + 1,
01766
01767         OPENMOVE_3 = OPENMOVE_BASE + 2,
01768
01769         OPENMOVE_4 = OPENMOVE_BASE + 3,
01770
01771         OPENMOVE_5 = OPENMOVE_BASE + 4,
01772     };
01773
01774     [Flags]
01775     public enum AxisStatus : UInt64
01776     {
01777         AS_LOOKAHEAD_OVERRIDE = ((UInt64)1 << 0 ),
01778
01779         AS_FE_WARNING = ((UInt64)1 << 1 ),
01780
01781         AS_COMMS_ERROR = ((UInt64)1 << 2 ),
01782
01783         AS_REMOTE_DRIVE_ERROR = ((UInt64)1 << 3 ),
01784
01785         AS_IN_FORWARD_LIMIT = ((UInt64)1 << 4 ),
01786
01787         AS_IN_REVERSE_LIMIT = ((UInt64)1 << 5 ),
01788
01789         AS_DATUMING = ((UInt64)1 << 6 ),
01790
01791         AS_FEEDHOLD = ((UInt64)1 << 7 ),
01792
01793         AS_FE_EXCEEDS_LIMIT = ((UInt64)1 << 8 ),
01794
01795         AS_FORWARD_SOFTWARE_LIMIT = ((UInt64)1 << 9 ),
01796
01797         AS_REVERSE_SOFTWARE_LIMIT = ((UInt64)1 << 10),
01798
01799         AS_CANCEL_CURRENT = ((UInt64)1 << 11),
01800
01801         AS_PULSE_OVER_SPEED = ((UInt64)1 << 12),
01802
01803         AS_OVERRIDE_TO_ZERO = ((UInt64)1 << 13),
01804
01805         AS_MOVE_CANCEL_DONE = ((UInt64)1 << 14),
01806
01807         AS_VOLUME_LIMIT = ((UInt64)1 << 15),
01808
01809         AS_AXIS_FS_LIMIT = ((UInt64)1 << 16),
01810
01811         AS_AXIS_RS_LIMIT = ((UInt64)1 << 17),
01812
01813         AS_ENCODER_OVER_I = ((UInt64)1 << 18),
01814
01815         AS_PSWITCH_FIFO_NOT_EMPTY = ((UInt64)1 << 19),
01816
01817         AS_PSWITCH_FIFO_FULL = ((UInt64)1 << 20),
01818
```

```
01819         AS_KINEMATICS_1HOUR = ((UInt64)1 << 21),
01820
01821         AS_BISS_CLEAN_WARNING = ((UInt64)1 << 22),
01822
01823         AS_WORLD_FS_LIMIT = ((UInt64)1 << 23),
01824
01825         AS_WORLD_RS_LIMIT = ((UInt64)1 << 24),
01826
01827         AS_CANCEL_CURRENT_AT_SPEED = ((UInt64)1 << 25),
01828
01829         AS_REMOTE_NODE_STATUS_ERROR = ((UInt64)1 << 26),
01830
01831         AS_PSWITCH_FIFO_HALF_FULL = ((UInt64)1 << 27),
01832
01833         AS_MOVEADD = ((UInt64)1 << 28),
01834
01835         AS_CANCEL_MOVEADD = ((UInt64)1 << 29),
01836
01837         AS_REMOTE_NODE_ERROR2 = ((UInt64)1 << 30),
01838
01839         AS_SINGULARITY = ((UInt64)1 << 31),
01840
01841         AS_OPEN_FRAME_ERROR = ((UInt64)1 << 32),
01842
01843         AS_VP_ERROR = ((UInt64)1 << 33),
01844
01845         AS_CANCEL_BUFFERED = ((UInt64)1 << 34),
01846
01847         AS_CANCEL_BUFFERED_DECEL_INTO_BUFFERED_MOVES = ((UInt64)1 << 35),
01848
01849         AS_CANCEL_BUFFERED_DECEL_WITHIN_CURRENT_MOVE = ((UInt64)1 << 36),
01850
01851         AS_CANCEL_LAST_BUFFERED = ((UInt64)1 << 37),
01852     };
01853
01854     [Flags]
01855     public enum RobotStatus : UInt64
01856     {
01857         RS_WORLD_FS_LIMIT = ((UInt64)1 << 0),
01858
01859         RS_WORLD_RS_LIMIT = ((UInt64)1 << 1),
01860
01861         RS_ROBOT_FS_LIMIT = ((UInt64)1 << 2),
01862
01863         RS_ROBOT_RS_LIMIT = ((UInt64)1 << 3),
01864
01865         RS_TCP_FS_LIMIT = ((UInt64)1 << 4),
01866
01867         RS_TCP_RS_LIMIT = ((UInt64)1 << 5),
01868
01869         RS_FE_WARNING = ((UInt64)1 << 6),
01870
01871         RS_FE_EXCEEDS_LIMIT = ((UInt64)1 << 7),
01872
01873         RS_WRIST_SINGULARITY = ((UInt64)1 << 8),
01874
01875         RS_ALIGNMENT_SINGULARITY = ((UInt64)1 << 9),
01876
01877         RS_ELBOW_SINGULARITY = ((UInt64)1 << 10),
01878
01879         RS_MAX_SPEED_LIMIT = ((UInt64)1 << 11),
01880
01881         RS_COLLIDED = ((UInt64)1 << 12),
01882
01883         RS_NOT_IN_POS = ((UInt64)1 << 13),
01884
01885         RS_ELBOW_CLOSED_SINGULARITY = ((UInt64)1 << 14),
01886
01887         RS_DISALLOW_JOG = ((UInt64)1 << 15),
01888     };
01889
01890     public enum ProcessStatus : UInt32
01891     {
01892         STOPPED,
01893
01894         READY,
01895
01896         STEPPING,
01897
01898         PAUSED,
01899
01900         PAUSING,
01901
01902         ABORTED,
01903
01904         SUSPEND,
01905
```

```
01906         SUSPENDED,
01907
01908         SLEEPING,
01909     };
01910
01911     public enum SyncControl : Int32
01912     {
01913         SC_WAITING_TO_START = -1,
01914
01915         SC_NO_SYNC = 0,
01916
01917         SC_STARTING_SYNC = 1,
01918
01919         SC_PERFORMING_SYNC = 2,
01920
01921         SC_SYNCHRONISED = 3,
01922
01923         SC_STOPPING_SYNC = 4,
01924
01925         SC_STARTING_INTERP_MOVE = 5,
01926
01927         SC_PERFORMING_INTERP_MOVE = 6,
01928
01929         SC_STARTING_RESYNC = 10,
01930
01931         SC_PERFORMING_RESYNC = 11,
01932
01933         SC_STARTING_RESYNC_TO_USERFRAME = 20,
01934
01935         SC_PERFORMING_RESYNC_TO_USERFRAME = 21,
01936     };
01937
01938 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
01939
01940     #endregion // Enumerations
01941
01942     #region Delegates and helper Structures
01943
01944     /// <summary>
01945     /// Delegate for connection related events.
01946     /// </summary>
01947     public delegate void ConnectionEventHandler(EventType eventType, long integerData, string
      stringData);
01948
01949     /// <summary>
01950     /// Data used with the delegate for processing the response of Dir command
01951     /// </summary>
01952     [StructLayout(LayoutKind.Sequential)]
01953     public struct FileInfo
01954     {
01955         /// <summary>
01956         /// Type of the file on the controller
01957         /// </summary>
01958         public FileType type;
01959         /// <summary>
01960         /// Text code size
01961         /// </summary>
01962         public uint size;
01963         /// <summary>
01964         /// Compiled binary size
01965         /// </summary>
01966         public uint compiled_size;
01967         /// <summary>
01968         /// Autorun process number
01969         /// </summary>
01970         public int autorun_process;
01971         /// <summary>
01972         /// Flags related to files and programs on the controller
01973         /// </summary>
01974         public FileFlags flags;
01975     };
01976
01977     /// <summary>
01978     /// Delegate for processing the response of Dir command
01979     /// </summary>
01980     public delegate void FileCallback(ref FileInfo file, string fileName);
01981
01982     /// <summary>
01983     /// Data used with the delegate for streaming back the response of commands that return string
      data
01984     /// </summary>
01985     [StructLayout(LayoutKind.Sequential)]
01986     public struct DataStreamPacket
01987     {
01988         /// <summary>
01989         /// The length of the stream, if available. 0 otherwise
```

```
01990         /// </summary>
01991         public uint total_length;
01992         /// <summary>
01993         /// The packet number
01994         /// </summary>
01995         public uint packet_number;
01996         /// <summary>
01997         /// The total count of the packets of the stream, if available. 0 otherwise.
01998         /// </summary>
01999         public uint packet_count;
02000         /// <summary>
02001         /// The number of bytes
02002         /// </summary>
02003         public uint data_length;
02004     };
02005
02006     /// <summary>
02007     /// Delegate for streaming back the response of commands that return string data
02008     /// </summary>
02009     public unsafe delegate void StreamHandlerCallback(ref DataStreamPacket packet, byte* buff);
02010
02011     /// <summary>
02012     /// Data used with the progress report
02013     /// </summary>
02014     [StructLayout(LayoutKind.Sequential)]
02015     public struct ProgressInfo
02016     {
02017         /// <summary>
02018         /// The length of the process. If 0 the length is unknown.
02019         /// </summary>
02020         public uint total_length;
02021         /// <summary>
02022         /// The current progress position.
02023         /// </summary>
02024         public uint current_pos;
02025         /// <summary>
02026         /// Progress event
02027         /// </summary>
02028         public ProgressEvent evt;
02029     };
02030
02031     /// <summary>
02032     /// Delegate for streaming back the response of commands that return string data
02033     /// </summary>
02034     public delegate void ProgressHandlerCallback(ref ProgressInfo progress);
02035
02036     #endregion // Delegates and helper Structures
02037
02038     #region TrioConnectionException
02039
02040     /// <summary>
02041     /// Exception raised when error occurs in communication library
02042     /// </summary>
02043     public class TrioConnectionException : Exception
02044     {
02045         /// <summary>
02046         /// Error code associated causing the exception
02047         /// </summary>
02048         public ErrorCode ErrorCode { get; }
02049
02050         /// <summary>
02051         /// Text representation of the error
02052         /// </summary>
02053         public override string Message => TrioConnectionFactory.ErrorStringGetter?.Invoke(ErrorCode)
      ?? "No error context - connection module not initialized";
02054
02055         /// <summary>
02056         /// Initializing constructor
02057         /// </summary>
02058         /// <param name="errorCode"></param>
02059         public TrioConnectionException(ErrorCode errorCode)
02060         {
02061             ErrorCode = errorCode;
02062         }
02063     }
02064
02065     #endregion // TrioConnectionException
02066
02067     #region ITrioConnection interface
02068
02069     /// <summary>
02070     /// Connection interface
02071     /// </summary>
02072     public interface ITrioConnection : IDisposable
02073     {
02074         /// <summary>
02075         /// Connection related event
```

```
02076         /// </summary>
02077         event ConnectionEventHandler ConnectionEvent;
02078 
02079         /// <summary>
02080         /// True if connection to Motion Coordinator was opened.
02081         /// </summary>
02082         bool IsConnected { get; }
02083 
02084         /// <summary>
02085         /// Connection type or current connection instance or Unknown if connection is disposed
02086         /// </summary>
02087         ConnectionType ConnectionType { get; }
02088 
02089 
      ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
02090 
02091         /// <summary>
02092         /// Set host name or IP to connection object. Valid only for TCP connection
02093         /// </summary>
02094         /// <param name="host">New value for the host name</param>
02095         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02096         void SetTcpHostName(String host);
02097 
02098         /// <summary>
02099         /// Get current host name or IP of connection object. Valid only for TCP connection
02100         /// </summary>
02101         /// <returns>Current value of the host name</returns>
02102         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02103         String GetTcpHostName();
02104 
02105         /// <summary>
02106         /// Set TCP port to connection object. Valid only for TCP connection.
02107         /// </summary>
02108         /// <param name="port">New value for the TCP port</param>
02109         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02110         void SetTcpPort(UInt16 port);
02111 
02112         /// <summary>
02113         /// Get current TCP port from connection object. Valid only for TCP connection.
02114         /// </summary>
02115         /// <returns>Current value of the TCP port</returns>
02116         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02117         UInt16 GetTcpPort();
02118 
02119         /// <summary>
02120         /// Set TCP connection timeout. Valid only for TCP connection.
02121         /// </summary>
02122         /// <param name="timeout_ms">New value for the TCP connection timeout in milliseconds</param>
02123         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02124         void SetTcpTimeout(UInt32 timeout_ms);
02125 
02126         /// <summary>
02127         /// Get current TCP connection timeout. Valid only for TCP connection.
02128         /// </summary>
02129         /// <returns>Current value of the TCP connection timeout in milliseconds</returns>
02130         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02131         UInt32 GetTcpTimeout();
02132 
02133         /// <summary>
02134         /// Get Unified API version info
02135         /// </summary>
02136         /// <returns>On success version info of the Unified API</returns>
02137         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02138         Version GetApiVersion();
02139 
02140         /// <summary>
02141         /// Get Unified API version string
02142         /// </summary>
02143         /// <returns>On success string representation of the Unified API version</returns>
02144         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02145         String GetApiVersionStr();
02146 
02147         /// <summary>
02148         /// Get the firmware version of the Motion Coordinator.
02149         /// </summary>
02150         /// <returns>On success Motion Coordinator version as a floating point number</returns>
02151         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02152         double GetFirmwareVersionNumber();
```

```
02153
02154         /// <summary>
02155         /// Open a connection to the Motion Coordinator.
02156         /// </summary>
02157         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02158         void OpenConnection();
02159
02160         /// <summary>
02161         /// Closes previously open connection to the Motion Coordinator.
02162         /// </summary>
02163         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02164         void CloseConnection();
02165
02166         /// <summary>
02167         /// Perform software reset on Motion Coordinator.
02168         /// </summary>
02169         /// <param name="mode">Reset mode. See the TrioBASIC EX documentation for more
      details.</param>
02170         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02171         void Ex(Int32 mode = 0);
02172
02173         /// <summary>
02174         /// Used to cancel current or buffered axis commands on an axis or an interpolating axis
      group.
02175         /// </summary>
02176         /// <param name="mode">Cancel mode</param>
02177         /// <param name="axis">Index of axis subject to cancel operation or -1 to cancel default base
      axis</param>
02178         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02179         void Cancel(Int32 mode, Int32 axis = -1);
02180
02181         /// <summary>
02182         /// Links the demand position of the base axis to the measured movements of the driving axes
      to produce an electronic gearbox.
02183         /// </summary>
02184         /// <param name="ratio">This parameter holds the number of edges the base axis is required to
      move per increment of the driving axis. The ratio value can be either positive or negative. The ratio
      is always specified as an encoder edge ratio</param>
02185         /// <param name="linkAxis">This parameter specifies the axis to link to.</param>
02186         /// <param name="mode">
02187         ///  0   The axis follows the measured position(MPOS) of the driving axis. (default)<br/>
02188         ///  1   The axis follows the demand position (DPOS) of the driving axis.
02189         /// </param>
02190         /// <param name="axis">Index of axis subject to cancel operation or -1 to cancel default base
      axis</param>
02191         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02192         void Connect(double ratio, Int32 linkAxis, ConnectMode mode = ConnectMode.MPOS, Int32 axis =
      -1);
02193
02194         /// <summary>
02195         /// Enables you to link to the path of an interpolated movement by linking the demand position
      of the base axis, to the interpolated path distance of the driving axis.
02196         /// </summary>
02197         /// <param name="ratio">This is the ratio between the interpolated distance moved on the
      driving axis to the distance moved on the base axis. </param>
02198         /// <param name="linkAxis">This parameter specifies the axis to link to.</param>
02199         /// <param name="axis">Index of axis subject to cancel operation or -1 to cancel default base
      axis</param>
02200         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02201         void ConnPath(double ratio, Int32 linkAxis, Int32 axis = -1);
02202
02203         /// <summary>
02204         /// The PSWITCH command allows an output to be set when a predefined position is reached, and
      to be reset when a second position is reached
02205         /// </summary>
02206         /// <param name="switchNumber">The switch number in the range 0..63 </param>
02207         /// <param name="mode">Mode of operation. See PSwitchMode enumeration for more
      information</param>
02208         /// <param name="axis">Axis to link the PSWITCH to, may be any real or virtual axis </param>
02209         /// <param name="outputNumber">Selects the output to set, can be any real or virtual output
      </param>
02210         /// <param name="outputState">False - turn the output OFF at setpos, or true - turn the output
      ON at setpos</param>
02211         /// <param name="setPosition">The position at which output is set or reset, in user units
      </param>
02212         /// <param name="resetPosition">The position at which output is set or reset, in user units
      </param>
02213         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02214         void PSwitch(int switchNumber, PSwitchMode mode, Int32 axis, UInt32 outputNumber, bool
      outputState, double setPosition, double resetPosition);
```

```
02215
02216         /// <summary>
02217         /// The PSWITCHOFF command is used to disable
02218         /// </summary>
02219         /// <param name="switchNumber">The switch number in the range 0..63 </param>
02220         /// <param name="hold">
02221         ///  false = The PSWITCH output will hold in the same state it was when the PSWITCH is set to
      OFF . (Default)<br/>
02222         ///  true = The PSWITCH output is forced OFF even if it was ON when the PSWTICH is set to OFF
      .
02223         /// </param>
02224         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02225         void PSwitchOff(int switchNumber, bool hold = false);
02226
02227         /// <summary>
02228         /// The PSWITCHMULTI command allows an output to be set that turns the output on and off at
      selected positions loaded from the TABLE memory.
02229         /// </summary>
02230         /// <param name="switchNumber">The switch number in the range 0..63 </param>
02231         /// <param name="mode">Mode of operation. See PSwitchMultiMode enumeration for more
      information</param>
02232         /// <param name="axis">Axis to link the PSWITCH to, may be any real or virtual axis </param>
02233         /// <param name="outputNumber">Selects the output to set, can be any real or virtual output
      </param>
02234         /// <param name="outputState">False - turn the output OFF at setpos, or true - turn the output
      ON at setpos</param>
02235         /// <param name="tableStart">TABLE index for start of PSWITCH data </param>
02236         /// <param name="tableEnd">TABLE index for end of PSWITCH data </param>
02237         /// <param name="pulseTime">Time in servo-cycles when pulse output mode bit 4 is set </param>
02238         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02239         void PSwitchMulti(int switchNumber, PSwitchMultiMode mode, Int32 axis, UInt32 outputNumber,
      bool outputState, UInt32 tableStart, UInt32 tableEnd, UInt32 pulseTime = 0);
02240
02241         /// <summary>
02242         /// Execute arbitrary command.
02243         /// </summary>
02244         /// <param name="command">Command to be executed</param>
02245         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02246         void Execute(String command);
02247
02248         /// <summary>
02249         /// Execute arbitrary command with response.
02250         /// </summary>
02251         /// <param name="command">Command to be executed</param>
02252         /// <param name="buff">Buffer to receive the response string characters</param>
02253         /// <param name="buff_size">Size of the buffer on input and on exit required buffer size
      including null terminator</param>
02254         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02255         unsafe void ExecuteWithResponse(String command, Byte* buff, ref UInt32 buff_size);
02256
02257         /// <summary>
02258         /// Execute arbitrary command with response.
02259         /// </summary>
02260         /// <param name="command">Command to be executed</param>
02261         /// <param name="handler">Callback function to receive response string chunks</param>
02262         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02263         void ExecuteWithResponse(String command, StreamHandlerCallback handler);
02264
02265         /// <summary>
02266         /// Execute arbitrary command with response.
02267         /// </summary>
02268         /// <param name="command">Command to be executed</param>
02269         /// <returns>Command response</returns>
02270         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02271         String ExecuteWithResponse(String command);
02272
02273         /// <summary>
02274         /// The BASE command is used to direct all subsequent motion commands and axis parameter
      read/writes to a particular axis, or group of axes. The default setting is a sequence: 0, 1, 2, 3...
02275         /// </summary>
02276         /// <param name="axis">The axes to send the motion commands to</param>
02277         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02278         void SetAxisBase(Int32 axis);
02279
02280         /// <summary>
02281         /// Set multiple base axes.
02282         /// </summary>
02283         /// <param name="axes">Array of axis indexes to set as base</param>
02284         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
```

```
02285         void SetMultiAxisBase(Int32[] axes);
02286
02287         /// <summary>
02288         /// Performs one of 6 homing sequences to locate an axis to an absolute position.
02289         /// </summary>
02290         /// <param name="sequence">Bit-mask for homing sequence to be executed</param>
02291         /// <param name="axis">Index of the axis to be subject of datum sequence or -1 to datum
      default base axis</param>
02292         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02293         void Datum(UInt32 sequence, Int32 axis = -1);
02294
02295         /// <summary>
02296         /// Defines the current axis position as a new absolute value
02297         /// </summary>
02298         /// <param name="pos">Absolute position to set on axis in user units</param>
02299         /// <param name="axis">Index of the axis for which the default position to be set or -1 to set
      position to default base axis</param>
02300         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02301         void DefPos(double pos, Int32 axis = -1);
02302
02303         /// <summary>
02304         /// Defines the current axis position as a new absolute value to multiple axes
02305         /// </summary>
02306         /// <param name="pos">Array of absolute positions in user units to set on axes defined by
      SetMultiAxisBase</param>
02307         /// <param name="axis">Index of the first axis for which the default position to be set or -1
      to set position to default base axis</param>
02308         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02309         void DefPosMulti(double[] pos, Int32 axis = -1);
02310
02311         /// <summary>
02312         /// Sets continuous forward movement.
02313         /// </summary>
02314         /// <param name="axis">Axis index to set forward movement to or -1 to apply to default base
      axis.</param>
02315         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02316         void Forward(Int32 axis = -1);
02317
02318         /// <summary>
02319         /// Sets continuous reverse movement.
02320         /// </summary>
02321         /// <param name="axis">Axis index to set reverse movement to or -1 to apply to default base
      axis.</param>
02322         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02323         void Reverse(Int32 axis = -1);
02324
02325         /// <summary>
02326         /// Incremental move of single axis
02327         /// </summary>
02328         /// <param name="pos">Distance to move axis in user units</param>
02329         /// <param name="axis">Index of the axis for which to be subject of move operation or -1 to
      use default base axis</param>
02330         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02331         void MoveRel(double pos, Int32 axis = -1);
02332
02333         /// <summary>
02334         /// Incremental move of multiple axes
02335         /// </summary>
02336         /// <param name="dist">Array of distances in user units to move axes defined by
      SetMultiAxisBase</param>
02337         /// <param name="axis">Index of the first axis for which to be subject of move operation or -1
      to use default base axis</param>
02338         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02339         void MoveRelMulti(double[] dist, Int32 axis = -1);
02340
02341         /// <summary>
02342         /// Incremental move of single axis with speed changed between moves
02343         /// </summary>
02344         /// <param name="pos">Distance to move axis in user units</param>
02345         /// <param name="axis">Index of the axis for which to be subject of move operation or -1 to
      use default base axis</param>
02346         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02347         void MoveRelSp(double pos, Int32 axis = -1);
02348
02349         /// <summary>
02350         /// Incremental move of multiple axes with speed changed between moves
02351         /// </summary>
02352         /// <param name="dist">Array of distances in user units to move axes defined by
      SetMultiAxisBase</param>
```

```
02353         /// <param name="axis">Index of the first axis for which to be subject of move operation or -1
      to use default base axis</param>
02354         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02355         void MoveRelSpMulti(double[] dist, Int32 axis = -1);
02356
02357         /// <summary>
02358         /// Absolute positional move of single axis
02359         /// </summary>
02360         /// <param name="pos">Absolute position to move axis to in user units</param>
02361         /// <param name="axis">Index of the axis for which to be subject of move operation or -1 to
      use default base axis</param>
02362         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02363         void MoveAbs(double pos, Int32 axis = -1);
02364
02365         /// <summary>
02366         /// Absolute positional move of multiple axes
02367         /// </summary>
02368         /// <param name="pos">Array of absolute positions in user units to move axes defined by
      SetMultiAxisBase</param>
02369         /// <param name="axis">Base axis for this command. If value is -1 then the default base axis
      is used.</param>
02370         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02371         void MoveAbsMulti(double[] pos, Int32 axis = -1);
02372
02373         /// <summary>
02374         /// Absolute positional move of single axis with speed changed between moves
02375         /// </summary>
02376         /// <param name="pos">Absolute position to move axis to in user units</param>
02377         /// <param name="axis">Index of the axis for which to be subject of move operation or -1 to
      use default base axis</param>
02378         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02379         void MoveAbsSp(double pos, Int32 axis = -1);
02380
02381         /// <summary>
02382         /// Absolute positional move of multiple axes with speed changed between moves
02383         /// </summary>
02384         /// <param name="pos">Array of absolute positions in user units to move axes defined by
      SetMultiAxisBase</param>
02385         /// <param name="axis">Index of the first axis for which to be subject of move operation or -1
      to use default base axis</param>
02386         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02387         void MoveAbsSpMulti(double[] pos, Int32 axis = -1);
02388
02389         /// <summary>
02390         /// Moves 2 orthogonal axes in such a way as to produce a circular arc at the tool point.
02391         ///  The length and radius of the arc are defined by the five parameters in the command line.
02392         ///  The move parameters are always relative to the end of the last specified move.
02393         ///  This is the start position on the circle circumference. Axis 1 is the current BASE axis.
02394         ///  Axis 2 is the next axis in the BASE array.
02395         ///  The first 4 distance parameters are scaled according to the current unit conversion
      factor for the BASE axis.
02396         /// </summary>
02397         /// <param name="axis">Base axis for this command. If value is -1 then the default base axis
      is used.</param>
02398         /// <param name="finishBase">Position on BASE axis to finish at.</param>
02399         /// <param name="finishNext">Position on next axis in BASE array to finish at.</param>
02400         /// <param name="centreBase">Position on BASE about which to move.</param>
02401         /// <param name="centreNext">Position on next axis in BASE array about which to move.</param>
02402         /// <param name="direction">
02403         ///  0 Arc is interpolated in an anti - clockwise direction<br/>
02404         ///  1 Arc is interpolated in a clockwise direction<br/>
02405         ///  2 Arc is interpolated using the shortest path to endpoint<br/>
02406         ///  3 Arc is interpolated using the longest path to endpoint
02407         /// </param>
02408         /// <param name="transitionAngle">Transition angle (in Radians) for any filleted arcs that are
      generated.  Set to 0 if no transition curve is to be applied.</param>
02409         /// <param name="tableOutput">Output data to TABLE. 0 means do not output data to TABLE,
      otherwise is the TABLE start index for the data.</param>
02410         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02411         void MoveCirc(Int32 axis, double finishBase, double finishNext, double centreBase, double
      centreNext, UInt32 direction, double transitionAngle = 0, UInt32 tableOutput =
      unchecked((UInt32)(-1)));
02412
02413         /// <summary>
02414         /// Works as MoveCirc and additionally allows vector speed to be changed when using multiple
      moves in the look ahead buffer when MERGE = ON , using additional parameters FORCE_SPEED and
      ENDMOVE_SPEED.
02415         ///  Optional parameters ta and output allow the use of transition curves.
02416         /// </summary>
02417         /// <param name="axis">Base axis for this command. If value is -1 then the default base axis
      is used.</param>
```

```
02418         /// <param name="finishBase">Position on BASE axis to finish at.</param>
02419         /// <param name="finishNext">Position on next axis in BASE array to finish at.</param>
02420         /// <param name="centreBase">Position on BASE about which to move.</param>
02421         /// <param name="centreNext">Position on next axis in BASE array about which to move.</param>
02422         /// <param name="direction">
02423         ///  0 Arc is interpolated in an anti - clockwise direction<br/>
02424         ///  1 Arc is interpolated in a clockwise direction<br/>
02425         ///  2 Arc is interpolated using the shortest path to endpoint<br/>
02426         ///  3 Arc is interpolated using the longest path to endpoint
02427         /// </param>
02428         /// <param name="transitionAngle">Transition angle (in Radians) for any filleted arcs that are
      generated.  Set to 0 if no transition curve is to be applied.</param>
02429         /// <param name="tableOutput">Output data to TABLE. 0 means do not output data to TABLE,
      otherwise is the TABLE start index for the data.</param>
02430         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02431         void MoveCircSp(Int32 axis, double finishBase, double finishNext, double centreBase, double
      centreNext, UInt32 direction, double transitionAngle = 0, UInt32 tableOutput = 0);
02432
02433         /// <summary>
02434         /// Performs a helical move.
02435         ///  Moves 2 orthogonal axes in such a way as to produce a circular arc at the tool point with
      a simultaneous linear move on a third axis.
02436         ///  The first 5 parameters are similar to those of an MOVECIRC command.The sixth parameter
      defines the simultaneous linear move.
02437         /// </summary>
02438         /// <param name="axis">Base axis for this command. If value is -1 then the default base axis
      is used.</param>
02439         /// <param name="finishBase">Position on the base axis to finish at.</param>
02440         /// <param name="finishNext">Position on the next axis in the base vector to finish
      at.</param>
02441         /// <param name="centreBase">Position on the base axis about which to move.</param>
02442         /// <param name="centreNext">Position on the next axis in the base vector about which to
      move.</param>
02443         /// <param name="direction">Direction in which to move around the arc.</param>
02444         /// <param name="linearDistance">Distance to move on the third axis in the base
      vector.</param>
02445         /// <param name="mode">Mode of operation. See the MHELICAL command documentation for more
      details.</param>
02446         /// <param name="transitionAngle">Transition angle (in Radians) for any filleted arcs that are
      generated.  Set to 0 if no transition curve is to be applied.</param>
02447         /// <param name="tableOutput">Output data to TABLE. 0 means do not output data to TABLE,
      otherwise is the TABLE start index for the data.</param>
02448         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02449         void MoveHelical(Int32 axis, double finishBase, double finishNext, double centreBase, double
      centreNext, UInt32 direction, double linearDistance, UInt32 mode, UInt32 transitionAngle = 0, UInt32
      tableOutput = 0);
02450
02451         /// <summary>
02452         /// Performs a helical move the same as MHELICAL and additionally allows vector speed to be
      changed when using multiple moves in the buffer.
02453         ///  Uses additional axis parameters FORCE_SPEED, ENDMOVE_SPEED and STARTMOVE_SPEED.
02454         /// </summary>
02455         /// <param name="axis">Base axis for this command. If value is -1 then the default base axis
      is used.</param>
02456         /// <param name="finishBase">Position on the base axis to finish at.</param>
02457         /// <param name="finishNext">Position on the next axis in the base vector to finish
      at.</param>
02458         /// <param name="centreBase">Position on the base axis about which to move.</param>
02459         /// <param name="centreNext">Position on the next axis in the base vector about which to
      move.</param>
02460         /// <param name="direction">Direction in which to move around the arc.</param>
02461         /// <param name="linearDistance">Distance to move on the third axis in the base
      vector.</param>
02462         /// <param name="mode">Mode of operation. See the MHELICAL command documentation for more
      details.</param>
02463         /// <param name="transitionAngle">Transition angle (in Radians) for any filleted arcs that are
      generated.  Set to 0 if no transition curve is to be applied.</param>
02464         /// <param name="tableOutput">Output data to TABLE. 0 means do not output data to TABLE,
      otherwise is the TABLE start index for the data.</param>
02465         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02466         void MoveHelicalSp(Int32 axis, double finishBase, double finishNext, double centreBase, double
      centreNext, UInt32 direction, double linearDistance, UInt32 mode, UInt32 transitionAngle = 0, UInt32
      tableOutput = 0);
02467
02468         /// <summary>
02469         /// Moves the three axis group defined in BASE along a spherical path with a vector speed
      determined by the SPEED set in the first axis of the BASE array.
02470         ///  There are 4 modes of operation with the option of finishing the move at an endpoint
      different to the start, or returning to the start point to complete a circle.
02471         ///  The path of the movement in 3D space can be defined either by specifying a point
      somewhere along the path, or by specifying the centre of the sphere.
02472         /// </summary>
02473         /// <param name="axis">Base axis for this command. If value is -1 then the default base axis
      is used.</param>
```

```
02474         /// <param name="endMidX">End position of the first axis</param>
02475         /// <param name="endMidY">End position of the second axis</param>
02476         /// <param name="endMidZ">End position of the third axis</param>
02477         /// <param name="centreX">Centre position of the first axis</param>
02478         /// <param name="centreY">Centre position of the second axis</param>
02479         /// <param name="centreZ">Centre position of the third axis</param>
02480         /// <param name="mode">Mode of operation. See the MSPHERICAL command documentation for more
      details.</param>
02481         /// <param name="gtpi">If this optional parameter is non zero, modes 0 and 1 will perform a
      move taking the opposite way around a 360 degree circle to the same endpoint</param>
02482         /// <param name="rotau">If this optional parameter is non zero, a fourth axis will perform
      linear interpolation at the same time as the spherical move. The axis is the next in the BASE
      sequence</param>
02483         /// <param name="rotav">If this optional parameter is non zero, a fifth axis will perform
      linear interpolation at the same time as the spherical move</param>
02484         /// <param name="rotaw">If this optional parameter is non zero, a sixth axis will perform
      linear interpolation at the same time as the spherical move</param>
02485         /// <param name="rotap">If this optional parameter is non zero, a seventh axis will perform
      linear interpolation at the same time as the spherical move</param>
02486         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02487         void MoveSpherical(Int32 axis, double endMidX, double endMidY, double endMidZ, double centreX,
      double centreY, double centreZ, UInt32 mode, UInt32 gtpi = 0, double rotau = 0, double rotav = 0,
      double rotaw = 0, double rotap = 0);
02488
02489         /// <summary>
02490         /// Performs a spherical move the same as MSPHERICAL and additionally allows vector speed to
      be changed when using multiple moves in the look ahead buffer
02491         ///  when MERGE = ON, using additional parameters FORCE_SPEED, ENDMOVE_SPEED and
      STARTMOVE_SPEED
02492         /// </summary>
02493         /// <param name="axis">Base axis for this command. If value is -1 then the default base axis
      is used.</param>
02494         /// <param name="endMidX">End position of the first axis</param>
02495         /// <param name="endMidY">End position of the second axis</param>
02496         /// <param name="endMidZ">End position of the third axis</param>
02497         /// <param name="centreX">Centre position of the first axis</param>
02498         /// <param name="centreY">Centre position of the second axis</param>
02499         /// <param name="centreZ">Centre position of the third axis</param>
02500         /// <param name="mode">Mode of operation. See the MSPHERICAL command documentation for more
      details.</param>
02501         /// <param name="gtpi">If this optional parameter is non zero, modes 0 and 1 will perform a
      move taking the opposite way around a 360 degree circle to the same endpoint</param>
02502         /// <param name="rotau">If this optional parameter is non zero, a fourth axis will perform
      linear interpolation at the same time as the spherical move. The axis is the next in the BASE
      sequence</param>
02503         /// <param name="rotav">If this optional parameter is non zero, a fifth axis will perform
      linear interpolation at the same time as the spherical move</param>
02504         /// <param name="rotaw">If this optional parameter is non zero, a sixth axis will perform
      linear interpolation at the same time as the spherical move</param>
02505         /// <param name="rotap">If this optional parameter is non zero, a seventh axis will perform
      linear interpolation at the same time as the spherical move</param>
02506         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02507         void MoveSphericalSp(Int32 axis, double endMidX, double endMidY, double endMidZ, double
      centreX, double centreY, double centreZ, UInt32 mode, UInt32 gtpi = 0, double rotau = 0, double rotav
      = 0, double rotaw = 0, double rotap = 0);
02508
02509         /// <summary>
02510         /// Function allows a sequence of relative movements to be loaded via TABLE values.
02511         ///  The moves can be automatically merged together using a circular or spherical arc.
02512         ///  Movement on 2 to 6 axes may be controlled in this way.
02513         /// </summary>
02514         /// <param name="axis">Base axis for this command. If value is -1 then the default base axis
      is used</param>
02515         /// <param name="tablePointer">Start of the move sequence in the TABLE data.</param>
02516         /// <param name="nAxes">Number of axes 2 to 6</param>
02517         /// <param name="nPoints">The number of points, each point requires 2 or 3 table
      values</param>
02518         /// <param name="options">Modifiers for the movements. See the TrioBASIC MOVEABSSEQ
      documentation for more details.</param>
02519         /// <param name="radius">The merging/filleting radius to be applied. 0 for no
      filleting</param>
02520         /// <param name="tableOutput">Output data to TABLE. 0 means do not output data to TABLE,
      otherwise is the TABLE start index for the data.</param>
02521         /// <param name="transitionAngle">Transition angle (in Radians) for any filleted arcs that are
      generated.  Set to 0 if no transition curve is to be applied.</param>
02522         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02523         void MoveRelSeq(Int32 axis, UInt32 tablePointer, UInt32 nAxes, UInt32 nPoints, UInt32 options,
      double radius, UInt32 tableOutput = 0, double transitionAngle = 0);
02524
02525         /// <summary>
02526         /// Function allows a sequence of absolute movements to be loaded via TABLE values.
02527         ///  The moves can be automatically merged together using a circular or spherical arc.
02528         ///  Movement on 2 to 6 axes may be controlled in this way.
02529         /// </summary>
```

```
02530         /// <param name="axis">Base axis for this command. If value is -1 then the default base axis
      is used</param>
02531         /// <param name="tablePointer">Start of the move sequence in the TABLE data.</param>
02532         /// <param name="nAxes">Number of axes 2 to 6</param>
02533         /// <param name="nPoints">The number of points, each point requires 2 or 3 table
      values</param>
02534         /// <param name="options">Modifiers for the movements. See the TrioBASIC MOVEABSSEQ
      documentation for more details.</param>
02535         /// <param name="radius">The merging/filleting radius to be applied. 0 for no
      filleting</param>
02536         /// <param name="tableOutput">Output data to TABLE. 0 means do not output data to TABLE,
      otherwise is the TABLE start index for the data.</param>
02537         /// <param name="transitionAngle">Transition angle (in Radians) for any filleted arcs that are
      generated.  Set to 0 if no transition curve is to be applied.</param>
02538         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02539         void MoveAbsSeq(Int32 axis, UInt32 tablePointer, UInt32 nAxes, UInt32 nPoints, UInt32 options,
      double radius, UInt32 tableOutput = 0, double transitionAngle = 0);
02540
02541         /// <summary>
02542         /// MOVEMODIFY will change the absolute end position of a single axis MOVE , MOVEABS , MOVESP
      , MOVEABSSP or MOVEMODIFY that is in the last position in the movement buffer.
02543         ///  If there is no motion command in the movement buffers or the last movement is not a
      single axis linear move, then MOVEMODIFY is loaded.
02544         /// </summary>
02545         /// <param name="position">Absolute position for the current move to complete at</param>
02546         /// <param name="axis">Base axis for this command. If value is -1 then the default base axis
      is used</param>
02547         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02548         void MoveModify(double position, Int32 axis = -1);
02549
02550         /// <summary>
02551         /// MOVEPICK is a new move type aimed at making it much simpler to program fast pick and place
      operations
02552         /// </summary>
02553         /// <param name="axis">Base axis for this command. If value is -1 then the default base axis
      is used</param>
02554         /// <param name="mode">Specify the number of axes 0 = xyz, 1 = xyzu, 2 = xyzuv, 3 =
      xyzuvw</param>
02555         /// <param name="x">Target x absolute position</param>
02556         /// <param name="y">Target y absolute position</param>
02557         /// <param name="z">Target z absolute position</param>
02558         /// <param name="withdraw">Incremental withdraw distance from the start position on z
      axis</param>
02559         /// <param name="overlapControlA">0..0.9 Specify rounding on corner A (optional)</param>
02560         /// <param name="overlapControlB">0..0.9 Specify rounding on corner B (optional)</param>
02561         /// <param name="u">Target u absolute position (optional)</param>
02562         /// <param name="v">Target v absolute position (optional)</param>
02563         /// <param name="w">Target w absolute position (optional)</param>
02564         /// <param name="withdrawControl">0..1 Adjust the speed for withdraw section
      (optional)</param>
02565         /// <param name="xyControl">0..1 Adjust the speed for xy section (optional)</param>
02566         /// <param name="approachControl">0..1 Adjust the speed for approach section
      (optional)</param>
02567         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02568         void MovePick(Int32 axis, UInt32 mode, double x, double y, double z, double withdraw, double
      overlapControlA = 0.5, double overlapControlB = 0.5, double u = 0, double v = 0, double w = 0, double
      withdrawControl = 1, double xyControl = 1, double approachControl = 1);
02569
02570         /// <summary>
02571         /// The linked move command is designed for controlling movements such as:
02572         ///  - Synchronisation to conveyors
02573         ///  - Flying shears
02574         ///  - Thread chasing, tapping etc.
02575         ///  - Coil winding
02576         ///  The motion consists of a linear movement with separately variable acceleration and
      deceleration phases linked via a software gearbox to the MEASURED position(MPOS) of another axis.The
      command uses the BASE and AXIS, and unit conversion factors in a similar way to other move commands.
02577         ///  The “link” axis may move in either direction to drive the output motion.The link
      distances specified are always positive.
02578         /// </summary>
02579         /// <param name="axis">Base axis for this command. If value is -1 then the default base axis
      is used</param>
02580         /// <param name="distance">Incremental distance in user units to be moved on the current base
      axis, as a result of the measured movement on the “input” axis which drives the move. </param>
02581         /// <param name="link_dist">Positive incremental distance in user units which is required to
      be measured on the “link” axis to result in the motion on the base axis.</param>
02582         /// <param name="link_acc">Positive incremental distance in user units on the input axis over
      which the base axis accelerates.</param>
02583         /// <param name="link_dec">Positive incremental distance in user units on the input axis over
      which the base axis decelerates.</param>
02584         /// <param name="link_axis">Specifies the axis to “link” to. It should be set to a value
      between 0 and the number of available axes.</param>
02585         /// <param name="link_options">
02586         /// Bit value options to customise how your MOVELINK operates:<br/>
```

```
02587         ///  Bit 0 (1) - link commences exactly when registration event MARK occurs on link axis<br/>
02588         ///  Bit 1 (2) - link commences at an absolute position on link axis(see link_pos for start
      position)<br/>
02589         ///  Bit 2 (4) - MOVELINK repeats automatically and bi - directionally when this bit is set.
      (This mode can be cleared by setting bit 1 of the REP_OPTION axis parameter)<br/>
02590         ///  Bit 4 (16)- If this bit is set the MOVELINK acceleration and deceleration phases are
      constructed using an "S" speed profile not a trapezoidal speed profile<br/>
02591         ///  Bit 5 (32)- Link is only active during a positive move on the link axis<br/>
02592         ///  Bit 8 (256) - link commences exactly when registration event MARKB occurs on link
      axis<br/>
02593         ///  Bit 9 (512) - link commences exactly when registration event R_MARK occurs on link axis.
      (see link_pos for channel number)<br/>
02594         ///  Bit 10 and 11 - S-Ramp mode selection for use when bit 4 is set.Figure in() shows peak
      acceleration compared to linear ramp.<br/>
02595         ///                 0 - SIN speed Profile(1.55)<br/>
02596         ///                 1 - Power 9 polynomial speed(2.42)<br/>
02597         ///                 2 - Power 7 polynomial speed(2.16)<br/>
02598         ///                 3 - Power 5 polynomial speed(1.86)<br/>
02599         ///  Bit 13 - Slave axis follows master axis DPOS<br/>
02600         ///  Bit 14 - Link is active during positive movement from a threshold.The threshold is
      updated to the highest positive value.
02601         /// </param>
02602         /// <param name="link_pos">Link_option bit 1 - the absolute position on the link axis in user
      UNITS where the MOVELINK is to be start. link_option bit 9 - the registration channel to start the
      movement on.</param>
02603         /// <param name="base_dist">
02604         /// Firmware V2.0253 onwards. The "base distance" is the distance that is to be travelled at
      the base speed and is part of the total first "distance" parameter. See Examples 4 and 5 below.<br/>
02605         ///  MOVELINK_MODIFY cannot be used when the base distance is in use.
02606         /// </param>
02607         /// <param name="y_dist">Same as base_dist, for 2nd base axis</param>
02608         /// <param name="z_dist">Same as base_dist, for 3rd base axis</param>
02609         /// <param name="u_dist">Same as base_dist, for 4th base axis</param>
02610         /// <param name="v_dist">Same as base_dist, for 5th base axis</param>
02611         /// <param name="w_dist">Same as base_dist, for 6th base axis</param>
02612         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02613         void MoveLink(Int32 axis, double distance, double link_dist, double link_acc, double link_dec,
      int link_axis = -1, UInt32 link_options = 0, double link_pos = 0, double base_dist = 0, double y_dist
      = 0, double z_dist = 0, double u_dist = 0, double v_dist = 0, double w_dist = 0);
02614
02615         /// <summary>
02616         /// Function used to pre-set the parameter list for a subsequent move using just one command.
02617         /// </summary>
02618         /// <param name="axis">Target axis for whose parameters are to be modified</param>
02619         /// <param name="speed">The value for SPEED to be set</param>
02620         /// <param name="accel">The value for ACCEL to be set</param>
02621         /// <param name="decel">The value for DECEL to be set</param>
02622         /// <param name="jerk">The value for JERK to be set</param>
02623         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02624         void SetMotion(Int32 axis, double speed, double accel, double decel, double jerk);
02625
02626         /// <summary>
02627         /// Function used to pre-set the parameter list for a subsequent move using just one command.
02628         /// </summary>
02629         /// <param name="axis">Target axis for whose parameters are to be modified</param>
02630         /// <param name="speed">The value for SPEED to be set</param>
02631         /// <param name="accel">The value for ACCEL to be set</param>
02632         /// <param name="decel">The value for DECEL to be set</param>
02633         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02634         void SetMotion(Int32 axis, double speed, double accel, double decel);
02635
02636         /// <summary>
02637         /// Function used to pre-set the parameter list for a subsequent move using just one command.
02638         /// </summary>
02639         /// <param name="axis">Target axis for whose parameters are to be modified</param>
02640         /// <param name="speed">The value for SPEED to be set</param>
02641         /// <param name="accel">The value for ACCEL to be set</param>
02642         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02643         void SetMotion(Int32 axis, double speed, double accel);
02644
02645         /// <summary>
02646         /// Function used to pre-set the parameter list for a subsequent move using just one command.
02647         /// </summary>
02648         /// <param name="axis">Target axis for whose parameters are to be modified</param>
02649         /// <param name="speed">The value for SPEED to be set</param>
02650         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02651         void SetMotion(Int32 axis, double speed);
02652
02653         /// <summary>
02654         /// SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP,
      MOVECIRCSP etc. using just one command.
02655         /// </summary>
```

```
02656         /// <param name="axis">Target axis for whose parameters are to be modified</param>
02657         /// <param name="forceSpeed">The value for FORCE_SPEED to be set</param>
02658         /// <param name="startMoveSpeed">The value for STARTMOVE_SPEED to be set</param>
02659         /// <param name="endMoveSpeed">The value for ENDMOVE_SPEED to be set</param>
02660         /// <param name="forceAccel">The value for FORCE_ACCEL to be set</param>
02661         /// <param name="forceDecel">The value for FORCE_DECEL to be set</param>
02662         /// <param name="forceJerk">The value for FORCE_JERK to be set</param>
02663         /// <param name="forceRamp">The value for FORCE_RAMP to be set</param>
02664         /// <param name="forceDwell">The value for FORCE_DWELL to be set</param>
02665         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02666         void SetMotionSp(Int32 axis, double forceSpeed, double startMoveSpeed, double endMoveSpeed,
      double forceAccel, double forceDecel, double forceJerk, double forceRamp, double forceDwell);
02667
02668         /// <summary>
02669         /// SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP,
      MOVECIRCSP etc. using just one command.
02670         /// </summary>
02671         /// <param name="axis">Target axis for whose parameters are to be modified</param>
02672         /// <param name="forceSpeed">The value for FORCE_SPEED to be set</param>
02673         /// <param name="startMoveSpeed">The value for STARTMOVE_SPEED to be set</param>
02674         /// <param name="endMoveSpeed">The value for ENDMOVE_SPEED to be set</param>
02675         /// <param name="forceAccel">The value for FORCE_ACCEL to be set</param>
02676         /// <param name="forceDecel">The value for FORCE_DECEL to be set</param>
02677         /// <param name="forceJerk">The value for FORCE_JERK to be set</param>
02678         /// <param name="forceRamp">The value for FORCE_RAMP to be set</param>
02679         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02680         void SetMotionSp(Int32 axis, double forceSpeed, double startMoveSpeed, double endMoveSpeed,
      double forceAccel, double forceDecel, double forceJerk, double forceRamp);
02681
02682         /// <summary>
02683         /// SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP,
      MOVECIRCSP etc. using just one command.
02684         /// </summary>
02685         /// <param name="axis">Target axis for whose parameters are to be modified</param>
02686         /// <param name="forceSpeed">The value for FORCE_SPEED to be set</param>
02687         /// <param name="startMoveSpeed">The value for STARTMOVE_SPEED to be set</param>
02688         /// <param name="endMoveSpeed">The value for ENDMOVE_SPEED to be set</param>
02689         /// <param name="forceAccel">The value for FORCE_ACCEL to be set</param>
02690         /// <param name="forceDecel">The value for FORCE_DECEL to be set</param>
02691         /// <param name="forceJerk">The value for FORCE_JERK to be set</param>
02692         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02693         void SetMotionSp(Int32 axis, double forceSpeed, double startMoveSpeed, double endMoveSpeed,
      double forceAccel, double forceDecel, double forceJerk);
02694
02695         /// <summary>
02696         /// SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP,
      MOVECIRCSP etc. using just one command.
02697         /// </summary>
02698         /// <param name="axis">Target axis for whose parameters are to be modified</param>
02699         /// <param name="forceSpeed">The value for FORCE_SPEED to be set</param>
02700         /// <param name="startMoveSpeed">The value for STARTMOVE_SPEED to be set</param>
02701         /// <param name="endMoveSpeed">The value for ENDMOVE_SPEED to be set</param>
02702         /// <param name="forceAccel">The value for FORCE_ACCEL to be set</param>
02703         /// <param name="forceDecel">The value for FORCE_DECEL to be set</param>
02704         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02705         void SetMotionSp(Int32 axis, double forceSpeed, double startMoveSpeed, double endMoveSpeed,
      double forceAccel, double forceDecel);
02706
02707         /// <summary>
02708         /// SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP,
      MOVECIRCSP etc. using just one command.
02709         /// </summary>
02710         /// <param name="axis">Target axis for whose parameters are to be modified</param>
02711         /// <param name="forceSpeed">The value for FORCE_SPEED to be set</param>
02712         /// <param name="startMoveSpeed">The value for STARTMOVE_SPEED to be set</param>
02713         /// <param name="endMoveSpeed">The value for ENDMOVE_SPEED to be set</param>
02714         /// <param name="forceAccel">The value for FORCE_ACCEL to be set</param>
02715         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02716         void SetMotionSp(Int32 axis, double forceSpeed, double startMoveSpeed, double endMoveSpeed,
      double forceAccel);
02717
02718         /// <summary>
02719         /// SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP,
      MOVECIRCSP etc. using just one command.
02720         /// </summary>
02721         /// <param name="axis">Target axis for whose parameters are to be modified</param>
02722         /// <param name="forceSpeed">The value for FORCE_SPEED to be set</param>
02723         /// <param name="startMoveSpeed">The value for STARTMOVE_SPEED to be set</param>
02724         /// <param name="endMoveSpeed">The value for ENDMOVE_SPEED to be set</param>
02725         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02726         void SetMotionSp(Int32 axis, double forceSpeed, double startMoveSpeed, double endMoveSpeed);
```

```
02727
02728         /// <summary>
02729         /// SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP,
      MOVECIRCSP etc. using just one command.
02730         /// </summary>
02731         /// <param name="axis">Target axis for whose parameters are to be modified</param>
02732         /// <param name="forceSpeed">The value for FORCE_SPEED to be set</param>
02733         /// <param name="startMoveSpeed">The value for STARTMOVE_SPEED to be set</param>
02734         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02735         void SetMotionSp(Int32 axis, double forceSpeed, double startMoveSpeed);
02736
02737         /// <summary>
02738         /// SET_MOTIONSP is used to pre-set the parameter list for a subsequent MOVESP, MOVEABSSP,
      MOVECIRCSP etc. using just one command.
02739         /// </summary>
02740         /// <param name="axis">Target axis for whose parameters are to be modified</param>
02741         /// <param name="forceSpeed">The value for FORCE_SPEED to be set</param>
02742         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02743         void SetMotionSp(Int32 axis, double forceSpeed);
02744
02745         /// <summary>
02746         /// Set single table value.
02747         /// </summary>
02748         /// <param name="index">Index of the table entry to be written</param>
02749         /// <param name="value">New value to be set</param>
02750         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02751         void SetTableValue(UInt32 index, double value);
02752
02753         /// <summary>
02754         /// Get single table value.
02755         /// </summary>
02756         /// <param name="index">Index of the table entry to be read</param>
02757         /// <returns>On success the value of table entry returned</returns>
02758         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02759         double GetTableValue(UInt32 index);
02760
02761         /// <summary>
02762         /// Set multiple table entries.
02763         /// </summary>
02764         /// <param name="start_index">Start index of the table range to be written</param>
02765         /// <param name="count">Number of table entries to be written</param>
02766         /// <param name="values">Array of values to be written</param>
02767         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02768         void SetMultiTableValues(UInt32 start_index, UInt32 count, double[] values);
02769
02770         /// <summary>
02771         /// Get multiple table entries.
02772         /// </summary>
02773         /// <param name="start_index">Start index of the table range to be read</param>
02774         /// <param name="count">Number of table entries to be read</param>
02775         /// <returns>Placeholder array for the values to be read</returns>
02776         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02777         double[] GetMultiTableValues(UInt32 start_index, UInt32 count);
02778
02779         /// <summary>
02780         /// Set single VR value.
02781         /// </summary>
02782         /// <param name="index">Index of the VR entry to be written</param>
02783         /// <param name="value">New value to be written</param>
02784         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02785         void SetVrValue(UInt32 index, double value);
02786
02787         /// <summary>
02788         /// Get single VR value.
02789         /// </summary>
02790         /// <param name="index">Index of the VR entry to be read</param>
02791         /// <returns>On success the value VR entry returned</returns>
02792         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02793         double GetVrValue(UInt32 index);
02794
02795         /// <summary>
02796         /// Set multiple VR entries.
02797         /// </summary>
02798         /// <param name="start_index">Start index of the VR range to be written</param>
02799         /// <param name="count">Number of VR entries to be written</param>
02800         /// <param name="values">Array of VR values to be written</param>
02801         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02802         void SetMultiVrValues(UInt32 start_index, UInt32 count, double[] values);
```

```
02803
02804         /// <summary>
02805         /// Get multiple VR entries.
02806         /// </summary>
02807         /// <param name="start_index">Start index of the VR range to be read</param>
02808         /// <param name="count">Number of VR entries to be read</param>
02809         /// <returns>Placeholder array for the values to be read</returns>
02810         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02811         double[] GetMultiVrValues(UInt32 start_index, UInt32 count);
02812
02813         /// <summary>
02814         /// The INVERT_IN command allows the input channels to be individually inverted in software.
02815         /// </summary>
02816         /// <param name="input">The input to invert</param>
02817         /// <param name="invert">True to invert, false not to invert</param>
02818         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02819         void SetInvertIN(UInt32 input, bool invert);
02820
02821         /// <summary>
02822         /// Gets the invert status set by INVERT_IN command.
02823         /// </summary>
02824         /// <param name="input">The input to invert</param>
02825         /// <returns>True if inverted, false if not inverted</returns>
02826         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02827         bool GetInvertIN(UInt32 input);
02828
02829         /// <summary>
02830         /// The RAPIDSTOP command operates like the CANCEL command but does so for ALL axes using a
      subset of cancel modes
02831         /// </summary>
02832         /// <param name="mode">
02833         ///  0 Cancels axis commands from the MTYPE buffers(default)<br/>
02834         ///  1 Cancels all buffered moves on all axis(excluding the PMOVE)<br/>
02835         ///  2 Cancels all active and buffered moves including the PMOVE
02836         /// </param>
02837         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02838         void RapidStop(Int32 mode = 0);
02839
02840         /// <summary>
02841         /// Function returns the list of programs and files in internal memory of the Motion
      Coordinator
02842         /// </summary>
02843         /// <param name="handler">Callback handler function executed for each item in internal
      memory</param>
02844         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02845         void Dir(FileCallback handler);
02846
02847         /// <summary>
02848         /// Set a null-terminated string into consecutive VRs
02849         /// </summary>
02850         /// <param name="vr">Starting VR index</param>
02851         /// <param name="str">String to be set</param>
02852         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02853         void SetVrString(UInt32 vr, String str);
02854
02855         /// <summary>
02856         /// Get the length of a null-terminated string from consecutive VRs
02857         /// </summary>
02858         /// <param name="vr">Starting VR index</param>
02859         /// <returns>Length of the string</returns>
02860         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02861         UInt32 GetVrStringLen(UInt32 vr);
02862
02863         /// <summary>
02864         /// Get a null-terminated string from consecutive VRs
02865         /// </summary>
02866         /// <param name="vr">Starting VR index</param>
02867         /// <param name="buff">Buffer to receive the string characters. If NULL function succeeds and
      returns expected buffers size in buff_size parameter including null terminator character</param>
02868         /// <param name="buff_size">Size of the buffer on input and on exit required buffer size
      including null terminator</param>
02869         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02870         unsafe void GetVrString(UInt32 vr, Byte* buff, ref UInt32 buff_size);
02871
02872         /// <summary>
02873         /// Get a null-terminated string from consecutive VRs using callback function to re-construct
      the string from received chunks
02874         /// </summary>
02875         /// <param name="vr">Starting VR index</param>
```

```
02876         /// <param name="handler">Callback function to receive string chunks</param>
02877         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02878         void GetVrString(UInt32 vr, StreamHandlerCallback handler);
02879
02880         /// <summary>
02881         /// Get a null-terminated string from consecutive VRs
02882         /// </summary>
02883         /// <param name="vr">Starting VR index</param>
02884         /// <returns>On success the string stored in consecutive VR entries</returns>
02885         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02886         String GetVrString(UInt32 vr);
02887
02888         /// <summary>
02889         /// Get a value from an Analogue Input.
02890         /// </summary>
02891         /// <param name="channel">Analogue input channel number</param>
02892         /// <returns>Current value of the analog input channel</returns>
02893         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02894         Int32 GetAIn(UInt32 channel);
02895
02896         /// <summary>
02897         /// Get the state of the Analogue Output
02898         /// </summary>
02899         /// <param name="channel">Analogue output channel number</param>
02900         /// <returns>Current value of the analogue output channel</returns>
02901         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02902         Int32 GetAOut(UInt32 channel);
02903
02904         /// <summary>
02905         /// Set the state of the Analogue Output
02906         /// </summary>
02907         /// <param name="channel">Analogue output channel number</param>
02908         /// <param name="value">New value for the analogue output channel</param>
02909         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02910         void SetAOut(UInt32 channel, Int32 value);
02911
02912         /// <summary>
02913         /// Get the state of the Digital Inputs
02914         /// </summary>
02915         /// <param name="firstChannel">First digital input channel number</param>
02916         /// <param name="lastChannel">Last digital input channel number. The difference between last
      and first channel should be at most 32</param>
02917         /// <returns>Binary sum of digital input states</returns>
02918         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02919         UInt32 GetDIn(UInt32 firstChannel, UInt32 lastChannel);
02920
02921         /// <summary>
02922         /// Get the state of the Digital Input
02923         /// </summary>
02924         /// <param name="channel">Digital input channel number</param>
02925         /// <returns>Current state of the digital input channel</returns>
02926         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02927         bool GetDIn(UInt32 channel);
02928
02929         /// <summary>
02930         /// Get the state of the first 32 Digital Inputs
02931         /// </summary>
02932         /// <returns>Bit sum of the states of the first 32 digital input channels</returns>
02933         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02934         UInt32 GetDIn32();
02935
02936         /// <summary>
02937         /// Get the state of the Digital Outputs
02938         /// </summary>
02939         /// <param name="firstChannel">First digital output channel number</param>
02940         /// <param name="lastChannel">Last digital output channel number. The difference between last
      and first channel should be at most 32</param>
02941         /// <returns>Binary sum of digital output states</returns>
02942         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02943         UInt32 GetDOut(UInt32 firstChannel, UInt32 lastChannel);
02944
02945         /// <summary>
02946         /// Get the state of the Digital Output
02947         /// </summary>
02948         /// <param name="channel">Digital output channel number</param>
02949         /// <returns>Current state of the digital output channel</returns>
02950         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
```

```
02951         bool GetDOut(UInt32 channel);
02952
02953         /// <summary>
02954         /// Get the state of the first 32 Digital Outputs
02955         /// </summary>
02956         /// <returns>Bit sum of the states of the first 32 digital output channels</returns>
02957         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02958         UInt32 GetDOut32();
02959
02960         /// <summary>
02961         /// Set the state of the Digital Outputs
02962         /// </summary>
02963         /// <param name="firstChannel">First digital output channel number</param>
02964         /// <param name="lastChannel">Last digital output channel number</param>
02965         /// <param name="states">Binary sum of digital output states</param>
02966         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02967         void SetDOut(UInt32 firstChannel, UInt32 lastChannel, UInt32 states);
02968
02969         /// <summary>
02970         /// Set the state of the Digital Output
02971         /// </summary>
02972         /// <param name="channel">Digital output channel number</param>
02973         /// <param name="state">New state for the digital output channel</param>
02974         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02975         void SetDOut(UInt32 channel, bool state);
02976
02977         /// <summary>
02978         /// Set the state of the first 32 Digital Outputs
02979         /// </summary>
02980         /// <param name="states32">Bit sum of new the states for the first 32 digital output
      channels</param>
02981         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02982         void SetDOut32(UInt32 states32);
02983
02984         /// <summary>
02985         /// Stops a program at its current line.
02986         /// </summary>
02987         /// <param name="name">Name of the program to be stopped.</param>
02988         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02989         void StopProgram(String name);
02990
02991         /// <summary>
02992         /// Stops a program at its current line.
02993         /// </summary>
02994         /// <param name="name">Name of the program to be stopped.</param>
02995         /// <param name="process_number">Optional process number to be used when multiple instances of
      the program are running and only one is to be stopped.</param>
02996         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
02997         void StopProgram(String name, Int32 process_number);
02998
02999         /// <summary>
03000         /// Adds the output from the servo control block of a secondary axis to the output of the base
      axis.
03001         ///  The resulting DAC_OUT of the base axis is then the sum of the two control loop outputs.
03002         /// </summary>
03003         /// <param name="axis">Base axis for to be subject to DAC_OUT superimposition</param>
03004         /// <param name="secondAxis">Axis which DAC_OUT value to superimpose. -1 breaks the link with
      the other axis. </param>
03005         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03006         void AddDac(Int32 axis, Int32 secondAxis);
03007
03008         /// <summary>
03009         /// Adds the output from the servo control block of a secondary axis to the output of the base
      axis.
03010         ///  The resulting DAC_OUT of the base axis is then the sum of the two control loop outputs.
03011         ///  Axis previously set by SetAxisBase() is subject to superimposition of DAC_OUT value.
03012         /// </summary>
03013         /// <param name="secondAxis">Axis which DAC_OUT value to superimpose. -1 breaks the link with
      the other axis. </param>
03014         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03015         void AddDac(Int32 secondAxis);
03016
03017         /// <summary>
03018         /// Function is used to superimpose 2 or more movements to build up a more complex movement
      profile.
03019         /// </summary>
03020         /// <param name="axis">Target axis for to be subject to superimposition</param>
03021         /// <param name="linkAxis">Axis to superimpose. -1 breaks the link with the other axis.
      </param>
```

```
03022         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03023         void AddAx(Int32 axis, Int32 linkAxis);
03024
03025         /// <summary>
03026         /// Function is used to superimpose 2 or more movements to build up a more complex movement
      profile.
03027         ///  Axis previously set by SetAxisBase() is subject to superimposition.
03028         /// </summary>
03029         /// <param name="linkAxis">Axis to superimpose. -1 breaks the link with the other axis.
      </param>
03030         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03031         void AddAx(Int32 linkAxis);
03032
03033         /// <summary>
03034         /// Sets all global (numbered) variables and VR values to 0 and sets local variables on the
      process on which command is run to 0.
03035         /// </summary>
03036         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03037         void ClearVariables();
03038
03039         /// <summary>
03040         /// Delete frame group. When the group is deleted FRAME is set to 0. See the TrioBASIC
      FRAME_GROUP documentation for more information
03041         /// </summary>
03042         /// <param name="group">Frame group number to delete</param>
03043         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03044         void DeleteFrameGroup(UInt32 group);
03045
03046         /// <summary>
03047         /// Define new frame group.
03048         /// </summary>
03049         /// <param name="group">Frame group number to define</param>
03050         /// <param name="tableOffset">Start offset of TABLE range used by the frame group</param>
03051         /// <param name="axes">Array of axes to be added to the frame group</param>
03052         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03053         void SetFrameGroup(UInt32 group, UInt32 tableOffset, Int32[] axes);
03054
03055         /// <summary>
03056         /// Get frame group info.
03057         /// </summary>
03058         /// <param name="group">Frame group number to get info</param>
03059         /// <returns>
03060         /// On success a tuple is returned containing following:<br/>
03061         ///  - Current start offset of TABLE range used by the frame group<br/>
03062         ///  - Array of axis indices currently belonging to the frame group
03063         /// </returns>
03064         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03065         Tuple<UInt32, Int32[]> GetFrameGroup(UInt32 group);
03066
03067         /// <summary>
03068         /// Initialise EtherCAT network, and put it onto operational mode. See Trio BASIC help for
      more details.
03069         /// </summary>
03070         /// <param name="slot">Slot number of EtherCAT port.</param>
03071         /// <param name="options">
03072         ///  bit 0: 0-Do not print progress information; 1-Print out progress information
      (default)<br/>
03073         ///  bit 1: 0-Normal state change (default); 1-Force a re-scan of the network<br/>
03074         ///  bit 2: 0-Standard startup; 1-Express startup; do not wait for PLL to lock between state
      changes
03075         /// </param>
03076         /// <param name="MAC_retries">Sets the number of times the master attempts to restart the
      Ethernet auto-negotiation. Default = 2</param>
03077         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03078         void Ethercat_StartNetwork(Int32 slot, EthercatStartOption options, UInt32 MAC_retries);
03079
03080         /// <summary>
03081         /// Initialise EtherCAT network, and put it onto operational mode. See Trio BASIC help for
      more details.
03082         /// </summary>
03083         /// <param name="slot">Slot number of EtherCAT port.</param>
03084         /// <param name="options">
03085         ///  bit 0: 0-Do not print progress information; 1-Print out progress information
      (default)<br/>
03086         ///  bit 1: 0-Normal state change (default); 1-Force a re-scan of the network<br/>
03087         ///  bit 2: 0-Standard startup; 1-Express startup; do not wait for PLL to lock between state
      changes
03088         /// </param>
03089         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
```

```
03090         void Ethercat_StartNetwork(Int32 slot, EthercatStartOption options);
03091
03092         /// <summary>
03093         /// Initialise EtherCAT network, and put it onto operational mode. See Trio BASIC help for
      more details.
03094         /// </summary>
03095         /// <param name="slot">Slot number of EtherCAT port.</param>
03096         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03097         void Ethercat_StartNetwork(Int32 slot);
03098
03099         /// <summary>
03100         /// Closedown the EtherCAT network.
03101         /// </summary>
03102         /// <param name="slot">Slot number of EtherCAT port.</param>
03103         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03104         void Ethercat_StopNetwork(Int32 slot);
03105
03106         /// <summary>
03107         /// Check whether given slave is online. See Trio BASIC help for more details.
03108         /// </summary>
03109         /// <param name="slot">Slot number of EtherCAT port.</param>
03110         /// <param name="slave">Position on the network of the slave </param>
03111         /// <param name="out_vr_index">Result is placed in the selected VR variable</param>
03112         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03113         void Ethercat_CheckSlaveOnline_VR(Int32 slot, Int32 slave, UInt32 out_vr_index);
03114
03115         /// <summary>
03116         /// Check whether given slave is online. See Trio BASIC help for more details.
03117         /// </summary>
03118         /// <param name="slot">Slot number of EtherCAT port.</param>
03119         /// <param name="slave">Position on the network of the slave </param>
03120         /// <returns>On success true if slave is online or false otherwise</returns>
03121         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03122         bool Ethercat_CheckSlaveOnline(Int32 slot, Int32 slave);
03123
03124         /// <summary>
03125         /// Determine the number of slaves on the bus. See Trio BASIC help for more details.
03126         /// </summary>
03127         /// <param name="slot">Slot number of EtherCAT port.</param>
03128         /// <param name="out_vr_index">Result is placed in the selected VR variable</param>
03129         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03130         void Ethercat_CheckNumberOfSlaves_VR(Int32 slot, UInt32 out_vr_index);
03131
03132         /// <summary>
03133         /// Determine the number of slaves on the bus. See Trio BASIC help for more details.
03134         /// </summary>
03135         /// <param name="slot">Slot number of EtherCAT port.</param>
03136         /// <returns>On success receives the number of slaves</returns>
03137         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03138         UInt32 Ethercat_CheckNumberOfSlaves(Int32 slot);
03139
03140         /// <summary>
03141         /// Determine the configured address of the slave identified by its position on the network.
      See Trio BASIC help for more details.
03142         /// </summary>
03143         /// <param name="slot">Slot number of EtherCAT port.</param>
03144         /// <param name="slave">Position on the network of the slave </param>
03145         /// <param name="out_vr_index">Result is placed in the selected VR variable</param>
03146         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03147         void Ethercat_GetSlaveAddress_VR(Int32 slot, Int32 slave, UInt32 out_vr_index);
03148
03149         /// <summary>
03150         /// Determine the configured address of the slave identified by its position on the network.
      See Trio BASIC help for more details.
03151         /// </summary>
03152         /// <param name="slot">Slot number of EtherCAT port.</param>
03153         /// <param name="slave">Position on the network of the slave </param>
03154         /// <returns>On success receives the configured slave address</returns>
03155         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03156         Int32 Ethercat_GetSlaveAddress(Int32 slot, Int32 slave);
03157
03158         /// <summary>
03159         /// Determine the axis of the slave identified by its position on the network. See Trio BASIC
      help for more details.
03160         /// </summary>
03161         /// <param name="slot">Slot number of EtherCAT port.</param>
03162         /// <param name="slave">Position on the network of the slave </param>
03163         /// <param name="out_vr_index">Result is placed in the selected VR variable</param>
03164         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
```

```
module.</exception>
03165         void Ethercat_GetSlaveAxis_VR(Int32 slot, Int32 slave, UInt32 out_vr_index);
03166
03167         /// <summary>
03168         /// Determine the axis of the slave identified by its position on the network. See Trio BASIC
help for more details.
03169         /// </summary>
03170         /// <param name="slot">Slot number of EtherCAT port.</param>
03171         /// <param name="slave">Position on the network of the slave </param>
03172         /// <returns>On success receives the axis assigned to the slave</returns>
03173         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
03174         Int32 Ethercat_GetSlaveAxis(Int32 slot, Int32 slave);
03175
03176         /// <summary>
03177         /// This function controls the EtherCAT State Machine (ESM).
03178         ///  It requests the master change to given EtherCAT 'state', and hence changes all slaves to
the same state. When a change to a higher state is made, the EtherCAT network will progress to the new
state through the in-between states to allow correct starting of the network. See Trio BASIC help for
more details.
03179         /// </summary>
03180         /// <param name="slot">Slot number of EtherCAT port.</param>
03181         /// <param name="state">
03182         ///  -1 : Reserved<br/>
03183         ///  0 : Initial (EtherCAT ESC value 0x01)<br/>
03184         ///  1 : Pre-Operational (0x02)<br/>
03185         ///  2 : Safe-Operational  (0x04)<br/>
03186         ///  3 : Operational  (0x08)
03187         /// </param>
03188         /// <param name="display">
03189         ///  0 : Writes state change information to the standard output stream.<br/>
03190         ///  1 : Do not write out state change information.
03191         /// </param>
03192         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
03193         void Ethercat_SetState(Int32 slot, EthercatState state, UInt32 display);
03194
03195         /// <summary>
03196         /// Gets the present state of the EtherCAT running on the defined slot. See Trio BASIC help
for more details.
03197         /// </summary>
03198         /// <param name="slot">Slot number of EtherCAT port.</param>
03199         /// <param name="out_vr_index">
03200         /// Result is placed in the selected VR variable. The returned value means:<br/>
03201         ///  0 : Initial (EtherCAT ESC value 0x01)<br/>
03202         ///  1 : Pre-Operational (0x02)<br/>
03203         ///  2 : Safe-Operational  (0x04)<br/>
03204         ///  3 : Operational  (0x08)
03205         /// </param>
03206         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
03207         void Ethercat_GetState_VR(Int32 slot, UInt32 out_vr_index);
03208
03209         /// <summary>
03210         /// Gets the present state of the EtherCAT running on the defined slot. See Trio BASIC help
for more details.
03211         /// </summary>
03212         /// <param name="slot">Slot number of EtherCAT port.</param>
03213         /// <returns>
03214         /// On success redeives the state. The returned value means:<br/>
03215         ///  0 : Initial (EtherCAT ESC value 0x01)<br/>
03216         ///  1 : Pre-Operational (0x02)<br/>
03217         ///  2 : Safe-Operational  (0x04)<br/>
03218         ///  3 : Operational  (0x08)
03219         /// </returns>
03220         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
03221         EthercatState Ethercat_GetState(Int32 slot);
03222
03223         /// <summary>
03224         /// Low-level write to the slave's ESC registers, addressing method used is auto-increment
addressing. See Trio BASIC help for more details.
03225         /// </summary>
03226         /// <param name="slot">Slot number of EtherCAT port.</param>
03227         /// <param name="slave">Auto-increment address </param>
03228         /// <param name="esc_offset">Register address</param>
03229         /// <param name="data_length">Number of data bytes to write</param>
03230         /// <param name="data">Data values</param>
03231         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
03232         unsafe void Ethercat_SetSlave_ESC_Register(Int32 slot, Int32 slave, UInt32 esc_offset, UInt32
data_length, Byte* data);
03233
03234         /// <summary>
03235         /// Low-level read from the slave's ESC registers, addressing method used is auto-increment
addressing. See Trio BASIC help for more details.
03236         /// </summary>
```

```
03237         /// <param name="slot">Slot number of EtherCAT port.</param>
03238         /// <param name="slave">Auto-increment address</param>
03239         /// <param name="esc_offset">Register address</param>
03240         /// <param name="data_length">Number of bytes in the register size</param>
03241         /// <param name="out_vr_index">Result is placed in the selected VR variable</param>
03242         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03243         void Ethercat_GetSlave_ESC_Register(Int32 slot, Int32 slave, UInt32 esc_offset, UInt32
      data_length, UInt32 out_vr_index);
03244
03245         /// <summary>
03246         /// Reset a slave error. This function runs the error reset sequence on the drive control
      word. DRIVE_CONTROLWORD bit 8 is toggled high then low. This will instruct the drive to reset any
      errors in the drive where the cause of the error has been removed.
03247         ///  See Trio BASIC help for more details.
03248         /// </summary>
03249         /// <param name="axis">The axis number of the drive to be reset</param>
03250         /// <param name="mode">
03251         ///  0 : The 'Fault Reset' (bit 7) of DS402 control word is set high and then set low again
      after a hard coded timeout. (default)<br/>
03252         ///  1 : Bit 7 is set high until the 'Fault Flag' (bit 3) of the status word goes low, or a
      timeout occurs.
03253         /// </param>
03254         /// <param name="timeout">Timeout in ms used during mode 1 operation. Default is 100 ms. Range
      is 1 to 10000 ms.</param>
03255         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03256         void Ethercat_SendResetSequence(Int32 axis, UInt32 mode, UInt32 timeout);
03257
03258         /// <summary>
03259         /// Reset a slave error. This function runs the error reset sequence on the drive control
      word. DRIVE_CONTROLWORD bit 8 is toggled high then low. This will instruct the drive to reset any
      errors in the drive where the cause of the error has been removed.
03260         ///  See Trio BASIC help for more details.
03261         /// </summary>
03262         /// <param name="axis">The axis number of the drive to be reset</param>
03263         /// <param name="mode">
03264         ///  0 : The 'Fault Reset' (bit 7) of DS402 control word is set high and then set low again
      after a hard coded timeout. (default)<br/>
03265         ///  1 : Bit 7 is set high until the 'Fault Flag' (bit 3) of the status word goes low, or a
      timeout occurs.
03266         /// </param>
03267         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03268         /// <remarks>
03269         /// Default timeout is 100 ms
03270         /// </remarks>
03271         void Ethercat_SendResetSequence(Int32 axis, UInt32 mode);
03272
03273         /// <summary>
03274         /// Reset a slave error. This function runs the error reset sequence on the drive control
      word. DRIVE_CONTROLWORD bit 8 is toggled high then low. This will instruct the drive to reset any
      errors in the drive where the cause of the error has been removed.
03275         ///  See Trio BASIC help for more details.
03276         /// </summary>
03277         /// <param name="axis">The axis number of the drive to be reset</param>
03278         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03279         /// <remarks>
03280         /// Default timeout is 100 ms
03281         /// </remarks>
03282         void Ethercat_SendResetSequence(Int32 axis);
03283
03284         /// <summary>
03285         /// This command enables the user to define a CoE Statusword (0x6041:00) mask and trigger
      pattern for raising the controller AXISSTATUS 'Remote Drive Status word Error' flag (bit 26.)
03286         ///  The mask defines which bits within the CoE status word are relevant, and the trigger
      pattern defines the state of these bits(0 or 1) which will trigger the controller AXISSTATUS flag.
03287         ///  If it is necessary to reset the watchdog(master enable) when this AXISISTATUS flag is set
      then the axis ERRORMASK must be set accordingly.
03288         ///  The slave mask and pattern can be set and modified from preop to operational state of the
      network(after the configured addresses have been assigned.) It must be reset if the network is
      restarted.
03289         ///  The slave must support the CoE statusword(0x6041:00) within the PDO cyclic data.
03290         ///  If the Mask is zero the controller will not execute any runtime check of the CoE
      statusword.
03291         ///  If the watchdog must be cleared to OFF when the 'remote drive statusword error' flag is
      raised then bit 26 of the axis ERRORMASK parameter must be set.
03292         ///  See Trio BASIC help for more details.
03293         /// </summary>
03294         /// <param name="slot">Slot number of EtherCAT port.</param>
03295         /// <param name="addr">Configured address of the slave</param>
03296         /// <param name="vr_index">VR index in range [0..max_vr_index-1]. The index of the VR which
      contains the mask and the next VR (vr_index+1) contains the trigger_pattern.</param>
03297         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03298         void Ethercat_SetStatusMaskAndTriggerPattern_VR(Int32 slot, Int32 addr, UInt32 vr_index);
```

```
03299
03300         /// <summary>
03301         /// This command enables the user to define a CoE Statusword (0x6041:00) mask and trigger
      pattern for raising the controller AXISSTATUS 'Remote Drive Status word Error' flag (bit 26.)
03302         ///  The mask defines which bits within the CoE status word are relevant, and the trigger
      pattern defines the state of these bits(0 or 1) which will trigger the controller AXISSTATUS flag.
03303         ///  If it is necessary to reset the watchdog(master enable) when this AXISISTATUS flag is set
      then the axis ERRORMASK must be set accordingly.
03304         ///  The slave mask and pattern can be set and modified from preop to operational state of the
      network(after the configured addresses have been assigned.) It must be reset if the network is
      restarted.
03305         ///  The slave must support the CoE statusword(0x6041:00) within the PDO cyclic data.
03306         ///  If the Mask is zero the controller will not execute any runtime check of the CoE
      statusword.
03307         ///  If the watchdog must be cleared to OFF when the 'remote drive statusword error' flag is
      raised then bit 26 of the axis ERRORMASK parameter must be set.
03308         ///  See Trio BASIC help for more details.
03309         /// </summary>
03310         /// <param name="slot">Slot number of EtherCAT port.</param>
03311         /// <param name="addr">Configured address of the slave</param>
03312         /// <param name="mask">CoE statusword mask.</param>
03313         /// <param name="trigger_pattern">Defines the state of the relevant bits which will trigger
      the controller AXISSTATUS 'remote drive statusword error' flag (bit 26.)</param>
03314         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03315         void Ethercat_SetStatusMaskAndTriggerPattern(Int32 slot, Int32 addr, UInt32 mask, UInt32
      trigger_pattern);
03316
03317         /// <summary>
03318         /// This command enables the user to display and check the mask and trigger pattern values set
      (if any) for a given slave.
03319         ///  See Trio BASIC help for more details.
03320         /// </summary>
03321         /// <param name="slot">Slot number of EtherCAT port.</param>
03322         /// <param name="addr">Configured address of the slave</param>
03323         /// <param name="out_vr_index">VR index in range [0..max_vr_index-2]. Index of the array of 2
      VRs where the mask and the trigger_pattern will be stored.</param>
03324         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03325         void Ethercat_GetStatusMaskAndTriggerPattern_VR(Int32 slot, Int32 addr, UInt32 out_vr_index);
03326
03327         /// <summary>
03328         /// This command enables the user to display and check the mask and trigger pattern values set
      (if any) for a given slave.
03329         ///  See Trio BASIC help for more details.
03330         /// </summary>
03331         /// <param name="slot">Slot number of EtherCAT port</param>
03332         /// <param name="addr">Configured address of the slave</param>
03333         /// <returns>
03334         /// On success a tuple is returned containing following:<br/>
03335         ///  - Current statusword mask<br/>
03336         ///  - Current trigger pattern
03337         /// </returns>
03338         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03339         Tuple<UInt32, UInt32> Ethercat_GetStatusMaskAndTriggerPattern(Int32 slot, Int32 addr);
03340
03341         /// <summary>
03342         /// Displays the network configuration to the command line terminal in Motion Perfect. See
      Trio BASIC help for more details.
03343         /// </summary>
03344         /// <param name="slot">Slot number of EtherCAT port.</param>
03345         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03346         void Ethercat_DisplayNetworkConfig(Int32 slot);
03347
03348         /// <summary>
03349         /// Set the network default Sync0 start time offset. The offset value is from the given VR.
03350         ///  The default start time is 1 second(1000000 microseconds).Add or subtract the required
      offset in microseconds.For example 1000150 is an offset of 150 microseconds and 999930 is an offset of
      -70 microseconds.The offset must not exceed ±50 % of the servo period.
03351         ///  The function returns TRUE / FALSE depending upon whether the operation was successful.
03352         ///  This command must be run before the EtherCAT network is started.
03353         ///   See Trio BASIC help for more details.
03354         /// </summary>
03355         /// <param name="vr_index">VR index in range [0..max_vr_index-1]. Use the value in the VR
      number specified.</param>
03356         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03357         void Ethercat_SetSync0StartTime_VR(UInt32 vr_index);
03358
03359         /// <summary>
03360         /// Set the network default Sync0 start time offset. If offset value is directly provided as a
      parameter.
03361         ///  The default start time is 1 second(1000000 microseconds).Add or subtract the required
      offset in microseconds.For example 1000150 is an offset of 150 microseconds and 999930 is an offset of
      -70 microseconds.The offset must not exceed ±50 % of the servo period.
```

```
03362         ///  The function returns TRUE / FALSE depending upon whether the operation was successful.
03363         ///  This command must be run before the EtherCAT network is started.
03364         ///   See Trio BASIC help for more details.
03365         /// </summary>
03366         /// <param name="offset">Direct value for the start time</param>
03367         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03368         void Ethercat_SetSync0StartTime(Int32 offset);
03369
03370         /// <summary>
03371         /// Set the EtherCAT master to enable the first DC enabled node to be the Distributed Clocks
      master. As this requires the Trio Motion Coordinator to synchronise to the external DC master, it does
      not function on P876, MC4N-ECAT or MC664(X) as these have FPGA based fixed timing hardware for the
      EtherCAT port.
03372         ///  The function returns TRUE / FALSE depending upon whether the operation was successful.
03373         ///  This command must be run before the EtherCAT network is started.
03374         ///  See Trio BASIC help for more details.
03375         /// </summary>
03376         /// <param name="slot">Slot number of EtherCAT port.</param>
03377         /// <param name="function">Distributed Clocks function. 2 = use first DC enabled node as
      clocks master.</param>
03378         /// <param name="vr_index">VR index in range [0..max_vr_index-1]. Use the value in the VR
      number specified.</param>
03379         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03380         void Ethercat_SetDistrClockMaster_VR(Int32 slot, UInt32 function, UInt32 vr_index);
03381
03382         /// <summary>
03383         /// Set the EtherCAT master to enable the first DC enabled node to be the Distributed Clocks
      master. As this requires the Trio Motion Coordinator to synchronise to the external DC master, it does
      not function on P876, MC4N-ECAT or MC664(X) as these have FPGA based fixed timing hardware for the
      EtherCAT port.
03384         ///  The function returns TRUE / FALSE depending upon whether the operation was successful.
03385         ///  This command must be run before the EtherCAT network is started.
03386         ///  See Trio BASIC help for more details.
03387         /// </summary>
03388         /// <param name="slot">Slot number of EtherCAT port.</param>
03389         /// <param name="function">Distributed Clocks function. 2 = use first DC enabled node as
      clocks master.</param>
03390         /// <param name="value">The value. 0 = use first DC enabled node as the clocks master.</param>
03391         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03392         void Ethercat_SetDistrClockMaster(Int32 slot, UInt32 function, UInt32 value);
03393
03394         /// <summary>
03395         /// Set the response of the master to an emergency telegram transmitted by a slave node. This
      function enables the user to determine whether emergency messages should trigger a system error alarm
      and cause the WDOG to be dropped.
03396         ///  See Trio BASIC help for more details.
03397         /// </summary>
03398         /// <param name="slot">Slot number of EtherCAT port.</param>
03399         /// <param name="mode">
03400         ///  0 : Log error and raise a System Error and drop WDOG off. (Default)<br/>
03401         ///  1 : Log error only.
03402         /// </param>
03403         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03404         void Ethercat_SetEmergencyMsgControl(Int32 slot, UInt32 mode);
03405
03406         /// <summary>
03407         /// Get the current value; indicating the response of the master to an emergency telegram
      transmitted by a slave node.
03408         ///  See Trio BASIC help for more details.
03409         /// </summary>
03410         /// <param name="slot">Slot number of EtherCAT port.</param>
03411         /// <param name="out_vr_index">Result is placed in the selected VR variable</param>
03412         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03413         void Ethercat_GetEmergencyMsgControl_VR(Int32 slot, UInt32 out_vr_index);
03414
03415         /// <summary>
03416         /// Get the current value; indicating the response of the master to an emergency telegram
      transmitted by a slave node.
03417         ///  See Trio BASIC help for more details.
03418         /// </summary>
03419         /// <param name="slot">Slot number of EtherCAT port.</param>
03420         /// <returns>On success the current value</returns>
03421         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03422         UInt32 Ethercat_GetEmergencyMsgControl(Int32 slot);
03423
03424         /// <summary>
03425         /// This function gets a CANopen-over-EtherCAT (CoE) object from the remote drive or I/O
      device. The object's index and sub-index are used to request a value and that value is either placed
      in the VR or is displayed in the Motion Perfect terminal if the VR number is set to -1.
03426         ///  See Trio BASIC help for more details.
03427         /// </summary>
```

```
03428         /// <param name="slot">Slot number of EtherCAT port.</param>
03429         /// <param name="address">Node address of the remote device on the network</param>
03430         /// <param name="object_index">CANopen Object index</param>
03431         /// <param name="object_subindex">CANopen Object sub-index</param>
03432         /// <param name="type">
03433         ///  1 : Boolean<br/>
03434         ///  2 : Integer 8<br/>
03435         ///  3 : Integer 16<br/>
03436         ///  4 : Integer 32<br/>
03437         ///  5 : Unsigned 8<br/>
03438         ///  6 : Unsigned 16<br/>
03439         ///  7 : Unsigned 32<br/>
03440         ///  8 : Real 32<br/>
03441         ///  17 : Real 64<br/>
03442         ///  21 : Integer 64<br/>
03443         ///  27 : Unsigned 64
03444         /// </param>
03445         /// <param name="out_vr_index">Result is placed in the selected VR variable</param>
03446         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03447         void Ethercat_CoRead(Int32 slot, Int32 address, UInt16 object_index, Byte object_subindex,
      Co_ObjectType type, Int32 out_vr_index);
03448
03449         /// <summary>
03450         /// This function sets a CANopen-over-EtherCAT (CoE) object in the remote drive or I/O device.
      The object's index and sub-index are used to write a value to that object. The value comes from a VR.
03451         ///  Use this function for integer values.
03452         ///  Refer to the remote device's manual for a list of available objects.
03453         ///  See Trio BASIC help for more details.
03454         /// </summary>
03455         /// <param name="slot">Slot number of EtherCAT port.</param>
03456         /// <param name="address">Node address of the remote device on the network</param>
03457         /// <param name="object_index">CANopen Object index</param>
03458         /// <param name="object_subindex">CANopen Object sub-index</param>
03459         /// <param name="type">
03460         ///  1 : Boolean<br/>
03461         ///  2 : Integer 8<br/>
03462         ///  3 : Integer 16<br/>
03463         ///  4 : Integer 32<br/>
03464         ///  5 : Unsigned 8<br/>
03465         ///  6 : Unsigned 16<br/>
03466         ///  7 : Unsigned 32<br/>
03467         ///  8 : Real 32<br/>
03468         ///  17 : Real 64<br/>
03469         ///  21 : Integer 64<br/>
03470         ///  27 : Unsigned 64
03471         /// </param>
03472         /// <param name="vr_index">VR number between 0 and max VR where the result will be stored. (-1
      means the value will be printed to the terminal)</param>
03473         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03474         void Ethercat_CoWrite_VR(Int32 slot, Int32 address, UInt16 object_index, Byte object_subindex,
      Co_ObjectType type, UInt32 vr_index);
03475
03476         /// <summary>
03477         /// This function sets a CANopen-over-EtherCAT (CoE) object in the remote drive or I/O device.
      The object's index and sub-index are used to write a value to that object.
03478         ///  Use this function for integer values.
03479         ///  Refer to the remote device's manual for a list of available objects.
03480         ///  See Trio BASIC help for more details.
03481         /// </summary>
03482         /// <param name="slot">Slot number of EtherCAT port.</param>
03483         /// <param name="address">Node address of the remote device on the network</param>
03484         /// <param name="object_index">CANopen Object index</param>
03485         /// <param name="object_subindex">CANopen Object sub-index</param>
03486         /// <param name="type">
03487         ///  1 : Boolean<br/>
03488         ///  2 : Integer 8<br/>
03489         ///  3 : Integer 16<br/>
03490         ///  4 : Integer 32<br/>
03491         ///  5 : Unsigned 8<br/>
03492         ///  6 : Unsigned 16<br/>
03493         ///  7 : Unsigned 32<br/>
03494         ///  8 : Real 32<br/>
03495         ///  17 : Real 64<br/>
03496         ///  21 : Integer 64<br/>
03497         ///  27 : Unsigned 64
03498         /// </param>
03499         /// <param name="value">The value</param>
03500         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03501         void Ethercat_CoWrite_Value(Int32 slot, Int32 address, UInt16 object_index, Byte
      object_subindex, Co_ObjectType type, Int64 value);
03502
03503         /// <summary>
03504         /// This function sets a CANopen-over-EtherCAT (CoE) object in the remote drive or I/O device.
      The object's index and sub-index are used to write a value to that object.
```

```
03505         ///  Use this function for floating-point values.
03506         ///  Refer to the remote device's manual for a list of available objects.
03507         ///  See Trio BASIC help for more details.
03508         /// </summary>
03509         /// <param name="slot">Slot number of EtherCAT port.</param>
03510         /// <param name="address">Node address of the remote device on the network</param>
03511         /// <param name="object_index">CANopen Object index</param>
03512         /// <param name="object_subindex">CANopen Object sub-index</param>
03513         /// <param name="type">
03514         ///  1 : Boolean<br/>
03515         ///  2 : Integer 8<br/>
03516         ///  3 : Integer 16<br/>
03517         ///  4 : Integer 32<br/>
03518         ///  5 : Unsigned 8<br/>
03519         ///  6 : Unsigned 16<br/>
03520         ///  7 : Unsigned 32<br/>
03521         ///  8 : Real 32<br/>
03522         ///  17 : Real 64<br/>
03523         ///  21 : Integer 64<br/>
03524         ///  27 : Unsigned 64
03525         /// </param>
03526         /// <param name="value">The value</param>
03527         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03528         void Ethercat_CoWrite_Value(Int32 slot, Int32 address, UInt16 object_index, Byte
      object_subindex, Co_ObjectType type, double value);
03529
03530         /// <summary>
03531         /// This function gets a CANopen-over-EtherCAT (CoE) object from the remote drive or I/O
      device. The object's index and sub-index are used to request a value and that value is either placed
      in the VR or is displayed in the Motion Perfect terminal if the VR number is set to -1.
03532         ///  Refer to the remote device's manual for a list of available objects.
03533         ///  See Trio BASIC help for more details.
03534         /// </summary>
03535         /// <param name="axis_number">Axis number of the EtherCAT drive.</param>
03536         /// <param name="object_index">CANopen Object index</param>
03537         /// <param name="object_subindex">CANopen Object sub-index</param>
03538         /// <param name="type">
03539         ///  1 : Boolean<br/>
03540         ///  2 : Integer 8<br/>
03541         ///  3 : Integer 16<br/>
03542         ///  4 : Integer 32<br/>
03543         ///  5 : Unsigned 8<br/>
03544         ///  6 : Unsigned 16<br/>
03545         ///  7 : Unsigned 32<br/>
03546         ///  8 : Real 32<br/>
03547         ///  17 : Real 64<br/>
03548         ///  21 : Integer 64<br/>
03549         ///  27 : Unsigned 64
03550         /// </param>
03551         /// <param name="out_vr_index">Result is placed in the selected VR variable</param>
03552         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03553         void Ethercat_CoReadAxis(Int32 axis_number, UInt16 object_index, Byte object_subindex,
      Co_ObjectType type, Int32 out_vr_index);
03554
03555         /// <summary>
03556         /// This function sets a CANopen-over-EtherCAT (CoE) object in the remote drive or I/O device.
      The object's index and sub-index are used to write a value to that object. The value comes from a VR.
03557         ///  Refer to the remote device's manual for a list of available objects.
03558         ///  See Trio BASIC help for more details.
03559         /// </summary>
03560         /// <param name="axis_number">Axis number of the EtherCAT drive.</param>
03561         /// <param name="object_index">CANopen Object index</param>
03562         /// <param name="object_subindex">CANopen Object sub-index</param>
03563         /// <param name="type">
03564         ///  1 : Boolean<br/>
03565         ///  2 : Integer 8<br/>
03566         ///  3 : Integer 16<br/>
03567         ///  4 : Integer 32<br/>
03568         ///  5 : Unsigned 8<br/>
03569         ///  6 : Unsigned 16<br/>
03570         ///  7 : Unsigned 32<br/>
03571         ///  8 : Real 32<br/>
03572         ///  17 : Real 64<br/>
03573         ///  21 : Integer 64<br/>
03574         ///  27 : Unsigned 64
03575         /// </param>
03576         /// <param name="vr_index">VR number between 0 and max VR where the value is stored.</param>
03577         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03578         void Ethercat_CoWriteAxis_VR(Int32 axis_number, UInt16 object_index, Byte object_subindex,
      Co_ObjectType type, UInt32 vr_index);
03579
03580         /// <summary>
03581         /// This function sets a CANopen-over-EtherCAT (CoE) object in the remote drive or I/O device.
      The object's index and sub-index are used to write a value to that object.
```

```
03582         ///  Use this function for integer values.
03583         ///  Refer to the remote device's manual for a list of available objects.
03584         ///  See Trio BASIC help for more details.
03585         /// </summary>
03586         /// <param name="axis_number">Axis number of the EtherCAT drive.</param>
03587         /// <param name="object_index">CANopen Object index</param>
03588         /// <param name="object_subindex">CANopen Object sub-index</param>
03589         /// <param name="type">
03590         ///  1 : Boolean<br/>
03591         ///  2 : Integer 8<br/>
03592         ///  3 : Integer 16<br/>
03593         ///  4 : Integer 32<br/>
03594         ///  5 : Unsigned 8<br/>
03595         ///  6 : Unsigned 16<br/>
03596         ///  7 : Unsigned 32<br/>
03597         ///  8 : Real 32<br/>
03598         ///  17 : Real 64<br/>
03599         ///  21 : Integer 64<br/>
03600         ///  27 : Unsigned 64
03601         /// </param>
03602         /// <param name="value">The value.</param>
03603         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03604         void Ethercat_CoWriteAxis_Value(Int32 axis_number, UInt16 object_index, Byte object_subindex,
      Co_ObjectType type, Int64 value);
03605
03606         /// <summary>
03607         /// This function sets a CANopen-over-EtherCAT (CoE) object in the remote drive or I/O device.
      The object's index and sub-index are used to write a value to that object.
03608         ///  Use this function for floating-point values.
03609         ///  Refer to the remote device's manual for a list of available objects.
03610         ///  See Trio BASIC help for more details.
03611         /// </summary>
03612         /// <param name="axis_number">Axis number of the EtherCAT drive.</param>
03613         /// <param name="object_index">CANopen Object index</param>
03614         /// <param name="object_subindex">CANopen Object sub-index</param>
03615         /// <param name="type">
03616         ///  1 : Boolean<br/>
03617         ///  2 : Integer 8<br/>
03618         ///  3 : Integer 16<br/>
03619         ///  4 : Integer 32<br/>
03620         ///  5 : Unsigned 8<br/>
03621         ///  6 : Unsigned 16<br/>
03622         ///  7 : Unsigned 32<br/>
03623         ///  8 : Real 32<br/>
03624         ///  17 : Real 64<br/>
03625         ///  21 : Integer 64<br/>
03626         ///  27 : Unsigned 64
03627         /// </param>
03628         /// <param name="value">The value.</param>
03629         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03630         void Ethercat_CoWriteAxis_Value(Int32 axis_number, UInt16 object_index, Byte object_subindex,
      Co_ObjectType type, double value);
03631
03632         /// <summary>
03633         /// Uploads a file to the Motion Coordinator.
03634         /// </summary>
03635         /// <param name="local_file">Local file name</param>
03636         /// <param name="remote_file">
03637         /// Remote file name. Can be prefixed by:<br/>
03638         ///  "RAM:" - the location of the file is in the controller RAM.<br/>
03639         ///  "SD:" - the location of the file is on the SD card.<br/>
03640         ///  "FIFO:" - The target file is a FIFO file on the controller. It is expected that the
      controller (a program) will consume the data.<br/>
03641         ///  If no prefix "RAM:" is assumed.
03642         /// </param>
03643         /// <param name="remote_file_type">The type of the file. Ignored when uploading to the SD card
      or with for FIFO transfers.</param>
03644         /// <param name="opt">File transfer option</param>
03645         /// <param name="handler">Callback handler with the progress advance</param>
03646         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03647         void UploadFile(String local_file, String remote_file, FileType remote_file_type,
      FileTransferOption opt, ProgressHandlerCallback handler);
03648
03649         /// <summary>
03650         /// Downloads a file from the Motion Coordinator.
03651         /// </summary>
03652         /// <param name="local_file">Local source file</param>
03653         /// <param name="remote_file">
03654         /// Remote file name. Can be prefixed by:<br/>
03655         ///  "RAM:" - the location of the file is in the controller RAM.<br/>
03656         ///  "SD:" - the location of the file is on the SD card.<br/>
03657         ///  "FIFO:" - The target file is a FIFO file on the controller.<br/>
03658         ///  If no prefix "RAM:" is assumed.
03659         /// </param>
```

```
03660         /// <param name="handler">Callback handler with the progress advance</param>
03661         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03662         void DownloadFile(String local_file, String remote_file, ProgressHandlerCallback handler);
03663
03664         /// <summary>
03665         /// Deletes a file on the Motion Coordinator.
03666         /// </summary>
03667         /// <param name="remote_file">The name of the file</param>
03668         /// <param name="compiled_code_only">In case of a program: when true only the compiled code of
      the program will be deleted, the source code will remain. When false all data related to the program
      will be deleted</param>
03669         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03670         void Delete(String remote_file, bool compiled_code_only = false);
03671
03672         /// <summary>
03673         /// Deletes a file, all files or the TABLE data the Motion Coordinator.
03674         /// </summary>
03675         /// <param name="target">
03676         /// The target to be cleared (deleted)<br/>
03677         ///  "file_name" - deletes the file with given name from the Motion Coordinator<br/>
03678         ///  "ALL" - deletes all user files from the Motion Coordinator. System files are not
      deleted.<br/>
03679         ///  "TABLE" - sets all TABLE values to 0 .
03680         /// </param>
03681         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03682         void New(String target);
03683
03684         /// <summary>
03685         /// Makes the named program the currently selected program on the Motion Coordinator, if the
      named program does not exist then it makes a program of that name.
03686         ///  The selected program cannot be changed when programs are running. When a program is
      selected any previously selected program is compiled.
03687         ///  When program_type is specified is to make a program of that name and type, it cannot be
      possible to change the type of an existing program.
03688         /// </summary>
03689         /// <param name="remote_file">The file (program) name</param>
03690         /// <param name="file_type">The type of the file (program) to be created</param>
03691         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03692         void Select(String remote_file, FileType file_type);
03693
03694         /// <summary>
03695         /// This axis function disables backlash compensation.
03696         /// </summary>
03697         /// <param name="axis">Base axis for this command. If value is -1 then the default base axis
      is used</param>
03698         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03699         void BacklashDisable(Int32 axis = -1);
03700
03701         /// <summary>
03702         /// This axis function allows backlash compensation to be loaded. This is achieved by applying
      an offset move when the motor demand is in one direction, then reversing the offset move when the
      motor demand is in the opposite direction.
03703         ///  These moves are superimposed on the commanded axis movements.
03704         ///  The backlash compensation is applied after a reversal of the direction of change of the
      DPOS parameter.
03705         ///  The backlash compensation can be seen in the AXIS_DPOS axis parameter.This is effectively
      DPOS + backlash compensation.
03706         /// </summary>
03707         /// <param name="distance">The distance to be offset in user units</param>
03708         /// <param name="speed">The ACCEL / DECEL rate at which is compensation move is applied in
      user units</param>
03709         /// <param name="acceleration">Enables/disables the backlash compensation</param>
03710         /// <param name="axis">Base axis for this command. If value is -1 then the default base axis
      is used</param>
03711         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03712         void Backlash(double distance, double speed, double acceleration, Int32 axis = -1);
03713
03714         /// <summary>
03715         /// This command allows the incoming encoder count to be scaled by a non integer ratio :
03716         ///  MPOS = (mpos_count / input_count) x encoder_edges_input
03717         ///  When using the servo loop you will need to adjust the gains to maintain performance and
      stability.
03718         ///  Unlike the UNITS parameter, which only affects the scaling seen by the user programs,
      ENCODER_RATIO affects all motion commands.
03719         ///  ENCODER_RATIO does not replace UNITS.Only use ENCODER_RATIO where absolutely
      necessary.PP_STEP and ENCODER_RATIO cannot be used at the same time on the same axis.
03720         /// </summary>
03721         /// <param name="mpos_count">An integer number which defines the numerator</param>
03722         /// <param name="input_count">An integer number which defines the denominator</param>
03723         /// <param name="axis">Base axis for this command. If value is -1 then the default base axis
      is used</param>
```

```
03724         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03725         void EncoderRatio(Int64 mpos_count, Int64 input_count, Int32 axis = -1);
03726
03727         /// <summary>
03728         /// This command sets up an integer ratio for the axis' demanded position.  Usually this is
      used on a stepper (pulse+direction) output.  It can also be applied to a networked axis in positon
      mode, such as EtherCAT and RTEX position servo.
03729         ///  Every servo-period the change in position is passed through the STEP_RATIO function
      before it goes to the step pulse or network drive output.
03730         ///  The STEP_RATIO function operates before the divide by 16 factor(divide by 256 in
      Flexslice P375) in the stepper axis.This maintains the good timing resolution of the stepper output
      circuit.
03731         ///  STEP_RATIO does not replace UNITS.
03732         /// </summary>
03733         /// <param name="output_count">Number of counts to output for the given dpos_count value.
      Range: 0 to 16777215</param>
03734         /// <param name="dpos_count">Change in DPOS value for corresponding output count.  Range: 0 to
      16777215.</param>
03735         /// <param name="axis">Base axis for this command. If value is -1 then the default base axis
      is used</param>
03736         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03737         void StepRatio(Int64 output_count, Int64 dpos_count, Int32 axis = -1);
03738
03739         /// <summary>
03740         /// This command is used to generate movement of an axis according to a table of positions
      which define a movement profile.
03741         /// </summary>
03742         /// <param name="tableStart">The start position of the cam profile in the TABLE.</param>
03743         /// <param name="tableStop">The end position of the cam profile in the TABLE.</param>
03744         /// <param name="tableMultiplier">The table values are multiplied by this value to generate
      the positions.</param>
03745         /// <param name="linkDistance">The distance parameter relates the speed of the axis to the
      time taken to complete the cam profile.</param>
03746         /// <param name="axis">Base axis for this command. If value is -1 then the default base axis
      is used</param>
03747         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03748         void Cam(int tableStart, int tableStop, double tableMultiplier, double linkDistance, Int32
      axis = -1);
03749
03750         /// <summary>
03751         /// This command is used to generate movement of an axis according to a table of POSITIONS
      which define the movement profile
03752         /// </summary>
03753         /// <param name="tableStart">The start position of the cam profile in the TABLE.</param>
03754         /// <param name="tableStop">The end position of the cam profile in the TABLE.</param>
03755         /// <param name="tableMultiplier">The table values are multiplied by this value to generate
      the positions.</param>
03756         /// <param name="linkDistance">The distance the link axis must move to complete CAMBOX
      profile.</param>
03757         /// <param name="linkAxis">The axis to link to.</param>
03758         /// <param name="linkOptions">Bit value options to customise how your CAMBOX operates.</param>
03759         /// <param name="linkPos">
03760         ///  linkOptions bit 1 - the absolute position on the link axis in user UNITS where the CAMBOX
      is to be start.<br/>
03761         ///  linkOptions bit 9 - the registration channel to start the movement on
03762         /// </param>
03763         /// <param name="offsetStart">The position defined on the link_dist where the profile will
      start. </param>
03764         /// <param name="axis">Base axis for this command. If value is -1 then the default base axis
      is used</param>
03765         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03766         void CamBox(int tableStart, int tableStop, double tableMultiplier, double linkDistance, int
      linkAxis, CamBoxLinkOptions linkOptions, double linkPos = 0, double offsetStart = 0, Int32 axis = -1);
03767
03768
      ////////////////////////////////////////////////////////////////////////////////////////////////////////////////
03769
03770
      ////////////////////////////////////////////////////////////////////////////////////////////////////////////////
03771
03772         /// <summary>
03773         /// Getter for AIN0 system parameter
03774         /// </summary>
03775         /// <returns>Current value of the parameter</returns>
03776         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03777         uint GetSystemParameter_AIN0();
03778
03779         /// <summary>
03780         /// Setter for AIN0 system parameter
03781         /// </summary>
03782         /// <param name="value">New value to set to parameter</param>
03783         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
```

```
module.</exception>
03784         void SetSystemParameter_AIN0(uint value);
03785
03786
03787         /// <summary>
03788         /// Getter for AIN1 system parameter
03789         /// </summary>
03790         /// <returns>Current value of the parameter</returns>
03791         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
03792         uint GetSystemParameter_AIN1();
03793
03794         /// <summary>
03795         /// Setter for AIN1 system parameter
03796         /// </summary>
03797         /// <param name="value">New value to set to parameter</param>
03798         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
03799         void SetSystemParameter_AIN1(uint value);
03800
03801
03802         /// <summary>
03803         /// Getter for AIN2 system parameter
03804         /// </summary>
03805         /// <returns>Current value of the parameter</returns>
03806         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
03807         uint GetSystemParameter_AIN2();
03808
03809         /// <summary>
03810         /// Setter for AIN2 system parameter
03811         /// </summary>
03812         /// <param name="value">New value to set to parameter</param>
03813         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
03814         void SetSystemParameter_AIN2(uint value);
03815
03816
03817         /// <summary>
03818         /// Getter for AIN3 system parameter
03819         /// </summary>
03820         /// <returns>Current value of the parameter</returns>
03821         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
03822         uint GetSystemParameter_AIN3();
03823
03824         /// <summary>
03825         /// Setter for AIN3 system parameter
03826         /// </summary>
03827         /// <param name="value">New value to set to parameter</param>
03828         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
03829         void SetSystemParameter_AIN3(uint value);
03830
03831
03832         /// <summary>
03833         /// Getter for AOUT0 system parameter
03834         /// </summary>
03835         /// <returns>Current value of the parameter</returns>
03836         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
03837         uint GetSystemParameter_AOUT0();
03838
03839         /// <summary>
03840         /// Setter for AOUT0 system parameter
03841         /// </summary>
03842         /// <param name="value">New value to set to parameter</param>
03843         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
03844         void SetSystemParameter_AOUT0(uint value);
03845
03846
03847         /// <summary>
03848         /// Getter for AOUT1 system parameter
03849         /// </summary>
03850         /// <returns>Current value of the parameter</returns>
03851         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
03852         uint GetSystemParameter_AOUT1();
03853
03854         /// <summary>
03855         /// Setter for AOUT1 system parameter
03856         /// </summary>
03857         /// <param name="value">New value to set to parameter</param>
03858         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
03859         void SetSystemParameter_AOUT1(uint value);
```

```
03860
03861
03862         /// <summary>
03863         /// Getter for AOUT2 system parameter
03864         /// </summary>
03865         /// <returns>Current value of the parameter</returns>
03866         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03867         uint GetSystemParameter_AOUT2();
03868
03869         /// <summary>
03870         /// Setter for AOUT2 system parameter
03871         /// </summary>
03872         /// <param name="value">New value to set to parameter</param>
03873         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03874         void SetSystemParameter_AOUT2(uint value);
03875
03876
03877         /// <summary>
03878         /// Getter for AOUT3 system parameter
03879         /// </summary>
03880         /// <returns>Current value of the parameter</returns>
03881         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03882         uint GetSystemParameter_AOUT3();
03883
03884         /// <summary>
03885         /// Setter for AOUT3 system parameter
03886         /// </summary>
03887         /// <param name="value">New value to set to parameter</param>
03888         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03889         void SetSystemParameter_AOUT3(uint value);
03890
03891
03892         /// <summary>
03893         /// Getter for AINBI0 system parameter
03894         /// </summary>
03895         /// <returns>Current value of the parameter</returns>
03896         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03897         uint GetSystemParameter_AINBI0();
03898
03899         /// <summary>
03900         /// Setter for AINBI0 system parameter
03901         /// </summary>
03902         /// <param name="value">New value to set to parameter</param>
03903         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03904         void SetSystemParameter_AINBI0(uint value);
03905
03906
03907         /// <summary>
03908         /// Getter for AINBI1 system parameter
03909         /// </summary>
03910         /// <returns>Current value of the parameter</returns>
03911         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03912         uint GetSystemParameter_AINBI1();
03913
03914         /// <summary>
03915         /// Setter for AINBI1 system parameter
03916         /// </summary>
03917         /// <param name="value">New value to set to parameter</param>
03918         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03919         void SetSystemParameter_AINBI1(uint value);
03920
03921
03922         /// <summary>
03923         /// Getter for AINBI2 system parameter
03924         /// </summary>
03925         /// <returns>Current value of the parameter</returns>
03926         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03927         uint GetSystemParameter_AINBI2();
03928
03929         /// <summary>
03930         /// Setter for AINBI2 system parameter
03931         /// </summary>
03932         /// <param name="value">New value to set to parameter</param>
03933         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03934         void SetSystemParameter_AINBI2(uint value);
03935
03936
```

```
03937         /// <summary>
03938         /// Getter for AINBI3 system parameter
03939         /// </summary>
03940         /// <returns>Current value of the parameter</returns>
03941         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03942         uint GetSystemParameter_AINBI3();
03943
03944         /// <summary>
03945         /// Setter for AINBI3 system parameter
03946         /// </summary>
03947         /// <param name="value">New value to set to parameter</param>
03948         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03949         void SetSystemParameter_AINBI3(uint value);
03950
03951
03952         /// <summary>
03953         /// Getter for INPUTS0 system parameter
03954         /// </summary>
03955         /// <returns>Current value of the parameter</returns>
03956         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03957         uint GetSystemParameter_INPUTS0();
03958
03959
03960         /// <summary>
03961         /// Getter for INPUTS1 system parameter
03962         /// </summary>
03963         /// <returns>Current value of the parameter</returns>
03964         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03965         uint GetSystemParameter_INPUTS1();
03966
03967
03968         /// <summary>
03969         /// Getter for SLOT0_TIME system parameter
03970         /// </summary>
03971         /// <returns>Current value of the parameter</returns>
03972         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03973         uint GetSystemParameter_SLOT0_TIME();
03974
03975         /// <summary>
03976         /// Setter for SLOT0_TIME system parameter
03977         /// </summary>
03978         /// <param name="value">New value to set to parameter</param>
03979         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03980         void SetSystemParameter_SLOT0_TIME(uint value);
03981
03982
03983         /// <summary>
03984         /// Getter for SLOT1_TIME system parameter
03985         /// </summary>
03986         /// <returns>Current value of the parameter</returns>
03987         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03988         uint GetSystemParameter_SLOT1_TIME();
03989
03990         /// <summary>
03991         /// Setter for SLOT1_TIME system parameter
03992         /// </summary>
03993         /// <param name="value">New value to set to parameter</param>
03994         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
03995         void SetSystemParameter_SLOT1_TIME(uint value);
03996
03997
03998         /// <summary>
03999         /// Getter for SLOT2_TIME system parameter
04000         /// </summary>
04001         /// <returns>Current value of the parameter</returns>
04002         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04003         uint GetSystemParameter_SLOT2_TIME();
04004
04005         /// <summary>
04006         /// Setter for SLOT2_TIME system parameter
04007         /// </summary>
04008         /// <param name="value">New value to set to parameter</param>
04009         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04010         void SetSystemParameter_SLOT2_TIME(uint value);
04011
04012
04013         /// <summary>
```

```
04014         /// Getter for SLOT3_TIME system parameter
04015         /// </summary>
04016         /// <returns>Current value of the parameter</returns>
04017         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04018         uint GetSystemParameter_SLOT3_TIME();
04019
04020         /// <summary>
04021         /// Setter for SLOT3_TIME system parameter
04022         /// </summary>
04023         /// <param name="value">New value to set to parameter</param>
04024         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04025         void SetSystemParameter_SLOT3_TIME(uint value);
04026
04027
04028         /// <summary>
04029         /// Getter for WDOG system parameter
04030         /// </summary>
04031         /// <returns>Current value of the parameter</returns>
04032         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04033         bool GetSystemParameter_WDOG();
04034
04035         /// <summary>
04036         /// Setter for WDOG system parameter
04037         /// </summary>
04038         /// <param name="value">New value to set to parameter</param>
04039         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04040         void SetSystemParameter_WDOG(bool value);
04041
04042
04043         /// <summary>
04044         /// Getter for SYSTEM_ERROR system parameter
04045         /// </summary>
04046         /// <returns>Current value of the parameter</returns>
04047         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04048         uint GetSystemParameter_SYSTEM_ERROR();
04049
04050
04051         /// <summary>
04052         /// Getter for CURRENT_JITTER system parameter
04053         /// </summary>
04054         /// <returns>Current value of the parameter</returns>
04055         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04056         long GetSystemParameter_CURRENT_JITTER();
04057
04058         /// <summary>
04059         /// Setter for CURRENT_JITTER system parameter
04060         /// </summary>
04061         /// <param name="value">New value to set to parameter</param>
04062         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04063         void SetSystemParameter_CURRENT_JITTER(long value);
04064
04065
04066         /// <summary>
04067         /// Getter for AVERAGE_JITTER system parameter
04068         /// </summary>
04069         /// <returns>Current value of the parameter</returns>
04070         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04071         long GetSystemParameter_AVERAGE_JITTER();
04072
04073         /// <summary>
04074         /// Setter for AVERAGE_JITTER system parameter
04075         /// </summary>
04076         /// <param name="value">New value to set to parameter</param>
04077         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04078         void SetSystemParameter_AVERAGE_JITTER(long value);
04079
04080
04081         /// <summary>
04082         /// Getter for MAX_JITTER system parameter
04083         /// </summary>
04084         /// <returns>Current value of the parameter</returns>
04085         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04086         long GetSystemParameter_MAX_JITTER();
04087
04088         /// <summary>
04089         /// Setter for MAX_JITTER system parameter
04090         /// </summary>
```

```
04091         /// <param name="value">New value to set to parameter</param>
04092         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04093         void SetSystemParameter_MAX_JITTER(long value);
04094
04095
04096         /// <summary>
04097         /// Getter for MISSED_INTERRUPTS system parameter
04098         /// </summary>
04099         /// <returns>Current value of the parameter</returns>
04100         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04101         long GetSystemParameter_MISSED_INTERRUPTS();
04102
04103         /// <summary>
04104         /// Setter for MISSED_INTERRUPTS system parameter
04105         /// </summary>
04106         /// <param name="value">New value to set to parameter</param>
04107         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04108         void SetSystemParameter_MISSED_INTERRUPTS(long value);
04109
04110
04111         /// <summary>
04112         /// Getter for MISSED_SEND_FRAMES system parameter
04113         /// </summary>
04114         /// <returns>Current value of the parameter</returns>
04115         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04116         long GetSystemParameter_MISSED_SEND_FRAMES();
04117
04118         /// <summary>
04119         /// Setter for MISSED_SEND_FRAMES system parameter
04120         /// </summary>
04121         /// <param name="value">New value to set to parameter</param>
04122         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04123         void SetSystemParameter_MISSED_SEND_FRAMES(long value);
04124
04125
04126         /// <summary>
04127         /// Getter for MISSED_RECV_FRAMES system parameter
04128         /// </summary>
04129         /// <returns>Current value of the parameter</returns>
04130         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04131         long GetSystemParameter_MISSED_RECV_FRAMES();
04132
04133         /// <summary>
04134         /// Setter for MISSED_RECV_FRAMES system parameter
04135         /// </summary>
04136         /// <param name="value">New value to set to parameter</param>
04137         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04138         void SetSystemParameter_MISSED_RECV_FRAMES(long value);
04139
04140
04141         /// <summary>
04142         /// Getter for LIMIT_BUFFERED system parameter
04143         /// </summary>
04144         /// <returns>Current value of the parameter</returns>
04145         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04146         short GetSystemParameter_LIMIT_BUFFERED();
04147
04148         /// <summary>
04149         /// Setter for LIMIT_BUFFERED system parameter
04150         /// </summary>
04151         /// <param name="value">New value to set to parameter</param>
04152         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04153         void SetSystemParameter_LIMIT_BUFFERED(short value);
04154
04155
04156         /// <summary>
04157         /// Getter for ADDRESS system parameter
04158         /// </summary>
04159         /// <returns>Current value of the parameter</returns>
04160         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04161         uint GetSystemParameter_ADDRESS();
04162
04163         /// <summary>
04164         /// Setter for ADDRESS system parameter
04165         /// </summary>
04166         /// <param name="value">New value to set to parameter</param>
04167         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
```

```
      module.</exception>
04168         void SetSystemParameter_ADDRESS(uint value);
04169
04170
04171         /// <summary>
04172         /// Getter for AUTO_ETHERCAT system parameter
04173         /// </summary>
04174         /// <returns>Current value of the parameter</returns>
04175         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04176         int GetSystemParameter_AUTO_ETHERCAT();
04177
04178         /// <summary>
04179         /// Setter for AUTO_ETHERCAT system parameter
04180         /// </summary>
04181         /// <param name="value">New value to set to parameter</param>
04182         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04183         void SetSystemParameter_AUTO_ETHERCAT(int value);
04184
04185
04186         /// <summary>
04187         /// Getter for CONTROL system parameter
04188         /// </summary>
04189         /// <returns>Current value of the parameter</returns>
04190         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04191         uint GetSystemParameter_CONTROL();
04192
04193
04194         /// <summary>
04195         /// Getter for CPU_EXCEPTIONS system parameter
04196         /// </summary>
04197         /// <returns>Current value of the parameter</returns>
04198         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04199         long GetSystemParameter_CPU_EXCEPTIONS();
04200
04201         /// <summary>
04202         /// Setter for CPU_EXCEPTIONS system parameter
04203         /// </summary>
04204         /// <param name="value">New value to set to parameter</param>
04205         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04206         void SetSystemParameter_CPU_EXCEPTIONS(long value);
04207
04208
04209         /// <summary>
04210         /// Getter for DISPLAY system parameter
04211         /// </summary>
04212         /// <returns>Current value of the parameter</returns>
04213         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04214         int GetSystemParameter_DISPLAY();
04215
04216         /// <summary>
04217         /// Setter for DISPLAY system parameter
04218         /// </summary>
04219         /// <param name="value">New value to set to parameter</param>
04220         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04221         void SetSystemParameter_DISPLAY(int value);
04222
04223
04224         /// <summary>
04225         /// Getter for FLASH_STATUS system parameter
04226         /// </summary>
04227         /// <returns>Current value of the parameter</returns>
04228         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04229         uint GetSystemParameter_FLASH_STATUS();
04230
04231
04232         /// <summary>
04233         /// Getter for ERROR_AXIS system parameter
04234         /// </summary>
04235         /// <returns>Current value of the parameter</returns>
04236         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04237         int GetSystemParameter_ERROR_AXIS();
04238
04239
04240         /// <summary>
04241         /// Getter for FPU_EXCEPTIONS system parameter
04242         /// </summary>
04243         /// <returns>Current value of the parameter</returns>
04244         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
```

```
      module.</exception>
04245         long GetSystemParameter_FPU_EXCEPTIONS();
04246
04247         /// <summary>
04248         /// Setter for FPU_EXCEPTIONS system parameter
04249         /// </summary>
04250         /// <param name="value">New value to set to parameter</param>
04251         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04252         void SetSystemParameter_FPU_EXCEPTIONS(long value);
04253
04254
04255         /// <summary>
04256         /// Getter for HMI_PROC system parameter
04257         /// </summary>
04258         /// <returns>Current value of the parameter</returns>
04259         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04260         int GetSystemParameter_HMI_PROC();
04261
04262         /// <summary>
04263         /// Setter for HMI_PROC system parameter
04264         /// </summary>
04265         /// <param name="value">New value to set to parameter</param>
04266         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04267         void SetSystemParameter_HMI_PROC(int value);
04268
04269
04270         /// <summary>
04271         /// Getter for IP_ADDRESS system parameter
04272         /// </summary>
04273         /// <returns>Current value of the parameter</returns>
04274         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04275         uint GetSystemParameter_IP_ADDRESS();
04276
04277         /// <summary>
04278         /// Setter for IP_ADDRESS system parameter
04279         /// </summary>
04280         /// <param name="value">New value to set to parameter</param>
04281         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04282         void SetSystemParameter_IP_ADDRESS(uint value);
04283
04284
04285         /// <summary>
04286         /// Getter for IP_GATEWAY system parameter
04287         /// </summary>
04288         /// <returns>Current value of the parameter</returns>
04289         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04290         uint GetSystemParameter_IP_GATEWAY();
04291
04292         /// <summary>
04293         /// Setter for IP_GATEWAY system parameter
04294         /// </summary>
04295         /// <param name="value">New value to set to parameter</param>
04296         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04297         void SetSystemParameter_IP_GATEWAY(uint value);
04298
04299
04300         /// <summary>
04301         /// Getter for IP_MAC system parameter
04302         /// </summary>
04303         /// <returns>Current value of the parameter</returns>
04304         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04305         ulong GetSystemParameter_IP_MAC();
04306
04307         /// <summary>
04308         /// Setter for IP_MAC system parameter
04309         /// </summary>
04310         /// <param name="value">New value to set to parameter</param>
04311         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04312         void SetSystemParameter_IP_MAC(ulong value);
04313
04314
04315         /// <summary>
04316         /// Getter for IP_MEMORY_CONFIG system parameter
04317         /// </summary>
04318         /// <returns>Current value of the parameter</returns>
04319         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04320         uint GetSystemParameter_IP_MEMORY_CONFIG();
```

```
04321
04322         /// <summary>
04323         /// Setter for IP_MEMORY_CONFIG system parameter
04324         /// </summary>
04325         /// <param name="value">New value to set to parameter</param>
04326         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04327         void SetSystemParameter_IP_MEMORY_CONFIG(uint value);
04328
04329
04330         /// <summary>
04331         /// Getter for IP_NETMASK system parameter
04332         /// </summary>
04333         /// <returns>Current value of the parameter</returns>
04334         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04335         uint GetSystemParameter_IP_NETMASK();
04336
04337         /// <summary>
04338         /// Setter for IP_NETMASK system parameter
04339         /// </summary>
04340         /// <param name="value">New value to set to parameter</param>
04341         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04342         void SetSystemParameter_IP_NETMASK(uint value);
04343
04344
04345         /// <summary>
04346         /// Getter for IP_PROTOCOL_CONFIG system parameter
04347         /// </summary>
04348         /// <returns>Current value of the parameter</returns>
04349         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04350         uint GetSystemParameter_IP_PROTOCOL_CONFIG();
04351
04352         /// <summary>
04353         /// Setter for IP_PROTOCOL_CONFIG system parameter
04354         /// </summary>
04355         /// <param name="value">New value to set to parameter</param>
04356         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04357         void SetSystemParameter_IP_PROTOCOL_CONFIG(uint value);
04358
04359
04360         /// <summary>
04361         /// Getter for IP_PROTOCOL_CTRL system parameter
04362         /// </summary>
04363         /// <returns>Current value of the parameter</returns>
04364         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04365         uint GetSystemParameter_IP_PROTOCOL_CTRL();
04366
04367         /// <summary>
04368         /// Setter for IP_PROTOCOL_CTRL system parameter
04369         /// </summary>
04370         /// <param name="value">New value to set to parameter</param>
04371         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04372         void SetSystemParameter_IP_PROTOCOL_CTRL(uint value);
04373
04374
04375         /// <summary>
04376         /// Getter for IP_TCP_TIMEOUT system parameter
04377         /// </summary>
04378         /// <returns>Current value of the parameter</returns>
04379         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04380         ulong GetSystemParameter_IP_TCP_TIMEOUT();
04381
04382         /// <summary>
04383         /// Setter for IP_TCP_TIMEOUT system parameter
04384         /// </summary>
04385         /// <param name="value">New value to set to parameter</param>
04386         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04387         void SetSystemParameter_IP_TCP_TIMEOUT(ulong value);
04388
04389
04390         /// <summary>
04391         /// Getter for IP_TCP_TX_THRESHOLD system parameter
04392         /// </summary>
04393         /// <returns>Current value of the parameter</returns>
04394         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04395         uint GetSystemParameter_IP_TCP_TX_THRESHOLD();
04396
04397         /// <summary>
```

```
04398         /// Setter for IP_TCP_TX_THRESHOLD system parameter
04399         /// </summary>
04400         /// <param name="value">New value to set to parameter</param>
04401         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04402         void SetSystemParameter_IP_TCP_TX_THRESHOLD(uint value);
04403
04404
04405         /// <summary>
04406         /// Getter for IP_TCP_TX_TIMEOUT system parameter
04407         /// </summary>
04408         /// <returns>Current value of the parameter</returns>
04409         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04410         uint GetSystemParameter_IP_TCP_TX_TIMEOUT();
04411
04412         /// <summary>
04413         /// Setter for IP_TCP_TX_TIMEOUT system parameter
04414         /// </summary>
04415         /// <param name="value">New value to set to parameter</param>
04416         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04417         void SetSystemParameter_IP_TCP_TX_TIMEOUT(uint value);
04418
04419
04420         /// <summary>
04421         /// Getter for LAST_AXIS system parameter
04422         /// </summary>
04423         /// <returns>Current value of the parameter</returns>
04424         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04425         int GetSystemParameter_LAST_AXIS();
04426
04427         /// <summary>
04428         /// Setter for LAST_AXIS system parameter
04429         /// </summary>
04430         /// <param name="value">New value to set to parameter</param>
04431         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04432         void SetSystemParameter_LAST_AXIS(int value);
04433
04434
04435         /// <summary>
04436         /// Getter for MODBUS_TX_DELAY system parameter
04437         /// </summary>
04438         /// <returns>Current value of the parameter</returns>
04439         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04440         ushort GetSystemParameter_MODBUS_TX_DELAY();
04441
04442         /// <summary>
04443         /// Setter for MODBUS_TX_DELAY system parameter
04444         /// </summary>
04445         /// <param name="value">New value to set to parameter</param>
04446         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04447         void SetSystemParameter_MODBUS_TX_DELAY(ushort value);
04448
04449
04450         /// <summary>
04451         /// Getter for MOTION_ERROR system parameter
04452         /// </summary>
04453         /// <returns>Current value of the parameter</returns>
04454         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04455         ulong GetSystemParameter_MOTION_ERROR();
04456
04457
04458         /// <summary>
04459         /// Getter for MPE_MODE system parameter
04460         /// </summary>
04461         /// <returns>Current value of the parameter</returns>
04462         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04463         short GetSystemParameter_MPE_MODE();
04464
04465
04466         /// <summary>
04467         /// Getter for NOP system parameter
04468         /// </summary>
04469         /// <returns>Current value of the parameter</returns>
04470         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04471         uint GetSystemParameter_NOP();
04472
04473
04474         /// <summary>
```

```
04475         /// Getter for NIN system parameter
04476         /// </summary>
04477         /// <returns>Current value of the parameter</returns>
04478         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04479         uint GetSystemParameter_NIN();
04480
04481
04482         /// <summary>
04483         /// Getter for NIO system parameter
04484         /// </summary>
04485         /// <returns>Current value of the parameter</returns>
04486         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04487         uint GetSystemParameter_NIO();
04488
04489         /// <summary>
04490         /// Setter for NIO system parameter
04491         /// </summary>
04492         /// <param name="value">New value to set to parameter</param>
04493         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04494         void SetSystemParameter_NIO(uint value);
04495
04496
04497         /// <summary>
04498         /// Getter for N_ANA_IN system parameter
04499         /// </summary>
04500         /// <returns>Current value of the parameter</returns>
04501         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04502         uint GetSystemParameter_N_ANA_IN();
04503
04504
04505         /// <summary>
04506         /// Getter for N_ANA_OUT system parameter
04507         /// </summary>
04508         /// <returns>Current value of the parameter</returns>
04509         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04510         uint GetSystemParameter_N_ANA_OUT();
04511
04512
04513         /// <summary>
04514         /// Getter for PCMCAT_PROC system parameter
04515         /// </summary>
04516         /// <returns>Current value of the parameter</returns>
04517         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04518         int GetSystemParameter_PCMCAT_PROC();
04519
04520         /// <summary>
04521         /// Setter for PCMCAT_PROC system parameter
04522         /// </summary>
04523         /// <param name="value">New value to set to parameter</param>
04524         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04525         void SetSystemParameter_PCMCAT_PROC(int value);
04526
04527
04528         /// <summary>
04529         /// Getter for PLC_CONFIG system parameter
04530         /// </summary>
04531         /// <returns>Current value of the parameter</returns>
04532         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04533         int GetSystemParameter_PLC_CONFIG();
04534
04535         /// <summary>
04536         /// Setter for PLC_CONFIG system parameter
04537         /// </summary>
04538         /// <param name="value">New value to set to parameter</param>
04539         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04540         void SetSystemParameter_PLC_CONFIG(int value);
04541
04542
04543         /// <summary>
04544         /// Getter for PLC_ERROR system parameter
04545         /// </summary>
04546         /// <returns>Current value of the parameter</returns>
04547         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04548         long GetSystemParameter_PLC_ERROR();
04549
04550
04551         /// <summary>
```

```
04552         /// Getter for PLC_OVERFLOW system parameter
04553         /// </summary>
04554         /// <returns>Current value of the parameter</returns>
04555         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04556         long GetSystemParameter_PLC_OVERFLOW();
04557
04558
04559         /// <summary>
04560         /// Getter for PLC_RUN system parameter
04561         /// </summary>
04562         /// <returns>Current value of the parameter</returns>
04563         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04564         long GetSystemParameter_PLC_RUN();
04565
04566
04567         /// <summary>
04568         /// Getter for REMOTE_PROC system parameter
04569         /// </summary>
04570         /// <returns>Current value of the parameter</returns>
04571         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04572         int GetSystemParameter_REMOTE_PROC();
04573
04574         /// <summary>
04575         /// Setter for REMOTE_PROC system parameter
04576         /// </summary>
04577         /// <param name="value">New value to set to parameter</param>
04578         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04579         void SetSystemParameter_REMOTE_PROC(int value);
04580
04581
04582         /// <summary>
04583         /// Getter for SCHEDULE_TYPE system parameter
04584         /// </summary>
04585         /// <returns>Current value of the parameter</returns>
04586         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04587         uint GetSystemParameter_SCHEDULE_TYPE();
04588
04589         /// <summary>
04590         /// Setter for SCHEDULE_TYPE system parameter
04591         /// </summary>
04592         /// <param name="value">New value to set to parameter</param>
04593         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04594         void SetSystemParameter_SCHEDULE_TYPE(uint value);
04595
04596
04597         /// <summary>
04598         /// Getter for SCOPE_CYCLE_COUNT system parameter
04599         /// </summary>
04600         /// <returns>Current value of the parameter</returns>
04601         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04602         int GetSystemParameter_SCOPE_CYCLE_COUNT();
04603
04604         /// <summary>
04605         /// Setter for SCOPE_CYCLE_COUNT system parameter
04606         /// </summary>
04607         /// <param name="value">New value to set to parameter</param>
04608         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04609         void SetSystemParameter_SCOPE_CYCLE_COUNT(int value);
04610
04611
04612         /// <summary>
04613         /// Getter for SCOPE_DELAY system parameter
04614         /// </summary>
04615         /// <returns>Current value of the parameter</returns>
04616         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04617         long GetSystemParameter_SCOPE_DELAY();
04618
04619         /// <summary>
04620         /// Setter for SCOPE_DELAY system parameter
04621         /// </summary>
04622         /// <param name="value">New value to set to parameter</param>
04623         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04624         void SetSystemParameter_SCOPE_DELAY(long value);
04625
04626
04627         /// <summary>
04628         /// Getter for SCOPE_POS system parameter
```

```
04629         /// </summary>
04630         /// <returns>Current value of the parameter</returns>
04631         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04632         int GetSystemParameter_SCOPE_POS();
04633
04634         /// <summary>
04635         /// Setter for SCOPE_POS system parameter
04636         /// </summary>
04637         /// <param name="value">New value to set to parameter</param>
04638         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04639         void SetSystemParameter_SCOPE_POS(int value);
04640
04641
04642         /// <summary>
04643         /// Getter for SCOPE_TRIGGER_POS system parameter
04644         /// </summary>
04645         /// <returns>Current value of the parameter</returns>
04646         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04647         int GetSystemParameter_SCOPE_TRIGGER_POS();
04648
04649
04650         /// <summary>
04651         /// Getter for SERIAL_NUMBER system parameter
04652         /// </summary>
04653         /// <returns>Current value of the parameter</returns>
04654         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04655         int GetSystemParameter_SERIAL_NUMBER();
04656
04657         /// <summary>
04658         /// Setter for SERIAL_NUMBER system parameter
04659         /// </summary>
04660         /// <param name="value">New value to set to parameter</param>
04661         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04662         void SetSystemParameter_SERIAL_NUMBER(int value);
04663
04664
04665         /// <summary>
04666         /// Getter for SERVO_OFFSET system parameter
04667         /// </summary>
04668         /// <returns>Current value of the parameter</returns>
04669         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04670         int GetSystemParameter_SERVO_OFFSET();
04671
04672         /// <summary>
04673         /// Setter for SERVO_OFFSET system parameter
04674         /// </summary>
04675         /// <param name="value">New value to set to parameter</param>
04676         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04677         void SetSystemParameter_SERVO_OFFSET(int value);
04678
04679
04680         /// <summary>
04681         /// Getter for SERVO_PERIOD system parameter
04682         /// </summary>
04683         /// <returns>Current value of the parameter</returns>
04684         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04685         int GetSystemParameter_SERVO_PERIOD();
04686
04687         /// <summary>
04688         /// Setter for SERVO_PERIOD system parameter
04689         /// </summary>
04690         /// <param name="value">New value to set to parameter</param>
04691         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04692         void SetSystemParameter_SERVO_PERIOD(int value);
04693
04694
04695         /// <summary>
04696         /// Getter for SYSTEM_ERROR_MASK system parameter
04697         /// </summary>
04698         /// <returns>Current value of the parameter</returns>
04699         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04700         uint GetSystemParameter_SYSTEM_ERROR_MASK();
04701
04702         /// <summary>
04703         /// Setter for SYSTEM_ERROR_MASK system parameter
04704         /// </summary>
04705         /// <param name="value">New value to set to parameter</param>
```

```
04706         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04707         void SetSystemParameter_SYSTEM_ERROR_MASK(uint value);
04708
04709
04710         /// <summary>
04711         /// Getter for SYSTEM_LOAD system parameter
04712         /// </summary>
04713         /// <returns>Current value of the parameter</returns>
04714         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04715         uint GetSystemParameter_SYSTEM_LOAD();
04716
04717
04718         /// <summary>
04719         /// Getter for SYSTEM_LOAD_MAX system parameter
04720         /// </summary>
04721         /// <returns>Current value of the parameter</returns>
04722         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04723         uint GetSystemParameter_SYSTEM_LOAD_MAX();
04724
04725         /// <summary>
04726         /// Setter for SYSTEM_LOAD_MAX system parameter
04727         /// </summary>
04728         /// <param name="value">New value to set to parameter</param>
04729         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04730         void SetSystemParameter_SYSTEM_LOAD_MAX(uint value);
04731
04732
04733         /// <summary>
04734         /// Getter for TEXT_FILE_LOADER_PROC system parameter
04735         /// </summary>
04736         /// <returns>Current value of the parameter</returns>
04737         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04738         int GetSystemParameter_TEXT_FILE_LOADER_PROC();
04739
04740         /// <summary>
04741         /// Setter for TEXT_FILE_LOADER_PROC system parameter
04742         /// </summary>
04743         /// <param name="value">New value to set to parameter</param>
04744         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04745         void SetSystemParameter_TEXT_FILE_LOADER_PROC(int value);
04746
04747
04748         /// <summary>
04749         /// Getter for TSIZE system parameter
04750         /// </summary>
04751         /// <returns>Current value of the parameter</returns>
04752         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04753         int GetSystemParameter_TSIZE();
04754
04755
04756         /// <summary>
04757         /// Getter for UNIT_ERROR system parameter
04758         /// </summary>
04759         /// <returns>Current value of the parameter</returns>
04760         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04761         uint GetSystemParameter_UNIT_ERROR();
04762
04763
04764         /// <summary>
04765         /// Getter for VERSION system parameter
04766         /// </summary>
04767         /// <returns>Current value of the parameter</returns>
04768         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04769         double GetSystemParameter_VERSION();
04770
04771
04772         /// <summary>
04773         /// Getter for VRSIZE system parameter
04774         /// </summary>
04775         /// <returns>Current value of the parameter</returns>
04776         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04777         int GetSystemParameter_VRSIZE();
04778
04779
04780         /// <summary>
04781         /// Getter for PROCNUMBER system parameter
04782         /// </summary>
```

```
04783         /// <returns>Current value of the parameter</returns>
04784         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04785         short GetSystemParameter_PROCNUMBER();
04786
04787
04788
      ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
04789
04790
      ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
04791
04792         /// <summary>
04793         /// Getter for ACCEL axis parameter
04794         /// </summary>
04795         /// <param name="axis">Target axis whose parameter is retrieved</param>
04796         /// <returns>Current value of the parameter</returns>
04797         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04798         double GetAxisParameter_ACCEL(int axis);
04799
04800         /// <summary>
04801         /// Setter for ACCEL axis parameter
04802         /// </summary>
04803         /// <param name="axis">Target axis whose parameter is modified</param>
04804         /// <param name="value">New value to set to parameter</param>
04805         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04806         void SetAxisParameter_ACCEL(int axis, double value);
04807
04808
04809         /// <summary>
04810         /// Getter for ADDAX_AXIS axis parameter
04811         /// </summary>
04812         /// <param name="axis">Target axis whose parameter is retrieved</param>
04813         /// <returns>Current value of the parameter</returns>
04814         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04815         int GetAxisParameter_ADDAX_AXIS(int axis);
04816
04817
04818         /// <summary>
04819         /// Getter for AFF_GAIN axis parameter
04820         /// </summary>
04821         /// <param name="axis">Target axis whose parameter is retrieved</param>
04822         /// <returns>Current value of the parameter</returns>
04823         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04824         double GetAxisParameter_AFF_GAIN(int axis);
04825
04826         /// <summary>
04827         /// Setter for AFF_GAIN axis parameter
04828         /// </summary>
04829         /// <param name="axis">Target axis whose parameter is modified</param>
04830         /// <param name="value">New value to set to parameter</param>
04831         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04832         void SetAxisParameter_AFF_GAIN(int axis, double value);
04833
04834
04835         /// <summary>
04836         /// Getter for ATYPE axis parameter
04837         /// </summary>
04838         /// <param name="axis">Target axis whose parameter is retrieved</param>
04839         /// <returns>Current value of the parameter</returns>
04840         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04841         AxisType GetAxisParameter_ATYPE(int axis);
04842
04843         /// <summary>
04844         /// Setter for ATYPE axis parameter
04845         /// </summary>
04846         /// <param name="axis">Target axis whose parameter is modified</param>
04847         /// <param name="value">New value to set to parameter</param>
04848         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04849         void SetAxisParameter_ATYPE(int axis, AxisType value);
04850
04851
04852         /// <summary>
04853         /// Getter for AXIS_ACCEL axis parameter
04854         /// </summary>
04855         /// <param name="axis">Target axis whose parameter is retrieved</param>
04856         /// <returns>Current value of the parameter</returns>
04857         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04858         double GetAxisParameter_AXIS_ACCEL(int axis);
```

```
04859
04860         /// <summary>
04861         /// Setter for AXIS_ACCEL axis parameter
04862         /// </summary>
04863         /// <param name="axis">Target axis whose parameter is modified</param>
04864         /// <param name="value">New value to set to parameter</param>
04865         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04866         void SetAxisParameter_AXIS_ACCEL(int axis, double value);
04867
04868
04869         /// <summary>
04870         /// Getter for AXIS_ADDRESS axis parameter
04871         /// </summary>
04872         /// <param name="axis">Target axis whose parameter is retrieved</param>
04873         /// <returns>Current value of the parameter</returns>
04874         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04875         int GetAxisParameter_AXIS_ADDRESS(int axis);
04876
04877         /// <summary>
04878         /// Setter for AXIS_ADDRESS axis parameter
04879         /// </summary>
04880         /// <param name="axis">Target axis whose parameter is modified</param>
04881         /// <param name="value">New value to set to parameter</param>
04882         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04883         void SetAxisParameter_AXIS_ADDRESS(int axis, int value);
04884
04885
04886         /// <summary>
04887         /// Getter for AXIS_DECEL axis parameter
04888         /// </summary>
04889         /// <param name="axis">Target axis whose parameter is retrieved</param>
04890         /// <returns>Current value of the parameter</returns>
04891         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04892         double GetAxisParameter_AXIS_DECEL(int axis);
04893
04894         /// <summary>
04895         /// Setter for AXIS_DECEL axis parameter
04896         /// </summary>
04897         /// <param name="axis">Target axis whose parameter is modified</param>
04898         /// <param name="value">New value to set to parameter</param>
04899         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04900         void SetAxisParameter_AXIS_DECEL(int axis, double value);
04901
04902
04903         /// <summary>
04904         /// Getter for AXIS_DPOS axis parameter
04905         /// </summary>
04906         /// <param name="axis">Target axis whose parameter is retrieved</param>
04907         /// <returns>Current value of the parameter</returns>
04908         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04909         double GetAxisParameter_AXIS_DPOS(int axis);
04910
04911
04912         /// <summary>
04913         /// Getter for AXIS_ENABLE axis parameter
04914         /// </summary>
04915         /// <param name="axis">Target axis whose parameter is retrieved</param>
04916         /// <returns>Current value of the parameter</returns>
04917         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04918         bool GetAxisParameter_AXIS_ENABLE(int axis);
04919
04920         /// <summary>
04921         /// Setter for AXIS_ENABLE axis parameter
04922         /// </summary>
04923         /// <param name="axis">Target axis whose parameter is modified</param>
04924         /// <param name="value">New value to set to parameter</param>
04925         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04926         void SetAxisParameter_AXIS_ENABLE(int axis, bool value);
04927
04928
04929         /// <summary>
04930         /// Getter for AXIS_ERROR_COUNT axis parameter
04931         /// </summary>
04932         /// <param name="axis">Target axis whose parameter is retrieved</param>
04933         /// <returns>Current value of the parameter</returns>
04934         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04935         int GetAxisParameter_AXIS_ERROR_COUNT(int axis);
04936
```

```
04937         /// <summary>
04938         /// Setter for AXIS_ERROR_COUNT axis parameter
04939         /// </summary>
04940         /// <param name="axis">Target axis whose parameter is modified</param>
04941         /// <param name="value">New value to set to parameter</param>
04942         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04943         void SetAxisParameter_AXIS_ERROR_COUNT(int axis, int value);
04944
04945
04946         /// <summary>
04947         /// Getter for AXIS_FS_LIMIT axis parameter
04948         /// </summary>
04949         /// <param name="axis">Target axis whose parameter is retrieved</param>
04950         /// <returns>Current value of the parameter</returns>
04951         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04952         double GetAxisParameter_AXIS_FS_LIMIT(int axis);
04953
04954         /// <summary>
04955         /// Setter for AXIS_FS_LIMIT axis parameter
04956         /// </summary>
04957         /// <param name="axis">Target axis whose parameter is modified</param>
04958         /// <param name="value">New value to set to parameter</param>
04959         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04960         void SetAxisParameter_AXIS_FS_LIMIT(int axis, double value);
04961
04962
04963         /// <summary>
04964         /// Getter for AXIS_MAX_JOG_SPEED axis parameter
04965         /// </summary>
04966         /// <param name="axis">Target axis whose parameter is retrieved</param>
04967         /// <returns>Current value of the parameter</returns>
04968         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04969         double GetAxisParameter_AXIS_MAX_JOG_SPEED(int axis);
04970
04971         /// <summary>
04972         /// Setter for AXIS_MAX_JOG_SPEED axis parameter
04973         /// </summary>
04974         /// <param name="axis">Target axis whose parameter is modified</param>
04975         /// <param name="value">New value to set to parameter</param>
04976         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04977         void SetAxisParameter_AXIS_MAX_JOG_SPEED(int axis, double value);
04978
04979
04980         /// <summary>
04981         /// Getter for AXIS_MODE axis parameter
04982         /// </summary>
04983         /// <param name="axis">Target axis whose parameter is retrieved</param>
04984         /// <returns>Current value of the parameter</returns>
04985         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04986         AxisMode GetAxisParameter_AXIS_MODE(int axis);
04987
04988         /// <summary>
04989         /// Setter for AXIS_MODE axis parameter
04990         /// </summary>
04991         /// <param name="axis">Target axis whose parameter is modified</param>
04992         /// <param name="value">New value to set to parameter</param>
04993         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
04994         void SetAxisParameter_AXIS_MODE(int axis, AxisMode value);
04995
04996
04997         /// <summary>
04998         /// Getter for AXIS_RS_LIMIT axis parameter
04999         /// </summary>
05000         /// <param name="axis">Target axis whose parameter is retrieved</param>
05001         /// <returns>Current value of the parameter</returns>
05002         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05003         double GetAxisParameter_AXIS_RS_LIMIT(int axis);
05004
05005         /// <summary>
05006         /// Setter for AXIS_RS_LIMIT axis parameter
05007         /// </summary>
05008         /// <param name="axis">Target axis whose parameter is modified</param>
05009         /// <param name="value">New value to set to parameter</param>
05010         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05011         void SetAxisParameter_AXIS_RS_LIMIT(int axis, double value);
05012
05013
05014         /// <summary>
```

```
05015         /// Getter for AXIS_SPEED axis parameter
05016         /// </summary>
05017         /// <param name="axis">Target axis whose parameter is retrieved</param>
05018         /// <returns>Current value of the parameter</returns>
05019         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05020         double GetAxisParameter_AXIS_SPEED(int axis);
05021
05022         /// <summary>
05023         /// Setter for AXIS_SPEED axis parameter
05024         /// </summary>
05025         /// <param name="axis">Target axis whose parameter is modified</param>
05026         /// <param name="value">New value to set to parameter</param>
05027         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05028         void SetAxisParameter_AXIS_SPEED(int axis, double value);
05029
05030
05031         /// <summary>
05032         /// Getter for AXIS_UNITS axis parameter
05033         /// </summary>
05034         /// <param name="axis">Target axis whose parameter is retrieved</param>
05035         /// <returns>Current value of the parameter</returns>
05036         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05037         double GetAxisParameter_AXIS_UNITS(int axis);
05038
05039         /// <summary>
05040         /// Setter for AXIS_UNITS axis parameter
05041         /// </summary>
05042         /// <param name="axis">Target axis whose parameter is modified</param>
05043         /// <param name="value">New value to set to parameter</param>
05044         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05045         void SetAxisParameter_AXIS_UNITS(int axis, double value);
05046
05047
05048         /// <summary>
05049         /// Getter for AXISSTATUS axis parameter
05050         /// </summary>
05051         /// <param name="axis">Target axis whose parameter is retrieved</param>
05052         /// <returns>Current value of the parameter</returns>
05053         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05054         AxisStatus GetAxisParameter_AXISSTATUS(int axis);
05055
05056
05057         /// <summary>
05058         /// Getter for BACKLASH_DIST axis parameter
05059         /// </summary>
05060         /// <param name="axis">Target axis whose parameter is retrieved</param>
05061         /// <returns>Current value of the parameter</returns>
05062         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05063         long GetAxisParameter_BACKLASH_DIST(int axis);
05064
05065         /// <summary>
05066         /// Setter for BACKLASH_DIST axis parameter
05067         /// </summary>
05068         /// <param name="axis">Target axis whose parameter is modified</param>
05069         /// <param name="value">New value to set to parameter</param>
05070         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05071         void SetAxisParameter_BACKLASH_DIST(int axis, long value);
05072
05073
05074         /// <summary>
05075         /// Getter for CHANGE_DIR_LAST axis parameter
05076         /// </summary>
05077         /// <param name="axis">Target axis whose parameter is retrieved</param>
05078         /// <returns>Current value of the parameter</returns>
05079         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05080         double GetAxisParameter_CHANGE_DIR_LAST(int axis);
05081
05082         /// <summary>
05083         /// Setter for CHANGE_DIR_LAST axis parameter
05084         /// </summary>
05085         /// <param name="axis">Target axis whose parameter is modified</param>
05086         /// <param name="value">New value to set to parameter</param>
05087         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05088         void SetAxisParameter_CHANGE_DIR_LAST(int axis, double value);
05089
05090
05091         /// <summary>
05092         /// Getter for CLOSE_WIN axis parameter
```

```
05093         /// </summary>
05094         /// <param name="axis">Target axis whose parameter is retrieved</param>
05095         /// <returns>Current value of the parameter</returns>
05096         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05097         double GetAxisParameter_CLOSE_WIN(int axis);
05098
05099         /// <summary>
05100         /// Setter for CLOSE_WIN axis parameter
05101         /// </summary>
05102         /// <param name="axis">Target axis whose parameter is modified</param>
05103         /// <param name="value">New value to set to parameter</param>
05104         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05105         void SetAxisParameter_CLOSE_WIN(int axis, double value);
05106
05107
05108         /// <summary>
05109         /// Getter for CLUTCH_RATE axis parameter
05110         /// </summary>
05111         /// <param name="axis">Target axis whose parameter is retrieved</param>
05112         /// <returns>Current value of the parameter</returns>
05113         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05114         double GetAxisParameter_CLUTCH_RATE(int axis);
05115
05116         /// <summary>
05117         /// Setter for CLUTCH_RATE axis parameter
05118         /// </summary>
05119         /// <param name="axis">Target axis whose parameter is modified</param>
05120         /// <param name="value">New value to set to parameter</param>
05121         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05122         void SetAxisParameter_CLUTCH_RATE(int axis, double value);
05123
05124
05125         /// <summary>
05126         /// Getter for CORNER_MODE axis parameter
05127         /// </summary>
05128         /// <param name="axis">Target axis whose parameter is retrieved</param>
05129         /// <returns>Current value of the parameter</returns>
05130         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05131         CornerMode GetAxisParameter_CORNER_MODE(int axis);
05132
05133         /// <summary>
05134         /// Setter for CORNER_MODE axis parameter
05135         /// </summary>
05136         /// <param name="axis">Target axis whose parameter is modified</param>
05137         /// <param name="value">New value to set to parameter</param>
05138         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05139         void SetAxisParameter_CORNER_MODE(int axis, CornerMode value);
05140
05141
05142         /// <summary>
05143         /// Getter for CORNER_STATE axis parameter
05144         /// </summary>
05145         /// <param name="axis">Target axis whose parameter is retrieved</param>
05146         /// <returns>Current value of the parameter</returns>
05147         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05148         CornerState GetAxisParameter_CORNER_STATE(int axis);
05149
05150         /// <summary>
05151         /// Setter for CORNER_STATE axis parameter
05152         /// </summary>
05153         /// <param name="axis">Target axis whose parameter is modified</param>
05154         /// <param name="value">New value to set to parameter</param>
05155         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05156         void SetAxisParameter_CORNER_STATE(int axis, CornerState value);
05157
05158
05159         /// <summary>
05160         /// Getter for CREEP axis parameter
05161         /// </summary>
05162         /// <param name="axis">Target axis whose parameter is retrieved</param>
05163         /// <returns>Current value of the parameter</returns>
05164         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05165         double GetAxisParameter_CREEP(int axis);
05166
05167         /// <summary>
05168         /// Setter for CREEP axis parameter
05169         /// </summary>
05170         /// <param name="axis">Target axis whose parameter is modified</param>
```

```
05171         /// <param name="value">New value to set to parameter</param>
05172         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05173         void SetAxisParameter_CREEP(int axis, double value);
05174
05175
05176         /// <summary>
05177         /// Getter for D_GAIN axis parameter
05178         /// </summary>
05179         /// <param name="axis">Target axis whose parameter is retrieved</param>
05180         /// <returns>Current value of the parameter</returns>
05181         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05182         double GetAxisParameter_D_GAIN(int axis);
05183
05184         /// <summary>
05185         /// Setter for D_GAIN axis parameter
05186         /// </summary>
05187         /// <param name="axis">Target axis whose parameter is modified</param>
05188         /// <param name="value">New value to set to parameter</param>
05189         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05190         void SetAxisParameter_D_GAIN(int axis, double value);
05191
05192
05193         /// <summary>
05194         /// Getter for DAC axis parameter
05195         /// </summary>
05196         /// <param name="axis">Target axis whose parameter is retrieved</param>
05197         /// <returns>Current value of the parameter</returns>
05198         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05199         int GetAxisParameter_DAC(int axis);
05200
05201         /// <summary>
05202         /// Setter for DAC axis parameter
05203         /// </summary>
05204         /// <param name="axis">Target axis whose parameter is modified</param>
05205         /// <param name="value">New value to set to parameter</param>
05206         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05207         void SetAxisParameter_DAC(int axis, int value);
05208
05209
05210         /// <summary>
05211         /// Getter for DAC_SCALE axis parameter
05212         /// </summary>
05213         /// <param name="axis">Target axis whose parameter is retrieved</param>
05214         /// <returns>Current value of the parameter</returns>
05215         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05216         long GetAxisParameter_DAC_SCALE(int axis);
05217
05218         /// <summary>
05219         /// Setter for DAC_SCALE axis parameter
05220         /// </summary>
05221         /// <param name="axis">Target axis whose parameter is modified</param>
05222         /// <param name="value">New value to set to parameter</param>
05223         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05224         void SetAxisParameter_DAC_SCALE(int axis, long value);
05225
05226
05227         /// <summary>
05228         /// Getter for DAC_OUT axis parameter
05229         /// </summary>
05230         /// <param name="axis">Target axis whose parameter is retrieved</param>
05231         /// <returns>Current value of the parameter</returns>
05232         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05233         int GetAxisParameter_DAC_OUT(int axis);
05234
05235
05236         /// <summary>
05237         /// Getter for DATUM_IN axis parameter
05238         /// </summary>
05239         /// <param name="axis">Target axis whose parameter is retrieved</param>
05240         /// <returns>Current value of the parameter</returns>
05241         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05242         short GetAxisParameter_DATUM_IN(int axis);
05243
05244         /// <summary>
05245         /// Setter for DATUM_IN axis parameter
05246         /// </summary>
05247         /// <param name="axis">Target axis whose parameter is modified</param>
05248         /// <param name="value">New value to set to parameter</param>
```

```
05249         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05250         void SetAxisParameter_DATUM_IN(int axis, short value);
05251
05252
05253         /// <summary>
05254         /// Getter for DECEL axis parameter
05255         /// </summary>
05256         /// <param name="axis">Target axis whose parameter is retrieved</param>
05257         /// <returns>Current value of the parameter</returns>
05258         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05259         double GetAxisParameter_DECEL(int axis);
05260
05261         /// <summary>
05262         /// Setter for DECEL axis parameter
05263         /// </summary>
05264         /// <param name="axis">Target axis whose parameter is modified</param>
05265         /// <param name="value">New value to set to parameter</param>
05266         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05267         void SetAxisParameter_DECEL(int axis, double value);
05268
05269
05270         /// <summary>
05271         /// Getter for DECEL_ANGLE axis parameter
05272         /// </summary>
05273         /// <param name="axis">Target axis whose parameter is retrieved</param>
05274         /// <returns>Current value of the parameter</returns>
05275         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05276         double GetAxisParameter_DECEL_ANGLE(int axis);
05277
05278         /// <summary>
05279         /// Setter for DECEL_ANGLE axis parameter
05280         /// </summary>
05281         /// <param name="axis">Target axis whose parameter is modified</param>
05282         /// <param name="value">New value to set to parameter</param>
05283         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05284         void SetAxisParameter_DECEL_ANGLE(int axis, double value);
05285
05286
05287         /// <summary>
05288         /// Getter for DEMAND_SPEED axis parameter
05289         /// </summary>
05290         /// <param name="axis">Target axis whose parameter is retrieved</param>
05291         /// <returns>Current value of the parameter</returns>
05292         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05293         double GetAxisParameter_DEMAND_SPEED(int axis);
05294
05295
05296         /// <summary>
05297         /// Getter for DPOS axis parameter
05298         /// </summary>
05299         /// <param name="axis">Target axis whose parameter is retrieved</param>
05300         /// <returns>Current value of the parameter</returns>
05301         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05302         double GetAxisParameter_DPOS(int axis);
05303
05304
05305         /// <summary>
05306         /// Getter for DRIVE_CONTROL axis parameter
05307         /// </summary>
05308         /// <param name="axis">Target axis whose parameter is retrieved</param>
05309         /// <returns>Current value of the parameter</returns>
05310         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05311         int GetAxisParameter_DRIVE_CONTROL(int axis);
05312
05313         /// <summary>
05314         /// Setter for DRIVE_CONTROL axis parameter
05315         /// </summary>
05316         /// <param name="axis">Target axis whose parameter is modified</param>
05317         /// <param name="value">New value to set to parameter</param>
05318         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05319         void SetAxisParameter_DRIVE_CONTROL(int axis, int value);
05320
05321
05322         /// <summary>
05323         /// Getter for DRIVE_CONTROLWORD axis parameter
05324         /// </summary>
05325         /// <param name="axis">Target axis whose parameter is retrieved</param>
05326         /// <returns>Current value of the parameter</returns>
```

```
05327         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05328         int GetAxisParameter_DRIVE_CONTROLWORD(int axis);
05329
05330         /// <summary>
05331         /// Setter for DRIVE_CONTROLWORD axis parameter
05332         /// </summary>
05333         /// <param name="axis">Target axis whose parameter is modified</param>
05334         /// <param name="value">New value to set to parameter</param>
05335         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05336         void SetAxisParameter_DRIVE_CONTROLWORD(int axis, int value);
05337
05338
05339         /// <summary>
05340         /// Getter for DRIVE_CW_MODE axis parameter
05341         /// </summary>
05342         /// <param name="axis">Target axis whose parameter is retrieved</param>
05343         /// <returns>Current value of the parameter</returns>
05344         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05345         int GetAxisParameter_DRIVE_CW_MODE(int axis);
05346
05347         /// <summary>
05348         /// Setter for DRIVE_CW_MODE axis parameter
05349         /// </summary>
05350         /// <param name="axis">Target axis whose parameter is modified</param>
05351         /// <param name="value">New value to set to parameter</param>
05352         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05353         void SetAxisParameter_DRIVE_CW_MODE(int axis, int value);
05354
05355
05356         /// <summary>
05357         /// Getter for DRIVE_CURRENT axis parameter
05358         /// </summary>
05359         /// <param name="axis">Target axis whose parameter is retrieved</param>
05360         /// <returns>Current value of the parameter</returns>
05361         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05362         short GetAxisParameter_DRIVE_CURRENT(int axis);
05363
05364
05365         /// <summary>
05366         /// Getter for DRIVE_ENABLE axis parameter
05367         /// </summary>
05368         /// <param name="axis">Target axis whose parameter is retrieved</param>
05369         /// <returns>Current value of the parameter</returns>
05370         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05371         long GetAxisParameter_DRIVE_ENABLE(int axis);
05372
05373         /// <summary>
05374         /// Setter for DRIVE_ENABLE axis parameter
05375         /// </summary>
05376         /// <param name="axis">Target axis whose parameter is modified</param>
05377         /// <param name="value">New value to set to parameter</param>
05378         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05379         void SetAxisParameter_DRIVE_ENABLE(int axis, long value);
05380
05381
05382         /// <summary>
05383         /// Getter for DRIVE_FE axis parameter
05384         /// </summary>
05385         /// <param name="axis">Target axis whose parameter is retrieved</param>
05386         /// <returns>Current value of the parameter</returns>
05387         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05388         long GetAxisParameter_DRIVE_FE(int axis);
05389
05390
05391         /// <summary>
05392         /// Getter for DRIVE_FE_LIMIT axis parameter
05393         /// </summary>
05394         /// <param name="axis">Target axis whose parameter is retrieved</param>
05395         /// <returns>Current value of the parameter</returns>
05396         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05397         long GetAxisParameter_DRIVE_FE_LIMIT(int axis);
05398
05399         /// <summary>
05400         /// Setter for DRIVE_FE_LIMIT axis parameter
05401         /// </summary>
05402         /// <param name="axis">Target axis whose parameter is modified</param>
05403         /// <param name="value">New value to set to parameter</param>
05404         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
```

```
module.</exception>
05405         void SetAxisParameter_DRIVE_FE_LIMIT(int axis, long value);
05406
05407
05408         /// <summary>
05409         /// Getter for DRIVE_INPUTS axis parameter
05410         /// </summary>
05411         /// <param name="axis">Target axis whose parameter is retrieved</param>
05412         /// <returns>Current value of the parameter</returns>
05413         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
05414         uint GetAxisParameter_DRIVE_INPUTS(int axis);
05415
05416
05417         /// <summary>
05418         /// Getter for DRIVE_MODE axis parameter
05419         /// </summary>
05420         /// <param name="axis">Target axis whose parameter is retrieved</param>
05421         /// <returns>Current value of the parameter</returns>
05422         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
05423         int GetAxisParameter_DRIVE_MODE(int axis);
05424
05425         /// <summary>
05426         /// Setter for DRIVE_MODE axis parameter
05427         /// </summary>
05428         /// <param name="axis">Target axis whose parameter is modified</param>
05429         /// <param name="value">New value to set to parameter</param>
05430         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
05431         void SetAxisParameter_DRIVE_MODE(int axis, int value);
05432
05433
05434         /// <summary>
05435         /// Getter for DRIVE_MONITOR axis parameter
05436         /// </summary>
05437         /// <param name="axis">Target axis whose parameter is retrieved</param>
05438         /// <returns>Current value of the parameter</returns>
05439         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
05440         int GetAxisParameter_DRIVE_MONITOR(int axis);
05441
05442
05443         /// <summary>
05444         /// Getter for DRIVE_PROFILE axis parameter
05445         /// </summary>
05446         /// <param name="axis">Target axis whose parameter is retrieved</param>
05447         /// <returns>Current value of the parameter</returns>
05448         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
05449         int GetAxisParameter_DRIVE_PROFILE(int axis);
05450
05451         /// <summary>
05452         /// Setter for DRIVE_PROFILE axis parameter
05453         /// </summary>
05454         /// <param name="axis">Target axis whose parameter is modified</param>
05455         /// <param name="value">New value to set to parameter</param>
05456         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
05457         void SetAxisParameter_DRIVE_PROFILE(int axis, int value);
05458
05459
05460         /// <summary>
05461         /// Getter for DRIVE_STATUS axis parameter
05462         /// </summary>
05463         /// <param name="axis">Target axis whose parameter is retrieved</param>
05464         /// <returns>Current value of the parameter</returns>
05465         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
05466         int GetAxisParameter_DRIVE_STATUS(int axis);
05467
05468
05469         /// <summary>
05470         /// Getter for DRIVE_TORQUE axis parameter
05471         /// </summary>
05472         /// <param name="axis">Target axis whose parameter is retrieved</param>
05473         /// <returns>Current value of the parameter</returns>
05474         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
05475         int GetAxisParameter_DRIVE_TORQUE(int axis);
05476
05477
05478         /// <summary>
05479         /// Getter for DRIVE_TYPE axis parameter
05480         /// </summary>
05481         /// <param name="axis">Target axis whose parameter is retrieved</param>
05482         /// <returns>Current value of the parameter</returns>
```

```
05483         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05484         int GetAxisParameter_DRIVE_TYPE(int axis);
05485
05486         /// <summary>
05487         /// Setter for DRIVE_TYPE axis parameter
05488         /// </summary>
05489         /// <param name="axis">Target axis whose parameter is modified</param>
05490         /// <param name="value">New value to set to parameter</param>
05491         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05492         void SetAxisParameter_DRIVE_TYPE(int axis, int value);
05493
05494
05495         /// <summary>
05496         /// Getter for DRIVE_VELOCITY axis parameter
05497         /// </summary>
05498         /// <param name="axis">Target axis whose parameter is retrieved</param>
05499         /// <returns>Current value of the parameter</returns>
05500         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05501         int GetAxisParameter_DRIVE_VELOCITY(int axis);
05502
05503
05504         /// <summary>
05505         /// Getter for ENCODER axis parameter
05506         /// </summary>
05507         /// <param name="axis">Target axis whose parameter is retrieved</param>
05508         /// <returns>Current value of the parameter</returns>
05509         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05510         int GetAxisParameter_ENCODER(int axis);
05511
05512
05513         /// <summary>
05514         /// Getter for ENCODER_BITS axis parameter
05515         /// </summary>
05516         /// <param name="axis">Target axis whose parameter is retrieved</param>
05517         /// <returns>Current value of the parameter</returns>
05518         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05519         ushort GetAxisParameter_ENCODER_BITS(int axis);
05520
05521         /// <summary>
05522         /// Setter for ENCODER_BITS axis parameter
05523         /// </summary>
05524         /// <param name="axis">Target axis whose parameter is modified</param>
05525         /// <param name="value">New value to set to parameter</param>
05526         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05527         void SetAxisParameter_ENCODER_BITS(int axis, ushort value);
05528
05529
05530         /// <summary>
05531         /// Getter for ENCODER_CONTROL axis parameter
05532         /// </summary>
05533         /// <param name="axis">Target axis whose parameter is retrieved</param>
05534         /// <returns>Current value of the parameter</returns>
05535         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05536         int GetAxisParameter_ENCODER_CONTROL(int axis);
05537
05538         /// <summary>
05539         /// Setter for ENCODER_CONTROL axis parameter
05540         /// </summary>
05541         /// <param name="axis">Target axis whose parameter is modified</param>
05542         /// <param name="value">New value to set to parameter</param>
05543         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05544         void SetAxisParameter_ENCODER_CONTROL(int axis, int value);
05545
05546
05547         /// <summary>
05548         /// Getter for ENCODER_FILTER axis parameter
05549         /// </summary>
05550         /// <param name="axis">Target axis whose parameter is retrieved</param>
05551         /// <returns>Current value of the parameter</returns>
05552         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05553         double GetAxisParameter_ENCODER_FILTER(int axis);
05554
05555         /// <summary>
05556         /// Setter for ENCODER_FILTER axis parameter
05557         /// </summary>
05558         /// <param name="axis">Target axis whose parameter is modified</param>
05559         /// <param name="value">New value to set to parameter</param>
05560         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
```

```
module.</exception>
05561         void SetAxisParameter_ENCODER_FILTER(int axis, double value);
05562
05563
05564         /// <summary>
05565         /// Getter for ENCODER_ID axis parameter
05566         /// </summary>
05567         /// <param name="axis">Target axis whose parameter is retrieved</param>
05568         /// <returns>Current value of the parameter</returns>
05569         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
05570         int GetAxisParameter_ENCODER_ID(int axis);
05571
05572         /// <summary>
05573         /// Setter for ENCODER_ID axis parameter
05574         /// </summary>
05575         /// <param name="axis">Target axis whose parameter is modified</param>
05576         /// <param name="value">New value to set to parameter</param>
05577         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
05578         void SetAxisParameter_ENCODER_ID(int axis, int value);
05579
05580
05581         /// <summary>
05582         /// Getter for ENCODER_STATUS axis parameter
05583         /// </summary>
05584         /// <param name="axis">Target axis whose parameter is retrieved</param>
05585         /// <returns>Current value of the parameter</returns>
05586         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
05587         int GetAxisParameter_ENCODER_STATUS(int axis);
05588
05589         /// <summary>
05590         /// Setter for ENCODER_STATUS axis parameter
05591         /// </summary>
05592         /// <param name="axis">Target axis whose parameter is modified</param>
05593         /// <param name="value">New value to set to parameter</param>
05594         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
05595         void SetAxisParameter_ENCODER_STATUS(int axis, int value);
05596
05597
05598         /// <summary>
05599         /// Getter for ENCODER_TURNS axis parameter
05600         /// </summary>
05601         /// <param name="axis">Target axis whose parameter is retrieved</param>
05602         /// <returns>Current value of the parameter</returns>
05603         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
05604         int GetAxisParameter_ENCODER_TURNS(int axis);
05605
05606         /// <summary>
05607         /// Setter for ENCODER_TURNS axis parameter
05608         /// </summary>
05609         /// <param name="axis">Target axis whose parameter is modified</param>
05610         /// <param name="value">New value to set to parameter</param>
05611         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
05612         void SetAxisParameter_ENCODER_TURNS(int axis, int value);
05613
05614
05615         /// <summary>
05616         /// Getter for END_DIR_LAST axis parameter
05617         /// </summary>
05618         /// <param name="axis">Target axis whose parameter is retrieved</param>
05619         /// <returns>Current value of the parameter</returns>
05620         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
05621         double GetAxisParameter_END_DIR_LAST(int axis);
05622
05623         /// <summary>
05624         /// Setter for END_DIR_LAST axis parameter
05625         /// </summary>
05626         /// <param name="axis">Target axis whose parameter is modified</param>
05627         /// <param name="value">New value to set to parameter</param>
05628         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
05629         void SetAxisParameter_END_DIR_LAST(int axis, double value);
05630
05631
05632         /// <summary>
05633         /// Getter for ENDMOVE axis parameter
05634         /// </summary>
05635         /// <param name="axis">Target axis whose parameter is retrieved</param>
05636         /// <returns>Current value of the parameter</returns>
05637         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
```

```
05638         long GetAxisParameter_ENDMOVE(int axis);
05639
05640         /// <summary>
05641         /// Setter for ENDMOVE axis parameter
05642         /// </summary>
05643         /// <param name="axis">Target axis whose parameter is modified</param>
05644         /// <param name="value">New value to set to parameter</param>
05645         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05646         void SetAxisParameter_ENDMOVE(int axis, long value);
05647
05648
05649         /// <summary>
05650         /// Getter for ENDMOVE_BUFFER axis parameter
05651         /// </summary>
05652         /// <param name="axis">Target axis whose parameter is retrieved</param>
05653         /// <returns>Current value of the parameter</returns>
05654         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05655         double GetAxisParameter_ENDMOVE_BUFFER(int axis);
05656
05657         /// <summary>
05658         /// Setter for ENDMOVE_BUFFER axis parameter
05659         /// </summary>
05660         /// <param name="axis">Target axis whose parameter is modified</param>
05661         /// <param name="value">New value to set to parameter</param>
05662         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05663         void SetAxisParameter_ENDMOVE_BUFFER(int axis, double value);
05664
05665
05666         /// <summary>
05667         /// Getter for ENDMOVE_SPEED axis parameter
05668         /// </summary>
05669         /// <param name="axis">Target axis whose parameter is retrieved</param>
05670         /// <returns>Current value of the parameter</returns>
05671         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05672         double GetAxisParameter_ENDMOVE_SPEED(int axis);
05673
05674         /// <summary>
05675         /// Setter for ENDMOVE_SPEED axis parameter
05676         /// </summary>
05677         /// <param name="axis">Target axis whose parameter is modified</param>
05678         /// <param name="value">New value to set to parameter</param>
05679         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05680         void SetAxisParameter_ENDMOVE_SPEED(int axis, double value);
05681
05682
05683         /// <summary>
05684         /// Getter for ERRORMASK axis parameter
05685         /// </summary>
05686         /// <param name="axis">Target axis whose parameter is retrieved</param>
05687         /// <returns>Current value of the parameter</returns>
05688         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05689         AxisStatus GetAxisParameter_ERRORMASK(int axis);
05690
05691         /// <summary>
05692         /// Setter for ERRORMASK axis parameter
05693         /// </summary>
05694         /// <param name="axis">Target axis whose parameter is modified</param>
05695         /// <param name="value">New value to set to parameter</param>
05696         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05697         void SetAxisParameter_ERRORMASK(int axis, AxisStatus value);
05698
05699
05700         /// <summary>
05701         /// Getter for FAST_JOG axis parameter
05702         /// </summary>
05703         /// <param name="axis">Target axis whose parameter is retrieved</param>
05704         /// <returns>Current value of the parameter</returns>
05705         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05706         short GetAxisParameter_FAST_JOG(int axis);
05707
05708         /// <summary>
05709         /// Setter for FAST_JOG axis parameter
05710         /// </summary>
05711         /// <param name="axis">Target axis whose parameter is modified</param>
05712         /// <param name="value">New value to set to parameter</param>
05713         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05714         void SetAxisParameter_FAST_JOG(int axis, short value);
05715
```

```
05716
05717         /// <summary>
05718         /// Getter for FASTDEC axis parameter
05719         /// </summary>
05720         /// <param name="axis">Target axis whose parameter is retrieved</param>
05721         /// <returns>Current value of the parameter</returns>
05722         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05723         double GetAxisParameter_FASTDEC(int axis);
05724
05725         /// <summary>
05726         /// Setter for FASTDEC axis parameter
05727         /// </summary>
05728         /// <param name="axis">Target axis whose parameter is modified</param>
05729         /// <param name="value">New value to set to parameter</param>
05730         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05731         void SetAxisParameter_FASTDEC(int axis, double value);
05732
05733
05734         /// <summary>
05735         /// Getter for FE axis parameter
05736         /// </summary>
05737         /// <param name="axis">Target axis whose parameter is retrieved</param>
05738         /// <returns>Current value of the parameter</returns>
05739         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05740         double GetAxisParameter_FE(int axis);
05741
05742
05743         /// <summary>
05744         /// Getter for FE_LIMIT axis parameter
05745         /// </summary>
05746         /// <param name="axis">Target axis whose parameter is retrieved</param>
05747         /// <returns>Current value of the parameter</returns>
05748         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05749         double GetAxisParameter_FE_LIMIT(int axis);
05750
05751         /// <summary>
05752         /// Setter for FE_LIMIT axis parameter
05753         /// </summary>
05754         /// <param name="axis">Target axis whose parameter is modified</param>
05755         /// <param name="value">New value to set to parameter</param>
05756         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05757         void SetAxisParameter_FE_LIMIT(int axis, double value);
05758
05759
05760         /// <summary>
05761         /// Getter for FE_LIMIT_MODE axis parameter
05762         /// </summary>
05763         /// <param name="axis">Target axis whose parameter is retrieved</param>
05764         /// <returns>Current value of the parameter</returns>
05765         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05766         short GetAxisParameter_FE_LIMIT_MODE(int axis);
05767
05768         /// <summary>
05769         /// Setter for FE_LIMIT_MODE axis parameter
05770         /// </summary>
05771         /// <param name="axis">Target axis whose parameter is modified</param>
05772         /// <param name="value">New value to set to parameter</param>
05773         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05774         void SetAxisParameter_FE_LIMIT_MODE(int axis, short value);
05775
05776
05777         /// <summary>
05778         /// Getter for FE_RANGE axis parameter
05779         /// </summary>
05780         /// <param name="axis">Target axis whose parameter is retrieved</param>
05781         /// <returns>Current value of the parameter</returns>
05782         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05783         double GetAxisParameter_FE_RANGE(int axis);
05784
05785         /// <summary>
05786         /// Setter for FE_RANGE axis parameter
05787         /// </summary>
05788         /// <param name="axis">Target axis whose parameter is modified</param>
05789         /// <param name="value">New value to set to parameter</param>
05790         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05791         void SetAxisParameter_FE_RANGE(int axis, double value);
05792
05793
```

```
05794         /// <summary>
05795         /// Getter for FE_LATCH axis parameter
05796         /// </summary>
05797         /// <param name="axis">Target axis whose parameter is retrieved</param>
05798         /// <returns>Current value of the parameter</returns>
05799         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05800         double GetAxisParameter_FE_LATCH(int axis);
05801
05802
05803         /// <summary>
05804         /// Getter for FHOLD_IN axis parameter
05805         /// </summary>
05806         /// <param name="axis">Target axis whose parameter is retrieved</param>
05807         /// <returns>Current value of the parameter</returns>
05808         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05809         short GetAxisParameter_FHOLD_IN(int axis);
05810
05811         /// <summary>
05812         /// Setter for FHOLD_IN axis parameter
05813         /// </summary>
05814         /// <param name="axis">Target axis whose parameter is modified</param>
05815         /// <param name="value">New value to set to parameter</param>
05816         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05817         void SetAxisParameter_FHOLD_IN(int axis, short value);
05818
05819
05820         /// <summary>
05821         /// Getter for FHSPEED axis parameter
05822         /// </summary>
05823         /// <param name="axis">Target axis whose parameter is retrieved</param>
05824         /// <returns>Current value of the parameter</returns>
05825         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05826         double GetAxisParameter_FHSPEED(int axis);
05827
05828         /// <summary>
05829         /// Setter for FHSPEED axis parameter
05830         /// </summary>
05831         /// <param name="axis">Target axis whose parameter is modified</param>
05832         /// <param name="value">New value to set to parameter</param>
05833         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05834         void SetAxisParameter_FHSPEED(int axis, double value);
05835
05836
05837         /// <summary>
05838         /// Getter for FORCE_DWELL axis parameter
05839         /// </summary>
05840         /// <param name="axis">Target axis whose parameter is retrieved</param>
05841         /// <returns>Current value of the parameter</returns>
05842         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05843         int GetAxisParameter_FORCE_DWELL(int axis);
05844
05845         /// <summary>
05846         /// Setter for FORCE_DWELL axis parameter
05847         /// </summary>
05848         /// <param name="axis">Target axis whose parameter is modified</param>
05849         /// <param name="value">New value to set to parameter</param>
05850         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05851         void SetAxisParameter_FORCE_DWELL(int axis, int value);
05852
05853
05854         /// <summary>
05855         /// Getter for FORCE_RAMP axis parameter
05856         /// </summary>
05857         /// <param name="axis">Target axis whose parameter is retrieved</param>
05858         /// <returns>Current value of the parameter</returns>
05859         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05860         double GetAxisParameter_FORCE_RAMP(int axis);
05861
05862         /// <summary>
05863         /// Setter for FORCE_RAMP axis parameter
05864         /// </summary>
05865         /// <param name="axis">Target axis whose parameter is modified</param>
05866         /// <param name="value">New value to set to parameter</param>
05867         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05868         void SetAxisParameter_FORCE_RAMP(int axis, double value);
05869
05870
05871         /// <summary>
```

```
05872         /// Getter for FORCE_SPEED axis parameter
05873         /// </summary>
05874         /// <param name="axis">Target axis whose parameter is retrieved</param>
05875         /// <returns>Current value of the parameter</returns>
05876         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05877         double GetAxisParameter_FORCE_SPEED(int axis);
05878
05879         /// <summary>
05880         /// Setter for FORCE_SPEED axis parameter
05881         /// </summary>
05882         /// <param name="axis">Target axis whose parameter is modified</param>
05883         /// <param name="value">New value to set to parameter</param>
05884         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05885         void SetAxisParameter_FORCE_SPEED(int axis, double value);
05886
05887
05888         /// <summary>
05889         /// Getter for FRAME_REP_DIST axis parameter
05890         /// </summary>
05891         /// <param name="axis">Target axis whose parameter is retrieved</param>
05892         /// <returns>Current value of the parameter</returns>
05893         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05894         double GetAxisParameter_FRAME_REP_DIST(int axis);
05895
05896         /// <summary>
05897         /// Setter for FRAME_REP_DIST axis parameter
05898         /// </summary>
05899         /// <param name="axis">Target axis whose parameter is modified</param>
05900         /// <param name="value">New value to set to parameter</param>
05901         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05902         void SetAxisParameter_FRAME_REP_DIST(int axis, double value);
05903
05904
05905         /// <summary>
05906         /// Getter for FRAME_SCALE axis parameter
05907         /// </summary>
05908         /// <param name="axis">Target axis whose parameter is retrieved</param>
05909         /// <returns>Current value of the parameter</returns>
05910         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05911         double GetAxisParameter_FRAME_SCALE(int axis);
05912
05913         /// <summary>
05914         /// Setter for FRAME_SCALE axis parameter
05915         /// </summary>
05916         /// <param name="axis">Target axis whose parameter is modified</param>
05917         /// <param name="value">New value to set to parameter</param>
05918         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05919         void SetAxisParameter_FRAME_SCALE(int axis, double value);
05920
05921
05922         /// <summary>
05923         /// Getter for FS_LIMIT axis parameter
05924         /// </summary>
05925         /// <param name="axis">Target axis whose parameter is retrieved</param>
05926         /// <returns>Current value of the parameter</returns>
05927         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05928         double GetAxisParameter_FS_LIMIT(int axis);
05929
05930         /// <summary>
05931         /// Setter for FS_LIMIT axis parameter
05932         /// </summary>
05933         /// <param name="axis">Target axis whose parameter is modified</param>
05934         /// <param name="value">New value to set to parameter</param>
05935         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05936         void SetAxisParameter_FS_LIMIT(int axis, double value);
05937
05938
05939         /// <summary>
05940         /// Getter for FULL_SP_RADIUS axis parameter
05941         /// </summary>
05942         /// <param name="axis">Target axis whose parameter is retrieved</param>
05943         /// <returns>Current value of the parameter</returns>
05944         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05945         double GetAxisParameter_FULL_SP_RADIUS(int axis);
05946
05947         /// <summary>
05948         /// Setter for FULL_SP_RADIUS axis parameter
05949         /// </summary>
```

```
05950         /// <param name="axis">Target axis whose parameter is modified</param>
05951         /// <param name="value">New value to set to parameter</param>
05952         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05953         void SetAxisParameter_FULL_SP_RADIUS(int axis, double value);
05954
05955
05956         /// <summary>
05957         /// Getter for FWD_IN axis parameter
05958         /// </summary>
05959         /// <param name="axis">Target axis whose parameter is retrieved</param>
05960         /// <returns>Current value of the parameter</returns>
05961         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05962         short GetAxisParameter_FWD_IN(int axis);
05963
05964         /// <summary>
05965         /// Setter for FWD_IN axis parameter
05966         /// </summary>
05967         /// <param name="axis">Target axis whose parameter is modified</param>
05968         /// <param name="value">New value to set to parameter</param>
05969         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05970         void SetAxisParameter_FWD_IN(int axis, short value);
05971
05972
05973         /// <summary>
05974         /// Getter for FWD_JOG axis parameter
05975         /// </summary>
05976         /// <param name="axis">Target axis whose parameter is retrieved</param>
05977         /// <returns>Current value of the parameter</returns>
05978         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05979         short GetAxisParameter_FWD_JOG(int axis);
05980
05981         /// <summary>
05982         /// Setter for FWD_JOG axis parameter
05983         /// </summary>
05984         /// <param name="axis">Target axis whose parameter is modified</param>
05985         /// <param name="value">New value to set to parameter</param>
05986         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05987         void SetAxisParameter_FWD_JOG(int axis, short value);
05988
05989
05990         /// <summary>
05991         /// Getter for I_GAIN axis parameter
05992         /// </summary>
05993         /// <param name="axis">Target axis whose parameter is retrieved</param>
05994         /// <returns>Current value of the parameter</returns>
05995         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
05996         double GetAxisParameter_I_GAIN(int axis);
05997
05998         /// <summary>
05999         /// Setter for I_GAIN axis parameter
06000         /// </summary>
06001         /// <param name="axis">Target axis whose parameter is modified</param>
06002         /// <param name="value">New value to set to parameter</param>
06003         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06004         void SetAxisParameter_I_GAIN(int axis, double value);
06005
06006
06007         /// <summary>
06008         /// Getter for IDLE axis parameter
06009         /// </summary>
06010         /// <param name="axis">Target axis whose parameter is retrieved</param>
06011         /// <returns>Current value of the parameter</returns>
06012         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06013         bool GetAxisParameter_IDLE(int axis);
06014
06015
06016         /// <summary>
06017         /// Getter for IN_POS axis parameter
06018         /// </summary>
06019         /// <param name="axis">Target axis whose parameter is retrieved</param>
06020         /// <returns>Current value of the parameter</returns>
06021         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06022         uint GetAxisParameter_IN_POS(int axis);
06023
06024         /// <summary>
06025         /// Setter for IN_POS axis parameter
06026         /// </summary>
06027         /// <param name="axis">Target axis whose parameter is modified</param>
```

```
06028         /// <param name="value">New value to set to parameter</param>
06029         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06030         void SetAxisParameter_IN_POS(int axis, uint value);
06031
06032
06033         /// <summary>
06034         /// Getter for IN_POS_DIST axis parameter
06035         /// </summary>
06036         /// <param name="axis">Target axis whose parameter is retrieved</param>
06037         /// <returns>Current value of the parameter</returns>
06038         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06039         long GetAxisParameter_IN_POS_DIST(int axis);
06040
06041         /// <summary>
06042         /// Setter for IN_POS_DIST axis parameter
06043         /// </summary>
06044         /// <param name="axis">Target axis whose parameter is modified</param>
06045         /// <param name="value">New value to set to parameter</param>
06046         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06047         void SetAxisParameter_IN_POS_DIST(int axis, long value);
06048
06049
06050         /// <summary>
06051         /// Getter for IN_POS_SPEED axis parameter
06052         /// </summary>
06053         /// <param name="axis">Target axis whose parameter is retrieved</param>
06054         /// <returns>Current value of the parameter</returns>
06055         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06056         double GetAxisParameter_IN_POS_SPEED(int axis);
06057
06058         /// <summary>
06059         /// Setter for IN_POS_SPEED axis parameter
06060         /// </summary>
06061         /// <param name="axis">Target axis whose parameter is modified</param>
06062         /// <param name="value">New value to set to parameter</param>
06063         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06064         void SetAxisParameter_IN_POS_SPEED(int axis, double value);
06065
06066
06067         /// <summary>
06068         /// Getter for INTERP_FACTOR axis parameter
06069         /// </summary>
06070         /// <param name="axis">Target axis whose parameter is retrieved</param>
06071         /// <returns>Current value of the parameter</returns>
06072         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06073         double GetAxisParameter_INTERP_FACTOR(int axis);
06074
06075         /// <summary>
06076         /// Setter for INTERP_FACTOR axis parameter
06077         /// </summary>
06078         /// <param name="axis">Target axis whose parameter is modified</param>
06079         /// <param name="value">New value to set to parameter</param>
06080         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06081         void SetAxisParameter_INTERP_FACTOR(int axis, double value);
06082
06083
06084         /// <summary>
06085         /// Getter for INVERT_STEP axis parameter
06086         /// </summary>
06087         /// <param name="axis">Target axis whose parameter is retrieved</param>
06088         /// <returns>Current value of the parameter</returns>
06089         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06090         int GetAxisParameter_INVERT_STEP(int axis);
06091
06092         /// <summary>
06093         /// Setter for INVERT_STEP axis parameter
06094         /// </summary>
06095         /// <param name="axis">Target axis whose parameter is modified</param>
06096         /// <param name="value">New value to set to parameter</param>
06097         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06098         void SetAxisParameter_INVERT_STEP(int axis, int value);
06099
06100
06101         /// <summary>
06102         /// Getter for JERK axis parameter
06103         /// </summary>
06104         /// <param name="axis">Target axis whose parameter is retrieved</param>
06105         /// <returns>Current value of the parameter</returns>
```

```
06106         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06107         double GetAxisParameter_JERK(int axis);
06108
06109         /// <summary>
06110         /// Setter for JERK axis parameter
06111         /// </summary>
06112         /// <param name="axis">Target axis whose parameter is modified</param>
06113         /// <param name="value">New value to set to parameter</param>
06114         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06115         void SetAxisParameter_JERK(int axis, double value);
06116
06117
06118         /// <summary>
06119         /// Getter for JOGSPEED axis parameter
06120         /// </summary>
06121         /// <param name="axis">Target axis whose parameter is retrieved</param>
06122         /// <returns>Current value of the parameter</returns>
06123         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06124         double GetAxisParameter_JOGSPEED(int axis);
06125
06126         /// <summary>
06127         /// Setter for JOGSPEED axis parameter
06128         /// </summary>
06129         /// <param name="axis">Target axis whose parameter is modified</param>
06130         /// <param name="value">New value to set to parameter</param>
06131         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06132         void SetAxisParameter_JOGSPEED(int axis, double value);
06133
06134
06135         /// <summary>
06136         /// Getter for LINK_AXIS axis parameter
06137         /// </summary>
06138         /// <param name="axis">Target axis whose parameter is retrieved</param>
06139         /// <returns>Current value of the parameter</returns>
06140         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06141         short GetAxisParameter_LINK_AXIS(int axis);
06142
06143
06144         /// <summary>
06145         /// Getter for LOADED axis parameter
06146         /// </summary>
06147         /// <param name="axis">Target axis whose parameter is retrieved</param>
06148         /// <returns>Current value of the parameter</returns>
06149         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06150         uint GetAxisParameter_LOADED(int axis);
06151
06152
06153         /// <summary>
06154         /// Getter for LOOKAHEAD_FACTOR axis parameter
06155         /// </summary>
06156         /// <param name="axis">Target axis whose parameter is retrieved</param>
06157         /// <returns>Current value of the parameter</returns>
06158         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06159         double GetAxisParameter_LOOKAHEAD_FACTOR(int axis);
06160
06161         /// <summary>
06162         /// Setter for LOOKAHEAD_FACTOR axis parameter
06163         /// </summary>
06164         /// <param name="axis">Target axis whose parameter is modified</param>
06165         /// <param name="value">New value to set to parameter</param>
06166         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06167         void SetAxisParameter_LOOKAHEAD_FACTOR(int axis, double value);
06168
06169
06170         /// <summary>
06171         /// Getter for MARK axis parameter
06172         /// </summary>
06173         /// <param name="axis">Target axis whose parameter is retrieved</param>
06174         /// <returns>Current value of the parameter</returns>
06175         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06176         int GetAxisParameter_MARK(int axis);
06177
06178
06179         /// <summary>
06180         /// Getter for MARKB axis parameter
06181         /// </summary>
06182         /// <param name="axis">Target axis whose parameter is retrieved</param>
06183         /// <returns>Current value of the parameter</returns>
```

```
06184         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06185         int GetAxisParameter_MARKB(int axis);
06186
06187
06188         /// <summary>
06189         /// Getter for MERGE axis parameter
06190         /// </summary>
06191         /// <param name="axis">Target axis whose parameter is retrieved</param>
06192         /// <returns>Current value of the parameter</returns>
06193         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06194         bool GetAxisParameter_MERGE(int axis);
06195
06196         /// <summary>
06197         /// Setter for MERGE axis parameter
06198         /// </summary>
06199         /// <param name="axis">Target axis whose parameter is modified</param>
06200         /// <param name="value">New value to set to parameter</param>
06201         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06202         void SetAxisParameter_MERGE(int axis, bool value);
06203
06204
06205         /// <summary>
06206         /// Getter for MOVE_COUNT axis parameter
06207         /// </summary>
06208         /// <param name="axis">Target axis whose parameter is retrieved</param>
06209         /// <returns>Current value of the parameter</returns>
06210         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06211         int GetAxisParameter_MOVE_COUNT(int axis);
06212
06213         /// <summary>
06214         /// Setter for MOVE_COUNT axis parameter
06215         /// </summary>
06216         /// <param name="axis">Target axis whose parameter is modified</param>
06217         /// <param name="value">New value to set to parameter</param>
06218         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06219         void SetAxisParameter_MOVE_COUNT(int axis, int value);
06220
06221
06222         /// <summary>
06223         /// Getter for MOVE_PA axis parameter
06224         /// </summary>
06225         /// <param name="axis">Target axis whose parameter is retrieved</param>
06226         /// <returns>Current value of the parameter</returns>
06227         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06228         double GetAxisParameter_MOVE_PA(int axis);
06229
06230         /// <summary>
06231         /// Setter for MOVE_PA axis parameter
06232         /// </summary>
06233         /// <param name="axis">Target axis whose parameter is modified</param>
06234         /// <param name="value">New value to set to parameter</param>
06235         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06236         void SetAxisParameter_MOVE_PA(int axis, double value);
06237
06238
06239         /// <summary>
06240         /// Getter for MOVE_PA_CONT axis parameter
06241         /// </summary>
06242         /// <param name="axis">Target axis whose parameter is retrieved</param>
06243         /// <returns>Current value of the parameter</returns>
06244         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06245         int GetAxisParameter_MOVE_PA_CONT(int axis);
06246
06247         /// <summary>
06248         /// Setter for MOVE_PA_CONT axis parameter
06249         /// </summary>
06250         /// <param name="axis">Target axis whose parameter is modified</param>
06251         /// <param name="value">New value to set to parameter</param>
06252         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06253         void SetAxisParameter_MOVE_PA_CONT(int axis, int value);
06254
06255
06256         /// <summary>
06257         /// Getter for MOVE_PA_IDLE axis parameter
06258         /// </summary>
06259         /// <param name="axis">Target axis whose parameter is retrieved</param>
06260         /// <returns>Current value of the parameter</returns>
06261         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
```

```
      module.</exception>
06262         double GetAxisParameter_MOVE_PA_IDLE(int axis);
06263
06264         /// <summary>
06265         /// Setter for MOVE_PA_IDLE axis parameter
06266         /// </summary>
06267         /// <param name="axis">Target axis whose parameter is modified</param>
06268         /// <param name="value">New value to set to parameter</param>
06269         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06270         void SetAxisParameter_MOVE_PA_IDLE(int axis, double value);
06271
06272
06273         /// <summary>
06274         /// Getter for MOVE_PB axis parameter
06275         /// </summary>
06276         /// <param name="axis">Target axis whose parameter is retrieved</param>
06277         /// <returns>Current value of the parameter</returns>
06278         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06279         double GetAxisParameter_MOVE_PB(int axis);
06280
06281         /// <summary>
06282         /// Setter for MOVE_PB axis parameter
06283         /// </summary>
06284         /// <param name="axis">Target axis whose parameter is modified</param>
06285         /// <param name="value">New value to set to parameter</param>
06286         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06287         void SetAxisParameter_MOVE_PB(int axis, double value);
06288
06289
06290         /// <summary>
06291         /// Getter for MOVE_PB_CONT axis parameter
06292         /// </summary>
06293         /// <param name="axis">Target axis whose parameter is retrieved</param>
06294         /// <returns>Current value of the parameter</returns>
06295         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06296         int GetAxisParameter_MOVE_PB_CONT(int axis);
06297
06298         /// <summary>
06299         /// Setter for MOVE_PB_CONT axis parameter
06300         /// </summary>
06301         /// <param name="axis">Target axis whose parameter is modified</param>
06302         /// <param name="value">New value to set to parameter</param>
06303         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06304         void SetAxisParameter_MOVE_PB_CONT(int axis, int value);
06305
06306
06307         /// <summary>
06308         /// Getter for MOVE_PB_IDLE axis parameter
06309         /// </summary>
06310         /// <param name="axis">Target axis whose parameter is retrieved</param>
06311         /// <returns>Current value of the parameter</returns>
06312         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06313         double GetAxisParameter_MOVE_PB_IDLE(int axis);
06314
06315         /// <summary>
06316         /// Setter for MOVE_PB_IDLE axis parameter
06317         /// </summary>
06318         /// <param name="axis">Target axis whose parameter is modified</param>
06319         /// <param name="value">New value to set to parameter</param>
06320         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06321         void SetAxisParameter_MOVE_PB_IDLE(int axis, double value);
06322
06323
06324         /// <summary>
06325         /// Getter for MOVELINK_MODIFY axis parameter
06326         /// </summary>
06327         /// <param name="axis">Target axis whose parameter is retrieved</param>
06328         /// <returns>Current value of the parameter</returns>
06329         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06330         double GetAxisParameter_MOVELINK_MODIFY(int axis);
06331
06332         /// <summary>
06333         /// Setter for MOVELINK_MODIFY axis parameter
06334         /// </summary>
06335         /// <param name="axis">Target axis whose parameter is modified</param>
06336         /// <param name="value">New value to set to parameter</param>
06337         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06338         void SetAxisParameter_MOVELINK_MODIFY(int axis, double value);
```

```
06339
06340
06341         /// <summary>
06342         /// Getter for MOVES_BUFFERED axis parameter
06343         /// </summary>
06344         /// <param name="axis">Target axis whose parameter is retrieved</param>
06345         /// <returns>Current value of the parameter</returns>
06346         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06347         short GetAxisParameter_MOVES_BUFFERED(int axis);
06348
06349
06350         /// <summary>
06351         /// Getter for MPOS axis parameter
06352         /// </summary>
06353         /// <param name="axis">Target axis whose parameter is retrieved</param>
06354         /// <returns>Current value of the parameter</returns>
06355         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06356         double GetAxisParameter_MPOS(int axis);
06357
06358
06359         /// <summary>
06360         /// Getter for MSPEED axis parameter
06361         /// </summary>
06362         /// <param name="axis">Target axis whose parameter is retrieved</param>
06363         /// <returns>Current value of the parameter</returns>
06364         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06365         double GetAxisParameter_MSPEED(int axis);
06366
06367
06368         /// <summary>
06369         /// Getter for MSPEED_FILTER axis parameter
06370         /// </summary>
06371         /// <param name="axis">Target axis whose parameter is retrieved</param>
06372         /// <returns>Current value of the parameter</returns>
06373         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06374         double GetAxisParameter_MSPEED_FILTER(int axis);
06375
06376         /// <summary>
06377         /// Setter for MSPEED_FILTER axis parameter
06378         /// </summary>
06379         /// <param name="axis">Target axis whose parameter is modified</param>
06380         /// <param name="value">New value to set to parameter</param>
06381         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06382         void SetAxisParameter_MSPEED_FILTER(int axis, double value);
06383
06384
06385         /// <summary>
06386         /// Getter for MSPEEDF axis parameter
06387         /// </summary>
06388         /// <param name="axis">Target axis whose parameter is retrieved</param>
06389         /// <returns>Current value of the parameter</returns>
06390         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06391         double GetAxisParameter_MSPEEDF(int axis);
06392
06393
06394         /// <summary>
06395         /// Getter for MTYPE axis parameter
06396         /// </summary>
06397         /// <param name="axis">Target axis whose parameter is retrieved</param>
06398         /// <returns>Current value of the parameter</returns>
06399         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06400         MotionType GetAxisParameter_MTYPE(int axis);
06401
06402
06403         /// <summary>
06404         /// Getter for NTYPE axis parameter
06405         /// </summary>
06406         /// <param name="axis">Target axis whose parameter is retrieved</param>
06407         /// <returns>Current value of the parameter</returns>
06408         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06409         MotionType GetAxisParameter_NTYPE(int axis);
06410
06411
06412         /// <summary>
06413         /// Getter for OFFPOS axis parameter
06414         /// </summary>
06415         /// <param name="axis">Target axis whose parameter is retrieved</param>
06416         /// <returns>Current value of the parameter</returns>
06417         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
```

```
module.</exception>
06418         double GetAxisParameter_OFFPOS(int axis);
06419
06420         /// <summary>
06421         /// Setter for OFFPOS axis parameter
06422         /// </summary>
06423         /// <param name="axis">Target axis whose parameter is modified</param>
06424         /// <param name="value">New value to set to parameter</param>
06425         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
06426         void SetAxisParameter_OFFPOS(int axis, double value);
06427
06428
06429         /// <summary>
06430         /// Getter for OPEN_WIN axis parameter
06431         /// </summary>
06432         /// <param name="axis">Target axis whose parameter is retrieved</param>
06433         /// <returns>Current value of the parameter</returns>
06434         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
06435         double GetAxisParameter_OPEN_WIN(int axis);
06436
06437         /// <summary>
06438         /// Setter for OPEN_WIN axis parameter
06439         /// </summary>
06440         /// <param name="axis">Target axis whose parameter is modified</param>
06441         /// <param name="value">New value to set to parameter</param>
06442         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
06443         void SetAxisParameter_OPEN_WIN(int axis, double value);
06444
06445
06446         /// <summary>
06447         /// Getter for OUTLIMIT axis parameter
06448         /// </summary>
06449         /// <param name="axis">Target axis whose parameter is retrieved</param>
06450         /// <returns>Current value of the parameter</returns>
06451         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
06452         int GetAxisParameter_OUTLIMIT(int axis);
06453
06454         /// <summary>
06455         /// Setter for OUTLIMIT axis parameter
06456         /// </summary>
06457         /// <param name="axis">Target axis whose parameter is modified</param>
06458         /// <param name="value">New value to set to parameter</param>
06459         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
06460         void SetAxisParameter_OUTLIMIT(int axis, int value);
06461
06462
06463         /// <summary>
06464         /// Getter for OV_GAIN axis parameter
06465         /// </summary>
06466         /// <param name="axis">Target axis whose parameter is retrieved</param>
06467         /// <returns>Current value of the parameter</returns>
06468         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
06469         double GetAxisParameter_OV_GAIN(int axis);
06470
06471         /// <summary>
06472         /// Setter for OV_GAIN axis parameter
06473         /// </summary>
06474         /// <param name="axis">Target axis whose parameter is modified</param>
06475         /// <param name="value">New value to set to parameter</param>
06476         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
06477         void SetAxisParameter_OV_GAIN(int axis, double value);
06478
06479
06480         /// <summary>
06481         /// Getter for P_GAIN axis parameter
06482         /// </summary>
06483         /// <param name="axis">Target axis whose parameter is retrieved</param>
06484         /// <returns>Current value of the parameter</returns>
06485         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
06486         double GetAxisParameter_P_GAIN(int axis);
06487
06488         /// <summary>
06489         /// Setter for P_GAIN axis parameter
06490         /// </summary>
06491         /// <param name="axis">Target axis whose parameter is modified</param>
06492         /// <param name="value">New value to set to parameter</param>
06493         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
06494         void SetAxisParameter_P_GAIN(int axis, double value);
```

```
06495
06496
06497         /// <summary>
06498         /// Getter for POS_DELAY axis parameter
06499         /// </summary>
06500         /// <param name="axis">Target axis whose parameter is retrieved</param>
06501         /// <returns>Current value of the parameter</returns>
06502         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06503         double GetAxisParameter_POS_DELAY(int axis);
06504
06505         /// <summary>
06506         /// Setter for POS_DELAY axis parameter
06507         /// </summary>
06508         /// <param name="axis">Target axis whose parameter is modified</param>
06509         /// <param name="value">New value to set to parameter</param>
06510         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06511         void SetAxisParameter_POS_DELAY(int axis, double value);
06512
06513
06514         /// <summary>
06515         /// Getter for PP_STEP axis parameter
06516         /// </summary>
06517         /// <param name="axis">Target axis whose parameter is retrieved</param>
06518         /// <returns>Current value of the parameter</returns>
06519         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06520         int GetAxisParameter_PP_STEP(int axis);
06521
06522         /// <summary>
06523         /// Setter for PP_STEP axis parameter
06524         /// </summary>
06525         /// <param name="axis">Target axis whose parameter is modified</param>
06526         /// <param name="value">New value to set to parameter</param>
06527         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06528         void SetAxisParameter_PP_STEP(int axis, int value);
06529
06530
06531         /// <summary>
06532         /// Getter for PS_ENCODER axis parameter
06533         /// </summary>
06534         /// <param name="axis">Target axis whose parameter is retrieved</param>
06535         /// <returns>Current value of the parameter</returns>
06536         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06537         int GetAxisParameter_PS_ENCODER(int axis);
06538
06539
06540         /// <summary>
06541         /// Getter for RAISE_ANGLE axis parameter
06542         /// </summary>
06543         /// <param name="axis">Target axis whose parameter is retrieved</param>
06544         /// <returns>Current value of the parameter</returns>
06545         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06546         double GetAxisParameter_RAISE_ANGLE(int axis);
06547
06548         /// <summary>
06549         /// Setter for RAISE_ANGLE axis parameter
06550         /// </summary>
06551         /// <param name="axis">Target axis whose parameter is modified</param>
06552         /// <param name="value">New value to set to parameter</param>
06553         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06554         void SetAxisParameter_RAISE_ANGLE(int axis, double value);
06555
06556
06557         /// <summary>
06558         /// Getter for REG_POS axis parameter
06559         /// </summary>
06560         /// <param name="axis">Target axis whose parameter is retrieved</param>
06561         /// <returns>Current value of the parameter</returns>
06562         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06563         double GetAxisParameter_REG_POS(int axis);
06564
06565
06566         /// <summary>
06567         /// Getter for REG_POSB axis parameter
06568         /// </summary>
06569         /// <param name="axis">Target axis whose parameter is retrieved</param>
06570         /// <returns>Current value of the parameter</returns>
06571         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06572         double GetAxisParameter_REG_POSB(int axis);
```

```
06573
06574
06575         /// <summary>
06576         /// Getter for REMAIN axis parameter
06577         /// </summary>
06578         /// <param name="axis">Target axis whose parameter is retrieved</param>
06579         /// <returns>Current value of the parameter</returns>
06580         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06581         double GetAxisParameter_REMAIN(int axis);
06582
06583
06584         /// <summary>
06585         /// Getter for REP_DIST axis parameter
06586         /// </summary>
06587         /// <param name="axis">Target axis whose parameter is retrieved</param>
06588         /// <returns>Current value of the parameter</returns>
06589         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06590         double GetAxisParameter_REP_DIST(int axis);
06591
06592         /// <summary>
06593         /// Setter for REP_DIST axis parameter
06594         /// </summary>
06595         /// <param name="axis">Target axis whose parameter is modified</param>
06596         /// <param name="value">New value to set to parameter</param>
06597         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06598         void SetAxisParameter_REP_DIST(int axis, double value);
06599
06600
06601         /// <summary>
06602         /// Getter for REP_OPTION axis parameter
06603         /// </summary>
06604         /// <param name="axis">Target axis whose parameter is retrieved</param>
06605         /// <returns>Current value of the parameter</returns>
06606         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06607         short GetAxisParameter_REP_OPTION(int axis);
06608
06609         /// <summary>
06610         /// Setter for REP_OPTION axis parameter
06611         /// </summary>
06612         /// <param name="axis">Target axis whose parameter is modified</param>
06613         /// <param name="value">New value to set to parameter</param>
06614         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06615         void SetAxisParameter_REP_OPTION(int axis, short value);
06616
06617
06618         /// <summary>
06619         /// Getter for REV_IN axis parameter
06620         /// </summary>
06621         /// <param name="axis">Target axis whose parameter is retrieved</param>
06622         /// <returns>Current value of the parameter</returns>
06623         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06624         short GetAxisParameter_REV_IN(int axis);
06625
06626         /// <summary>
06627         /// Setter for REV_IN axis parameter
06628         /// </summary>
06629         /// <param name="axis">Target axis whose parameter is modified</param>
06630         /// <param name="value">New value to set to parameter</param>
06631         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06632         void SetAxisParameter_REV_IN(int axis, short value);
06633
06634
06635         /// <summary>
06636         /// Getter for REV_JOG axis parameter
06637         /// </summary>
06638         /// <param name="axis">Target axis whose parameter is retrieved</param>
06639         /// <returns>Current value of the parameter</returns>
06640         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06641         short GetAxisParameter_REV_JOG(int axis);
06642
06643         /// <summary>
06644         /// Setter for REV_JOG axis parameter
06645         /// </summary>
06646         /// <param name="axis">Target axis whose parameter is modified</param>
06647         /// <param name="value">New value to set to parameter</param>
06648         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06649         void SetAxisParameter_REV_JOG(int axis, short value);
06650
```

```
06651
06652         /// <summary>
06653         /// Getter for ROBOT_DPOS axis parameter
06654         /// </summary>
06655         /// <param name="axis">Target axis whose parameter is retrieved</param>
06656         /// <returns>Current value of the parameter</returns>
06657         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06658         double GetAxisParameter_ROBOT_DPOS(int axis);
06659
06660
06661         /// <summary>
06662         /// Getter for ROBOT_FS_LIMIT axis parameter
06663         /// </summary>
06664         /// <param name="axis">Target axis whose parameter is retrieved</param>
06665         /// <returns>Current value of the parameter</returns>
06666         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06667         double GetAxisParameter_ROBOT_FS_LIMIT(int axis);
06668
06669         /// <summary>
06670         /// Setter for ROBOT_FS_LIMIT axis parameter
06671         /// </summary>
06672         /// <param name="axis">Target axis whose parameter is modified</param>
06673         /// <param name="value">New value to set to parameter</param>
06674         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06675         void SetAxisParameter_ROBOT_FS_LIMIT(int axis, double value);
06676
06677
06678         /// <summary>
06679         /// Getter for ROBOT_RS_LIMIT axis parameter
06680         /// </summary>
06681         /// <param name="axis">Target axis whose parameter is retrieved</param>
06682         /// <returns>Current value of the parameter</returns>
06683         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06684         double GetAxisParameter_ROBOT_RS_LIMIT(int axis);
06685
06686         /// <summary>
06687         /// Setter for ROBOT_RS_LIMIT axis parameter
06688         /// </summary>
06689         /// <param name="axis">Target axis whose parameter is modified</param>
06690         /// <param name="value">New value to set to parameter</param>
06691         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06692         void SetAxisParameter_ROBOT_RS_LIMIT(int axis, double value);
06693
06694
06695         /// <summary>
06696         /// Getter for ROBOT_SP_MODE axis parameter
06697         /// </summary>
06698         /// <param name="axis">Target axis whose parameter is retrieved</param>
06699         /// <returns>Current value of the parameter</returns>
06700         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06701         short GetAxisParameter_ROBOT_SP_MODE(int axis);
06702
06703         /// <summary>
06704         /// Setter for ROBOT_SP_MODE axis parameter
06705         /// </summary>
06706         /// <param name="axis">Target axis whose parameter is modified</param>
06707         /// <param name="value">New value to set to parameter</param>
06708         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06709         void SetAxisParameter_ROBOT_SP_MODE(int axis, short value);
06710
06711
06712         /// <summary>
06713         /// Getter for ROBOTSTATUS axis parameter
06714         /// </summary>
06715         /// <param name="axis">Target axis whose parameter is retrieved</param>
06716         /// <returns>Current value of the parameter</returns>
06717         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06718         RobotStatus GetAxisParameter_ROBOTSTATUS(int axis);
06719
06720
06721         /// <summary>
06722         /// Getter for ROBOT_UNITS axis parameter
06723         /// </summary>
06724         /// <param name="axis">Target axis whose parameter is retrieved</param>
06725         /// <returns>Current value of the parameter</returns>
06726         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06727         double GetAxisParameter_ROBOT_UNITS(int axis);
06728
```

```
06729         /// <summary>
06730         /// Setter for ROBOT_UNITS axis parameter
06731         /// </summary>
06732         /// <param name="axis">Target axis whose parameter is modified</param>
06733         /// <param name="value">New value to set to parameter</param>
06734         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06735         void SetAxisParameter_ROBOT_UNITS(int axis, double value);
06736
06737
06738         /// <summary>
06739         /// Getter for RS_LIMIT axis parameter
06740         /// </summary>
06741         /// <param name="axis">Target axis whose parameter is retrieved</param>
06742         /// <returns>Current value of the parameter</returns>
06743         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06744         double GetAxisParameter_RS_LIMIT(int axis);
06745
06746         /// <summary>
06747         /// Setter for RS_LIMIT axis parameter
06748         /// </summary>
06749         /// <param name="axis">Target axis whose parameter is modified</param>
06750         /// <param name="value">New value to set to parameter</param>
06751         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06752         void SetAxisParameter_RS_LIMIT(int axis, double value);
06753
06754
06755         /// <summary>
06756         /// Getter for SERVO axis parameter
06757         /// </summary>
06758         /// <param name="axis">Target axis whose parameter is retrieved</param>
06759         /// <returns>Current value of the parameter</returns>
06760         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06761         bool GetAxisParameter_SERVO(int axis);
06762
06763         /// <summary>
06764         /// Setter for SERVO axis parameter
06765         /// </summary>
06766         /// <param name="axis">Target axis whose parameter is modified</param>
06767         /// <param name="value">New value to set to parameter</param>
06768         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06769         void SetAxisParameter_SERVO(int axis, bool value);
06770
06771
06772         /// <summary>
06773         /// Getter for SLOT_NUMBER axis parameter
06774         /// </summary>
06775         /// <param name="axis">Target axis whose parameter is retrieved</param>
06776         /// <returns>Current value of the parameter</returns>
06777         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06778         short GetAxisParameter_SLOT_NUMBER(int axis);
06779
06780         /// <summary>
06781         /// Setter for SLOT_NUMBER axis parameter
06782         /// </summary>
06783         /// <param name="axis">Target axis whose parameter is modified</param>
06784         /// <param name="value">New value to set to parameter</param>
06785         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06786         void SetAxisParameter_SLOT_NUMBER(int axis, short value);
06787
06788
06789         /// <summary>
06790         /// Getter for SPEED axis parameter
06791         /// </summary>
06792         /// <param name="axis">Target axis whose parameter is retrieved</param>
06793         /// <returns>Current value of the parameter</returns>
06794         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06795         double GetAxisParameter_SPEED(int axis);
06796
06797         /// <summary>
06798         /// Setter for SPEED axis parameter
06799         /// </summary>
06800         /// <param name="axis">Target axis whose parameter is modified</param>
06801         /// <param name="value">New value to set to parameter</param>
06802         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06803         void SetAxisParameter_SPEED(int axis, double value);
06804
06805
06806         /// <summary>
```

```
06807         /// Getter for SRAMP axis parameter
06808         /// </summary>
06809         /// <param name="axis">Target axis whose parameter is retrieved</param>
06810         /// <returns>Current value of the parameter</returns>
06811         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06812         int GetAxisParameter_SRAMP(int axis);
06813
06814         /// <summary>
06815         /// Setter for SRAMP axis parameter
06816         /// </summary>
06817         /// <param name="axis">Target axis whose parameter is modified</param>
06818         /// <param name="value">New value to set to parameter</param>
06819         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06820         void SetAxisParameter_SRAMP(int axis, int value);
06821
06822
06823         /// <summary>
06824         /// Getter for S_REF axis parameter
06825         /// </summary>
06826         /// <param name="axis">Target axis whose parameter is retrieved</param>
06827         /// <returns>Current value of the parameter</returns>
06828         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06829         int GetAxisParameter_S_REF(int axis);
06830
06831         /// <summary>
06832         /// Setter for S_REF axis parameter
06833         /// </summary>
06834         /// <param name="axis">Target axis whose parameter is modified</param>
06835         /// <param name="value">New value to set to parameter</param>
06836         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06837         void SetAxisParameter_S_REF(int axis, int value);
06838
06839
06840         /// <summary>
06841         /// Getter for S_REF_OUT axis parameter
06842         /// </summary>
06843         /// <param name="axis">Target axis whose parameter is retrieved</param>
06844         /// <returns>Current value of the parameter</returns>
06845         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06846         int GetAxisParameter_S_REF_OUT(int axis);
06847
06848         /// <summary>
06849         /// Setter for S_REF_OUT axis parameter
06850         /// </summary>
06851         /// <param name="axis">Target axis whose parameter is modified</param>
06852         /// <param name="value">New value to set to parameter</param>
06853         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06854         void SetAxisParameter_S_REF_OUT(int axis, int value);
06855
06856
06857         /// <summary>
06858         /// Getter for START_DIR_LAST axis parameter
06859         /// </summary>
06860         /// <param name="axis">Target axis whose parameter is retrieved</param>
06861         /// <returns>Current value of the parameter</returns>
06862         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06863         double GetAxisParameter_START_DIR_LAST(int axis);
06864
06865         /// <summary>
06866         /// Setter for START_DIR_LAST axis parameter
06867         /// </summary>
06868         /// <param name="axis">Target axis whose parameter is modified</param>
06869         /// <param name="value">New value to set to parameter</param>
06870         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06871         void SetAxisParameter_START_DIR_LAST(int axis, double value);
06872
06873
06874         /// <summary>
06875         /// Getter for STARTMOVE_SPEED axis parameter
06876         /// </summary>
06877         /// <param name="axis">Target axis whose parameter is retrieved</param>
06878         /// <returns>Current value of the parameter</returns>
06879         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06880         double GetAxisParameter_STARTMOVE_SPEED(int axis);
06881
06882         /// <summary>
06883         /// Setter for STARTMOVE_SPEED axis parameter
06884         /// </summary>
```

```
06885         /// <param name="axis">Target axis whose parameter is modified</param>
06886         /// <param name="value">New value to set to parameter</param>
06887         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06888         void SetAxisParameter_STARTMOVE_SPEED(int axis, double value);
06889
06890
06891         /// <summary>
06892         /// Getter for STOP_ANGLE axis parameter
06893         /// </summary>
06894         /// <param name="axis">Target axis whose parameter is retrieved</param>
06895         /// <returns>Current value of the parameter</returns>
06896         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06897         double GetAxisParameter_STOP_ANGLE(int axis);
06898
06899         /// <summary>
06900         /// Setter for STOP_ANGLE axis parameter
06901         /// </summary>
06902         /// <param name="axis">Target axis whose parameter is modified</param>
06903         /// <param name="value">New value to set to parameter</param>
06904         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06905         void SetAxisParameter_STOP_ANGLE(int axis, double value);
06906
06907
06908         /// <summary>
06909         /// Getter for SYNC_AXIS axis parameter
06910         /// </summary>
06911         /// <param name="axis">Target axis whose parameter is retrieved</param>
06912         /// <returns>Current value of the parameter</returns>
06913         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06914         int GetAxisParameter_SYNC_AXIS(int axis);
06915
06916         /// <summary>
06917         /// Setter for SYNC_AXIS axis parameter
06918         /// </summary>
06919         /// <param name="axis">Target axis whose parameter is modified</param>
06920         /// <param name="value">New value to set to parameter</param>
06921         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06922         void SetAxisParameter_SYNC_AXIS(int axis, int value);
06923
06924
06925         /// <summary>
06926         /// Getter for SYNC_CONTROL axis parameter
06927         /// </summary>
06928         /// <param name="axis">Target axis whose parameter is retrieved</param>
06929         /// <returns>Current value of the parameter</returns>
06930         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06931         SyncControl GetAxisParameter_SYNC_CONTROL(int axis);
06932
06933         /// <summary>
06934         /// Setter for SYNC_CONTROL axis parameter
06935         /// </summary>
06936         /// <param name="axis">Target axis whose parameter is modified</param>
06937         /// <param name="value">New value to set to parameter</param>
06938         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06939         void SetAxisParameter_SYNC_CONTROL(int axis, SyncControl value);
06940
06941
06942         /// <summary>
06943         /// Getter for SYNC_DPOS axis parameter
06944         /// </summary>
06945         /// <param name="axis">Target axis whose parameter is retrieved</param>
06946         /// <returns>Current value of the parameter</returns>
06947         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06948         double GetAxisParameter_SYNC_DPOS(int axis);
06949
06950         /// <summary>
06951         /// Setter for SYNC_DPOS axis parameter
06952         /// </summary>
06953         /// <param name="axis">Target axis whose parameter is modified</param>
06954         /// <param name="value">New value to set to parameter</param>
06955         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06956         void SetAxisParameter_SYNC_DPOS(int axis, double value);
06957
06958
06959         /// <summary>
06960         /// Getter for SYNC_DWELL axis parameter
06961         /// </summary>
06962         /// <param name="axis">Target axis whose parameter is retrieved</param>
```

```
06963         /// <returns>Current value of the parameter</returns>
06964         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06965         int GetAxisParameter_SYNC_DWELL(int axis);
06966
06967         /// <summary>
06968         /// Setter for SYNC_DWELL axis parameter
06969         /// </summary>
06970         /// <param name="axis">Target axis whose parameter is modified</param>
06971         /// <param name="value">New value to set to parameter</param>
06972         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06973         void SetAxisParameter_SYNC_DWELL(int axis, int value);
06974
06975
06976         /// <summary>
06977         /// Getter for SYNC_FLIGHT axis parameter
06978         /// </summary>
06979         /// <param name="axis">Target axis whose parameter is retrieved</param>
06980         /// <returns>Current value of the parameter</returns>
06981         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06982         int GetAxisParameter_SYNC_FLIGHT(int axis);
06983
06984         /// <summary>
06985         /// Setter for SYNC_FLIGHT axis parameter
06986         /// </summary>
06987         /// <param name="axis">Target axis whose parameter is modified</param>
06988         /// <param name="value">New value to set to parameter</param>
06989         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06990         void SetAxisParameter_SYNC_FLIGHT(int axis, int value);
06991
06992
06993         /// <summary>
06994         /// Getter for SYNC_PAUSE axis parameter
06995         /// </summary>
06996         /// <param name="axis">Target axis whose parameter is retrieved</param>
06997         /// <returns>Current value of the parameter</returns>
06998         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
06999         int GetAxisParameter_SYNC_PAUSE(int axis);
07000
07001         /// <summary>
07002         /// Setter for SYNC_PAUSE axis parameter
07003         /// </summary>
07004         /// <param name="axis">Target axis whose parameter is modified</param>
07005         /// <param name="value">New value to set to parameter</param>
07006         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07007         void SetAxisParameter_SYNC_PAUSE(int axis, int value);
07008
07009
07010         /// <summary>
07011         /// Getter for SYNC_TIME axis parameter
07012         /// </summary>
07013         /// <param name="axis">Target axis whose parameter is retrieved</param>
07014         /// <returns>Current value of the parameter</returns>
07015         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07016         int GetAxisParameter_SYNC_TIME(int axis);
07017
07018         /// <summary>
07019         /// Setter for SYNC_TIME axis parameter
07020         /// </summary>
07021         /// <param name="axis">Target axis whose parameter is modified</param>
07022         /// <param name="value">New value to set to parameter</param>
07023         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07024         void SetAxisParameter_SYNC_TIME(int axis, int value);
07025
07026
07027         /// <summary>
07028         /// Getter for SYNC_TIMER axis parameter
07029         /// </summary>
07030         /// <param name="axis">Target axis whose parameter is retrieved</param>
07031         /// <returns>Current value of the parameter</returns>
07032         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07033         int GetAxisParameter_SYNC_TIMER(int axis);
07034
07035
07036         /// <summary>
07037         /// Getter for SYNC_WITHDRAW axis parameter
07038         /// </summary>
07039         /// <param name="axis">Target axis whose parameter is retrieved</param>
07040         /// <returns>Current value of the parameter</returns>
```

```
07041         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07042         double GetAxisParameter_SYNC_WITHDRAW(int axis);
07043
07044         /// <summary>
07045         /// Setter for SYNC_WITHDRAW axis parameter
07046         /// </summary>
07047         /// <param name="axis">Target axis whose parameter is modified</param>
07048         /// <param name="value">New value to set to parameter</param>
07049         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07050         void SetAxisParameter_SYNC_WITHDRAW(int axis, double value);
07051
07052
07053         /// <summary>
07054         /// Getter for TABLE_POINTER axis parameter
07055         /// </summary>
07056         /// <param name="axis">Target axis whose parameter is retrieved</param>
07057         /// <returns>Current value of the parameter</returns>
07058         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07059         double GetAxisParameter_TABLE_POINTER(int axis);
07060
07061         /// <summary>
07062         /// Setter for TABLE_POINTER axis parameter
07063         /// </summary>
07064         /// <param name="axis">Target axis whose parameter is modified</param>
07065         /// <param name="value">New value to set to parameter</param>
07066         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07067         void SetAxisParameter_TABLE_POINTER(int axis, double value);
07068
07069
07070         /// <summary>
07071         /// Getter for TANG_DIRECTION axis parameter
07072         /// </summary>
07073         /// <param name="axis">Target axis whose parameter is retrieved</param>
07074         /// <returns>Current value of the parameter</returns>
07075         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07076         double GetAxisParameter_TANG_DIRECTION(int axis);
07077
07078         /// <summary>
07079         /// Setter for TANG_DIRECTION axis parameter
07080         /// </summary>
07081         /// <param name="axis">Target axis whose parameter is modified</param>
07082         /// <param name="value">New value to set to parameter</param>
07083         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07084         void SetAxisParameter_TANG_DIRECTION(int axis, double value);
07085
07086
07087         /// <summary>
07088         /// Getter for T_REF axis parameter
07089         /// </summary>
07090         /// <param name="axis">Target axis whose parameter is retrieved</param>
07091         /// <returns>Current value of the parameter</returns>
07092         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07093         int GetAxisParameter_T_REF(int axis);
07094
07095         /// <summary>
07096         /// Setter for T_REF axis parameter
07097         /// </summary>
07098         /// <param name="axis">Target axis whose parameter is modified</param>
07099         /// <param name="value">New value to set to parameter</param>
07100         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07101         void SetAxisParameter_T_REF(int axis, int value);
07102
07103
07104         /// <summary>
07105         /// Getter for T_REF_OUT axis parameter
07106         /// </summary>
07107         /// <param name="axis">Target axis whose parameter is retrieved</param>
07108         /// <returns>Current value of the parameter</returns>
07109         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07110         int GetAxisParameter_T_REF_OUT(int axis);
07111
07112         /// <summary>
07113         /// Setter for T_REF_OUT axis parameter
07114         /// </summary>
07115         /// <param name="axis">Target axis whose parameter is modified</param>
07116         /// <param name="value">New value to set to parameter</param>
07117         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
```

```
07118         void SetAxisParameter_T_REF_OUT(int axis, int value);
07119
07120
07121         /// <summary>
07122         /// Getter for UNITS axis parameter
07123         /// </summary>
07124         /// <param name="axis">Target axis whose parameter is retrieved</param>
07125         /// <returns>Current value of the parameter</returns>
07126         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07127         double GetAxisParameter_UNITS(int axis);
07128
07129         /// <summary>
07130         /// Setter for UNITS axis parameter
07131         /// </summary>
07132         /// <param name="axis">Target axis whose parameter is modified</param>
07133         /// <param name="value">New value to set to parameter</param>
07134         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07135         void SetAxisParameter_UNITS(int axis, double value);
07136
07137
07138         /// <summary>
07139         /// Getter for VECTOR_BUFFERED axis parameter
07140         /// </summary>
07141         /// <param name="axis">Target axis whose parameter is retrieved</param>
07142         /// <returns>Current value of the parameter</returns>
07143         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07144         double GetAxisParameter_VECTOR_BUFFERED(int axis);
07145
07146         /// <summary>
07147         /// Setter for VECTOR_BUFFERED axis parameter
07148         /// </summary>
07149         /// <param name="axis">Target axis whose parameter is modified</param>
07150         /// <param name="value">New value to set to parameter</param>
07151         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07152         void SetAxisParameter_VECTOR_BUFFERED(int axis, double value);
07153
07154
07155         /// <summary>
07156         /// Getter for VERIFY axis parameter
07157         /// </summary>
07158         /// <param name="axis">Target axis whose parameter is retrieved</param>
07159         /// <returns>Current value of the parameter</returns>
07160         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07161         short GetAxisParameter_VERIFY(int axis);
07162
07163         /// <summary>
07164         /// Setter for VERIFY axis parameter
07165         /// </summary>
07166         /// <param name="axis">Target axis whose parameter is modified</param>
07167         /// <param name="value">New value to set to parameter</param>
07168         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07169         void SetAxisParameter_VERIFY(int axis, short value);
07170
07171
07172         /// <summary>
07173         /// Getter for VFF_GAIN axis parameter
07174         /// </summary>
07175         /// <param name="axis">Target axis whose parameter is retrieved</param>
07176         /// <returns>Current value of the parameter</returns>
07177         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07178         double GetAxisParameter_VFF_GAIN(int axis);
07179
07180         /// <summary>
07181         /// Setter for VFF_GAIN axis parameter
07182         /// </summary>
07183         /// <param name="axis">Target axis whose parameter is modified</param>
07184         /// <param name="value">New value to set to parameter</param>
07185         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07186         void SetAxisParameter_VFF_GAIN(int axis, double value);
07187
07188
07189         /// <summary>
07190         /// Getter for VP_ACCEL axis parameter
07191         /// </summary>
07192         /// <param name="axis">Target axis whose parameter is retrieved</param>
07193         /// <returns>Current value of the parameter</returns>
07194         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07195         double GetAxisParameter_VP_ACCEL(int axis);
```

```
07196
07197
07198         /// <summary>
07199         /// Getter for VP_JERK axis parameter
07200         /// </summary>
07201         /// <param name="axis">Target axis whose parameter is retrieved</param>
07202         /// <returns>Current value of the parameter</returns>
07203         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07204         double GetAxisParameter_VP_JERK(int axis);
07205
07206
07207         /// <summary>
07208         /// Getter for VP_MODE axis parameter
07209         /// </summary>
07210         /// <param name="axis">Target axis whose parameter is retrieved</param>
07211         /// <returns>Current value of the parameter</returns>
07212         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07213         int GetAxisParameter_VP_MODE(int axis);
07214
07215         /// <summary>
07216         /// Setter for VP_MODE axis parameter
07217         /// </summary>
07218         /// <param name="axis">Target axis whose parameter is modified</param>
07219         /// <param name="value">New value to set to parameter</param>
07220         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07221         void SetAxisParameter_VP_MODE(int axis, int value);
07222
07223
07224         /// <summary>
07225         /// Getter for VP_POSITION axis parameter
07226         /// </summary>
07227         /// <param name="axis">Target axis whose parameter is retrieved</param>
07228         /// <returns>Current value of the parameter</returns>
07229         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07230         double GetAxisParameter_VP_POSITION(int axis);
07231
07232
07233         /// <summary>
07234         /// Getter for VP_SPEED axis parameter
07235         /// </summary>
07236         /// <param name="axis">Target axis whose parameter is retrieved</param>
07237         /// <returns>Current value of the parameter</returns>
07238         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07239         double GetAxisParameter_VP_SPEED(int axis);
07240
07241
07242         /// <summary>
07243         /// Getter for WORLD_ACCEL axis parameter
07244         /// </summary>
07245         /// <param name="axis">Target axis whose parameter is retrieved</param>
07246         /// <returns>Current value of the parameter</returns>
07247         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07248         double GetAxisParameter_WORLD_ACCEL(int axis);
07249
07250         /// <summary>
07251         /// Setter for WORLD_ACCEL axis parameter
07252         /// </summary>
07253         /// <param name="axis">Target axis whose parameter is modified</param>
07254         /// <param name="value">New value to set to parameter</param>
07255         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07256         void SetAxisParameter_WORLD_ACCEL(int axis, double value);
07257
07258
07259         /// <summary>
07260         /// Getter for WORLD_DECEL axis parameter
07261         /// </summary>
07262         /// <param name="axis">Target axis whose parameter is retrieved</param>
07263         /// <returns>Current value of the parameter</returns>
07264         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07265         double GetAxisParameter_WORLD_DECEL(int axis);
07266
07267         /// <summary>
07268         /// Setter for WORLD_DECEL axis parameter
07269         /// </summary>
07270         /// <param name="axis">Target axis whose parameter is modified</param>
07271         /// <param name="value">New value to set to parameter</param>
07272         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07273         void SetAxisParameter_WORLD_DECEL(int axis, double value);
```

```
07274
07275
07276         /// <summary>
07277         /// Getter for WORLD_DPOS axis parameter
07278         /// </summary>
07279         /// <param name="axis">Target axis whose parameter is retrieved</param>
07280         /// <returns>Current value of the parameter</returns>
07281         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07282         double GetAxisParameter_WORLD_DPOS(int axis);
07283
07284
07285         /// <summary>
07286         /// Getter for WORLD_FS_LIMIT axis parameter
07287         /// </summary>
07288         /// <param name="axis">Target axis whose parameter is retrieved</param>
07289         /// <returns>Current value of the parameter</returns>
07290         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07291         double GetAxisParameter_WORLD_FS_LIMIT(int axis);
07292
07293         /// <summary>
07294         /// Setter for WORLD_FS_LIMIT axis parameter
07295         /// </summary>
07296         /// <param name="axis">Target axis whose parameter is modified</param>
07297         /// <param name="value">New value to set to parameter</param>
07298         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07299         void SetAxisParameter_WORLD_FS_LIMIT(int axis, double value);
07300
07301
07302         /// <summary>
07303         /// Getter for WORLD_JOGSPEED axis parameter
07304         /// </summary>
07305         /// <param name="axis">Target axis whose parameter is retrieved</param>
07306         /// <returns>Current value of the parameter</returns>
07307         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07308         double GetAxisParameter_WORLD_JOGSPEED(int axis);
07309
07310         /// <summary>
07311         /// Setter for WORLD_JOGSPEED axis parameter
07312         /// </summary>
07313         /// <param name="axis">Target axis whose parameter is modified</param>
07314         /// <param name="value">New value to set to parameter</param>
07315         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07316         void SetAxisParameter_WORLD_JOGSPEED(int axis, double value);
07317
07318
07319         /// <summary>
07320         /// Getter for WORLD_RS_LIMIT axis parameter
07321         /// </summary>
07322         /// <param name="axis">Target axis whose parameter is retrieved</param>
07323         /// <returns>Current value of the parameter</returns>
07324         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07325         double GetAxisParameter_WORLD_RS_LIMIT(int axis);
07326
07327         /// <summary>
07328         /// Setter for WORLD_RS_LIMIT axis parameter
07329         /// </summary>
07330         /// <param name="axis">Target axis whose parameter is modified</param>
07331         /// <param name="value">New value to set to parameter</param>
07332         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07333         void SetAxisParameter_WORLD_RS_LIMIT(int axis, double value);
07334
07335
07336         /// <summary>
07337         /// Getter for WORLD_SPEED axis parameter
07338         /// </summary>
07339         /// <param name="axis">Target axis whose parameter is retrieved</param>
07340         /// <returns>Current value of the parameter</returns>
07341         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07342         double GetAxisParameter_WORLD_SPEED(int axis);
07343
07344         /// <summary>
07345         /// Setter for WORLD_SPEED axis parameter
07346         /// </summary>
07347         /// <param name="axis">Target axis whose parameter is modified</param>
07348         /// <param name="value">New value to set to parameter</param>
07349         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07350         void SetAxisParameter_WORLD_SPEED(int axis, double value);
07351
```

```
07352
07353         /// <summary>
07354         /// Getter for WORLD_UNITS axis parameter
07355         /// </summary>
07356         /// <param name="axis">Target axis whose parameter is retrieved</param>
07357         /// <returns>Current value of the parameter</returns>
07358         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07359         double GetAxisParameter_WORLD_UNITS(int axis);
07360
07361         /// <summary>
07362         /// Setter for WORLD_UNITS axis parameter
07363         /// </summary>
07364         /// <param name="axis">Target axis whose parameter is modified</param>
07365         /// <param name="value">New value to set to parameter</param>
07366         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07367         void SetAxisParameter_WORLD_UNITS(int axis, double value);
07368
07369
07370         /// <summary>
07371         /// Getter for AXIS_ENABLE_OVERRIDE axis parameter
07372         /// </summary>
07373         /// <param name="axis">Target axis whose parameter is retrieved</param>
07374         /// <returns>Current value of the parameter</returns>
07375         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07376         bool GetAxisParameter_AXIS_ENABLE_OVERRIDE(int axis);
07377
07378         /// <summary>
07379         /// Setter for AXIS_ENABLE_OVERRIDE axis parameter
07380         /// </summary>
07381         /// <param name="axis">Target axis whose parameter is modified</param>
07382         /// <param name="value">New value to set to parameter</param>
07383         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07384         void SetAxisParameter_AXIS_ENABLE_OVERRIDE(int axis, bool value);
07385
07386
07387         /// <summary>
07388         /// Getter for AXIS_DISPLAY axis parameter
07389         /// </summary>
07390         /// <param name="axis">Target axis whose parameter is retrieved</param>
07391         /// <returns>Current value of the parameter</returns>
07392         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07393         short GetAxisParameter_AXIS_DISPLAY(int axis);
07394
07395         /// <summary>
07396         /// Setter for AXIS_DISPLAY axis parameter
07397         /// </summary>
07398         /// <param name="axis">Target axis whose parameter is modified</param>
07399         /// <param name="value">New value to set to parameter</param>
07400         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07401         void SetAxisParameter_AXIS_DISPLAY(int axis, short value);
07402
07403
07404         /// <summary>
07405         /// Getter for AXIS_Z_OUTPUT axis parameter
07406         /// </summary>
07407         /// <param name="axis">Target axis whose parameter is retrieved</param>
07408         /// <returns>Current value of the parameter</returns>
07409         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07410         int GetAxisParameter_AXIS_Z_OUTPUT(int axis);
07411
07412         /// <summary>
07413         /// Setter for AXIS_Z_OUTPUT axis parameter
07414         /// </summary>
07415         /// <param name="axis">Target axis whose parameter is modified</param>
07416         /// <param name="value">New value to set to parameter</param>
07417         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07418         void SetAxisParameter_AXIS_Z_OUTPUT(int axis, int value);
07419
07420
07421         /// <summary>
07422         /// Getter for DEMAND_EDGES axis parameter
07423         /// </summary>
07424         /// <param name="axis">Target axis whose parameter is retrieved</param>
07425         /// <returns>Current value of the parameter</returns>
07426         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07427         long GetAxisParameter_DEMAND_EDGES(int axis);
07428
07429
```

```
07430         /// <summary>
07431         /// Getter for DRIVE_INDEX axis parameter
07432         /// </summary>
07433         /// <param name="axis">Target axis whose parameter is retrieved</param>
07434         /// <returns>Current value of the parameter</returns>
07435         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07436         int GetAxisParameter_DRIVE_INDEX(int axis);
07437
07438         /// <summary>
07439         /// Setter for DRIVE_INDEX axis parameter
07440         /// </summary>
07441         /// <param name="axis">Target axis whose parameter is modified</param>
07442         /// <param name="value">New value to set to parameter</param>
07443         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07444         void SetAxisParameter_DRIVE_INDEX(int axis, int value);
07445
07446
07447         /// <summary>
07448         /// Getter for DRIVE_NEG_TORQUE axis parameter
07449         /// </summary>
07450         /// <param name="axis">Target axis whose parameter is retrieved</param>
07451         /// <returns>Current value of the parameter</returns>
07452         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07453         int GetAxisParameter_DRIVE_NEG_TORQUE(int axis);
07454
07455         /// <summary>
07456         /// Setter for DRIVE_NEG_TORQUE axis parameter
07457         /// </summary>
07458         /// <param name="axis">Target axis whose parameter is modified</param>
07459         /// <param name="value">New value to set to parameter</param>
07460         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07461         void SetAxisParameter_DRIVE_NEG_TORQUE(int axis, int value);
07462
07463
07464         /// <summary>
07465         /// Getter for DRIVE_PARAMETER axis parameter
07466         /// </summary>
07467         /// <param name="axis">Target axis whose parameter is retrieved</param>
07468         /// <returns>Current value of the parameter</returns>
07469         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07470         int GetAxisParameter_DRIVE_PARAMETER(int axis);
07471
07472         /// <summary>
07473         /// Setter for DRIVE_PARAMETER axis parameter
07474         /// </summary>
07475         /// <param name="axis">Target axis whose parameter is modified</param>
07476         /// <param name="value">New value to set to parameter</param>
07477         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07478         void SetAxisParameter_DRIVE_PARAMETER(int axis, int value);
07479
07480
07481         /// <summary>
07482         /// Getter for DRIVE_POS_TORQUE axis parameter
07483         /// </summary>
07484         /// <param name="axis">Target axis whose parameter is retrieved</param>
07485         /// <returns>Current value of the parameter</returns>
07486         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07487         int GetAxisParameter_DRIVE_POS_TORQUE(int axis);
07488
07489         /// <summary>
07490         /// Setter for DRIVE_POS_TORQUE axis parameter
07491         /// </summary>
07492         /// <param name="axis">Target axis whose parameter is modified</param>
07493         /// <param name="value">New value to set to parameter</param>
07494         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07495         void SetAxisParameter_DRIVE_POS_TORQUE(int axis, int value);
07496
07497
07498         /// <summary>
07499         /// Getter for DRIVE_SET_VAL axis parameter
07500         /// </summary>
07501         /// <param name="axis">Target axis whose parameter is retrieved</param>
07502         /// <returns>Current value of the parameter</returns>
07503         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07504         int GetAxisParameter_DRIVE_SET_VAL(int axis);
07505
07506         /// <summary>
07507         /// Setter for DRIVE_SET_VAL axis parameter
```

```
07508         /// </summary>
07509         /// <param name="axis">Target axis whose parameter is modified</param>
07510         /// <param name="value">New value to set to parameter</param>
07511         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07512         void SetAxisParameter_DRIVE_SET_VAL(int axis, int value);
07513
07514
07515         /// <summary>
07516         /// Getter for DRIVE_VALUE axis parameter
07517         /// </summary>
07518         /// <param name="axis">Target axis whose parameter is retrieved</param>
07519         /// <returns>Current value of the parameter</returns>
07520         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07521         int GetAxisParameter_DRIVE_VALUE(int axis);
07522
07523         /// <summary>
07524         /// Setter for DRIVE_VALUE axis parameter
07525         /// </summary>
07526         /// <param name="axis">Target axis whose parameter is modified</param>
07527         /// <param name="value">New value to set to parameter</param>
07528         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07529         void SetAxisParameter_DRIVE_VALUE(int axis, int value);
07530
07531
07532         /// <summary>
07533         /// Getter for FEED_OVERRIDE axis parameter
07534         /// </summary>
07535         /// <param name="axis">Target axis whose parameter is retrieved</param>
07536         /// <returns>Current value of the parameter</returns>
07537         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07538         double GetAxisParameter_FEED_OVERRIDE(int axis);
07539
07540         /// <summary>
07541         /// Setter for FEED_OVERRIDE axis parameter
07542         /// </summary>
07543         /// <param name="axis">Target axis whose parameter is modified</param>
07544         /// <param name="value">New value to set to parameter</param>
07545         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07546         void SetAxisParameter_FEED_OVERRIDE(int axis, double value);
07547
07548
07549         /// <summary>
07550         /// Getter for FORCE_ACCEL axis parameter
07551         /// </summary>
07552         /// <param name="axis">Target axis whose parameter is retrieved</param>
07553         /// <returns>Current value of the parameter</returns>
07554         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07555         double GetAxisParameter_FORCE_ACCEL(int axis);
07556
07557         /// <summary>
07558         /// Setter for FORCE_ACCEL axis parameter
07559         /// </summary>
07560         /// <param name="axis">Target axis whose parameter is modified</param>
07561         /// <param name="value">New value to set to parameter</param>
07562         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07563         void SetAxisParameter_FORCE_ACCEL(int axis, double value);
07564
07565
07566         /// <summary>
07567         /// Getter for FORCE_DECEL axis parameter
07568         /// </summary>
07569         /// <param name="axis">Target axis whose parameter is retrieved</param>
07570         /// <returns>Current value of the parameter</returns>
07571         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07572         double GetAxisParameter_FORCE_DECEL(int axis);
07573
07574         /// <summary>
07575         /// Setter for FORCE_DECEL axis parameter
07576         /// </summary>
07577         /// <param name="axis">Target axis whose parameter is modified</param>
07578         /// <param name="value">New value to set to parameter</param>
07579         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07580         void SetAxisParameter_FORCE_DECEL(int axis, double value);
07581
07582
07583         /// <summary>
07584         /// Getter for FORCE_JERK axis parameter
07585         /// </summary>
```

```
07586         /// <param name="axis">Target axis whose parameter is retrieved</param>
07587         /// <returns>Current value of the parameter</returns>
07588         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07589         double GetAxisParameter_FORCE_JERK(int axis);
07590
07591         /// <summary>
07592         /// Setter for FORCE_JERK axis parameter
07593         /// </summary>
07594         /// <param name="axis">Target axis whose parameter is modified</param>
07595         /// <param name="value">New value to set to parameter</param>
07596         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07597         void SetAxisParameter_FORCE_JERK(int axis, double value);
07598
07599
07600         /// <summary>
07601         /// Getter for FRAME axis parameter
07602         /// </summary>
07603         /// <param name="axis">Target axis whose parameter is retrieved</param>
07604         /// <returns>Current value of the parameter</returns>
07605         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07606         int GetAxisParameter_FRAME(int axis);
07607
07608         /// <summary>
07609         /// Setter for FRAME axis parameter
07610         /// </summary>
07611         /// <param name="axis">Target axis whose parameter is modified</param>
07612         /// <param name="value">New value to set to parameter</param>
07613         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07614         void SetAxisParameter_FRAME(int axis, int value);
07615
07616
07617         /// <summary>
07618         /// Getter for FWD_START axis parameter
07619         /// </summary>
07620         /// <param name="axis">Target axis whose parameter is retrieved</param>
07621         /// <returns>Current value of the parameter</returns>
07622         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07623         short GetAxisParameter_FWD_START(int axis);
07624
07625         /// <summary>
07626         /// Setter for FWD_START axis parameter
07627         /// </summary>
07628         /// <param name="axis">Target axis whose parameter is modified</param>
07629         /// <param name="value">New value to set to parameter</param>
07630         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07631         void SetAxisParameter_FWD_START(int axis, short value);
07632
07633
07634         /// <summary>
07635         /// Getter for POSI_SEQ_DELAY axis parameter
07636         /// </summary>
07637         /// <param name="axis">Target axis whose parameter is retrieved</param>
07638         /// <returns>Current value of the parameter</returns>
07639         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07640         int GetAxisParameter_POSI_SEQ_DELAY(int axis);
07641
07642         /// <summary>
07643         /// Setter for POSI_SEQ_DELAY axis parameter
07644         /// </summary>
07645         /// <param name="axis">Target axis whose parameter is modified</param>
07646         /// <param name="value">New value to set to parameter</param>
07647         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07648         void SetAxisParameter_POSI_SEQ_DELAY(int axis, int value);
07649
07650
07651         /// <summary>
07652         /// Getter for POSI_SEQ_MODE axis parameter
07653         /// </summary>
07654         /// <param name="axis">Target axis whose parameter is retrieved</param>
07655         /// <returns>Current value of the parameter</returns>
07656         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07657         int GetAxisParameter_POSI_SEQ_MODE(int axis);
07658
07659         /// <summary>
07660         /// Setter for POSI_SEQ_MODE axis parameter
07661         /// </summary>
07662         /// <param name="axis">Target axis whose parameter is modified</param>
07663         /// <param name="value">New value to set to parameter</param>
```

```
07664         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07665         void SetAxisParameter_POSI_SEQ_MODE(int axis, int value);
07666
07667
07668         /// <summary>
07669         /// Getter for PWM_CYCLE axis parameter
07670         /// </summary>
07671         /// <param name="axis">Target axis whose parameter is retrieved</param>
07672         /// <returns>Current value of the parameter</returns>
07673         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07674         uint GetAxisParameter_PWM_CYCLE(int axis);
07675
07676         /// <summary>
07677         /// Setter for PWM_CYCLE axis parameter
07678         /// </summary>
07679         /// <param name="axis">Target axis whose parameter is modified</param>
07680         /// <param name="value">New value to set to parameter</param>
07681         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07682         void SetAxisParameter_PWM_CYCLE(int axis, uint value);
07683
07684
07685         /// <summary>
07686         /// Getter for PWM_MARK axis parameter
07687         /// </summary>
07688         /// <param name="axis">Target axis whose parameter is retrieved</param>
07689         /// <returns>Current value of the parameter</returns>
07690         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07691         uint GetAxisParameter_PWM_MARK(int axis);
07692
07693         /// <summary>
07694         /// Setter for PWM_MARK axis parameter
07695         /// </summary>
07696         /// <param name="axis">Target axis whose parameter is modified</param>
07697         /// <param name="value">New value to set to parameter</param>
07698         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07699         void SetAxisParameter_PWM_MARK(int axis, uint value);
07700
07701
07702         /// <summary>
07703         /// Getter for REG_INPUTS axis parameter
07704         /// </summary>
07705         /// <param name="axis">Target axis whose parameter is retrieved</param>
07706         /// <returns>Current value of the parameter</returns>
07707         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07708         int GetAxisParameter_REG_INPUTS(int axis);
07709
07710         /// <summary>
07711         /// Setter for REG_INPUTS axis parameter
07712         /// </summary>
07713         /// <param name="axis">Target axis whose parameter is modified</param>
07714         /// <param name="value">New value to set to parameter</param>
07715         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07716         void SetAxisParameter_REG_INPUTS(int axis, int value);
07717
07718
07719         /// <summary>
07720         /// Getter for REGIST_CONTROL axis parameter
07721         /// </summary>
07722         /// <param name="axis">Target axis whose parameter is retrieved</param>
07723         /// <returns>Current value of the parameter</returns>
07724         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07725         int GetAxisParameter_REGIST_CONTROL(int axis);
07726
07727         /// <summary>
07728         /// Setter for REGIST_CONTROL axis parameter
07729         /// </summary>
07730         /// <param name="axis">Target axis whose parameter is modified</param>
07731         /// <param name="value">New value to set to parameter</param>
07732         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07733         void SetAxisParameter_REGIST_CONTROL(int axis, int value);
07734
07735
07736         /// <summary>
07737         /// Getter for REGIST_DELAY axis parameter
07738         /// </summary>
07739         /// <param name="axis">Target axis whose parameter is retrieved</param>
07740         /// <returns>Current value of the parameter</returns>
07741         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
```

```
module.</exception>
07742         double GetAxisParameter_REGIST_DELAY(int axis);
07743
07744         /// <summary>
07745         /// Setter for REGIST_DELAY axis parameter
07746         /// </summary>
07747         /// <param name="axis">Target axis whose parameter is modified</param>
07748         /// <param name="value">New value to set to parameter</param>
07749         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
07750         void SetAxisParameter_REGIST_DELAY(int axis, double value);
07751
07752
07753         /// <summary>
07754         /// Getter for REGIST_SPEED axis parameter
07755         /// </summary>
07756         /// <param name="axis">Target axis whose parameter is retrieved</param>
07757         /// <returns>Current value of the parameter</returns>
07758         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
07759         double GetAxisParameter_REGIST_SPEED(int axis);
07760
07761
07762         /// <summary>
07763         /// Getter for REGIST_SPEEDB axis parameter
07764         /// </summary>
07765         /// <param name="axis">Target axis whose parameter is retrieved</param>
07766         /// <returns>Current value of the parameter</returns>
07767         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
07768         double GetAxisParameter_REGIST_SPEEDB(int axis);
07769
07770
07771         /// <summary>
07772         /// Getter for REV_START axis parameter
07773         /// </summary>
07774         /// <param name="axis">Target axis whose parameter is retrieved</param>
07775         /// <returns>Current value of the parameter</returns>
07776         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
07777         short GetAxisParameter_REV_START(int axis);
07778
07779         /// <summary>
07780         /// Setter for REV_START axis parameter
07781         /// </summary>
07782         /// <param name="axis">Target axis whose parameter is modified</param>
07783         /// <param name="value">New value to set to parameter</param>
07784         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
07785         void SetAxisParameter_REV_START(int axis, short value);
07786
07787
07788         /// <summary>
07789         /// Getter for TCP_DPOS axis parameter
07790         /// </summary>
07791         /// <param name="axis">Target axis whose parameter is retrieved</param>
07792         /// <returns>Current value of the parameter</returns>
07793         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
07794         double GetAxisParameter_TCP_DPOS(int axis);
07795
07796
07797         /// <summary>
07798         /// Getter for TCP_FS_LIMIT axis parameter
07799         /// </summary>
07800         /// <param name="axis">Target axis whose parameter is retrieved</param>
07801         /// <returns>Current value of the parameter</returns>
07802         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
07803         double GetAxisParameter_TCP_FS_LIMIT(int axis);
07804
07805         /// <summary>
07806         /// Setter for TCP_FS_LIMIT axis parameter
07807         /// </summary>
07808         /// <param name="axis">Target axis whose parameter is modified</param>
07809         /// <param name="value">New value to set to parameter</param>
07810         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
07811         void SetAxisParameter_TCP_FS_LIMIT(int axis, double value);
07812
07813
07814         /// <summary>
07815         /// Getter for TCP_RS_LIMIT axis parameter
07816         /// </summary>
07817         /// <param name="axis">Target axis whose parameter is retrieved</param>
07818         /// <returns>Current value of the parameter</returns>
07819         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
```

```
module.</exception>
07820         double GetAxisParameter_TCP_RS_LIMIT(int axis);
07821
07822         /// <summary>
07823         /// Setter for TCP_RS_LIMIT axis parameter
07824         /// </summary>
07825         /// <param name="axis">Target axis whose parameter is modified</param>
07826         /// <param name="value">New value to set to parameter</param>
07827         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
07828         void SetAxisParameter_TCP_RS_LIMIT(int axis, double value);
07829
07830
07831         /// <summary>
07832         /// Getter for TCP_UNITS axis parameter
07833         /// </summary>
07834         /// <param name="axis">Target axis whose parameter is retrieved</param>
07835         /// <returns>Current value of the parameter</returns>
07836         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
07837         double GetAxisParameter_TCP_UNITS(int axis);
07838
07839         /// <summary>
07840         /// Setter for TCP_UNITS axis parameter
07841         /// </summary>
07842         /// <param name="axis">Target axis whose parameter is modified</param>
07843         /// <param name="value">New value to set to parameter</param>
07844         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
07845         void SetAxisParameter_TCP_UNITS(int axis, double value);
07846
07847
07848         /// <summary>
07849         /// Getter for ROBOT_FE axis parameter
07850         /// </summary>
07851         /// <param name="axis">Target axis whose parameter is retrieved</param>
07852         /// <returns>Current value of the parameter</returns>
07853         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
07854         double GetAxisParameter_ROBOT_FE(int axis);
07855
07856
07857         /// <summary>
07858         /// Getter for ROBOT_FE_LATCH axis parameter
07859         /// </summary>
07860         /// <param name="axis">Target axis whose parameter is retrieved</param>
07861         /// <returns>Current value of the parameter</returns>
07862         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
07863         double GetAxisParameter_ROBOT_FE_LATCH(int axis);
07864
07865
07866         /// <summary>
07867         /// Getter for ROBOT_FE_RANGE axis parameter
07868         /// </summary>
07869         /// <param name="axis">Target axis whose parameter is retrieved</param>
07870         /// <returns>Current value of the parameter</returns>
07871         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
07872         double GetAxisParameter_ROBOT_FE_RANGE(int axis);
07873
07874         /// <summary>
07875         /// Setter for ROBOT_FE_RANGE axis parameter
07876         /// </summary>
07877         /// <param name="axis">Target axis whose parameter is modified</param>
07878         /// <param name="value">New value to set to parameter</param>
07879         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
07880         void SetAxisParameter_ROBOT_FE_RANGE(int axis, double value);
07881
07882
07883
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
07884
07885
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
07886
07887         /// <summary>
07888         /// Getter for PROC_STATUS process parameter
07889         /// </summary>
07890         /// <param name="process">Target process whose parameter is retrieved</param>
07891         /// <returns>Current value of the parameter</returns>
07892         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
module.</exception>
07893         ProcessStatus GetProcessParameter_PROC_STATUS(int process);
07894
07895
```

```
07896         /// <summary>
07897         /// Getter for ERROR_LINE process parameter
07898         /// </summary>
07899         /// <param name="process">Target process whose parameter is retrieved</param>
07900         /// <returns>Current value of the parameter</returns>
07901         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07902         int GetProcessParameter_ERROR_LINE(int process);
07903
07904
07905         /// <summary>
07906         /// Getter for INDEVICE process parameter
07907         /// </summary>
07908         /// <param name="process">Target process whose parameter is retrieved</param>
07909         /// <returns>Current value of the parameter</returns>
07910         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07911         int GetProcessParameter_INDEVICE(int process);
07912
07913         /// <summary>
07914         /// Setter for INDEVICE process parameter
07915         /// </summary>
07916         /// <param name="process">Target process whose parameter is modified</param>
07917         /// <param name="value">New value to set to parameter</param>
07918         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07919         void SetProcessParameter_INDEVICE(int process, int value);
07920
07921
07922         /// <summary>
07923         /// Getter for OUTDEVICE process parameter
07924         /// </summary>
07925         /// <param name="process">Target process whose parameter is retrieved</param>
07926         /// <returns>Current value of the parameter</returns>
07927         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07928         int GetProcessParameter_OUTDEVICE(int process);
07929
07930         /// <summary>
07931         /// Setter for OUTDEVICE process parameter
07932         /// </summary>
07933         /// <param name="process">Target process whose parameter is modified</param>
07934         /// <param name="value">New value to set to parameter</param>
07935         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07936         void SetProcessParameter_OUTDEVICE(int process, int value);
07937
07938
07939         /// <summary>
07940         /// Getter for PMOVE process parameter
07941         /// </summary>
07942         /// <param name="process">Target process whose parameter is retrieved</param>
07943         /// <returns>Current value of the parameter</returns>
07944         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07945         bool GetProcessParameter_PMOVE(int process);
07946
07947
07948         /// <summary>
07949         /// Getter for PROC_LINE process parameter
07950         /// </summary>
07951         /// <param name="process">Target process whose parameter is retrieved</param>
07952         /// <returns>Current value of the parameter</returns>
07953         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07954         int GetProcessParameter_PROC_LINE(int process);
07955
07956
07957         /// <summary>
07958         /// Getter for RUN_ERROR process parameter
07959         /// </summary>
07960         /// <param name="process">Target process whose parameter is retrieved</param>
07961         /// <returns>Current value of the parameter</returns>
07962         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07963         int GetProcessParameter_RUN_ERROR(int process);
07964
07965
07966         /// <summary>
07967         /// Getter for TICKS process parameter
07968         /// </summary>
07969         /// <param name="process">Target process whose parameter is retrieved</param>
07970         /// <returns>Current value of the parameter</returns>
07971         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07972         long GetProcessParameter_TICKS(int process);
07973
```

```
07974         /// <summary>
07975         /// Setter for TICKS process parameter
07976         /// </summary>
07977         /// <param name="process">Target process whose parameter is modified</param>
07978         /// <param name="value">New value to set to parameter</param>
07979         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07980         void SetProcessParameter_TICKS(int process, long value);
07981
07982
07983
      ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
07984
07985
      ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
07986
07987         /// <summary>
07988         /// Getter for SERIAL_NUMBER slot parameter
07989         /// </summary>
07990         /// <param name="slot">Target slot whose parameter is retrieved</param>
07991         /// <returns>Current value of the parameter</returns>
07992         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
07993         int GetSlotParameter_SERIAL_NUMBER(int slot);
07994
07995         /// <summary>
07996         /// Setter for SERIAL_NUMBER slot parameter
07997         /// </summary>
07998         /// <param name="slot">Target slot whose parameter is modified</param>
07999         /// <param name="value">New value to set to parameter</param>
08000         /// <exception cref="TrioConnectionException">Thrown if error occurs in communication
      module.</exception>
08001         void SetSlotParameter_SERIAL_NUMBER(int slot, int value);
08002
08003
08004
      ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
08005
08006
      ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
08007
      ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
08008     }
08009
08010     #endregion  // IConnection interface
08011
08012 } // namespace Trio.UnifiedApi.Managed
```

7.3 TrioConnectionFactory.cs File Reference

Classes

- class Trio.UnifiedApi.TrioConnectionFactory

  Factory for creating communication instances.

Namespaces

- namespace Trio
- namespace Trio.UnifiedApi

7.4 TrioConnectionFactory.cs

Go to the documentation of this file.

```
00001 using System;
00002 using System.Diagnostics;
00003 using System.Globalization;
00004 using System.IO;
00005 using System.Reflection;
00006 using System.Threading.Tasks;
```

```
00007
00008 namespace Trio.UnifiedApi
00009 {
00010     /// <summary>
00011     /// Factory for creating communication instances
00012     /// </summary>
00013     public class TrioConnectionFactory
00014     {
00015         /// <summary>
00016         /// Delegate for transforming error code to message
00017         /// </summary>
00018         public static Func<ErrorCode, string> ErrorStringGetter =>
      TrioConnectionWrapperBase.ErrorStringGetter ?? TrioConnectionWrapperTCP.GetErrorString;
00019
00020         /// <summary>
00021         /// Create TCP based connection to Motion Coordinator
00022         /// </summary>
00023         /// <param name="hostName">Host name or IP of the Motion Coordinator to connect via TCP
      protocol</param>
00024         /// <param name="port">Port on Motion Coordinator open for TCP connections. It is normally
      3240 unless Motion Coorinator has custom configuration</param>
00025         /// <returns>On success an instance of connection interface is returned</returns>
00026         public static ITrioConnection CreateConnectionTCP(string hostName = null, ushort port = 0)
00027         {
00028             return TrioConnectionWrapperTCP.Create(hostName, port);
00029         }
00030
00031         /// <summary>
00032         /// Create PC-MCAT based connection to Motion Coordinator
00033         /// </summary>
00034         /// <returns>On success an instance of connection interface is returned</returns>
00035         public static ITrioConnection CreateConnectionPCMCAT()
00036         {
00037             return TrioConnectionWrapperPCMCAT.Create();
00038         }
00039
00040         /// <summary>
00041         /// Create Flex7 based connection to Motion Coordinator
00042         /// </summary>
00043         /// <returns>On success an instance of connection interface is returned</returns>
00044         public static ITrioConnection CreateConnectionFlex7()
00045         {
00046             return TrioConnectionWrapperFlex7.Create();
00047         }
00048     }
00049 }
```

Index

_ApiError
Trio.UnifiedApi, 24
_ApiLast
Trio.UnifiedApi, 24
_ControllerError
Trio.UnifiedApi, 24
_SystemError
Trio.UnifiedApi, 24
_Unknown
Trio.UnifiedApi, 24
__Size
Trio.UnifiedApi, 35

Abort
Trio.UnifiedApi, 24
ABORTED
Trio.UnifiedApi, 38
ACCPROFILE
Trio.UnifiedApi, 36
AddAx
Trio.UnifiedApi.ITrioConnection, 75
AddDac
Trio.UnifiedApi.ITrioConnection, 75, 76
AFFDACout
Trio.UnifiedApi, 20
Alarms
Trio.UnifiedApi, 35
ALLOW_CONNECT_MOVES_IN_LIMIT
Trio.UnifiedApi, 17
ANALout_ANALin
Trio.UnifiedApi, 19
Api_Unknown
Trio.UnifiedApi, 24
AS_AXIS_FS_LIMIT
Trio.UnifiedApi, 18
AS_AXIS_RS_LIMIT
Trio.UnifiedApi, 18
AS_BISS_CLEAN_WARNING
Trio.UnifiedApi, 18
AS_CANCEL_BUFFERED
Trio.UnifiedApi, 18
AS_CANCEL_BUFFERED_DECEL_INTO_BUFFERED_MOVES
Trio.UnifiedApi, 18
AS_CANCEL_BUFFERED_DECEL_WITHIN_CURRENT_MOVE
Trio.UnifiedApi, 18
AS_CANCEL_CURRENT
Trio.UnifiedApi, 18
AS_CANCEL_CURRENT_AT_SPEED
Trio.UnifiedApi, 18
AS_CANCEL_LAST_BUFFERED
Trio.UnifiedApi, 18
AS_CANCEL_MOVEADD
Trio.UnifiedApi, 18
AS_COMMS_ERROR
Trio.UnifiedApi, 17
AS_DATUMING
Trio.UnifiedApi, 18
AS_ENCODER_OVER_I
Trio.UnifiedApi, 18
AS_FE_EXCEEDS_LIMIT
Trio.UnifiedApi, 18
AS_FE_WARNING
Trio.UnifiedApi, 17
AS_FEEDHOLD
Trio.UnifiedApi, 18
AS_FORWARD_SOFTWARE_LIMIT
Trio.UnifiedApi, 18
AS_IN_FORWARD_LIMIT
Trio.UnifiedApi, 18
AS_IN_REVERSE_LIMIT
Trio.UnifiedApi, 18
AS_KINEMATICS_1HOUR
Trio.UnifiedApi, 18
AS_LOOKAHEAD_OVERRIDE
Trio.UnifiedApi, 17
AS_MOVE_CANCEL_DONE
Trio.UnifiedApi, 18
AS_MOVEADD
Trio.UnifiedApi, 18
AS_OPEN_FRAME_ERROR
Trio.UnifiedApi, 18
AS_OVERRIDE_TO_ZERO
Trio.UnifiedApi, 18
AS_PSWITCH_FIFO_FULL
Trio.UnifiedApi, 18
AS_PSWITCH_FIFO_HALF_FULL
Trio.UnifiedApi, 18
AS_PSWITCH_FIFO_NOT_EMPTY
Trio.UnifiedApi, 18
AS_PULSE_OVER_SPEED
Trio.UnifiedApi, 18
AS_REMOTE_DRIVE_ERROR
Trio.UnifiedApi, 17
AS_REMOTE_NODE_ERROR2
Trio.UnifiedApi, 18
AS_REMOTE_NODE_STATUS_ERROR
Trio.UnifiedApi, 18
AS_REVERSE_SOFTWARE_LIMIT
Trio.UnifiedApi, 18

AS_SINGULARITY
Trio.UnifiedApi, 18
AS_VOLUME_LIMIT
Trio.UnifiedApi, 18
AS_VP_ERROR
Trio.UnifiedApi, 18
AS_WORLD_FS_LIMIT
Trio.UnifiedApi, 18
AS_WORLD_RS_LIMIT
Trio.UnifiedApi, 18
AUTO_BLEND_MODE_0_USING_CORNER_RADIUS
Trio.UnifiedApi, 23
AUTO_BLEND_MODE_1_USING_CORNER_RADIUS
Trio.UnifiedApi, 23
AUTO_RADIUS_SPEED_CONTROL
Trio.UnifiedApi, 23
AUTO_RADIUS_SPEED_CONTROL_CENTRIPETAL_MODE
Trio.UnifiedApi, 23
AUTO_RADIUS_SPEED_CONTROL_CENTRIPETAL_MODE_USE_ACCEL
Trio.UnifiedApi, 23
AUTO_SPEED_CONTROL
Trio.UnifiedApi, 23
Autorun
Trio.UnifiedApi, 34
autorun_process
Trio.UnifiedApi.FileInfo, 45
AxesDiff
Trio.UnifiedApi, 20
AxisMode
Trio.UnifiedApi, 17
AxisParam
Trio.UnifiedApi, 38
AxisStatus
Trio.UnifiedApi, 17
AxisType
Trio.UnifiedApi, 18

Backlash
Trio.UnifiedApi.ITrioConnection, 76
BacklashDisable
Trio.UnifiedApi.ITrioConnection, 77
BadArgument
Trio.UnifiedApi, 24
BadResponse
Trio.UnifiedApi, 24
BASIC
Trio.UnifiedApi, 34
BASICLibrary
Trio.UnifiedApi, 35
Binary
Trio.UnifiedApi, 34
BISSin
Trio.UnifiedApi, 20
Boolean
Trio.UnifiedApi, 22
BufferTooSmall
Trio.UnifiedApi, 24

CAM
Trio.UnifiedApi, 35
Cam
Trio.UnifiedApi.ITrioConnection, 77
CAMBOX
Trio.UnifiedApi, 35
CamBox
Trio.UnifiedApi.ITrioConnection, 77
CamBoxLinkOptions
Trio.UnifiedApi, 21
CAMIN
Trio.UnifiedApi, 36
CAMLINK
Trio.UnifiedApi, 36
CAN
Trio.UnifiedApi, 19
CAN_Comms
Trio.UnifiedApi, 19
CAN_Remote
Trio.UnifiedApi, 19
Cancel
Trio.UnifiedApi.ITrioConnection, 78
CANCEL_ADDAX_ON_LIMIT
Trio.UnifiedApi, 17
CANCEL_MTYPE_ON_MASTER_IF_ADDAX_HITS_LIM
Trio.UnifiedApi, 17
ClearVariables
Trio.UnifiedApi.ITrioConnection, 79
CloseConnection
Trio.UnifiedApi.ITrioConnection, 79
Co_ObjectType
Trio.UnifiedApi, 22
COMBINEAXES
Trio.UnifiedApi, 36
COMPENSATE_XY
Trio.UnifiedApi, 36
compiled_size
Trio.UnifiedApi.FileInfo, 45
CompressDecompress
Trio.UnifiedApi, 34
Compression
Trio.UnifiedApi, 24
CompressOnly
Trio.UnifiedApi, 34
CONNECT
Trio.UnifiedApi, 35
Connect
Trio.UnifiedApi.ITrioConnection, 79
ConnectionContext
Trio.UnifiedApi, 24
ConnectionEvent
Trio.UnifiedApi.ITrioConnection, 337
ConnectionEventHandler
Trio.UnifiedApi, 40
ConnectionModule
Trio.UnifiedApi, 24
ConnectionType
Trio.UnifiedApi, 22
Trio.UnifiedApi.ITrioConnection, 337

ConnectMode
Trio.UnifiedApi, 23
CONNPATH
Trio.UnifiedApi, 36
ConnPath
Trio.UnifiedApi.ITrioConnection, 79
ControllerError_APPEND_BLOCK_INCONSISTENT
Trio.UnifiedApi, 27
ControllerError_AS_EXPECTED
Trio.UnifiedApi, 27
ControllerError_ASSIGNMENT_EXPECTED
Trio.UnifiedApi, 24
ControllerError_ATYPE_MUST_BE_CONFIGURED_USING_MC_CONFIG_COMMAND_FOR_THIS_MODE
Trio.UnifiedApi, 31
ControllerError_AXIS_LOCKED_TO_A_ROBOT
Trio.UnifiedApi, 31
ControllerError_BASE_AXIS_DOES_NOT_BELONG_TO_ANY_KINEMATIC_CONSTANT
Trio.UnifiedApi, 30
ControllerError_BASIC_TOKEN_DEFINITION_INCORRECT
Trio.UnifiedApi, 27
ControllerError_BIT_NUMBER_OUT_OF_RANGE
Trio.UnifiedApi, 29
ControllerError_CAM_TABLE_ID_NOT_DEFINED
Trio.UnifiedApi, 31
ControllerError_CANNOT_EDIT_PROGRAM
Trio.UnifiedApi, 26
ControllerError_CANNOT_MODIOFY_SPECIAL_PROGRAM
Trio.UnifiedApi, 25
ControllerError_CANNOT_NEST_ONE_LINE_IF
Trio.UnifiedApi, 26
ControllerError_CANNOT_RESET_WHEN_DRIVE_SERVO_ON
Trio.UnifiedApi, 26
ControllerError_CANNOT_SELECT_PROGRAM
Trio.UnifiedApi, 25
ControllerError_CANNOT_SPECIFY_PROGRAM_TYPE_FOR_AN_EXISTING_PROGRAM
Trio.UnifiedApi, 27
ControllerError_CASE_OR_END_CASE_WITHOUT_PREVIOUS_SELECT_CASE
Trio.UnifiedApi, 31
ControllerError_CLOSE_ROUND_BRACKET_EXPECTED
Trio.UnifiedApi, 25
ControllerError_CLOSE_SQUARE_BRACKET_EXPECTED
Trio.UnifiedApi, 25
ControllerError_COLLISION_OBJECTS_CANNOT_BE_MODIFIED_OR_DELETED_WHILE_LOCKED
Trio.UnifiedApi, 31
ControllerError_COMMA_EXPECTED
Trio.UnifiedApi, 25
ControllerError_COMMAND_INHIBITED_ROBOT_MUST_BE_USED_TO_ACTIVATE_KINEMATIC_CONSTANT
Trio.UnifiedApi, 31
ControllerError_COMMAND_IS_COMMAND_LINE_ONLY
Trio.UnifiedApi, 25
ControllerError_COMMAND_IS_RUNTIME_ONLY
Trio.UnifiedApi, 25
ControllerError_COMMAND_LINE_BROKEN_BY_ESC
Trio.UnifiedApi, 25
ControllerError_COMMAND_LINE_TOO_LONG
Trio.UnifiedApi, 31
ControllerError_COMMAND_NOT_ALLOWED_IN_FRAME
Trio.UnifiedApi, 31
ControllerError_COMMAND_NOT_ALLOWED_WHEN_RUNNING_MOTIO
Trio.UnifiedApi, 26
ControllerError_COMMAND_NOT_ALLOWED_WITH_CURRENT_ATYPE
Trio.UnifiedApi, 28
ControllerError_COMMAND_NOT_ALLOWED_WITHIN_INCLUDE_FILE
Trio.UnifiedApi, 27
ControllerError_COMMAND_NOT_PERMITTED_IN_BASIC_LIBRARY
Trio.UnifiedApi, 31
ControllerError_COMMAND_NOT_PERMITTED_IN_FUNCTION
Trio.UnifiedApi, 30
ControllerError_COMMAND_NOT_RECOGNISED
Trio.UnifiedApi, 24
ControllerError_COMMAND_NOT_VALID_WHEN_RUNNING_MOTI
Trio.UnifiedApi, 28
ControllerError_COMMAND_OUT_OF_CONTEXT
Trio.UnifiedApi, 25
ControllerError_CONSTANT_VALUE_EXPECTED
Trio.UnifiedApi, 32
ControllerError_CONTINUE_WITHOUT_PREVIOUS_FOR_WHILE_OR_R
Trio.UnifiedApi, 30
ControllerError_CYCLIC_INCLUDE_DEPENDENCY
Trio.UnifiedApi, 30
ControllerError_DATA_TYPE_EXPECTED
Trio.UnifiedApi, 27
ControllerError_DEVICE_ERROR__COMMAND_NOT_SUPPORTED_BY
Trio.UnifiedApi, 27
ControllerError_DEVICE_ERROR__CRC_ERROR
Trio.UnifiedApi, 27
ControllerError_DEVICE_ERROR__ERROR_WRITING_TO_DEVICE
Trio.UnifiedApi, 28
ControllerError_DEVICE_ERROR__INVALID_RESPONSE_FROM_DEVI
Trio.UnifiedApi, 28
ControllerError_DEVICE_ERROR__TIMEOUT_WAITING_FOR_DEVICE
Trio.UnifiedApi, 27
ControllerError_DEVICE_ERROR_NO_SDCARD_INSERTED
Trio.UnifiedApi, 29
ControllerError_DEVICE_ERROR_SDCARD_WRITE_PROTECTED
Trio.UnifiedApi, 29
ControllerError_DIRECTORY_ALREADY_LOCKED
Trio.UnifiedApi, 26
ControllerError_DIRECTORY_IS_LOCKED
Trio.UnifiedApi, 26
ControllerError_DIRECTORY_NOT_LOCKED
Trio.UnifiedApi, 26
ControllerError_DIRECTORY_STRUCTURE_INVALID
Trio.UnifiedApi, 26
ControllerError_DISK_ERROR_DISK_MUST_BE_FAT32
Trio.UnifiedApi, 29
ControllerError_DISK_ERROR__INVALID_BOOT_SECTOR
Trio.UnifiedApi, 28
ControllerError_DISK_ERROR__INVALID_MBR
Trio.UnifiedApi, 28
ControllerError_DISK_ERROR__INVALID_SECTOR_CLUSTER_REFERE
Trio.UnifiedApi, 28
ControllerError_DIVISION_BY_ZERO
Trio.UnifiedApi, 24
ControllerError_DOWNLOAD_ABORT_OR_TIMEOUT
Trio.UnifiedApi, 27

ControllerError_DUPLICATE_AXES_NOT_PERMITTED
Trio.UnifiedApi, 26
ControllerError_DUPLICATE_IDENTIFIER
Trio.UnifiedApi, 25
ControllerError_DUPLICATE_NAME_USED_IN_LABEL_AND_VARIABLE
Trio.UnifiedApi, 30
ControllerError_DUPLICATE_VR_INDEX
Trio.UnifiedApi, 32
ControllerError_ELSE_ELSEIF_ENDIF_WITHOUT_PREVIOUS_IF
Trio.UnifiedApi, 25
ControllerError_ENCRYPTION_AUTHORITY_NOT_FOUND
Trio.UnifiedApi, 31
ControllerError_END_OF_FILE
Trio.UnifiedApi, 28
ControllerError_ENDFUNC_WITHOUT_PREVIOUS_FUNCTION
Trio.UnifiedApi, 29
ControllerError_ETHERCAT_AXES_CANNOT_BE_SET_USING_ATYPE_FLEXSERVO_DRIVE
Trio.UnifiedApi, 29
ControllerError_EVALUATION_ERROR
Trio.UnifiedApi, 26
ControllerError_EXIT_WITHOUT_PREVIOUS_FOR_WHILE_OR_REPEAT
Trio.UnifiedApi, 30
ControllerError_EXTRA_CHARS_AT_EOLN
Trio.UnifiedApi, 24
ControllerError_FAILED_ATTEMPT_1
Trio.UnifiedApi, 29
ControllerError_FAILED_ATTEMPT_2
Trio.UnifiedApi, 29
ControllerError_FAILED_OPEN_CONNECTION
Trio.UnifiedApi, 29
ControllerError_FAILED_TOO_MANY_TIMES_PLEASE_RESET
Trio.UnifiedApi, 30
ControllerError_FEATURES_ENABLED_DO_NOT_ALLOW_ATYPE_CHANGE
Trio.UnifiedApi, 26
ControllerError_FILE_ALREADY_OPEN
Trio.UnifiedApi, 28
ControllerError_FILE_ERROR__DIRECTORY_FULL
Trio.UnifiedApi, 28
ControllerError_FILE_ERROR__DISK_FULL
Trio.UnifiedApi, 28
ControllerError_FILE_ERROR__FILE_NOT_FOUND
Trio.UnifiedApi, 28
ControllerError_FILE_ERROR__FILENAME_ALREADY_EXISTS
Trio.UnifiedApi, 28
ControllerError_FILE_ERROR__INVALID_FILENAME
Trio.UnifiedApi, 28
ControllerError_FILE_NOT_OPEN
Trio.UnifiedApi, 28
ControllerError_FIRMWARE_ERROR__CANNOT_REFERENCE_DATA_OUTSIDE_ARRAY_BLOCK
Trio.UnifiedApi, 28
ControllerError_FIRMWARE_ERROR__DEVICE_IN_USE
Trio.UnifiedApi, 27
ControllerError_FIXED_INTEGER_OR_ELSE_EXPECTED
Trio.UnifiedApi, 31
ControllerError_FLASH_ERASE_ERROR
Trio.UnifiedApi, 25
ControllerError_FLASH_PROGRAMMING_ERROR
Trio.UnifiedApi, 24
ControllerError_FLASH_STICK_BLANK
Trio.UnifiedApi, 26
ControllerError_FLASH_STICK_NOT_AVAILABLE_ON_THIS_CONTROL
Trio.UnifiedApi, 27
ControllerError_FLASH_VALUE_CANNOT_BE_SET
Trio.UnifiedApi, 26
ControllerError_FOR_EXPECTED
Trio.UnifiedApi, 28
ControllerError_FORWARD_TRANSFORM_FUNCTION_NOT_DEFINED
Trio.UnifiedApi, 32
ControllerError_FRAME_NOT_SUPPORTED_BY_RPS
Trio.UnifiedApi, 32
ControllerError_FUNCTION_EXPECTED
Trio.UnifiedApi, 31
ControllerError_FUNCTION_MUST_BE_DEFINED_WITHIN_A_BASIC_L
Trio.UnifiedApi, 29
ControllerError_FUNCTION_NAME_EXPECTED
Trio.UnifiedApi, 29
ControllerError_GLOBAL_DEFINITION_IS_DIFFERENT
Trio.UnifiedApi, 27
ControllerError_GOTO_GOSUB_EXEC_EXPECTED
Trio.UnifiedApi, 25
ControllerError_HASH_EXPECTED
Trio.UnifiedApi, 28
ControllerError_ID_NOT_DEFINED
Trio.UnifiedApi, 30
ControllerError_IDENTIFIER_EXPECTED
Trio.UnifiedApi, 25
ControllerError_IEC_ERROR__INVALID_VARIABLE_ACCESS
Trio.UnifiedApi, 28
ControllerError_IEC_IO_BINDING_INDEX_OUT_OF_RANGE
Trio.UnifiedApi, 29
ControllerError_IEC_RUNTIME_LIMITED_TO_1_HOUR_FEC_21
Trio.UnifiedApi, 28
ControllerError_IEC_TABLE_BINDING_INDEX_OUT_OF_RANGE
Trio.UnifiedApi, 29
ControllerError_IEC_TASK_MEMORY_ALLOCATION_ERROR
Trio.UnifiedApi, 29
ControllerError_IEC_VR_BINDING_INDEX_OUT_OF_RANGE
Trio.UnifiedApi, 29
ControllerError_INCLUDE_FILE_MUST_CONTAIN_DEFCONST_DIM_ST
Trio.UnifiedApi, 32
ControllerError_INCOMPATIBLE_ARRAY_DIMENSIONS
Trio.UnifiedApi, 29
ControllerError_INCORRECT_SYMBOL_TYPE
Trio.UnifiedApi, 27
ControllerError_INPUT__OUTPUT__APPEND__FIFO_READ__FIFO_WR
Trio.UnifiedApi, 28
ControllerError_INSUFFICIENT_ARRAY_DIMENSION
Trio.UnifiedApi, 32
ControllerError_INVALID_COFF_IMAGE_FILE
Trio.UnifiedApi, 27
ControllerError_INVALID_FLOATING_POINT_OPERATION
Trio.UnifiedApi, 28
ControllerError_INVALID_FORMAT_SPECIFIER
Trio.UnifiedApi, 32
ControllerError_INVALID_FPGA_VERSION
Trio.UnifiedApi, 29

ControllerError_INVALID_FRAME_NUMBER
Trio.UnifiedApi, 32
ControllerError_INVALID_IO_CHANNEL
Trio.UnifiedApi, 26
ControllerError_INVALID_LINE_NUMBER
Trio.UnifiedApi, 26
ControllerError_INVALID_MIX_OF_DATA_TYPES
Trio.UnifiedApi, 28
ControllerError_INVALID_MOVEC_INPUT
Trio.UnifiedApi, 31
ControllerError_INVALID_MOVEOPTION
Trio.UnifiedApi, 32
ControllerError_INVALID_MSPHERICAL_INPUT
Trio.UnifiedApi, 27
ControllerError_INVALID_OPEN_ARCH_FUNC_DEFINITION
Trio.UnifiedApi, 32
ControllerError_INVALID_PROGRAM_NAME
Trio.UnifiedApi, 27
ControllerError_INVALID_PROGRAM_TYPE
Trio.UnifiedApi, 27
ControllerError_INVALID_RANGE_SPECIFIED
Trio.UnifiedApi, 27
ControllerError_INVALID_STARTUP_CONFIGURATION_COMMAND
Trio.UnifiedApi, 28
ControllerError_INVALID_STORAGE_AREA
Trio.UnifiedApi, 28
ControllerError_INVALID_SYSTEM_CODE_WRONG_CONTROLLER
Trio.UnifiedApi, 28
ControllerError_INVALID_TRANSFER_TYPE
Trio.UnifiedApi, 24
ControllerError_INVALID_USE_OF_DOLAR_SIGN
Trio.UnifiedApi, 26
ControllerError_INVERSE_TRANSFORM_FUNCTION_NOT_DEFINED
Trio.UnifiedApi, 32
ControllerError_KEYWORD_FUNCTION_NOT_DEFINED
Trio.UnifiedApi, 32
ControllerError_KEYWORD_INACTIVE
Trio.UnifiedApi, 32
ControllerError_KEYWORD_SAME_AS_SYSTEM_WORD
Trio.UnifiedApi, 32
ControllerError_KINEMATICS_RUNTIME_LIMITED_TO_1_GROUP
Trio.UnifiedApi, 28
ControllerError_LABEL_EXPECTED
Trio.UnifiedApi, 25
ControllerError_LABEL_MUST_BE_AT_START_OF_LINE
Trio.UnifiedApi, 26
ControllerError_LABEL_NOT_FOUND
Trio.UnifiedApi, 26
ControllerError_LINE_MUST_BE_INSIDE_FUNCTION
Trio.UnifiedApi, 30
ControllerError_LINE_NUMBER_CANNOT_HAVE_DECIMAL_POINT
Trio.UnifiedApi, 26
ControllerError_LOCAL_TARGET_VALUE_HAS_NOT_BEEN_DEFINED
Trio.UnifiedApi, 31
ControllerError_MAC_EEPROM_IS_LOCKED
Trio.UnifiedApi, 28
ControllerError_MASTER_ERROR
Trio.UnifiedApi, 27
ControllerError_MATRIX_IS_SINGULAR
Trio.UnifiedApi, 29
ControllerError_MAX_OPEN_KEYWORDS_REACHED
Trio.UnifiedApi, 32
ControllerError_MAX_OPEN_MOVES_REACHED
Trio.UnifiedApi, 32
ControllerError_MAXIMUM_NUMBER_OF_PROGRAM_INSTANCES_AC
Trio.UnifiedApi, 26
ControllerError_MCCONFIG_CHANGED_EX_REQUIRED
Trio.UnifiedApi, 29
ControllerError_MISSING_ENDFUNC
Trio.UnifiedApi, 29
ControllerError_MODIFIER_NOT_ALLOWED
Trio.UnifiedApi, 25
ControllerError_MORE_THAN_ONE_PROGRAM_RUNNING_WHEN_TR
Trio.UnifiedApi, 27
ControllerError_MOVE_TRIGGER_NOT_SPECIFIED
Trio.UnifiedApi, 32
ControllerError_MOVE_TRIGGER_RESPONSE_NOT_SPECIFIED
Trio.UnifiedApi, 32
ControllerError_MULTIPLE_MOVE_TRIGGERS_SPECIFIED
Trio.UnifiedApi, 31
ControllerError_NETWORK_PROTOCOL_ERROR
Trio.UnifiedApi, 27
ControllerError_NETWORK_TIMEOUT
Trio.UnifiedApi, 27
ControllerError_NEXT_WITHOUT_FOR
Trio.UnifiedApi, 25
ControllerError_NO_CODE_AVAILABLE_FOR_PROGRAM
Trio.UnifiedApi, 25
ControllerError_NO_FREE_ENCRYPTION_AUTHORITY
Trio.UnifiedApi, 31
ControllerError_NO_MORE_PROGRAMS_AVAILABLE
Trio.UnifiedApi, 25
ControllerError_NO_PROCESS_AVAILABLE
Trio.UnifiedApi, 25
ControllerError_NO_PROGRAM_SELECTED
Trio.UnifiedApi, 25
ControllerError_NO_RESPONSE_FROM_CAN_MODULE
Trio.UnifiedApi, 29
ControllerError_NONE
Trio.UnifiedApi, 24
ControllerError_NUMBER_EXPECTED
Trio.UnifiedApi, 27
ControllerError_NUMERICAL_ERROR__INF_USED
Trio.UnifiedApi, 28
ControllerError_NUMERICAL_ERROR__NAN_USED
Trio.UnifiedApi, 28
ControllerError_NUMERICAL_ERROR__SUBNORMAL_VALUE_USED
Trio.UnifiedApi, 28
ControllerError_OBJECT_FRAME_ALREADY_ATTACHED
Trio.UnifiedApi, 30
ControllerError_OBJECT_FRAME_ALREADY_SYNCED
Trio.UnifiedApi, 30
ControllerError_OBJECT_FRAME_CANNOT_BE_MODIFIED_OR_DELE
Trio.UnifiedApi, 30
ControllerError_OBJECT_FRAME_HAS_NOT_BEEN_ATTACHED
Trio.UnifiedApi, 30

ControllerError_OBJECT_FRAME_ID_NOT_DEFINED
Trio.UnifiedApi, 30
ControllerError_OPEN_BRACKET_EXPECTED
Trio.UnifiedApi, 27
ControllerError_OPEN_MOVE_FUNCTIONS_NOT_DEFINED
Trio.UnifiedApi, 32
ControllerError_OPERAND_EXPECTED
Trio.UnifiedApi, 24
ControllerError_OPERATION_CANNOT_BE_PERFORMED_UNTIL_EDLE
Trio.UnifiedApi, 30
ControllerError_OUT_OF_MEMORY
Trio.UnifiedApi, 25
ControllerError_PARAMETER_CAN_ONLY_BE_CHANGED_USING_MO_COMMAND
Trio.UnifiedApi, 29
ControllerError_PARAMETER_CAN_ONLY_BE_CHANGED_USING_MO_COMMAND
Trio.UnifiedApi, 29
ControllerError_PARAMETER_EXPECTED
Trio.UnifiedApi, 27
ControllerError_PARAMETER_NAME_EXPECTED
Trio.UnifiedApi, 29
ControllerError_PARAMETER_OUT_OF_RANGE
Trio.UnifiedApi, 25
ControllerError_PLC_TYPE_IS_INVALID
Trio.UnifiedApi, 26
ControllerError_PROCESS_ALREADY_SELECTED
Trio.UnifiedApi, 26
ControllerError_PROCESS_IN_USE
Trio.UnifiedApi, 32
ControllerError_PROGRAM_ALREADY_EXISTS
Trio.UnifiedApi, 26
ControllerError_PROGRAM_CORRUPT
Trio.UnifiedApi, 27
ControllerError_PROGRAM_ENCRYPTED
Trio.UnifiedApi, 27
ControllerError_PROGRAM_FAILED_TO_RUN
Trio.UnifiedApi, 30
ControllerError_PROGRAM_IS_LOCKED
Trio.UnifiedApi, 25
ControllerError_PROGRAM_IS_NOT_EXECUTABLE_TYPE
Trio.UnifiedApi, 29
ControllerError_PROGRAM_IS_STOPPED
Trio.UnifiedApi, 25
ControllerError_PROGRAM_IS_STOPPED_HALT_FORCED
Trio.UnifiedApi, 29
ControllerError_PROGRAM_NOT_FOUND
Trio.UnifiedApi, 25
ControllerError_PROGRAM_NOT_PAUSED
Trio.UnifiedApi, 26
ControllerError_PROGRAM_NOT_PAUSED_ON_THIS_PROCESS
Trio.UnifiedApi, 26
ControllerError_PROGRAM_NOT_RUNNING
Trio.UnifiedApi, 26
ControllerError_PROGRAM_NOT_RUNNING_ON_THIS_PROCESS
Trio.UnifiedApi, 26
ControllerError_PROGRAMS_RUNNING
Trio.UnifiedApi, 25
ControllerError_QUOTES_EXPECTED
Trio.UnifiedApi, 24
ControllerError_REMOTE_AXIS_IS_IN_USE
Trio.UnifiedApi, 25
ControllerError_RESERVED_KEYWORD_NOT_AVAILABLE_ON_THIS_C
Trio.UnifiedApi, 26
ControllerError_RETURN_STATEMENT_EXPECTED_BEFORE_ENDFUN
Trio.UnifiedApi, 30
ControllerError_RETURN_WITHOUT_GOSUB
Trio.UnifiedApi, 26
ControllerError_ROBOT_CONFIGURATION_CANNOT_BE_CHANGED_W
Trio.UnifiedApi, 30
ControllerError_ROBOT_CONTROLLED_BY_ANOTHER_PROCESS
Trio.UnifiedApi, 31
ControllerError_ROBOT_FRAME_CANNOT_BE_MODIFIED_OR_DELET
Trio.UnifiedApi, 30
ControllerError_ROBOT_FRAME_NOT_DEFINED
Trio.UnifiedApi, 30
ControllerError_ROBOT_MOVES_CANNOT_BE_PERFORMED_UNTIL_J
Trio.UnifiedApi, 30
ControllerError_ROBOT_NOT_DEFINED_USING_KINEMATIC_GROUP
Trio.UnifiedApi, 31
ControllerError_ROBOT_PROGRAM_STOPPED
Trio.UnifiedApi, 30
ControllerError_ROBOTICS_RUNTIME_LIMITED_TO_1_HOUR
Trio.UnifiedApi, 31
ControllerError_RPS_KERNEL_V2_NOT_ENABLED
Trio.UnifiedApi, 32
ControllerError_RUNNING_OF_RPS_PROGRAMS_INHIBITED_USE_DIS
Trio.UnifiedApi, 31
ControllerError_SCOPE_PERIOD_SHOULD_EXCEED_NUMBER_OF_AI
Trio.UnifiedApi, 26
ControllerError_SERIAL_NUMBER_MUST_BE_MINUS_1
Trio.UnifiedApi, 27
ControllerError_SINGULARITY
Trio.UnifiedApi, 30
ControllerError_SLAVE_ERROR
Trio.UnifiedApi, 27
ControllerError_STACK_OVERFLOW
Trio.UnifiedApi, 24
ControllerError_START_OF_EXPRESSION_EXPECTED
Trio.UnifiedApi, 25
ControllerError_STRING_EXCEEDS_PERMITTED_LENGTH
Trio.UnifiedApi, 26
ControllerError_STRING_EXPECTED
Trio.UnifiedApi, 27
ControllerError_STRUCTURE_FIELD_NAME_NOT_RECOGNISED
Trio.UnifiedApi, 30
ControllerError_STRUCTURE_NESTING_ERROR
Trio.UnifiedApi, 25
ControllerError_SYMBOL_IS_NOT_A_VARIABLE
Trio.UnifiedApi, 28
ControllerError_SYMBOL_TABLE_LOCKED
Trio.UnifiedApi, 27
ControllerError_SYNC_IN_PROCESS
Trio.UnifiedApi, 30
ControllerError_TABLE_INDEX_RANGE_ERROR
Trio.UnifiedApi, 26
ControllerError_TABLE_LOCATION_WRITE_PROTECTED
Trio.UnifiedApi, 31

ControllerError_TABLE_MEMORY_CONFLICT
Trio.UnifiedApi, 32
ControllerError_TARGET_OBJECT_FRAME_MISMATCH
Trio.UnifiedApi, 32
ControllerError_TARGET_POINT_NOT_ACTIVATED
Trio.UnifiedApi, 30
ControllerError_TARGET_POINT_OUT_OF_ROBOT_WORKING_AREA
Trio.UnifiedApi, 31
ControllerError_TARGET_ROBOT_CONFIG_MISMATCH
Trio.UnifiedApi, 32
ControllerError_TARGET_ROBOT_FRAME_MISMATCH
Trio.UnifiedApi, 32
ControllerError_TARGET_TAUGHT_WITH_ANOTHER_ROBOT
Trio.UnifiedApi, 32
ControllerError_TARGET_TOOL_OFFSET_MISMATCH
Trio.UnifiedApi, 32
ControllerError_THE_SPECIFIED_ROBOT_CONFIGURATION_IS_NOT_AVAILABLE
Trio.UnifiedApi, 31
ControllerError_THEN_EXPECTED_IN_IF_ELSEIF
Trio.UnifiedApi, 25
ControllerError_THIS_COMMAND_CAN_BE_EXECUTED_ONLY_ONCE
Trio.UnifiedApi, 32
ControllerError_TO_EXPECTED_AFTER_FOR
Trio.UnifiedApi, 25
ControllerError_TOO_MANY_AXES_ACTIVATED
Trio.UnifiedApi, 31
ControllerError_TOO_MANY_CASE_OPTIONS
Trio.UnifiedApi, 31
ControllerError_TOO_MANY_CHARS_IN_STR
Trio.UnifiedApi, 32
ControllerError_TOO_MANY_COLLISION_OBJECTS_ACTIVE
Trio.UnifiedApi, 31
ControllerError_TOO_MANY_EXIT_STATEMENTS_WITHIN_SINGLE_LOOP
Trio.UnifiedApi, 30
ControllerError_TOO_MANY_FUNCTION_PARAMETERS
Trio.UnifiedApi, 30
ControllerError_TOO_MANY_FUNCTIONS_DEFINED
Trio.UnifiedApi, 29
ControllerError_TOO_MANY_ITEMS_DEFINED_FOR_BLOCK
Trio.UnifiedApi, 27
ControllerError_TOO_MANY_LABELS
Trio.UnifiedApi, 27
ControllerError_TOO_MANY_NESTED_FOR_NEXT
Trio.UnifiedApi, 25
ControllerError_TOO_MANY_NESTED_GOSUB
Trio.UnifiedApi, 26
ControllerError_TOO_MANY_NESTED_OPERANDS
Trio.UnifiedApi, 26
ControllerError_TOO_MANY_NESTED_STRUCTURES
Trio.UnifiedApi, 25
ControllerError_TOO_MANY_VARIABLES
Trio.UnifiedApi, 24
ControllerError_TOOL_CANNOT_BE_MODIFIED_OR_DELETED_WHILE_SELECTED
Trio.UnifiedApi, 30
ControllerError_TOOL_OFFSET_ID_NOT_DEFINED
Trio.UnifiedApi, 30
ControllerError_U_AND_V_ORIENTATION_INHIBITED_WHILE_WRIST_SINGULARITY
Trio.UnifiedApi, 31
ControllerError_UNDEFINED_VARIABLE_IN_DIM_INITIALISER
Trio.UnifiedApi, 31
ControllerError_UNEXPECTED_CHARACTERS_AFTER_LINE_CONTINU
Trio.UnifiedApi, 32
ControllerError_UNTIL_IDLE_LOADED_EXPECTED_AFTER_WAIT
Trio.UnifiedApi, 25
ControllerError_UNTIL_WITHOUT_PREVIOUS_REPEAT
Trio.UnifiedApi, 25
ControllerError_USER_DEFINED_ERROR
Trio.UnifiedApi, 32
ControllerError_VALUE_IS_INCORRECT
Trio.UnifiedApi, 26
ControllerError_VALUE_IS_READ_ONLY
Trio.UnifiedApi, 25
ControllerError_VARIABLE_DEFINED_OUTSIDE_INCLUDE_FILE
Trio.UnifiedApi, 27
ControllerError_VARIABLE_NOT_CREATED_USING_GLOBAL_CONSTA
Trio.UnifiedApi, 29
ControllerError_VARIABLE_NOT_DEFINED_USING_DIM_OR_GLOBAL_
Trio.UnifiedApi, 29
ControllerError_VARIABLE_NOT_DEFINED_USING_DIM_WITHIN_INCL
Trio.UnifiedApi, 29
ControllerError_VARIABLE_NOT_FOUND
Trio.UnifiedApi, 26
ControllerError_VARIABLES_NOT_PERMITTED_ON_COMMAND_LINE
Trio.UnifiedApi, 27
ControllerError_VR_LOCATION_WRITE_PROTECTED
Trio.UnifiedApi, 32
ControllerError_VR_X_OR_STRING_VARIABLE_EXPECTED
Trio.UnifiedApi, 26
ControllerError_WEND_WITHOUT_PREVIOUS_WHILE
Trio.UnifiedApi, 25
ControllerError_WILDCARD_LENGTH_MUST_BE_1
Trio.UnifiedApi, 28
CORNER_STATE_MACHINE
Trio.UnifiedApi, 23
CornerMode
Trio.UnifiedApi, 23
CornerState
Trio.UnifiedApi, 23
CreateConnectionFlex7
Trio.UnifiedApi.TrioConnectionFactory, 341
CreateConnectionPCMCAT
Trio.UnifiedApi.TrioConnectionFactory, 341
CreateConnectionTCP
Trio.UnifiedApi.TrioConnectionFactory, 341
current_pos
Trio.UnifiedApi.ProgressInfo, 338

DACout_ANALin
Trio.UnifiedApi, 19
DACout_BISSin
Trio.UnifiedApi, 20
DACout_ENDATin
Trio.UnifiedApi, 19
DACout_QUADin
Trio.UnifiedApi, 19
DACout_QUADin_HD
Trio.UnifiedApi, 21

DACout_QUADin_Zout
Trio.UnifiedApi, 20
DACout_SDFREQin
Trio.UnifiedApi, 20
DACout_SDin
Trio.UnifiedApi, 20
DACout_SDin_HD
Trio.UnifiedApi, 21
DACout_SSIin
Trio.UnifiedApi, 20
DACout_SSISLAVEin
Trio.UnifiedApi, 20
DACout_TAMAin
Trio.UnifiedApi, 19
DACPWMout_QUADin
Trio.UnifiedApi, 20
data_length
Trio.UnifiedApi.DataStreamPacket, 43
DATUM
Trio.UnifiedApi, 35
Datum
Trio.UnifiedApi.ITrioConnection, 80
DefPos
Trio.UnifiedApi.ITrioConnection, 80
DefPosMulti
Trio.UnifiedApi.ITrioConnection, 81
DELAY_POSITION_EVALUATION_TO_LOAD_TIME
Trio.UnifiedApi, 17
Delete
Trio.UnifiedApi.ITrioConnection, 81
DeleteFrameGroup
Trio.UnifiedApi.ITrioConnection, 81
Dir
Trio.UnifiedApi.ITrioConnection, 82
DIRECTION_MODE_3D
Trio.UnifiedApi, 17
DIRECTION_SHORTEST
Trio.UnifiedApi, 17
DISABLE_FE_ACCUMULATION_DURING_MOVE
Trio.UnifiedApi, 17
DISABLE_LOOKAHEAD
Trio.UnifiedApi, 17
Disconnected
Trio.UnifiedApi, 24
DLink_Comms
Trio.UnifiedApi, 19
DLink_Remote
Trio.UnifiedApi, 19
DownloadFile
Trio.UnifiedApi.ITrioConnection, 82
DPOS
Trio.UnifiedApi, 23, 39
DS402PosType1
Trio.UnifiedApi, 19
DS402PosType2
Trio.UnifiedApi, 19
DS402VelType1
Trio.UnifiedApi, 19
DS402VelType2
Trio.UnifiedApi, 19
DUAL_AXIS_AVERAGE_POS
Trio.UnifiedApi, 17
Dummy107
Trio.UnifiedApi, 21
Dummy108
Trio.UnifiedApi, 21
Dummy109
Trio.UnifiedApi, 21
Dummy122
Trio.UnifiedApi, 21
Dummy123
Trio.UnifiedApi, 21
Dummy124
Trio.UnifiedApi, 21
Dummy125
Trio.UnifiedApi, 21
Dummy126
Trio.UnifiedApi, 21
Dummy127
Trio.UnifiedApi, 21
Dummy128
Trio.UnifiedApi, 21
Dummy129
Trio.UnifiedApi, 21
Dummy15
Trio.UnifiedApi, 19
Dummy16
Trio.UnifiedApi, 19
Dummy17
Trio.UnifiedApi, 19
Dummy24
Trio.UnifiedApi, 19
Dummy25
Trio.UnifiedApi, 19
Dummy28
Trio.UnifiedApi, 19
Dummy29
Trio.UnifiedApi, 19
Dummy32
Trio.UnifiedApi, 19
Dummy33
Trio.UnifiedApi, 19
Dummy34
Trio.UnifiedApi, 19
Dummy38
Trio.UnifiedApi, 19
Dummy39
Trio.UnifiedApi, 19
Dummy96
Trio.UnifiedApi, 20
Dummy98
Trio.UnifiedApi, 21
Dummy99
Trio.UnifiedApi, 21

Editable
Trio.UnifiedApi, 34

Encoder
Trio.UnifiedApi, 19
EncoderOut
Trio.UnifiedApi, 19
EncoderRatio
Trio.UnifiedApi.ITrioConnection, 83
EncryptedBASIC
Trio.UnifiedApi, 34
EncryptedBASICibrary
Trio.UnifiedApi, 35
EncryptedIecLibrary
Trio.UnifiedApi, 35
EncryptedIecProgram
Trio.UnifiedApi, 35
EncryptedRealtimeBasicLibrary
Trio.UnifiedApi, 35
ENDATin
Trio.UnifiedApi, 20
Error
Trio.UnifiedApi, 33
ErrorCode
Trio.UnifiedApi, 23
Trio.UnifiedApi.TrioConnectionException, 340
ErrorStringGetter
Trio.UnifiedApi.TrioConnectionFactory, 342
Ethercat_CheckNumberOfSlaves
Trio.UnifiedApi.ITrioConnection, 83
Ethercat_CheckNumberOfSlaves_VR
Trio.UnifiedApi.ITrioConnection, 83
Ethercat_CheckSlaveOnline
Trio.UnifiedApi.ITrioConnection, 84
Ethercat_CheckSlaveOnline_VR
Trio.UnifiedApi.ITrioConnection, 84
Ethercat_CoRead
Trio.UnifiedApi.ITrioConnection, 85
Ethercat_CoReadAxis
Trio.UnifiedApi.ITrioConnection, 85
Ethercat_CoWrite_Value
Trio.UnifiedApi.ITrioConnection, 86, 87
Ethercat_CoWrite_VR
Trio.UnifiedApi.ITrioConnection, 88
Ethercat_CoWriteAxis_Value
Trio.UnifiedApi.ITrioConnection, 88, 89
Ethercat_CoWriteAxis_VR
Trio.UnifiedApi.ITrioConnection, 90
Ethercat_DisplayNetworkConfig
Trio.UnifiedApi.ITrioConnection, 90
Ethercat_GetEmergencyMsgControl
Trio.UnifiedApi.ITrioConnection, 91
Ethercat_GetEmergencyMsgControl_VR
Trio.UnifiedApi.ITrioConnection, 91
Ethercat_GetSlave_ESC_Register
Trio.UnifiedApi.ITrioConnection, 91
Ethercat_GetSlaveAddress
Trio.UnifiedApi.ITrioConnection, 92
Ethercat_GetSlaveAddress_VR
Trio.UnifiedApi.ITrioConnection, 92
Ethercat_GetSlaveAxis
Trio.UnifiedApi.ITrioConnection, 93
Ethercat_GetSlaveAxis_VR
Trio.UnifiedApi.ITrioConnection, 93
Ethercat_GetState
Trio.UnifiedApi.ITrioConnection, 95
Ethercat_GetState_VR
Trio.UnifiedApi.ITrioConnection, 95
Ethercat_GetStatusMaskAndTriggerPattern
Trio.UnifiedApi.ITrioConnection, 96
Ethercat_GetStatusMaskAndTriggerPattern_VR
Trio.UnifiedApi.ITrioConnection, 96
Ethercat_SendResetSequence
Trio.UnifiedApi.ITrioConnection, 97
Ethercat_SetDistrClockMaster
Trio.UnifiedApi.ITrioConnection, 98
Ethercat_SetDistrClockMaster_VR
Trio.UnifiedApi.ITrioConnection, 98
Ethercat_SetEmergencyMsgControl
Trio.UnifiedApi.ITrioConnection, 99
Ethercat_SetSlave_ESC_Register
Trio.UnifiedApi.ITrioConnection, 99
Ethercat_SetState
Trio.UnifiedApi.ITrioConnection, 101
Ethercat_SetStatusMaskAndTriggerPattern
Trio.UnifiedApi.ITrioConnection, 101
Ethercat_SetStatusMaskAndTriggerPattern_VR
Trio.UnifiedApi.ITrioConnection, 102
Ethercat_SetSync0StartTime
Trio.UnifiedApi.ITrioConnection, 102
Ethercat_SetSync0StartTime_VR
Trio.UnifiedApi.ITrioConnection, 103
Ethercat_StartNetwork
Trio.UnifiedApi.ITrioConnection, 103, 104
Ethercat_StopNetwork
Trio.UnifiedApi.ITrioConnection, 104
EthercatDummy5
Trio.UnifiedApi, 20
EthercatDummy6
Trio.UnifiedApi, 20
EthercatEncoderIn
Trio.UnifiedApi, 20
EthercatInAnalOut
Trio.UnifiedApi, 20
EthercatOpenSpeed
Trio.UnifiedApi, 20
EthercatPos
Trio.UnifiedApi, 20
EthercatPosTrio
Trio.UnifiedApi, 20
EthercatSpeed
Trio.UnifiedApi, 20
EthercatSpeedTrio
Trio.UnifiedApi, 20
EthercatStartOption
Trio.UnifiedApi, 32
EthercatState
Trio.UnifiedApi, 33
EthercatTorque

Trio.UnifiedApi, 20
EventType
Trio.UnifiedApi, 33
evt
Trio.UnifiedApi.ProgressInfo, 338
Ex
Trio.UnifiedApi.ITrioConnection, 105
Execute
Trio.UnifiedApi.ITrioConnection, 105
ExecuteWithResponse
Trio.UnifiedApi.ITrioConnection, 105, 106
ExpressStartup
Trio.UnifiedApi, 33

Fail
Trio.UnifiedApi, 24
FifoFile
Trio.UnifiedApi, 34
FileCallback
Trio.UnifiedApi, 40
FileFlags
Trio.UnifiedApi, 33
FileTransferOption
Trio.UnifiedApi, 34
FileType
Trio.UnifiedApi, 34
Finish
Trio.UnifiedApi, 38
First_Comms
Trio.UnifiedApi, 19
flags
Trio.UnifiedApi.FileInfo, 45
FLEX7
Trio.UnifiedApi, 22
FLEXLINK
Trio.UnifiedApi, 36
ForceRescan
Trio.UnifiedApi, 33
FORWARD
Trio.UnifiedApi, 35
Forward
Trio.UnifiedApi.ITrioConnection, 107
FORWARD_JOG
Trio.UnifiedApi, 35

G5Pos
Trio.UnifiedApi, 20
G5Speed
Trio.UnifiedApi, 20
G5Torque
Trio.UnifiedApi, 20
GCode
Trio.UnifiedApi, 34
GEARIN
Trio.UnifiedApi, 36
GetAIn
Trio.UnifiedApi.ITrioConnection, 107
GetAOut
Trio.UnifiedApi.ITrioConnection, 107
GetApiVersion
Trio.UnifiedApi.ITrioConnection, 108
GetApiVersionStr
Trio.UnifiedApi.ITrioConnection, 108
GetAxisParameter_ACCEL
Trio.UnifiedApi.ITrioConnection, 109
GetAxisParameter_ADDAX_AXIS
Trio.UnifiedApi.ITrioConnection, 109
GetAxisParameter_AFF_GAIN
Trio.UnifiedApi.ITrioConnection, 109
GetAxisParameter_ATYPE
Trio.UnifiedApi.ITrioConnection, 110
GetAxisParameter_AXIS_ACCEL
Trio.UnifiedApi.ITrioConnection, 110
GetAxisParameter_AXIS_ADDRESS
Trio.UnifiedApi.ITrioConnection, 111
GetAxisParameter_AXIS_DECEL
Trio.UnifiedApi.ITrioConnection, 111
GetAxisParameter_AXIS_DISPLAY
Trio.UnifiedApi.ITrioConnection, 111
GetAxisParameter_AXIS_DPOS
Trio.UnifiedApi.ITrioConnection, 112
GetAxisParameter_AXIS_ENABLE
Trio.UnifiedApi.ITrioConnection, 112
GetAxisParameter_AXIS_ENABLE_OVERRIDE
Trio.UnifiedApi.ITrioConnection, 113
GetAxisParameter_AXIS_ERROR_COUNT
Trio.UnifiedApi.ITrioConnection, 113
GetAxisParameter_AXIS_FS_LIMIT
Trio.UnifiedApi.ITrioConnection, 113
GetAxisParameter_AXIS_MAX_JOG_SPEED
Trio.UnifiedApi.ITrioConnection, 114
GetAxisParameter_AXIS_MODE
Trio.UnifiedApi.ITrioConnection, 114
GetAxisParameter_AXIS_RS_LIMIT
Trio.UnifiedApi.ITrioConnection, 115
GetAxisParameter_AXIS_SPEED
Trio.UnifiedApi.ITrioConnection, 115
GetAxisParameter_AXIS_UNITS
Trio.UnifiedApi.ITrioConnection, 116
GetAxisParameter_AXIS_Z_OUTPUT
Trio.UnifiedApi.ITrioConnection, 116
GetAxisParameter_AXISSTATUS
Trio.UnifiedApi.ITrioConnection, 116
GetAxisParameter_BACKLASH_DIST
Trio.UnifiedApi.ITrioConnection, 117
GetAxisParameter_CHANGE_DIR_LAST
Trio.UnifiedApi.ITrioConnection, 117
GetAxisParameter_CLOSE_WIN
Trio.UnifiedApi.ITrioConnection, 118
GetAxisParameter_CLUTCH_RATE
Trio.UnifiedApi.ITrioConnection, 118
GetAxisParameter_CORNER_MODE
Trio.UnifiedApi.ITrioConnection, 118
GetAxisParameter_CORNER_STATE
Trio.UnifiedApi.ITrioConnection, 119
GetAxisParameter_CREEP
Trio.UnifiedApi.ITrioConnection, 119

GetAxisParameter_D_GAIN
Trio.UnifiedApi.ITrioConnection, 120
GetAxisParameter_DAC
Trio.UnifiedApi.ITrioConnection, 120
GetAxisParameter_DAC_OUT
Trio.UnifiedApi.ITrioConnection, 121
GetAxisParameter_DAC_SCALE
Trio.UnifiedApi.ITrioConnection, 121
GetAxisParameter_DATUM_IN
Trio.UnifiedApi.ITrioConnection, 121
GetAxisParameter_DECEL
Trio.UnifiedApi.ITrioConnection, 122
GetAxisParameter_DECEL_ANGLE
Trio.UnifiedApi.ITrioConnection, 122
GetAxisParameter_DEMAND_EDGES
Trio.UnifiedApi.ITrioConnection, 123
GetAxisParameter_DEMAND_SPEED
Trio.UnifiedApi.ITrioConnection, 123
GetAxisParameter_DPOS
Trio.UnifiedApi.ITrioConnection, 123
GetAxisParameter_DRIVE_CONTROL
Trio.UnifiedApi.ITrioConnection, 124
GetAxisParameter_DRIVE_CONTROLWORD
Trio.UnifiedApi.ITrioConnection, 124
GetAxisParameter_DRIVE_CURRENT
Trio.UnifiedApi.ITrioConnection, 125
GetAxisParameter_DRIVE_CW_MODE
Trio.UnifiedApi.ITrioConnection, 125
GetAxisParameter_DRIVE_ENABLE
Trio.UnifiedApi.ITrioConnection, 126
GetAxisParameter_DRIVE_FE
Trio.UnifiedApi.ITrioConnection, 126
GetAxisParameter_DRIVE_FE_LIMIT
Trio.UnifiedApi.ITrioConnection, 126
GetAxisParameter_DRIVE_INDEX
Trio.UnifiedApi.ITrioConnection, 127
GetAxisParameter_DRIVE_INPUTS
Trio.UnifiedApi.ITrioConnection, 127
GetAxisParameter_DRIVE_MODE
Trio.UnifiedApi.ITrioConnection, 128
GetAxisParameter_DRIVE_MONITOR
Trio.UnifiedApi.ITrioConnection, 128
GetAxisParameter_DRIVE_NEG_TORQUE
Trio.UnifiedApi.ITrioConnection, 128
GetAxisParameter_DRIVE_PARAMETER
Trio.UnifiedApi.ITrioConnection, 129
GetAxisParameter_DRIVE_POS_TORQUE
Trio.UnifiedApi.ITrioConnection, 129
GetAxisParameter_DRIVE_PROFILE
Trio.UnifiedApi.ITrioConnection, 130
GetAxisParameter_DRIVE_SET_VAL
Trio.UnifiedApi.ITrioConnection, 130
GetAxisParameter_DRIVE_STATUS
Trio.UnifiedApi.ITrioConnection, 131
GetAxisParameter_DRIVE_TORQUE
Trio.UnifiedApi.ITrioConnection, 131
GetAxisParameter_DRIVE_TYPE
Trio.UnifiedApi.ITrioConnection, 131
GetAxisParameter_DRIVE_VALUE
Trio.UnifiedApi.ITrioConnection, 132
GetAxisParameter_DRIVE_VELOCITY
Trio.UnifiedApi.ITrioConnection, 132
GetAxisParameter_ENCODER
Trio.UnifiedApi.ITrioConnection, 133
GetAxisParameter_ENCODER_BITS
Trio.UnifiedApi.ITrioConnection, 133
GetAxisParameter_ENCODER_CONTROL
Trio.UnifiedApi.ITrioConnection, 133
GetAxisParameter_ENCODER_FILTER
Trio.UnifiedApi.ITrioConnection, 134
GetAxisParameter_ENCODER_ID
Trio.UnifiedApi.ITrioConnection, 134
GetAxisParameter_ENCODER_STATUS
Trio.UnifiedApi.ITrioConnection, 135
GetAxisParameter_ENCODER_TURNS
Trio.UnifiedApi.ITrioConnection, 135
GetAxisParameter_END_DIR_LAST
Trio.UnifiedApi.ITrioConnection, 136
GetAxisParameter_ENDMOVE
Trio.UnifiedApi.ITrioConnection, 136
GetAxisParameter_ENDMOVE_BUFFER
Trio.UnifiedApi.ITrioConnection, 136
GetAxisParameter_ENDMOVE_SPEED
Trio.UnifiedApi.ITrioConnection, 137
GetAxisParameter_ERRORMASK
Trio.UnifiedApi.ITrioConnection, 137
GetAxisParameter_FAST_JOG
Trio.UnifiedApi.ITrioConnection, 138
GetAxisParameter_FASTDEC
Trio.UnifiedApi.ITrioConnection, 138
GetAxisParameter_FE
Trio.UnifiedApi.ITrioConnection, 138
GetAxisParameter_FE_LATCH
Trio.UnifiedApi.ITrioConnection, 139
GetAxisParameter_FE_LIMIT
Trio.UnifiedApi.ITrioConnection, 139
GetAxisParameter_FE_LIMIT_MODE
Trio.UnifiedApi.ITrioConnection, 140
GetAxisParameter_FE_RANGE
Trio.UnifiedApi.ITrioConnection, 140
GetAxisParameter_FEED_OVERRIDE
Trio.UnifiedApi.ITrioConnection, 141
GetAxisParameter_FHOLD_IN
Trio.UnifiedApi.ITrioConnection, 141
GetAxisParameter_FHSPEED
Trio.UnifiedApi.ITrioConnection, 141
GetAxisParameter_FORCE_ACCEL
Trio.UnifiedApi.ITrioConnection, 142
GetAxisParameter_FORCE_DECEL
Trio.UnifiedApi.ITrioConnection, 142
GetAxisParameter_FORCE_DWELL
Trio.UnifiedApi.ITrioConnection, 143
GetAxisParameter_FORCE_JERK
Trio.UnifiedApi.ITrioConnection, 143
GetAxisParameter_FORCE_RAMP
Trio.UnifiedApi.ITrioConnection, 143

GetAxisParameter_FORCE_SPEED
Trio.UnifiedApi.ITrioConnection, 144
GetAxisParameter_FRAME
Trio.UnifiedApi.ITrioConnection, 144
GetAxisParameter_FRAME_REP_DIST
Trio.UnifiedApi.ITrioConnection, 145
GetAxisParameter_FRAME_SCALE
Trio.UnifiedApi.ITrioConnection, 145
GetAxisParameter_FS_LIMIT
Trio.UnifiedApi.ITrioConnection, 146
GetAxisParameter_FULL_SP_RADIUS
Trio.UnifiedApi.ITrioConnection, 146
GetAxisParameter_FWD_IN
Trio.UnifiedApi.ITrioConnection, 146
GetAxisParameter_FWD_JOG
Trio.UnifiedApi.ITrioConnection, 147
GetAxisParameter_FWD_START
Trio.UnifiedApi.ITrioConnection, 147
GetAxisParameter_I_GAIN
Trio.UnifiedApi.ITrioConnection, 148
GetAxisParameter_IDLE
Trio.UnifiedApi.ITrioConnection, 148
GetAxisParameter_IN_POS
Trio.UnifiedApi.ITrioConnection, 148
GetAxisParameter_IN_POS_DIST
Trio.UnifiedApi.ITrioConnection, 149
GetAxisParameter_IN_POS_SPEED
Trio.UnifiedApi.ITrioConnection, 149
GetAxisParameter_INTERP_FACTOR
Trio.UnifiedApi.ITrioConnection, 150
GetAxisParameter_INVERT_STEP
Trio.UnifiedApi.ITrioConnection, 150
GetAxisParameter_JERK
Trio.UnifiedApi.ITrioConnection, 151
GetAxisParameter_JOGSPEED
Trio.UnifiedApi.ITrioConnection, 151
GetAxisParameter_LINK_AXIS
Trio.UnifiedApi.ITrioConnection, 151
GetAxisParameter_LOADED
Trio.UnifiedApi.ITrioConnection, 152
GetAxisParameter_LOOKAHEAD_FACTOR
Trio.UnifiedApi.ITrioConnection, 152
GetAxisParameter_MARK
Trio.UnifiedApi.ITrioConnection, 153
GetAxisParameter_MARKB
Trio.UnifiedApi.ITrioConnection, 153
GetAxisParameter_MERGE
Trio.UnifiedApi.ITrioConnection, 153
GetAxisParameter_MOVE_COUNT
Trio.UnifiedApi.ITrioConnection, 154
GetAxisParameter_MOVE_PA
Trio.UnifiedApi.ITrioConnection, 154
GetAxisParameter_MOVE_PA_CONT
Trio.UnifiedApi.ITrioConnection, 155
GetAxisParameter_MOVE_PA_IDLE
Trio.UnifiedApi.ITrioConnection, 155
GetAxisParameter_MOVE_PB
Trio.UnifiedApi.ITrioConnection, 156
GetAxisParameter_MOVE_PB_CONT
Trio.UnifiedApi.ITrioConnection, 156
GetAxisParameter_MOVE_PB_IDLE
Trio.UnifiedApi.ITrioConnection, 156
GetAxisParameter_MOVELINK_MODIFY
Trio.UnifiedApi.ITrioConnection, 157
GetAxisParameter_MOVES_BUFFERED
Trio.UnifiedApi.ITrioConnection, 157
GetAxisParameter_MPOS
Trio.UnifiedApi.ITrioConnection, 158
GetAxisParameter_MSPEED
Trio.UnifiedApi.ITrioConnection, 158
GetAxisParameter_MSPEED_FILTER
Trio.UnifiedApi.ITrioConnection, 158
GetAxisParameter_MSPEEDF
Trio.UnifiedApi.ITrioConnection, 159
GetAxisParameter_MTYPE
Trio.UnifiedApi.ITrioConnection, 159
GetAxisParameter_NTYPE
Trio.UnifiedApi.ITrioConnection, 160
GetAxisParameter_OFFPOS
Trio.UnifiedApi.ITrioConnection, 160
GetAxisParameter_OPEN_WIN
Trio.UnifiedApi.ITrioConnection, 161
GetAxisParameter_OUTLIMIT
Trio.UnifiedApi.ITrioConnection, 161
GetAxisParameter_OV_GAIN
Trio.UnifiedApi.ITrioConnection, 161
GetAxisParameter_P_GAIN
Trio.UnifiedApi.ITrioConnection, 162
GetAxisParameter_POS_DELAY
Trio.UnifiedApi.ITrioConnection, 162
GetAxisParameter_POSI_SEQ_DELAY
Trio.UnifiedApi.ITrioConnection, 163
GetAxisParameter_POSI_SEQ_MODE
Trio.UnifiedApi.ITrioConnection, 163
GetAxisParameter_PP_STEP
Trio.UnifiedApi.ITrioConnection, 163
GetAxisParameter_PS_ENCODER
Trio.UnifiedApi.ITrioConnection, 164
GetAxisParameter_PWM_CYCLE
Trio.UnifiedApi.ITrioConnection, 164
GetAxisParameter_PWM_MARK
Trio.UnifiedApi.ITrioConnection, 165
GetAxisParameter_RAISE_ANGLE
Trio.UnifiedApi.ITrioConnection, 165
GetAxisParameter_REG_INPUTS
Trio.UnifiedApi.ITrioConnection, 166
GetAxisParameter_REG_POS
Trio.UnifiedApi.ITrioConnection, 166
GetAxisParameter_REG_POSB
Trio.UnifiedApi.ITrioConnection, 166
GetAxisParameter_REGIST_CONTROL
Trio.UnifiedApi.ITrioConnection, 167
GetAxisParameter_REGIST_DELAY
Trio.UnifiedApi.ITrioConnection, 167
GetAxisParameter_REGIST_SPEED
Trio.UnifiedApi.ITrioConnection, 168

GetAxisParameter_REGIST_SPEEDB
Trio.UnifiedApi.ITrioConnection, 168
GetAxisParameter_REMAIN
Trio.UnifiedApi.ITrioConnection, 168
GetAxisParameter_REP_DIST
Trio.UnifiedApi.ITrioConnection, 169
GetAxisParameter_REP_OPTION
Trio.UnifiedApi.ITrioConnection, 169
GetAxisParameter_REV_IN
Trio.UnifiedApi.ITrioConnection, 170
GetAxisParameter_REV_JOG
Trio.UnifiedApi.ITrioConnection, 170
GetAxisParameter_REV_START
Trio.UnifiedApi.ITrioConnection, 171
GetAxisParameter_ROBOT_DPOS
Trio.UnifiedApi.ITrioConnection, 171
GetAxisParameter_ROBOT_FE
Trio.UnifiedApi.ITrioConnection, 171
GetAxisParameter_ROBOT_FE_LATCH
Trio.UnifiedApi.ITrioConnection, 172
GetAxisParameter_ROBOT_FE_RANGE
Trio.UnifiedApi.ITrioConnection, 172
GetAxisParameter_ROBOT_FS_LIMIT
Trio.UnifiedApi.ITrioConnection, 173
GetAxisParameter_ROBOT_RS_LIMIT
Trio.UnifiedApi.ITrioConnection, 173
GetAxisParameter_ROBOT_SP_MODE
Trio.UnifiedApi.ITrioConnection, 173
GetAxisParameter_ROBOT_UNITS
Trio.UnifiedApi.ITrioConnection, 174
GetAxisParameter_ROBOTSTATUS
Trio.UnifiedApi.ITrioConnection, 174
GetAxisParameter_RS_LIMIT
Trio.UnifiedApi.ITrioConnection, 175
GetAxisParameter_S_REF
Trio.UnifiedApi.ITrioConnection, 175
GetAxisParameter_S_REF_OUT
Trio.UnifiedApi.ITrioConnection, 176
GetAxisParameter_SERVO
Trio.UnifiedApi.ITrioConnection, 176
GetAxisParameter_SLOT_NUMBER
Trio.UnifiedApi.ITrioConnection, 176
GetAxisParameter_SPEED
Trio.UnifiedApi.ITrioConnection, 177
GetAxisParameter_SRAMP
Trio.UnifiedApi.ITrioConnection, 177
GetAxisParameter_START_DIR_LAST
Trio.UnifiedApi.ITrioConnection, 178
GetAxisParameter_STARTMOVE_SPEED
Trio.UnifiedApi.ITrioConnection, 178
GetAxisParameter_STOP_ANGLE
Trio.UnifiedApi.ITrioConnection, 178
GetAxisParameter_SYNC_AXIS
Trio.UnifiedApi.ITrioConnection, 179
GetAxisParameter_SYNC_CONTROL
Trio.UnifiedApi.ITrioConnection, 179
GetAxisParameter_SYNC_DPOS
Trio.UnifiedApi.ITrioConnection, 180
GetAxisParameter_SYNC_DWELL
Trio.UnifiedApi.ITrioConnection, 180
GetAxisParameter_SYNC_FLIGHT
Trio.UnifiedApi.ITrioConnection, 181
GetAxisParameter_SYNC_PAUSE
Trio.UnifiedApi.ITrioConnection, 181
GetAxisParameter_SYNC_TIME
Trio.UnifiedApi.ITrioConnection, 181
GetAxisParameter_SYNC_TIMER
Trio.UnifiedApi.ITrioConnection, 182
GetAxisParameter_SYNC_WITHDRAW
Trio.UnifiedApi.ITrioConnection, 182
GetAxisParameter_T_REF
Trio.UnifiedApi.ITrioConnection, 183
GetAxisParameter_T_REF_OUT
Trio.UnifiedApi.ITrioConnection, 183
GetAxisParameter_TABLE_POINTER
Trio.UnifiedApi.ITrioConnection, 183
GetAxisParameter_TANG_DIRECTION
Trio.UnifiedApi.ITrioConnection, 184
GetAxisParameter_TCP_DPOS
Trio.UnifiedApi.ITrioConnection, 184
GetAxisParameter_TCP_FS_LIMIT
Trio.UnifiedApi.ITrioConnection, 185
GetAxisParameter_TCP_RS_LIMIT
Trio.UnifiedApi.ITrioConnection, 185
GetAxisParameter_TCP_UNITS
Trio.UnifiedApi.ITrioConnection, 186
GetAxisParameter_UNITS
Trio.UnifiedApi.ITrioConnection, 186
GetAxisParameter_VECTOR_BUFFERED
Trio.UnifiedApi.ITrioConnection, 186
GetAxisParameter_VERIFY
Trio.UnifiedApi.ITrioConnection, 187
GetAxisParameter_VFF_GAIN
Trio.UnifiedApi.ITrioConnection, 187
GetAxisParameter_VP_ACCEL
Trio.UnifiedApi.ITrioConnection, 188
GetAxisParameter_VP_JERK
Trio.UnifiedApi.ITrioConnection, 188
GetAxisParameter_VP_MODE
Trio.UnifiedApi.ITrioConnection, 188
GetAxisParameter_VP_POSITION
Trio.UnifiedApi.ITrioConnection, 189
GetAxisParameter_VP_SPEED
Trio.UnifiedApi.ITrioConnection, 189
GetAxisParameter_WORLD_ACCEL
Trio.UnifiedApi.ITrioConnection, 190
GetAxisParameter_WORLD_DECEL
Trio.UnifiedApi.ITrioConnection, 190
GetAxisParameter_WORLD_DPOS
Trio.UnifiedApi.ITrioConnection, 191
GetAxisParameter_WORLD_FS_LIMIT
Trio.UnifiedApi.ITrioConnection, 191
GetAxisParameter_WORLD_JOGSPEED
Trio.UnifiedApi.ITrioConnection, 191
GetAxisParameter_WORLD_RS_LIMIT
Trio.UnifiedApi.ITrioConnection, 192

GetAxisParameter_WORLD_SPEED
Trio.UnifiedApi.ITrioConnection, 192
GetAxisParameter_WORLD_UNITS
Trio.UnifiedApi.ITrioConnection, 193
GetDIn
Trio.UnifiedApi.ITrioConnection, 193
GetDIn32
Trio.UnifiedApi.ITrioConnection, 194
GetDOut
Trio.UnifiedApi.ITrioConnection, 194, 195
GetDOut32
Trio.UnifiedApi.ITrioConnection, 195
GetFirmwareVersionNumber
Trio.UnifiedApi.ITrioConnection, 195
GetFrameGroup
Trio.UnifiedApi.ITrioConnection, 196
GetInvertIN
Trio.UnifiedApi.ITrioConnection, 196
GetMultiTableValues
Trio.UnifiedApi.ITrioConnection, 197
GetMultiVrValues
Trio.UnifiedApi.ITrioConnection, 197
GetProcessParameter_ERROR_LINE
Trio.UnifiedApi.ITrioConnection, 198
GetProcessParameter_INDEVICE
Trio.UnifiedApi.ITrioConnection, 198
GetProcessParameter_OUTDEVICE
Trio.UnifiedApi.ITrioConnection, 198
GetProcessParameter_PMOVE
Trio.UnifiedApi.ITrioConnection, 199
GetProcessParameter_PROC_LINE
Trio.UnifiedApi.ITrioConnection, 199
GetProcessParameter_PROC_STATUS
Trio.UnifiedApi.ITrioConnection, 200
GetProcessParameter_RUN_ERROR
Trio.UnifiedApi.ITrioConnection, 200
GetProcessParameter_TICKS
Trio.UnifiedApi.ITrioConnection, 200
GetSlotParameter_SERIAL_NUMBER
Trio.UnifiedApi.ITrioConnection, 201
GetSystemParameter_ADDRESS
Trio.UnifiedApi.ITrioConnection, 201
GetSystemParameter_AIN0
Trio.UnifiedApi.ITrioConnection, 202
GetSystemParameter_AIN1
Trio.UnifiedApi.ITrioConnection, 202
GetSystemParameter_AIN2
Trio.UnifiedApi.ITrioConnection, 202
GetSystemParameter_AIN3
Trio.UnifiedApi.ITrioConnection, 203
GetSystemParameter_AINBI0
Trio.UnifiedApi.ITrioConnection, 203
GetSystemParameter_AINBI1
Trio.UnifiedApi.ITrioConnection, 203
GetSystemParameter_AINBI2
Trio.UnifiedApi.ITrioConnection, 203
GetSystemParameter_AINBI3
Trio.UnifiedApi.ITrioConnection, 204
GetSystemParameter_AOUT0
Trio.UnifiedApi.ITrioConnection, 204
GetSystemParameter_AOUT1
Trio.UnifiedApi.ITrioConnection, 204
GetSystemParameter_AOUT2
Trio.UnifiedApi.ITrioConnection, 205
GetSystemParameter_AOUT3
Trio.UnifiedApi.ITrioConnection, 205
GetSystemParameter_AUTO_ETHERCAT
Trio.UnifiedApi.ITrioConnection, 205
GetSystemParameter_AVERAGE_JITTER
Trio.UnifiedApi.ITrioConnection, 206
GetSystemParameter_CONTROL
Trio.UnifiedApi.ITrioConnection, 206
GetSystemParameter_CPU_EXCEPTIONS
Trio.UnifiedApi.ITrioConnection, 206
GetSystemParameter_CURRENT_JITTER
Trio.UnifiedApi.ITrioConnection, 206
GetSystemParameter_DISPLAY
Trio.UnifiedApi.ITrioConnection, 207
GetSystemParameter_ERROR_AXIS
Trio.UnifiedApi.ITrioConnection, 207
GetSystemParameter_FLASH_STATUS
Trio.UnifiedApi.ITrioConnection, 207
GetSystemParameter_FPU_EXCEPTIONS
Trio.UnifiedApi.ITrioConnection, 208
GetSystemParameter_HMI_PROC
Trio.UnifiedApi.ITrioConnection, 208
GetSystemParameter_INPUTS0
Trio.UnifiedApi.ITrioConnection, 208
GetSystemParameter_INPUTS1
Trio.UnifiedApi.ITrioConnection, 209
GetSystemParameter_IP_ADDRESS
Trio.UnifiedApi.ITrioConnection, 209
GetSystemParameter_IP_GATEWAY
Trio.UnifiedApi.ITrioConnection, 209
GetSystemParameter_IP_MAC
Trio.UnifiedApi.ITrioConnection, 209
GetSystemParameter_IP_MEMORY_CONFIG
Trio.UnifiedApi.ITrioConnection, 210
GetSystemParameter_IP_NETMASK
Trio.UnifiedApi.ITrioConnection, 210
GetSystemParameter_IP_PROTOCOL_CONFIG
Trio.UnifiedApi.ITrioConnection, 210
GetSystemParameter_IP_PROTOCOL_CTRL
Trio.UnifiedApi.ITrioConnection, 211
GetSystemParameter_IP_TCP_TIMEOUT
Trio.UnifiedApi.ITrioConnection, 211
GetSystemParameter_IP_TCP_TX_THRESHOLD
Trio.UnifiedApi.ITrioConnection, 211
GetSystemParameter_IP_TCP_TX_TIMEOUT
Trio.UnifiedApi.ITrioConnection, 212
GetSystemParameter_LAST_AXIS
Trio.UnifiedApi.ITrioConnection, 212
GetSystemParameter_LIMIT_BUFFERED
Trio.UnifiedApi.ITrioConnection, 212
GetSystemParameter_MAX_JITTER
Trio.UnifiedApi.ITrioConnection, 212

GetSystemParameter_MISSED_INTERRUPTS
Trio.UnifiedApi.ITrioConnection, 213
GetSystemParameter_MISSED_RECV_FRAMES
Trio.UnifiedApi.ITrioConnection, 213
GetSystemParameter_MISSED_SEND_FRAMES
Trio.UnifiedApi.ITrioConnection, 213
GetSystemParameter_MODBUS_TX_DELAY
Trio.UnifiedApi.ITrioConnection, 214
GetSystemParameter_MOTION_ERROR
Trio.UnifiedApi.ITrioConnection, 214
GetSystemParameter_MPE_MODE
Trio.UnifiedApi.ITrioConnection, 214
GetSystemParameter_N_ANA_IN
Trio.UnifiedApi.ITrioConnection, 215
GetSystemParameter_N_ANA_OUT
Trio.UnifiedApi.ITrioConnection, 215
GetSystemParameter_NIN
Trio.UnifiedApi.ITrioConnection, 215
GetSystemParameter_NIO
Trio.UnifiedApi.ITrioConnection, 215
GetSystemParameter_NOP
Trio.UnifiedApi.ITrioConnection, 216
GetSystemParameter_PCMCAT_PROC
Trio.UnifiedApi.ITrioConnection, 216
GetSystemParameter_PLC_CONFIG
Trio.UnifiedApi.ITrioConnection, 216
GetSystemParameter_PLC_ERROR
Trio.UnifiedApi.ITrioConnection, 217
GetSystemParameter_PLC_OVERFLOW
Trio.UnifiedApi.ITrioConnection, 217
GetSystemParameter_PLC_RUN
Trio.UnifiedApi.ITrioConnection, 217
GetSystemParameter_PROCNUMBER
Trio.UnifiedApi.ITrioConnection, 218
GetSystemParameter_REMOTE_PROC
Trio.UnifiedApi.ITrioConnection, 218
GetSystemParameter_SCHEDULE_TYPE
Trio.UnifiedApi.ITrioConnection, 218
GetSystemParameter_SCOPE_CYCLE_COUNT
Trio.UnifiedApi.ITrioConnection, 218
GetSystemParameter_SCOPE_DELAY
Trio.UnifiedApi.ITrioConnection, 219
GetSystemParameter_SCOPE_POS
Trio.UnifiedApi.ITrioConnection, 219
GetSystemParameter_SCOPE_TRIGGER_POS
Trio.UnifiedApi.ITrioConnection, 219
GetSystemParameter_SERIAL_NUMBER
Trio.UnifiedApi.ITrioConnection, 220
GetSystemParameter_SERVO_OFFSET
Trio.UnifiedApi.ITrioConnection, 220
GetSystemParameter_SERVO_PERIOD
Trio.UnifiedApi.ITrioConnection, 220
GetSystemParameter_SLOT0_TIME
Trio.UnifiedApi.ITrioConnection, 221
GetSystemParameter_SLOT1_TIME
Trio.UnifiedApi.ITrioConnection, 221
GetSystemParameter_SLOT2_TIME
Trio.UnifiedApi.ITrioConnection, 221
GetSystemParameter_SLOT3_TIME
Trio.UnifiedApi.ITrioConnection, 221
GetSystemParameter_SYSTEM_ERROR
Trio.UnifiedApi.ITrioConnection, 222
GetSystemParameter_SYSTEM_ERROR_MASK
Trio.UnifiedApi.ITrioConnection, 222
GetSystemParameter_SYSTEM_LOAD
Trio.UnifiedApi.ITrioConnection, 222
GetSystemParameter_SYSTEM_LOAD_MAX
Trio.UnifiedApi.ITrioConnection, 223
GetSystemParameter_TEXT_FILE_LOADER_PROC
Trio.UnifiedApi.ITrioConnection, 223
GetSystemParameter_TSIZE
Trio.UnifiedApi.ITrioConnection, 223
GetSystemParameter_UNIT_ERROR
Trio.UnifiedApi.ITrioConnection, 224
GetSystemParameter_VERSION
Trio.UnifiedApi.ITrioConnection, 224
GetSystemParameter_VRSIZE
Trio.UnifiedApi.ITrioConnection, 224
GetSystemParameter_WDOG
Trio.UnifiedApi.ITrioConnection, 224
GetTableValue
Trio.UnifiedApi.ITrioConnection, 225
GetTcpHostName
Trio.UnifiedApi.ITrioConnection, 225
GetTcpPort
Trio.UnifiedApi.ITrioConnection, 225
GetTcpTimeout
Trio.UnifiedApi.ITrioConnection, 226
GetVrString
Trio.UnifiedApi.ITrioConnection, 226, 227
GetVrStringLen
Trio.UnifiedApi.ITrioConnection, 227
GetVrValue
Trio.UnifiedApi.ITrioConnection, 228

HELD
Trio.UnifiedApi, 23
HighestMotionType
Trio.UnifiedApi, 19
HmiDesign
Trio.UnifiedApi, 35
HmiLibrary
Trio.UnifiedApi, 35

IDLE
Trio.UnifiedApi, 35
IecLibrary
Trio.UnifiedApi, 35
IecProgram
Trio.UnifiedApi, 34
Initial
Trio.UnifiedApi, 33
Integer16
Trio.UnifiedApi, 22
Integer32
Trio.UnifiedApi, 22
Integer64

Trio.UnifiedApi, 22
Integer8
Trio.UnifiedApi, 22
Invert
Trio.UnifiedApi, 39
INVERT_AXIS_DIRECTION
Trio.UnifiedApi, 17
IsConnected
Trio.UnifiedApi.ITrioConnection, 337
ITrioConnection.cs, 343
size_t, 353

Jacquard
Trio.UnifiedApi, 20

LinkAbsolute
Trio.UnifiedApi, 22
LinkForceStart
Trio.UnifiedApi, 22
LinkMark
Trio.UnifiedApi, 21
LinkMarkB
Trio.UnifiedApi, 22
LinkPattern
Trio.UnifiedApi, 22
LinkPositiveMove
Trio.UnifiedApi, 22
LinkR_Mark
Trio.UnifiedApi, 22
LinkRepeat
Trio.UnifiedApi, 22
LinkThreshold
Trio.UnifiedApi, 22
LOAD
Trio.UnifiedApi, 23
LOAD_ZERO_SPEED
Trio.UnifiedApi, 23
LowLevelError
Trio.UnifiedApi, 24

Mask
Trio.UnifiedApi, 39
MCConfig
Trio.UnifiedApi, 34
MechatroInv
Trio.UnifiedApi, 20
MechatroPos
Trio.UnifiedApi, 19
MechatroSpeed
Trio.UnifiedApi, 19
MechatroTorque
Trio.UnifiedApi, 19
Message
Trio.UnifiedApi, 33
Trio.UnifiedApi.TrioConnectionException, 340
MHELICAL
Trio.UnifiedApi, 35
MHELICALSP
Trio.UnifiedApi, 35
MONITOR_FE
Trio.UnifiedApi, 17
MotionType
Trio.UnifiedApi, 35
MOVE
Trio.UnifiedApi, 35
MOVEABS
Trio.UnifiedApi, 35
MoveAbs
Trio.UnifiedApi.ITrioConnection, 228
MoveAbsMulti
Trio.UnifiedApi.ITrioConnection, 228
MOVEABSSEQ
Trio.UnifiedApi, 36
MoveAbsSeq
Trio.UnifiedApi.ITrioConnection, 229
MOVEABSSEQSP
Trio.UnifiedApi, 36
MOVEABSSP
Trio.UnifiedApi, 35
MoveAbsSp
Trio.UnifiedApi.ITrioConnection, 229
MOVEABSSPCONT
Trio.UnifiedApi, 36
MoveAbsSpMulti
Trio.UnifiedApi.ITrioConnection, 230
MOVEB
Trio.UnifiedApi, 36
MOVEBLEND
Trio.UnifiedApi, 36
MOVEC
Trio.UnifiedApi, 36
MOVECIRC
Trio.UnifiedApi, 35
MoveCirc
Trio.UnifiedApi.ITrioConnection, 230
MOVECIRCSP
Trio.UnifiedApi, 35
MoveCircSp
Trio.UnifiedApi.ITrioConnection, 231
MOVECONTOUR
Trio.UnifiedApi, 36
MOVEDISPENSE
Trio.UnifiedApi, 36
MoveHelical
Trio.UnifiedApi.ITrioConnection, 232
MoveHelicalSp
Trio.UnifiedApi.ITrioConnection, 232
MOVEJ
Trio.UnifiedApi, 36
MOVEJREL
Trio.UnifiedApi, 36
MOVEL
Trio.UnifiedApi, 36
MOVELINK
Trio.UnifiedApi, 36
MoveLink
Trio.UnifiedApi.ITrioConnection, 233

MOVELREL
Trio.UnifiedApi, 36
MOVEMODIFY
Trio.UnifiedApi, 35
MoveModify
Trio.UnifiedApi.ITrioConnection, 235
MOVEP
Trio.UnifiedApi, 36
MOVEPICK
Trio.UnifiedApi, 36
MovePick
Trio.UnifiedApi.ITrioConnection, 235
MoveRel
Trio.UnifiedApi.ITrioConnection, 236
MoveRelMulti
Trio.UnifiedApi.ITrioConnection, 236
MoveRelSeq
Trio.UnifiedApi.ITrioConnection, 237
MoveRelSp
Trio.UnifiedApi.ITrioConnection, 237
MoveRelSpMulti
Trio.UnifiedApi.ITrioConnection, 238
MOVESEQ
Trio.UnifiedApi, 36
MOVESEQSP
Trio.UnifiedApi, 36
MOVESP
Trio.UnifiedApi, 35
MOVESPCONT
Trio.UnifiedApi, 36
MoveSpherical
Trio.UnifiedApi.ITrioConnection, 238
MoveSphericalSp
Trio.UnifiedApi.ITrioConnection, 239
MOVETANG
Trio.UnifiedApi, 36
MOVETEST
Trio.UnifiedApi, 36
MOVETURN
Trio.UnifiedApi, 36
MOVEU
Trio.UnifiedApi, 36
MPOS
Trio.UnifiedApi, 23
MSPHERICAL
Trio.UnifiedApi, 36
MSPHERICALSP
Trio.UnifiedApi, 36
MTRANSCIRC
Trio.UnifiedApi, 36
MTRANSCIRCSP
Trio.UnifiedApi, 36

NegativeDirection
Trio.UnifiedApi, 39
New
Trio.UnifiedApi.ITrioConnection, 240
NoError
Trio.UnifiedApi, 24
None
Trio.UnifiedApi, 18, 21, 33, 34, 39
NOT_USED_1
Trio.UnifiedApi, 17
NOT_USED_2
Trio.UnifiedApi, 17
NOT_USED_3
Trio.UnifiedApi, 17
NotSupported
Trio.UnifiedApi, 24

OpenConnection
Trio.UnifiedApi.ITrioConnection, 240
OPENMOVE_1
Trio.UnifiedApi, 36
OPENMOVE_2
Trio.UnifiedApi, 36
OPENMOVE_3
Trio.UnifiedApi, 36
OPENMOVE_4
Trio.UnifiedApi, 36
OPENMOVE_5
Trio.UnifiedApi, 36
OPENMOVE_BASE
Trio.UnifiedApi, 36
Operational
Trio.UnifiedApi, 33
OVERRIDE_FE_LIMITS
Trio.UnifiedApi, 17

packet_count
Trio.UnifiedApi.DataStreamPacket, 43
packet_number
Trio.UnifiedApi.DataStreamPacket, 44
PanasonicPos
Trio.UnifiedApi, 20
PanasonicSpeed
Trio.UnifiedApi, 20
PanasonicTorque
Trio.UnifiedApi, 20
ParamBlockType
Trio.UnifiedApi, 36
PAUSED
Trio.UnifiedApi, 38
PAUSING
Trio.UnifiedApi, 38
PCMCAT
Trio.UnifiedApi, 22
PICKLINK
Trio.UnifiedApi, 36
PLM32
Trio.UnifiedApi, 20
PLMvirtual
Trio.UnifiedApi, 21
POSPROFILE
Trio.UnifiedApi, 36
PreOprational
Trio.UnifiedApi, 33
PrintOutProgress

Trio.UnifiedApi, 33
ProcessStatus
Trio.UnifiedApi, 38
Profibus_Comms
Trio.UnifiedApi, 19
ProgramVariable
Trio.UnifiedApi, 38
Progress
Trio.UnifiedApi, 38
ProgressEvent
Trio.UnifiedApi, 38
ProgressHandlerCallback
Trio.UnifiedApi, 40
Project
Trio.UnifiedApi, 34
PSwitch
Trio.UnifiedApi, 19
Trio.UnifiedApi.ITrioConnection, 240
PSwitchMode
Trio.UnifiedApi, 38
PSwitchMulti
Trio.UnifiedApi.ITrioConnection, 241
PSwitchMultiMode
Trio.UnifiedApi, 39
PSwitchOff
Trio.UnifiedApi.ITrioConnection, 242
PWMout_SDFREQin
Trio.UnifiedApi, 19
PWMout_SDin
Trio.UnifiedApi, 19

QUADin
Trio.UnifiedApi, 20
QUADin_HD
Trio.UnifiedApi, 21
QUADin_Zout
Trio.UnifiedApi, 21
QUADout_BISSin_HD
Trio.UnifiedApi, 21
QUADout_ENDATin_HD
Trio.UnifiedApi, 21
QUADout_HD
Trio.UnifiedApi, 21
QUADout_QUADin_HD
Trio.UnifiedApi, 21
QUADout_SDin_HD
Trio.UnifiedApi, 21
QUADout_SSIin_HD
Trio.UnifiedApi, 21
QUADout_TAMAin_HD
Trio.UnifiedApi, 21
QUADout_VFFDACout
Trio.UnifiedApi, 20
QUADout_Zin
Trio.UnifiedApi, 20
QUADout_Zout
Trio.UnifiedApi, 19

RapidStop
Trio.UnifiedApi.ITrioConnection, 242
READY
Trio.UnifiedApi, 38
Real32
Trio.UnifiedApi, 22
Real64
Trio.UnifiedApi, 22
RealtimeBASICLibrary
Trio.UnifiedApi, 35
Recipe
Trio.UnifiedApi, 35
Remote_TokenBuffer
Trio.UnifiedApi, 24
Remote_UnknownToken
Trio.UnifiedApi, 24
Reopen
Trio.UnifiedApi, 24
RESERVED
Trio.UnifiedApi, 23
Resolver
Trio.UnifiedApi, 19
REVERSE
Trio.UnifiedApi, 35
Reverse
Trio.UnifiedApi.ITrioConnection, 242
REVERSE_JOG
Trio.UnifiedApi, 35
Robot
Trio.UnifiedApi, 35
RobotBasic
Trio.UnifiedApi, 35
RobotBasicLibrary
Trio.UnifiedApi, 35
RobotLibrary
Trio.UnifiedApi, 35
RobotStatus
Trio.UnifiedApi, 39
RS_ALIGNMENT_SINGULARITY
Trio.UnifiedApi, 39
RS_COLLIDED
Trio.UnifiedApi, 39
RS_DISALLOW_JOG
Trio.UnifiedApi, 39
RS_ELBOW_CLOSED_SINGULARITY
Trio.UnifiedApi, 39
RS_ELBOW_SINGULARITY
Trio.UnifiedApi, 39
RS_FE_EXCEEDS_LIMIT
Trio.UnifiedApi, 39
RS_FE_WARNING
Trio.UnifiedApi, 39
RS_MAX_SPEED_LIMIT
Trio.UnifiedApi, 39
RS_NOT_IN_POS
Trio.UnifiedApi, 39
RS_ROBOT_FS_LIMIT
Trio.UnifiedApi, 39
RS_ROBOT_RS_LIMIT

Trio.UnifiedApi, 39
RS_TCP_FS_LIMIT
Trio.UnifiedApi, 39
RS_TCP_RS_LIMIT
Trio.UnifiedApi, 39
RS_WORLD_FS_LIMIT
Trio.UnifiedApi, 39
RS_WORLD_RS_LIMIT
Trio.UnifiedApi, 39
RS_WRIST_SINGULARITY
Trio.UnifiedApi, 39

SafeOperational
Trio.UnifiedApi, 33
SC_NO_SYNC
Trio.UnifiedApi, 40
SC_PERFORMING_INTERP_MOVE
Trio.UnifiedApi, 40
SC_PERFORMING_RESYNC
Trio.UnifiedApi, 40
SC_PERFORMING_RESYNC_TO_USERFRAME
Trio.UnifiedApi, 40
SC_PERFORMING_SYNC
Trio.UnifiedApi, 40
SC_STARTING_INTERP_MOVE
Trio.UnifiedApi, 40
SC_STARTING_RESYNC
Trio.UnifiedApi, 40
SC_STARTING_RESYNC_TO_USERFRAME
Trio.UnifiedApi, 40
SC_STARTING_SYNC
Trio.UnifiedApi, 40
SC_STOPPING_SYNC
Trio.UnifiedApi, 40
SC_SYNCHRONISED
Trio.UnifiedApi, 40
SC_WAITING_TO_START
Trio.UnifiedApi, 40
SDFREQout_ANALin
Trio.UnifiedApi, 20
SDin
Trio.UnifiedApi, 20
SDout_BISSin_HD
Trio.UnifiedApi, 21
SDout_ENDATin_HD
Trio.UnifiedApi, 21
SDout_HD
Trio.UnifiedApi, 21
SDout_PWMout
Trio.UnifiedApi, 19
SDout_PWMout2
Trio.UnifiedApi, 19
SDout_QUADin_HD
Trio.UnifiedApi, 21
SDout_SDin_HD
Trio.UnifiedApi, 21
SDout_SSIin_HD
Trio.UnifiedApi, 21
SDout_TAMAin
Trio.UnifiedApi, 20
SDout_TAMAin_HD
Trio.UnifiedApi, 21
SDout_VFFDACout
Trio.UnifiedApi, 20
SDout_Zin
Trio.UnifiedApi, 20
SDout_Zout
Trio.UnifiedApi, 19
Select
Trio.UnifiedApi.ITrioConnection, 244
SercosEncoder
Trio.UnifiedApi, 20
SercosOpen
Trio.UnifiedApi, 20
SercosPos
Trio.UnifiedApi, 20
SercosPosReg
Trio.UnifiedApi, 20
SercosSpeed
Trio.UnifiedApi, 20
SercosSpeedReg
Trio.UnifiedApi, 20
SercosTorque
Trio.UnifiedApi, 20
Servo
Trio.UnifiedApi, 18
Servo2
Trio.UnifiedApi, 19
SetAOut
Trio.UnifiedApi.ITrioConnection, 244
SetAxisBase
Trio.UnifiedApi.ITrioConnection, 244
SetAxisParameter_ACCEL
Trio.UnifiedApi.ITrioConnection, 245
SetAxisParameter_AFF_GAIN
Trio.UnifiedApi.ITrioConnection, 245
SetAxisParameter_ATYPE
Trio.UnifiedApi.ITrioConnection, 246
SetAxisParameter_AXIS_ACCEL
Trio.UnifiedApi.ITrioConnection, 246
SetAxisParameter_AXIS_ADDRESS
Trio.UnifiedApi.ITrioConnection, 246
SetAxisParameter_AXIS_DECEL
Trio.UnifiedApi.ITrioConnection, 247
SetAxisParameter_AXIS_DISPLAY
Trio.UnifiedApi.ITrioConnection, 247
SetAxisParameter_AXIS_ENABLE
Trio.UnifiedApi.ITrioConnection, 247
SetAxisParameter_AXIS_ENABLE_OVERRIDE
Trio.UnifiedApi.ITrioConnection, 248
SetAxisParameter_AXIS_ERROR_COUNT
Trio.UnifiedApi.ITrioConnection, 248
SetAxisParameter_AXIS_FS_LIMIT
Trio.UnifiedApi.ITrioConnection, 249
SetAxisParameter_AXIS_MAX_JOG_SPEED
Trio.UnifiedApi.ITrioConnection, 249
SetAxisParameter_AXIS_MODE

Trio.UnifiedApi.ITrioConnection, 249
SetAxisParameter_AXIS_RS_LIMIT
Trio.UnifiedApi.ITrioConnection, 250
SetAxisParameter_AXIS_SPEED
Trio.UnifiedApi.ITrioConnection, 250
SetAxisParameter_AXIS_UNITS
Trio.UnifiedApi.ITrioConnection, 250
SetAxisParameter_AXIS_Z_OUTPUT
Trio.UnifiedApi.ITrioConnection, 251
SetAxisParameter_BACKLASH_DIST
Trio.UnifiedApi.ITrioConnection, 251
SetAxisParameter_CHANGE_DIR_LAST
Trio.UnifiedApi.ITrioConnection, 252
SetAxisParameter_CLOSE_WIN
Trio.UnifiedApi.ITrioConnection, 252
SetAxisParameter_CLUTCH_RATE
Trio.UnifiedApi.ITrioConnection, 252
SetAxisParameter_CORNER_MODE
Trio.UnifiedApi.ITrioConnection, 253
SetAxisParameter_CORNER_STATE
Trio.UnifiedApi.ITrioConnection, 253
SetAxisParameter_CREEP
Trio.UnifiedApi.ITrioConnection, 253
SetAxisParameter_D_GAIN
Trio.UnifiedApi.ITrioConnection, 254
SetAxisParameter_DAC
Trio.UnifiedApi.ITrioConnection, 254
SetAxisParameter_DAC_SCALE
Trio.UnifiedApi.ITrioConnection, 255
SetAxisParameter_DATUM_IN
Trio.UnifiedApi.ITrioConnection, 255
SetAxisParameter_DECEL
Trio.UnifiedApi.ITrioConnection, 255
SetAxisParameter_DECEL_ANGLE
Trio.UnifiedApi.ITrioConnection, 256
SetAxisParameter_DRIVE_CONTROL
Trio.UnifiedApi.ITrioConnection, 256
SetAxisParameter_DRIVE_CONTROLWORD
Trio.UnifiedApi.ITrioConnection, 256
SetAxisParameter_DRIVE_CW_MODE
Trio.UnifiedApi.ITrioConnection, 257
SetAxisParameter_DRIVE_ENABLE
Trio.UnifiedApi.ITrioConnection, 257
SetAxisParameter_DRIVE_FE_LIMIT
Trio.UnifiedApi.ITrioConnection, 258
SetAxisParameter_DRIVE_INDEX
Trio.UnifiedApi.ITrioConnection, 258
SetAxisParameter_DRIVE_MODE
Trio.UnifiedApi.ITrioConnection, 258
SetAxisParameter_DRIVE_NEG_TORQUE
Trio.UnifiedApi.ITrioConnection, 259
SetAxisParameter_DRIVE_PARAMETER
Trio.UnifiedApi.ITrioConnection, 259
SetAxisParameter_DRIVE_POS_TORQUE
Trio.UnifiedApi.ITrioConnection, 259
SetAxisParameter_DRIVE_PROFILE
Trio.UnifiedApi.ITrioConnection, 260
SetAxisParameter_DRIVE_SET_VAL
Trio.UnifiedApi.ITrioConnection, 260
SetAxisParameter_DRIVE_TYPE
Trio.UnifiedApi.ITrioConnection, 261
SetAxisParameter_DRIVE_VALUE
Trio.UnifiedApi.ITrioConnection, 261
SetAxisParameter_ENCODER_BITS
Trio.UnifiedApi.ITrioConnection, 261
SetAxisParameter_ENCODER_CONTROL
Trio.UnifiedApi.ITrioConnection, 262
SetAxisParameter_ENCODER_FILTER
Trio.UnifiedApi.ITrioConnection, 262
SetAxisParameter_ENCODER_ID
Trio.UnifiedApi.ITrioConnection, 262
SetAxisParameter_ENCODER_STATUS
Trio.UnifiedApi.ITrioConnection, 263
SetAxisParameter_ENCODER_TURNS
Trio.UnifiedApi.ITrioConnection, 263
SetAxisParameter_END_DIR_LAST
Trio.UnifiedApi.ITrioConnection, 264
SetAxisParameter_ENDMOVE
Trio.UnifiedApi.ITrioConnection, 264
SetAxisParameter_ENDMOVE_BUFFER
Trio.UnifiedApi.ITrioConnection, 264
SetAxisParameter_ENDMOVE_SPEED
Trio.UnifiedApi.ITrioConnection, 265
SetAxisParameter_ERRORMASK
Trio.UnifiedApi.ITrioConnection, 265
SetAxisParameter_FAST_JOG
Trio.UnifiedApi.ITrioConnection, 265
SetAxisParameter_FASTDEC
Trio.UnifiedApi.ITrioConnection, 266
SetAxisParameter_FE_LIMIT
Trio.UnifiedApi.ITrioConnection, 266
SetAxisParameter_FE_LIMIT_MODE
Trio.UnifiedApi.ITrioConnection, 267
SetAxisParameter_FE_RANGE
Trio.UnifiedApi.ITrioConnection, 267
SetAxisParameter_FEED_OVERRIDE
Trio.UnifiedApi.ITrioConnection, 267
SetAxisParameter_FHOLD_IN
Trio.UnifiedApi.ITrioConnection, 268
SetAxisParameter_FHSPEED
Trio.UnifiedApi.ITrioConnection, 268
SetAxisParameter_FORCE_ACCEL
Trio.UnifiedApi.ITrioConnection, 268
SetAxisParameter_FORCE_DECEL
Trio.UnifiedApi.ITrioConnection, 269
SetAxisParameter_FORCE_DWELL
Trio.UnifiedApi.ITrioConnection, 269
SetAxisParameter_FORCE_JERK
Trio.UnifiedApi.ITrioConnection, 270
SetAxisParameter_FORCE_RAMP
Trio.UnifiedApi.ITrioConnection, 270
SetAxisParameter_FORCE_SPEED
Trio.UnifiedApi.ITrioConnection, 270
SetAxisParameter_FRAME
Trio.UnifiedApi.ITrioConnection, 271
SetAxisParameter_FRAME_REP_DIST

Trio.UnifiedApi.ITrioConnection, 271
SetAxisParameter_FRAME_SCALE
Trio.UnifiedApi.ITrioConnection, 271
SetAxisParameter_FS_LIMIT
Trio.UnifiedApi.ITrioConnection, 272
SetAxisParameter_FULL_SP_RADIUS
Trio.UnifiedApi.ITrioConnection, 272
SetAxisParameter_FWD_IN
Trio.UnifiedApi.ITrioConnection, 273
SetAxisParameter_FWD_JOG
Trio.UnifiedApi.ITrioConnection, 273
SetAxisParameter_FWD_START
Trio.UnifiedApi.ITrioConnection, 273
SetAxisParameter_I_GAIN
Trio.UnifiedApi.ITrioConnection, 274
SetAxisParameter_IN_POS
Trio.UnifiedApi.ITrioConnection, 274
SetAxisParameter_IN_POS_DIST
Trio.UnifiedApi.ITrioConnection, 274
SetAxisParameter_IN_POS_SPEED
Trio.UnifiedApi.ITrioConnection, 275
SetAxisParameter_INTERP_FACTOR
Trio.UnifiedApi.ITrioConnection, 275
SetAxisParameter_INVERT_STEP
Trio.UnifiedApi.ITrioConnection, 276
SetAxisParameter_JERK
Trio.UnifiedApi.ITrioConnection, 276
SetAxisParameter_JOGSPEED
Trio.UnifiedApi.ITrioConnection, 276
SetAxisParameter_LOOKAHEAD_FACTOR
Trio.UnifiedApi.ITrioConnection, 277
SetAxisParameter_MERGE
Trio.UnifiedApi.ITrioConnection, 277
SetAxisParameter_MOVE_COUNT
Trio.UnifiedApi.ITrioConnection, 277
SetAxisParameter_MOVE_PA
Trio.UnifiedApi.ITrioConnection, 278
SetAxisParameter_MOVE_PA_CONT
Trio.UnifiedApi.ITrioConnection, 278
SetAxisParameter_MOVE_PA_IDLE
Trio.UnifiedApi.ITrioConnection, 279
SetAxisParameter_MOVE_PB
Trio.UnifiedApi.ITrioConnection, 279
SetAxisParameter_MOVE_PB_CONT
Trio.UnifiedApi.ITrioConnection, 279
SetAxisParameter_MOVE_PB_IDLE
Trio.UnifiedApi.ITrioConnection, 280
SetAxisParameter_MOVELINK_MODIFY
Trio.UnifiedApi.ITrioConnection, 280
SetAxisParameter_MSPEED_FILTER
Trio.UnifiedApi.ITrioConnection, 280
SetAxisParameter_OFFPOS
Trio.UnifiedApi.ITrioConnection, 281
SetAxisParameter_OPEN_WIN
Trio.UnifiedApi.ITrioConnection, 281
SetAxisParameter_OUTLIMIT
Trio.UnifiedApi.ITrioConnection, 282
SetAxisParameter_OV_GAIN
Trio.UnifiedApi.ITrioConnection, 282
SetAxisParameter_P_GAIN
Trio.UnifiedApi.ITrioConnection, 282
SetAxisParameter_POS_DELAY
Trio.UnifiedApi.ITrioConnection, 283
SetAxisParameter_POSI_SEQ_DELAY
Trio.UnifiedApi.ITrioConnection, 283
SetAxisParameter_POSI_SEQ_MODE
Trio.UnifiedApi.ITrioConnection, 283
SetAxisParameter_PP_STEP
Trio.UnifiedApi.ITrioConnection, 284
SetAxisParameter_PWM_CYCLE
Trio.UnifiedApi.ITrioConnection, 284
SetAxisParameter_PWM_MARK
Trio.UnifiedApi.ITrioConnection, 285
SetAxisParameter_RAISE_ANGLE
Trio.UnifiedApi.ITrioConnection, 285
SetAxisParameter_REG_INPUTS
Trio.UnifiedApi.ITrioConnection, 285
SetAxisParameter_REGIST_CONTROL
Trio.UnifiedApi.ITrioConnection, 286
SetAxisParameter_REGIST_DELAY
Trio.UnifiedApi.ITrioConnection, 286
SetAxisParameter_REP_DIST
Trio.UnifiedApi.ITrioConnection, 286
SetAxisParameter_REP_OPTION
Trio.UnifiedApi.ITrioConnection, 287
SetAxisParameter_REV_IN
Trio.UnifiedApi.ITrioConnection, 287
SetAxisParameter_REV_JOG
Trio.UnifiedApi.ITrioConnection, 288
SetAxisParameter_REV_START
Trio.UnifiedApi.ITrioConnection, 288
SetAxisParameter_ROBOT_FE_RANGE
Trio.UnifiedApi.ITrioConnection, 288
SetAxisParameter_ROBOT_FS_LIMIT
Trio.UnifiedApi.ITrioConnection, 289
SetAxisParameter_ROBOT_RS_LIMIT
Trio.UnifiedApi.ITrioConnection, 289
SetAxisParameter_ROBOT_SP_MODE
Trio.UnifiedApi.ITrioConnection, 289
SetAxisParameter_ROBOT_UNITS
Trio.UnifiedApi.ITrioConnection, 290
SetAxisParameter_RS_LIMIT
Trio.UnifiedApi.ITrioConnection, 290
SetAxisParameter_S_REF
Trio.UnifiedApi.ITrioConnection, 291
SetAxisParameter_S_REF_OUT
Trio.UnifiedApi.ITrioConnection, 291
SetAxisParameter_SERVO
Trio.UnifiedApi.ITrioConnection, 291
SetAxisParameter_SLOT_NUMBER
Trio.UnifiedApi.ITrioConnection, 292
SetAxisParameter_SPEED
Trio.UnifiedApi.ITrioConnection, 292
SetAxisParameter_SRAMP
Trio.UnifiedApi.ITrioConnection, 292
SetAxisParameter_START_DIR_LAST

Trio.UnifiedApi.ITrioConnection, 293
SetAxisParameter_STARTMOVE_SPEED
Trio.UnifiedApi.ITrioConnection, 293
SetAxisParameter_STOP_ANGLE
Trio.UnifiedApi.ITrioConnection, 294
SetAxisParameter_SYNC_AXIS
Trio.UnifiedApi.ITrioConnection, 294
SetAxisParameter_SYNC_CONTROL
Trio.UnifiedApi.ITrioConnection, 294
SetAxisParameter_SYNC_DPOS
Trio.UnifiedApi.ITrioConnection, 295
SetAxisParameter_SYNC_DWELL
Trio.UnifiedApi.ITrioConnection, 295
SetAxisParameter_SYNC_FLIGHT
Trio.UnifiedApi.ITrioConnection, 295
SetAxisParameter_SYNC_PAUSE
Trio.UnifiedApi.ITrioConnection, 296
SetAxisParameter_SYNC_TIME
Trio.UnifiedApi.ITrioConnection, 296
SetAxisParameter_SYNC_WITHDRAW
Trio.UnifiedApi.ITrioConnection, 297
SetAxisParameter_T_REF
Trio.UnifiedApi.ITrioConnection, 297
SetAxisParameter_T_REF_OUT
Trio.UnifiedApi.ITrioConnection, 297
SetAxisParameter_TABLE_POINTER
Trio.UnifiedApi.ITrioConnection, 298
SetAxisParameter_TANG_DIRECTION
Trio.UnifiedApi.ITrioConnection, 298
SetAxisParameter_TCP_FS_LIMIT
Trio.UnifiedApi.ITrioConnection, 298
SetAxisParameter_TCP_RS_LIMIT
Trio.UnifiedApi.ITrioConnection, 299
SetAxisParameter_TCP_UNITS
Trio.UnifiedApi.ITrioConnection, 299
SetAxisParameter_UNITS
Trio.UnifiedApi.ITrioConnection, 300
SetAxisParameter_VECTOR_BUFFERED
Trio.UnifiedApi.ITrioConnection, 300
SetAxisParameter_VERIFY
Trio.UnifiedApi.ITrioConnection, 300
SetAxisParameter_VFF_GAIN
Trio.UnifiedApi.ITrioConnection, 301
SetAxisParameter_VP_MODE
Trio.UnifiedApi.ITrioConnection, 301
SetAxisParameter_WORLD_ACCEL
Trio.UnifiedApi.ITrioConnection, 301
SetAxisParameter_WORLD_DECEL
Trio.UnifiedApi.ITrioConnection, 302
SetAxisParameter_WORLD_FS_LIMIT
Trio.UnifiedApi.ITrioConnection, 302
SetAxisParameter_WORLD_JOGSPEED
Trio.UnifiedApi.ITrioConnection, 303
SetAxisParameter_WORLD_RS_LIMIT
Trio.UnifiedApi.ITrioConnection, 303
SetAxisParameter_WORLD_SPEED
Trio.UnifiedApi.ITrioConnection, 303
SetAxisParameter_WORLD_UNITS
Trio.UnifiedApi.ITrioConnection, 304
SetDOut
Trio.UnifiedApi.ITrioConnection, 304
SetDOut32
Trio.UnifiedApi.ITrioConnection, 305
SetFrameGroup
Trio.UnifiedApi.ITrioConnection, 305
SetInvertIN
Trio.UnifiedApi.ITrioConnection, 306
SetMotion
Trio.UnifiedApi.ITrioConnection, 306, 307
SetMotionSp
Trio.UnifiedApi.ITrioConnection, 308–311
SetMultiAxisBase
Trio.UnifiedApi.ITrioConnection, 312
SetMultiTableValues
Trio.UnifiedApi.ITrioConnection, 312
SetMultiVrValues
Trio.UnifiedApi.ITrioConnection, 313
SetProcessParameter_INDEVICE
Trio.UnifiedApi.ITrioConnection, 313
SetProcessParameter_OUTDEVICE
Trio.UnifiedApi.ITrioConnection, 313
SetProcessParameter_TICKS
Trio.UnifiedApi.ITrioConnection, 314
SetSlotParameter_SERIAL_NUMBER
Trio.UnifiedApi.ITrioConnection, 314
SetSystemParameter_ADDRESS
Trio.UnifiedApi.ITrioConnection, 314
SetSystemParameter_AIN0
Trio.UnifiedApi.ITrioConnection, 315
SetSystemParameter_AIN1
Trio.UnifiedApi.ITrioConnection, 315
SetSystemParameter_AIN2
Trio.UnifiedApi.ITrioConnection, 315
SetSystemParameter_AIN3
Trio.UnifiedApi.ITrioConnection, 316
SetSystemParameter_AINBI0
Trio.UnifiedApi.ITrioConnection, 316
SetSystemParameter_AINBI1
Trio.UnifiedApi.ITrioConnection, 316
SetSystemParameter_AINBI2
Trio.UnifiedApi.ITrioConnection, 317
SetSystemParameter_AINBI3
Trio.UnifiedApi.ITrioConnection, 317
SetSystemParameter_AOUT0
Trio.UnifiedApi.ITrioConnection, 317
SetSystemParameter_AOUT1
Trio.UnifiedApi.ITrioConnection, 318
SetSystemParameter_AOUT2
Trio.UnifiedApi.ITrioConnection, 318
SetSystemParameter_AOUT3
Trio.UnifiedApi.ITrioConnection, 318
SetSystemParameter_AUTO_ETHERCAT
Trio.UnifiedApi.ITrioConnection, 319
SetSystemParameter_AVERAGE_JITTER
Trio.UnifiedApi.ITrioConnection, 319
SetSystemParameter_CPU_EXCEPTIONS

Trio.UnifiedApi.ITrioConnection, 319
SetSystemParameter_CURRENT_JITTER
Trio.UnifiedApi.ITrioConnection, 320
SetSystemParameter_DISPLAY
Trio.UnifiedApi.ITrioConnection, 320
SetSystemParameter_FPU_EXCEPTIONS
Trio.UnifiedApi.ITrioConnection, 320
SetSystemParameter_HMI_PROC
Trio.UnifiedApi.ITrioConnection, 321
SetSystemParameter_IP_ADDRESS
Trio.UnifiedApi.ITrioConnection, 321
SetSystemParameter_IP_GATEWAY
Trio.UnifiedApi.ITrioConnection, 321
SetSystemParameter_IP_MAC
Trio.UnifiedApi.ITrioConnection, 322
SetSystemParameter_IP_MEMORY_CONFIG
Trio.UnifiedApi.ITrioConnection, 322
SetSystemParameter_IP_NETMASK
Trio.UnifiedApi.ITrioConnection, 322
SetSystemParameter_IP_PROTOCOL_CONFIG
Trio.UnifiedApi.ITrioConnection, 323
SetSystemParameter_IP_PROTOCOL_CTRL
Trio.UnifiedApi.ITrioConnection, 323
SetSystemParameter_IP_TCP_TIMEOUT
Trio.UnifiedApi.ITrioConnection, 323
SetSystemParameter_IP_TCP_TX_THRESHOLD
Trio.UnifiedApi.ITrioConnection, 324
SetSystemParameter_IP_TCP_TX_TIMEOUT
Trio.UnifiedApi.ITrioConnection, 324
SetSystemParameter_LAST_AXIS
Trio.UnifiedApi.ITrioConnection, 324
SetSystemParameter_LIMIT_BUFFERED
Trio.UnifiedApi.ITrioConnection, 325
SetSystemParameter_MAX_JITTER
Trio.UnifiedApi.ITrioConnection, 325
SetSystemParameter_MISSED_INTERRUPTS
Trio.UnifiedApi.ITrioConnection, 325
SetSystemParameter_MISSED_RECV_FRAMES
Trio.UnifiedApi.ITrioConnection, 326
SetSystemParameter_MISSED_SEND_FRAMES
Trio.UnifiedApi.ITrioConnection, 326
SetSystemParameter_MODBUS_TX_DELAY
Trio.UnifiedApi.ITrioConnection, 326
SetSystemParameter_NIO
Trio.UnifiedApi.ITrioConnection, 327
SetSystemParameter_PCMCAT_PROC
Trio.UnifiedApi.ITrioConnection, 327
SetSystemParameter_PLC_CONFIG
Trio.UnifiedApi.ITrioConnection, 327
SetSystemParameter_REMOTE_PROC
Trio.UnifiedApi.ITrioConnection, 328
SetSystemParameter_SCHEDULE_TYPE
Trio.UnifiedApi.ITrioConnection, 328
SetSystemParameter_SCOPE_CYCLE_COUNT
Trio.UnifiedApi.ITrioConnection, 328
SetSystemParameter_SCOPE_DELAY
Trio.UnifiedApi.ITrioConnection, 329
SetSystemParameter_SCOPE_POS
Trio.UnifiedApi.ITrioConnection, 329
SetSystemParameter_SERIAL_NUMBER
Trio.UnifiedApi.ITrioConnection, 329
SetSystemParameter_SERVO_OFFSET
Trio.UnifiedApi.ITrioConnection, 330
SetSystemParameter_SERVO_PERIOD
Trio.UnifiedApi.ITrioConnection, 330
SetSystemParameter_SLOT0_TIME
Trio.UnifiedApi.ITrioConnection, 330
SetSystemParameter_SLOT1_TIME
Trio.UnifiedApi.ITrioConnection, 331
SetSystemParameter_SLOT2_TIME
Trio.UnifiedApi.ITrioConnection, 331
SetSystemParameter_SLOT3_TIME
Trio.UnifiedApi.ITrioConnection, 331
SetSystemParameter_SYSTEM_ERROR_MASK
Trio.UnifiedApi.ITrioConnection, 332
SetSystemParameter_SYSTEM_LOAD_MAX
Trio.UnifiedApi.ITrioConnection, 332
SetSystemParameter_TEXT_FILE_LOADER_PROC
Trio.UnifiedApi.ITrioConnection, 332
SetSystemParameter_WDOG
Trio.UnifiedApi.ITrioConnection, 333
SetTableValue
Trio.UnifiedApi.ITrioConnection, 333
SetTcpHostName
Trio.UnifiedApi.ITrioConnection, 333
SetTcpPort
Trio.UnifiedApi.ITrioConnection, 334
SetTcpTimeout
Trio.UnifiedApi.ITrioConnection, 334
SetVrString
Trio.UnifiedApi.ITrioConnection, 334
SetVrValue
Trio.UnifiedApi.ITrioConnection, 335
size
Trio.UnifiedApi.FileInfo, 45
size_t
ITrioConnection.cs, 353
SLEEPING
Trio.UnifiedApi, 38
SLM32
Trio.UnifiedApi, 20
SLMvirtual
Trio.UnifiedApi, 21
SocketRead
Trio.UnifiedApi, 24
SocketWrite
Trio.UnifiedApi, 24
SOFTWARE_LIMIT_USES_DPOS
Trio.UnifiedApi, 17
SSI
Trio.UnifiedApi, 19
SSIin
Trio.UnifiedApi, 20
SSIin_HD
Trio.UnifiedApi, 21
Start

Trio.UnifiedApi, 38
Stepper
Trio.UnifiedApi, 18
StepperEncoderInt
Trio.UnifiedApi, 19
STEPPING
Trio.UnifiedApi, 38
StepRatio
Trio.UnifiedApi.ITrioConnection, 335
STOPPED
Trio.UnifiedApi, 38
StopProgram
Trio.UnifiedApi.ITrioConnection, 336
StreamHandlerCallback
Trio.UnifiedApi, 40
SUSPEND
Trio.UnifiedApi, 38
SUSPENDED
Trio.UnifiedApi, 38
SyncControl
Trio.UnifiedApi, 40
Sys_Unknown
Trio.UnifiedApi, 24
SystemParameters
Trio.UnifiedApi, 38

TABLE
Trio.UnifiedApi, 38
TAMAin
Trio.UnifiedApi, 20
TAMAin_HD
Trio.UnifiedApi, 21
TCP
Trio.UnifiedApi, 22
TempFile
Trio.UnifiedApi, 34
TempTextFile
Trio.UnifiedApi, 35
TempVolatileFile
Trio.UnifiedApi, 35
TextFile
Trio.UnifiedApi, 34
Timeout
Trio.UnifiedApi, 24
TimerMode
Trio.UnifiedApi, 39
TokenTable
Trio.UnifiedApi, 24
total_length
Trio.UnifiedApi.DataStreamPacket, 44
Trio.UnifiedApi.ProgressInfo, 338
Trio, 9
Trio.UnifiedApi, 9
_ApiError, 24
_ApiLast, 24
_ControllerError, 24
_SystemError, 24
_Unknown, 24
__Size, 35

Abort, 24
ABORTED, 38
ACCPROFILE, 36
AFFDACout, 20
Alarms, 35
ALLOW_CONNECT_MOVES_IN_LIMIT, 17
ANALout_ANALin, 19
Api_Unknown, 24
AS_AXIS_FS_LIMIT, 18
AS_AXIS_RS_LIMIT, 18
AS_BISS_CLEAN_WARNING, 18
AS_CANCEL_BUFFERED, 18
AS_CANCEL_BUFFERED_DECEL_INTO_BUFFERED_MOVES, 18
AS_CANCEL_BUFFERED_DECEL_WITHIN_CURRENT_MOVE, 18
AS_CANCEL_CURRENT, 18
AS_CANCEL_CURRENT_AT_SPEED, 18
AS_CANCEL_LAST_BUFFERED, 18
AS_CANCEL_MOVEADD, 18
AS_COMMS_ERROR, 17
AS_DATUMING, 18
AS_ENCODER_OVER_I, 18
AS_FE_EXCEEDS_LIMIT, 18
AS_FE_WARNING, 17
AS_FEEDHOLD, 18
AS_FORWARD_SOFTWARE_LIMIT, 18
AS_IN_FORWARD_LIMIT, 18
AS_IN_REVERSE_LIMIT, 18
AS_KINEMATICS_1HOUR, 18
AS_LOOKAHEAD_OVERRIDE, 17
AS_MOVE_CANCEL_DONE, 18
AS_MOVEADD, 18
AS_OPEN_FRAME_ERROR, 18
AS_OVERRIDE_TO_ZERO, 18
AS_PSWITCH_FIFO_FULL, 18
AS_PSWITCH_FIFO_HALF_FULL, 18
AS_PSWITCH_FIFO_NOT_EMPTY, 18
AS_PULSE_OVER_SPEED, 18
AS_REMOTE_DRIVE_ERROR, 17
AS_REMOTE_NODE_ERROR2, 18
AS_REMOTE_NODE_STATUS_ERROR, 18
AS_REVERSE_SOFTWARE_LIMIT, 18
AS_SINGULARITY, 18
AS_VOLUME_LIMIT, 18
AS_VP_ERROR, 18
AS_WORLD_FS_LIMIT, 18
AS_WORLD_RS_LIMIT, 18
AUTO_BLEND_MODE_0_USING_CORNER_RADIUS, 23
AUTO_BLEND_MODE_1_USING_CORNER_RADIUS, 23
AUTO_RADIUS_SPEED_CONTROL, 23
AUTO_RADIUS_SPEED_CONTROL_CENTRIPETAL_MODE, 23
AUTO_RADIUS_SPEED_CONTROL_CENTRIPETAL_MODE_USE_ 23
AUTO_SPEED_CONTROL, 23

Autorun, 34
AxesDiff, 20
AxisMode, 17
AxisParam, 38
AxisStatus, 17
AxisType, 18
BadArgument, 24
BadResponse, 24
BASIC, 34
BASICLibrary, 35
Binary, 34
BISSin, 20
Boolean, 22
BufferTooSmall, 24
CAM, 35
CAMBOX, 35
CamBoxLinkOptions, 21
CAMIN, 36
CAMLINK, 36
CAN, 19
CAN_Comms, 19
CAN_Remote, 19
CANCEL_ADDAX_ON_LIMIT, 17
CANCEL_MTYPE_ON_MASTER_IF_ADDAX_HITS_LIM, 17
Co_ObjectType, 22
COMBINEAXES, 36
COMPENSATE_XY, 36
CompressDecompress, 34
Compression, 24
CompressOnly, 34
CONNECT, 35
ConnectionContext, 24
ConnectionEventHandler, 40
ConnectionModule, 24
ConnectionType, 22
ConnectMode, 23
CONNPATH, 36
ControllerError_APPEND_BLOCK_INCONSISTENT, 27
ControllerError_AS_EXPECTED, 27
ControllerError_ASSIGNMENT_EXPECTED, 24
ControllerError_ATYPE_MUST_BE_CONFIGURED_USING_MC_CONFIG_PARAMETER_DRIVE_MODE, 31
ControllerError_AXIS_LOCKED_TO_A_ROBOT, 31
ControllerError_BASE_AXIS_DOES_NOT_BELONG_TO_ANY_KINEMATIC_GROUP, 30
ControllerError_BASIC_TOKEN_DEFINITION_INCORRECT, 27
ControllerError_BIT_NUMBER_OUT_OF_RANGE, 29
ControllerError_CAM_TABLE_ID_NOT_DEFINED, 31
ControllerError_CANNOT_EDIT_PROGRAM, 26
ControllerError_CANNOT_MODIOFY_SPECIAL_PROGRAM, 25
ControllerError_CANNOT_NEST_ONE_LINE_IF, 26
ControllerError_CANNOT_RESET_WHEN_DRIVE_SERVO_ON, 26
ControllerError_CANNOT_SELECT_PROGRAM, 25
ControllerError_CANNOT_SPECIFY_PROGRAM_TYPE_FOR_AN_E 27
ControllerError_CASE_OR_END_CASE_WITHOUT_PREVIOUS_SE 31
ControllerError_CLOSE_ROUND_BRACKET_EXPECTED, 25
ControllerError_CLOSE_SQUARE_BRACKET_EXPECTED_IN_PRI 25
ControllerError_COLLISION_OBJECTS_CANNOT_BE_MODIFIED_ 31
ControllerError_COMMA_EXPECTED, 25
ControllerError_COMMAND_INHIBITED_ROBOT_MUST_BE_USED 31
ControllerError_COMMAND_IS_COMMAND_LINE_ONLY, 25
ControllerError_COMMAND_IS_RUNTIME_ONLY, 25
ControllerError_COMMAND_LINE_BROKEN_BY_ESC, 25
ControllerError_COMMAND_LINE_TOO_LONG, 31
ControllerError_COMMAND_NOT_ALLOWED_IN_FRAME, 31
ControllerError_COMMAND_NOT_ALLOWED_WHEN_RUNNING_M 26
ControllerError_COMMAND_NOT_ALLOWED_WITH_CURRENT_AT 28
ControllerError_COMMAND_NOT_ALLOWED_WITHIN_INCLUDE_F 27
ControllerError_COMMAND_NOT_PERMITTED_IN_BASIC_LIBRAR 31
ControllerError_COMMAND_NOT_PERMITTED_IN_FUNCTION, 30
ControllerError_COMMAND_NOT_RECOGNISED, 24
ControllerError_COMMAND_ONLY_ALLOWED_WHEN_RUNNING_ 28
ControllerError_COMMAND_OUT_OF_CONTEXT, 25
ControllerError_CONSTANT_VALUE_EXPECTED, 32
ControllerError_CONTINUE_WITHOUT_PREVIOUS_FOR_WHILE_ 30
ControllerError_CYCLIC_INCLUDE_DEPENDENCY, 30
ControllerError_DATA_TYPE_EXPECTED, 27
ControllerError_DEVICE_ERROR__COMMAND_NOT_SUPPORTED 27
ControllerError_DEVICE_ERROR__CRC_ERROR, 27
ControllerError_DEVICE_ERROR__ERROR_WRITING_TO_DEVICE 28

ControllerError_DEVICE_ERROR__INVALID_RESPONSE_FROM_DEVICE, 28
ControllerError_DEVICE_ERROR__TIMEOUT_WAITING_FOR_DEVICE, 27
ControllerError_DEVICE_ERROR_NO_SDCARD_INSERTED, 29
ControllerError_DEVICE_ERROR_SDCARD_WRITE_PROTECTED, 29
ControllerError_DIRECTORY_ALREADY_LOCKED, 26
ControllerError_DIRECTORY_IS_LOCKED, 26
ControllerError_DIRECTORY_NOT_LOCKED, 26
ControllerError_DIRECTORY_STRUCTURE_INVALID, 26
ControllerError_DISK_ERROR__FORMAT_MUST_BE_FAT32, 29
ControllerError_DISK_ERROR__INVALID_BOOT_SECTOR, 28
ControllerError_DISK_ERROR__INVALID_MBR, 28
ControllerError_DISK_ERROR__INVALID_SECTOR_CLUSTER_REFERENCE, 28
ControllerError_DIVISION_BY_ZERO, 24
ControllerError_DOWNLOAD_ABORT_OR_TIMEOUT, 27
ControllerError_DUPLICATE_AXES_NOT_PERMITTED, 26
ControllerError_DUPLICATE_IDENTIFIER, 25
ControllerError_DUPLICATE_NAME_USED_IN_LABEL_AND_VARIABLE, 30
ControllerError_DUPLICATE_VR_INDEX, 32
ControllerError_ELSE_ELSEIF_ENDIF_WITHOUT_PREVIOUS_IF, 25
ControllerError_ENCRYPTION_AUTHORITY_NOT_FOUND, 31
ControllerError_END_OF_FILE, 28
ControllerError_ENDFUNC_WITHOUT_PREVIOUS_FUNCTION, 29
ControllerError_ETHERCAT_AXES_CANNOT_BE_SET_USING_ATYPE_PLEASE_USE_DRIVE_MODE, 29
ControllerError_EVALUATION_ERROR, 26
ControllerError_EXIT_WITHOUT_PREVIOUS_FOR_WHILE_OR_REPEAT, 30
ControllerError_EXTRA_CHARS_AT_EOLN, 24
ControllerError_FAILED_ATTEMPT_1, 29
ControllerError_FAILED_ATTEMPT_2, 29
ControllerError_FAILED_OPEN_CONNECTION, 29
ControllerError_FAILED_TOO_MANY_TIMES_PLEASE_RESET, 30
ControllerError_FEATURES_ENABLED_DO_NOT_ALLOW_ATYPE_CHANGE, 26
ControllerError_FILE_ALREADY_OPEN, 28
ControllerError_FILE_ERROR__DIRECTORY_FULL, 28
ControllerError_FILE_ERROR__DISK_FULL, 28
ControllerError_FILE_ERROR__FILE_NOT_FOUND, 28
ControllerError_FILE_ERROR__FILENAME_ALREADY_EXISTS, 28
ControllerError_FILE_ERROR__INVALID_FILENAME, 28
ControllerError_FILE_NOT_OPEN, 28
ControllerError_FIRMWARE_ERROR__CANNOT_REFERENCE_DA
28,
ControllerError_FIRMWARE_ERROR__DEVICE_IN_USE, 27
ControllerError_FIXED_INTEGER_OR_ELSE_EXPECTED, 31
ControllerError_FLASH_ERASE_ERROR, 25
ControllerError_FLASH_PROGRAMMING_ERROR, 24
ControllerError_FLASH_STICK_BLANK, 26
ControllerError_FLASH_STICK_NOT_AVAILABLE_ON_THIS_CONT
27
ControllerError_FLASH_VALUE_CANNOT_BE_SET, 26
ControllerError_FOR_EXPECTED, 28
ControllerError_FORWARD_TRANSFORM_FUNCTION_NOT_DEFI
32
ControllerError_FRAME_NOT_SUPPORTED_BY_RPS, 32
ControllerError_FUNCTION_EXPECTED, 31
ControllerError_FUNCTION_MUST_BE_DEFINED_WITHIN_A_BAS
29
ControllerError_FUNCTION_NAME_EXPECTED, 29
ControllerError_GLOBAL_DEFINITION_IS_DIFFERENT, 27
ControllerError_GOTO_GOSUB_EXEC_EXPECTED, 25
ControllerError_HASH_EXPECTED, 28
ControllerError_ID_NOT_DEFINED, 30
ControllerError_IDENTIFIER_EXPECTED, 25
ControllerError_IEC_ERROR__INVALID_VARIABLE_ACCESS, 28
ControllerError_IEC_FB_INSTANCE_INDEX_OUT_OF_RANGE, 29
ControllerError_IEC_RUNTIME_LIMITED_TO_1_HOUR_FEC_21, 29
ControllerError_IEC_TABLE_BINDING_INDEX_OUT_OF_RANGE, 29
ControllerError_IEC_TASK_MEMORY_ALLOCATION_ERROR, 29
ControllerError_IEC_VR_BINDING_INDEX_OUT_OF_RANGE, 29
ControllerError_INCLUDE_FILE_MUST_CONTAIN_DEFCONST_DI
32
ControllerError_INCOMPATIBLE_ARRAY_DIMENSIONS, 29
ControllerError_INCORRECT_SYMBOL_TYPE, 27
ControllerError_INPUT__OUTPUT__APPEND__FIFO_READ__FIFO
28
ControllerError_INVALID_ARRAY_DIMENSION, 32

ControllerError_INVALID_COFF_IMAGE_FILE, 27
ControllerError_INVALID_FLOATING_POINT_OPERATION, 28
ControllerError_INVALID_FORMAT_SPECIFIER, 32
ControllerError_INVALID_FPGA_VERSION, 29
ControllerError_INVALID_FRAME_NUMBER, 32
ControllerError_INVALID_IO_CHANNEL, 26
ControllerError_INVALID_LINE_NUMBER, 26
ControllerError_INVALID_MIX_OF_DATA_TYPES, 28
ControllerError_INVALID_MOVEC_INPUT, 31
ControllerError_INVALID_MOVEOPTION, 32
ControllerError_INVALID_MSPHERICAL_INPUT, 27
ControllerError_INVALID_OPEN_ARCH_FUNC_DEFINITION, 32
ControllerError_INVALID_PROGRAM_NAME, 27
ControllerError_INVALID_PROGRAM_TYPE, 27
ControllerError_INVALID_RANGE_SPECIFIED, 27
ControllerError_INVALID_STARTUP_CONFIGURATION_COMMAND, 28
ControllerError_INVALID_STORAGE_AREA, 28
ControllerError_INVALID_SYSTEM_CODE_WRONG_CONTROLLER, 28
ControllerError_INVALID_TRANSFER_TYPE, 24
ControllerError_INVALID_USE_OF_DOLAR_SIGN, 26
ControllerError_INVERSE_TRANSFORM_FUNCTION_NOT_DEFINED, 32
ControllerError_KEYWORD_FUNCTION_NOT_DEFINED, 32
ControllerError_KEYWORD_INACTIVE, 32
ControllerError_KEYWORD_SAME_AS_SYSTEM_WORD, 32
ControllerError_KINEMATICS_RUNTIME_LIMITED_TO_1_HOUR, 28
ControllerError_LABEL_EXPECTED, 25
ControllerError_LABEL_MUST_BE_AT_START_OF_LINE, 26
ControllerError_LABEL_NOT_FOUND, 26
ControllerError_LINE_MUST_BE_INSIDE_FUNCTION, 30
ControllerError_LINE_NUMBER_CANNOT_HAVE_DECIMAL_POINT, 26
ControllerError_LOCAL_TARGET_VALUE_HAS_NOT_BEEN_DEFINED, 31
ControllerError_MAC_EEPROM_IS_LOCKED, 28
ControllerError_MASTER_ERROR, 27
ControllerError_MATRIX_IS_SINGULAR, 29
ControllerError_MAX_OPEN_KEYWORDS_REACHED, 32
ControllerError_MAX_OPEN_MOVES_REACHED, 32
ControllerError_MAXIMUM_NUMBER_OF_PROGRAM_INSTANCES_ACTIVE, 26
ControllerError_MCCONFIG_CHANGED_EX_REQUIRED, 29
ControllerError_MISSING_ENDFUNC, 29
ControllerError_MODIFIER_NOT_ALLOWED, 25
ControllerError_MORE_THAN_ONE_PROGRAM_RUNNING_WHEN, 27
ControllerError_MOVE_TRIGGER_NOT_SPECIFIED, 32
ControllerError_MOVE_TRIGGER_RESPONSE_NOT_SPECIFIED, 32
ControllerError_MULTIPLE_MOVE_TRIGGERS_SPECIFIED, 31
ControllerError_NETWORK_PROTOCOL_ERROR, 27
ControllerError_NETWORK_TIMEOUT, 27
ControllerError_NEXT_WITHOUT_FOR, 25
ControllerError_NO_CODE_AVAILABLE_FOR_PROGRAM, 25
ControllerError_NO_FREE_ENCRYPTION_AUTHORITY, 31
ControllerError_NO_MORE_PROGRAMS_AVAILABLE, 25
ControllerError_NO_PROCESS_AVAILABLE, 25
ControllerError_NO_PROGRAM_SELECTED, 25
ControllerError_NO_RESPONSE_FROM_CAN_MODULE, 29
ControllerError_NONE, 24
ControllerError_NUMBER_EXPECTED, 27
ControllerError_NUMERICAL_ERROR__INF_USED, 28
ControllerError_NUMERICAL_ERROR__NAN_USED, 28
ControllerError_NUMERICAL_ERROR__SUBNORMAL_VALUE_USED, 28
ControllerError_OBJECT_FRAME_ALREADY_ATTACHED, 30
ControllerError_OBJECT_FRAME_ALREADY_SYNCED, 30
ControllerError_OBJECT_FRAME_CANNOT_BE_MODIFIED_OR_D, 30
ControllerError_OBJECT_FRAME_HAS_NOT_BEEN_ATTACHED, 30
ControllerError_OBJECT_FRAME_ID_NOT_DEFINED, 30
ControllerError_OPEN_BRACKET_EXPECTED, 27
ControllerError_OPEN_MOVE_FUNCTIONS_NOT_DEFINED, 32
ControllerError_OPERAND_EXPECTED, 24
ControllerError_OPERATION_CANNOT_BE_PERFORMED_UNTIL_, 30
ControllerError_OUT_OF_MEMORY, 25
ControllerError_PARAMETER_CAN_ONLY_BE_CHANGED_USING_, 29
ControllerError_PARAMETER_CAN_ONLY_BE_CHANGED_USING_, 29
ControllerError_PARAMETER_EXPECTED, 27
ControllerError_PARAMETER_NAME_EXPECTED, 29
ControllerError_PARAMETER_OUT_OF_RANGE,

25
ControllerError_PLC_TYPE_IS_INVALID, 26
ControllerError_PROCESS_ALREADY_SELECTED, 26
ControllerError_PROCESS_IN_USE, 32
ControllerError_PROGRAM_ALREADY_EXISTS, 26
ControllerError_PROGRAM_CORRUPT, 27
ControllerError_PROGRAM_ENCRYPTED, 27
ControllerError_PROGRAM_FAILED_TO_RUN, 30
ControllerError_PROGRAM_IS_LOCKED, 25
ControllerError_PROGRAM_IS_NOT_EXECUTABLE_TYPE, 29
ControllerError_PROGRAM_IS_STOPPED, 25
ControllerError_PROGRAM_IS_STOPPED_HALT_FORCED, 29
ControllerError_PROGRAM_NOT_FOUND, 25
ControllerError_PROGRAM_NOT_PAUSED, 26
ControllerError_PROGRAM_NOT_PAUSED_ON_THIS_PROCESS, 26
ControllerError_PROGRAM_NOT_RUNNING, 26
ControllerError_PROGRAM_NOT_RUNNING_ON_THIS_PROCESS, 26
ControllerError_PROGRAMS_RUNNING, 25
ControllerError_QUOTES_EXPECTED, 24
ControllerError_REMOTE_AXIS_IS_IN_USE, 25
ControllerError_RESERVED_KEYWORD_NOT_AVAILABLE_ON_THIS_CONTROLLER, 26
ControllerError_RETURN_STATEMENT_EXPECTED_BEFORE_END_FUNCTION, 30
ControllerError_RETURN_WITHOUT_GOSUB, 26
ControllerError_ROBOT_CONFIGURATION_CANNOT_BE_CHANGED_WHILE_MOVEL, 30
ControllerError_ROBOT_CONTROLLED_BY_ANOTHER_PROCESS, 31
ControllerError_ROBOT_FRAME_CANNOT_BE_MODIFIED_OR_DELETED_WHILE_SELECTED, 30
ControllerError_ROBOT_FRAME_ID_NOT_DEFINED, 30
ControllerError_ROBOT_MOVES_CANNOT_BE_PERFORMED_UNTIL_JOG_MANIPULATOR_IS_ACTIVATED, 30
ControllerError_ROBOT_NOT_DEFINED_USING_KINEMATIC_GROUP, 31
ControllerError_ROBOT_PROGRAM_STOPPED, 30
ControllerError_ROBOTICS_RUNTIME_LIMITED_TO_1_HOUR, 31
ControllerError_RPS_KERNEL_V2_NOT_ENABLED, 32
ControllerError_RUNNING_OF_RPS_PROGRAMS_INHIBITED, 31
ControllerError_SCOPE_PERIOD_SHOULD_EXCEED_NUMBER_OF_AIN_PARAMS, 26
ControllerError_SERIAL_NUMBER_MUST_BE_MINUS_1, 27
ControllerError_SINGULARITY, 30
ControllerError_SLAVE_ERROR, 27
ControllerError_STACK_OVERFLOW, 24
ControllerError_START_OF_EXPRESSION_EXPECTED, 25
ControllerError_STRING_EXCEEDS_PERMITTED_LENGTH, 26
ControllerError_STRING_EXPECTED, 27
ControllerError_STRUCTURE_FIELD_NAME_NOT_RECOGNISED, 30
ControllerError_STRUCTURE_NESTING_ERROR, 25
ControllerError_SYMBOL_IS_NOT_A_VARIABLE, 28
ControllerError_SYMBOL_TABLE_LOCKED, 27
ControllerError_SYNC_IN_PROCESS, 30
ControllerError_TABLE_INDEX_RANGE_ERROR, 26
ControllerError_TABLE_LOCATION_WRITE_PROTECTED, 31
ControllerError_TABLE_MEMORY_CONFLICT, 32
ControllerError_TARGET_OBJECT_FRAME_MISMATCH, 32
ControllerError_TARGET_POINT_NOT_ACTIVATED, 32
ControllerError_TARGET_POINT_OUT_OF_ROBOT_WORKING_AR
31
ControllerError_TARGET_ROBOT_CONFIG_MISMATCH, 32
ControllerError_TARGET_ROBOT_FRAME_MISMATCH, 32
ControllerError_TARGET_TAUGHT_WITH_ANOTHER_ROBOT, 32
ControllerError_TARGET_TOOL_OFFSET_MISMATCH, 32
ControllerError_THE_SPECIFIED_ROBOT_CONFIGURATION_IS_N
31
ControllerError_THEN_EXPECTED_IN_IF_ELSEIF, 25
ControllerError_THIS_COMMAND_CAN_BE_EXECUTED_ONLY_O
32
ControllerError_TO_EXPECTED_AFTER_FOR, 25
ControllerError_TOO_MANY_AXES_ASSIGNED, 31
ControllerError_TOO_MANY_CASE_OPTIONS, 31
ControllerError_TOO_MANY_CHARS_IN_STR, 32
ControllerError_TOO_MANY_COLLISION_OBJECTS_ACTIVE, 31
ControllerError_TOO_MANY_EXIT_STATEMENTS_WITHIN_SINGLE
30
ControllerError_TOO_MANY_FUNCTION_PARAMETERS, 30
ControllerError_TOO_MANY_FUNCTIONS_DEFINED, 28
ControllerError_TOO_MANY_ITEMS_DEFINED_FOR_BLOCK, 27
ControllerError_TOO_MANY_LABELS, 27
ControllerError_TOO_MANY_NESTED_FOR_NEXT, 25
ControllerError_TOO_MANY_NESTED_GOSUB,

26
ControllerError_TOO_MANY_NESTED_OPERANDS, 26
ControllerError_TOO_MANY_NESTED_STRUCTURES, 25
ControllerError_TOO_MANY_VARIABLES, 24
ControllerError_TOOL_CANNOT_BE_MODIFIED_OR_DELETED, 30
ControllerError_TOOL_OFFSET_ID_NOT_DEFINED, 30
ControllerError_U_AND_V_ORIENTATION_INHIBITED_WHILE_WRIST_SINGULARITY, 31
ControllerError_UNDEFINED_VARIABLE_IN_DIM_INITIALISER, 31
ControllerError_UNEXPECTED_CHARACTERS_AFTER_LINE_CONTINUATION, 32
ControllerError_UNTIL_IDLE_LOADED_EXPECTED_AFTER_WAIT, 25
ControllerError_UNTIL_WITHOUT_PREVIOUS_REPEAT, 25
ControllerError_USER_DEFINED_ERROR, 32
ControllerError_VALUE_IS_INCORRECT, 26
ControllerError_VALUE_IS_READ_ONLY, 25
ControllerError_VARIABLE_DEFINED_OUTSIDE_INCLUDE_FILE, 27
ControllerError_VARIABLE_NOT_CREATED_USING_GLOBAL_CONSTANT, 29
ControllerError_VARIABLE_NOT_DEFINED_USING_DIM_OR_GLOBAL, 29
ControllerError_VARIABLE_NOT_DEFINED_USING_DIM_WITHIN_INCLUDE_FILE, 29
ControllerError_VARIABLE_NOT_FOUND, 26
ControllerError_VARIABLES_NOT_PERMITTED_ON_COMMAND_LINE, 27
ControllerError_VR_LOCATION_WRITE_PROTECTED, 32
ControllerError_VR_X_OR_STRING_VARIABLE_EXPECTED, 26
ControllerError_WEND_WITHOUT_PREVIOUS_WHILE, 25
ControllerError_WILDCARD_LENGTH_MUST_BE_1, 28
CORNER_STATE_MACHINE, 23
CornerMode, 23
CornerState, 23
DACout_ANALin, 19
DACout_BISSin, 20
DACout_ENDATin, 19
DACout_QUADin, 19
DACout_QUADin_HD, 21
DACout_QUADin_Zout, 20
DACout_SDFREQin, 20
DACout_SDin, 20
DACout_SDin_HD, 21
DACout_SSIin, 20
DACout_SSISLAVEin, 20
DACout_TAMAin, 19
DACPWMout_QUADin, 20
DATUM, 35
DELAY_POSITION_EVALUATION_TO_LOAD_TIME, 17
DIRECTION_MODE_3D, 17
DIRECTION_SHORTEST, 17
DISABLE_FE_ACCUMULATION_DURING_MOVE, 17
DISABLE_FE_WHILE_SELECTED, 17
DISABLE_LOOKAHEAD, 17
Disconnected, 24
DLink_Comms, 19
DLink_Master, 19
DPOS, 23, 39
DS402PosType1, 19
DS402PosType2, 19
DS402VelType1, 19
DS402VelType2, 19
DRIVE_AXIS_AVERAGE_POS, 17
Dummy107, 21
Dummy108, 21
Dummy109, 21
Dummy122, 21
Dummy123, 21
Dummy124, 21
Dummy125, 21
Dummy126, 21
Dummy127, 21
Dummy128, 21
Dummy129, 21
Dummy15, 19
Dummy16, 19
Dummy17, 19
Dummy24, 19
Dummy25, 19
Dummy28, 19
Dummy29, 19
Dummy32, 19
Dummy33, 19
Dummy34, 19
Dummy38, 19
Dummy39, 19
Dummy96, 20
Dummy98, 21
Dummy99, 21
Editable, 34
Encoder, 19
EncoderOut, 19
EncryptedBASIC, 34
EncryptedBASICibrary, 35
EncryptedIecLibrary, 35
EncryptedIecProgram, 35
EncryptedRealtimeBasicLibrary, 35
ENDATin, 20
Error, 33
ErrorCode, 23
EthercatDummy5, 20
EthercatDummy6, 20
EthercatEncoderIn, 20
EthercatInAnalOut, 20

EthercatOpenSpeed, 20
EthercatPos, 20
EthercatPosTrio, 20
EthercatSpeed, 20
EthercatSpeedTrio, 20
EthercatStartOption, 32
EthercatState, 33
EthercatTorque, 20
EventType, 33
ExpressStartup, 33
Fail, 24
FifoFile, 34
FileCallback, 40
FileFlags, 33
FileTransferOption, 34
FileType, 34
Finish, 38
First_Comms, 19
FLEX7, 22
FLEXLINK, 36
ForceRescan, 33
FORWARD, 35
FORWARD_JOG, 35
G5Pos, 20
G5Speed, 20
G5Torque, 20
GCode, 34
GEARIN, 36
HELD, 23
HighestMotionType, 19
HmiDesign, 35
HmiLibrary, 35
IDLE, 35
IecLibrary, 35
IecProgram, 34
Initial, 33
Integer16, 22
Integer32, 22
Integer64, 22
Integer8, 22
Invert, 39
INVERT_AXIS_DIRECTION, 17
Jacquard, 20
LinkAbsolute, 22
LinkForceStart, 22
LinkMark, 21
LinkMarkB, 22
LinkPattern, 22
LinkPositiveMove, 22
LinkR_Mark, 22
LinkRepeat, 22
LinkThreshold, 22
LOAD, 23
LOAD_ZERO_SPEED, 23
LowLevelError, 24
Mask, 39
MCConfig, 34
MechatroInv, 20
MechatroPos, 19
MechatroSpeed, 19
MechatroTorque, 19
Message, 33
MHELICAL, 35
MHELICALSP, 35
MONITOR_FE, 17
MotionType, 35
MOVE, 35
MOVEABS, 35
MOVEABSSEQ, 36
MOVEABSSEQSP, 36
MOVEABSSP, 35
MOVEABSSPCONT, 36
MOVEB, 36
MOVEBLEND, 36
MOVEC, 36
MOVECIRC, 35
MOVECIRCSP, 35
MOVECONTOUR, 36
MOVEDISPENSE, 36
MOVEJ, 36
MOVEJREL, 36
MOVEL, 36
MOVELINK, 36
MOVELREL, 36
MOVEMODIFY, 35
MOVEP, 36
MOVEPICK, 36
MOVESEQ, 36
MOVESEQSP, 36
MOVESP, 35
MOVESPCONT, 36
MOVETANG, 36
MOVETEST, 36
MOVETURN, 36
MOVEU, 36
MPOS, 23
MSPHERICAL, 36
MSPHERICALSP, 36
MTRANSCIRC, 36
MTRANSCIRCSP, 36
NegativeDirection, 39
NoError, 24
None, 18, 21, 33, 34, 39
NOT_USED_1, 17
NOT_USED_2, 17
NOT_USED_3, 17
NotSupported, 24
OPENMOVE_1, 36
OPENMOVE_2, 36
OPENMOVE_3, 36
OPENMOVE_4, 36
OPENMOVE_5, 36
OPENMOVE_BASE, 36
Operational, 33
OVERRIDE_FE_LIMITS, 17
PanasonicPos, 20

PanasonicSpeed, 20
PanasonicTorque, 20
ParamBlockType, 36
PAUSED, 38
PAUSING, 38
PCMCAT, 22
PICKLINK, 36
PLM32, 20
PLMvirtual, 21
POSPROFILE, 36
PreOprational, 33
PrintOutProgress, 33
ProcessStatus, 38
Profibus_Comms, 19
ProgramVariable, 38
Progress, 38
ProgressEvent, 38
ProgressHandlerCallback, 40
Project, 34
PSwitch, 19
PSwitchMode, 38
PSwitchMultiMode, 39
PWMout_SDFREQin, 19
PWMout_SDin, 19
QUADin, 20
QUADin_HD, 21
QUADin_Zout, 21
QUADout_BISSin_HD, 21
QUADout_ENDATin_HD, 21
QUADout_HD, 21
QUADout_QUADin_HD, 21
QUADout_SDin_HD, 21
QUADout_SSIin_HD, 21
QUADout_TAMAin_HD, 21
QUADout_VFFDACout, 20
QUADout_Zin, 20
QUADout_Zout, 19
READY, 38
Real32, 22
Real64, 22
RealtimeBASICLibrary, 35
Recipe, 35
Remote_TokenBuffer, 24
Remote_UnknownToken, 24
Reopen, 24
RESERVED, 23
Resolver, 19
REVERSE, 35
REVERSE_JOG, 35
Robot, 35
RobotBasic, 35
RobotBasicLibrary, 35
RobotLibrary, 35
RobotStatus, 39
RS_ALIGNMENT_SINGULARITY, 39
RS_COLLIDED, 39
RS_DISALLOW_JOG, 39
RS_ELBOW_CLOSED_SINGULARITY, 39
RS_ELBOW_SINGULARITY, 39
RS_FE_EXCEEDS_LIMIT, 39
RS_FE_WARNING, 39
RS_MAX_SPEED_LIMIT, 39
RS_NOT_IN_POS, 39
RS_ROBOT_FS_LIMIT, 39
RS_ROBOT_RS_LIMIT, 39
RS_TCP_FS_LIMIT, 39
RS_TCP_RS_LIMIT, 39
RS_WORLD_FS_LIMIT, 39
RS_WORLD_RS_LIMIT, 39
RS_WRIST_SINGULARITY, 39
SafeOperational, 33
SC_NO_SYNC, 40
SC_PERFORMING_INTERP_MOVE, 40
SC_PERFORMING_RESYNC, 40
SC_PERFORMING_RESYNC_TO_USERFRAME, 40
SC_PERFORMING_SYNC, 40
SC_STARTING_INTERP_MOVE, 40
SC_STARTING_RESYNC, 40
SC_STARTING_RESYNC_TO_USERFRAME, 40
SC_STARTING_SYNC, 40
SC_STOPPING_SYNC, 40
SC_SYNCHRONISED, 40
SC_WAITING_TO_START, 40
SDFREQout_ANALin, 20
SDin, 20
SDout_BISSin_HD, 21
SDout_ENDATin_HD, 21
SDout_HD, 21
SDout_PWMout, 19
SDout_PWMout2, 19
SDout_QUADin_HD, 21
SDout_SDin_HD, 21
SDout_SSIin_HD, 21
SDout_TAMAin, 20
SDout_TAMAin_HD, 21
SDout_VFFDACout, 20
SDout_Zin, 20
SDout_Zout, 19
SercosEncoder, 20
SercosOpen, 20
SercosPos, 20
SercosPosReg, 20
SercosSpeed, 20
SercosSpeedReg, 20
SercosTorque, 20
Servo, 18
Servo2, 19
SLEEPING, 38
SLM32, 20
SLMvirtual, 21
SocketRead, 24
SocketWrite, 24
SOFTWARE_LIMIT_USES_DPOS, 17
SSI, 19
SSIin, 20

SSIin_HD, 21
Start, 38
Stepper, 18
StepperEncoderInt, 19
STEPPING, 38
STOPPED, 38
StreamHandlerCallback, 40
SUSPEND, 38
SUSPENDED, 38
SyncControl, 40
Sys_Unknown, 24
SystemParameters, 38
TABLE, 38
TAMAin, 20
TAMAin_HD, 21
TCP, 22
TempFile, 34
TempTextFile, 35
TempVolatileFile, 35
TextFile, 34
Timeout, 24
TimerMode, 39
TokenTable, 24
Undefined, 34
Unknown, 22
UnknownFloatingPointFormat, 24
Unsigned16, 22
Unsigned32, 22
Unsigned64, 22
Unsigned8, 22
UnsupportedController, 24
USB_Comms, 19
USE_FASTDEC_WHEN_CANCELLING_RPS_JOG, 17
USE_NON_SIGNEXTENDED_ANALOGUE_FEEDBACK, 17
VELPROFILE, 36
VersionMismatch, 24
VOut, 19
VR, 38
Warning, 33
WETCAM, 35
Yaskawa, 19
ZOut, 39
Trio.UnifiedApi.DataStreamPacket, 43
data_length, 43
packet_count, 43
packet_number, 44
total_length, 44
Trio.UnifiedApi.FileInfo, 44
autorun_process, 45
compiled_size, 45
flags, 45
size, 45
type, 45
Trio.UnifiedApi.ITrioConnection, 46
AddAx, 75
AddDac, 75, 76
Backlash, 76
BacklashDisable, 77
Cam, 77
CamBox, 77
Cancel, 78
ClearVariables, 79
CloseConnection, 79
Connect, 79
ConnectionEvent, 337
ConnectionType, 337
ConnPath, 79
Datum, 80
DefPos, 80
DefPosMulti, 81
Delete, 81
DeleteFrameGroup, 81
Dir, 82
DownloadFile, 82
EncoderRatio, 83
Ethercat_CheckNumberOfSlaves, 83
Ethercat_CheckNumberOfSlaves_VR, 83
Ethercat_CheckSlaveOnline, 84
Ethercat_CheckSlaveOnline_VR, 84
Ethercat_CoRead, 85
Ethercat_CoReadAxis, 85
Ethercat_CoWrite_Value, 86, 87
Ethercat_CoWrite_VR, 88
Ethercat_CoWriteAxis_Value, 88, 89
Ethercat_CoWriteAxis_VR, 90
Ethercat_DisplayNetworkConfig, 90
Ethercat_GetEmergencyMsgControl, 91
Ethercat_GetEmergencyMsgControl_VR, 91
Ethercat_GetSlave_ESC_Register, 91
Ethercat_GetSlaveAddress, 92
Ethercat_GetSlaveAddress_VR, 92
Ethercat_GetSlaveAxis, 93
Ethercat_GetSlaveAxis_VR, 93
Ethercat_GetState, 95
Ethercat_GetState_VR, 95
Ethercat_GetStatusMaskAndTriggerPattern, 96
Ethercat_GetStatusMaskAndTriggerPattern_VR, 96
Ethercat_SendResetSequence, 97
Ethercat_SetDistrClockMaster, 98
Ethercat_SetDistrClockMaster_VR, 98
Ethercat_SetEmergencyMsgControl, 99
Ethercat_SetSlave_ESC_Register, 99
Ethercat_SetState, 101
Ethercat_SetStatusMaskAndTriggerPattern, 101
Ethercat_SetStatusMaskAndTriggerPattern_VR, 102
Ethercat_SetSync0StartTime, 102
Ethercat_SetSync0StartTime_VR, 103
Ethercat_StartNetwork, 103, 104
Ethercat_StopNetwork, 104
Ex, 105
Execute, 105
ExecuteWithResponse, 105, 106

Forward, 107
GetAIn, 107
GetAOut, 107
GetApiVersion, 108
GetApiVersionStr, 108
GetAxisParameter_ACCEL, 109
GetAxisParameter_ADDAX_AXIS, 109
GetAxisParameter_AFF_GAIN, 109
GetAxisParameter_ATYPE, 110
GetAxisParameter_AXIS_ACCEL, 110
GetAxisParameter_AXIS_ADDRESS, 111
GetAxisParameter_AXIS_DECEL, 111
GetAxisParameter_AXIS_DISPLAY, 111
GetAxisParameter_AXIS_DPOS, 112
GetAxisParameter_AXIS_ENABLE, 112
GetAxisParameter_AXIS_ENABLE_OVERRIDE, 113
GetAxisParameter_AXIS_ERROR_COUNT, 113
GetAxisParameter_AXIS_FS_LIMIT, 113
GetAxisParameter_AXIS_MAX_JOG_SPEED, 114
GetAxisParameter_AXIS_MODE, 114
GetAxisParameter_AXIS_RS_LIMIT, 115
GetAxisParameter_AXIS_SPEED, 115
GetAxisParameter_AXIS_UNITS, 116
GetAxisParameter_AXIS_Z_OUTPUT, 116
GetAxisParameter_AXISSTATUS, 116
GetAxisParameter_BACKLASH_DIST, 117
GetAxisParameter_CHANGE_DIR_LAST, 117
GetAxisParameter_CLOSE_WIN, 118
GetAxisParameter_CLUTCH_RATE, 118
GetAxisParameter_CORNER_MODE, 118
GetAxisParameter_CORNER_STATE, 119
GetAxisParameter_CREEP, 119
GetAxisParameter_D_GAIN, 120
GetAxisParameter_DAC, 120
GetAxisParameter_DAC_OUT, 121
GetAxisParameter_DAC_SCALE, 121
GetAxisParameter_DATUM_IN, 121
GetAxisParameter_DECEL, 122
GetAxisParameter_DECEL_ANGLE, 122
GetAxisParameter_DEMAND_EDGES, 123
GetAxisParameter_DEMAND_SPEED, 123
GetAxisParameter_DPOS, 123
GetAxisParameter_DRIVE_CONTROL, 124
GetAxisParameter_DRIVE_CONTROLWORD, 124
GetAxisParameter_DRIVE_CURRENT, 125
GetAxisParameter_DRIVE_CW_MODE, 125
GetAxisParameter_DRIVE_ENABLE, 126
GetAxisParameter_DRIVE_FE, 126
GetAxisParameter_DRIVE_FE_LIMIT, 126
GetAxisParameter_DRIVE_INDEX, 127
GetAxisParameter_DRIVE_INPUTS, 127
GetAxisParameter_DRIVE_MODE, 128
GetAxisParameter_DRIVE_MONITOR, 128
GetAxisParameter_DRIVE_NEG_TORQUE, 128
GetAxisParameter_DRIVE_PARAMETER, 129
GetAxisParameter_DRIVE_POS_TORQUE, 129
GetAxisParameter_DRIVE_PROFILE, 130
GetAxisParameter_DRIVE_SET_VAL, 130
GetAxisParameter_DRIVE_STATUS, 131
GetAxisParameter_DRIVE_TORQUE, 131
GetAxisParameter_DRIVE_TYPE, 131
GetAxisParameter_DRIVE_VALUE, 132
GetAxisParameter_DRIVE_VELOCITY, 132
GetAxisParameter_ENCODER, 133
GetAxisParameter_ENCODER_BITS, 133
GetAxisParameter_ENCODER_CONTROL, 133
GetAxisParameter_ENCODER_FILTER, 134
GetAxisParameter_ENCODER_ID, 134
GetAxisParameter_ENCODER_STATUS, 135
GetAxisParameter_ENCODER_TURNS, 135
GetAxisParameter_END_DIR_LAST, 136
GetAxisParameter_ENDMOVE, 136
GetAxisParameter_ENDMOVE_BUFFER, 136
GetAxisParameter_ENDMOVE_SPEED, 137
GetAxisParameter_ERRORMASK, 137
GetAxisParameter_FAST_JOG, 138
GetAxisParameter_FASTDEC, 138
GetAxisParameter_FE, 138
GetAxisParameter_FE_LATCH, 139
GetAxisParameter_FE_LIMIT, 139
GetAxisParameter_FE_LIMIT_MODE, 140
GetAxisParameter_FE_RANGE, 140
GetAxisParameter_FEED_OVERRIDE, 141
GetAxisParameter_FHOLD_IN, 141
GetAxisParameter_FHSPEED, 141
GetAxisParameter_FORCE_ACCEL, 142
GetAxisParameter_FORCE_DECEL, 142
GetAxisParameter_FORCE_DWELL, 143
GetAxisParameter_FORCE_JERK, 143
GetAxisParameter_FORCE_RAMP, 143
GetAxisParameter_FORCE_SPEED, 144
GetAxisParameter_FRAME, 144
GetAxisParameter_FRAME_REP_DIST, 145
GetAxisParameter_FRAME_SCALE, 145
GetAxisParameter_FS_LIMIT, 146
GetAxisParameter_FULL_SP_RADIUS, 146
GetAxisParameter_FWD_IN, 146
GetAxisParameter_FWD_JOG, 147
GetAxisParameter_FWD_START, 147
GetAxisParameter_I_GAIN, 148
GetAxisParameter_IDLE, 148
GetAxisParameter_IN_POS, 148
GetAxisParameter_IN_POS_DIST, 149
GetAxisParameter_IN_POS_SPEED, 149
GetAxisParameter_INTERP_FACTOR, 150
GetAxisParameter_INVERT_STEP, 150
GetAxisParameter_JERK, 151
GetAxisParameter_JOGSPEED, 151
GetAxisParameter_LINK_AXIS, 151
GetAxisParameter_LOADED, 152
GetAxisParameter_LOOKAHEAD_FACTOR, 152
GetAxisParameter_MARK, 153
GetAxisParameter_MARKB, 153
GetAxisParameter_MERGE, 153
GetAxisParameter_MOVE_COUNT, 154

GetAxisParameter_MOVE_PA, 154
GetAxisParameter_MOVE_PA_CONT, 155
GetAxisParameter_MOVE_PA_IDLE, 155
GetAxisParameter_MOVE_PB, 156
GetAxisParameter_MOVE_PB_CONT, 156
GetAxisParameter_MOVE_PB_IDLE, 156
GetAxisParameter_MOVELINK_MODIFY, 157
GetAxisParameter_MOVES_BUFFERED, 157
GetAxisParameter_MPOS, 158
GetAxisParameter_MSPEED, 158
GetAxisParameter_MSPEED_FILTER, 158
GetAxisParameter_MSPEEDF, 159
GetAxisParameter_MTYPE, 159
GetAxisParameter_NTYPE, 160
GetAxisParameter_OFFPOS, 160
GetAxisParameter_OPEN_WIN, 161
GetAxisParameter_OUTLIMIT, 161
GetAxisParameter_OV_GAIN, 161
GetAxisParameter_P_GAIN, 162
GetAxisParameter_POS_DELAY, 162
GetAxisParameter_POSI_SEQ_DELAY, 163
GetAxisParameter_POSI_SEQ_MODE, 163
GetAxisParameter_PP_STEP, 163
GetAxisParameter_PS_ENCODER, 164
GetAxisParameter_PWM_CYCLE, 164
GetAxisParameter_PWM_MARK, 165
GetAxisParameter_RAISE_ANGLE, 165
GetAxisParameter_REG_INPUTS, 166
GetAxisParameter_REG_POS, 166
GetAxisParameter_REG_POSB, 166
GetAxisParameter_REGIST_CONTROL, 167
GetAxisParameter_REGIST_DELAY, 167
GetAxisParameter_REGIST_SPEED, 168
GetAxisParameter_REGIST_SPEEDB, 168
GetAxisParameter_REMAIN, 168
GetAxisParameter_REP_DIST, 169
GetAxisParameter_REP_OPTION, 169
GetAxisParameter_REV_IN, 170
GetAxisParameter_REV_JOG, 170
GetAxisParameter_REV_START, 171
GetAxisParameter_ROBOT_DPOS, 171
GetAxisParameter_ROBOT_FE, 171
GetAxisParameter_ROBOT_FE_LATCH, 172
GetAxisParameter_ROBOT_FE_RANGE, 172
GetAxisParameter_ROBOT_FS_LIMIT, 173
GetAxisParameter_ROBOT_RS_LIMIT, 173
GetAxisParameter_ROBOT_SP_MODE, 173
GetAxisParameter_ROBOT_UNITS, 174
GetAxisParameter_ROBOTSTATUS, 174
GetAxisParameter_RS_LIMIT, 175
GetAxisParameter_S_REF, 175
GetAxisParameter_S_REF_OUT, 176
GetAxisParameter_SERVO, 176
GetAxisParameter_SLOT_NUMBER, 176
GetAxisParameter_SPEED, 177
GetAxisParameter_SRAMP, 177
GetAxisParameter_START_DIR_LAST, 178
GetAxisParameter_STARTMOVE_SPEED, 178
GetAxisParameter_STOP_ANGLE, 178
GetAxisParameter_SYNC_AXIS, 179
GetAxisParameter_SYNC_CONTROL, 179
GetAxisParameter_SYNC_DPOS, 180
GetAxisParameter_SYNC_DWELL, 180
GetAxisParameter_SYNC_FLIGHT, 181
GetAxisParameter_SYNC_PAUSE, 181
GetAxisParameter_SYNC_TIME, 181
GetAxisParameter_SYNC_TIMER, 182
GetAxisParameter_SYNC_WITHDRAW, 182
GetAxisParameter_T_REF, 183
GetAxisParameter_T_REF_OUT, 183
GetAxisParameter_TABLE_POINTER, 183
GetAxisParameter_TANG_DIRECTION, 184
GetAxisParameter_TCP_DPOS, 184
GetAxisParameter_TCP_FS_LIMIT, 185
GetAxisParameter_TCP_RS_LIMIT, 185
GetAxisParameter_TCP_UNITS, 186
GetAxisParameter_UNITS, 186
GetAxisParameter_VECTOR_BUFFERED, 186
GetAxisParameter_VERIFY, 187
GetAxisParameter_VFF_GAIN, 187
GetAxisParameter_VP_ACCEL, 188
GetAxisParameter_VP_JERK, 188
GetAxisParameter_VP_MODE, 188
GetAxisParameter_VP_POSITION, 189
GetAxisParameter_VP_SPEED, 189
GetAxisParameter_WORLD_ACCEL, 190
GetAxisParameter_WORLD_DECEL, 190
GetAxisParameter_WORLD_DPOS, 191
GetAxisParameter_WORLD_FS_LIMIT, 191
GetAxisParameter_WORLD_JOGSPEED, 191
GetAxisParameter_WORLD_RS_LIMIT, 192
GetAxisParameter_WORLD_SPEED, 192
GetAxisParameter_WORLD_UNITS, 193
GetDIn, 193
GetDIn32, 194
GetDOut, 194, 195
GetDOut32, 195
GetFirmwareVersionNumber, 195
GetFrameGroup, 196
GetInvertIN, 196
GetMultiTableValues, 197
GetMultiVrValues, 197
GetProcessParameter_ERROR_LINE, 198
GetProcessParameter_INDEVICE, 198
GetProcessParameter_OUTDEVICE, 198
GetProcessParameter_PMOVE, 199
GetProcessParameter_PROC_LINE, 199
GetProcessParameter_PROC_STATUS, 200
GetProcessParameter_RUN_ERROR, 200
GetProcessParameter_TICKS, 200
GetSlotParameter_SERIAL_NUMBER, 201
GetSystemParameter_ADDRESS, 201
GetSystemParameter_AIN0, 202
GetSystemParameter_AIN1, 202
GetSystemParameter_AIN2, 202
GetSystemParameter_AIN3, 203

GetSystemParameter_AINBI0, 203
GetSystemParameter_AINBI1, 203
GetSystemParameter_AINBI2, 203
GetSystemParameter_AINBI3, 204
GetSystemParameter_AOUT0, 204
GetSystemParameter_AOUT1, 204
GetSystemParameter_AOUT2, 205
GetSystemParameter_AOUT3, 205
GetSystemParameter_AUTO_ETHERCAT, 205
GetSystemParameter_AVERAGE_JITTER, 206
GetSystemParameter_CONTROL, 206
GetSystemParameter_CPU_EXCEPTIONS, 206
GetSystemParameter_CURRENT_JITTER, 206
GetSystemParameter_DISPLAY, 207
GetSystemParameter_ERROR_AXIS, 207
GetSystemParameter_FLASH_STATUS, 207
GetSystemParameter_FPU_EXCEPTIONS, 208
GetSystemParameter_HMI_PROC, 208
GetSystemParameter_INPUTS0, 208
GetSystemParameter_INPUTS1, 209
GetSystemParameter_IP_ADDRESS, 209
GetSystemParameter_IP_GATEWAY, 209
GetSystemParameter_IP_MAC, 209
GetSystemParameter_IP_MEMORY_CONFIG, 210
GetSystemParameter_IP_NETMASK, 210
GetSystemParameter_IP_PROTOCOL_CONFIG, 210
GetSystemParameter_IP_PROTOCOL_CTRL, 211
GetSystemParameter_IP_TCP_TIMEOUT, 211
GetSystemParameter_IP_TCP_TX_THRESHOLD, 211
GetSystemParameter_IP_TCP_TX_TIMEOUT, 212
GetSystemParameter_LAST_AXIS, 212
GetSystemParameter_LIMIT_BUFFERED, 212
GetSystemParameter_MAX_JITTER, 212
GetSystemParameter_MISSED_INTERRUPTS, 213
GetSystemParameter_MISSED_RECV_FRAMES, 213
GetSystemParameter_MISSED_SEND_FRAMES, 213
GetSystemParameter_MODBUS_TX_DELAY, 214
GetSystemParameter_MOTION_ERROR, 214
GetSystemParameter_MPE_MODE, 214
GetSystemParameter_N_ANA_IN, 215
GetSystemParameter_N_ANA_OUT, 215
GetSystemParameter_NIN, 215
GetSystemParameter_NIO, 215
GetSystemParameter_NOP, 216
GetSystemParameter_PCMCAT_PROC, 216
GetSystemParameter_PLC_CONFIG, 216
GetSystemParameter_PLC_ERROR, 217
GetSystemParameter_PLC_OVERFLOW, 217
GetSystemParameter_PLC_RUN, 217
GetSystemParameter_PROCNUMBER, 218
GetSystemParameter_REMOTE_PROC, 218
GetSystemParameter_SCHEDULE_TYPE, 218
GetSystemParameter_SCOPE_CYCLE_COUNT, 218
GetSystemParameter_SCOPE_DELAY, 219
GetSystemParameter_SCOPE_POS, 219
GetSystemParameter_SCOPE_TRIGGER_POS, 219
GetSystemParameter_SERIAL_NUMBER, 220
GetSystemParameter_SERVO_OFFSET, 220
GetSystemParameter_SERVO_PERIOD, 220
GetSystemParameter_SLOT0_TIME, 221
GetSystemParameter_SLOT1_TIME, 221
GetSystemParameter_SLOT2_TIME, 221
GetSystemParameter_SLOT3_TIME, 221
GetSystemParameter_SYSTEM_ERROR, 222
GetSystemParameter_SYSTEM_ERROR_MASK, 222
GetSystemParameter_SYSTEM_LOAD, 222
GetSystemParameter_SYSTEM_LOAD_MAX, 223
GetSystemParameter_TEXT_FILE_LOADER_PROC, 223
GetSystemParameter_TSIZE, 223
GetSystemParameter_UNIT_ERROR, 224
GetSystemParameter_VERSION, 224
GetSystemParameter_VRSIZE, 224
GetSystemParameter_WDOG, 224
GetTableValue, 225
GetTcpHostName, 225
GetTcpPort, 225
GetTcpTimeout, 226
GetVrString, 226, 227
GetVrStringLen, 227
GetVrValue, 228
IsConnected, 337
MoveAbs, 228
MoveAbsMulti, 228
MoveAbsSeq, 229
MoveAbsSp, 229
MoveAbsSpMulti, 230
MoveCirc, 230
MoveCircSp, 231
MoveHelical, 232
MoveHelicalSp, 232
MoveLink, 233
MoveModify, 235
MovePick, 235
MoveRel, 236
MoveRelMulti, 236
MoveRelSeq, 237
MoveRelSp, 237
MoveRelSpMulti, 238
MoveSpherical, 238
MoveSphericalSp, 239
New, 240
OpenConnection, 240
PSwitch, 240
PSwitchMulti, 241
PSwitchOff, 242

RapidStop, 242
Reverse, 242
Select, 244
SetAOut, 244
SetAxisBase, 244
SetAxisParameter_ACCEL, 245
SetAxisParameter_AFF_GAIN, 245
SetAxisParameter_ATYPE, 246
SetAxisParameter_AXIS_ACCEL, 246
SetAxisParameter_AXIS_ADDRESS, 246
SetAxisParameter_AXIS_DECEL, 247
SetAxisParameter_AXIS_DISPLAY, 247
SetAxisParameter_AXIS_ENABLE, 247
SetAxisParameter_AXIS_ENABLE_OVERRIDE, 248
SetAxisParameter_AXIS_ERROR_COUNT, 248
SetAxisParameter_AXIS_FS_LIMIT, 249
SetAxisParameter_AXIS_MAX_JOG_SPEED, 249
SetAxisParameter_AXIS_MODE, 249
SetAxisParameter_AXIS_RS_LIMIT, 250
SetAxisParameter_AXIS_SPEED, 250
SetAxisParameter_AXIS_UNITS, 250
SetAxisParameter_AXIS_Z_OUTPUT, 251
SetAxisParameter_BACKLASH_DIST, 251
SetAxisParameter_CHANGE_DIR_LAST, 252
SetAxisParameter_CLOSE_WIN, 252
SetAxisParameter_CLUTCH_RATE, 252
SetAxisParameter_CORNER_MODE, 253
SetAxisParameter_CORNER_STATE, 253
SetAxisParameter_CREEP, 253
SetAxisParameter_D_GAIN, 254
SetAxisParameter_DAC, 254
SetAxisParameter_DAC_SCALE, 255
SetAxisParameter_DATUM_IN, 255
SetAxisParameter_DECEL, 255
SetAxisParameter_DECEL_ANGLE, 256
SetAxisParameter_DRIVE_CONTROL, 256
SetAxisParameter_DRIVE_CONTROLWORD, 256
SetAxisParameter_DRIVE_CW_MODE, 257
SetAxisParameter_DRIVE_ENABLE, 257
SetAxisParameter_DRIVE_FE_LIMIT, 258
SetAxisParameter_DRIVE_INDEX, 258
SetAxisParameter_DRIVE_MODE, 258
SetAxisParameter_DRIVE_NEG_TORQUE, 259
SetAxisParameter_DRIVE_PARAMETER, 259
SetAxisParameter_DRIVE_POS_TORQUE, 259
SetAxisParameter_DRIVE_PROFILE, 260
SetAxisParameter_DRIVE_SET_VAL, 260
SetAxisParameter_DRIVE_TYPE, 261
SetAxisParameter_DRIVE_VALUE, 261
SetAxisParameter_ENCODER_BITS, 261
SetAxisParameter_ENCODER_CONTROL, 262
SetAxisParameter_ENCODER_FILTER, 262
SetAxisParameter_ENCODER_ID, 262
SetAxisParameter_ENCODER_STATUS, 263
SetAxisParameter_ENCODER_TURNS, 263
SetAxisParameter_END_DIR_LAST, 264
SetAxisParameter_ENDMOVE, 264
SetAxisParameter_ENDMOVE_BUFFER, 264
SetAxisParameter_ENDMOVE_SPEED, 265
SetAxisParameter_ERRORMASK, 265
SetAxisParameter_FAST_JOG, 265
SetAxisParameter_FASTDEC, 266
SetAxisParameter_FE_LIMIT, 266
SetAxisParameter_FE_LIMIT_MODE, 267
SetAxisParameter_FE_RANGE, 267
SetAxisParameter_FEED_OVERRIDE, 267
SetAxisParameter_FHOLD_IN, 268
SetAxisParameter_FHSPEED, 268
SetAxisParameter_FORCE_ACCEL, 268
SetAxisParameter_FORCE_DECEL, 269
SetAxisParameter_FORCE_DWELL, 269
SetAxisParameter_FORCE_JERK, 270
SetAxisParameter_FORCE_RAMP, 270
SetAxisParameter_FORCE_SPEED, 270
SetAxisParameter_FRAME, 271
SetAxisParameter_FRAME_REP_DIST, 271
SetAxisParameter_FRAME_SCALE, 271
SetAxisParameter_FS_LIMIT, 272
SetAxisParameter_FULL_SP_RADIUS, 272
SetAxisParameter_FWD_IN, 273
SetAxisParameter_FWD_JOG, 273
SetAxisParameter_FWD_START, 273
SetAxisParameter_I_GAIN, 274
SetAxisParameter_IN_POS, 274
SetAxisParameter_IN_POS_DIST, 274
SetAxisParameter_IN_POS_SPEED, 275
SetAxisParameter_INTERP_FACTOR, 275
SetAxisParameter_INVERT_STEP, 276
SetAxisParameter_JERK, 276
SetAxisParameter_JOGSPEED, 276
SetAxisParameter_LOOKAHEAD_FACTOR, 277
SetAxisParameter_MERGE, 277
SetAxisParameter_MOVE_COUNT, 277
SetAxisParameter_MOVE_PA, 278
SetAxisParameter_MOVE_PA_CONT, 278
SetAxisParameter_MOVE_PA_IDLE, 279
SetAxisParameter_MOVE_PB, 279
SetAxisParameter_MOVE_PB_CONT, 279
SetAxisParameter_MOVE_PB_IDLE, 280
SetAxisParameter_MOVELINK_MODIFY, 280
SetAxisParameter_MSPEED_FILTER, 280
SetAxisParameter_OFFPOS, 281
SetAxisParameter_OPEN_WIN, 281
SetAxisParameter_OUTLIMIT, 282
SetAxisParameter_OV_GAIN, 282
SetAxisParameter_P_GAIN, 282
SetAxisParameter_POS_DELAY, 283
SetAxisParameter_POSI_SEQ_DELAY, 283
SetAxisParameter_POSI_SEQ_MODE, 283
SetAxisParameter_PP_STEP, 284
SetAxisParameter_PWM_CYCLE, 284
SetAxisParameter_PWM_MARK, 285
SetAxisParameter_RAISE_ANGLE, 285
SetAxisParameter_REG_INPUTS, 285
SetAxisParameter_REGIST_CONTROL, 286

SetAxisParameter_REGIST_DELAY, 286
SetAxisParameter_REP_DIST, 286
SetAxisParameter_REP_OPTION, 287
SetAxisParameter_REV_IN, 287
SetAxisParameter_REV_JOG, 288
SetAxisParameter_REV_START, 288
SetAxisParameter_ROBOT_FE_RANGE, 288
SetAxisParameter_ROBOT_FS_LIMIT, 289
SetAxisParameter_ROBOT_RS_LIMIT, 289
SetAxisParameter_ROBOT_SP_MODE, 289
SetAxisParameter_ROBOT_UNITS, 290
SetAxisParameter_RS_LIMIT, 290
SetAxisParameter_S_REF, 291
SetAxisParameter_S_REF_OUT, 291
SetAxisParameter_SERVO, 291
SetAxisParameter_SLOT_NUMBER, 292
SetAxisParameter_SPEED, 292
SetAxisParameter_SRAMP, 292
SetAxisParameter_START_DIR_LAST, 293
SetAxisParameter_STARTMOVE_SPEED, 293
SetAxisParameter_STOP_ANGLE, 294
SetAxisParameter_SYNC_AXIS, 294
SetAxisParameter_SYNC_CONTROL, 294
SetAxisParameter_SYNC_DPOS, 295
SetAxisParameter_SYNC_DWELL, 295
SetAxisParameter_SYNC_FLIGHT, 295
SetAxisParameter_SYNC_PAUSE, 296
SetAxisParameter_SYNC_TIME, 296
SetAxisParameter_SYNC_WITHDRAW, 297
SetAxisParameter_T_REF, 297
SetAxisParameter_T_REF_OUT, 297
SetAxisParameter_TABLE_POINTER, 298
SetAxisParameter_TANG_DIRECTION, 298
SetAxisParameter_TCP_FS_LIMIT, 298
SetAxisParameter_TCP_RS_LIMIT, 299
SetAxisParameter_TCP_UNITS, 299
SetAxisParameter_UNITS, 300
SetAxisParameter_VECTOR_BUFFERED, 300
SetAxisParameter_VERIFY, 300
SetAxisParameter_VFF_GAIN, 301
SetAxisParameter_VP_MODE, 301
SetAxisParameter_WORLD_ACCEL, 301
SetAxisParameter_WORLD_DECEL, 302
SetAxisParameter_WORLD_FS_LIMIT, 302
SetAxisParameter_WORLD_JOGSPEED, 303
SetAxisParameter_WORLD_RS_LIMIT, 303
SetAxisParameter_WORLD_SPEED, 303
SetAxisParameter_WORLD_UNITS, 304
SetDOut, 304
SetDOut32, 305
SetFrameGroup, 305
SetInvertIN, 306
SetMotion, 306, 307
SetMotionSp, 308–311
SetMultiAxisBase, 312
SetMultiTableValues, 312
SetMultiVrValues, 313
SetProcessParameter_INDEVICE, 313
SetProcessParameter_OUTDEVICE, 313
SetProcessParameter_TICKS, 314
SetSlotParameter_SERIAL_NUMBER, 314
SetSystemParameter_ADDRESS, 314
SetSystemParameter_AIN0, 315
SetSystemParameter_AIN1, 315
SetSystemParameter_AIN2, 315
SetSystemParameter_AIN3, 316
SetSystemParameter_AINBI0, 316
SetSystemParameter_AINBI1, 316
SetSystemParameter_AINBI2, 317
SetSystemParameter_AINBI3, 317
SetSystemParameter_AOUT0, 317
SetSystemParameter_AOUT1, 318
SetSystemParameter_AOUT2, 318
SetSystemParameter_AOUT3, 318
SetSystemParameter_AUTO_ETHERCAT, 319
SetSystemParameter_AVERAGE_JITTER, 319
SetSystemParameter_CPU_EXCEPTIONS, 319
SetSystemParameter_CURRENT_JITTER, 320
SetSystemParameter_DISPLAY, 320
SetSystemParameter_FPU_EXCEPTIONS, 320
SetSystemParameter_HMI_PROC, 321
SetSystemParameter_IP_ADDRESS, 321
SetSystemParameter_IP_GATEWAY, 321
SetSystemParameter_IP_MAC, 322
SetSystemParameter_IP_MEMORY_CONFIG, 322
SetSystemParameter_IP_NETMASK, 322
SetSystemParameter_IP_PROTOCOL_CONFIG, 323
SetSystemParameter_IP_PROTOCOL_CTRL, 323
SetSystemParameter_IP_TCP_TIMEOUT, 323
SetSystemParameter_IP_TCP_TX_THRESHOLD, 324
SetSystemParameter_IP_TCP_TX_TIMEOUT, 324
SetSystemParameter_LAST_AXIS, 324
SetSystemParameter_LIMIT_BUFFERED, 325
SetSystemParameter_MAX_JITTER, 325
SetSystemParameter_MISSED_INTERRUPTS, 325
SetSystemParameter_MISSED_RECV_FRAMES, 326
SetSystemParameter_MISSED_SEND_FRAMES, 326
SetSystemParameter_MODBUS_TX_DELAY, 326
SetSystemParameter_NIO, 327
SetSystemParameter_PCMCAT_PROC, 327
SetSystemParameter_PLC_CONFIG, 327
SetSystemParameter_REMOTE_PROC, 328
SetSystemParameter_SCHEDULE_TYPE, 328
SetSystemParameter_SCOPE_CYCLE_COUNT, 328
SetSystemParameter_SCOPE_DELAY, 329
SetSystemParameter_SCOPE_POS, 329
SetSystemParameter_SERIAL_NUMBER, 329
SetSystemParameter_SERVO_OFFSET, 330

SetSystemParameter_SERVO_PERIOD, 330
SetSystemParameter_SLOT0_TIME, 330
SetSystemParameter_SLOT1_TIME, 331
SetSystemParameter_SLOT2_TIME, 331
SetSystemParameter_SLOT3_TIME, 331
SetSystemParameter_SYSTEM_ERROR_MASK, 332
SetSystemParameter_SYSTEM_LOAD_MAX, 332
SetSystemParameter_TEXT_FILE_LOADER_PROC, 332
SetSystemParameter_WDOG, 333
SetTableValue, 333
SetTcpHostName, 333
SetTcpPort, 334
SetTcpTimeout, 334
SetVrString, 334
SetVrValue, 335
StepRatio, 335
StopProgram, 336
UploadFile, 336
Trio.UnifiedApi.ProgressInfo, 338
current_pos, 338
evt, 338
total_length, 338
Trio.UnifiedApi.TrioConnectionException, 339
ErrorCode, 340
Message, 340
TrioConnectionException, 339
Trio.UnifiedApi.TrioConnectionFactory, 340
CreateConnectionFlex7, 341
CreateConnectionPCMCAT, 341
CreateConnectionTCP, 341
ErrorStringGetter, 342
TrioConnectionException
Trio.UnifiedApi.TrioConnectionException, 339
TrioConnectionFactory.cs, 457
type
Trio.UnifiedApi.FileInfo, 45

Undefined
Trio.UnifiedApi, 34
Unknown
Trio.UnifiedApi, 22
UnknownFloatingPointFormat
Trio.UnifiedApi, 24
Unsigned16
Trio.UnifiedApi, 22
Unsigned32
Trio.UnifiedApi, 22
Unsigned64
Trio.UnifiedApi, 22
Unsigned8
Trio.UnifiedApi, 22
UnsupportedController
Trio.UnifiedApi, 24
UploadFile
Trio.UnifiedApi.ITrioConnection, 336
USB_Comms
Trio.UnifiedApi, 19
USE_FASTDEC_WHEN_CANCELLING_RPS_JOG
Trio.UnifiedApi, 17
USE_NON_SIGNEXTENDED_ANALOGUE_FEEDBACK
Trio.UnifiedApi, 17

VELPROFILE
Trio.UnifiedApi, 36
VersionMismatch
Trio.UnifiedApi, 24
VOut
Trio.UnifiedApi, 19
VR
Trio.UnifiedApi, 38

Warning
Trio.UnifiedApi, 33
WETCAM
Trio.UnifiedApi, 35

Yaskawa
Trio.UnifiedApi, 19

ZOut
Trio.UnifiedApi, 39