

기 술 자 료

Encoder Loss Detection

Encoder Loss Detection

엔코더 문제 발생 감지

April 22, 2009

ENCODER LOSS DETECTION

Encoder Loss Detection 개요

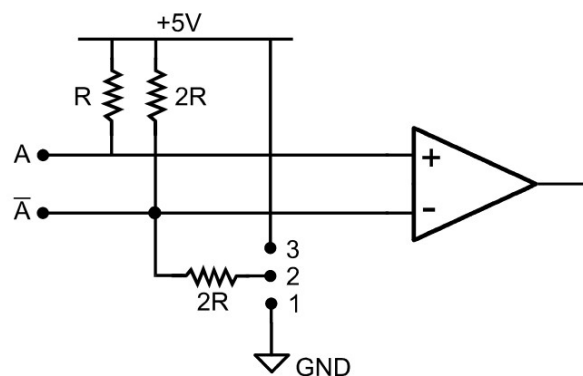
대부분의 Turbo PMAC 제품들은 각각의 Encoder 입력채널에 Encoder loss detection 회로를 가지고 있습니다. Encoder loss detection 기능을 사용할 경우 차동 라인드라이브 방식의 Encoder 사용시에만 가능하며 Single-ended 방식의 Encoder 사용시에는 Encoder loss detection 기능을 사용할 수 없습니다.

Encoder 입력의 각각의 A 상, /A 상의 입력을 XOR(exclusive-or) 하여 상태를 확인합니다.

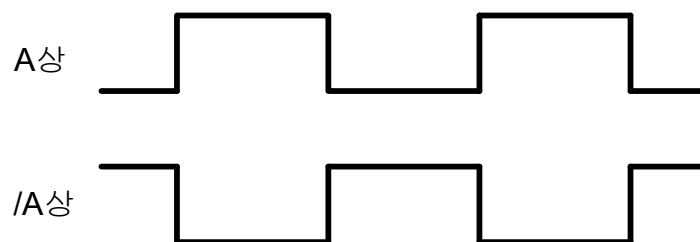
Encoder 의 연결 상태가 정상적이라고 한다면 입력 두 상의 상태가 항상 반대이며 XOR gate 의 출력이 항상 1 을 유지합니다.

Note:

제품 출하시 설정되어있지 않음, USER 가 사용함에 따라 변경하여야함.



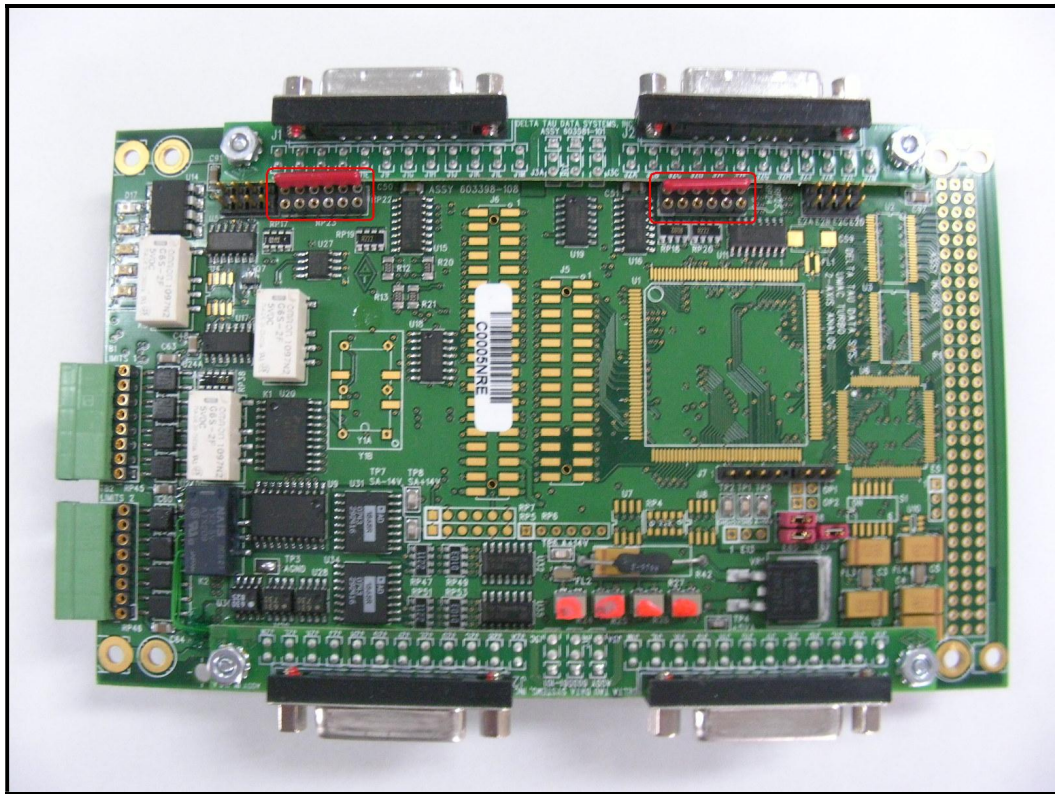
<PMAC Encoder 입력 회로>



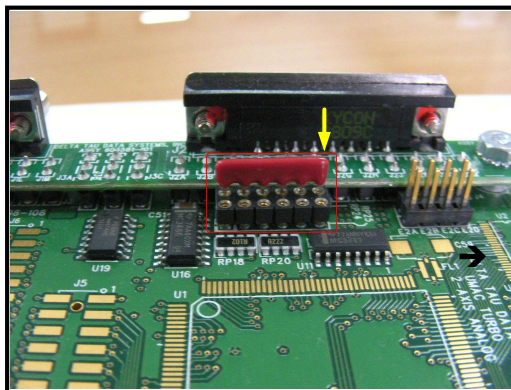
<차동 라인드라이브 입력 방식>

Encoder loss detection 설정

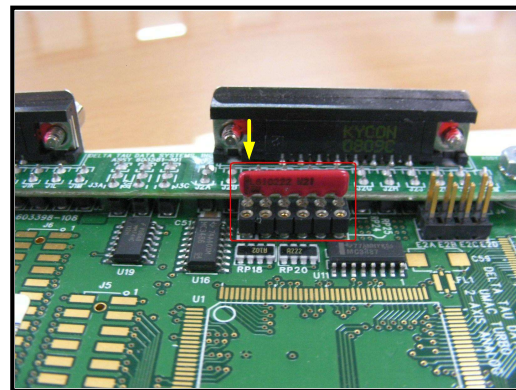
Encoder loss detection 기능을 사용하기 위하여 하드웨어적인 부분을 설정을 하셔야 합니다. 해당하는 채널의 어레이 저항을 반대로 뒤집어 삽입을 하여야 합니다.



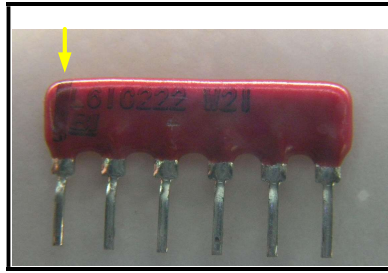
<ACC-24E2A의 어레이 저항 위치>



< Default 상태 >



<반대로 삽입>



<어레이 저항>

Note:

노란 화살표 방향이 1 핀임,
1 번핀을 확인 후 주의하여 방향에 맞게 삽입하여야함.

Device	Ch. 1	Ch. 2	Ch. 3	Ch. 4	Ch. 5/1	Ch. 6/2	Ch. 7/3	Ch. 8/4
PMAC-PCI	RP60	RP62	RP66	RP68	RP97	RP99	RP103	RP105
PMAC2-PCI	RP43	RP48	RP44	RP49	RP104	RP109	RP105	RP110
QMAC	RP55	RP57	RP66	RP68	-	-	-	-
Acc-24P2	RP74	RP75	RP85	RP86	RP139	RP140	RP150	RP151
Acc-24E2	RP22	RP24	RP22*	RP24*	RP22	RP24	RP22*	RP24*
Acc-24E2A	RP22	RP24	RP22*	RP24*	RP22	RP24	RP22*	RP24*
Acc-24E2S	RP19	RP21	RP27	RP29	RP19	RP21	RP27	RP29
Acc-24C2A	RP33	RP34	RP63	RP64	RP33	RP34	RP63	RP64

* On the daughter board for the accessory module

<해당하는 제품의 어레이 저항>

Device	Ch. 1	Ch. 2	Ch. 3	Ch. 4	Ch. 5/1	Ch. 6/2	Ch. 7/3	Ch. 8/4
PMAC-PCI	\$70801,1	\$70801,2	\$70801,3	\$70801,4	\$70801,5	\$70801,6	\$70801,6	\$70801,7
PMAC2-PCI	\$78403,8	\$78403,9	\$78403,10	\$78403,11	\$78403,8	\$78403,9	\$78403,10	\$78403,11
QMAC	\$78403,8	\$78403,9	\$78403,10	\$78403,11	-	-	-	-
Acc-24P2	\$7xF00,0	\$7xF00,1	\$7xF00,2	\$7xF00,3	\$7xF00,4	\$7xF00,5	\$7xF00,6	\$7xF00,7
Acc-24E2	\$7xF08,5	\$7xF09,5	\$7xF0A,5	\$7xF0B,5	\$7xF0C,5	\$7xF0D,5	\$7xF0E,5	\$7xF0F,5
Acc-24E2A	\$7xF08,5	\$7xF09,5	\$7xF0A,5	\$7xF0B,5	\$7xF0C,5	\$7xF0D,5	\$7xF0E,5	\$7xF0F,5
Acc-24E2S	\$7xF08,5	\$7xF09,5	\$7xF0A,5	\$7xF0B,5	\$7xF0C,5	\$7xF0D,5	\$7xF0E,5	\$7xF0F,5
Acc-24C2A	\$7xF08,5	\$7xF09,5	\$7xF0A,5	\$7xF0B,5	\$7xF0C,5	\$7xF0D,5	\$7xF0E,5	\$7xF0F,5

<해당하는 제품의 Error bit 어드레스>

• M 변수 정의 예

; UMAC Servo IC 2 (ACC-24E2A) 의 첫번째 채널, #1, #2, #3, #4

M180→Y:\$078F08,5,1 ; #1 첫번째 채널
M280→Y:\$078F09,5,1 ; #2 두번째 채널
M380→Y:\$078F0A,5,1 ; #3 세번째 채널
M480→Y:\$078F0B,5,1 ; #4 네번째 채널

; UMAC Servo IC 3 (ACC-24E2A) 의 첫번째 채널, #1, #2, #3, #4

M580→Y:\$078F0C,5,1 ; #1 첫번째 채널
M680→Y:\$078F0D,5,1 ; #2 두번째 채널

M780->Y:\$078F0E,5,1 ; #3 세번째 채널
M880->Y:\$078F0F,5,1 ; #4 네번째 채널

- 예제 PLC 프로그램

; Substitutions and Definitions

```
#define Mtr1OpenLoop M138 ; Motor status bit
Mtr1OpenLoop->Y:$0000B0,18,1 ; Standard definition
#define Enc1LossIn M180 ; Input loss-detection bit
Enc1LossIn->Y:$078F08,5,1 ; UMAC SIC2 Ch1 loss bit
#define Mtr1EncLossStatus P180 ; Internal latched status
#define Lost 0 ; Low-true fault here
#define OK 1 ; High is encoder present
#define Mtr1EncLossLatch P999 ; Latch
```

; Program to Check for and React to Encoder Loss

Mtr1EncLossLatch = 0 ; Latch initialize

OPEN PLCC 18 CLEAR

; Logic to Disable and Set Fault Status

```
IF (Mtr1OpenLoop=0 AND Enc1LossIn=Lost) ; Closed loop, no enc
    Mtr1EncLossStatus=1 ; Encoder loss detected
```

```
IF (Mtr1EncLossStatus = 1 and Mtr1EncLossLatch = 0)
```

```
    Mtr1EncLossLatch = 1
```

```
    CMD^K ; Kill all motors
```

```
Endif
```

```
Endif
```

```
IF (Mtr1OpenLoop=0 AND Enc1LossIn=OK) ; Normal operation status
```

```
    Mtr1EncLossLatch = 0 ; Latch clear
```

```
Endif
```

```
CLOSE
```

Analog Encoder Loss Detection (ACC-51E)

Analog Encoder 사용시 ACC-51E (Interpolator, X4096)를 사용하여 인터페이스를 합니다. 이때 ACC-24E2A, ACC-24E2 처럼 H/W 적인 Encoder Loss Detection 기능을 갖고 있지 않기 때문에 Analog Encoder 의 출력인 Sin, Cos 축력을 받아서 Encoder 들어오는 상태를 확인을 합니다.

Analog Encoder loss detection 설정

Turbo PMAC 은 Encoder 처리를 위해서 Encoder Conversion Table 을 사용합니다. Encoder Conversion Table 에서 Analog Encoder 입력 상태를 확인 가능하며 , 모터 구동 중 Analog Encoder 입력 Result 상태에 따른 조건으로 모션을 Kill 시킬 수 있습니다.

$$\text{Result} = (\text{SineADC} + \text{SineBias})^2 + (\text{CosineADC} + \text{CosineBias})^2$$

- High-Res Interpolator Diagnostic Entry First Lines For PMAC2-Style Servo ICs

Servo IC #	Channel 1	Channel 2	Channel 3	Channel 4
2	\$F78205	\$F7820D	\$F78215	\$F7821D
3	\$F78305	\$F7830D	\$F78315	\$F7831D
4	\$F79205	\$F7920D	\$F79215	\$F7921D
5	\$F79305	\$F7930D	\$F79315	\$F7931D
6	\$F7A205	\$F7A20D	\$F7A215	\$F7A21D
7	\$F7A305	\$F7A30D	\$F7A315	\$F7A31D
8	\$F7B205	\$F7B20D	\$F7B215	\$F7B21D
9	\$F7B305	\$F7B30D	\$F7B315	\$F7B31D

- Encoder Conversion Table 설정 (5 Entry)

High -Resolution Interpolation Diagnostic Entry (\$F/\$1)

Ex)

I8000 = \$78200 // Servo IC 2 ch1
I8001 = \$78208 // Servo IC 2 ch2
I8002 = \$78210 // Servo IC 2 ch3
I8003 = \$78218 // Servo IC 2 ch4
I8004 = \$78300 // Servo IC 3 ch1, ACC-51E
I8005 = \$78308 // Servo IC 3 ch2, ACC-51E
I8006 = \$78310 // Servo IC 3 ch3, ACC-51E
I8007 = \$78318 // Servo IC 3 ch4, ACC-51E
I8008 = \$F78305
I8009 = \$100000

```
I8010 = $0
I8011 = $0
I8012 = $0 // Result Address $351D
```

```
M4000->X:$351D,0,24 // Analog Encoder 입력 상태 Result M 변수
```

정상 sin, cos 신호 인경우 M4000 의 값이 \$100000 이상이며,
상위 4Bit 를 Mask 한 후 Mask 된 결과 값이 0 이면 Encoder Loss,
0 이 아니면 Encoder 정상으로 판단 하시면 됩니다.

- 예제 PLC 프로그램

```
; Substitutions and Definitions
```

```
#define Mtr1EncLossResult      M4000
```

```
#define Mtr1EncLossMaskResult  P4000
```

```
#define Mtr1EncLossLatch      P999
```

```
#define Mtr1EncLossStatus      P998
```

```
#define Mtr1OpenLoop  M138
```

```
Mtr1OpenLoop->Y:$0000B0,18,1
```

```
Mtr1EncLossLatch = 0
```

```
Open plc 18 clear
```

```
Mtr1EncLossMaskResult = Mtr1EncLossResult && $F00000
```

```
If (Mtr1OpenLoop = 0 and Mtr1EncLossMaskResult = 0)
```

```
    Mtr1EncLossStatus = 1
```

```
    If (Mtr1EncLossStatus = 1 and Mtr1EncLossLatch = 0)
```

```
        Mtr1EncLossLatch = 1
```

```
        CMD^K
```

```
    Endif
```

```
Endif
```

```
If (Mtr1OpenLoop and Mtr1EncLossMaskResult != 0)
```

```
    Mtr1EncLossLatch = 0
```

```
Endif
```

```
close
```