

```
1  PROGRAM AxisInterface
2  VAR
3      // LREAL Feedback signal calculated from iBallFeedback integer type
4      rBallFeedback : REAL ;
5      // Raw BallFeedback value from analog input on Drive. In INT type
6      iBallFeedback : INT ;
7      // Raw motor temperature value from Drive. In INT type
8      iTempMotor : INT ;
9      // LREAL motor temperature calculated to deg C from INT type
10     rTempMotor : REAL ;
11     // Raw motor force value from Drive. In INT type
12     iForce : INT ;
13     // LREAL motor force calculated to absolute %
14     rForce : REAL ;
15     // Input from HMI to clear drive errors
16     bClearError : BOOL ;
17     // Setting of beam position command dependent on bBeamZero
18     rBeamPosCmd : LREAL ;
19     rAxis_X_position : LREAL ;
20 END_VAR
21
```

```
1  //-----
2  //----- Axis-access with AxisInterface
3  //-----
4  IF NOT TE_AxisInterfaceMainProg . InitDone OR TE_AxisInterfaceMainProg . Error
5  THEN
6      RETURN ; // do not continue, when Motion is not in RUN or initialization
7      failed.
8  END_IF
9
10 // Interlock if Emergency stop in HMI has been activated. Alarm must be
11 // acknowledged before enable can be set
12 // Additionally no drive enable can be set until axis has no errors
13 IF gvl . iAcknowledgeEMStop = 2 OR gvl . bError_Axis THEN
14     gvl . bEnable := 0 ;
15 END_IF
16
17 // Calculation of variables
18 rBallFeedback := ( -1 * iBallFeedback + 5000 ) * 77.57 ; // Invert, move from
19 // minus and scale to real size
20 rBallFeedback := ( rBallFeedback / 1000 ) ; // Move decimals
21 gvl . rBallFeedback := rBallFeedback - 65 ; // moving zero point with an offset
22 gvl . rBallFeedback3D := gvl . rBallFeedback - 330 ; // moving the ball from center
23 // to start of beam
24
```

```
22
23 // Motor temperature conversion from integer to real value
24 rTempMotor := iTempMotor ;
25 gvl.rTempMotorCalc := rTempMotor / 10 ;
26 // Motor force conversion from integer to real value
27 rForce := iForce ;
28 gvl.rForceMotor := ABS ( rForce / 10 ) ;
29
30 // Manual adjustment of the beam to balance the ball on a horizontal beam
31 // Set the position command signal for the beam
32 // Set beam to 0 deg
33 IF gvl.bBeamZero THEN
34     // Set the position command signal to 0
35     rBeamPosCmd := 0.0 ;
36     // Disable the PID controller
37     gvl.bEnablePID := FALSE ;
38 ELSE
39     // Set the position command signal to the PID control output
40     rBeamPosCmd := gvl.rPIDCtrlOut ;
41 END_IF
42
43
44
45 // Axis error handling
46 arAxisCtrl_gb [ 1 ] . Admin . ClearError := gvl.bClearError ; //Clears axis
47                                     errors
48 ImcCtrl . Admin . ClearError := gvl.bClearError ; // Clears the red light, goes
49                                     to green
50 gvl.bError_Axis := arAxisStatus_gb [ 1 ] . Diag . Error ;
51
52 //-----
53 // Axis control
54 //-----
55
56 IF bRemoteOn_gb = FALSE THEN
57     //arAxisCtrl_gb[1].PosMode.Position := gvl.rPIDCtrlOut;
58     arAxisCtrl_gb [ 1 ] . PosMode . Position := rBeamPosCmd + gvl.rJogStep ; //
59     position cmd of axis
60     arAxisCtrl_gb [ 1 ] . PosMode . Velocity := gvl.rAxisMaxRpm ; // velocity cmd
61     of axis
62     arAxisCtrl_gb [ 1 ] . PosMode . DynValues . Acceleration := gvl.rAxisAccel ; //
63     axis acceleration cmd
64     arAxisCtrl_gb [ 1 ] . PosMode . DynValues . Deceleration := gvl.rAxisDecel ; //
65     axis deceleration cmd
66     arAxisCtrl_gb [ 1 ] . Admin . _OpModeBits . MODE_POS_ABS := gvl.bEnable ; //
67     enabling the axis
68
69 //-----
70 // Axis-Command: Switch all axes to ModeAH with bStopAll = TRUE
71 //-----
```

```
66     IF gvl.bStopAll = TRUE THEN
67         IF arAxisStatus_gb [ 1 ] . Admin . Active THEN
68             //arAxisCtrl_gb[1].Admin._OpMode := ModeAH;
69             gvl.bEnable := FALSE ;
70         END_IF
71     ELSE
72         // set to Mode Ab when bStop All = FALSE
73         IF arAxisStatus_gb [ 1 ] . Admin . Active AND arAxisStatus_gb [ 1 ] . Admin .
74         _OpModeAckBits . MODE_AH THEN
75             arAxisCtrl_gb [ 1 ] . Admin . _OpMode := ModeAb ;
76         END_IF
77     END_IF
78 END_IF
79
```