Application Example 0026

PositionServo Sample Program - Servo Valve Control (Analog Ref)

Concept:

This sample program demonstrates the use of the PositionServo Drive Analog Voltage Reference (Analog Input) to control the absolute position of the motor shaft and hence control the open / closed position of a precision Servo Valve system. Servo Valve control is growing in popularity due to it offering increasingly fast response times, high repeatability, high accuracy, and reduced maintenance compared to most pneumatic or hydraulic solutions.

This example program takes the voltage reference on Analog input 1 (Ain1), converts it to a position reference, and applies it to the drive's TPOS (Target Position) register. The result is that the orientation (phasing) of the motor shaft is directly controlled by the voltage reference. Typically the control voltage would be supplied by some form of analog controller but could be derived from inputs direct to the indexer program for manual control.

This example applies the concept to precision analog valves that operate from a rotational actuator coupled directly to the motor (or gearbox) shaft; however the software program is easily adapted to linear valve actuators, or to other applications that require positioning from an analog reference. The example can also be adapted to allow set-up and control via an HMI screen.

System Description:

A picture of a valve connected to a servo motor is illustrated in Figure 1. The motor shaft is connected via a gearbox to the valve actuator which needs to rotate between the maximum and minimum angle entered into the Indexer program (fully open and fully closed) with reference to an absolute home position.

For this example, the following mechanical limits and values are considered:

Valve position fully open = 100°

Valve position fully closed <10°

Gearbox Ratio = 10:1

Changes in valve position should occur smoothly with defined acceleration, and deceleration.

Figure 1: Valve Connected to Servo Motor

Software Description:

The program starts with the execution of a homing routine. The homing routine moves the valve to the position where it is mechanically prevented from moving further. This does not necessarily represent the valve being in the fully closed position. In this example the home position is considered to be 10 degrees beyond the closed position. This is achieved (without homing sensor) by monitoring current demand (using PhCur variable) on the drive and detecting the current increase when the valve is mechanically locked. A separate example of this method of homing (AE0024 – Homing to Hard Machine Limits) is available in the Lenze- AC Tech Technical Library.

Once homed, further position moves are relative to that home or zero position.

Homing code is located in a subroutine in the subroutine section at the bottom of the program.

The homing routine will wait until the analog position reference is returned below the minimum value before exiting to ensure the valve does not immediately open on execution of the main program.

Following successful execution of the homing routine program execution is returned to the main program and halted (using halt statement). Execution of the valve position control is handled by two events (event driven code) that restart executing when triggered.

The first event is to perform the update (input and scaling) of the reference present on the analog input (Ain1). This event is named Anal_Input_Update.

The second event handles the application of the reference (generated by event Anal_Input_Update) to the TPOS register, and controls rate of changes of velocity (ramps), and position limits for the reference. This event is named TPOS_Update.

Four other events are contained in the code. One event monitors the input request to re-home the drive (IN_A1), another monitors the run signal to the PositionServo drive (IN_A4), a third turns on an indicator when the analog reference drops below a minimum value, and the fourth turns on an indicator that shows the valve is in the fully closed position.

In order to respond immediately to changing position references on the analog input (from the controller), this program is required to write directly to the TPOS (Target Position) register of the PositionServo drive.

NOTE - Writing to the TPOS variable

When writing directly to the TPOS register, the 'Motion Profiler' (used for such commands as MoveD, MoveP, Move While, Move Until, etc) within the PositionServo is bypassed. The profile generator normally applies the acceleration, deceleration, and maximum velocity to the motion command as specified in the ACCEL, DECEL, and MAXV variables. Because the profile generator is by-passed when writing to the TPOS register, the Accel, Decel, and MaxV variables have no effect on the resultant motion that is produced. When writing directly to the TPOS register in this way, code should be included that integrates the current and demanded Position to create velocity ramps and to prevent position error trips. This is demonstrated in this example with the TPOS_Update event.

Refer to Appendix 1 for a basic flow chart of program execution.

Refer to Appendix 2 for a block diagram of program functions and variables within the two events that control valve position and for a description of how they link together.

Fault Handling:

In the event of a fault, the code will be restarted. The operator must switch the system run / stop input (A4) to the off position. Once this is done the fault reset input (C3) should be activated and released. Programming execution will then be restarted with execution of the homing routine.

Inputs and Outputs:

Input A1 is the request to re-home input. Once the drive has been homed and the drive is operating (enabled) an event will monitor this input and cause a jump back to the homing procedure if the input is activated.

Input A3 is still the safety enable input. Input must be present before program will try to enable drive. Deactivation while program is running will cause fault 36.

Input A4 is the system run input. It is the primary input for enable / disable of the drive and running of the motor. During program execution input A4 is monitored by an event and de-activation will cause the drive to disable and to wait until the input is re-activated. The status of this input is indicated by output 2 (see below).

Input C3 is used to reset / recover from a system fault. The system run input must be set to off before this switch can be used to restart program execution. The trip code for the fault will remain on the display until the next drive enable.

Output 1 shows the homing routine has been completed and the drive is homed.

Output 2 shows the current status of the system run input (A4), more information is given in Table 1.

Output 3 indicates the valve is within the specified limits of the closed position (is effectively closed).

Output 4 indicates the status of the analog input Ain1. When lit this indicator shows that the input is below the minimum voltage level set (Valve is commanded to close completely).

Table 1: Summary of Status Indicators

Indicator	Condition	Cause
Display	F_36	Input A3 (safety Input) deactivated during operation
Display	Dis	Drive disable, Waiting for A4 to Start
Output 1	Output on	Homing has been completed on the Valve
Output 2	Output off	No System Run Input to Drive
Output 2	Output on	System Run Input to Drive Active
Output 2	Output Flashing Fast	Program waiting for Safety Enable to be turned on
Output 2	Output Flashing Slow	Program waiting for Safety Enable to be turned off
Output 3	Output off	Valve not within tolerance of the full closed Position
Output 3	Output on	Valve within tolerance of the full closed Position
Output 4	Output off	Analog Reference Below the Minimum Value Set
Output 4	Output on	Analog Reference Above the Minimum Value Set

Normal Operating Sequence for Test and Simulation:

1. Ensure all Inputs are off.
2. Run the Indexer Program.
3. Switch on the Safety Enable (A3).
4. Switch on System Run Input (A4) – Drive will start homing.
5. When drive hits hard limit homing will be completed.
6. In order to exit the homing routine analog input 1 must be below the minimum voltage limit. Reduce analog input 1 until output 4 indicator lights.
7. Drive shaft position can now be controlled directly by analog input 1 between the maximum and minimum position limits set (as referenced to the gearbox shaft).
8. Re-homing of the axis can be request at any time by toggling the request homing input (A1).
9. The drive can be disabled by system run input (A4).

I/O:

IN_A1: Homing Request
IN_A3: Safety Enable / stop - Connected to safety Guards/Devices
IN_A4: System Run / Stop Input
IN_C3: Reset from System Fault

Out1: Homing Complete
Out2: System Run Input Status
Out3: Valve Closed Indicator
Out4: Analog reference off indicator

Connection:

Figure 2 illustrates the P3 terminals that need to be connected for this example to work.

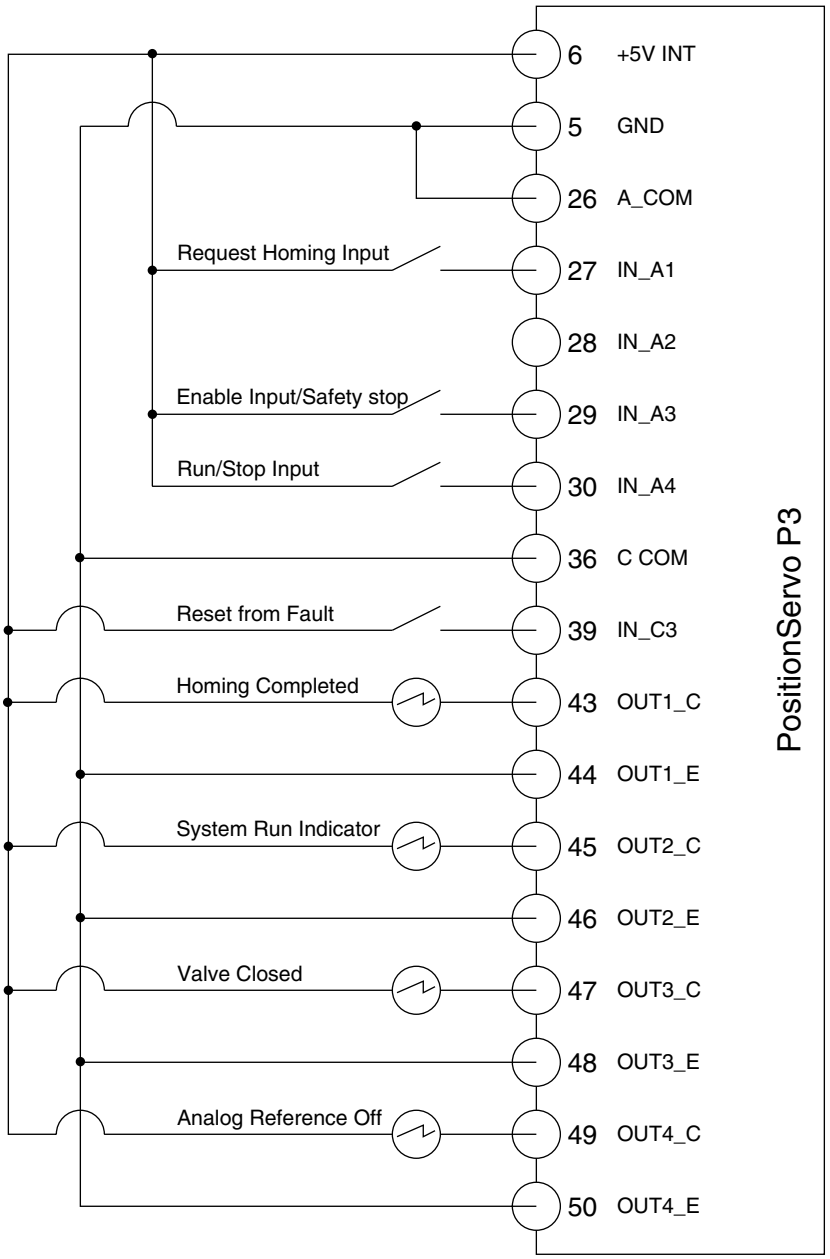

Figure 2: Connection Diagram- Servo Valve Control (Analog Ref)

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 Servo Valve Control (Analog Ref) Example Program
;Author            :    Lenze-ACTech International Ltd
;Description       :    Program uses Analog Voltage Reference (AIN1) to determine Position (phasing)
;                  :    of the Motor Shaft which is coupled to and controls the position (opening) of
;                  :    a precision valve. Programmer can set Positioning and Voltage limits.
;                  :    Motion can be Scaled to Gearbox shaft or motor shaft.
;Version Number    :    V1.0.0
;Date              :    16/12/08
;
;***********************************************  I/O List  *********************************************************
;    Input A1      -    Homing Request
;    Input A2      -    not used
;    Input A3      -    Safety Enable
;    Input A4      -    System Run
;    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      -    Fault Reset
;    Input C4      -    not used
;
;    Output 1      -    Homing Completed
;    Output 2      -    System Run Indicator
;    Output 3      -    Valve Closed
;    Output 4      -    Analog Reference Off
;
;    Analog In 1   -    not used
;    Analog In 2   -    not used
;    Analog Out    -    not used
;
;    Encoder Out   -    not used
;
;
;
;******************************* Initialize and Set Variables ***************************************************
; Define Constants and Variables. Assign I/O and Initialize Variable Values

; Define Constants
Define Gear_Ratio 10                          ; Enter Gearbox Ratio 10:1 (Set to 1 for no gearing)
Define Home_Velocity 30                       ; Velocity for homing Move: deg/s - Gearbox Shaft
Define Normal_Acceleration 1440               ; Acceleration for standard Moves: deg/s/s - Gearbox shaft
Define Homing_Current 1                       ; Define Current limit to be used during Homing to 'hard limit': Amps
Define Home_Dectection_Threshold 0.5          ; Define Detection Threshold for drive hitting hard limit during homing: Amps

Define Max_Angle 100                          ; Set / Define maximum angle of Gearbox shaft rotation: Degrees
Define Min_Angle 10                           ; Set / Define minimum angle of Gearbox shaft rotation: Degrees
Define Angle_Range (Max_Angle - Min_Angle)    ; Determine (Calculate) Valid Gearbox shaft Angle Range

Define Max_Volts 4.5                          ; Define Maximum Valid Voltage on Analog Input
Define Min_Volts 0.5                          ; Define Minimum Valid Voltage on analog Input
Define Volts_Range (Max_Volts - Min_Volts)    ; Determine (Calculate) Valid Volts Range

Define Anal_IP_Loop_Time 10                   ; Define Loop rate for updating reference from Analog input 1 (mS)
Define TPOS_Update_Loop_Time 15               ; Define Loop rate for update of Target Position Loop (mS)

Define PosLimit_Saftey_Margin 0.75
; Set safety limit for Position Error, percentage of maximum to ensure trip doesn't occur

; Define Variables
Define Anal_IP_Loop_Ratio V1
; Define Ratio between New and Previous Value of Analog Reference input (integration rate)
Define Target_Pos_New V2                      ; Variable to Store New commanded Position Value from Analog Input Update Loop
Define Target_Pos_Current V3                  ; Variable to Store Previous commanded Position Value from Analog Input Update Loop
Define Recorded_TPOS V4                       ; Record for Value of TPOS when TPOS update Loop is executed
Define Pos_Error_Check V5                     ; Variable stores calculation of Position Error
Define Copy_Pos_Error_Check V6                ; Variable stores calculation of Position Error prior to it being manipulated
Define PLimit_PosError_Units V7               ; Variable Stores maximum permissible level for position error
;                                             ; generated by updates to TPOS register
Define Record_Nominal_Current V8              ; Nominal current setting for normal operation - following homing
Define Record_Peak_Current16 V9               ; Peak current setting at 16kHz O/P Freq for normal operation - following homing
Define Record_Peak_Current8 V10               ; Peak current setting at 8kHz O/P Freq for normal operation - following homing
```

```
; Define Digital I/O
Define Homing_Request In_A1                    ; Define Label for Digital Input A1, Homing Request Input
Define Safety_Enable In_A3                     ; Define Label for Digital Input A3, Safety Device Input
Define System_Run In_A4                        ; Define Label for Digital Input A4, system Run Input
Define Fault_Reset_IP In_C3                    ; Define input to allow reset from fault
Define Home_Completed Out1                     ; Define Label for Digital Output Out1, Homing Completed Output
Define System_Run_Indicator Out2               ; Define Label for digital output out2, System run indicator - on = run signal present
Define Value_Closed Out3                       ; Define Label for digital output out3, indicates valve in closed position
Define Analog_Ref_Off Out4                     ; Define Label for digital output out4, Analog Reference Below Minimum Value

; Set Initial Values
Accel = 720                                    ; Accel for Homing (Not used for Positions written directly to TPOS)
Decel = 720                                    ; Decel for Homing (Not used for Positions written directly to TPOS)
Anal_IP_Loop_Ratio = 0.9                       ; Set Anal Loop Ratio, New Value = (Current Value * ###) + (Old Value * (1 - ###))
Target_Pos_New = 0                             ; Reset (Flush) New Position Value from Analog Input Update Loop
Target_Pos_Current = 0                         ; Reset (Flush) Current Position Value from Analog Input Update Loop

Var_Reference = 1                              ; Set Reference to Internal
Var_DriveMode = 2                              ; Set Operating mode to Position mode
Var_Enable_Switch_Type = 0                     ; Enable switch function set to "Inhibit"

Units = (1 / 360) * Gear_Ratio                 ; Put units into Degree's (Gearbox shaft) Set Gear Ratio to 1 for no gearing

; Calculate maximum permissible position error in pulses - save to variable
PLimit_PosError_Units = (Var_PLimit_PosError  *  PosLimit_Saftey_Margin) / (Var_M_Encoder * Units)

; Calculate Max increment that can be applied to TPOS Register to not exceed max permissible velocity
Define Position_Increment (Normal_Acceleration) * TPOS_Update_Loop_Time * 0.001


;********************************************** Events ***************************************************************

; Event to Detect valve in the fully closed position and turn on output (out2)
Event Valve_Closed_Out Time 250                ; Run Event every 250mS
     IF     (APOS - Min_Angle) < 1             ; If gearbox shaft within 1 degree of closed position
          Value_Closed = 1                     ; Turn on output (Out2) to indicate valve fully closed
     Else                                      ; If gearbox shaft greater than 1 degree of closed position
          Value_Closed = 0                     ; Turn off output (Out2) to indicate valve not fully closed
     Endif
EndEvent

; Event to Detect Analog reference below minimum value and on output (out4)
Event Anal_Min_Ref Time 250                    ; Run Event every 250mS
     IF Ain1 < Min_Volts                       ; If reference below the minimum threshold set
          Analog_Ref_Off = 1                   ; Turn on output (Out4) to indicate no analog voltage reference
     Else                                      ; If reference above the minimum threshold set
          Analog_Ref_Off = 0                   ; Turn on output (Out4) to indicate analog voltage reference present
     Endif
EndEvent

; Event to Detect Re-home Instruction being Applied during normal operation
Event Rehome Input Homing_Request Rise         ; If Input A1 Rising Edge Detected (0-1)
     Jump Homing                               ; Jump to Label 'Homing'
EndEvent


; Event to Detect Run Instruction being removed during normal operation
Event Run_IP_Deactivate Input System_Run Fall  ; If INput A4 Falling Edge Detected (1-0)
     Jump Program_Start                        ; Jump to Label 'Program_Start'
EndEvent


; Event to Read present value on Analog Input and Calculate New Target Position
Event Anal_Input_Update Time Anal_IP_Loop_Time ; Timed event, Time taken from Previously Defined Constant

; *** VOLTAGE LIMITS FUNCTION ***
                                               ; Read Analog Ref, scale to voltage and angular permissible ranges
     Target_Pos_New = ((Angle_Range / Volts_Range) * (Ain1 - Min_Volts)) + Min_Angle
; *** END FUNCTION ***

; *** POSITION LIMITS FUNCTION ***
     If Target_Pos_New > Max_Angle
; If new Calculated Position is Greater than Maximum allowed Position then set to Maximum
          Target_Pos_New = Max_Angle           ; Replace Target_Result with Maximum Position
     EndIf

     If Target_Pos_New < Min_Angle
; If Calculated Position is Less than Minimum allowed Position then set to Minimum
          Target_Pos_New = Min_Angle           ; Replace Target_Result with Minimum Position
     EndIf
; *** END FUNCTION ***
```

```
; *** POSITION REFERENCE INTEGRATOR ***
; Integrate new commanded position (calculated above) with previous applied Position using predefined ratio
    Target_Pos_Current = (Target_Pos_Current * Anal_IP_Loop_Ratio) + (Target_Pos_New * (1 - Anal_IP_Loop_Ratio))
; *** END FUNCTION ***

    Jump Main_Loop                                  ; Return to Main Program
EndEvent


; Event to take position commands generated in Anal_Input_Update event, verify with position limits and apply to TPos register
Event TPOS_Update Time TPOS_Update_Loop_Time        ; Timed event, Time taken from Previously Defined Constant

    Recorded_TPOS = TPOS                            ; Record current value of TPOS - to be used during Event

; *** TARGET POSITION DIRECTION COMPARATOR ***
; *** RAMP UP / DOWN VELOCITY FUNCTION ***
; If a new target position (from Anal_Input_Update event) is greater than the current target position
    If Target_Pos_Current > Recorded_TPOS
        Recorded_TPOS = Recorded_TPOS + Position_Increment      ; Increment Recorded value by max value (create velocity limit)
        If Recorded_TPOS > Target_Pos_Current                   ; If result greater than the new commanded Position
                Recorded_TPOS = Target_Pos_Current              ; Set Recorded value to the actual commanded Position
        Endif

; If a new target position (from Anal_Input_Update event) is less than the current target position
    Else
        Recorded_TPOS = Recorded_TPOS - Position_Increment      ; Decrement Recorded value by max value (create velocity limit)
        If Recorded_TPOS < Target_Pos_Current                   ; If result Less than the new commanded Position
                Recorded_TPOS = Target_Pos_Current              ; Set Recorded value to the actual commanded Position
        Endif
    EndIf
; *** END FUNCTION ***
; *** END FUNCTION ***

; *** POSITION ERROR LIMITS FUNCTION ***
; Check max position error against limit set, do not increment / decrement target position if it will exceed position error limit
    Pos_Error_Check  = Recorded_TPOS - APos             ; Calculate Position Error that will result from applying new target position
    Copy_Pos_Error_Check = Pos_Error_Check              ; Store copy of Pos_Error_Check before it is manipulated (below)

    If Pos_Error_Check < 0                              ; If new Position Error is Negative Value
                Pos_Error_Check = -Pos_Error_Check      ; Invert Position Error (make Positive)
    EndIf

; *** UPDATE TARGET POSITION FUNCTION ***
    If Pos_Error_Check > PLimit_PosError_Units          ; If new Position Error is greater than is permissible
        If Copy_Pos_Error_Check > 0                     ; If Position Error is Positive
                TPOS = APOS + PLimit_PosError_Units     ; Set new target position of actual position + limit so max is not exceeded
        Else                                            ; If Position Error is Negative
                TPOS = APOS - PLimit_PosError_Units     ; Set new target position of actual position - limit so max is not exceeded
        Endif
    Else                                                ; If new Position Error is less than is permissible
                TPOS = Recorded_TPOS                    ; Apply the calculated new target position to the target position register
    Endif
; *** END FUNCTION ***
; *** END FUNCTION ***

    Jump Main_Loop                                  ; Return to Main Program
EndEvent


;****************************************** Main Program ********************************************************************

    Outputs = 0                                     ; Ensure all outputs off at start of program
    While System_Run == 1                           ; Ensure Run input is Removed before main program is allowed to start
        System_Run_Indicator = !System_Run_Indicator  ; Toggle System run output slowly (indicate waiting for deactivation)
        Wait time 1000                              ; Time for output 'blink' 1000mS (Slow)
    EndWhile

Program_Start:                                      ; Program Start Label
    Disable                                         ; Disable Drive Output
    Event Anal_Input_Update off                     ; Ensure Event to read analog input and update position reference is off
    Event TPOS_Update off                           ; Ensure Event to check and apply new target positions is off
    Event Run_IP_Deactivate Off                     ; Event disable detect Run input going off
    Event ReHome Off                                ; Event disable detect Home Request input coming on

    While System_Run == 0                           ; Wait for Run input to be present before continuing
        System_Run_Indicator = !System_Run_Indicator  ; Toggle System run output quickly (indicate waiting for activation)
        Wait time 200                               ; Time for output 'blink' 200mS (Fast)
    EndWhile
    System_Run_Indicator = 1                        ; Turn on System Run indicator
    Event Run_IP_Deactivate On                      ; Event Enable detect Run input going off

Homing:
```

```
     GoSub Homing_Sub                                ; Go to Homing Subroutine, homing completed on return

     Event Valve_Closed_Out on                       ; Turn on Event to detect and indicate when valve in fully closed position
     Event Anal_Min_Ref on                           ; Turn on Event to detect when analog signal drops below minimum value
     Event Anal_Input_Update on                      ; Turn on Event to read analog input and update position reference
     Event TPOS_Update on                            ; Turn on Event to check and apply new target positions

Main_Loop:                                           ; Section to halt program execution and have event driven processing
     Halt                                            ; Halt main program execution (with events running)

END                                                  ; End Command - Never Reached

Fault_Code:                                          ; Fault handler section
     While System_Run == 1                           ; Ensure Run input is Removed before main program is allowed to restart
          System_Run_Indicator = !System_Run_Indicator ; Toggle System run output slowly (indicate waiting for deactivation)
          Wait time 1000                             ; Time for output 'blink' 1000mS (Slow)
     EndWhile
     Wait until Fault_Reset_IP == 1                  ; Wait for Fault reset input to go high
     Wait until Fault_Reset_IP == 0                  ; Wait for Fault reset input to go low
     Goto Program_Start                              ; Return to start of program


;****************************************** Sub-Routines ********************************************************************
Homing_Sub:
     Event Anal_Input_Update off                     ; Ensure Event to read analog input and update position reference is off
     Event TPOS_Update off                           ; Ensure Event to check and apply new target positions is off
     Event ReHome off                                ; Disable Event to detect Home Request input coming on (as already homing)
     Event Valve_Closed_Out off                      ; Turn off Event to detect when valve in the fully closed position
     Enable                                          ; Enable Drive
     Home_Completed = 0                              ; Turn off homing completed indicator
     MaxV = Home_Velocity                            ; set slow velocity for homing - Ensure Safe Velocity for hitting stop position
     Record_Nominal_Current = Var_CurrentLimit       ; Save current settings for Nominal Current (restore after homing)
     Record_Peak_Current16 = Var_PeakCurrentLimit16  ; Save current settings for peak Current 16kHz (restore after homing)
     Record_Peak_Current8 = Var_PeakCurrentLimit     ; Save current settings for peak Current 8kHz (restore after homing)
     Var_CurrentLimit = Homing_Current               ; Set Nominal Current (Torque) for Homing
     Var_PeakCurrentLimit = Homing_Current           ; Set Peak current (Torque) at 8 kHz for Homing
     Var_PeakCurrentLimit16 = Homing_Current         ; Set Peak current (Torque) at 16 kHz for Homing

     move back until PhCur > Home_Dectection_Threshold ; move back at slow velocity until shaft hits stop (closed) position

     Var_CurrentLimit = Record_Nominal_Current       ; Restore current settings for Nominal Current (restore after homing)
     Var_PeakCurrentLimit16 = Record_Peak_Current16  ; Restore current settings for peak Current 16kHz (restore after homing)
     Var_PeakCurrentLimit = Record_Peak_Current8     ; Restore current settings for peak Current 8kHz (restore after homing)

     APOS = 0                                        ; Set actual position register to 0 - Reference Home Position
     Wait while Homing_Request == 1                  ; wait for the homing request signal to be removed
     While Ain1 > Min_Volts                          ; While there is a position ref command on Ain 1 (stops immediate move after homing)
          Analog_Ref_Off = !Analog_Ref_Off           ; Flash indicator to show reference must be removed
          Wait Time 1000                             ; Blink time for output flash slow (1S)
     EndWhile
     Analog_Ref_Off = 1                              ; Turn on Analog reference off indicator
Home_Completed = 1                                   ; Turn on homing completed indicator
     Event ReHome On                                 ; Event Enable detect Home Request input coming on
Return                                               ; Return to Main Program


;************************************ Fault Handler Routine ************************************************************************
ON FAULT                                             ; Fault handling sequence
     var_exstatus = var_exstatus | 0x00800000        ; Do not reset Fault Display on Exiting Handler
     Resume Fault_Code                               ; Resume at Program Start

ENDFAULT
```

Appendix A - Basic Program Flow Chart

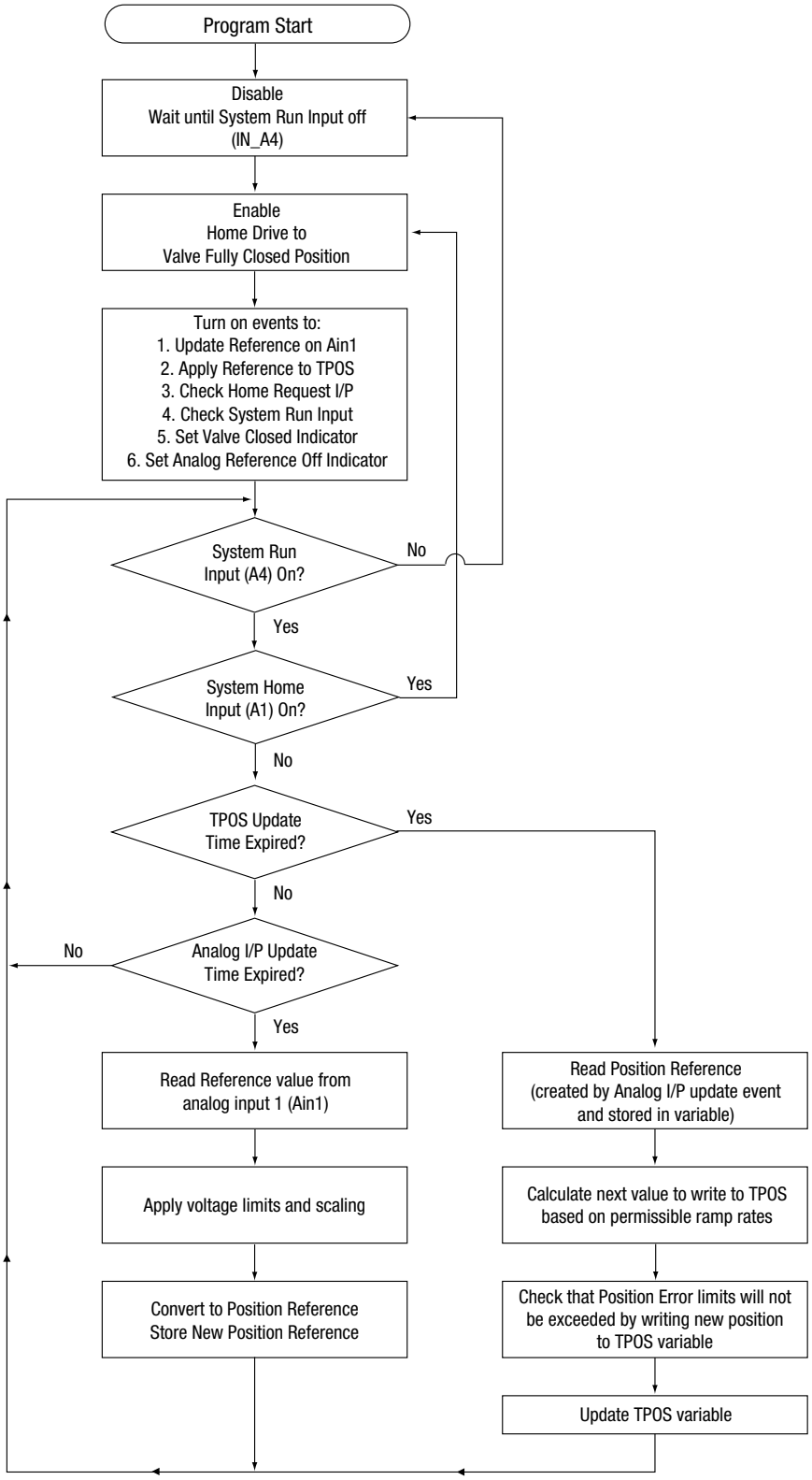

Figure 3: Servo Valve Control (Analog Ref) Program Flow Chart

Appendix B - Block Diagram Valve Positioning Events

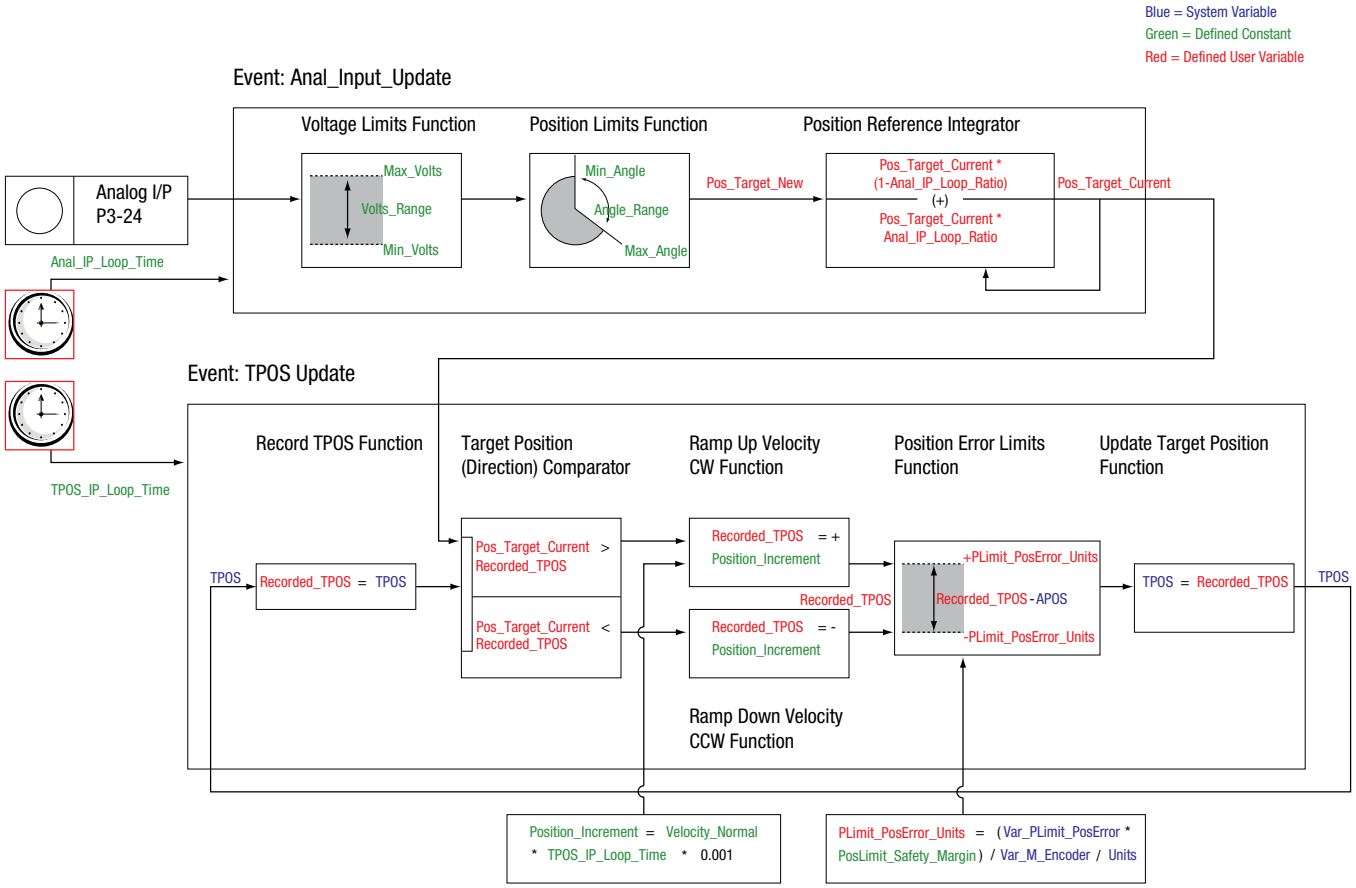

Figure 4: Block Diagram - Valve Positioning Events