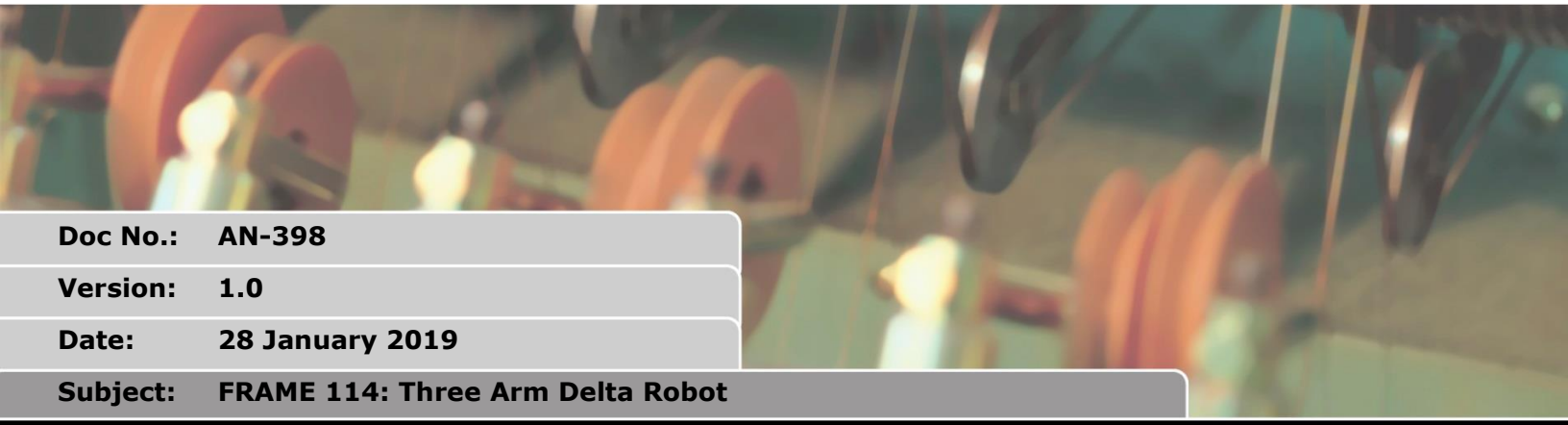
Shannon Way,
Tewkesbury,
Gloucestershire. GL20 8ND
United Kingdom
Tel: +44 (0)1684 292 333
Fax: +44 (0)1684 297 929

187 Northpointe Blvd,
Suite 105
Freeport, PA 16229
United States of America
Tel: +1 724-472-4100
Fax: +1 724-472-4101

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

SCMC House
16/6 Vishal Nagar
Pimpale Nilakh, Wakad, Pune
PIN 411027
INDIA
Tel: +91 827 506 5441



Doc No.: **AN-398**

Version: **1.0**

Date: **28 January 2019**

Subject: **FRAME 114: Three Arm Delta Robot**

APPLICATION NOTE

1. Changes:

Version 1.0 Completed draft 23rd Jan 2019

2. Introduction

FRAME = 114 enables the transformation for a 3 Arm Delta Robot. This allows you to define the end position of the wrist in X, Y, Z, and wrist angle (relative to the Z axis). The FRAME 114 transformation is available for MC4XX controllers (including MC464, EURO408 and MC405) with the Robotic Function FEC (feature enable code) fitted.

FRAME 114 is similar to FRAME 14 but with some additions. These additions include the inclusion of several optional parameters in the table data defining the robot. Further, while frame 14 is inflexible in that the robot parameters must be programmed in Micrometres, Frame 114 allows the robot to be defined in any desired unit, which will then be the units in which the robot is programmed.

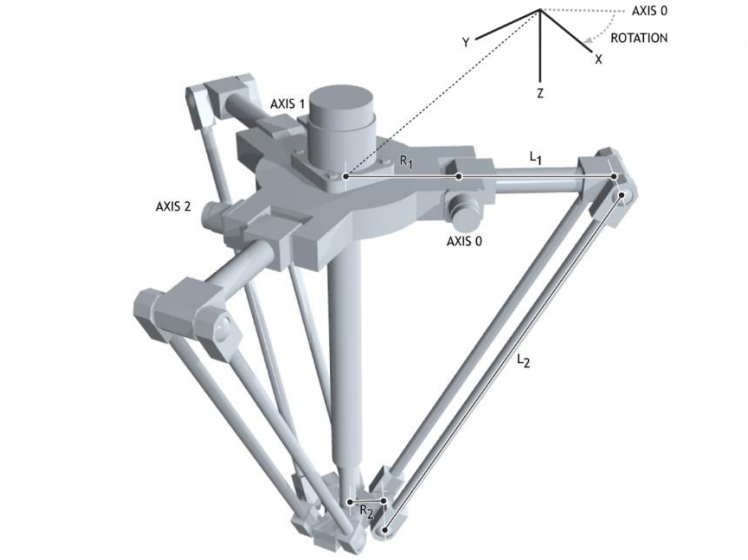
3. Axis order

The axes within the allocated FRAME_GROUP have the following functions:

First Axis	Cartesian X axis
Second Axis	Cartesian Y axis
Third Axis	Cartesian Z axis
Fourth Axis	End effector rotation axis

4. Diagram

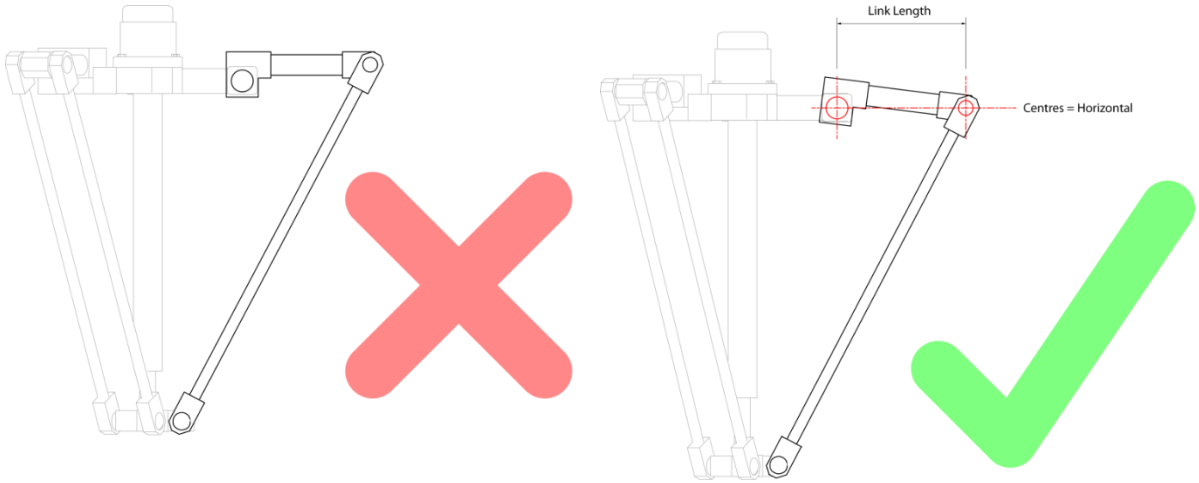
The following pictures show the types of machine that can be controlled using this FRAME:



Once the FRAME is enabled set the UNITS to FRAME_SCALE so that the Cartesian movements use the same scale as that used in the table data. So, if the TABLE data is programmed in mm then when UNITS is set to FRAME_SCALE then the robot can be programmed in mm.
Set the UNITS on the rotational (wrist) axes to FRAME_SCALE * 2PI / 360 so that they are programmed in degrees. You can of course set UNITS for all axis to any suitable scale.

5. Datuming

When datuming, (homing) it is necessary for each of the arms of the delta robot to be in the horizontal position (i.e. with the joints at either end of the link at the same level with respect to the robot's base).



6. Parameterization

13 parameters needed to be set into a sequence of TABLE values for use by the transformation mathematics. The base TABLE position is set using FRAME_GROUP.

Table data	0	Top radius to joint (R1)
	1	Wrist radius to joint (R2)
	2	Upper arm length (L1)
	3	Lower arm length (L2)
	4	Edges per radian
	5	Angle of rotation in radians (Rotation)
	6	Linkx (optional with 4 or 5 axis)
	7	Linky (optional with 4 or 5 axis)
	8	Linkz (optional with 4 or 5 axis)
	9	Encoder edges/radian (optional Z rotation)
	10	Encoder edges/radian (optional Y rotation)

Note that R1, R2, L1, L2 should all be defined in the same UNITS that the robot will be programmed in. Also, note that the angle of rotation in radians, is the positive rotation of the X axis about the robot's base, with respect to the direction in which the axis 0 arm is pointing.

7. Example

```
BASE(0)


REPEAT

  IF VR(1)=2 THEN
    GOSUB frame_114 'Move the robot in Cartesian system
    VR(1)=0
  ELSEIF VR(1)=1 THEN
    GOSUB frame_0 'Move the robot in Joint mode
    VR(1)=0
  ENDIF

UNTIL FALSE

STOP

frame_0:

'configure for frame 0:
```

```
arm1_encoder = 16000 'Encoder counts per revolution
arm1_gearbox = 360 '50
arm1_edges_per_radian = arm1_encoder * arm1_gearbox / (2 * PI)
arm1_edges_per_degree = arm1_encoder * arm1_gearbox / (360)

arm2_encoder = 16000 'Encoder counts per revolution
arm2_gearbox = 360 '50
arm2_edges_per_radian = arm2_encoder * arm2_gearbox / (2 * PI)
arm2_edges_per_degree = arm2_encoder * arm2_gearbox / (360)

arm3_encoder = 16000 'Encoder counts per revolution
arm3_gearbox = 360 '50
arm3_edges_per_radian = arm3_encoder * arm3_gearbox / (2 * PI)
arm3_edges_per_degree = arm3_encoder * arm3_gearbox / (360)

rot_encoder = 200*32*16*80/360 'Encoder counts per revolution
rot _gearbox = 360 '50
rot _edges_per_radian = rot_encoder * rot _gearbox / (2 * PI)
rot _edges_per_degree = rot _encoder * rot _gearbox / (360)


FRAME = 0

BASE(0)
UNITS = arm1_edges_per_degree 'degrees
AXIS_UNITS = UNITS
RS_LIMIT=-1000
FS_LIMIT=1000

BASE(1)
UNITS = arm2_edges_per_degree 'degrees
AXIS_UNITS = UNITS
RS_LIMIT=-1000
FS_LIMIT=1000

BASE(2)
UNITS = arm3_edges_per_mm 'mm
AXIS_UNITS = UNITS
RS_LIMIT=-1000
FS_LIMIT=1000

BASE(3)
UNITS=rot_edges_per_degree
AXIS_UNITS = UNITS
RS_LIMIT=-360
FS_LIMIT=360


FOR ax=0 TO 3
  SPEED AXIS(ax)=15
  ACC(200) AXIS(ax)
NEXT ax

RETURN

frame_114:
```

```
    TABLE(0,400)
    TABLE(0, 115)
    TABLE(1, 44)
    TABLE(2, 260)
    TABLE(3, 715)
    TABLE(4, 131072)
    TABLE(5, pi/4)


'configure for frame 114:

    FRAME_GROUP(0,0,0,1,2,3)'Delta robot
    FRAME = 114

    BASE(0)
    UNITS = FRAME_SCALE 'mm
    RS_LIMIT=-1000
    FS_LIMIT=1000
    BASE(1)
    UNITS = FRAME_SCALE 'mm
    RS_LIMIT=-1000
    FS_LIMIT=1000
    BASE(2)
    UNITS = FRAME_SCALE 'mm
    RS_LIMIT=-1000
    FS_LIMIT=1000
    BASE(3)
    UNITS =( FRAME_SCALE * 2 * PI) / (360)'degrees
    RS_LIMIT=-360
    FS_LIMIT=360

    FOR ax=0 TO 3
      SPEED AXIS(ax)=10
      ACC(200) AXIS(ax)
    NEXT ax

RETURN
```