

PositionServo - Linear Axis with Logic Selectable Limit Switches

Concept:

This is a simple example of a linear belt axis application (derived from programming Library application number 7, Linear Belt Axis Example). 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.

This application utilizes code and Events within the Indexer program to create limit switch functionality rather than using the pre-defined function that can be enabled within the drive. The advantages of this are: we can define logic (active high or low) easily for the limit switches; we can use the user fault codes to better indicate the precise fault that has occurred when one of the switches is activated; and we can easily define the sequence of recovery from a limit switch fault for the system.

The application has 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.

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. With Positive Limit switch logic the indexer program moves onto the switch looking for a high-low transition, and then off again looking for a low high transition. For negative logic limit switches this logic is reversed.

After homing, the drive moves to a home position away from the Left Limit switch and the actual position register is reset so that future absolute motion commands are referenced to this home position.

Once homing is done to the left limit switch (the carriage has moved off the limit switch), events to detect further activation of either limit switch are enabled.

There is an input provided to re-home the axis (Input B1).

Limit Switch Functionality (Inputs A1 and A2):

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

Logic for the limit switches is defined by a definition in 'Initialize and Set Variables' section of the indexer program. The line of code at the bottom of this section should be set to False (as shown) for Limit Switches that are triggered when they transition from logic high to logic low:

```
H_Limit_Logic = False ; Set to False (0) for Negative Logic & True (1) for Positive Logic
```

This line of code should be changed to true for Limit Switches that are triggered when they transition from logic low to logic High.

```
H_Limit_Logic = True ; Set to False (0) for Negative Logic & True (1) for Positive Logic
```

Each of the conditions listed in Table 1 is given an Event in order to create the Limit Switch Functionality.

Table 1: Limit Switch Functionality

Switch Logic (H_Limit_Logic)	Left Limit Switch (A1)	Right Limit Switch (A2)	Event Name	Fault Condition
False	Off	On	Neg_Limit_Left_Active	Fault 201
False	On	Off	Neg_Limit_Right_Active	Fault 202
True	On	Off	Pos_Limit_Left_Active	Fault 203
True	Off	On	Pos_Limit_Right_Active	Fault 204

Only two events, either those for positive limit switch functionality, or those for negative limit switch functionality are enabled within the program at any one time and dependent on the setting of the H_Limit_Logic definition (as given in Table 1).

We use the Fault command to create our own system faults based on each of these logic conditions or activations of the limit switches. When we use the fault command to create these fault conditions, program execution is passed to the fault handler and the fault code shown in Table 1 is shown on the drive display for operator reference.

Fault Handling:

The fault handler differentiates between two types of faults, those caused by activation of a limit switch and any other drive fault that occurs. Faults caused by limit switch activation and defined by fault codes 201 thru 204 are handled by a jump to a section of the main program that will attempt to recover the limit switch fault by moving back into the safe area of the axis. All other faults are dealt with by a restart of the program. Output 1 indicates a general fault condition (not caused by limit switch trip) and output 2 shows a fault caused by limit switch activation.

Inputs and Outputs:

Input A1 and Input A2 are used as hardware limit switches left and right respectively. Detection is handled by four events defined in the events section and dependent on the switch logic selected by the programmer.

Input A3 is still the safety enable input. It's present status is indicated by output 3 (see below). Input must be preset before program will try to enable drive. De-activation 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 a wait until the input is re-activated. The present status is indicated by output 4 (see below).

Input B1 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 B2 allows the operator to switch between an automation mode (pre-defined sequence of motion) and a manual mode (inputs used to define motion). The input is monitored by an event and a change of state will cause the event to force a jump to the section of code that performs the newly selected mode.

Input B3 and input B4 are the inputs that define direction of motion when the drive is in manual mode (see above). Inputs are not active in automatic mode. Activation of input B3 will cause motion in the left direction, and input B4 in the right direction. Activation of neither or both inputs simultaneously will result in no motion commanded.

Input C3 is used to start reset / recovery from a limit switch fault. It is only active when the drive is showing one of trips (201 thru 204) on the display.

Output 1 shows a general fault condition on the drive (any fault other than limit switch activation).

Output 2 shows a Limit Switch fault condition on the drive (faults 201 thru 204).

Output 3 indicates the status of the safety enable input A3. This output comes on to indicate the safety enable input is currently off, and goes off when the input is present. This input will flash slowly (1 sec blink) if the program is in a wait condition because this output is not on (program is waiting for this input).

Output 4 indicates the status of the system run input A4. This output comes on to indicate the system run input is currently off, and goes off when the input is present. This input will flash quickly (250mS blink) if the program is in a wait condition because this output is on (program is waiting while this input is on).

It will flash slowly (1 sec blink) if the program is in a wait condition because this output is not on (program is waiting until this input comes on).

Table 2: Summary of Status Indicators

Indicator	Condition	Cause
Display	Display shows '201'	Limit switch Left is active (active low)
Display	Display shows '202'	Limit switch Right is active (active low)
Display	Display shows '203'	Limit switch Left is active (active high)
Display	Display shows '204'	Limit switch Right is active (active high)
Display	F_36	Input A3 (safety Input) deactivated during operation
Display	DiS	Drive disable, Waiting for either A3, A4 or both Inputs
Output 1	Output on	Drive in General Fault condition, see drive display
Output 2	Output on	Drive in Limit Switch Fault Condition, See drive display
Output 3	Output on	No Safety Enable Input to Drive
Output 3	Output Flashing	Slow Program waiting for Safety Enable
Output 4	Output on	No System Run Input to Drive
Output 4	Output Flashing	Fast Program waiting for Safety Enable to be turned off
Output 4	Output Flashing	Slow Program waiting for Safety Enable to be turned on

Normal Operating Sequence for test and simulation:

1. Ensure correct limit switch logic has been selected and limit switches (A1 & A2) are operational.
2. Ensure all Inputs (other than Limit switches are off)
3. Run the Indexer Program
4. Switch on the Safety Enable (A3)
5. Switch on System Run Input (A4) – Drive will start homing
6. Activate Left Limit switch (A1) – Drive will reverse direction
7. Release Left Limit switch (A2) – Drive will move to home position.
8. With the auto/man switch (B2) set to off, the drive is in manual mode. Move left and move right inputs (B3 & B4) can be used to command motion.
9. Switch auto/man switch (B2) to on, drive is now in auto mode and will perform a predefined sequence of moves.
10. Re-homing of the axis can be requested at any time by toggling the request homing input (B1).
11. The drive can be disabled by the system run input (A4).

Simulating and recovering from a Limit Switch Trip:

With the drive in normal operation (see sequence above):

1. Activate either limit switch input (A1 or A2) – Drive will trip on appropriate trip code.
2. System Run indicator output (out4) will flash quickly. Turn off system Run input.
3. Press the reset for the limit switch fault button to start recovery. Drive will begin slow motion away from limit switch and back into safe zone.
4. Release the activation limit switch. Drive will return to ready / disabled status.
5. Activate system run input (A4) to restart program execution. System will be re-homed on enable.

I/O Summary:

IN_A1: Left Limit Switch

IN_A2: Right Limit Switch

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: Activate Reset from a Limit Fault Condition

OUT1: Fault Indication (General)

OUT2: Fault Indication (Limit Switch Active)

OUT3: Safety Enable Input Status

OUT4: System Run Input Status

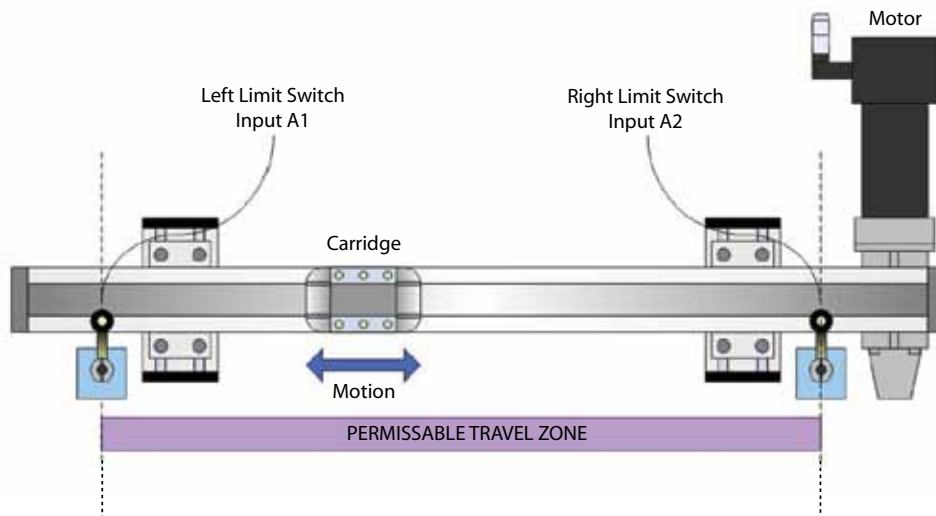


Figure 1: Linear Axis with Logic Selectable Limit Switches

Connection:

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

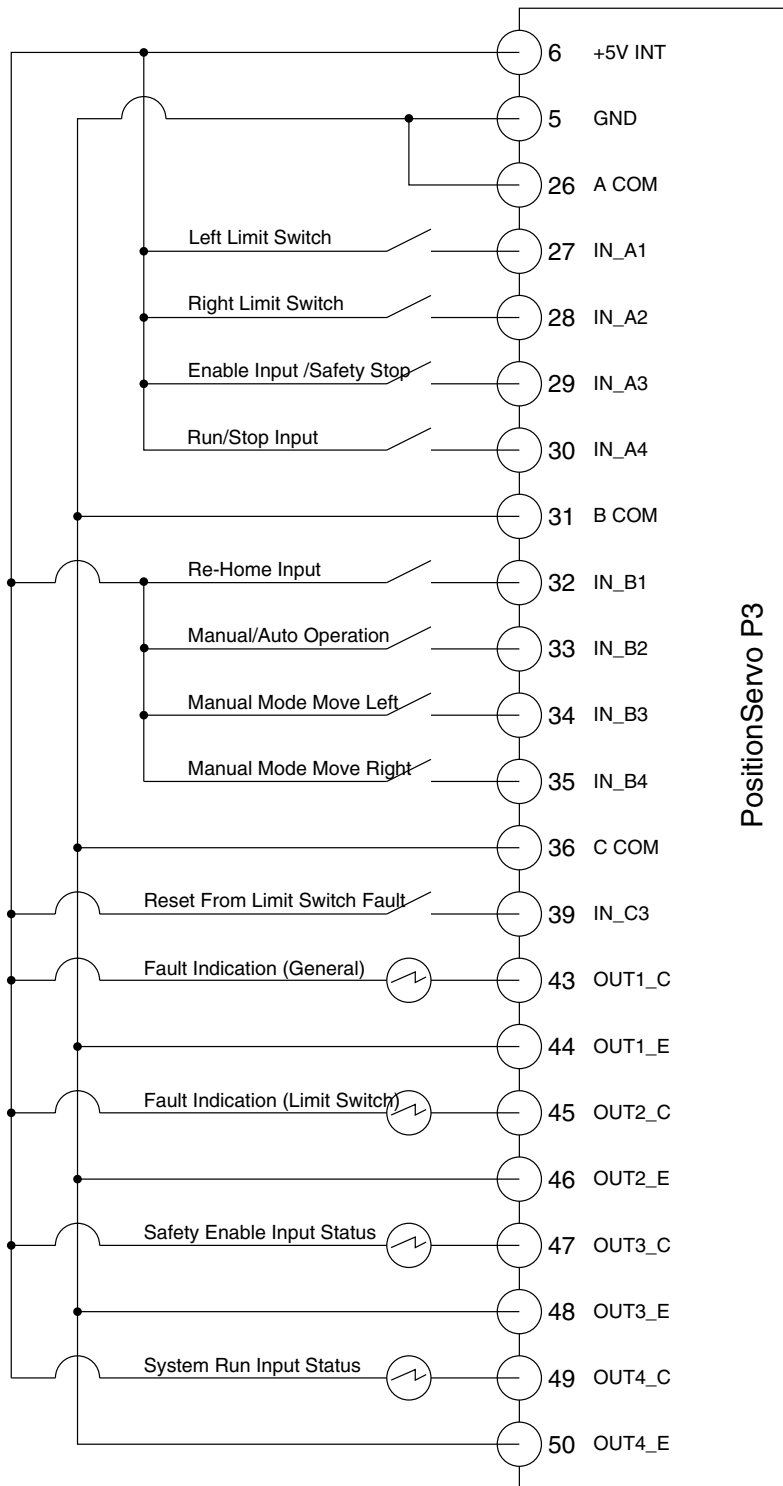


Figure 2: Connection Diagram- Linear Axis with Selectable Limit Switches

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 with Selectable Limit Switch Logic
;Author     :      Lenze_AC Tech 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.
;           :      Limit switches are utilized through events in the Index code and are logic Selectable
;           :      Limit Switches are given user fault code. Recovery from activation is from fault handler
;Version Number :      V1.0.0
;Date       :      10/12/08

;***** 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 - Reset from Limit Switch Trip
; Input C4 - not used
;
; Output 1 - Fault Indication (General)
; Output 2 - Fault Indication (Limit Switch Active)
; Output 3 - Safety Enable Off
; Output 4 - System Run Input Off
;
; Analog In 1 - not used
; Analog In 2 - not used
; Analog Out - not used
;
; Encoder Out - not used
;

;***** Initialize and Set Variables *****

UNITS = 1 ; Units in RPS
Accel = 50 ; Accelerate 50 RPS/S
Decel = 50 ; Decelerate 50 RPS/s
VAR_REFERENCE = 1 ; set Reference to Internal
VAR_DRIVEMODE = 2 ; Set Operating mode to Position mode

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 ; Define IN_A3 as 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 Limit_Switch_Reset 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
Define Safety_IP_Indicator Out3 ; Define output to indicate Safety Enable Input (A3) is off: Out on = Input off
Define System_Run_Indicator Out4 ; Define output to indicate System Run Input (A4) is off: Out on = Input off

Define True 1 ; Define Word True as constant value 1
Define False 0 ; Define Word False as constant value 0

Define Current_Mode V0 ; Stores Variable that Records present mode of operation - Man or Auto
Define H_Limit_Logic V1 ; Variable stores Logic setting for Hardware Limit Switches - Pos or Neg

H_Limit_Logic = False ; Set to False (0) for Negative Logic & True (1) for Positive Logic

```

```

Current_Mode = Man_Auto_Switch      ; Set Present Mode Variable to equal current value of Mode selection switch

;***** Events *****
; LIMIT SWITCH DETECTION EVENTS
; If any logic true in these events then fault forced and program execution given to fault handler

; Event to Handle Left Limit Switch Activation - Negative Logic
Event Neg_Limit_Left_Active Input Limit_Switch_Left Fall
    Fault 201                                ; Cause Fault 201 - Negative Logic Left Limit Switch Activated
EndEvent

; Event to Handle Right Limit Switch Activation - Negative Logic
Event Neg_Limit_Right_Active Input Limit_Switch_Right Fall
    Fault 202                                ; Cause Fault 202 - Negative Logic Right Limit Switch Activated
EndEvent

; Event to Handle Left Limit Switch Activation - Positive Logic
Event Pos_Limit_Left_Active Input Limit_Switch_Left Rise
    Fault 203                                ; Cause Fault 203 - Positive Logic Left Limit Switch Activated
EndEvent

; Event to Handle Right Limit Switch Activation - Positive Logic
Event Pos_Limit_Right_Active Input Limit_Switch_Right Rise
    Fault 204                                ; Cause Fault 204 - Positive Logic Right Limit Switch Activated
EndEvent

; Events looks for Homing Request Input coming on and Jumps to Homing Code
Event Homing_Call Input Homing_Request Rise
    Jump Homing                               ; Jump to Label 'Homing'
EndEvent

; Events looks for Run Input going off and Jumps to Program Start
Event Run_Input_Off Input Run_Input Fall
    Jump Program_Start                       ; Jump to Label 'Program Start'
EndEvent

; Events looks for a change in the Mode Selection Switch
Event Mode_Change Current_Mode <> Man_Auto_Switch
    Current_Mode = Man_Auto_Switch          ; Update Mode selection Variable with New Value
    Jump Mode_Select                       ; Jump to Label 'Mode_Select'
EndEvent

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

Program_Start:
    Outputs = False                        ; Start of Code
    ; Turn all Outputs Off
    If Safety_Enable == False              ; If Safety Enable INput off then turn on Out (3)
        Safety_IP_Indicator = True        ; Turn on Output (3)
    EndIf
    If Run_Input == False                   ; If system Run INput off then turn on Out (4)
        System_Run_Indicator = True      ; Turn on Output (4)
    EndIf
    Disable                                 ; Disable Drive / Make sure Drive is disabled
    Event Homing_call off                  ; Turn off event to check for homing request input
    Event Mode_Change off                  ; Turn 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

    While Safety_Enable == False           ; Wait for drive safety enable to be active - Safe Condition
        Safety_IP_Indicator = !Safety_IP_Indicator ; Toggle Indicate LED to show Safety Enable not Present
        Wait Time 1000                     ; Set time delay for indicator Flash
    EndWhile
    Safety_IP_Indicator = False            ; Turn off Output (3)

    While Run_Input == True                 ; Wait for drive command to run to not be active
        System_Run_Indicator = !System_Run_Indicator ; Toggle Indicate LED to show waiting on system Run Logic
        Wait Time 250                       ; Set time delay for indicator Flash
    EndWhile
    While Run_Input == False               ; Wait for drive command to run to be active (Toggle) - System Start Command
        System_Run_Indicator = !System_Run_Indicator ; Toggle Indicate LED to show waiting on system Run Logic
        Wait Time 1000                       ; Set time delay for indicator Flash
    EndWhile

    wait while Run_Input == False          ; Wait for system command to run
    System_Run_Indicator = False           ; Turn off Output (4)
    Event Run_Input_Off On                 ; Turn on event to detect system Run input switching off

```

```

Homing:
;
Event Homing_call off
Event Neg_Limit_Left_Active off
Event Neg_Limit_Right_Active off
Event Pos_Limit_Left_Active off
Event Pos_Limit_Right_Active off
Enable
Velocity Off
MaxV = 10
If H_Limit_Logic == True
    Move back until Limit_Switch_Left == True
Else
    Move back until Limit_Switch_Left == False
EndIf
maxv = 2
If H_Limit_Logic == True
    Move until Limit_Switch_Left == False
    Event Pos_Limit_Left_Active on
    Event Pos_Limit_Right_Active on
Else
    Move until Limit_Switch_Left == True
    Event Neg_Limit_Left_Active on
    Event Neg_Limit_Right_Active on
EndIf
Maxv = 50
MoveD 20
APOS = 0
Wait while Homing_Request == True
Event Homing_call on
Event Mode_Change on

; Homing Routine Performed at Start Up and after Homing Routine
; Request input goes high
; Turn off event to check for homing request input
; Turn off event to detect Left Limit Switch - Negative Logic
; Turn off event to detect Right Limit Switch - Negative Logic
; Turn off event to detect Left Limit Switch - Positive Logic
; Turn off event to detect Right Limit Switch - Positive Logic
; Enable Drive
; Turn Velocity Mode Off, Ensure Drive is not still in Velocity Mode
; set slow velocity for homing
; If Limit Switch Logic is set to Positive
; move back until homing sensor is activated
; If Limit Switch Logic is set to Negative
; move back at slow velocity until homing sensor is de-activated
; set very slow velocity
; If Limit Switch Logic is set to Positive
; move back at very slow velocity until homing sensor is activated
; Turn on event to detect Left Limit Switch - Positive Logic
; Turn on event to detect Right Limit Switch - Positive Logic
; If Limit Switch Logic is set to Negative
; move back at very slow velocity until homing sensor is de-activated
; Turn off event to detect Left Limit Switch - Negative Logic
; Turn off event to detect Right Limit Switch - Negative Logic
; Set velocity back to normal speed
; Move to start position in center of axis
; Set actual position register to 0
; wait for the homing request signal to be removed
; Turn on event to check for homing request input
; Turn on event to check for change of operating mode Input

Mode_Select:
If Man_Auto_Switch == False
    Wait Time 1000
    Vel = 0
    Velocity ON
    Goto Move_Profile_Manual
Else
    Wait Time 1000
    Velocity OFF
    Goto Move_Profile_Auto
Endif

; If Manual Mode Requested
; Wait 1 second to prevent fast switch toggle
; Initialise Vel (Velocity) variable to 0 - set Velocity to 0
; Turn on Velocity Mode
; Jump to the Manual Control Program Loop
; If Auto Mode Requested
; Wait 1 second to prevent fast switch toggle
; Turn off Velocity Mode
; Jump to the Automatic Control Program Loop

Move_Profile_Auto:
MoveP 0
Wait Time 200
MoveD 10
Wait Time 500
MoveD -15
Wait Time 1500
Goto Move_Profile_Auto

; Automatic position loop
; move to position 1
; wait 200 ms
; move to position 2
; wait 500 ms
; move to position 3
; wait 1500 ms
; restart automatic positioning loop

Move_Profile_Manual:
If Man_Left_Move == True && Man_Right_Move == False
    Vel = 20
Endif
If Man_Left_Move == False && Man_Right_Move == True
    Vel = -20
Endif
If Man_Left_Move == False && Man_Right_Move == False
    Vel = 0
Endif
If Man_Left_Move == True && Man_Right_Move == True
    Vel = 0
Endif
Goto Move_Profile_Manual

; Basic position loop
; If move Left button is pressed in Manual Mode
; Move Clockwise command, at set velocity
; If move Right button is pressed in Manual Mode
; Move Counter-Clockwise command, at set velocity
; If neither manual move buttons are pressed in Manual Mode
; Invalid Input - don't allow movement
; If both manual move buttons are pressed in Manual Mode
; Invalid Input - don't allow movement
; restart positioning loop

```

```

Limit_Switch_Recovery:
  While Run_Input == True
    System_Run_Indicator = !System_Run_Indicator
    Wait Time 250
  EndWhile
  System_Run_Indicator = True

  Wait until Limit_Switch_Reset == True
  Vel = 0
  Enable
  Velocity on

  If Limit_Switch_Left == False && H_Limit_Logic == False
    Vel = -2
    Wait Until Limit_Switch_Left == True
  EndIf

  If Limit_Switch_Right == False && H_Limit_Logic == False
    Vel = 2
    Wait Until Limit_Switch_Right == True
  EndIf

  If Limit_Switch_Left == True && H_Limit_Logic == True
    Vel = -2
    Wait Until Limit_Switch_Left == False
  EndIf

  If Limit_Switch_Right == True && H_Limit_Logic == True
    Vel = 2
    Wait Until Limit_Switch_Right == False
  EndIf

  Vel = 0
  Velocity Off
  Limit_Fault = False
  Goto Program_Start

Fault_Section:
  While Run_Input == True
    System_Run_Indicator = !System_Run_Indicator
    Wait Time 250
  EndWhile
  General_Fault = False

  Goto Program_Start

;***** Sub-Routines *****

;***** Fault Handler Routine *****
ON FAULT
  var_exstatus = var_exstatus | 0x00800000
  If Dfaults >= 201 && Dfaults <=204
    Limit_Fault = True
    Resume Limit_Switch_Recovery
  else
    If Dfaults == 36
      Safety_IP_Indicator = True
    EndIf
    General_Fault = True
    Resume Fault_Section
  endif
ENDFAULT

```