8 Position Break-Point Interrupt

Position Break-Point Activated Outputs

The position break-point interrupt is helpful in applications where an interrupt is to be generated based on the position of an axis passing a programmed set point while a move is in progress. The DSPL command which initiates such interrupt is EN_POSBRK. In addition to generation of interrupt, DSPL command POSBRK_OUT sets the programmed logic outputs.

The following DSPL program enables a position break point interrupt. This is done after clearing the corresponding interrupt register and programming the outputs to turn on (see POSBRK_OUT) at the break-point position. The position break-point interrupt is enabled to trigger at x=15000 and at y=15000. This is followed by a trapezoidal move command AXMOVE to move both axes to positions 28000. Clearly, in the process of achieving 28000, they must pass 15000 at which point interrupt is generated. The receipt of this interrupt is acknowledged by seven (7) output signals turned on. Next the position break-point interrupt is re-enabled to trigger at location x=3000 y=3000. The second AXMOVE command moves axes 1 and 2 to positions 0 and 0. The program waits until a position break-point interrupt is generated. This happens while move is in progress. The receipt of this interrupt is acknowledged by turning off all previously turned on signals.

```
plc_program:

  run_m_program (set_output_logic)

end
```

```
set_output_logic:

  int_reg_clr(0x0002, 0x3)        ;clear the pos_brk int register
  posbrk_out(0x1,0x1555,0x0000)   ;set output on mask
  en_posbrk(0x3, 15000, 15000)    ;enable position interrupt for
                                  ; axes 1,2 to set at x=15000, y=
                                  ; 15000

  axmove(0x3, .1, 28000, 5, .1, 28000, 5)
  wait_until(posbrk_reg & 0x0003) ;wait until position passed 15000

  int_reg_clr(0x0002, 0x3)        ;clear the pos_brk int register
  posbrk_out(0x1,0x0000,0x1555)   ;set outputs off
  en_posbrk (0x3, 3000, 3000)     ;enable position break-point
                                  ; to set at x=3000, y= 3000
  axmove(0x3, .1, 0, 5, .1, 0, 5)
  wait_until(posbrk_reg & 0x0003) ;wait until position passed 3000

end
```

Axis Exceeds Set Position Interrupt

Position break-point interrupt is helpful in applications where interrupt is generated based on the position of an axis passing a programmed set point during the move. The DSPL command that will initiate such an interrupt is `EN_POSBRK`

The program first enables position break-point interrupt. This is done after clearing the corresponding interrupt register. The positions break-point interrupt is enabled to trigger at x=15000 and y=15000. This is followed by a trapezoidal move command `AXMOVE` to move both axes to position 28000. Clearly, in the process of achieving 28000, position will pass 15000 at which point interrupt is generated. The receipt of this interrupt is acknowledged by presetting axis 4 to 444. Make sure axis 4 is not connected to an amplifier. Next the position break-point interrupt is re-enabled to trigger at location x=3000 y=3000. The second `AXMOVE` command moves axes 1 and 2 to positions 0 and 0. The program waits until a position break-point interrupt is generated. This happens while move command is in progress. The receipt of this interrupt is acknowledged by presetting axis 4 to 555.

```
plc_program:

   run_m_program (issue_position_int)

end


issue_position_int:
   int_reg_clr(0x0002, 0x3)      ;clear the pos_brk int register
   en_posbrk(0x3, 15000, 15000)  ;enable position interrupt for axes 1,2
                                 ;to set at x=15000, y= 15000

   axmove(0x3, .1, 28000, 5, .1, 28000, 5)
   wait_until(posbrk_reg & 0x0003)   ;wait until position passed 15000

   pos_preset(0x8, 444)          ;indicate the occurrence of the interrupt

   int_reg_clr(0x0002, 0x3)      ;clear the pos_brk int register
   en_posbrk (0x3, 3000, 3000)   ;enable position break-point
                                 ;to set at x=3000, y= 3000

   axmove(0x3, .1, 0, 5, .1, 0, 5)
   wait_until(posbrk_reg & 0x0003)   ;wait until position passed 3000
   pos_preset(0x8, 555)          ;indicate the occurrence of this interrupt

end
```

This page intentionally blank.