

기술 자료

# Position Compare

Position Compare

Auto Incremental, 예제 PLC

June, 05, 2009

# POSITION COMPARE

---

## Position Compare

---

PMAC 은 Servo IC 레벨의 Compare Register 에서 Hardware Count 를 직접 비교 하기 때문에 처리 속도나 정도면에서 뛰어난 Position Compare 기능을 제공합니다.

한 Servo IC 의 채널당 각각의 EQU 출력을 가지고 있으며 각각의 Position Compare Register 의 스케일은 Hardware Encoder Count 에 의해 결정됩니다.

Hardware Encoder Count 는 I7mn0(Servo IC m Channel n Encoder/Timer Decode Control)에 의해 정의되며 통상적으로 4 채널 사용하며 파라미터의 디폴트 또한 4 채널입니다.

Hardware Encoder Count 는 PMAC 의 전원투입시나 Reset(온라인 명령 : \$\$\$)시 영점이 되며 Home 명령시에는 변하지 않습니다

## Position Compare 설정

---

### - Position Compare 설정 관련 M-변수 (Suggested M-Variable)

M101 -> Y:\$78201,0,24,S	; Hardware Position Value (counts)
M108 -> Y:\$78207,0,24,S	; Position Compare A Value (counts)
M109 -> X:\$78207,0,24,S	; Position Compare B Value (counts)
M110 -> X:\$78206,0,24,S	; Auto-Incremental Value (counts)
M111 -> X:\$78205,11	; Position Compare Write Enable
M112 -> X:\$78205,12	; Position Compare Direct Write (Initial State) Value
M116 -> X:\$78200,9	; Position Compare Output State

### - Single Pulse Output

설정 가능한 특정 한 순간에만 출력을 요구하는 어플리케이션에 적용

- Position Compare A Value 설정 (Front edge)
- Position Compare B Value 설정 (Front edge)
- Auto Incremental Value 를 0 으로 설정
- Position Compare Direct Write (Initial State) Value 설정
- Position Compare Write Enable 설정

Hardware Encoder Count 의 위치를 기준으로 하여 원하고자 하는 위치에서 원하고자 하는 범위를 Position Compare A 와 B Value 값을 설정을 합니다.

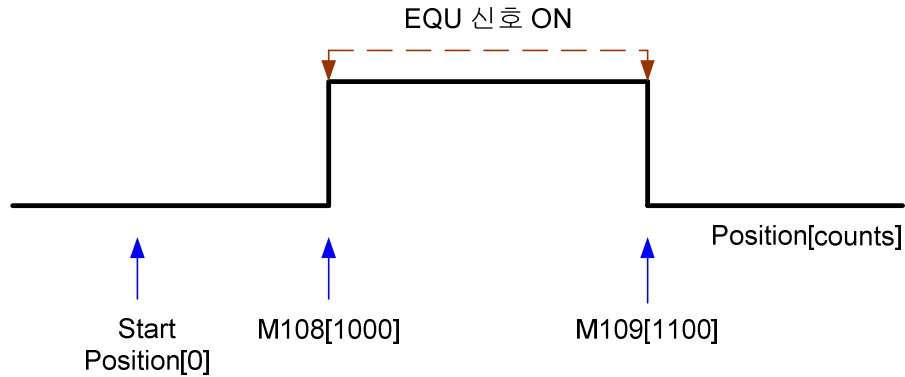
설정 예)

1000Cts 시점부터 1100Cts 까지 의 EQU 출력을 냄

M108 = 1000 ; EQU Start Position

M109 = 1100 ; EQU End Position

M110 = 0 ; Auto Incremental value  
M112 = 0 ; Initial value of 0  
M111 = 1 ; Enable direct write



Example PLC)

정해진 방향으로 진행을 하면서 특정 구간에서 출력을 내는 PLC 입니다.

Close  
Del gat  
End gat

```
#define EQU_INITIAL          P100
#define EQU_ENABLE          P101
#define CURR_OUTPUT         P200
#define PREV_OUTPUT         P201
#define USER_POS_A         P300
#define USER_POS_B         P301
#define EQU_POS_A           M108
#define EQU_POS_B           M109
#define WRITE_ENABLE        M111
#define DIRECT_WRITE_VALUE  M112
#define OUTPUT_STATUS       M116
#define HW_ENC_RESET        M113
#define REPORT_ERR          P999
```

M113->X:\$78205,10 ; Hardware encoder count reset register  
;(Amp Enable 이 되어있는 상태에서는 Interlock 이 걸렸기 때문에 Hardware encoder count reset 이 되지 않음)

```
Open plc 0 clear
REPORT_ERR = 0 ; Error Report 초기화
EQU_ENABLE = 0 ; EQU 시작 초기화
```

```
IF (M138 = 1) ; #1 Open loop 시
    REPORT_ERR = 1 ; Error Report 세팅
    DISABLE PLC0 ; PLC 0 Disable
    RETURN ; PLC 0 Return
ENDIF
```

```
PREV_OUTPUT = OUTPUT_STAUS ; 이전 EQU 출력 상태 초기화
DIRECT_WRITE_VALUE = 0 ; EQU 시작 레벨 초기화 설정
WRITE_ENABLE = 1 ; EQU 시작 레벨 초기화 세팅
EQU_ENABLE = 1 ; EQU 시작 세팅
EQU_POS_A = USER_POS_A ; User 가 원하는 Front Edge 설정
EQU_POS_B = USER_POS_B ; User 가 원하는 Back edge 설정
```

```
WHILE (EQU_ENABLE = 1)
    CURR_OUTPUT = OUTPUT_STATUS ; 현재 EQU 출력상태 저장
    IF (CURR_OUTPUT = 0 AND PREV_OUTPUT = 1) ; Front edge 에서 Back edge 로 바뀌는 순간이면
        EQU_ENABLE = 0 ; EQU 끝냄
```

```

EQU_POS_A = -8388608 ; EQU Front Edge 를 최대로 설정
EQU_POS_B = 8388607 ; EQU Back edge 를 최대로 설정
ENDIF
PREV_OUTPUT = CURR_OUTPUT ; 현재 EQU 출력 상태를 이전 EQU 출력 상태로 저장
ENDWHILE

DISABLE PLC0 ; PLC 0 Disable
CLOSE

```

### - Multiple Pulse Outputs (Auto Incremental)

원하고자 하는 시점에서부터 등간격으로 EQU 출력을 요구하는 어플리케이션에 적용

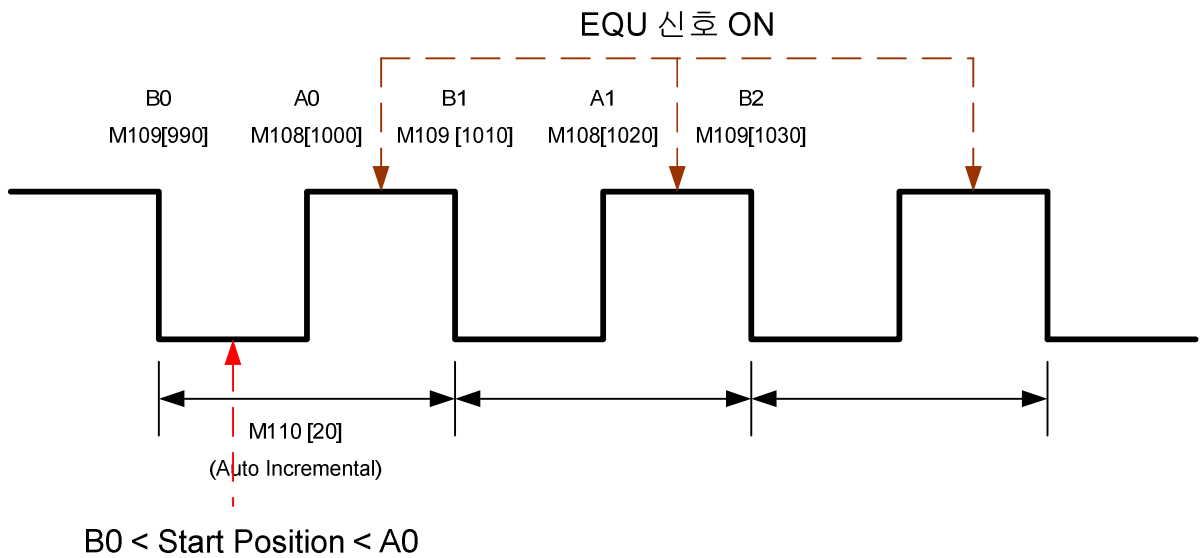
- First front edge 설정
- First back edge 설정
- Auto Incremental Value 설정
- Position Compare Direct Write (Initial State) Value 설정
- Position Compare Write Enable 설정

설정 예)

```

M108 = 1000 ; First front edge at 1000 counts
M109 = 1010 - M110 ; First back edge at 1010 counts -> 첫번째 Off 시점을 알려주기 위한 설정
M110 = 20 ; Auto incremental of 20 encoder counts
M112 = 0 ; Initial value of 0
M111 = 1 ; Enable direct write

```



Example PLC)

이 예제는 정해 놓은 위치부터 “Auto-Incremental”(자동증가) 기능을 수행하고, 정해진 카운트만큼 EQU 출력이 나가면 자동으로 Auto Increment 를 Disable 하도록 되어 있습니다. 이 과정에서 EQU 카운트를 세야 하기 때문에, EQU 를 사용할 때의 운전 속도에 제한이 있습니다.

현재 예제에서 최소 100Cnt 구간에서 EQU 출력 상태를 체크해야 하기 때문에 대략 50 ~ 70 Cnt 마다 EQU 출력 상태를 체크 해야 합니다.

이를 감안한다면, 사용자가 Default Servo Update Cycle 인 442usec 를 사용할 경우에 최대 속도를 100 ~ 140K Cts 정도를 사용할수 있는 것입니다.

따라서, 보다 빠른 속도에서 EQU 를 사용하기 위해서는 (예제에서 사용하는 방법) Servo Update Cycle 를 보다 짧게 설정해야 합니다.

(이와 관련한 부분은 “Turbo SRM”을 참고하여 주시기 바랍니다.)

```
CLOSE
END GAT
DEL GAT

#define SV_CNT      M101
#define COMP_A     M108
#define COMP_B     M109
#define COMP_AUTOINC M110
#define COMP_WE    M111
#define COMP_IS    M112
#define COMP_STAT  M116
#define MTR_POS    M162
#define MTR_BIAS  M164
#define MTR_SCALE  I108
#define COMP_ON_POS P108
#define COMP_OUT_WIDTH P109
#define COMP_INCVALUE P110
#define COMP_OUT_COUNT P111
#define OFFSET_CNT P164
#define TEMP_CNT   P101
#define TEMP_OUT_CNT P102
#define CURR_EQU  P116
#define PREV_EQU  P117

I8 = 0
I5 = 3

COMP_ON_POS = 30000 ; cnt
COMP_OUT_WIDTH = 100 ; cnt
COMP_INCVALUE = 1000 ; cnt
COMP_OUT_COUNT = 100 ; 회

OPEN PLC 0 CLEAR
; Actual 위치와 Servo IC 카운터간의 Offset 을 구합니다.
TEMP_CNT = (MTR_POS+MTR_BIAS)/(MTR_SCALE*32) ; cnt
OFFSET_CNT = SV_CNT - TEMP_CNT
; COMP_A, COMP_B 의 위치를 설정합니다.
COMP_A = COMP_ON_POS+OFFSET_CNT
COMP_B = COMP_A + COMP_OUT_WIDTH
COMP_AUTOINC = 0
; EQU 출력을 '0'으로 초기화 합니다.
COMP_IS = 0
COMP_WE = 1
; EQU 출력 상태가 '0' -> '1'로 바뀔 때까지 기다립니다.
; 여기서 실제로 모터가 움직이고 있을 것입니다.
TEMP_OUT_CNT = 0
CURR_EQU = COMP_STAT
PREV_EQU = CURR_EQU
While (CURR_EQU = PREV_EQU)
CURR_EQU = COMP_STAT
EndWhile
; 출력 상태가 바뀌면 Auto Increment 값을 설정합니다.
```

```

COMP_AUTOINC = COMP_INCVALUE
PREV_EQU = CURR_EQU
; 여기서 부터는 EQU 출력 개수를 카운트 합니다.
WHILE(TEMP_OUT_CNT<COMP_OUT_COUNT)
CURR_EQU = COMP_STAT
IF ( CURR_EQU!=PREV_EQU ); EQU 출력 상태가 '1'-'0' 으로 바뀔 때마다 카운트 합니다.
AND ( CURR_EQU=0 )
TEMP_OUT_CNT = TEMP_OUT_CNT+1
ENDIF
PREV_EQU = CURR_EQU
ENDWHILE

; Compare Disable
COMP_A = 8388607
COMP_B = -8388608
COMP_AUTOINC = -8388608

; PLC 를 종료합니다.
DISABLE PLC 0
CLOSE

```

## - Fractional Count Compare

Fractional count compare 기능은 2002 년 이후의 Revision D 타입 DSPGATE1 PMAC2 타입부터 가능하며, Firmware 버전 또한 V1.937 이후부터 가능합니다. PMAC1 타입은 이 기능을 사용 할 수 없습니다.

Servo IC 는 매 Encoder sample clock 주기 내에서 1/T timer-based 기능에 의해 Fractional count 를 생성됩니다. Sinusoidal encoder 를 채택 가능한 ACC-51E 를 사용 시에는 높은 채택에 의해서 Servo feedback resolution 은 크게 증가를 하나 실제 Compare 기능은 Hardware Count 값에 의한 것이기 때문에 Hardware count 의 resolution 은 증가하지 않습니다. 일반 Digital quadrature encoder 와 같다고 보시면 됩니다.

Fractional count compare 기능을 사용하기 위해서는 Servo IC 변수인 I7mn9(Servo IC m Channel n Hardware 1/T Control) 의 값을 1로 세팅을 해야 합니다.

I7mn9 변수가 1로 세팅한다는 것은 두 채널의 24Bit Timer register 로 바꾼다는 것입니다. 이것은 Encoder conversion table 의 일반적인 Software 1/T 채택을 쓰는 것이 아니라 Hardware 1/T 채택을 사용한다는 것인데 이는 4 개의 12Bit 값을 같은 두개의 24Bit Register 를 사용하게 됩니다. 만약 이 모드에서 디지털 4 체재 엔코더를 사용한다면 hardware 1/T 채택 확장 메소드를 사용하여야 합니다.

Servo IC 0 의 채널 1 의 Suggested M-Variable

M118 -> Y:\$78001,0,12,U ; Compare A Fractional Count

M119 -> Y:\$78000,0,12,U ; Compare B Fractional Count

위의 Register 의 값은 12Bit 이므로 0 부터 4096 을 최대로 갖습니다.

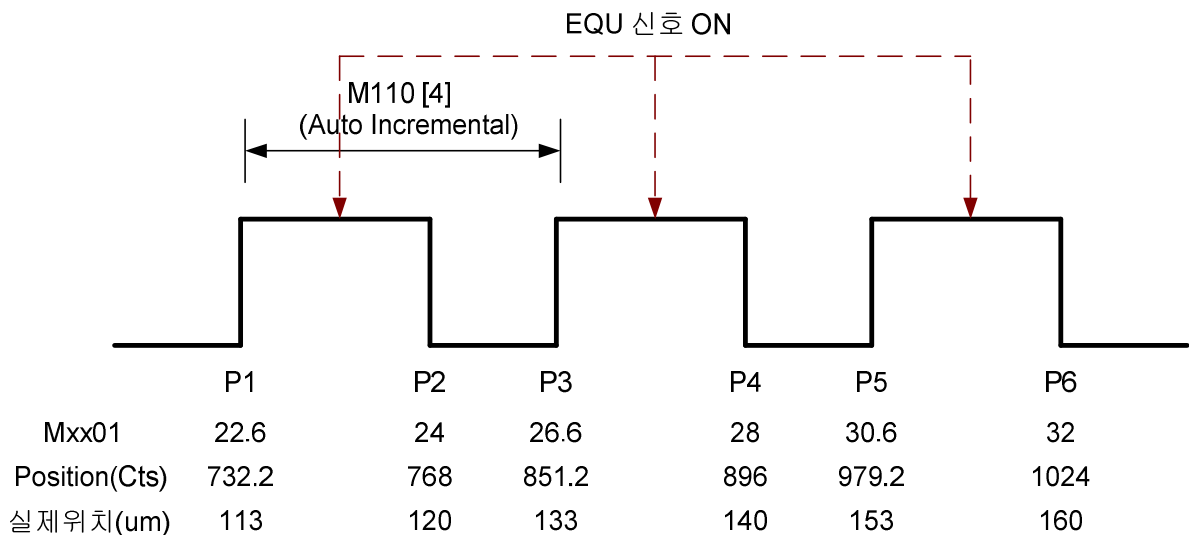
EQU 출력을 원하는 포지션이 소수점 일 경우(ACC-51E 를 이용한 20um 의 Analog Encoder 사용)

Compare A position : 100.5 (5um \* 100.5 = 502.5um)

Compare B position : 120.7 (5um \* 120.7 = 603.5um)

이러면 소수점 단위에 대한 처리가 필요 합니다. 이때 Fractional count compare 의 기능을 사용해야 하며 위의 Compare fractional count register 에 값을 적용해야 합니다.

설정 예)



위의 그림과 같은 경우 일 때

M108 = 22

M109 = 24

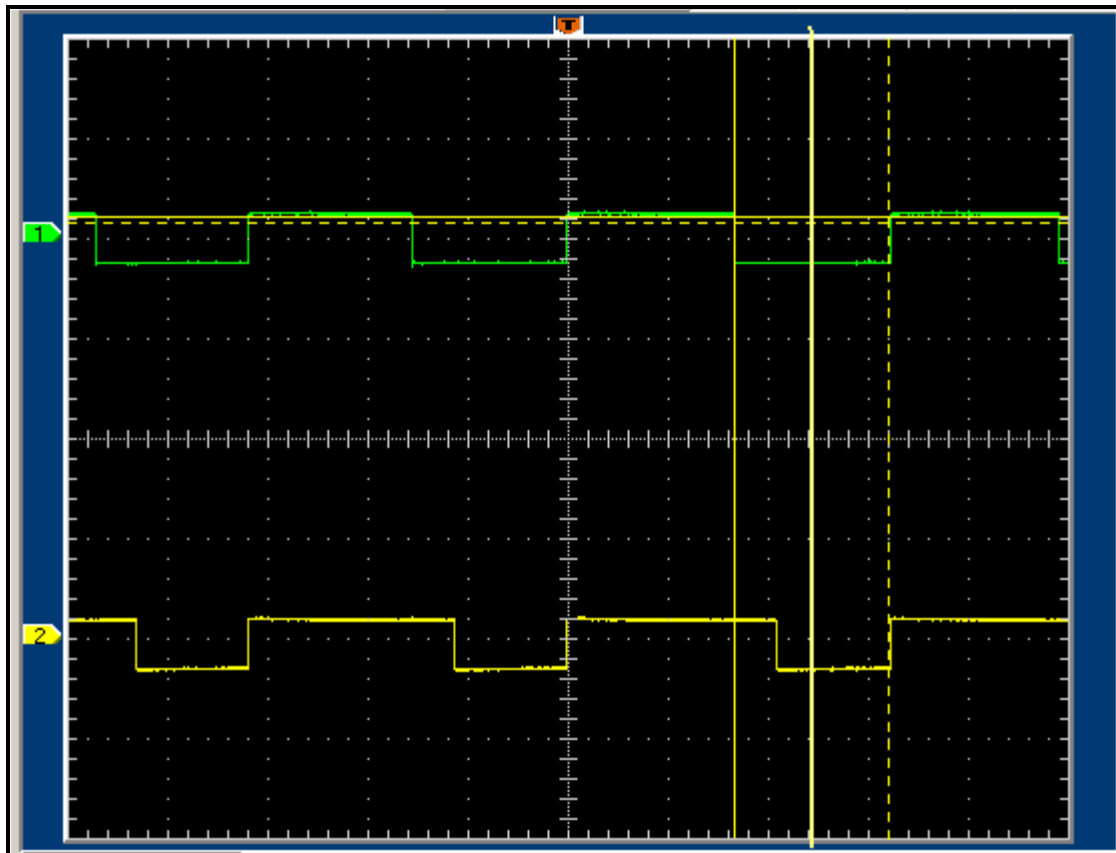
M110 = 4

M188 = 0.6 \* 4096

M189 = 0

※ 주의 사항

최소 Hardware 1 count 이상의 값을 설정해야 합니다. 1 count 내에서의 트리거는 한번 밖에 되지 않기 때문입니다. 예를 들어 105.3 과 105.5 에서는 Trigger 가 가능하나 다음 위치가 105.7 에서는 트리거가 되지 않습니다. 105.3 과 105.5 에서 Trigger 가 되면 다음 위치는 106 이상의 값이어야 합니다.



<CH1 : 정수배로 떨어지는 위치(No fraction), CH2 : 소수점 단위로 떨어지는 위치(Fraction)>

Example PLC)

위의 내용을 바탕으로 하여 PLC 입니다.

Close  
End gat  
Del gat

```
#define INIT_SET          P600
#define ENC6_RAWDATA     M501
#define CURR_POS         P601
#define CURR_POS_WITHFRAC P602

#define ENC6_CMPPOSA_Mxx08 M608
#define ENC6_CMPPOSB_Mxx09 M609
#define ENC6_PREV_CMPPOSA P8000
#define ENC6_PREV_CMPPOSB P8001

#define COMPARE_POSA     P603
#define COMPARE_POSB     P604
#define ENC6_FRAC_CMPPOSA_Mxx08 M688
#define ENC6_FRAC_CMPPOSB_Mxx09 M689
```



```

#define CURR_CMP_STATUS      P8004
#define PREV_CMP_STATUS     P8005
#define CMP_INDEX           P8006
#define 1um_PULSE_DIS      P8008
#define ENC6_CMP_OUT       M616

#define ENC6_CMP_WRITEENABLE M611
#define ENC_CMP_INIT       M612
#define HARDWARE_BASE_CNT  5
#define OUTPUT_DIS         6

#define NEXT_CMPPOSA       P8015
#define NEXT_CMPPOSB      P8016
#define PREV_CMPPOSA      P8017
#define PREV_CMPPOSB      P8018

```

Open plc0 clear

IF (INIT\_SET = 1)

```

COMPARE_POSA = 6
COMPARE_POSB = 7

```

```

NEXT_CMPPOSA = COMPARE_POSA
NEXT_CMPPOSB = COMPARE_POSB
PREV_CMPPOSA = NEXT_CMPPOSA
PREV_CMPPOSB = NEXT_CMPPOSB

```

```

1um_PULSE_DIS = (1/HARDWARE_BASE_CNT) * 4096
CURR_POS = ENC5_RAWDATA * HARDWARE_BASE_CNT
CURR_POS_WITHFRAC = ENC5_RAWDATA * 4096

```

```

ENC6_CMPPOSA_Mxx08 = INT((COMPARE_POSA + CURR_POS) / HARDWARE_BASE_CNT)
ENC6_CMPPOSB_Mxx09 = INT((COMPARE_POSB + CURR_POS) / HARDWARE_BASE_CNT)
ENC6_FRAC_CMPPOSA_Mxx88 = (COMPARE_POSA * 1um_PULSE_DIS) - CURR_POS_WITHFRAC
ENC6_FRAC_CMPPOSB_Mxx89 = (COMPARE_POSB * 1um_PULSE_DIS) - CURR_POS_WITHFRAC

```

```

ENC6_PREV_CMPPOSA = ENC6_CMPPOSA_Mxx08
ENC6_PREV_CMPPOSB = ENC6_CMPPOSB_Mxx09
ENC6_CMP_INIT = 0
ENC6_CMP_WRITEENABLE = 1
CURR_CMP_STATUS = 0
PREV_CMP_STATUS = 0

```

```

INIT_SET = 2
CMP_INDEX = 0
ENDIF

```

WHILE (INIT\_SET = 2)

```

CURR_CMP_STATUS = ENC6_CMP_OUT
CURR_POS_WITHFRAC = ENC6_RAWDATA * 4096

```

```

IF(CURR_CMP_STATUS = 1 AND PREV_CMP_STATUS = 0)
  CMP_INDEX = CMP_INDEX + 1

```

```

NEXT_CMPPOSA = PREV_CMPPOSA + OUTPUT_DIS
NEXT_CMPPOSB = PREV_CMPPOSB + OUTPUT_DIS

```

```

ENC6_CMPPOSA_Mxx08 = ENC6_PREV_CMPPOSA + INT(NEXT_CMPPOSA / HARDWARE_BASE_CNT)
ENC6_CMPPOSB_Mxx09 = ENC6_PREV_CMPPOSB + INT(NEXT_CMPPOSB / HARDWARE_BASE_CNT)
ENC6_FRAC_CMPPOSA_Mxx88 = (NEXT_CMPPOSA * 1um_PULSE_DIS) - CURR_POS_WITHFRAC
ENC6_FRAC_CMPPOSB_Mxx89 = (NEXT_CMPPOSB * 1um_PULSE_DIS) - CURR_POS_WITHFRAC
ENDIF

```

```

PREV_CMPPOSA = NEXT_CMPPOSA
PREV_CMPPOSB = NEXT_CMPPOSB
PREV_CMP_STATUS = CURR_CMP_STATUS

```

ENDWHILE

CLOSE

- Suggested M - Variable Definitions for whole count and fractional count data

IC #	Chan #	Mode I-Var	Compare A Wole Count (Mxx08)	Compare A Fractional Count (Mxx88)	Compare B Wole Count (Mxx09)	Compare B Fractional Count (Mxx89)
0	1	I7019	Y:\$078007,0,24,S	Y:\$078001,0,12,S	X:\$078007,0,24,S	Y:\$078000,0,12,S
0	2	I7029	Y:\$07800F,0,24,S	Y:\$078009,0,12,S	X:\$07800F,0,24,S	Y:\$078008,0,12,S
0	3	I7039	Y:\$078017,0,24,S	Y:\$078011,0,12,S	X:\$078017,0,24,S	Y:\$078010,0,12,S
0	4	I7049	Y:\$07801F,0,24,S	Y:\$078019,0,12,S	X:\$07801F,0,24,S	Y:\$078018,0,12,S
1	1	I7119	Y:\$078107,0,24,S	Y:\$078101,0,12,S	X:\$078107,0,24,S	Y:\$078100,0,12,S
1	2	I7129	Y:\$07810F,0,24,S	Y:\$078109,0,12,S	X:\$07810F,0,24,S	Y:\$078108,0,12,S
1	3	I7139	Y:\$078117,0,24,S	Y:\$078111,0,12,S	X:\$078117,0,24,S	Y:\$078110,0,12,S
1	4	I7149	Y:\$07811F,0,24,S	Y:\$078119,0,12,S	X:\$07811F,0,24,S	Y:\$078118,0,12,S
2	1	I7219	Y:\$078207,0,24,S	Y:\$078201,0,12,S	X:\$078207,0,24,S	Y:\$078200,0,12,S
2	2	I7229	Y:\$07820F,0,24,S	Y:\$078209,0,12,S	X:\$07820F,0,24,S	Y:\$078208,0,12,S
2	3	I7239	Y:\$078217,0,24,S	Y:\$078211,0,12,S	X:\$078217,0,24,S	Y:\$078210,0,12,S
2	4	I7249	Y:\$07821F,0,24,S	Y:\$078219,0,12,S	X:\$07821F,0,24,S	Y:\$078218,0,12,S
3	1	I7319	Y:\$078307,0,24,S	Y:\$078301,0,12,S	X:\$078307,0,24,S	Y:\$078300,0,12,S
3	2	I7329	Y:\$07830F,0,24,S	Y:\$078309,0,12,S	X:\$07830F,0,24,S	Y:\$078308,0,12,S
3	3	I7339	Y:\$078317,0,24,S	Y:\$078311,0,12,S	X:\$078317,0,24,S	Y:\$078310,0,12,S
3	4	I7349	Y:\$07831F,0,24,S	Y:\$078319,0,12,S	X:\$07831F,0,24,S	Y:\$078318,0,12,S
4	1	I7419	Y:\$079207,0,24,S	Y:\$079201,0,12,S	X:\$079207,0,24,S	Y:\$079200,0,12,S
4	2	I7429	Y:\$07920F,0,24,S	Y:\$079209,0,12,S	X:\$07920F,0,24,S	Y:\$079208,0,12,S
4	3	I7439	Y:\$079217,0,24,S	Y:\$079211,0,12,S	X:\$079217,0,24,S	Y:\$079210,0,12,S
4	4	I7449	Y:\$07921F,0,24,S	Y:\$079219,0,12,S	X:\$07921F,0,24,S	Y:\$079218,0,12,S
5	1	I7519	Y:\$079307,0,24,S	Y:\$079301,0,12,S	X:\$079307,0,24,S	Y:\$079300,0,12,S
5	2	I7529	Y:\$07930F,0,24,S	Y:\$079309,0,12,S	X:\$07930F,0,24,S	Y:\$079308,0,12,S
5	3	I7539	Y:\$079317,0,24,S	Y:\$079311,0,12,S	X:\$079317,0,24,S	Y:\$079310,0,12,S
5	4	I7549	Y:\$07931F,0,24,S	Y:\$079319,0,12,S	X:\$07931F,0,24,S	Y:\$079318,0,12,S
6	1	I7619	Y:\$07A207,0,24,S	Y:\$07A201,0,12,S	X:\$07A207,0,24,S	Y:\$07A200,0,12,S
6	2	I7629	Y:\$07A20F,0,24,S	Y:\$07A209,0,12,S	X:\$07A20F,0,24,S	Y:\$07A208,0,12,S
6	3	I7639	Y:\$07A207,0,24,S	Y:\$07A211,0,12,S	X:\$07A207,0,24,S	Y:\$07A210,0,12,S
6	4	I7649	Y:\$07A21F,0,24,S	Y:\$07A219,0,12,S	X:\$07A21F,0,24,S	Y:\$07A218,0,12,S
7	1	I7719	Y:\$07A307,0,24,S	Y:\$07A301,0,12,S	X:\$07A307,0,24,S	Y:\$07A300,0,12,S
7	2	I7729	Y:\$07A30F,0,24,S	Y:\$07A309,0,12,S	X:\$07A30F,0,24,S	Y:\$07A308,0,12,S
7	3	I7739	Y:\$07A317,0,24,S	Y:\$07A311,0,12,S	X:\$07A317,0,24,S	Y:\$07A310,0,12,S
7	4	I7749	Y:\$07A31F,0,24,S	Y:\$07A319,0,12,S	X:\$07A31F,0,24,S	Y:\$07A318,0,12,S
8	1	I7819	Y:\$07B207,0,24,S	Y:\$07B201,0,12,S	X:\$07B207,0,24,S	Y:\$07B200,0,12,S
8	2	I7829	Y:\$07B20F,0,24,S	Y:\$07B209,0,12,S	X:\$07B20F,0,24,S	Y:\$07B208,0,12,S
8	3	I7839	Y:\$07B217,0,24,S	Y:\$07B211,0,12,S	X:\$07B217,0,24,S	Y:\$07B210,0,12,S
8	4	I7849	Y:\$07B21F,0,24,S	Y:\$07B219,0,12,S	X:\$07B21F,0,24,S	Y:\$07B218,0,12,S
9	1	I7919	Y:\$07B307,0,24,S	Y:\$07B301,0,12,S	X:\$07B307,0,24,S	Y:\$07B300,0,12,S
9	2	I7929	Y:\$07B30F,0,24,S	Y:\$07B309,0,12,S	X:\$07B30F,0,24,S	Y:\$07B308,0,12,S
9	3	I7939	Y:\$07B317,0,24,S	Y:\$07B311,0,12,S	X:\$07B317,0,24,S	Y:\$07B310,0,12,S
9	4	I7949	Y:\$07B31F,0,24,S	Y:\$07B319,0,12,S	X:\$07B31F,0,24,S	Y:\$07B318,0,12,S

