

PositionServo Sample Program - Standard Internal Velocity Mode

Concept:

This sample program is an example of how to use the PositionServo in 'Internal Velocity Mode', using the analog input (Ain1) for the velocity reference (as is done automatically when using external velocity mode).

The program uses the Enable Input on A3 as a safety enable for safety devices on the system. Input A4 is defined as the Run / Stop input to cause the drive to start / stop.

Digital output 1 provides an inverse fault indication (0 = Fault, 1 – Healthy) to the outside world.

Analog Input 1 is used as the velocity reference and is accessed through drive variable AIN1 (in volts: +10 to -10). Note that Analog input 1 scaling, dead-band, and offset (as set in MotionView or through associated variables) are not applied to the AIN1 variable and this functionality has to be created within the user program.

The analog output is set to indicate motor output velocity by setting the relevant variable to select this 'pre-defined' functionality.

The PositionServo is placed in Velocity mode with internal reference through the relevant variables at the start of the program.

The program uses IREF (internal Reference) as its velocity reference when in internal velocity mode. Therefore Analog input 1 (AIN1) is transferred to IREF within the main program loop.

AIN1 is in volts and IREF in RPS (revolutions per second) so AIN1 must be scaled before it can be applied to IREF. This program uses the analog scaling entered in the MotionView Parameter as this allows the customer to make parametric changes to the scaling from within MotionView without having to alter the user program.

The program applies any offset entered into the MotionView parameter in the same way as for scaling.

Lastly the program applies a dead-band in RPS before the calculated velocity is transferred to the Internal Reference variable and is output to the motor.

The drive will continually execute this code loop until the drive run input is switched off.

An event is used to detect the run / stop input transitioning from run to stop once the drive is in a run condition.

No subroutines are used in this example.

Motor Mechanics:

For the purpose of the demonstration a motor should be used with a disc fitted to the motor shaft so that the user can see the Velocity output executed by the PositionServo.

Fault Handling:

In the event of a fault, the code will be restarted. The operator must switch the drive run /stop input off and on again for the Velocity loop to restart.

I/O:

IN_A3: Safety Enable / stop - Connected to machine safety Guards/Devices

IN_A4: System Run / Stop Input

AIN1: Drive Velocity Reference

AOUT: Drive Output Velocity

Connection:

Figure 1 illustrates the P3 terminals that need to be connected for this example to work.

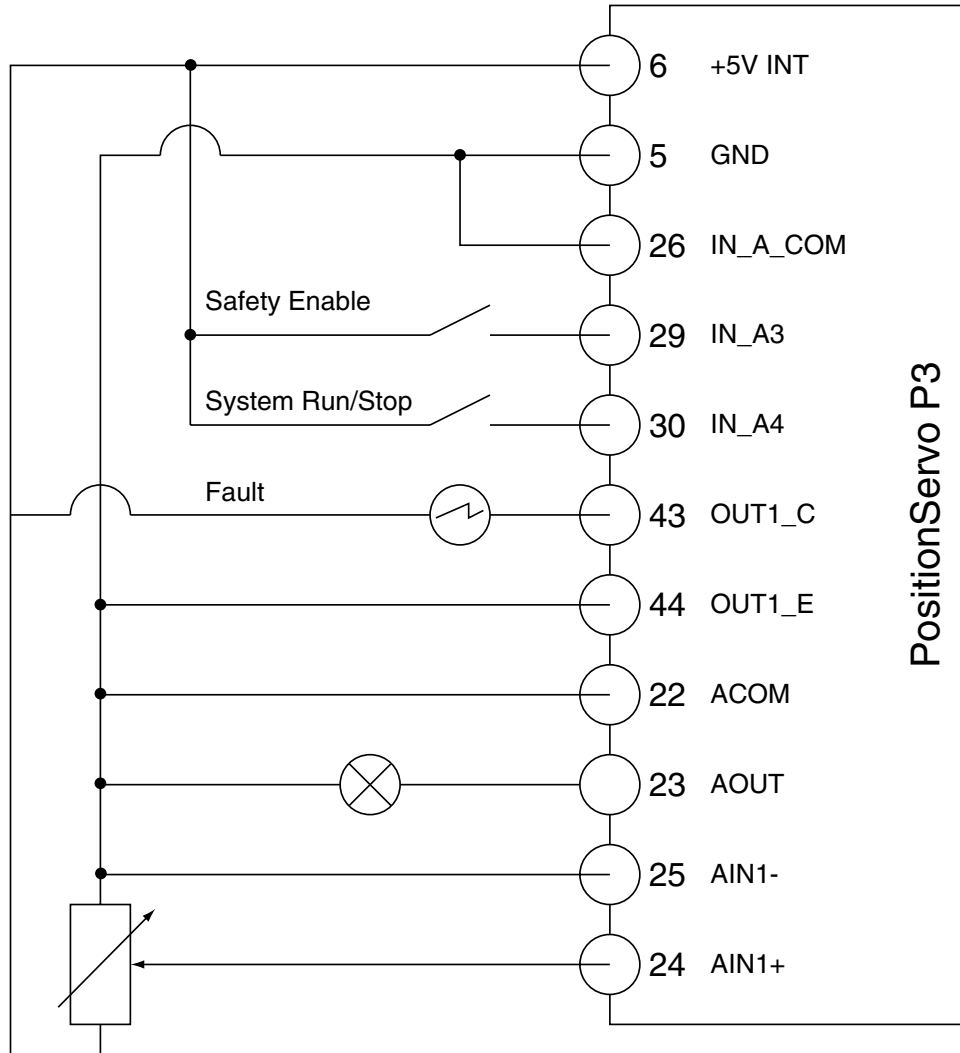


Figure 1: Connection Diagram- Standard Internal Velocity Mode

Example Program:

The program code is color-coded for quick recognition of the various parts of the indexer program. The color coding is not in accordance with, or representative of, any national or international standard.

```

;***** PositionServo User Indexing Program *****
;***** Header *****
;Title      :      PositionServo Training Internal Velocity Example Program
;Author     :      AC Technology International Ltd
;Description :      Program runs from analog voltage reference (AIN1) tranfered to IREF Variable
;           :      Fault indication provided by OUT1
;           :      RUN Input provided on A4
;           :      Simulated encoder passed to master Controller
;Version Number :      V1.0.1
;Date      :      22/11/06
;
;***** I/O List *****
;  Input A1 -      not used
;  Input A2 -      not used
;  Input A3 -      Safety stop
;  Input A4 -      Run/stop
;  Input B1 -      not used
;  Input B2 -      not used
;  Input B3 -      not used
;  Input B4 -      not used
;  Input C1 -      not used
;  Input C2 -      not used
;  Input C3 -      not used
;  Input C4 -      not used
;
;  Output 1 -      Drive Tripped Output (Event)
;  Output 2 -      not used
;  Output 3 -      not used
;  Output 4 -      not used
;
;  Analog In 1 -   Analog Speed Reference
;  Analog In 2 -   not used
;  Analog Out  -   Speed Reference to Controller - Assigned in Parameter
;
;  Encoder Out -   Past to External Customer Controller (If Required)
;
;
;***** Events *****
;Event to detect RUN input (A4) going off
Event Run_Input In_A4 == 0
  Disable
  Jump Program_Start
Endevent

;***** Initialize and Set Variables *****
; Define Constants and Variables. Assign I/O and Initialize Variable Values

UNITS = 1 ; Units in RPS
Define Vel_Calc V0 ; Define Variable for Velocity Calculations

Out1 = 1 ; Fault status 1 = ok, 0 = Fault
Var_Enable_acceldec = 1 ; Enable Ramps
Var_Accel_limit = 300000 ; Set Accel Ramp Rate
Var_decel_limit = 300000 ; Set Decel Ramp Rate

VAR_REFERENCE = 1 ;set Reference to Internal
VAR_DRIVEMODE = 1 ;Set Operating mode to velocity mode
VAR_ENABLE_SWITCH_TYPE=0 ;enable switch function set to "Inhibit"
VAR_AOUT_FUNCTION = 3 ; Set Analog Output to indicate Motor Velocity

```

```

;***** Main Program *****
PROGRAM_START:
Event Run_Input off           ; Turn off Checking for A4 Run Input (Run Input Always off Here)

IREF = 0                       ; Reset the Internal Reference Variable of any Previous Reference

Wait While IN_A4 == 0         ; Wait for run input to come on
Enable                        ; Enable Drive
Event Run_Input On           ; Turn on event to detect run input going off

VELOCITY_LOOP:
; Update the velocity reference from Ain1

Vel_Calc = Ain1                ; Copy Ain1 to Vel_Calc Variable, Vel_Calc now in volts
Vel_Calc = Vel_Calc + (Var_Ain1_Offset / 1000) ; Add MotionView Offset for Ain1 (offset in mV hence /1000)
Vel_Calc = Vel_Calc * (Var_velocity_scale / 60) ; Apply Scaling set in MotionView (V/kRPM)
;                               ; to determine speed reference, divide by 60 for RPS

; Vel_Calc = Vel_Calc * -1      ; Add Line to Reverse Direction of motor

; If statement Applies deadband in RPS
If Vel_Calc <= 0.6 && Vel_Calc >= -0.6      ; Deadband +/- 0.6 RPS = 360RPM
    Vel_Calc = 0                          ; Zero Velocity Reference
Endif

IREF = Vel_Calc                  ; Transfer Calculation to Internal Velocity Reference

GOTO VELOCITY_LOOP              ; Return and Re-calculate
END

Fault_Reset:
    wait until in_a4 == 0           ; Wait until Drive run input is disabled
    out1 = 1                       ; Fault status: 1 = ok, 0 = Fault
Goto Program_Start

;***** Sub-Routines *****
; Enter Sub-Routine code here

;***** Fault Handler Routine *****
; Enter Fault Handler code here
ON FAULT

    Out1 = 0                       ; Fault status: 1 = ok, 0 = Fault
    var_exstatus = var_exstatus | 0x00800000 ; Do not reset Fault on Exiting Handler
    Resume Fault_Reset

ENDFAULT

```