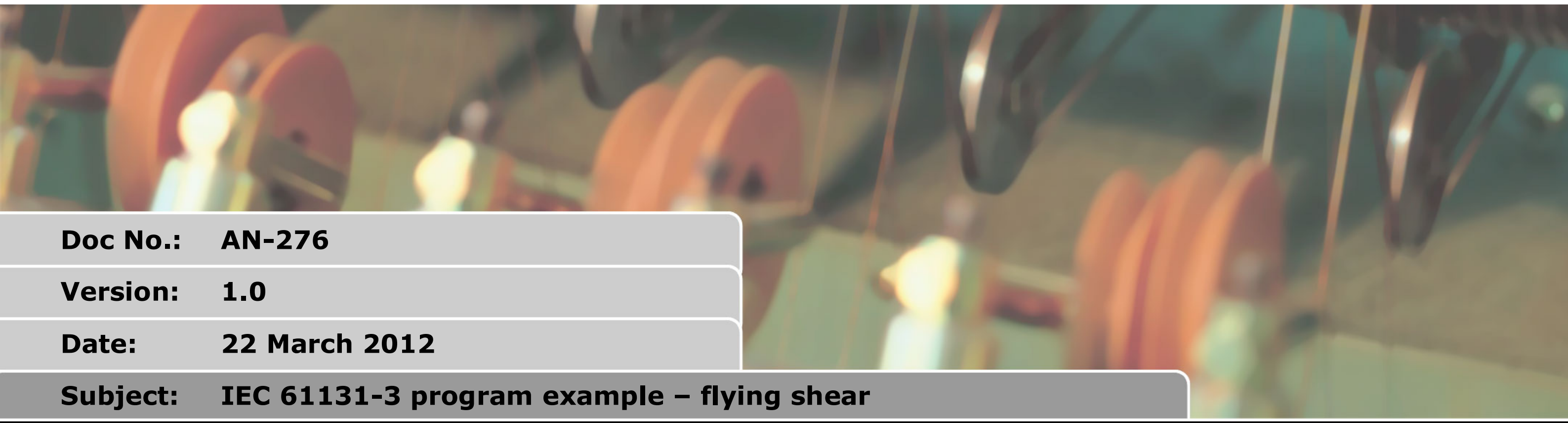
Shannon Way,
Tewkesbury,
Gloucestershire. GL20 8ND
United Kingdom
Tel: +44 (0)1684 292 333
Fax: +44 (0)1684 297 929

187 Northpointe Blvd,
Suite 105
Freeport, PA 16229
United States of America
Tel: +1 724-540-5018
Fax: +1 724-540-5098

Tomson Centre
118 Zhang Yang Rd., B1701
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: +91 206 811 4902

TRIO MOTION TECHNOLOGY



Doc No.:	**AN-276**
Version:	**1.0**
Date:	**22 March 2012**
Subject:	**IEC 61131-3 program example – flying shear**

APPLICATION NOTE

1. Introduction

This example takes a flying shear program written in Trio BASIC and presents the same functions as both Function Block Diagram (FBD) and Structured Text (ST). The flying shear will be familiar to programmers who already use Trio BASIC and so this is intended as an aid to Trio BASIC users trying the IEC 61131-3 language for the first time.

The actual way the flying shear works and the details of the motion functions are not covered. Details about motion functions can be found in the Trio BASIC reference manuals.

Also not covered is the actual way to open, write and generate the IEC 61131-3 programs with Motion Perfect v3. The IEC 61131-3 help provided with MPv3 gives instructions on using the IEC 61131-3 editors.

Hint: IEC 61131-3 Variables must be defined in the "Variables" window. IEC 61131-3 functions must be added to a program from the "Toolbox" window.

2. Flying Shear in Trio BASIC

The program shown below is the core of a small flying shear program. First it datums (homes) the axis, then 2 MOVELINK commands control the forward movement of the shear carrier and the reverse movement to return to home. Both MOVELINKs are following a master axis which is presumed to be the conveyor or transport mechanism feeding material into the shear.

```
' simple flying shear example
' shear axis = axis 0
' feed axis = axis 1

' datum the shear axis (homes on Z mark)
BASE(0)
CREEP=60
DATUM(3)
WAIT IDLE

IF MTYPE AXIS(1)=0 THEN FORWARD AXIS(1)

' VR definitions
cut_length=20
```

```
dist=21
link_dist=22
acc_dist=23
dec_dist=24
ret_link_dist=25
cut_posn=26

' IO defintions
ready=9
trig_ml1=10
cut=11 ' use output 11 for cutter

BASE(1)
REP_OPTION = 1

REPEAT
  BASE(1)
  REP_DIST = VR(cut_length)
  BASE(0)
  MOVELINK(VR(dist),VR(link_dist),VR(acc_dist),VR(dec_dist),1,2,0)
  WAIT UNTIL MPOS>VR(cut_posn)
  OP(cut, ON)
  WAIT IDLE
  OP(cut, OFF)
  rad=VR(ret_link_dist)/2
  MOVELINK(-VR(dist),VR(ret_link_dist),rad,rad,1)
  WAIT IDLE
UNTIL FALSE
```

3. Axis setup program in Trio BASIC

This program is provided simply to set up 2 virtual axes so that the main demonstration programs can be run.

```
'  Sets up 2 virtual axes for the IEC61131-3 demos

BASE(0)
ATYPE=0
REP_OPTION=0
SERVO=ON
DATUM_IN=8

BASE(1)
ATYPE=0
SERVO=ON
REP_OPTION=1
IF MTYPE=0 THEN
  FORWARD ' represents the movement of the conveyor
ENDIF

WDOG=ON
```

4. VR initialisation

Each demo program uses the VRs as the input values to the MOVELINKs. The VRs must therefore be initialised. The values here are just to get started.

```
VR(20)=1000 ' rep_dist
```

```
VR(21)=200 ' dist
VR(22)=300 ' link_dist
VR(23)=50 ' link acc dist
VR(24)=50 ' link dec dist
VR(25)=200 ' return link dist
VR(26)=30 ' shear trigger position
```

5. Function Block Diagram

Here is the flying shear as a FBD. The same sequence exists as in the Trio BASIC version, but due to the way the IEC program works, each motion function block is executed every PLC cycle. So there is a “Done” output from each block which is then used to trigger the next stage of the motion cycle.

For example, the MOVELINK execute inputs are ANDed with a master enable BOOL value which does not got TRUE until the DATUM function has completed.

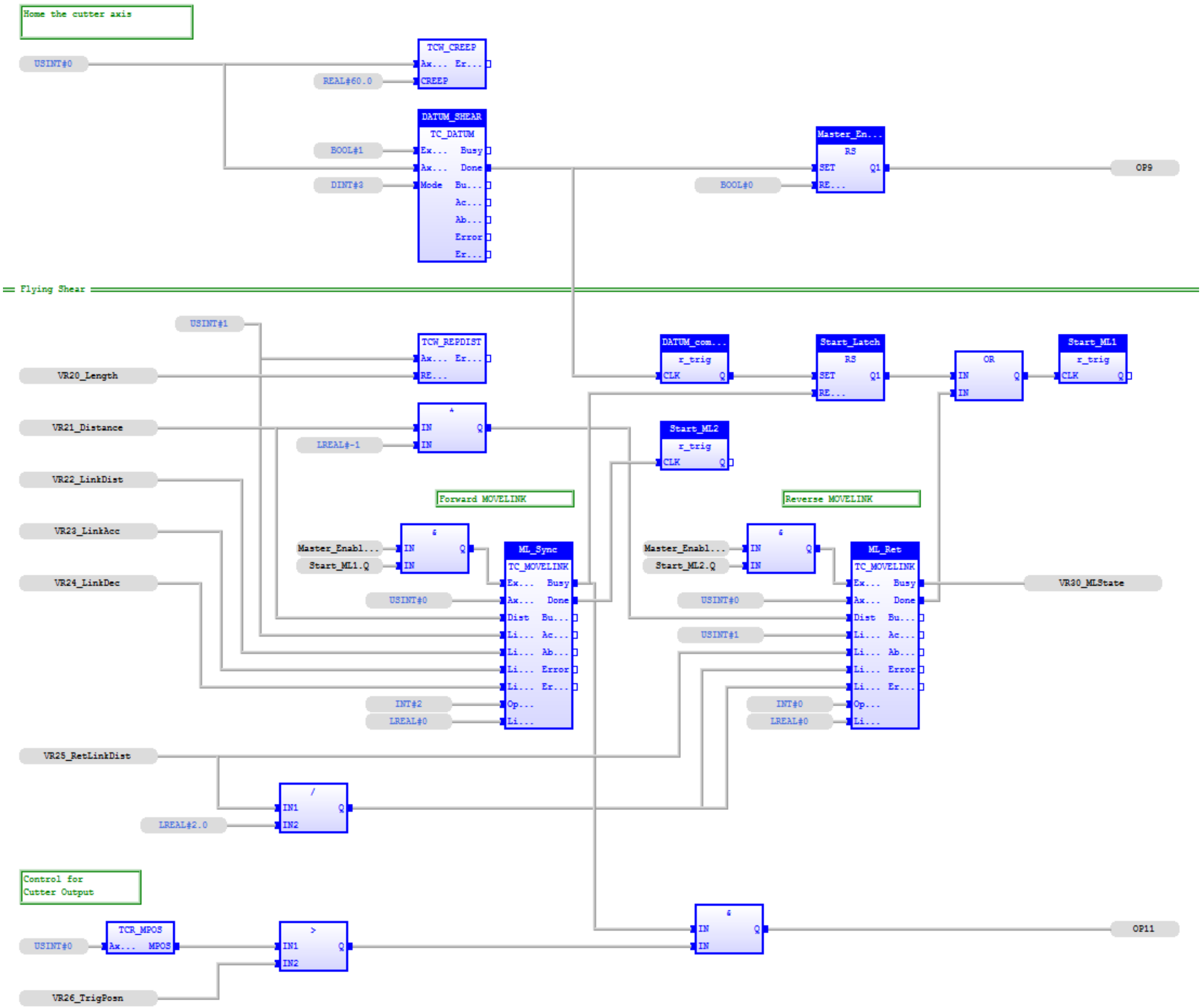


Figure 1. Flying Shear FBD

Output 11 is the cutter trigger signal. It is output after the shear carrier axis has passed the position where synchronised speed has been reached. The cutter will “fire” on the rising edge of the signal.

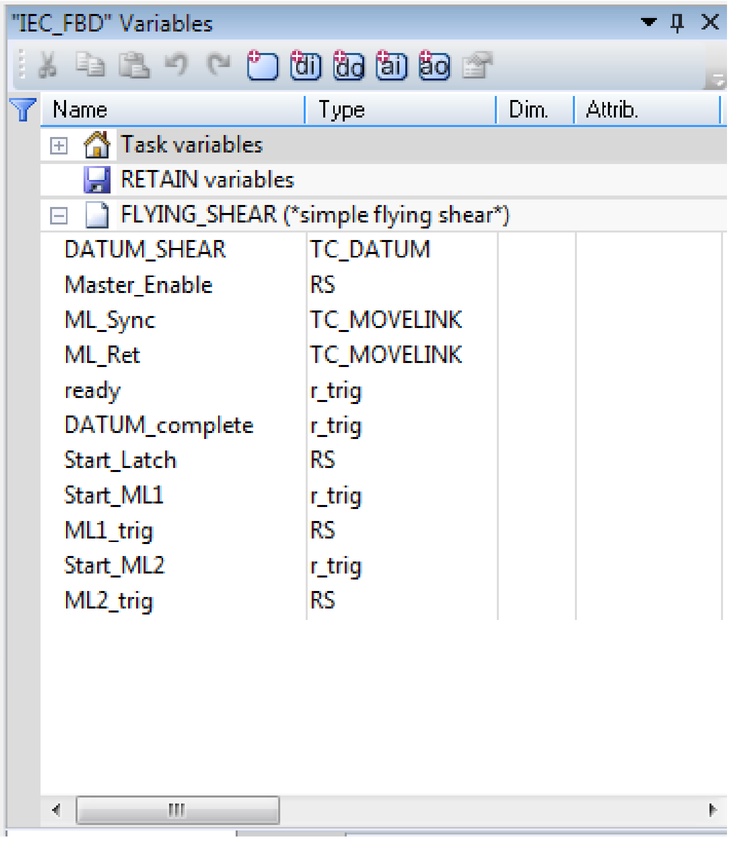


Figure 2. Local Variable List

Notice that there is more than one instance of some function blocks. Each instance is given its own name.

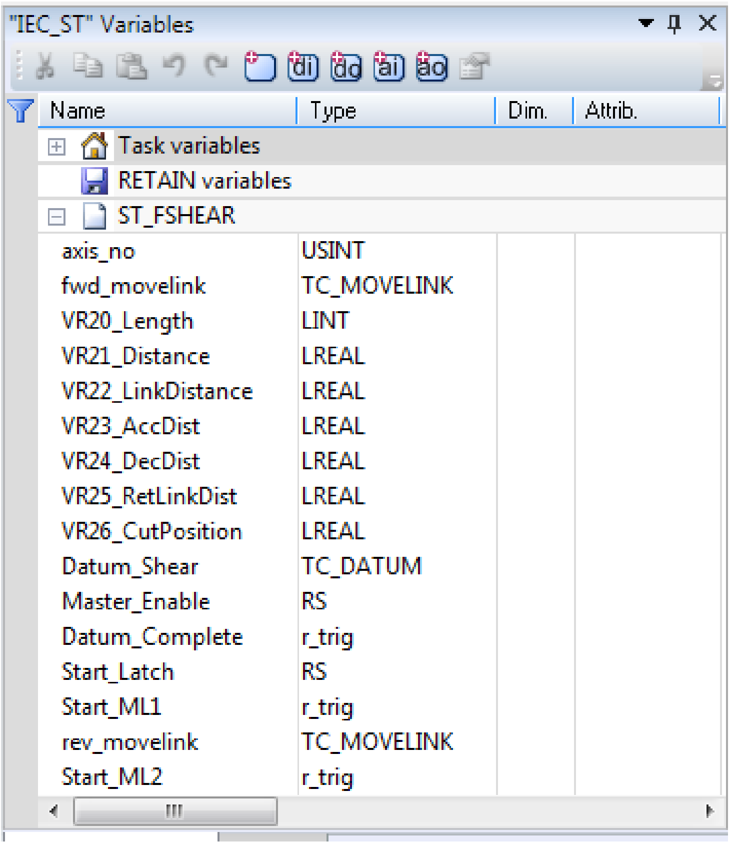
6. Structured Text

The same functions when written in ST look like this:

```
 axis_no := 0;

 TCW_CREEP(AxisNo:=axis_no, CREEP:=60.0);
 Datum_Shear(Execute:=1, AxisNo:=axis_no, Mode:=3);

 Master_Enable(SET:=Datum_Shear.Done, RESET1:=0);
 OP9:=Master_Enable.Q1;

 Datum_Complete(CLK:=Datum_Shear.Done); (* rising edge trigger function *)
 Start_Latch(SET:=Datum_Complete.Q, RESET1:=fwd_movelink.busy);

(* Flying Shear part starts here *)
 TCW_REPDIST( AxisNo:=1, REP_DIST:=VR20_Length ); (* REP_DIST of master axis *)

 Start_ML1(CLK:=(Start_Latch.Q1 OR rev_movelink.done));

 fwd_movelink(Execute:=(Master_Enable.Q1 AND Start_ML1.Q),
              AxisNo:=axis_no,
              Dist:=VR21_Distance,
              LinkDist:=VR22_LinkDistance,
              LinkAccDist:=VR23_AccDist,
              LinkDecDist:=VR24_DecDist,
              LinkAxis:=1,
              Options:=2,
              LinkPos:=0
```

```
               );

 Start_ML2(CLK:=fwd_movelink.Done);

 rev_movelink(Execute:=(Master_Enable.Q1 AND Start_ML2.Q),
               AxisNo:=axis_no,
               Dist:=-VR21_Distance,
               LinkDist:=VR25_RetLinkDist,
               LinkAccDist:=VR25_RetLinkDist/2,
               LinkDecDist:=VR25_RetLinkDist/2,
               LinkAxis:=1,
               Options:=0,
               LinkPos:=0
               );

(* Control for cutter output *)

 IF (TCR_MPOS( AxisNo:=axis_no ) > VR26_CutPosition) AND (fwd_movelink.busy = TRUE)
THEN
    OP11:=1;
ELSE
    OP11:=0;
END_IF;
```

Note that unlike Trio BASIC, this program is continuously scanned from top to bottom at the PLC rate. It is therefore truly a textual representation of a Function Block Diagram rather than a set of linear code.



Name	Type	Dim.	Attrib.
axis_no	USINT		
fwd_movelink	TC_MOVELINK		
VR20_Length	LINT		
VR21_Distance	LREAL		
VR22_LinkDistance	LREAL		
VR23_AccDist	LREAL		
VR24_DecDist	LREAL		
VR25_RetLinkDist	LREAL		
VR26_CutPosition	LREAL		
Datum_Shear	TC_DATUM		
Master_Enable	RS		
Datum_Complete	r_trig		
Start_Latch	RS		
Start_ML1	r_trig		
rev_movelink	TC_MOVELINK		
Start_ML2	r_trig		

Figure 3. Variable list for the ST Flying Shear

7. Alternative axis configuration (ST)

The axis configuration that is done in Trio BASIC, see section 3, can be made in ST instead. Here is the code from section 3 in ST:

```
(* Setup 2 virtual axes for the IEC61131-3 demos *)

TCW_ATYPE( AxisNo:=0, ATYPE:=0 ) (* set to virtual axis type *)
TCW_REPOPTION( AxisNo:=0, REP_OPTION:=0 )
TCW_SERVO( AxisNo:=0, SERVO:=1 )

TCW_DATUMIN( AxisNo:=0, DATUM_IN:=8)

TCW_ATYPE( AxisNo:=1, ATYPE:=0 ) (* set to virtual axis type *)
TCW_REPOPTION( AxisNo:=1, REP_OPTION:=1 )
TCW_SERVO( AxisNo:=1, SERVO:=1 )

TCW_WDOG( WDOG:=1 )
```

This program should be run in a separate task so that it can be run once and then stopped. If it is put in the main ST program, then it will scan continuously and that is not necessary.

Alternatively this part can be made into a function of its own which is triggered once by the main program. More information about making functions will follow in a future document.

8. Programs

The programs used to make this document can be downloaded from the Trio website. www.triomotion.com

Location: Support -> Technical Notes.

Click the button: SOFTWARE

Open the folder IEC 61131

Download the ZIP file IEC_Flying_Shear.zip

Unzip the file to your MPV3 Projects folder and load the project using Motion Perfect v3