Trio Motion Technology Ltd.
Shannon Way, Tewkesbury,
Gloucestershire. GL20 8ND
United Kingdom
Tel: +44 (0)1684 292 333
Fax: +44 (0)1684 297 929

1000 Gamma Drive
Suite 206
Pittsburgh, PA 15238
United States of America
Tel: +1 412 968 9744
Fax: +1 412 968 9746

B1602 Tomson Centre
188 Zhang Yang Rd.,
Pudong New Area, Shanghai,
Postal code: 200122
CHINA
Tel/Fax: +86 21 587 97659

Doc No.: AN-310
Version: 1.1
Date: 17 August 2017
Subject: Rotating Product Programming using FRAME 12

APPLICATION NOTE

1. Changes:

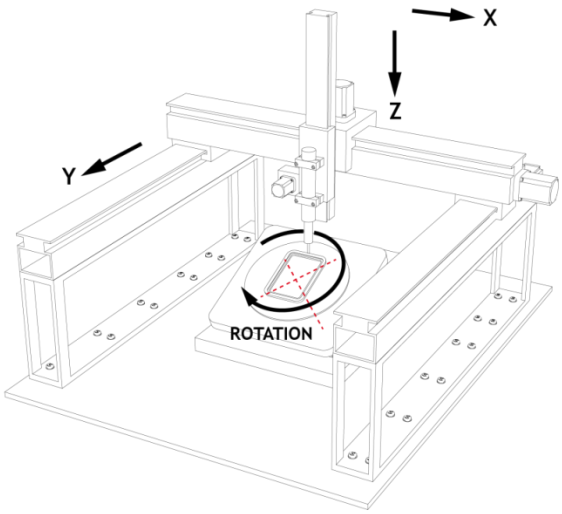
Version 1.0: Initial version of document.
Version 1.1: Updated to match system software version 2.0240

2. Introduction

The FRAME 12 transformation is an adaptation of FRAME 11 specifically designed for dispensing on to a rotating product tilted at an angle under an XYZ machine. This frame has been built for a mobile phone dispensing application and requires an MC4XX Motion Coordinator with system software version 2.0240 or higher. Kinematics FEC 22 are required to be enabled on the MC4XX to use this feature.

3. Diagram

The following picture shows the type of machine:



As the machine has 4 axes the tool above can be maintained at a specific angle to the product surface

by the XYZ motion of the table.

4. Initialization and datuming:

The machine axis directions MUST match those shown in the diagram above. The tilt angle should be positive. If necessary change the machine axis directions inside the drive or using STEP_RATIO(-1,1) inside the Motion Coordinator.

```
' In this example configuration:

' Axis 0 – X
' Axis 1 – Y
' Axis 2 – Z
' Axis 3 - Rotation

FRAME=0' This clears any set frame from previous machine running

' FRAME 12 requires 2 parameters, here we load it into TABLE positions 100 and 101:
tilt = 40 * PI / 180
TABLE(100,8000*180/PI,tilt)' set number of encoder edges/radian on rotation axis

INTERP_FACTOR AXIS(0)=1
INTERP_FACTOR AXIS(1)=1
INTERP_FACTOR AXIS(2)=1
INTERP_FACTOR AXIS(3)=1

UNITS AXIS(0)=2500' set number of encoder edges/mm on X
UNITS AXIS(1)=2500' set number of encoder edges/mm on Y
UNITS AXIS(2)=2500' set number of encoder edges/mm on Z
UNITS AXIS(3)=8000' set number of encoder edges/degree for rotation

SPEED AXIS(0)=10
SPEED AXIS(1)=10
SPEED AXIS(2)=10
SPEED AXIS(3)=10

ACC(100) AXIS(0)
ACC(100) AXIS(1)
ACC(100) AXIS(2)
ACC(100) AXIS(3)
' Put machine datuming here
' Datuming is done with FRAME=0 so axes are un-coupled

' After datuming a DEFPOS effectively defines the machine position.
' NB: The XY=(0,0) position is always the centre of rotation

DEFPOS(0,-35,0)' This is the tool position relative to the rotation centre

FRAME_GROUP(0,100,0,1,2,3)' Define FRAME data position and axes involved

FRAME=12' Activate the transformation
```

5. Checking Axis Directions:

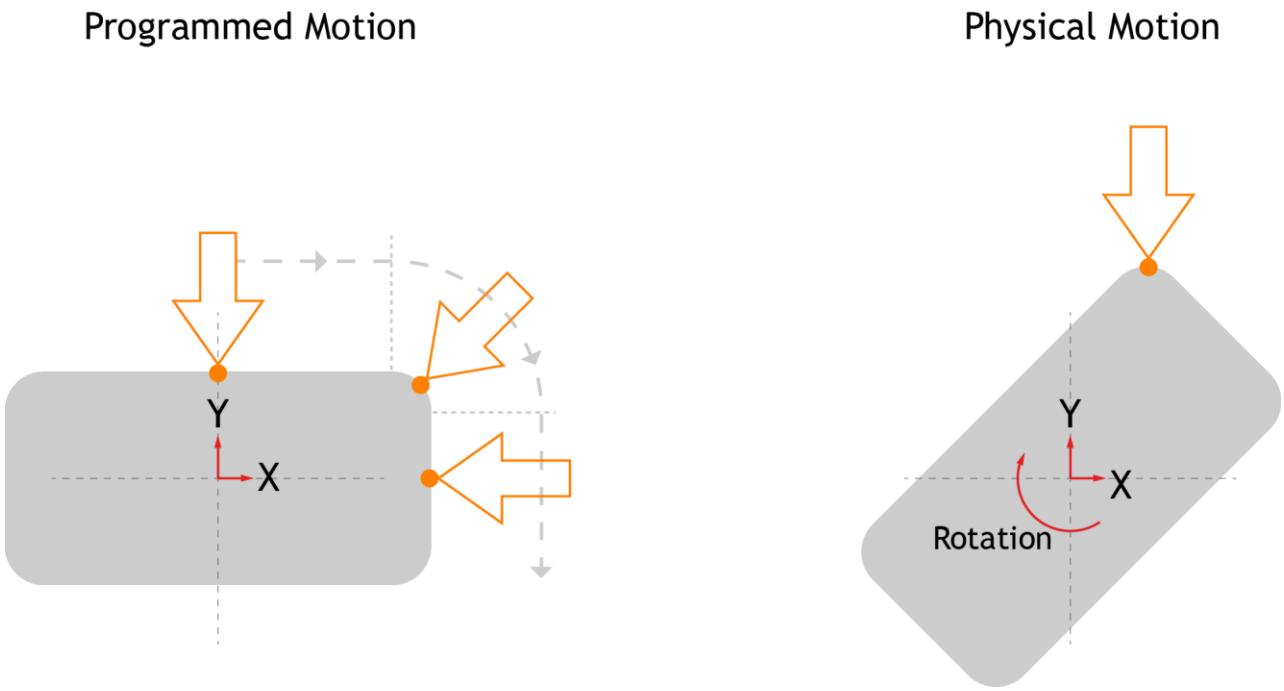
The XYZ and Rotation directions must be checked before and after activating FRAME=12.

With the "tilt" setting = 0 the rotation compensation can be checked in XY only first. After adding the "tilt" setting programming motion in Y will result in motion on both Y and Z.

6. Programming with FRAME=12 active:

After activating the FRAME=12 command the user can program the motion as the XY position along the part outline. If rotations are required to match the tool to the surface of the part these must be programmed by the user program. Note that although the FRAME must be aware of the encoder edges/radian of the machine for the rotation, the user can program the rotations in the more easily understandable degrees or other units.

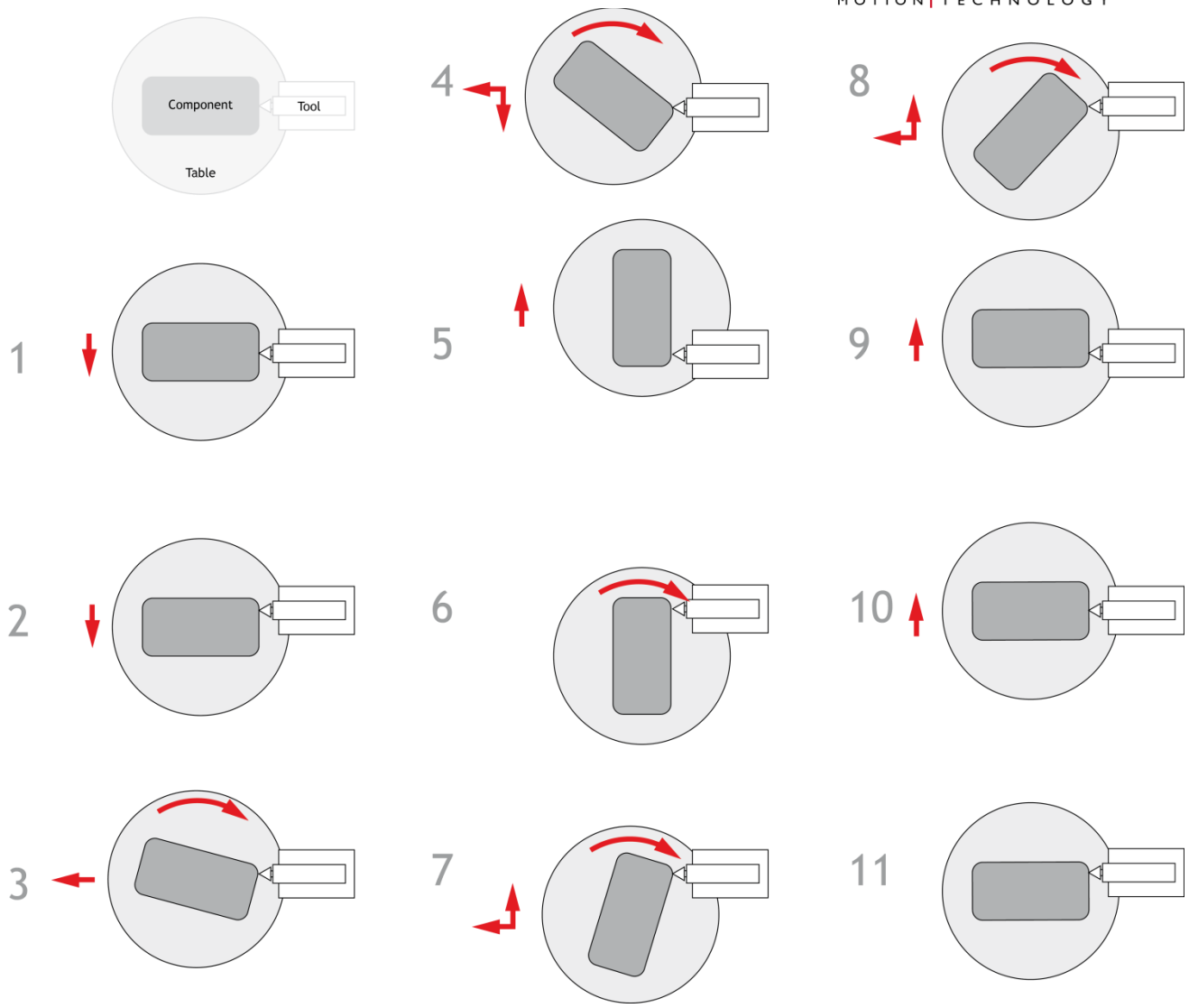
NOTE: IN THE DIAGRAM BELOW THE Y DIRECTION DOES NOT MATCH THE MACHINE AXIS



The programmed motion is planned as XY + tool rotation. The physical motion is fixed tool and XY + rotation motion of the part.

```
rot_axis=3
SPEED AXIS(0)=3

LIMIT_BUFFERED=64 ' Long enough for all moves to load into buffer
INTERP_FACTOR AXIS(rot_axis)=1
MERGE=ON
SRAMP AXIS(rot_axis)=30
long=30
short=20
```

```
radius=10
arclength=radius*PI/2
approach=3 ' Calc accel and speed for rotation allowing for approach:
arc_time = arclength / SPEED AXIS(0)
rota_speed = 90 / arc_time
ramp_dist = 2 * approach
ramp_time = ramp_dist / SPEED AXIS(0)
ramp_acc = rota_speed / ramp_time

SPEED AXIS(rot_axis)=rota_speed
ACC(ramp_acc) AXIS(rot_axis)

MOVE_COUNT=0
prev_move_count=0

DEFPOS(0) AXIS(rot_axis)

' Load moves:

MOVE(long/2-approach,0)
MOVE(approach,0)'2
MOVECIRC(radius,radius,0,radius,0)
MOVE(0,short-approach)
MOVE(0,approach)'5
MOVECIRC(-radius,radius,-radius,0,0)
MOVE(-long+approach,0)
MOVE(-approach,0)'8
MOVECIRC(-radius,-radius,0,-radius,0)
MOVE(0,-short+approach)
MOVE(0,-approach)'11
MOVECIRC(radius,-radius,radius,0,0)
MOVE(long/2,0) '

REPEAT
   WAIT UNTIL MOVE_COUNT<>prev_move_count
   IF MOVE_COUNT=2 THEN MOVE(-90) AXIS(rot_axis)
   IF MOVE_COUNT=5 THEN MOVE(-90) AXIS(rot_axis)
   IF MOVE_COUNT=8 THEN MOVE(-90) AXIS(rot_axis)
   IF MOVE_COUNT=11 THEN MOVE(-90) AXIS(rot_axis)
   PRINT#5,MOVE_COUNT
   prev_move_count=MOVE_COUNT
UNTIL MOVE_COUNT > 12
```



7. FRAME_GROUP and FRAME

Please refer to the latest MC4XX programming manual for details of the FRAME_GROUP and FRAME commands.

8. Fiducials and Orientation

Please refer to the latest MC4XX programming manual for details of the FRAME_GROUP and FRAME commands.