Trio Motion Technology Ltd., Shannon Way, Tewkesbury, Gloucestershire, GL20 8ND United Kingdom Tel: +44 (0)1684-292333 Trio Motion Technology LLC., 187 Northpointe Blvd, Suite 105 Freeport, PA 16229, USA Tel: +1 724 472 4100 Trio Motion Technology (Shanghai) Co. Ltd., A1104 Yunding International Commercial Plaza, 800 Chengshan Rd, Pudong New Area, Shanghai, CHINA CHINA Tel: +86 21 587 976 59 Trio Motion Technology, Teerth Business Center, 3rd Floor, Unit No. 7, Block EL - 15, MIDC, Bhosari, Pimpri-Chinchwad, Pune, 411026, INDIA Tel: +91 827 506 5446





1. History

V1.0 22 March 2022 First release

2. Introduction

The Modbus RTU protocol is a means of communication that allows data exchange between programmable logic controllers (PLCs) and peripheral devices. Electronic devices can exchange information over serial lines using the Modbus protocol.

The protocol was developed and published by Modicon® in 1979 for use with its programmable logic controllers. It is built using a master/slave architecture and supports serial devices that use the RS232/RS485/RS422 protocols. Modbus has been adopted by many industrial automation vendors and has become an open standard within the industry. Industrial automation and supervisory control and data acquisition (SCADA) systems often employ the Modbus protocol.

Trio has had Modbus RTU server node handlers in our products for more than 2 decades but the more recent introduction of Modbus RTU client functions allows the Motion Coordinator to access many more types of device using Serial ports.

3. Principles

Modbus is a request-response protocol with defined data packets containing the address and a number of 16 bit words followed by a CRC. The MODBUS command in the Trio calls a function in firmware which sends one request and interprets the response received from the server.

Most Modbus clients send requests on a cyclic basis. Therefore it is normal to write a program which sends one or more requests, repeated with a time interval between each.

The example program emulates a typical HMI which opens the connection by polling a single value from the server and once a valid response is received, goes into a loop polling for the required data cyclically. The initial opening request is sent every 0.5 seconds and once the full cyclic operation starts the data is polled more frequently at 20 msec intervals.

4. Program

The program is written in Trio BASIC and runs on the Motion Coordinator. A PC running Motion Perfect



is needed to connect to the Motion Coordinator, write and de-bug the program.

4.1. Initialisation

The serial port must be set up using the SETCOM command. Mode 11 is needed for Modbus Client operation.

SETCOM syntax: SETCOM(baudrate, databits, stopbits, parity, port, mode, 0, timeout, linetype)

Example 1, RS232;

```
' set the RS232 serial port 1 parameters, and start
' Modbus RTU client protocol (mode: 11)
SETCOM(19200, 8, 1, 0, 1, 11) ' RS232 port 1
```

Example 2, RS485 with 2-wire echo suppression;

```
' set the RS485 serial port 2 parameters, and start
' Modbus RTU client protocol (mode: 11)
SETCOM(19200, 8, 1, 0, 2, 11, 0, 3, 1) ' RS485 port 2
```

4.2. Open Modbus Channel

A channel must be opened within the Motion Coordinator. The MODBUS open function returns a handle for the program to use when referencing the channel.

```
' Open client on RS232, and obtain handle
nw_type = -1 ' -1 for RS232, -2 for 2 wire RS485, -3 for 4 wire RS422
cn_type = 0 'connection type: 0 = client, 1 = server
IF MODBUS($0, -1, nw_type, cn_type, 20) = FALSE THEN ' handle is in VR(20)
PRINT #5, "Failed to open Modbus Client session"
STOP
ELSE
PRINT #5, "Opened Modbus Client with handle "; VR(20) [0]
mbh = VR(20)
ENDIF
```

4.3. Data exchange

The simplest construction of data exchange is to put the MODBUS function 3 command in a loop with a short time delay.

```
DEFCONST retry state 1
DEFCONST cyclic state 2
mb_cycle = retry_state
WHILE TRUE
  SELECT CASE mb cycle
   CASE retry state
      ' send first enquiry to slave device
      ' keep sending until communication is established
      PRINT #5, "Retry Modbus "; mbh[0]
      addr = 1
      mb status = MODBUS(3, -1, mbh, $03, addr, 1, 1, 1)
      WA(500)
      IF mb status = TRUE THEN
       mb cycle = cyclic state
       PRINT #5, "Start cyclic"
      ENDIF
```



```
CASE cyclic_state
    ' Read holding registers starting at #100, (40101) quantity 10
    ' Modbus function $03
    addr = 1
    mb_status = MODBUS(3, -1, mbh, $03, addr, 100, 10, 1005)
    WA(20)
    IF mb_status = FALSE THEN
    mb_cycle = retry_state
    ENDIF
    END_CASE
WEND
```

5. Appendix

Sometimes it is necessary to close any outstanding open channels. This routine will check and close 12 channels if found to be open.

```
' close all connections
FOR i = 0 TO 11
 IF NOT MODBUS(2, -1, i, 20) THEN
    'Failed to execute MODBUS function 2
   PRINT #5, "Error: failed to check port status, handle: "; i[0]
 ELSE
    'Read connection status ok, now close if it is open
   IF VR(20) = 1 THEN
     IF NOT MODBUS (1, -1, i) THEN
        'Error failed to close the port
       PRINT #5, "Error: failed to close connection, handle "; i[0]
       STOP
     ELSE
       PRINT #5, "Closed connection, handle "; i[0]
     ENDIF
   ENDIF
 ENDIF
```

NEXT i

6. References

Trio BASIC help (.chm) file in Motion Perfect version 5.2.1 contains further information for all commands. Especially;

MODBUS SETCOM PRINT DEFCONST SELECT CASE

Modbus RTU definition can be found here: <u>https://modbus.org/specs.php</u>

Motion Coordinators supporting Modbus Client: <u>www.triomotion.com</u>