

## PositionServo Sample Program - Linear Belt Axis

### Concept:

This is a simple example of a Linear belt axis application. A carriage is mounted onto the belt axis drive system on which a variety of different devices / tools can be attached to allow the customer to perform different applications.

The belt passes around pulleys at either end of the axis. The right pulley is driven directly by the motor / servo mounted to the application.

When triggered (with input A4), the motor will move the cartridge to different locations within the limits set by the hardware limit switches.

The customer would like two modes of operation selectable via an input (Input B2). A manual mode where the customer can move the axis left and right manually by means of push buttons (Inputs B3, and B4), and an automatic mode where the PositionServo will run the motor / belt through a predefined sequence of position moves.

A linear belt axis is shown in Figure 1.

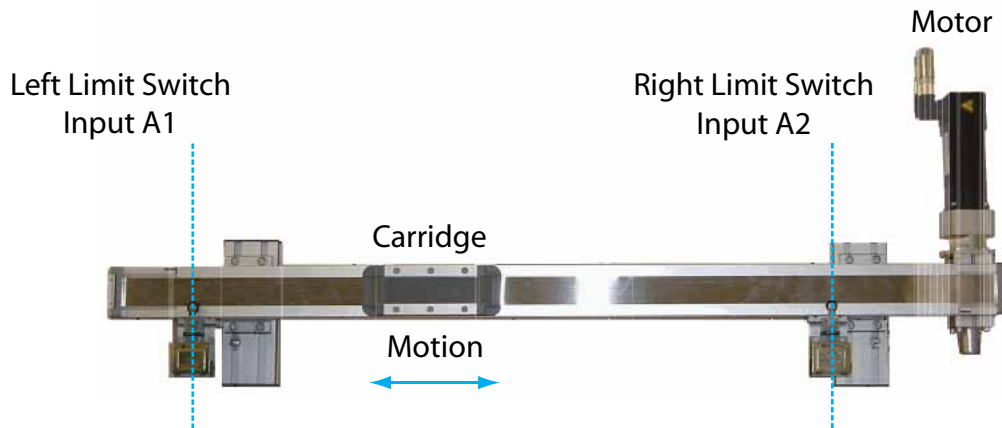


Figure 1: Linear Belt Axis

### Motor Mechanics:

The motor is mounted so that it turns one of the pulleys around which the belt is placed. The carriage is attached to the belt and moved horizontally between the limit switches. Due to some gearing between the motor shaft and the pulley one turn of the motor shaft will result in 1 cm of belt travel.

Acceleration and Deceleration are set independently.

### Homing:

The axis must be homed before the process can begin. The PositionServo uses the limit switch on the left side of the axis (Input A1) to Home the Axis.

After homing, the actual position register should be reset so that future absolute motion commands are reference to this left limit switch. The carriage should be moved off the limit switch so that hardware limit switch functionality can be enabled in the drive.

The application should only return to the home position if the application is required to re-home (Through Input B1).

## Limit Switch Functionality (Inputs A1, and A2):

The limit switches are disabled for homing and re-enabled once homing is complete.

For the purpose of demonstration the limit switches can be activated in normal operation. In manual mode the limit switches can be activated by simply moving the carriage via the inputs until one of the limit switches is triggered. In automatic mode an input to the drive is used to 'force' as positional move that will trigger one of the switches.

When a limit switch is triggered the drive should be capable of resetting the fault and of moving back into the normal operating zone.

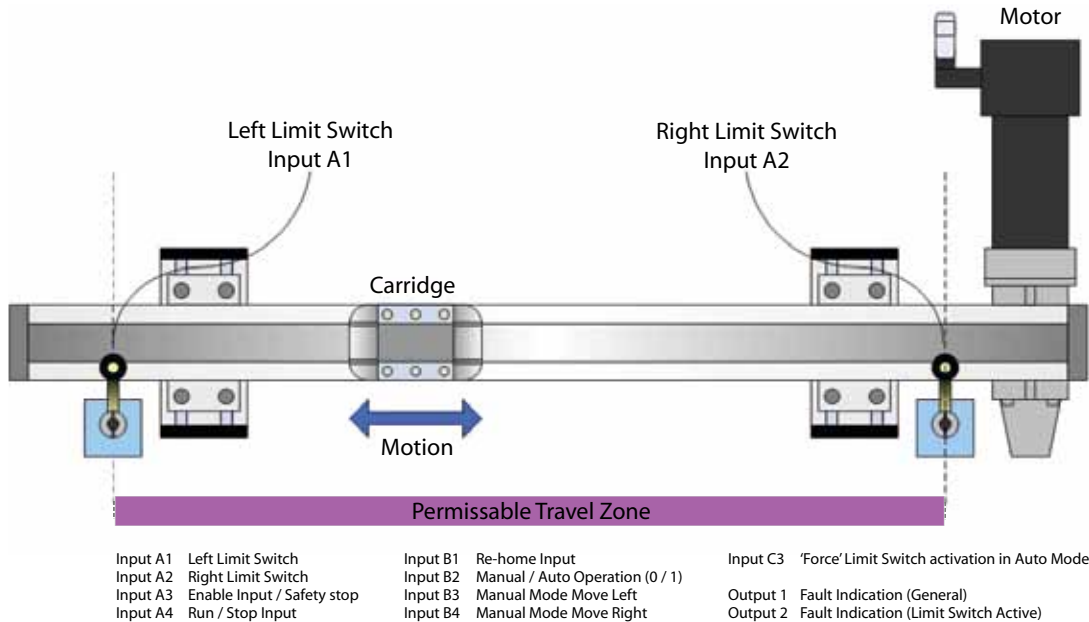


Figure 2: I/O for Linear Belt Axis Application

## Fault Handling:

In the event of a fault, the system should trip and stop program execution, unless the fault is caused by either of the hardware limit switches being triggered. In the case of the hardware limit switches the drive should recover from the fault in the way detailed in the limit switch functionality section.

## I/O:

IN\_A1: Left Limit Switch

OUT1: Fault Indication (General)

IN\_A2: Right Limit Switch

OUT2: Fault Indication (Limit Switch Active)

IN\_A3: Enable Input / Safety stop

IN\_A4: Run / Stop Input

IN\_B1: Re-Home Input

IN\_B2: Manual / Automatic Operation (0/1)

IN\_B3: Manual Mode Move Left

IN\_B4: Manual Mode Move Right

IN\_C3: 'Force' Limit Switch activation in Automatic Mode

Figure 3 illustrates the sample control panel for the Linear Belt Axis application.

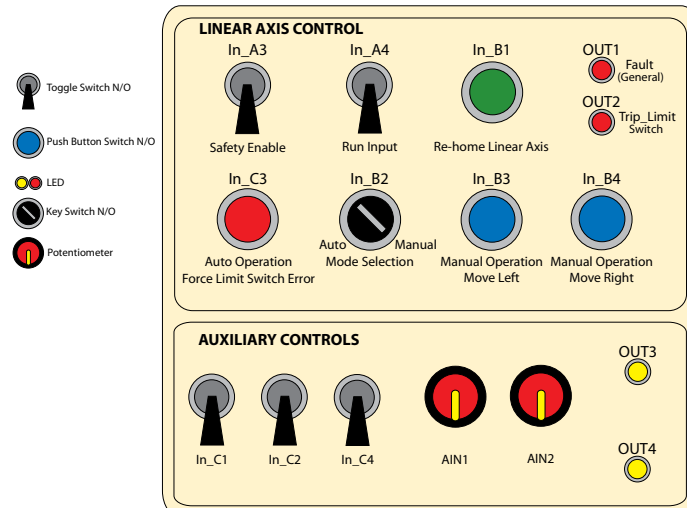


Figure 3: Control Panel for Linear Belt Axis Application

**Connection:**

Figure 4 illustrates the P3 terminals that need to be connected for this example to work.

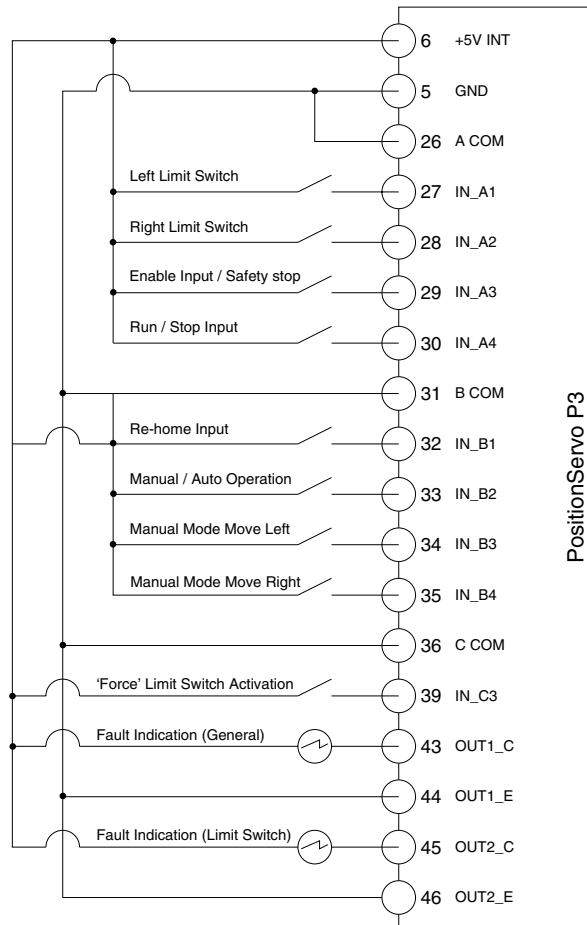


Figure 4: Connection Diagram- Homing to Hard Machine Limits

## 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      :      Linear Belt Axis Exercises Trainig Exercise
;Author     :      AC Technology International Ltd
;Description :      Program Performs Simple Homing Routine followed by Preset Sequence of Moves
;           :      on a Linear Axis. Drive operates in both manual (inputs) and automatic mode.
;           :      In automatic mode the limit switches are triggered intentionally to show
;           :      fault recovery process.
;Version Number :      V1.0.0
;Date      :      28/02/07
;
;***** I/O List *****
; Input A1 -      Left Limit Switch
; Input A2 -      Right Limit Switch
; Input A3 -      Enable Input / Safety stop
; Input A4 -      Run / Stop Input
; Input B1 -      Re-Home Input
; Input B2 -      Manual / Automatic Operation (0/1)
; Input B3 -      Manual Mode Move Left
; Input B4 -      Manual Mode Move Right
; Input C1 -      not used
; Input C2 -      not used
; Input C3 -      'Force' Limit Switch activation in Automatic Mode
; Input C4 -      not used
;
; Output 1 -      Fault Indication (General)
; Output 2 -      Fault Indication (Limit Switch Active)
; Output 3 -      not used
; Output 4 -      not used
;
; 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

UNITS = 1          ; Units in RPS
Accel = 50        ; Accelerate 50 RPS/S
Decel = 50        ; Decelerate 50 RPS/s
Define Limit_Switch_Left In_A1 ; Define IN_A1 as Left Limit Switch
Define Limit_Switch_Right In_A2 ; Define IN_A2 as Right Limit Switch
Define Safety_Enable IN_A3      ; A3 Hardware Enable Input
Define Run_Input In_A4          ; Define IN_A4 as system Run / Stop Input (command)
Define Homing_Request In_B1     ; Define IN_B1 as Input to Request Repeat of Homing Routine
Define Man_Auto_Switch In_B2    ; Define In_B2 to Switch from Manual to Automatic Operation
Define Man_Left_Move In_B3      ; Manual Operating Mode - Input Causes Left
Define Man_Right_Move In_B4     ; Manual Operating Mode - Input Causes Right
Define Auto_Error_Switch In_C3  ; Automatic Mode - Inputs causes a move that will trigger a Limit Switch

Define General_Fault Out1      ; Define output to indicate general faults (not limit switches)
Define Limit_Fault Out2       ; Define output to indicate Limit Switch Triggered fault

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"

Define Current_Mode V0        ;Stores Variable that Records present mode of operation - Man or Auto
Current_Mode = Man_Auto_Switch ;Set Present Mode Variable to equal current value of Mode selection switch

```

```

;***** Events *****
Event Homing_Call Input Homing_Request Rise ; Events looks for Homing Request Input coming on and Jumps to Homing Code
    jump Homing
Endevent

Event Run_Input_Off Input Run_Input Fall ; Events looks for Run Input going off and Jumps to Homing Code
    jump Program_Start
Endevent

Event Mode_Change Current_Mode <> Man_Auto_Switch ; Events looks a change in the Mode Selection Switch
    Current_Mode = Man_Auto_Switch ; Update Mode selection Variable with New Value
    jump Mode_Select
Endevent

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

PROGRAM_START:
Outputs = 0 ; Turn all Outputs Off
Disable ; Disable Drive / Make sure Drive is disabled
Event Homing_call off ; Switch off event to check for homing request input
Event Mode_Change off ; Switch off event to check for change of operating mode Input
Event Run_Input_Off Off ; Turn off event to detect system Run input switching off

Wait While Safety_Enable == 0 ; Wait for drive safety enable to be active - Safe Condition
wait while Run_Input == 0 ; Wait for system command to run
Event Run_Input_Off On ; Turn on event to detect system Run input switching off

Homing: ; Homing Routine, Performed at Start Up and after Homing Routine Request input goes high
Disable ; Disable because setting HLS function when enabled causes trip
VAR_HLS_Mode = 0 ; Defines Functionality of Hardware Limits as 'Disabled' for Input Switches A1 and A2
Enable ; Re-enable after setting HLS functionality
Velocity Off ; Turn Velocity Mode Off
Event homing_call off ; Switch off event to check for homing request input
maxv = 10 ; set slow velocity for homing
move back until Limit_Switch_Left ; move back at slow velocity until homing sensor is activate
maxv = 2 ; set very slow velocity
move until Limit_Switch_Left == 0 ; move forward off homing sensor
maxv = 50 ; Set velocity back to normal speed
MoveD 20 ; Move to start position in center of axis
APOS = 0 ; Set actual position register to 0
wait while Homing_Request == 1 ; wait for the homing request signal to be removed
Event Homing_call on ; Switch on event to check for homing request input
Disable ; Disable because enabling HLS function when enabled causes trip
VAR_HLS_Mode = 1 ; Defines Functionality of Hardware Limits as 'Enabled' for Input Switches A1 and A2
Enable ; Re-enable after setting HLS functionality
Event Mode_Change on ; Switch on event to check for change of operating mode Input

Mode_Select:
if Man_Auto_Switch == 0 ; If Manual Mode Requested
    Vel = 0 ; Initialise Vel (Velocity) variable to 0
    Velocity ON ; Turn on Velocity Mode
    Goto Move_Profile_Manual ; Jump to the Manual Control Loop
Else ; If Auto Mode Requested
    Apos = 0 ; Reset the APOS Register
    Velocity OFF ; Turn off Velocity Mode
    Goto Move_Profile_Auto ; Jump to the Automatic Control Loop
Endif
End

```

```

Move_Profile_Auto:           ; Automatic position loop
MoveP 5                     ; move to position 1
wait time 200               ; wait 200 ms
moveP 10                    ; move to position 2
wait time 500               ; wait 500 ms
moveP -15                   ; move to position -3
wait time 1500              ; wait 1500 ms
If Auto_Error_Switch == 1   ; If the 'force hardware limit fault' input is on
    moveP -50                ; Move to Position 4 - Position 4 activates the Right Limit Switch
endif
goto Move_Profile_Auto      ; restart positioning loop
End

Move_Profile_Manual:        ; Basic position loop
if Man_Left_Move == 1 && Man_Right_Move == 0
    Vel = 20                 ; Move Clockwise command - don't change the calculation
endif
if Man_Left_Move == 0 && Man_Right_Move == 1
    Vel = -20                ; Move Counter Clockwise command - change direction of the calculation
endif
if Man_Left_Move == 0 && Man_Right_Move == 0
    Vel = 0                  ; Invalid Input - don't allow movement
endif
if Man_Left_Move == 1 && Man_Right_Move == 1
    Vel = 0                  ; Invalid Input - don't allow movement
endif
Goto Move_Profile_Manual    ; restart positioning loop
End

fault_section:
Velocity Off
Wait While Run_Input == 1   ; wait until enable input is removed
Limit_Fault = 0             ; Switch off Indication for Limit Switch fault

goto program_start          ; restart program

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

;***** Fault Handler Routine *****
; Enter Fault Handler code here
ON FAULT
    if dfaulsts >= 32 && dfaulsts <= 35
        Limit_Fault = 1      ; If limit switches caused fault
        Resume fault_section ; Indicate Limit Switch fault
    else
        General_Fault = 1    ; Indicate general fault
    endif
ENDFEAULT

```