

보 조 자 료

UMAC Manual

UMAC Manual

UMAC 기본 자료

October 15, 2010

차 례

UMAC TURBO CPU / COMMUNICATIONS BOARD 설정	1
E3 Jumper : Re-Initialization on Reset Control.....	1
E6 Jumper : USB/Ethernet Micro-Controller Firmware Reload Enable.....	2
E23 Jumper : Firmware Reload Enable.....	2
ACC-24E2A BOARD 설정	3
Channel 의 입력과 출력 위치.....	3
E13 Jumper : phase and servo clocks(receive / transmit).....	4
E1A, E1B, E1C, E1D Jumper.....	5
Dip Switch : Servo IC Number 설정.....	5
Array 저항 : Encoder Loss Detection 설정.....	6
ACC-24E2S BOARD 설정	8
각 Channel 위치.....	8
E16 Jumper : phase and servo clocks(receive / transmit).....	8
Dip Switch : Servo IC Number 설정.....	9
ACC-11E BOARD 설정	10
Channel 의 입/출력 위치.....	10
E1, E2, E3, E4 Jumper : UMAC 에서의 기본 address 설정.....	10
ACC-51E BOARD 설정	11
Dip Switch : Servo IC Number 설정.....	11
20μm analog encoder 를 사용하였을 경우의 Count 계산.....	12
UMAC BOARD 설정 예	13
PEWIN32PRO 를 이용한 UMAC 과 HOST PC 와의 통신연결	16
Serial Cable 연결.....	17
USB Cable 연결(권장하지 않음).....	18
LAN(Ethernet) 연결(권장).....	21
PEWIN32PRO 의 시작	26
Pewin 32Pro 소개.....	26
Download Suggested M-variables.....	26
View 메뉴에서 Terminal, Watch Window, Position, Motor Status 실행.....	28
빠른 응답을 위한 Thread Period, Priority 설정.....	29
UMAC CONFIG PRO 를 이용한 SERVO IC 설정상태 확인	29
DATA BACKUP	30
SERVO IC 와 SERVO IC CHANNEL 표기 설명(M,N)	32
MOTOR 구동	33
토크모드(Torque Mode).....	33
속도모드(Velocity Mode).....	39
Stepping Motor Control.....	41
모터 구동 전 확인 및 주의 사항.....	44
Jog 명령.....	46
모터 구동.....	46
HOMING MOTORS(MOTOR 의 HOME 위치 이동)	47
Home 의 위치 결정 방법(I7mn3).....	48
Homing Search 방법.....	48
Homing 관련 I variables 와 명령어.....	48

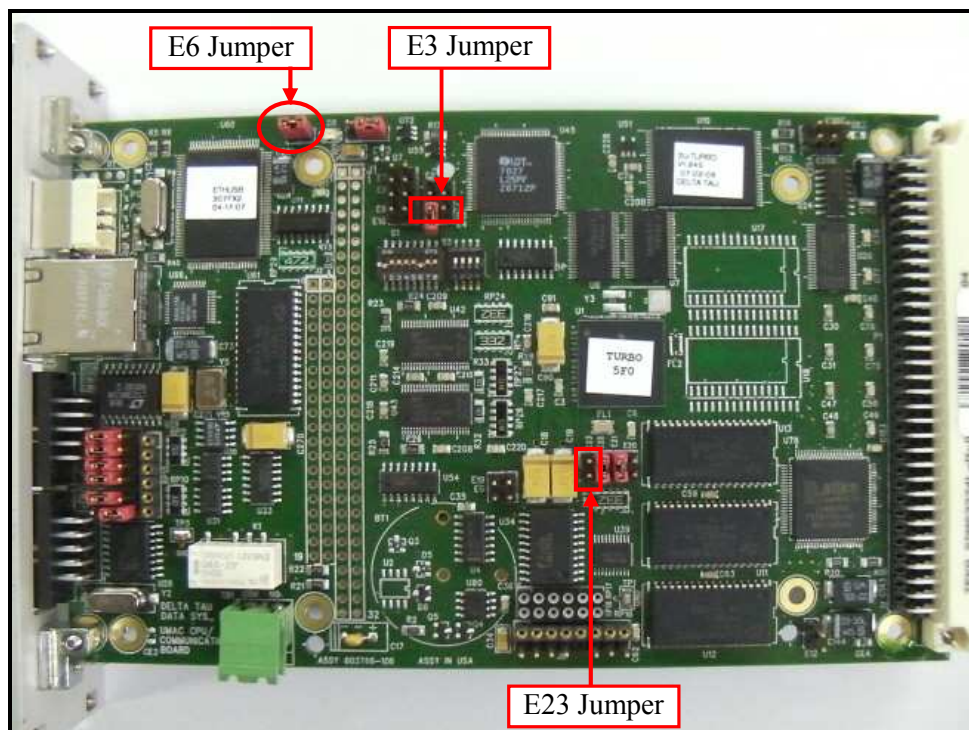
Homing 관련 Motor 2 의 M variables.....	50
Homing Motors 예제(Home Sensor, (+,-) Limit Sensor 를 사용하는 경우).....	50
DATA GATHERING(자료 수집).....	53
PLC PROGRAM 의 작성과 실행.....	56
MOTION PROGRAM 의 작성과 실행.....	58
비상 정지 절차.....	60

UMAC Turbo CPU / Communications Board 설정

UMAC 은 기존에 있던 PMAC Series 와 달리 CPU 기능과 Communications 기능을 단일 Board 에 구성하여 기존 ACC-54E(통신보드)가 불필요합니다. Base Board 는 80MHz 의 모토로라 DSP56303 CPU 를 사용하고 Option 에 따라 160MHz, 240MHz 의 CPU 를 사용할 수 있습니다. 100 Mbit/sec UDP/IP, TCP/IP Ethernet 통신, 480 Mbit/sec USB 2.0 통신을 지원하고 보조 Serial Port 2 개가 장착되어있습니다.

Note:

단, USB 의 경우 HOST 와의 전위차에 따른 USB 고장이 발생할 수 있으므로 사용을 권장하지 않습니다.



< UMAC Turbo CPU Board 의 점퍼 위치 >

E3 Jumper : Re-Initialization on Reset Control

UMAC 에 내부 프로그램 문제로 인하여 통신이 단절되었을 경우에 하드웨어적으로 RESET 을 할 때 사용하는 Jumper 로서, 온라인 명령어 \$\$\$*** 와 같은 기능을 합니다. 하드웨어적으로 RESET 이 필요한 경우에 Jumper 를 Close(쇼트)로 사용하면 됩니다. 연결 후에 UMAC 에 삽입하여 전원을 키고 PEWIN 32 Pro 를 실행하면 초기화 된 상태로 실행하게 됩니다.

E Point and Physical Layout	Description	Default
2 1	Remove jumper for normal reset mode (default). Jump pins 1 to 2 for re-initialization on reset.	No jumper installed

< E3 : Re-Initialization On Reset Control >

Note:

초기화 완료 및 저장 후 에는 CPU 카드를 다시 분리하여 E3 Jumper 를 제거하고 구동해야 합니다. E3 Jumper 가 연결된 상태로 있으면 UMAC 을 구동할 때 마다 다시 초기화가 됩니다. 만약, UMAC 이 부팅할 때 마다 초기화가 된다면 E3 Jumper 를 확인해야 합니다.

E6 Jumper : USB/Ethernet Micro-Controller Firmware Reload Enable

새로운 버전의 USB/Ethernet Micro-Controller Firmware 를 reload 시키기 위해서 사용하는 Jumper 입니다. 초기에는 Jumper 가 연결된 상태이며, 필요한 경우에 Jumper 를 제거하여 사용합니다. Jumper 제거 후 UMAC 에 삽입하여 전원을 켜 상태에서 Jumper 를 삽입 후 configure Ethernet for PMAC 프로그램을 이용하여 새로운 버전의 firmware 을 reload 시킵니다.

E Point and Physical Layout	Description	Default
2 1	Remove jumper to reload firmware on power-up\reset. Install jumper for normal operations.	Pins 1-2 jumpered

< E6 : USB/Ethernet Micro-Controller Firmware Reload Enable >

E23 Jumper : Firmware Reload Enable

새로운 버전의 CPU Firmware 로 reload 시키기 위해서 사용하는 Jumper 입니다. 초기에는 Jumper 가 연결이 안 되어있으며, 필요한 경우에 Jumper 를 연결하여 사용합니다. Jumper 연결 후 UMAC 에 삽입하여 전원을 투입 후 PEWIN32Pro 를 실행하면 초기에 Firmware Reload 를 위한 창이 뜨며 해당 firmware 파일을 Reload 하시면 됩니다.

E Point and Physical Layout	Description	Default
2 1	Install jumper to reload firmware through the communications port. Remove jumper for normal operations.	No jumper installed

< E23 : Firmware Reload Enable >

Note:

Reload 완료 후에는 CPU 카드를 다시 분리하여 E23 Jumper 를 제거하고 구동합니다.
 PEWIN 32Pro 를 실행했을 때 실행이 안 되고 Firmware Reload 에 관련된 창이 뜨면
 E23 Jumper 를 확인해야 합니다.

ACC-24E2A Board 설정

ACC-24E2A 는 $\pm 10V$ DAC 출력을 이용한 일반적인 토크, 속도모드제어, 2-Phase Command 방식 제어를 위한 Board 입니다.

한 개의 ACC-24E2A Board 는 4 Channel(4 축)을 제어 가능하며, Base Board 는 2 Channel 을 제어 가능하므로 Interface 용 Option Board (OPT-1A)를 추가하여 4 Channel 을 제어할 수 있습니다.

Servo IC 는 PMAC CPU 부와 외부 서보 신호의 중간에서 인터페이스 및 중요한 서보 기능을 수행하는 전용 IC 이며 Encoder Counter, Position Capture, Position Compare 기능 등 처리 속도에 민감한 기능들을 CPU 의 관여 없이 독립적으로 수행합니다.

- 기능

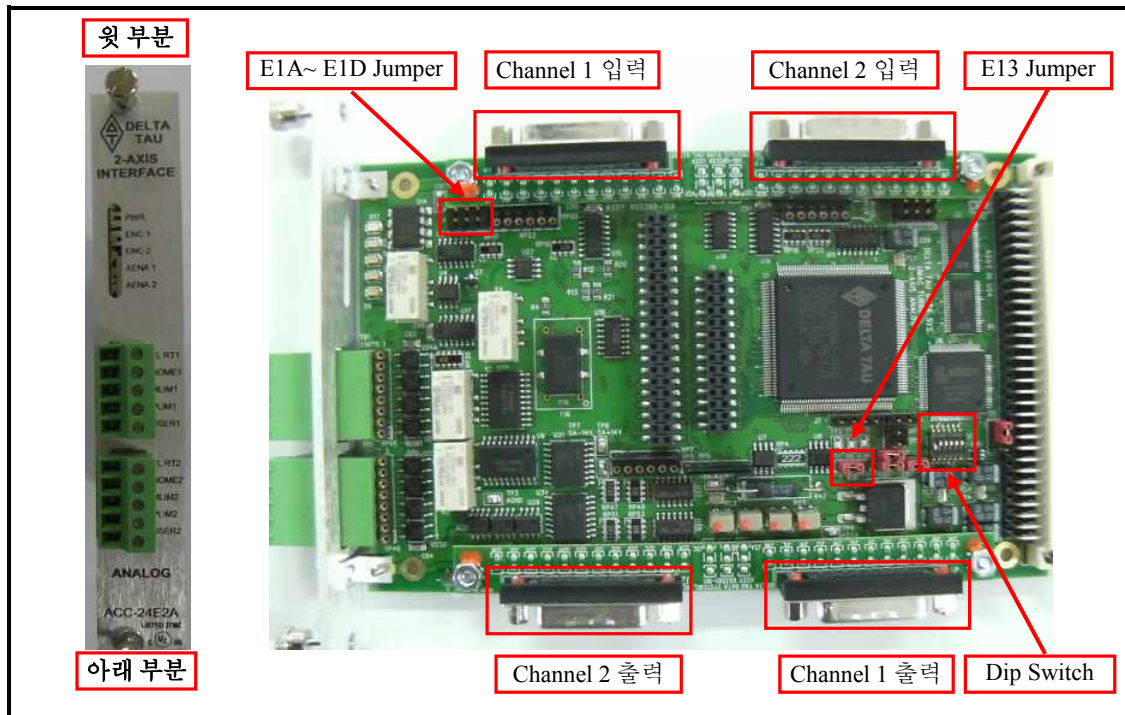
- Analog +/- 10V velocity commands
- Analog +/- 10V torque commands
- Sinusoidal analog +/- 10V phase current commands
- Pulse and Direction commands
- Differential/single-ended encoder input
- TTL Level U,V,W(Hall sensor) input
- +/- Limit, Home, User Flag input

Channel 의 입력과 출력 Connector 위치

Base board 1 개(2 channel)에 interface 용 option board 를 추가(4 channel)하여 사용할 수 있습니다. 전면부의 글씨가 정상 상태로 보일 때의 윗 부분이 입력(Digital Encoder)이며 아래 부분이 출력(Command output)입니다. 윗 부분에서 전면부쪽이 채널 1 의 입력이고 뒷 쪽이 채널 2 의 입력입니다. 아래 부분은 윗 부분과 반대로 전면부쪽이 채널 2 의 출력이고 뒷 쪽이 채널 1 의 출력입니다. 헷갈릴 수 있으므로 베젤에 실크 인쇄되어 있는 부분을 참고하십시오.

Note:

각 Channel 의 입력과 출력 포트를 연결할 때는 반드시 정확한 위치를 확인하고
연결해야 합니다. (각각의 Accessory 에 따라 위치가 다르기 때문)



<ACC-24E2A Board Jumper 및 Connector, DIP Switch 위치>

E13 Jumper : phase and servo clocks(receive / transmit)

여러 장의 카드를 쓸 경우에 모든 카드의 clock 을 동기화시켜야 하는데, 이 때 사용하게 되는 Jumper 입니다. 마스터 카드는 Transmit mode 로 설정하고 다른 카드들은 Receive mode 로 설정하여 Clock 을 동기시켜 사용해야 합니다. Transmit mode 로 설정하려면 E13 의 1,2,3 번 중에서 2,3 번을 연결하고, receive 로 설정하려면 1,2 번을 연결하면 됩니다. (점퍼를 이용하여 쇼트시킴)

Jumper	Config.	Description	Default
E13	1-2-3	Jump 1-2 to receive phase and servo clocks Jump 2-3 to transmit phase and servo clocks	Factory set

< E13 Jumper : phase and servo clocks(receive / transmit) >

Note:

카드의 교체 및 추가 시에 E13 Jumper 를 확인해서 clock 을 동기화 시켜야 합니다.
단, 당사 초기 UMAC 조립 후 출하시 CPU 옆에 있는 첫번째 카드(Servo IC 2)를
Master clock 카드로 설정하여 출하를 합니다.

E1A, E1B, E1C, E1D Jumper

Jumper 가 삽입되어 있지 않을 때는 Motor 의 hall sensor (TTL Level) 값을 받는 역할을 하며, Jumper 가 삽입되어 있는 경우에는 Pulse and Direction 신호를 출력하는 역할을 합니다.

Jumper	Config.	Description	Default
E1A	1-2	No Jumper for TTL Level input for CHU1 flag Jumper 1-2 for DIR1+ output in Stepper Mode	No jumper
E1B	1-2	No Jumper for TTL Level input for CHV1 flag Jumper 1-2 for DIR1- output in Stepper Mode	No jumper
E1C	1-2	No Jumper for TTL Level input for CHW1 flag Jumper 1-2 for PUL1+ output in Stepper Mode	No jumper
E1D	1-2	No Jumper for TTL Level input for CHT1 flag Jumper 1-2 for PUL1- output in Stepper Mode	No jumper

< E1A, E1B, E1C, E1D Jumper >

Dip Switch : Servo IC Number 설정

Servo IC 의 Base address 는 UMAC 에 삽입되어져 있는 순서와 관계없이 Dip Switch 의 설정으로 정해집니다. 아래의 표와 같이 Dip Switch 를 모두 on 키시면 Servo IC Number 가 2 번 으로 설정되고 1 번 Channel 의 주소가 \$78200 으로 설정됩니다.
다른 Number 설정을 위해서는 아래 표와 같이 설정을 하십시오.

Interp SW1 Settings						Turbo PMAC Servo IC # (m)	1 st Channe l	2 nd Channel	3 rd Channel	4 th Channel	I-Variables	CS16 Ident. Address
6	5	4	3	2	1							
on	on	on	on	on	on	2	\$78200	\$78208	\$78210	\$78218	I7200-I7249	\$78F08
on	on	on	on	on	off	3	\$78300	\$78308	\$78010	\$78318	I7300-I7349	\$78F0C
on	on	on	off	on	on	4	\$79200	\$79208	\$79210	\$79218	I7400-I7449	\$79F08
on	on	on	off	on	off	5	\$79300	\$79308	\$79310	\$79318	I7500-I7549	\$79F0C
on	on	off	on	on	on	6	\$7A200	\$7A208	\$7A210	\$7A218	I7600-I7649	\$7AF08
on	on	off	on	on	off	7	\$7A300	\$7A308	\$7A310	\$7A318	I7700-I7749	\$7AF0C
on	on	off	off	on	on	8	\$7B200	\$7B208	\$7B210	\$7B218	I7800-I7849	\$7BF08
on	on	off	off	on	off	9	\$7B300	\$7B308	\$7B310	\$7B318	I7900-I7949	\$7BF0C
on	on	on	on	off	on	2*	\$78220	\$78228	\$78230	\$78238	I7250-I7299	\$78F28
on	on	on	on	off	off	3*	\$78320	\$78328	\$78030	\$78338	I7350-I7399	\$78F2C
on	on	on	off	off	on	4*	\$79220	\$79228	\$79230	\$79238	I7450-I7499	\$79F28
on	on	on	off	off	off	5*	\$79320	\$79328	\$79330	\$79338	I7550-I7599	\$79F2C
on	on	off	on	off	on	6*	\$7A220	\$7A228	\$7A230	\$7A238	I7650-I7699	\$7AF28
on	on	off	on	off	off	7*	\$7A320	\$7A328	\$7A330	\$7A338	I7750-I7799	\$7AF2C
on	on	off	off	off	on	8*	\$7B220	\$7B228	\$7B230	\$7B238	I7850-I7899	\$7BF28
on	on	off	off	off	off	9*	\$7B320	\$7B328	\$7B330	\$7B338	I7950-I7999	\$7BF2C

< Dip Switch 설정에 따른 Servo IC Number 및 Base address >

Note:

카드의 교체 및 추가 시에 Dip Switch 를 확인해서 Servo IC Numbe 및
Base address 가 중복되지 않게 설정해야 합니다.
헛갈리지 않기 위하여 Servo IC 순서대로 삽입을 하십시오.
기본 출하시 순서대로 조립 되어져 있으며 CPU 에서 가장 가까운쪽부터
Servo IC 2 번으로 시작됩니다

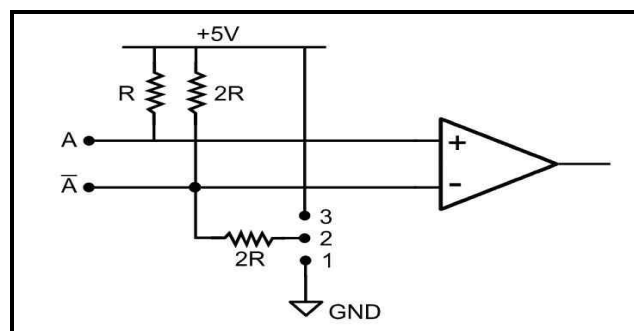
Array 저항 : Encoder Loss Detection 설정

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

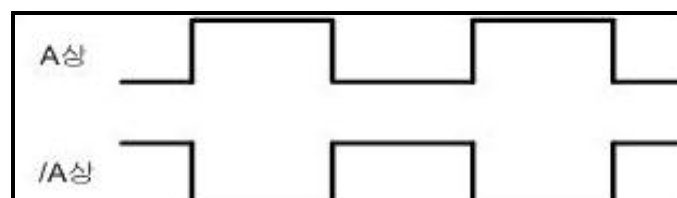
Encoder 입력의 각각의 A 상, /A 상의 입력을 XOR(exclusive-or) 하여 상태를 확인 합니다. Encoder 의 연결 상태가 정상적이라고 한다면 입력 두 상의 상태가 항상 반대이며 XOR gate 의 출력이 항상 1 을 유지합니다.

Note:

제품 출하시에 Encoder loss detection 기능을 사용하도록 설정되어있기 때문에 사용하지 않을 경우에는 USER 가 변경하여야 합니다.

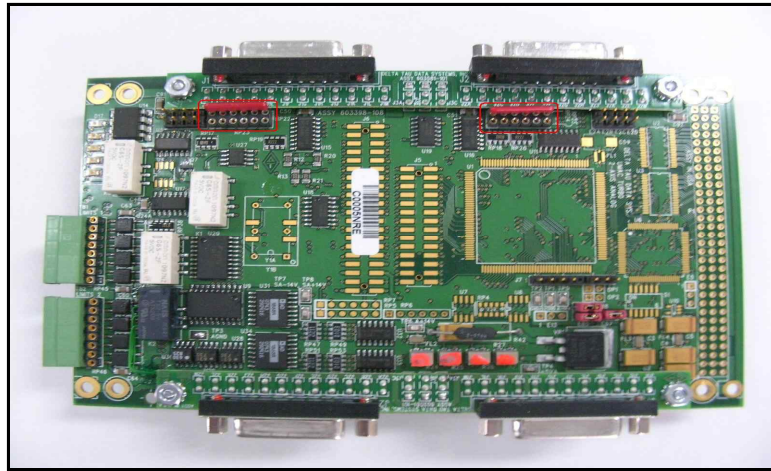


< PMAC Encoder 입력 회로 >

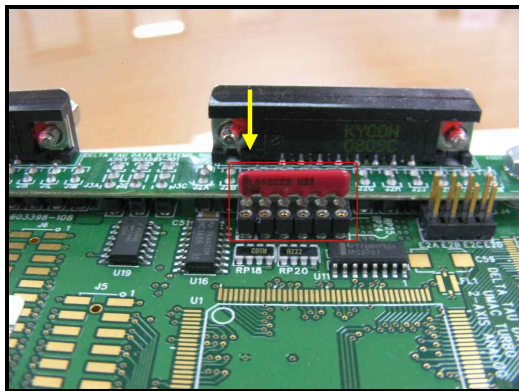


< 차동 라인드라이브 입력 방식 >

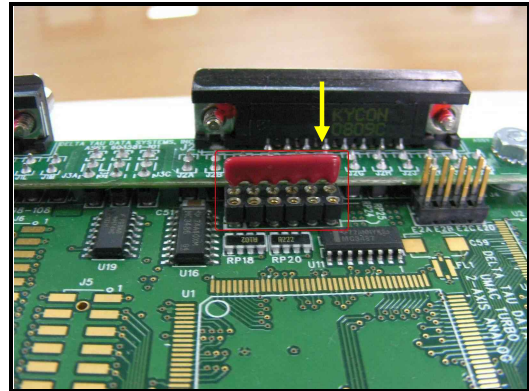
Encoder loss detection 기능을 해제하기 위해서는 하드웨어적인 부분을 설정을 하셔야 합니다. 해당하는 채널의 어레이 저항을 반대로 뒤집어 삽입을 하여야 합니다.



< ACC-24E2A 의 어레이 저항 위치 >



< Default 상태 >



<반대로 삽입>



< 어레이 저항 >

Note:

노란 화살표 방향이 1 핀입니다.

1 번핀을 확인 후 주의하여 방향에 맞게 삽입해야합니다.

ACC-24E2S Board 설정

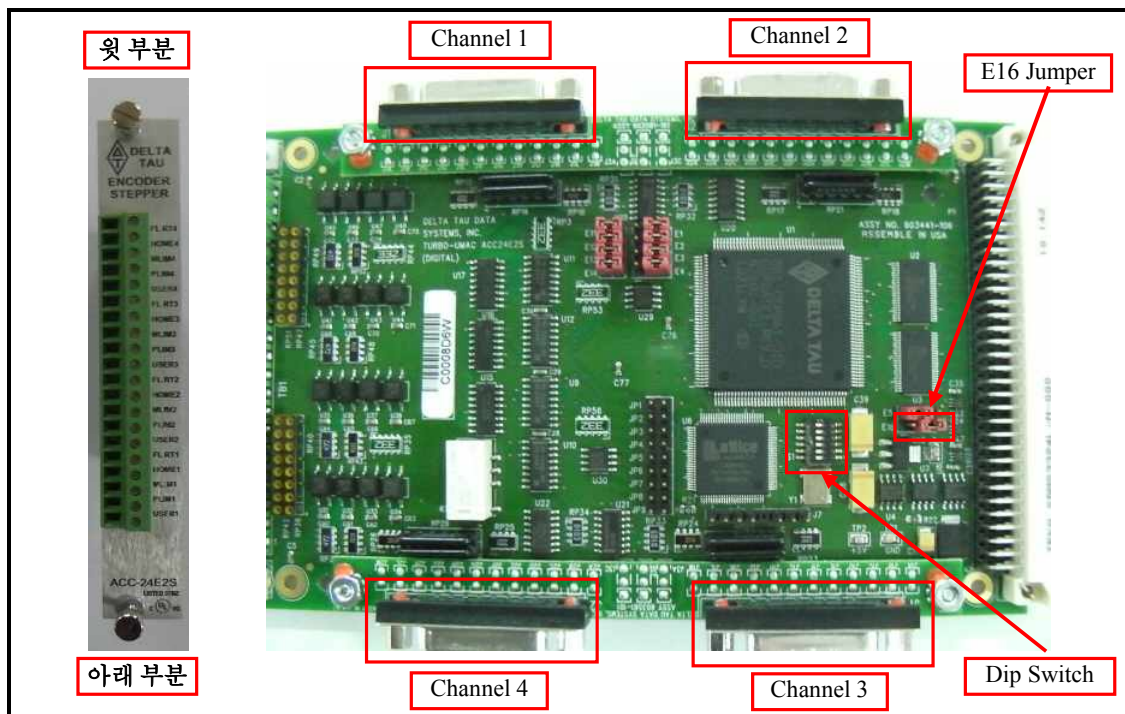
Stepping Motor 구동(일반적인 위치제어) 을 위해 Pulse & Direction 출력하는 Servo IC Board 입니다. Base Board 에 4 개의 channel 이 있습니다.

- 기능

- Pulse and Direction commands
- Differential/single-ended encoder input
- +/- Limit, Home, User Flag input

각 Channel 위치

전면부의 글씨가 정상 상태로 보일 때의 위쪽에서 전면부쪽이 1 번 채널, 그 뒤가 2 번 채널, 아래쪽에서 전면부쪽이 4 번 채널, 그 뒤가 3 번 채널입니다.



< ACC-24E2S Board 의 Jumper 및 connector 위치 >

E16 Jumper : phase and servo clocks(receive / transmit)

여러 장의 카드를 쓸 경우에 모든 카드의 clock 을 동기화시켜야 하는데, 이 때 사용하게 되는 Jumper 입니다. 마스터 카드는 Transmit mode 로 설정하고 다른 카드들은 Receive mode 로 설정하여 Clock 를 동기시켜 사용해야 합니다. Transmit mode 로

설정하려면 E16 의 1,2,3 번 중에서 2,3 번을 연결하고, receive 로 설정하려면 1,2 번을 연결하면 됩니다. (점퍼를 이용하여 쇼트시킴)

Jumper	Config.	Description	Default
E16	1-2-3	Jump 1-2 to receive phase and servo clocks Jump 2-3 to transmit phase and servo clocks	Factory set

< E16 Jumper : phase and servo clocks(receive / transmit) >

Note:

교체 및 추가 시에 E16 Jumper 를 확인해서 clock 을 동기화 시켜야 합니다.

단, 당사 초기 UMAC 조립 후 출하시 CPU 옆에 있는 첫번째 카드(Servo IC 2)가

Master clock 카드로 설정하여 출하를 합니다.

Dip Switch : Servo IC Number 설정

Servo IC 의 Base address 는 UMAC 에 삽입되어져 있는 순서와 관계없이 Dip Switch 의 설정으로 정해집니다. 아래의 표와 같이 Dip Switch 를 모두 on 키시면 Servo IC Number 가 2 번 으로 설정되고 1 번 Channel 의 주소가 \$78200 으로 설정됩니다. 다른 Number 설정을 위해서는 아래 표와 같이 설정을 하십시오.

Interp SW1 Settings						Turbo PMAC Servo IC # (m)	1 st Channel 1	2 nd Channel	3 rd Channel	4 th Channel	I-Variables	CS16 Ident. Address
6	5	4	3	2	1							
on	on	on	on	on	on	2	\$78200	\$78208	\$78210	\$78218	I7200-I7249	\$78F08
on	on	on	on	on	off	3	\$78300	\$78308	\$78010	\$78318	I7300-I7349	\$78F0C
on	on	on	off	on	on	4	\$79200	\$79208	\$79210	\$79218	I7400-I7449	\$79F08
on	on	on	off	on	off	5	\$79300	\$79308	\$79310	\$79318	I7500-I7549	\$79F0C
on	on	off	on	on	on	6	\$7A200	\$7A208	\$7A210	\$7A218	I7600-I7649	\$7AF08
on	on	off	on	on	off	7	\$7A300	\$7A308	\$7A310	\$7A318	I7700-I7749	\$7AF0C
on	on	off	off	on	on	8	\$7B200	\$7B208	\$7B210	\$7B218	I7800-I7849	\$7BF08
on	on	off	off	on	off	9	\$7B300	\$7B308	\$7B310	\$7B318	I7900-I7949	\$7BF0C
on	on	on	on	off	on	2*	\$78220	\$78228	\$78230	\$78238	I7250-I7299	\$78F28
on	on	on	on	off	off	3*	\$78320	\$78328	\$78030	\$78338	I7350-I7399	\$78F2C
on	on	on	off	off	on	4*	\$79220	\$79228	\$79230	\$79238	I7450-I7499	\$79F28
on	on	on	off	off	off	5*	\$79320	\$79328	\$79330	\$79338	I7550-I7599	\$79F2C
on	on	off	on	off	on	6*	\$7A220	\$7A228	\$7A230	\$7A238	I7650-I7699	\$7AF28
on	on	off	on	off	off	7*	\$7A320	\$7A328	\$7A330	\$7A338	I7750-I7799	\$7AF2C
on	on	off	off	off	on	8*	\$7B220	\$7B228	\$7B230	\$7B238	I7850-I7899	\$7BF28
on	on	off	off	off	off	9*	\$7B320	\$7B328	\$7B330	\$7B338	I7950-I7999	\$7BF2C

< Dip Switch 설정에 따른 Servo IC Number 및 Base address >

Note:

카드의 교체 및 추가 시에 Dip Switch 를 확인해서 Servo IC Number 를 중복되지 않게 설정해야 합니다.

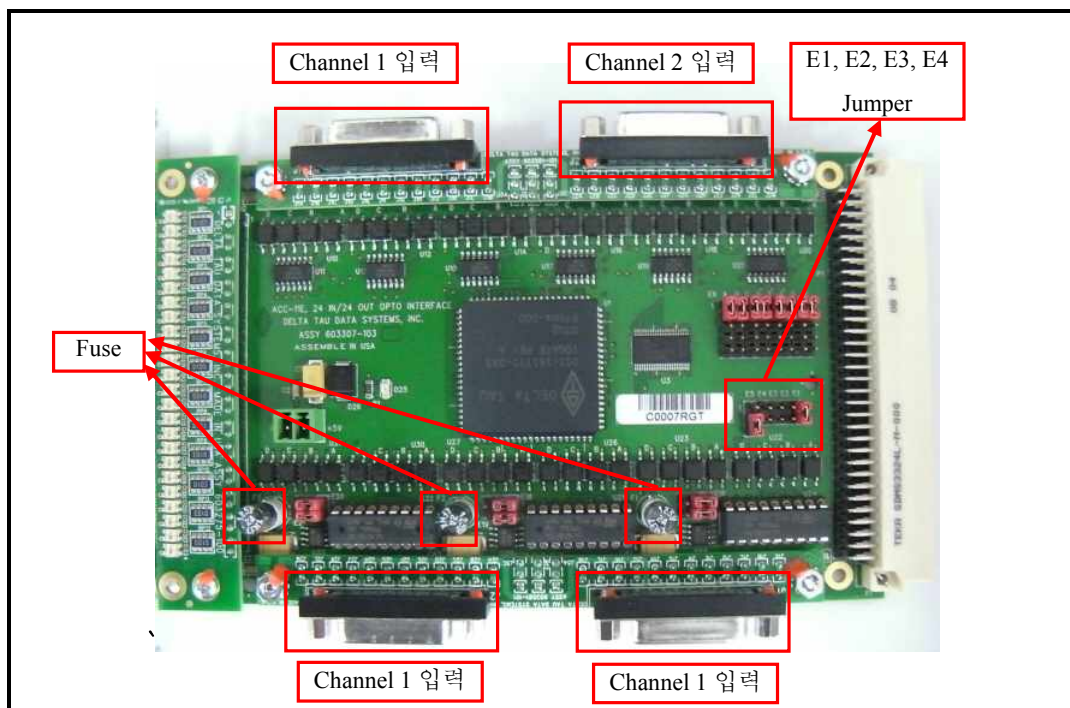
ACC-11E Board 설정

ACC-11E Board 는 general Purpose input/output board 입니다. 24 개의 입력이 있고 24 개의 출력이 있습니다.

Channel 의 입/출력 위치

전면부의 글씨가 정상 상태로 보일 때의 윗 부분이 I/O Input 이고 아래 부분이 I/O Output 입니다.

윗 부분에서 전면부쪽이 채널 1의 입력이고 뒷 쪽이 채널 2의 입력입니다. 아래 부분도 마찬가지로 전면부쪽이 채널 1의 출력이고 뒷 쪽이 채널 2의 출력입니다.



< ACC-11E Board Jumper, Connector, Fuse 위치 >

Note:

각 Channel 의 입력과 출력 포트를 연결할 때는 반드시 정확한 위치를 확인하고 연결해야 합니다.

E1, E2, E3, E4 Jumper : UMAC 에서의 기본 Address 설정

UMAC 에 11E 카드를 추가하여 사용하게 될 경우에 기존에 있던 11E 카드와 address 가 충돌할 수 있습니다. 이 때 E1, E2, E3, E4 Jumper 를 사용하여 각각의 11E 카드의 기본 address 를 설정하여 충돌을 막을 수 있습니다. 아래의 표와 같이 E1 Jumper 를 연결했을 때는 UMAC Turbo 에서 \$078C00 의 기본 주소를 갖게 됩니다. 추가되는

11E 카드들은 E2 부터 차례로 연결하여 기본 주소를 다르게 하면 됩니다.

Jumper	Chip Select	UMAC MACRO	UMAC Turbo
E1	10	\$FFE0 or \$8800*	\$078C00 (default)
E2	12	\$FFE8 or \$8840*	\$078D00
E3	14	\$FFF0 or \$8880*	\$078E00
E4	16	\$88C0*	\$078F00

* Could not be used with legacy MACRO CPUs (rev 100 – rev 104) but \$FFE0, \$FFE8 and \$FFF0 can be used with legacy MACRO CPUs.

< E1 – E4 : I/O Gate Transfer Jumpers >

Note:

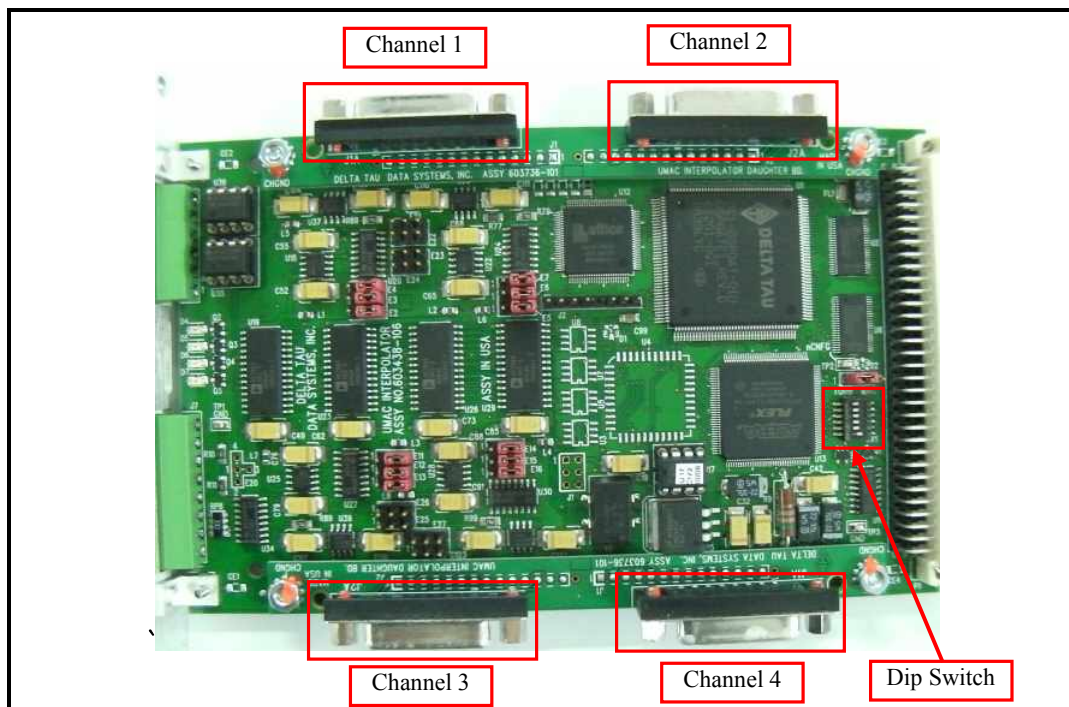
카드의 교체 및 추가 시에 기존에 있던 카드의 E1, E2, E3, E4 Jumper 를 확인해서 기본 address 가 중복되지 않게 설정해야 합니다.

ACC-51E Board 설정

Analog(Sine input) Encoder 를 사용할 수 있는 Interpolator board 입니다.

Analog Encoder 에서 sine wave 를 받아서 4096 체배를 하여 High resolution 의 더욱 정밀한 제어를 가능하게 합니다.

기본 사양으로는 2-Channel 입력이 가능하며 option-1 을 추가하여 4-Channel 입력이 가능합니다.



< ACC-51E Board Dip Switch, Connector 위치 >

Dip Switch : Servo IC Number 설정

Servo IC 의 Base address 는 UMAC 에 삽입되어져 있는 순서와 관계없이 Dip Switch 의 설정으로 정해집니다. 아래의 표와 같이 Dip Switch 를 모두 on 키시면 Servo IC Number 가 2 번 으로 설정되고 1 번 Channel 의 주소가 \$78200 으로 설정됩니다. 다른 Number 설정을 위해서는 아래 표와 같이 설정을 하십시오.

Interp SW1 Settings						Turbo PMAC Servo IC # (m)	1 st Channe l	2 nd Channel	3 rd Channel	4 th Channel	I-Variables	CS16 Ident. Address
6	5	4	3	2	1							
on	on	on	on	on	on	2	\$78200	\$78208	\$78210	\$78218	I7200-I7249	\$78F08
on	on	on	on	on	off	3	\$78300	\$78308	\$78010	\$78318	I7300-I7349	\$78F0C
on	on	on	off	on	on	4	\$79200	\$79208	\$79210	\$79218	I7400-I7449	\$79F08
on	on	on	off	on	off	5	\$79300	\$79308	\$79310	\$79318	I7500-I7549	\$79F0C
on	on	off	on	on	on	6	\$7A200	\$7A208	\$7A210	\$7A218	I7600-I7649	\$7AF08
on	on	off	on	on	off	7	\$7A300	\$7A308	\$7A310	\$7A318	I7700-I7749	\$7AF0C
on	on	off	off	on	on	8	\$7B200	\$7B208	\$7B210	\$7B218	I7800-I7849	\$7BF08
on	on	off	off	on	off	9	\$7B300	\$7B308	\$7B310	\$7B318	I7900-I7949	\$7BF0C
on	on	on	on	off	on	2*	\$78220	\$78228	\$78230	\$78238	I7250-I7299	\$78F28
on	on	on	on	off	off	3*	\$78320	\$78328	\$78030	\$78338	I7350-I7399	\$78F2C
on	on	on	off	off	on	4*	\$79220	\$79228	\$79230	\$79238	I7450-I7499	\$79F28
on	on	on	off	off	off	5*	\$79320	\$79328	\$79330	\$79338	I7550-I7599	\$79F2C
on	on	off	on	off	on	6*	\$7A220	\$7A228	\$7A230	\$7A238	I7650-I7699	\$7AF28
on	on	off	on	off	off	7*	\$7A320	\$7A328	\$7A330	\$7A338	I7750-I7799	\$7AF2C
on	on	off	off	off	on	8*	\$7B220	\$7B228	\$7B230	\$7B238	I7850-I7899	\$7BF28
on	on	off	off	off	off	9*	\$7B320	\$7B328	\$7B330	\$7B338	I7950-I7999	\$7BF2C

< Dip Switch 설정에 따른 Servo IC Number 및 Base address >

Note:

카드의 교체 및 추가 시에 Dip Switch 를 확인해서 Servo IC Number 를 중복되지 않게 설정해야 합니다.

20μm analog encoder 를 사용하였을 경우의 Count 계산

ACC-51E 는 Analog Quadrature 체배(4 체배)를 하고, 하드웨어 32 체배, 소프트웨어 32 체배를 하여 4096 체배가 되는 것입니다.

A,B 상의 4 체배 * 하드웨어 32 체배 * 소프트웨어(소수점 이하) 32 체배 = 4096 체배 입니다.

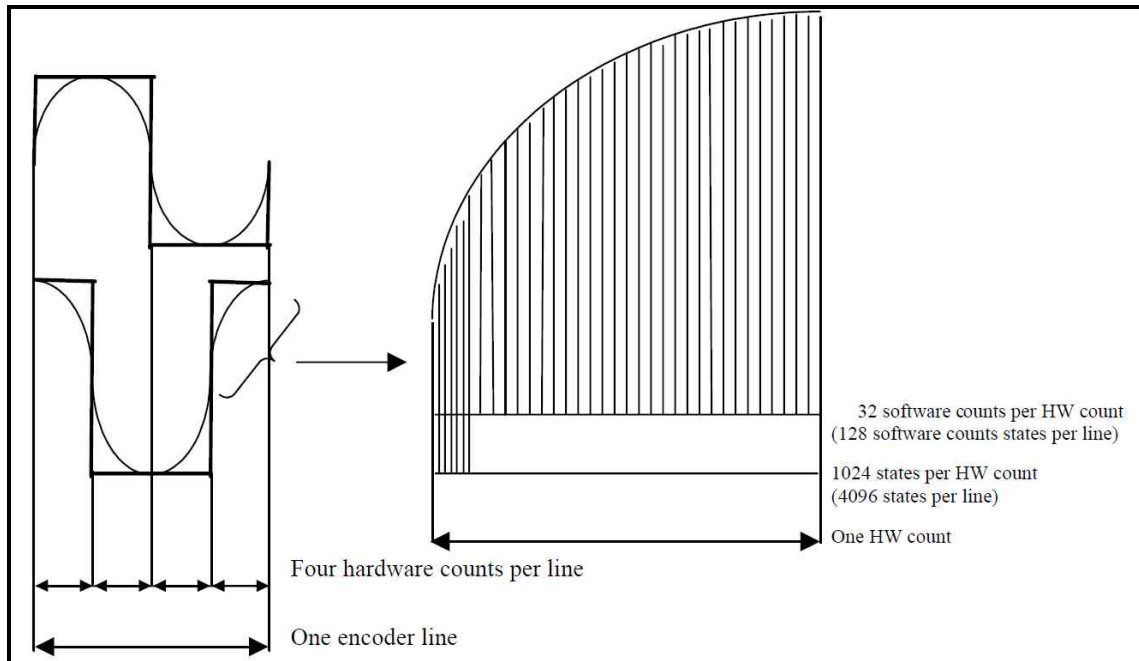
그러므로 쉽게 설명을 드리면 128(정수부) * 32(소수부) 가 되는 것입니다.

이때 20μm analog encoder 를 사용했을 경우

$20\mu\text{m} / 128 = 0.15625\mu\text{m}$ (정수부) = 1 cts

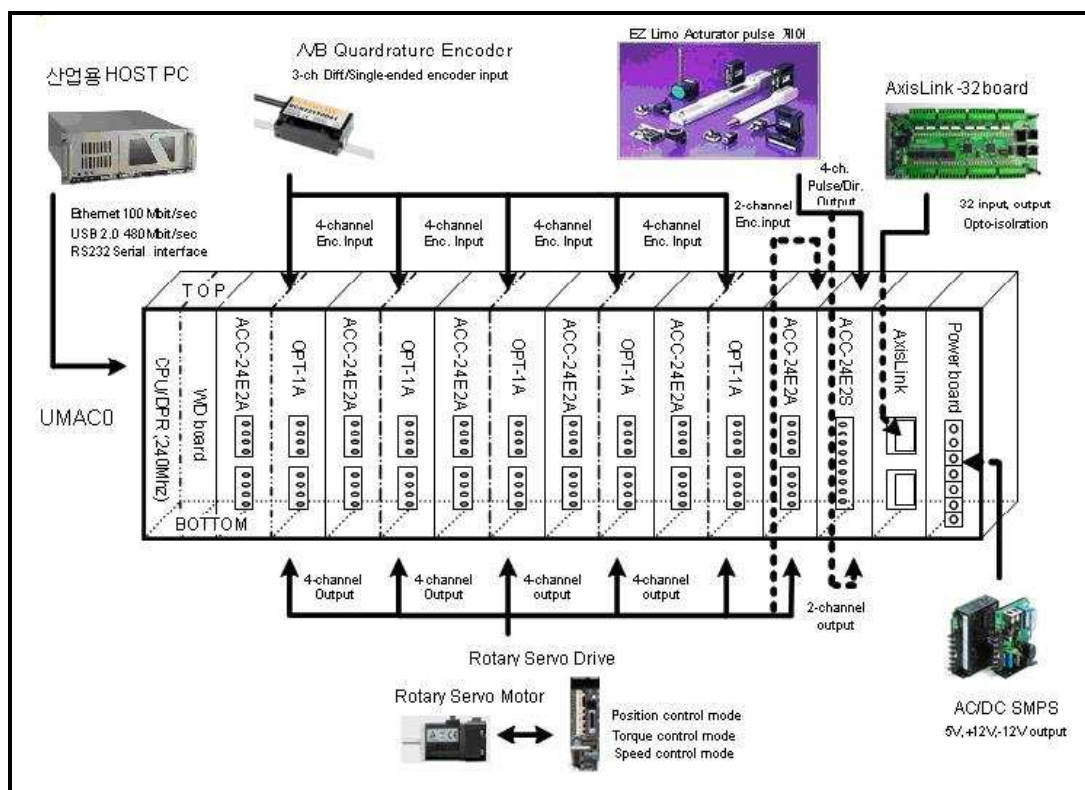
$0.15625 / 32 = 4.8828125\text{nm}$ (소수부) = 0.xx cts 입니다.

결과적으로 1cts = 0.15625 cts 이며, ACC-51E 를 사용시 이상적인 제어 가능한 최대 Resolution 은 대략 0.04cts = 0.0015625 μ m * 4 정도가 되는 것입니다.



< ACC-51E 4096 체배 원리 >

UMAC Board 설정 예



< UMAC 에 연결된 Board 들의 예 >

UMAC 에 연결된 Board 들의 종류와 수량

1. CPU/Communication Board 1 개
2. ACC-24E2A(OPT-1A) Board 5 개
3. ACC-24E2A Board 1 개
4. ACC-24E2S Board 1 개

Board 별 설정사항

1. CPU/Communication Board
 - E3 Jumper : No Jumper (Re-Initialization on Reset Control Disable)
 - E6 Jumper : Close (USB/Ethernet Micro-Controller Firmware Reload Disable)
 - E23 Jumper : No Jumper (Firmware Reload Disable)
2. 첫번째 ACC-24E2A(OPT-1A) Board - 4channel
 - E13 Jumper : 2,3 번 연결(Transmit - Master card)
 - E1A, E1B, E1C, E1D Jumper : No Jumper
(Motor 의 hall sensor (TTL Level) 값을 받습니다.)
 - Dip Switch : 6(ON) 5(ON) 4(ON) 3(ON) 2(ON) 1(ON)
Channel 1 Base Address : \$78200 -> motor 1
Channel 2 Base Address : \$78208 -> motor 2
Channel 3 Base Address : \$78210 -> motor 3
Channel 4 Base Address : \$78218 -> motor 4
 - Array 저항 : Encoder Loss Detection 기능 사용여부에 따라 결정합니다.
3. 두번째 ACC-24E2A(OPT-1A) Board - 4channel
 - E13 Jumper : 1,2 번 연결(Receive - Slave card)
 - E1A, E1B, E1C, E1D Jumper : No Jumper
(Motor 의 hall sensor (TTL Level) 값을 받습니다.)
 - Dip Switch : 6(ON) 5(ON) 4(ON) 3(ON) 2(ON) 1(OFF)
Channel 1 Base Address : \$78300 -> motor 5
Channel 2 Base Address : \$78308 -> motor 6
Channel 3 Base Address : \$78310 -> motor 7
Channel 4 Base Address : \$78318 -> motor 8
 - Array 저항 : Encoder Loss Detection 기능 사용여부에 따라 결정합니다.

4. 세번째 ACC-24E2A(OPT-1A) Board - 4channel

- E13 Jumper : 1,2 번 연결(Receive - Slave card)
- E1A, E1B, E1C, E1D Jumper : No Jumper
(Motor 의 hall sensor (TTL Level) 값을 받습니다.)
- Dip Switch : 6(ON) 5(ON) 4(ON) 3(OFF) 2(ON) 1(ON)
Channel 1 Base Address : \$79200 -> motor 9
Channel 2 Base Address : \$79208 -> motor 10
Channel 3 Base Address : \$79210 -> motor 11
Channel 4 Base Address : \$79218 -> motor 12
- Array 저항 : Encoder Loss Detection 기능 사용여부에 따라 결정합니다.

5. 네번째 ACC-24E2A(OPT-1A) Board - 4channel

- E13 Jumper : 1,2 번 연결(Receive - Slave card)
- E1A, E1B, E1C, E1D Jumper : No Jumper
(Motor 의 hall sensor (TTL Level) 값을 받습니다.)
- Dip Switch : 6(ON) 5(ON) 4(ON) 3(OFF) 2(ON) 1(OFF)
Channel 1 Base Address : \$79300 -> motor 13
Channel 2 Base Address : \$79308 -> motor 14
Channel 3 Base Address : \$79310 -> motor 15
Channel 4 Base Address : \$79318 -> motor 16
- Array 저항 : Encoder Loss Detection 기능 사용여부에 따라 결정합니다.

6. 다섯번째 ACC-24E2A(OPT-1A) Board - 4channel

- E13 Jumper : 1,2 번 연결(Receive - Slave card)
- E1A, E1B, E1C, E1D Jumper : No Jumper
(Motor 의 hall sensor (TTL Level) 값을 받습니다.)
- Dip Switch : 6(ON) 5(ON) 4(OFF) 3(ON) 2(ON) 1(ON)
Channel 1 Base Address : \$7A200 -> motor 17
Channel 2 Base Address : \$7A208 -> motor 18
Channel 3 Base Address : \$7A210 -> motor 19
Channel 4 Base Address : \$7A218 -> motor 20
- Array 저항 : Encoder Loss Detection 기능 사용여부에 따라 결정합니다.

7. ACC-24E2A (No OPT-1A) Board – 2channel

- E13 Jumper : 1,2 번 연결(Receive – Slave card)
- E1A, E1B, E1C, E1D Jumper : No Jumper
(Motor 의 hall sensor (TTL Level) 값을 받습니다.)
- Dip Switch : 6(ON) 5(ON) 4(OFF) 3(ON) 2(ON) 1(OFF)
Channel 1 Base Address : \$7A300 -> motor 21
Channel 2 Base Address : \$7A308 -> motor 22
- Array 저항 : Encoder Loss Detection 기능 사용여부에 따라 결정합니다.

8. ACC-24E2S Board

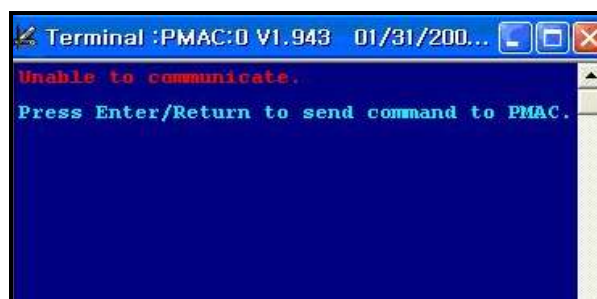
- E16 Jumper : 1,2 번 연결(Receive – Slave card)
- Dip Switch : 6(ON) 5(ON) 4(OFF) 3(OFF) 2(ON) 1(ON)
Channel 1 Base Address : \$7B200 -> motor 25
Channel 2 Base Address : \$7B208 -> motor 26
Channel 3 Base Address : \$7B210 -> motor 27
Channel 4 Base Address : \$7B218 -> motor 28

Pewin32Pro 를 이용한 UMAC 과 HOST PC 와의 통신연결

Pewin32Pro 프로그램은 PMAC 의 셋업(Setup), 디버그(Debug), 및 진단(Diagnose) 을 위한 필수적인 개발 프로그램입니다.

프로그램을 모두 설치한 후 PEWIN32 Pro 를 실행한다고 하여 바로 UMAC 과 통신 연결되지는 않습니다.

UMAC 과 PC 가 서로 통신하도록 Ethernet/USB 케이블을 연결하고 설정을 해야 하며 아래와 같이 3 가지 방법을 이용하여 UMAC 과 HOST PC 와의 통신 연결이 가능합니다. 아래의 방법으로 통신 연결이 완료되면 PEWIN32Pro 의 View 메뉴에서 Terminal 을 클릭하여 Terminal 창을 활성화시켜서 연결여부를 확인할 수 있습니다. 통신 연결이 완료되면 아래와 같은 메시지가 Terminal 창에 나옵니다.



< 통신 연결 여부가 확인된 상태의 Terminal 창 >

Unable to communicate. → 통신 연결이 안 되었을 때 나오는 메세지입니다.

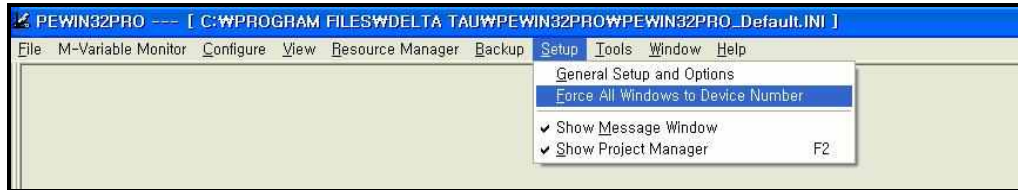
Press Enter/Return to send command to PMAC. → 통신 연결이 되었을 때 나오는 메세지입니다.

Serial Cable 연결

1. UMAC과 PC를 Serial Cable로 연결합니다.

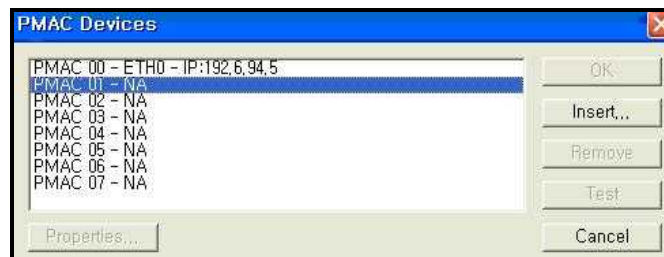
(홈페이지 자주묻는 질문란에 Serial Cable 제작 도면 참조)

2. PEWIN32Pro의 Setup 메뉴에서 Force all windows to device number를 클릭합니다.



< Force All Windows to Device Number 위치 >

3. 새로 열린 PMAC Device 창에서 NA로 되어있는 부분을 클릭하고 Insert를 클릭합니다.



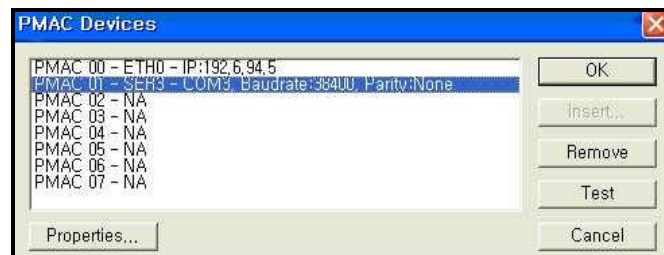
< PMAC Device 창에서 Insert 위치 >

4. 새로 열린 Available PMAC Device창에서 COM Port를 선택하고 OK를 클릭합니다.



< Available PMAC Device 창에서 COM Port 선택 >

5. PMAC Device창에서 생성된 Serial 연결(SER3-COM3, Baudrate:38400, Parity:None)을 선택하고 Test를 실행한 뒤 성공하면 OK를 클릭하고 종료합니다.



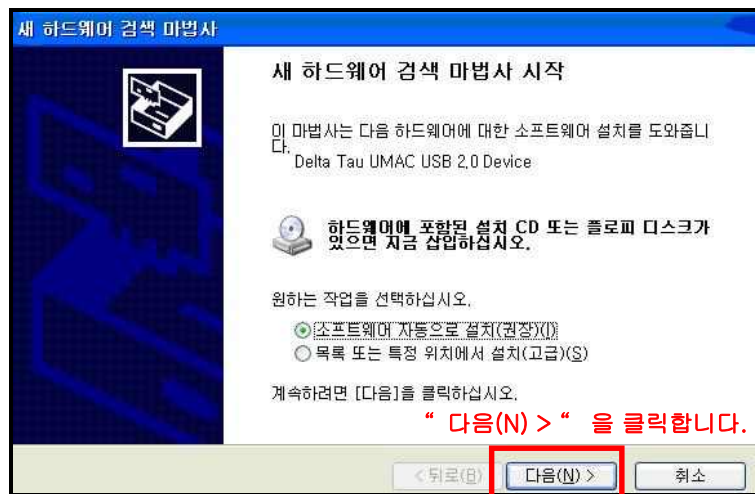
< PMAC Device 창에서 생성된 Serial 연결 선택 >

USB Cable 연결 (권장하지 않음)

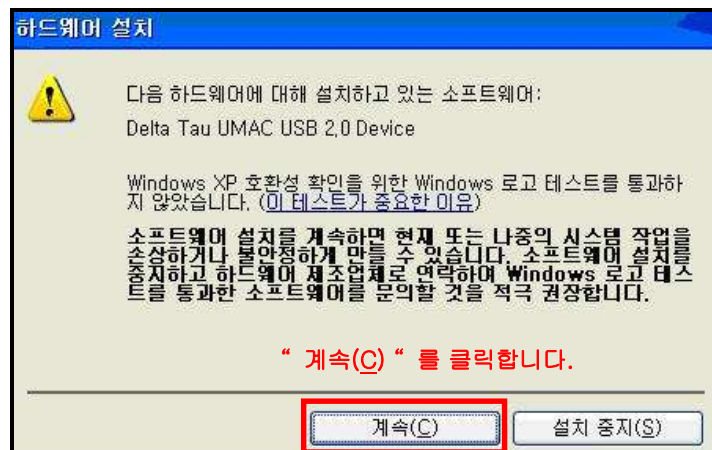
1. UMAC과 PC를 USB Cable로 연결하고 PC에서 Delta Tau UMAC USB 2.0 Device를 인식하면 소프트웨어를 설치합니다.



< PC 에서 Delta Tau UMAC USB 2.0 Device 인식 >



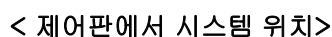
< Delta Tau UMAC USB 2.0 Device 소프트웨어 검색 >



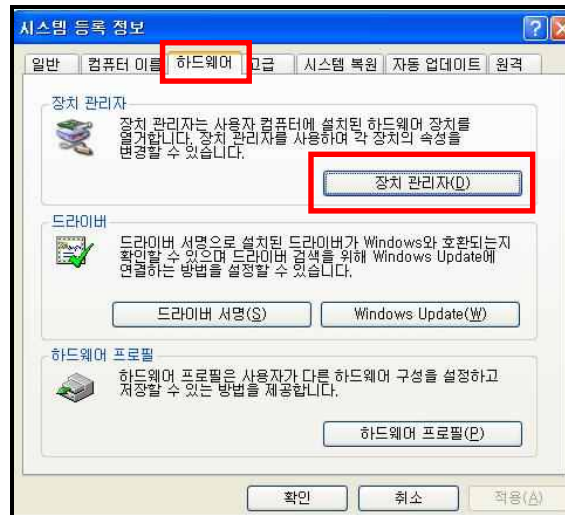
< Delta Tau UMAC USB 2.0 Device 소프트웨어 설치 >



2. PC의 장치관리자에서 범용 직렬 버스 컨트롤러 부분에 Delta tau UMAC USB 2.0 Device가 있는지 확인합니다.
 - PC의 제어판에서 시스템을 클릭합니다.

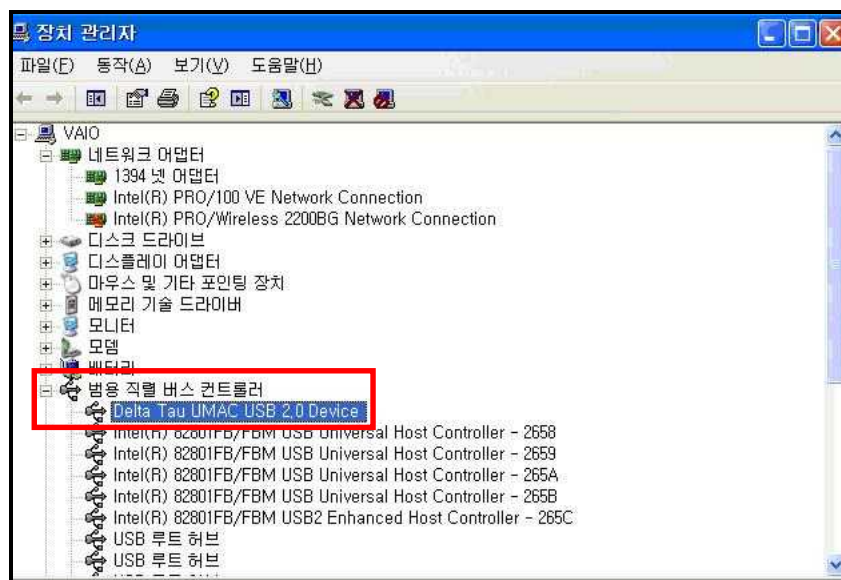


- 시스템 등록정보에서 하드웨어 메뉴의 장치관리자를 클릭합니다.



< 시스템 등록정보에서 장치관리자 위치 >

- 장치관리자에서 범용 직렬 버스 컨트롤러 부분에 Delta Tau UMAC USB 2.0 Device 가 있는지 확인합니다.



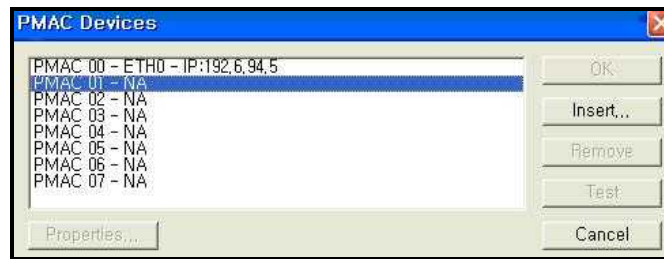
< 장치관리자에서 Delta Tau UMAC USB 2.0 Device 확인 >

3. PEWIN32Pro의 Setup 메뉴에서 Force all windows to device number를 클릭합니다.



< Force All Windows to Device Number 위치 >

4. 새로 열린 PMAC Device 창에서 NA로 되어있는 부분을 클릭하고 Insert를 클릭합니다.



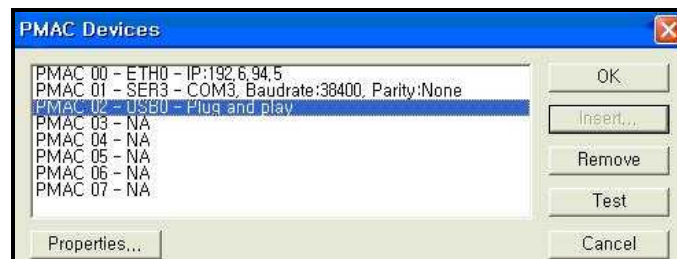
< PMAC Device 창에서 Insert 위치 >

5. 새로 뜬 Available PMAC Device창에서 Pmac USB0 을 선택하고 OK를 클릭합니다.



< Available PMAC Device 창에서 Pmac USB0 선택 >

6. PMAC Device창에서 생성된 USB 연결(USB0-Plug and Play)를 선택하고 Test를 실행한 뒤 성공하면 OK를 클릭해서 종료합니다.



< PMAC Device 창에서 생성된 USB 연결 선택 >

7. PC의 ground와 UMAC의 ground가 전위차가 있을 경우 위험할 수 있기 때문에 권장하지 않습니다.

LAN(Ethernet) 연결(권장)

최초 Pewin32Pro 를 설치 후 통신이 가능하기 위해서는 Window register 에 UMAC 의 IP 에 대한 정보를 입력해야 하며, UMAC 과 HOST PC 와의 Ping TEST 를 하여 케이블 상태를 확인 합니다. 이에 대해 문제가 없을 경우에 Pewin32Pro 를 실행 한 후 통신 테스트를 마친 후 연결을 시도합니다.

1. UMAC 의 IP 주소는 192.6.94.5 입니다. 이 주소를 PC window registry 에 저장합니다.
 - PC 에서 시작 → 프로그램 → PEWIN32Pro → Configure Ethernet 2.0 for PMAC 를 클릭합니다.

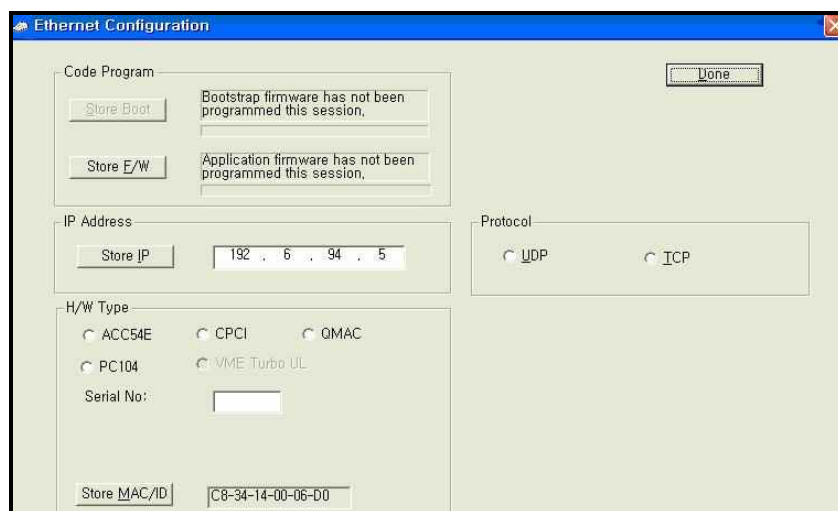


< Configure Ethernet 2.0 for PMAC 위치 >

- 새로 뜨는 Eth2Configure 창에서 확인을 누르고 기다리면 Ethernet Configuration 창이 뜹니다.

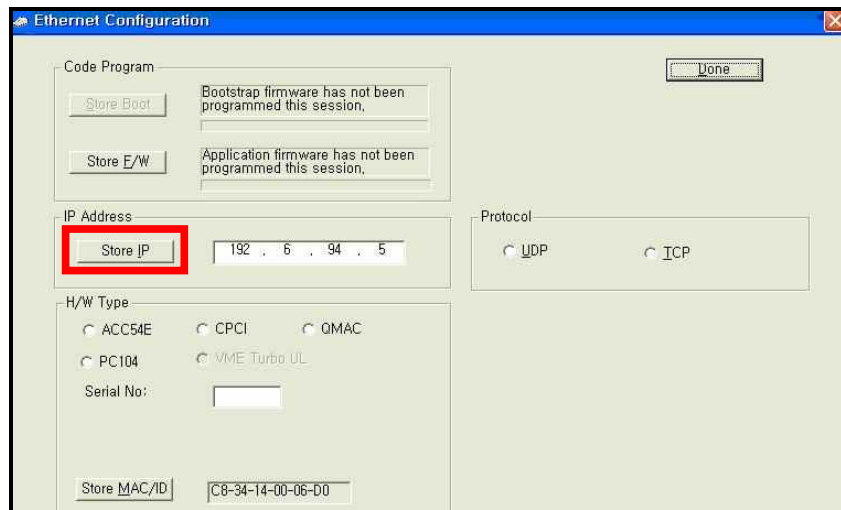


< 새로 뜨는 Eth2Configure 창 >



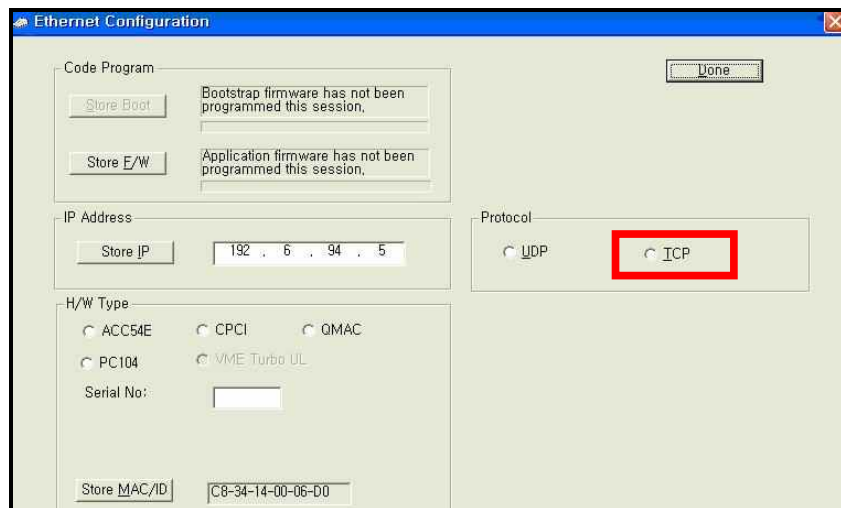
< 확인을 누르고 기다리면 나오는 Ethernet Configuration 창 >

- Store IP 부분에 현재 UMAC 의 IP 주소가 있습니다. Store IP 를 클릭하고, 새롭게 뜨는 창들은 '예' 또는 'OK' 를 클릭합니다.



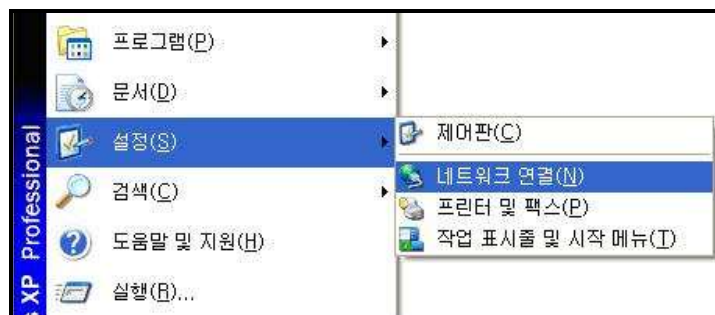
< Ethernet Configuration창에서 Store IP 부분 >

- TCP 부분을 클릭하고 새롭게 뜨는 창들은 ‘ 예 ’ 또는 ‘ OK ’ 를 클릭합니다.



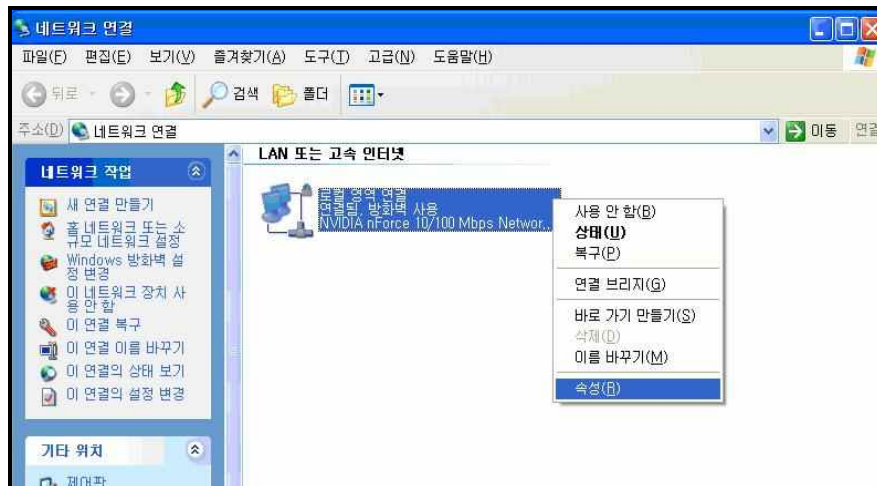
< Ethernet Configuration창에서 TCP 부분 >

- Done 을 클릭해서 창을 닫습니다.
2. PC의 IP 주소를 192.6.94.2(5 를 제외한 나머지 숫자)로 설정하고 서브넷 마스크는 255.255.255.0 로 설정하고 나머지 항목은 비웁니다.
- PC 에서 시작 → 설정 → 네트워크 연결을 클릭합니다.



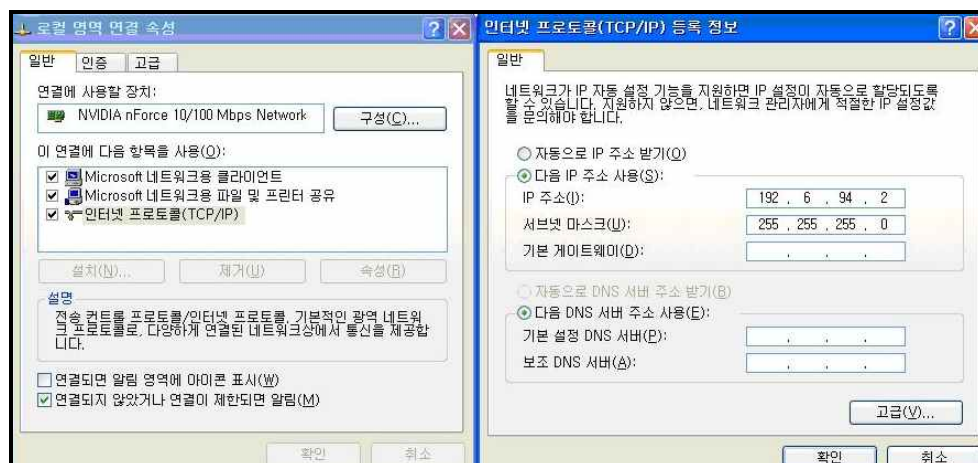
< 네트워크 연결 위치 >

- 네트워크 연결 창에서 로컬 영역 연결에 마우스 오른쪽 버튼을 클릭하고 속성을 클릭합니다.



< 로컬 영역 연결에서 속성 위치 >

- 로컬 영역 연결 속성에서 인터넷 프로토콜(TCP /IP)를 클릭하고 속성을 클릭합니다.
- 인터넷 프로토콜(TCP /IP) 등록 정보 창에서 IP 주소와 서브넷 마스크를 입력합니다.
- IP 주소 : 192.6.94.2(5 를 제외한 나머지 숫자), 서브넷 마스크 : 255.255.255.0

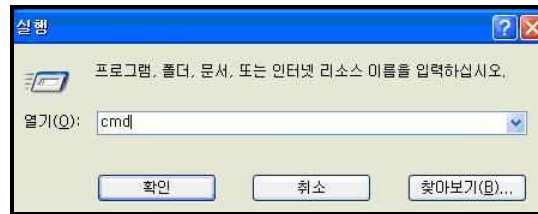


< IP주소와 서브넷 마스크 설정 위치 >

- UMAC과 PC를 LAN(Ethernet) Cable 연결하고 UMAC전원을 켭니다.
- PC의 Network 카드가 활성화 되어 있는지 확인합니다.
- Ping test를 해서 연결이 잘 되었는지 확인합니다.(PC에서 UMAC으로)
 - PC에서 시작 → 실행을 클릭하여 실행창에서 cmd를 입력합니다.

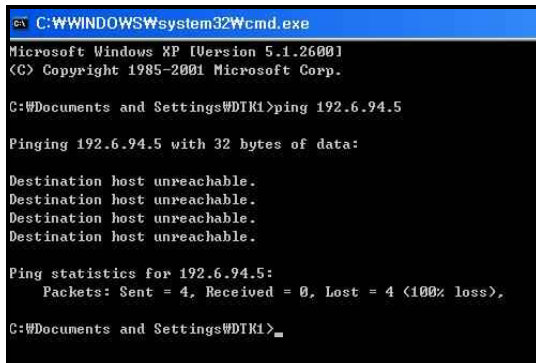


< 실행창 위치 >

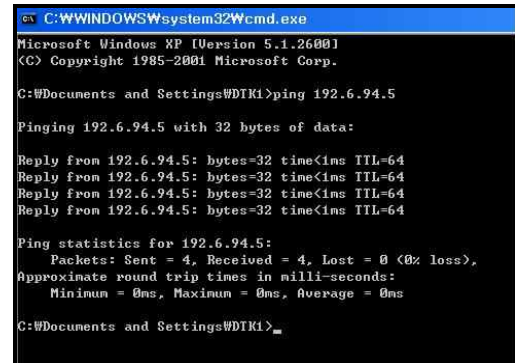


< 실행 창 >

- 새롭게 열린 창에서 ping 192.6.94.5 를 입력하여 실행시켜서 연결 여부를 확인합니다.
연결이 안 될 경우에는 위의 과정을 다시 한 번 확인하고 Service Pack 5.1 까지 설치되어 있는지 확인합니다.

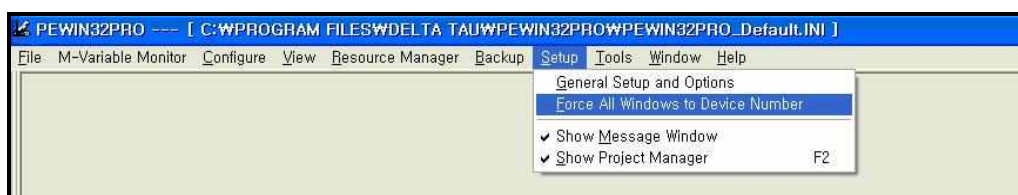


< 연결 실패 >



< 연결 성공 >

6. PEWIN32Pro의 Setup메뉴에서 Force all windows to device number를 클릭합니다.



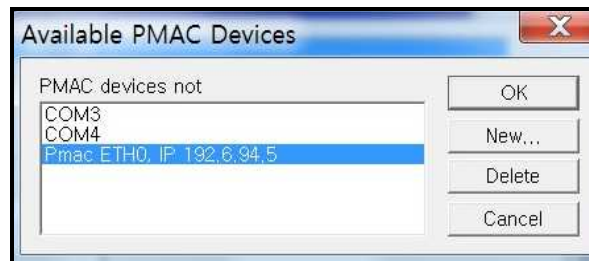
< Force All Windows to Device Number 위치 >

7. 새로 열린 PMAC Device 창에서 NA로 되어있는 부분을 클릭하고 Insert를 클릭합니다.



< PMAC Device 창에서 Insert 위치 >

8. 새로 뜬 Available PMAC Device창에서 Pmac ETH0, IP 192.6.94.5 을 선택하고 OK를 클릭합니다.



< Available PMAC Device 창에서 Pmac ETH0, IP 192.6.94.5 선택 >

9. PMAC Device창에서 생성된 Ethernet 연결(ETH0, IP 192.6.94.5)를 선택하고 Test를 실행한 뒤 성공하면 OK를 클릭해서 종료합니다.



< PMAC Device 창에서 생성된 Ethernet 연결 선택 >

PEWIN32Pro 의 시작

PEWIN32 Pro 소개

PMAC 의 셋업(setup), 디버그(debug) 및 진단(diagnose)을 위한 필수적인 프로그램으로 PMAC Plot pro, PMAC Tuning Pro 와 같이 사용합니다.

- 기능

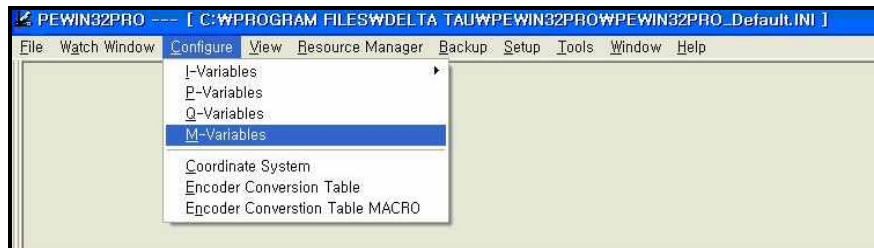
- PMAC 에 온라인 명령사용
- 위치(positions), 속도(velocities), 및 추종에러 (following errors) 표시
- 모터, 좌표계 및 모든 상태정보를 표시
- 모든 PMAC 변수의 표시, 변경 및 조회 가능
- 모든 PMAC 정보의 백업(backup) 및 환원(restore)

Download Suggested M-variables

M 변수는 PMAC 의 메모리와 I/O 를 액세스하는 용도로 사용하는 변수로서, UMAC 의 메모리와 I/O 에 대해서 기본적으로 설정되어 있는 M 변수를 UMAC 에 Download 하는

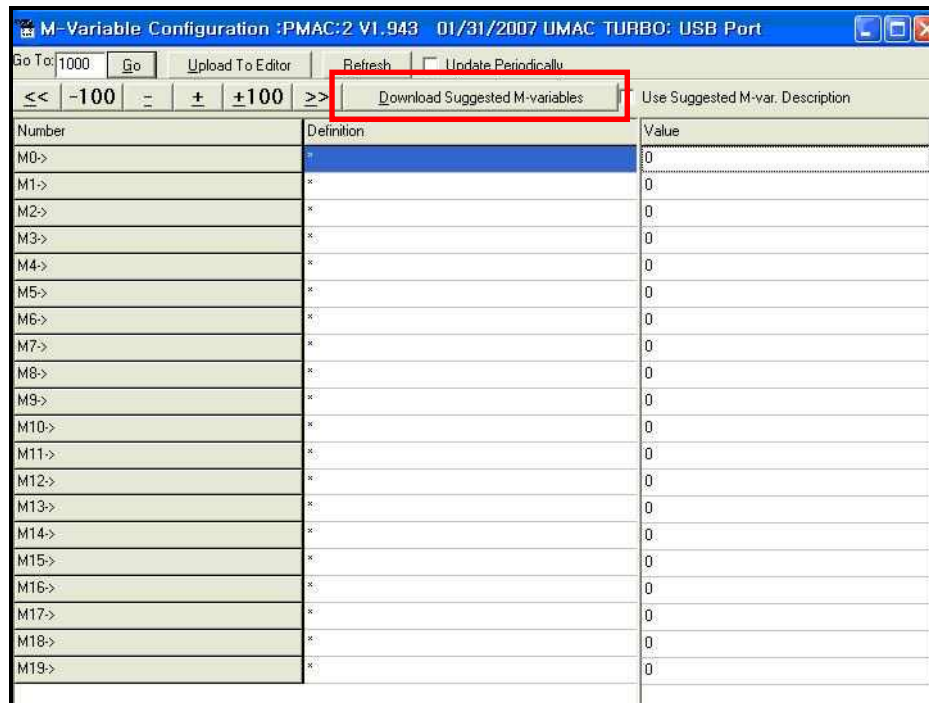
작업입니다. Download 하지 않고 사용자가 임의로 설정할 수 있지만 UMAC 구동중에 문제가 발생하여 당사에서 디버깅을 할 경우 M 변수의 의미를 이해할 수 없어 많은 시간이 소요됩니다.

1. PEWIN 32 Pro 의 Configure 메뉴에서 M-variables 를 클릭합니다.



< Configure 메뉴의 M-variables 위치 >

2. M-variables Configuration 창에서 Download Suggested M-variables 를 클릭합니다.



< M-variables Configuration 창에서 Download Suggested M-variables 위치 >

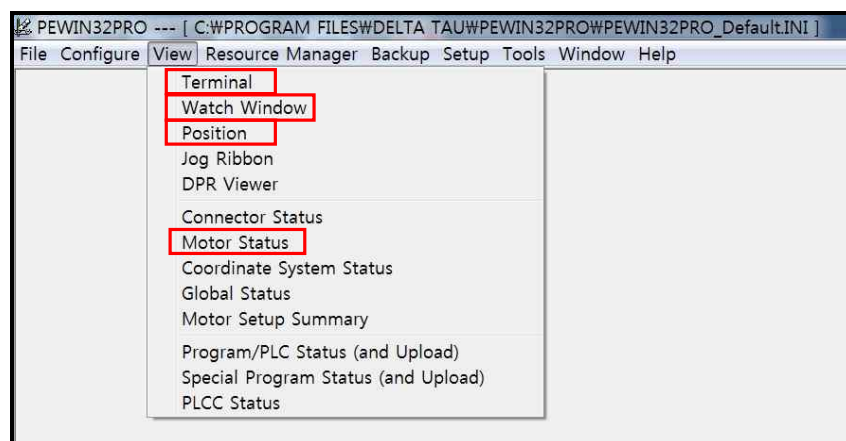
3. Use Suggested M-var. Description 을 클릭하면 각각의 M 변수 설명을 확인할 수 있습니다.

Number	Description	Definition	Value
M0->	I/O00 Data Line; J3 Pin 1	Y:\$78400,0	1
M1->	I/O01 Data Line; J3 Pin 2	Y:\$78400,1	1
M2->	I/O02 Data Line; J3 Pin 3	Y:\$78400,2	1
M3->	I/O03 Data Line; J3 Pin 4	Y:\$78400,3	1
M4->	I/O04 Data Line; J3 Pin 5	Y:\$78400,4	1
M5->	I/O05 Data Line; J3 Pin 6	Y:\$78400,5	1
M6->	I/O06 Data Line; J3 Pin 7	Y:\$78400,6	1
M7->	I/O07 Data Line; J3 Pin 8	Y:\$78400,7	1
M8->	I/O08 Data Line; J3 Pin 9	Y:\$78400,8	1
M9->	I/O09 Data Line; J3 Pin 10	Y:\$78400,9	1
M10->	Data Line; J3 Pin 11	Y:\$78400,10	1
M11->	I/O11 Data Line; J3 Pin 12	Y:\$78400,11	1
M12->	I/O12 Data Line; J3 Pin 13	Y:\$78400,12	1
M13->	I/O13 Data Line; J3 Pin 14	Y:\$78400,13	1
M14->	I/O14 Data Line; J3 Pin 15	Y:\$78400,14	1
M15->	I/O15 Data Line; J3 Pin 16	Y:\$78400,15	1
M16->	I/O16 Data Line; J3 Pin 17	Y:\$78400,16	1
M17->	I/O17 Data Line; J3 Pin 18	Y:\$78400,17	1
M18->	I/O18 Data Line; J3 Pin 19	Y:\$78400,18	1
M19->	I/O19 Data Line; J3 Pin 20	Y:\$78400,19	1

< M-variables Configuration 창에서 Description 표시 >

View 메뉴에서 Terminal, Watch Window, Position, Motor Status 실행

1. Terminal : 온라인 명령 실행, 변수 설정 등을 할 수 있는 창입니다..
2. Watch Window : I, M, P, Q 변수의 값을 모니터 할 수 있는 창입니다.
3. Position : 모터들의 현재 위치, 속도, following error 를 확인 할 수 있는 창입니다.
(position 창에서 마우스 오른쪽 클릭 → Velocity 와 Following Error 클릭)
4. Motor Status : AMP Enable 이나 Home Complete 등 모터의 현재 상태를 확인 할 수 있는 창입니다.



< Pwin 32Pro 의 View 메뉴>

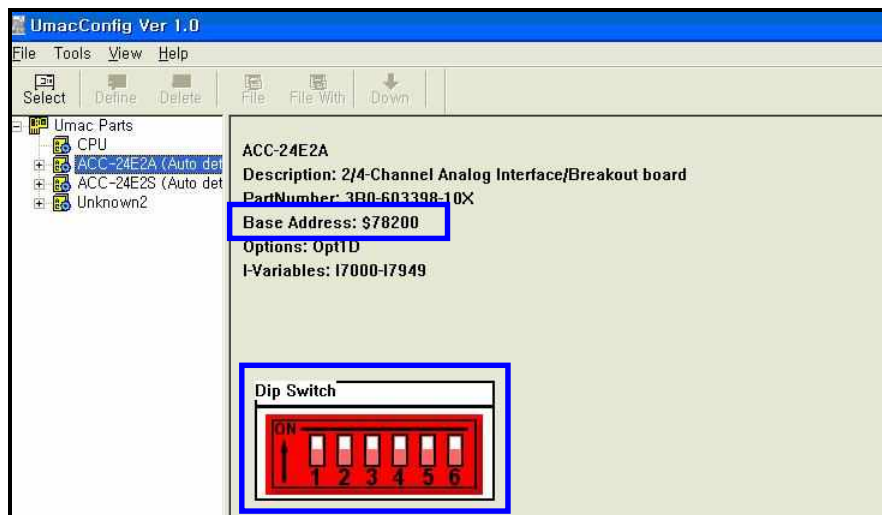
빠른 응답을 위한 Thread Period, priority 설정

Terminal, Watch Window, Position, Motor Status 창에서 마우스 오른쪽 버튼을 클릭해서 set thread priority 나 set thread period delay 를 클릭해서 모두 100 으로 설정합니다. Default 상태인 1000 으로 설정되어 있으면 각 창의 업데이트가 1 초마다 실행됩니다.

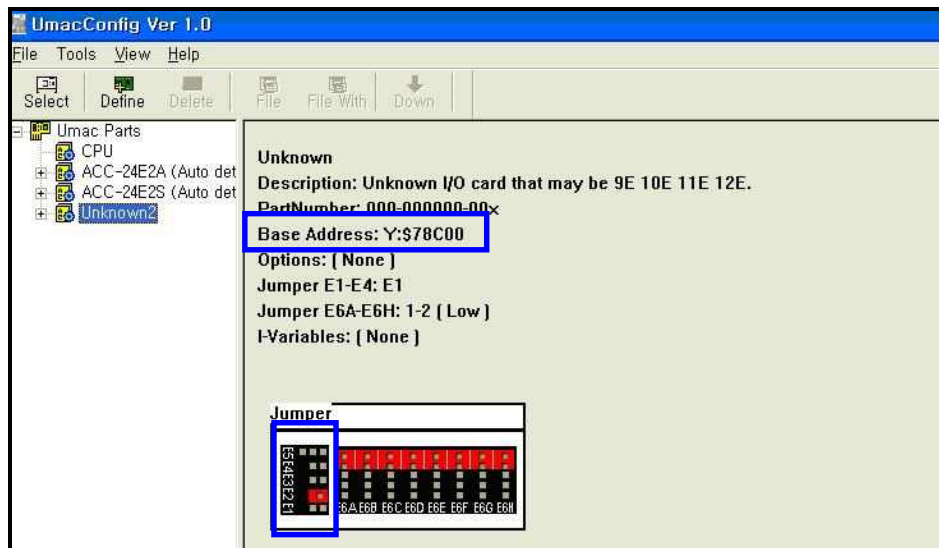
UMAC Config Pro 를 이용한 Servo IC 설정상태 확인

UMAC Config Pro 를 실행하면 UMAC 에 연결되어 있는 카드들의 종류와 수량, 각각의 카드들에 대한 정보를 확인할 수 있습니다.

1. UMAC Config Pro 실행하기 -> PEWIN32Pro 의 Tool 메뉴에서 UMAC Config Pro 클릭
2. 카드들의 정보 확인 : UMAC Config 창에서 좌측의 카드 이름을 클릭하면 해당 카드에 대한 정보를 확인할 수 있습니다. ACC-24E2A 는 Dip Switch 및 Base Address 를 확인할 수 있고 ACC-11E 는 Base Address 와 E1~E4 Jumper 를 확인할 수 있습니다.



< UMAC Config 에서 ACC-24E2A 의 Base Address, Dip Switch 확인>



< UMAC Config 에서 ACC-11E 의 E1 ~ E4 Jumper 확인 >

Data Backup

현재 UMAC 에서 실행중인 프로그램 및 각종 변수들 등을 PC 로 Backup 하거나 PC 로 Backup 한 파일을 다시 UMAC 에서 실행하기 위한 방법이 PEWIN32Pro 의 Backup 메뉴에 있습니다. Backup 메뉴를 실행할 때는 사용중인 PLC 를 모두 Disable 시키고 Jog Ribbon 을 종료해야 합니다.

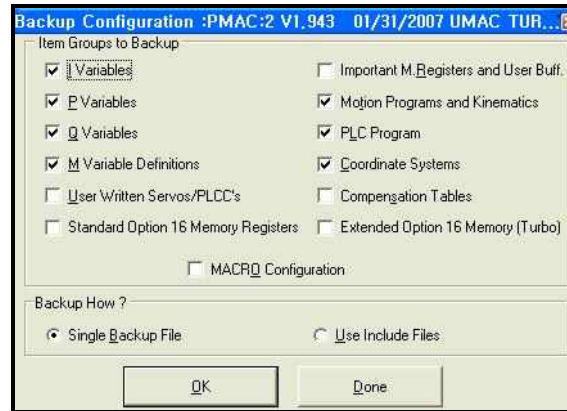


< PEWIN 32 Pro 의 Backup 메뉴 위치 >

Upload Configuration

현재 UMAC 에서 설정하고 사용중인 data(변수, 프로그램 등)중에서 Backup 을 원하는 항목을 선택하여 PC 로 전송하여 파일로 저장합니다.

일반적으로 Default 상태로 설정되어있는 I 변수, P 변수, Q 변수, M 변수, Motion 프로그램, PLC 프로그램, Coordinate Systems 을 선택하여 Upload 합니다.



< Backup data 선택 화면 >

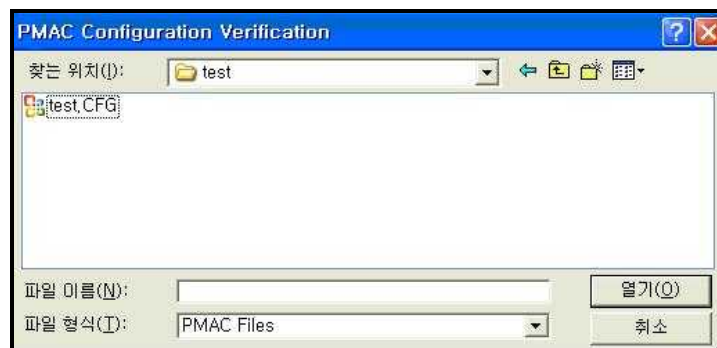
Restore configuration

PC 에 Backup 한 파일(확장자가 CFG 인 파일)을 다시 UMAC 으로 Restore 시킵니다.

Verify configuration

현재 UMAC 의 data 와 Backup 된 파일의 data 에서 다른 부분을 확인합니다.

- PC 에서 backup 된 파일(확장자가 CFG 인 파일)을 선택합니다.



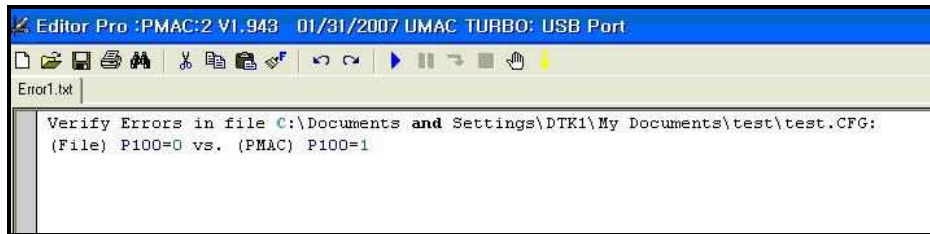
< Backup 된 파일 선택 >

- verify comfiguration 이 완료되었을 때 현재 UMAC data 와 backup 파일이 서로 같으면 PEWIN 32 Pro 의 하단에 아래 그림과 같이 표시됩니다.



< Verify configuration 완료 (다른 부분 없음) >

- verify comfiguration 이 완료되었을 때 현재 UMAC data 와 backup 파일이 서로 다르면 PEWIN 32 Pro 의 Editor 창이 생기고 어떤 data 가 다른지 표시됩니다.



< Verify configuration 완료 (다른 부분 있음) >

Servo IC 와 Servo IC channel 표기 설명(m,n)

1. Servo IC 와 관련된 7000 번대 i 변수 표기를 할 때에는 i7mn□과 같이 m 과 n 을 써서 표기를 합니다. 여기서 m 은 Servo IC Number 이고, n 은 Servo IC Channel Number 입니다. □는 숫자입니다.
2. Servo IC Number : m
Servo IC Board 의 dip switch 로 설정합니다. UMAC 에서는 2 부터 시작합니다.
UMAC 에 2 개의 Servo IC Board 가 있는 경우에 첫 번째 Servo IC Board 는 2 로 설정하고 두 번째 Servo IC Board 는 3 으로 설정합니다.
3. Servo IC channel Number : n
각 Servo IC Board 는 4 개의 channel 을 사용가능하기 때문에 Servo IC channel Number 는 1 ~ 4 까지 있습니다.
4. Servo IC channel 번호와 Motor 번호
일반적으로 첫번째 Servo IC Board(m:2)의 첫번째 channel(n:1)이 1 번 모터가 됩니다.
M=2, n=1 → 1 번 모터
M=2, n=2 → 2 번 모터
M=2, n=3 → 3 번 모터
M=2, n=4 → 4 번 모터
M=3, n=1 → 5 번 모터
M=3, n=2 → 6 번 모터
M=3, n=3 → 7 번 모터
M=3, n=4 → 8 번 모터

Motor 구동

모터를 구동하는 방식에는 속도모드, 토크모드, 위치모드(Stepping Motor Control) 세 가지 방식이 있습니다. 상위제어기인 PMAC 과 Servo Driver 의 제어방식에 따라 나누어지고 모터를 튜닝하는 방법도 다릅니다.

토크모드(Torque Mode)

1. 위치제어와 속도제어까지 상위제어기인 PMAC 에서 실행합니다.
2. ACC-24E2A Board 를 사용하며 $\pm 10V$ DAC 출력을 발생합니다.
3. HOST Controller 에서 PID 튜닝을 하며 드라이버에서 튜닝이 필요 없습니다.
(단, 드라이버의 제한에 관련한 파라미터를 풀어 주셔야 합니다.)
4. I 변수 설정
 - lxx00 : Motor xx Activation Control (xx = 모터 번호)
모터를 구동하기 위해서는 lxx00 변수를 1 로 설정해서 모터를 Active 시켜야 합니다. 모터 1 의 경우 초기값이 1 로 설정되어 있지만 다른 모터들은 초기값이 0 으로 설정되어 있습니다.
 - lxx19 : 최대 허용 조그 가속도 (Maximum Permitted Jog Acceleration)
(units : cts/msec²)
모터의 스펙에 맞는 최대로 허용 가능한 가속도 값보다 낮게 설정합니다.
모터 스펙 이상의 값을 설정할 경우 모터가 고장날 수 있습니다.
 - lxx20 : 가속시간 (Acceleration Time)
(units: msec; integer)
S-curve 시간 없이 가속시간만 있을 경우에는 가속도 변화에 있어서 불연속점이 생겨서 모터가 불안정해질 수 있습니다. 시스템에 맞게 적절한 가속시간과 S-curve 시간을 설정해야 합니다.
 - lxx21 : “ S ” 커브 시간 (S-Curve Time)
(units: msec; integer)
lxx21 * 2 값이 lxx20 보다 큰 경우 lxx20 의 값을 무시하고 lxx21 * 2 값이 전체 가속도 시간이 되며, 이 때를 full S라고 합니다. 이 경우에 S-Curve 중간부분에서 최대 가속도가 발생하고, 토크가 약한 모터의 경우 문제가 발생할 수 있습니다.
 - lxx22 : 조그 속도(Motor xx Jog Speed)
(units : counts / msec)

모터의 구동 속도를 결정합니다. 튜닝전에는 저속(lxx22=1)으로 설정하고
조금씩 증가시키면서 모터를 구동합니다.

- I7mn6 : Servo IC m channel n Output Mode Select

(m : Servo IC 번호, n : channel 번호)

I7mn6 = 0 : output A, B, C 를 PWM 으로 사용합니다.

I7mn6 = 1 : output A, B 는 DAC 로 사용하고 output C 는 PWM 으로
설정합니다.

I7mn6 = 2 : output A, B 는 PWM 으로 사용하고 output C 는 PFM 으로
설정합니다.

I7mn6 = 3 : output A, B 는 DAC 로 사용하고 output C 는 PFM 으로
설정합니다.

토크모드에서는 DAC(Digital to Analog Converter)로 사용하기 때문에
1 또는 3 으로 설정합니다. (I7mn6=1 or I7mn6=3)

Note:

lxx00 : 메뉴얼 TURBO SRM 124 페이지

lxx19 ~ lxx22 : 메뉴얼 TURBO SRM 142 ~ 144 페이지

I7mn6 : 메뉴얼 TURBO SRM 259 페이지

5. PID 튜닝

- PID Tuning 을 안 하면 모터 구동 명령을 실행해도 모터가 조금만 움직이고 멈추게
되고 PID Tuning 을 정확히 안 하면 명령만큼 정확히 움직이지 않습니다. 정확한 모터
구동을 위해서는 PID Tuning 을 적절하게 해야합니다.
- PEWIN 32 Pro 의 Tool 메뉴(ALT+T)에 들어가서 PMAC Tuning Pro 를 실행합니다.



< PMAC Tuning Pro 위치 >

- PMAC Tunig Pro 에서 모터를 선택하고 Interactive Tuning 을 실행합니다.



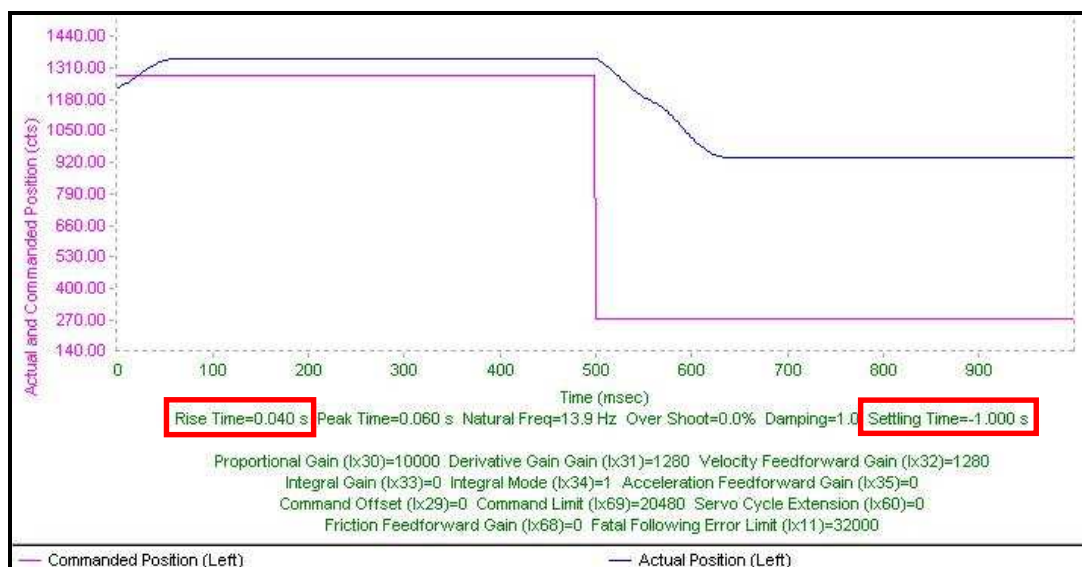
< 모터 선택 및 Interactive Tuning 위치 >

- lxx34 (IM)=1 (항상 I Gain 을 보상하는 기능입니다. Tuning 중에만 1 로 설정합니다.)
- lxx30 (P gain) 을 2~3 배 증가시키면서 Do A step Move 를 클릭해서 그래프가 Commanded position 그래프 형태와 비슷해지고 Rise time 이 빨라지도록 모터에서 진동이나 소음이 없을때까지 증가시킵니다.



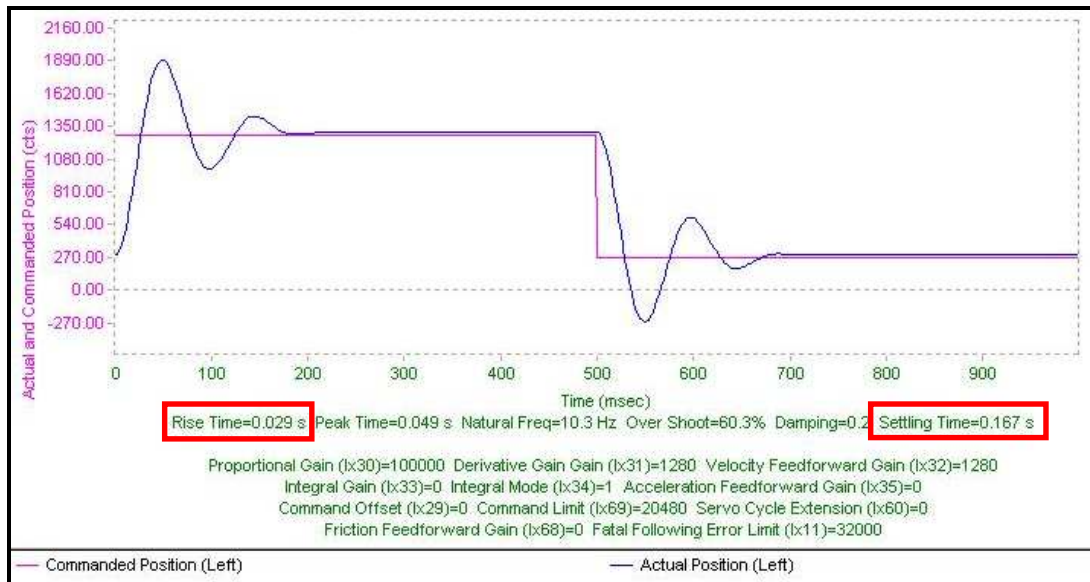
< “ Do A Step Move ” 위치 >

- lxx30 (P gain)이 낮아서 Rise Time 이 낮고 Commanded Position 을 따라가지 못합니다.



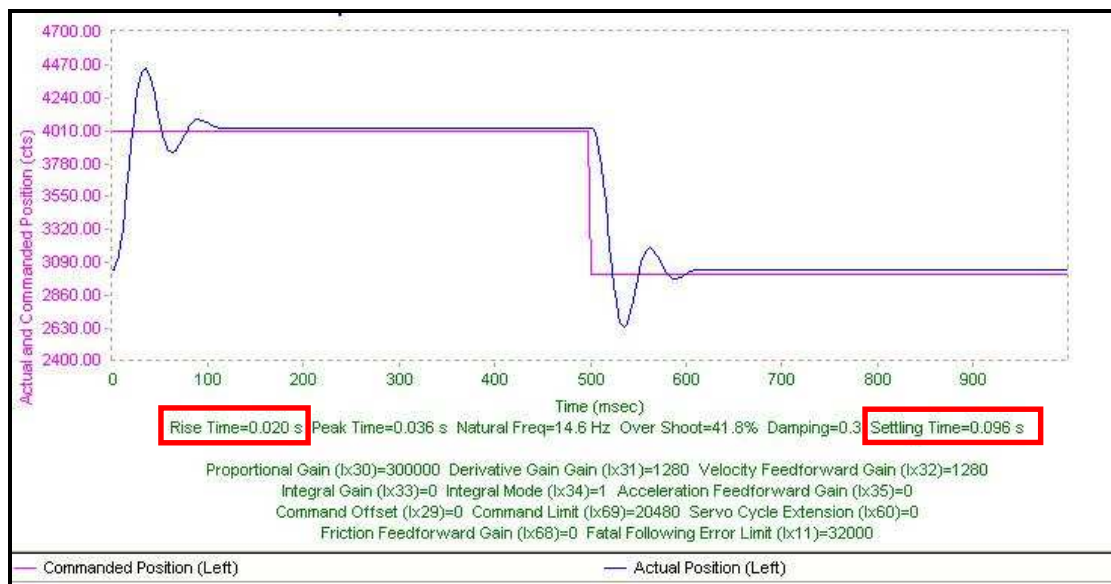
< lxx30=10000, lxx31=1280, lxx32=1280, lxx33=0, lxx34=1, lxx35=0 Graph >

- lxx30 (P gain)을 10 배 증가시킨 Graph 입니다. 진동(Oscilation)이 발생했지만 Rise Time 이 빨라지고 Commanded Position 을 따라가고 있습니다.



< lxx30=100000, lxx31=1280, lxx32=1280, lxx33=0, lxx34=1, lxx35=0 Graph >

- lxx30 (P gain)을 다시 3 배 증가시킨 Graph 입니다. 진동(Oscilation)이 있지만 Rise Time 과 Settling Time 이 더 빨라진 것을 확인할 수 있습니다.

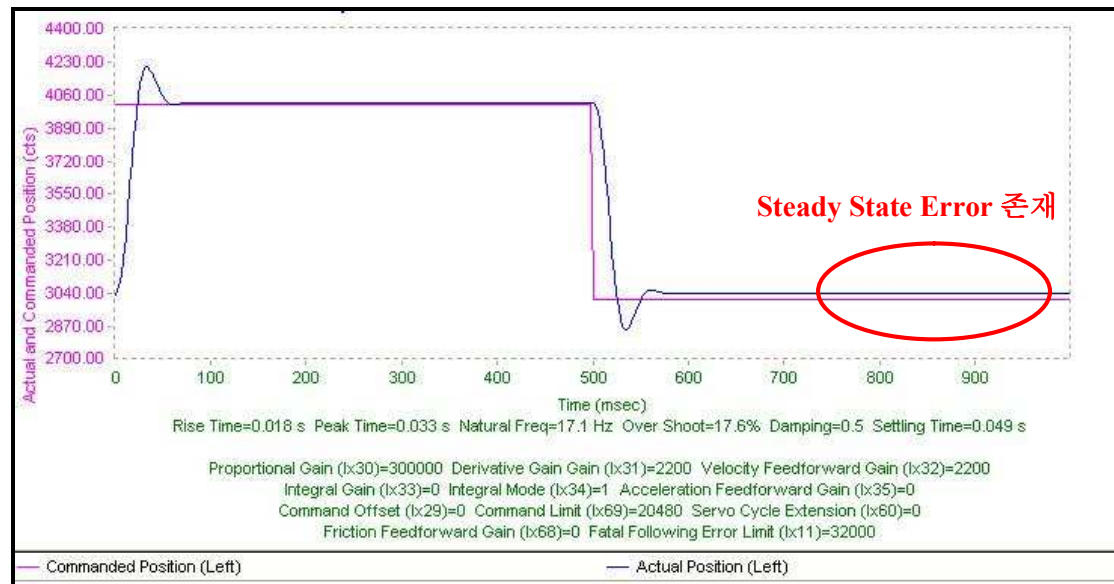


< lxx30=300000, lxx31=1280, lxx32=1280, lxx33=0, lxx34=1, lxx35=0 Graph >

- lxx31 (D gain) 과 lxx32 (Vff)를 같이 증가시켜서(100~200 씩) 그래프의 진동을 줄입니다.
- lxx31 (D gain) 과 lxx32 (Vff)를 2200 으로 증가시킨 Graph 입니다. 진동이 많이 줄어든

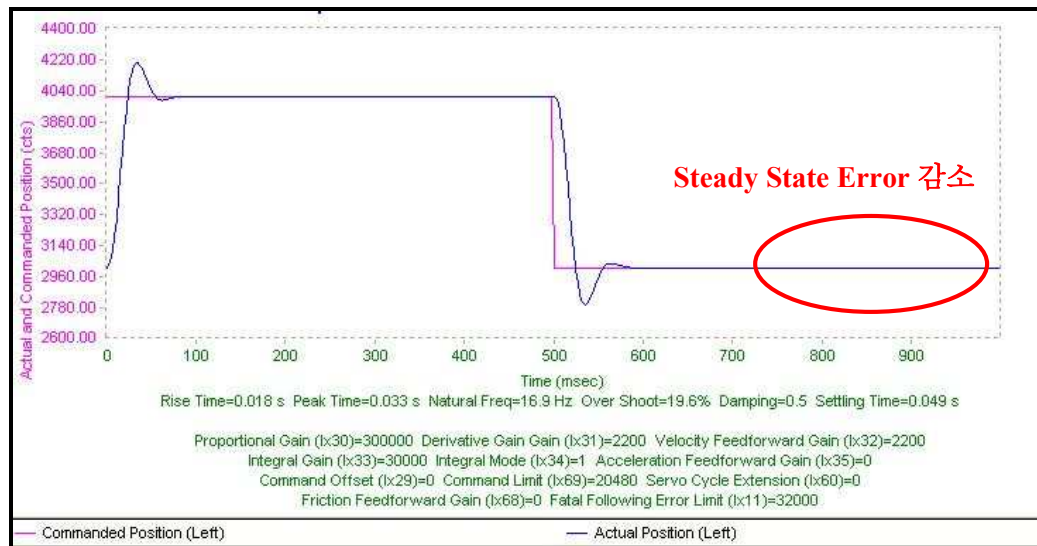
것을 확인할 수 있습니다. 하지만 Steady State Error 가 존재합니다.

* Steady State Error : Commanded Position 값과 Actual Position 값의 차이



< lxx30=300000, lxx31=2200, lxx32=2200, lxx33=0, lxx34=1, lxx35=0 Graph >

- lxx30 은 커도 좋지만 lxx31 은 필요 이상으로 크게 하지 않습니다.
- Steady State Error 값을 줄이기 위해서 lxx33 (I gain)을 30000 으로 설정합니다.
(overshoot 이 큰 경우는 작은 값으로 설정합니다.)



< lxx30=300000, lxx31=2200, lxx32=2200, lxx33=30000, lxx34=1, lxx35=0 Graph >

- lxx35 (Aff)설정을 위해서 Parabolic Move 를 실행합니다.

Trajectory Selection 에서 Parabolic Velocity 를 선택하고 Move Time 을 200ms 로 설정합니다.

Present PID Terms

lxx30 (Kp) : 300000
lxx31 (Kd) : 2200
lxx32 (Kvff) : 2200
lxx33 (Ki) : 30000
lxx34 (IM) : 1
lxx35 (Kaff) : 0
lxx29 : 0
lxx69 : 20480
lxx60 : 0
lxx68 : 0
lxx11 : 32000

Implement Auto-Tuning Gains
Implement Original Gains

PID Diagram

Trajectory Selection
☐ Position Step
☐ Position Ramp
☒ **Parabolic Velocity**
☐ Trapezoidal Velocity
☐ S-Curve Velocity
☐ Sinusoidal
☐ Sine Sweep
☐ User Defined

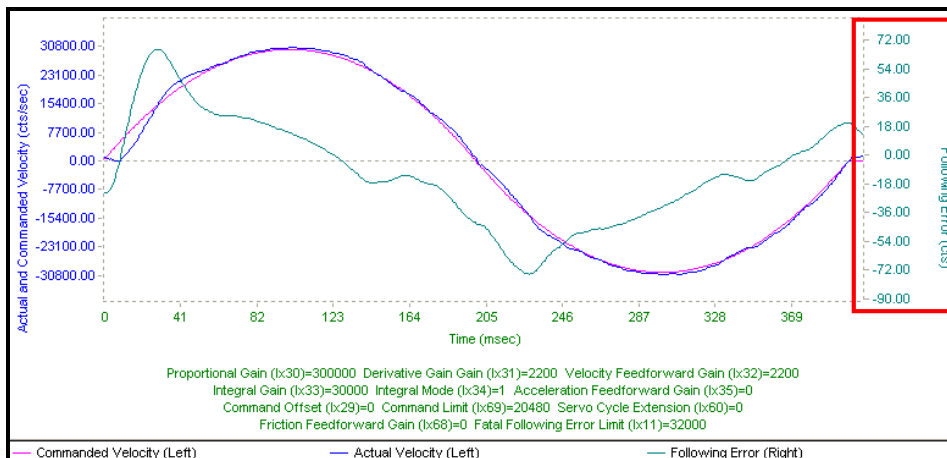
Parabolic Move
Move Size (cts): 4000
Move Time (ms): 200
Do A Parabolic Move
☐ Kill Motor After Parabolic Move
☐ Move in only one direction
☐ Dwell Time After Move : 500 (ms)

Left Axis Plot: Velocity
Right Axis Plot: Following Error
Coordinate System definition not saved.

Enable Motor #1
Exit
<< >>

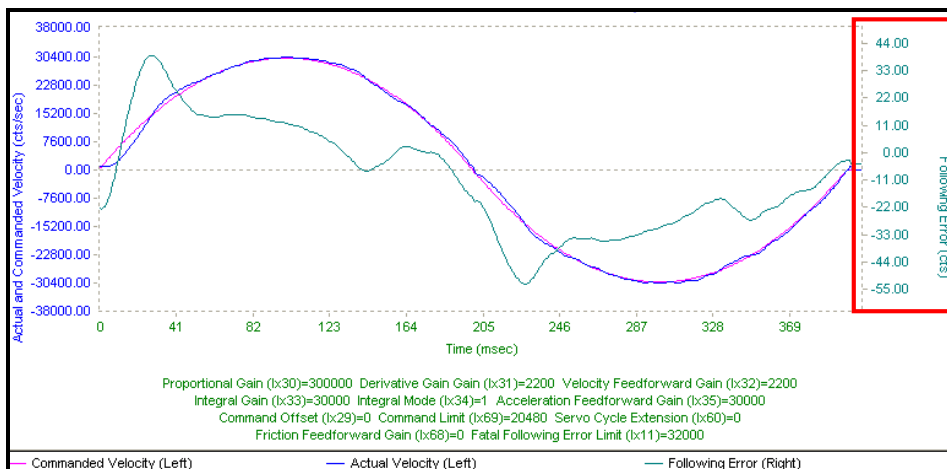
Notch Filter Calculator **Low Pass Filter Calculator**
Notch/Low Pass Filter Setup
Notch/Low Pass Filter Inactive

< Parabolic Move 설정 >



< Parabolic Move 실행 결과 >

- lxx35 (Aff)를 증가시켜서 Following Error 를 줄입니다. Following Error 파형이 Actual and command Velocity 파형과 반대의 형태가 될 경우에는 lxx35 (Aff) 값을 줄입니다.



< lxx30=300000, lxx31=2200, lxx32=2200, lxx33=30000, lxx34=1, lxx35=30000 Graph >

- lxx34 (IM) =0 (정지할때만 I Gain 보상. Tuning 이 끝나면 0 으로 설정합니다.)

Note:

lxx30 ~ lxx35(PID Tuning 관련 I 변수) : 메뉴얼 TURBO SRM 153 ~ 155 페이지

6. 토크모드 변수 설정 예(1 번 모터)

- l100=1 // 1 번 모터 activation
- l7216=3 // Output Mode : DAC
- l119=1 // 최대 허용 조그 가속도 = 1counts/msec²
- l120=100 // 가속시간 = 100msec
- l121=20 // S 커브 시간 = 20*2 = 40msec
- l122=32 // 조그 속도 = 32counts/msec
- l130=300000 // P gain = 300000
- l131=2200 // D gain = 2200
- l132=2200 // Vff = 2200
- l133=30000 // I gain = 30000
- l134=0 // IM = 0
- l135=0 // Aff = 0

속도모드(Velocity Mode)

1. 위치제어는 상위제어기인 PMAC 에서 실행합니다.
2. ACC-24E2A Board 를 사용하며 ±10V DAC 출력을 발생합니다.
3. Servo Driver 에서 속도제어루프 튜닝이 필요합니다. Servo Driver 에서 튜닝이 안 되어 있으면 UMAC 에서 PID 튜닝을 해도 정확한 튜닝이 어렵습니다.
먼저 Servo Driver 에서 튜닝을 완료하고 UMAC 에서 튜닝을 시작합니다.

4. I 변수 설정

- lxx00 : Motor xx Activation Control (xx = 모터 번호)
모터를 구동하기 위해서는 lxx00 변수를 1 로 설정해서 모터를 Active 시켜야 합니다. 모터 1 의 경우 초기값이 1 로 설정되어 있지만 다른 모터들은 초기값이 0 으로 설정되어 있습니다.
- lxx19 : 최대 허용 조그 가속도 (Maximum Permitted Jog Acceleration)
(units : cts/msec²)
모터의 스펙에 맞는 최대로 허용 가능한 가속도 값보다 낮게 설정합니다.
모터 스펙 이상의 값을 설정할 경우 모터가 고장날 수 있습니다.

- lxx20 : 가속시간 (Acceleration Time)
(units: msec; integer)
S-curve 시간 없이 가속시간만 있을 경우에는 가속도 변화에 있어서 불연속점이 생겨서 모터가 불안정해질 수 있습니다. 시스템에 맞게 적절한 가속시간과 S-curve 시간을 설정해야 합니다.
- lxx21 : “ S ” 커브 시간 (S-Curve Time)
(units: msec; integer)
lxx21 * 2 값이 lxx20 보다 큰 경우 lxx20 의 값을 무시하고 lxx21 * 2 값이 전체 가속도 시간이 되며, 이 때를 full S라고 합니다. 이 경우에 S-Curve 중간부분에서 최대 가속도가 발생하고, 토크가 약한 모터의 경우 문제가 발생할 수 있습니다.
- lxx22 : 조그 속도(Motor xx Jog Speed)
(units : counts / msec)
모터의 구동 속도를 결정합니다. 튜닝전에는 저속(lxx22=1)으로 설정하고 조금씩 증가시키면서 모터를 구동합니다.
- l7mn6 : Servo IC m channel n Output Mode Select
(m : Servo IC 번호, n : channel 번호)
l7mn6 = 0 : output A, B, C 를 PWM 으로 사용합니다.
l7mn6 = 1 : output A, B 는 DAC 로 사용하고 output C 는 PWM 으로 설정합니다.
l7mn6 = 2 : output A, B 는 PWM 으로 사용하고 output C 는 PFM 으로 설정합니다.
l7mn6 = 3 : output A, B 는 DAC 로 사용하고 output C 는 PFM 으로 설정합니다.
속도모드에서는 DAC(Digital to Analog Converter)로 사용하기 때문에 1 또는 3 으로 설정합니다. (l7mn6=1 or l7mn6=3)

Note:

lxx00 : 메뉴얼 TURBO SRM 124 페이지

lxx19 ~ lxx22 : 메뉴얼 TURBO SRM 142 ~ 144 페이지

l7mn6 : 메뉴얼 TURBO SRM 259 페이지

5. PID 튜닝

- PID 튜닝을 안 하면 모터 구동 명령을 실행해도 모터가 조금만 움직이고 멈추게 되고 PID 튜닝을 정확히 안 하면 명령만큼 정확히 움직이지 않습니다. 정확한 모터 구동을 위해서는 PID 튜닝을 적절하게 해야합니다.
- 속도모드에서의 PID 튜닝은 먼저 Servo Driver 튜닝을 끝내고 lxx31 (D gain)을 0 에 가까운 낮은 값으로 설정한 뒤 토크모드와 같은 방식으로 튜닝을 합니다.

Note:

lxx30 ~ lxx35(PID Tuning 관련 I 변수) : 메뉴얼 TURBO SRM 153 ~ 155 페이지

6. 속도모드 변수 설정 예(1 번 모터)

- l100=1 // 1 번 모터 activation
- l7216=3 // Output Mode : DAC
- l119=1 // 최대 허용 조그 가속도 = 1counts/msec²
- l120=100 // 가속시간 = 100msec
- l121=20 // S 커브 시간 = 20*2 = 40msec
- l122=32 // 조그 속도 = 32counts/msec
- l130=20000 // P gain = 200000
- l131=100 // D gain = 100
- l132=5000 // Vff = 5000
- l133=3000 // I gain = 3000
- l134=0 // IM = 0
- l135=10000 // Aff = 10000

Stepping Motor Control

Stepping Motor (일반적인 위치제어)를 제어 하는 경우에 사용하며 UMAC 에 ACC-24E2S 를 이용하여 제어 가능하며, 시뮬레이션 테스트를 위하여 설정가능합니다. 단, PMAC 에서는 Pulse and Direction (1 pulse 방식) 만 지원하므로 드라이버 선정 시 유의하여 지원여부를 확인하여야 합니다.

1. PMAC 과 상관없이 Servo Driver 에서 위치제어, 속도제어, 전류제어를 실행합니다.
2. 펄스신호의 수는 위치를 나타내고 펄스신호의 주파수는 속도를 나타냅니다.
3. I 변수 설정

- lxx00 : Motor xx Activation Control (xx = 모터 번호)

모터를 구동하기 위해서는 lxx00 변수를 1 로 설정해서 모터를 Active 시켜야

합니다. 모터 1 의 경우 초기값이 1 로 설정되어 있지만 다른 모터들은 초기값이 0 으로 설정되어 있습니다.

- lxx19 : 최대 허용 조그 가속도 (Maximum Permitted Jog Acceleration)
(units : cts/msec²)
모터의 스펙에 맞는 최대로 허용 가능한 가속도 값보다 낮게 설정합니다.
모터 스펙 이상의 값을 설정할 경우 모터가 고장날 수 있습니다.
- lxx20 : 가속시간 (Acceleration Time)
(units: msec; integer)
S-curve 시간 없이 가속시간만 있을 경우에는 가속도 변화에 있어서 불연속점이 생겨서 모터가 불안정해질 수 있습니다. 시스템에 맞게 적절한 가속시간과 S-curve 시간을 설정해야 합니다.
- lxx21 : “ S ” 커브 시간 (S-Curve Time)
(units: msec; integer)
lxx21 * 2 값이 lxx20 보다 큰 경우 lxx20 의 값을 무시하고 lxx21 * 2 값이 전체 가속도 시간이 되며, 이 때를 full S라고 합니다. 이 경우에 S-Curve 중간부분에서 최대 가속도가 발생하고, 토크가 약한 모터의 경우 문제가 발생할 수 있습니다.
- lxx22 : 조그 속도(Motor xx Jog Speed)
(units : counts / msec)
모터의 구동 속도를 결정합니다. 튜닝전에는 저속(lxx22=1)으로 설정하고 조금씩 증가시키면서 모터를 구동합니다.
- l7mn6 : Servo IC m channel n Output Mode Select
 - l7mn6 = 0 : output A, B, C 를 PWM 으로 사용합니다.
 - l7mn6 = 1 : output A, B 는 DAC 로 사용하고 output C 는 PWM 으로 설정합니다.
 - l7mn6 = 2 : output A, B 는 PWM 으로 사용하고 output C 는 PFM 으로 설정합니다.
 - l7mn6 = 3 : output A, B 는 DAC 로 사용하고 output C 는 PFM 으로 설정합니다.
 - Stepping Motor Control 에서는 PFM 으로 사용하기 때문에 2 또는 3 으로 설정하고 lxx02 변수 설정을 해서 output 을 C 로 변경합니다.
- l7mn0 : Servo IC m Channel n Encoder/Timer Decoder Control
Internal Pulse and Direction 을 사용하고 open loop stepper system 에서 Phantom closed loop 를 생성하는 8 번으로 설정합니다.
- lxx02 : Motor xx Command Output Address
기본값에 +2 를 해서 output A 를 output C 로 변경합니다.
(lxx02=lxx02+2 → Command output setting for stepper)

Note:

lxx00 : 메뉴얼 TURBO SRM 124 페이지

lxx19 ~ lxx22 : 메뉴얼 TURBO SRM 142 ~ 144 페이지

l7mn6 : 메뉴얼 TURBO SRM 259 페이지

l7mn0 : 메뉴얼 TURBO SRM 255 페이지

lxx02 : 메뉴얼 TURBO SRM 125 페이지

4. PID 튜닝

- Stepping Motor Control 의 경우에는 PID 튜닝이 필요없으며 특정 수식에 따른 lxx30 부터 lxx35 까지의 설정이 고정되어 있습니다.

- lxx30 : Motor xx Propotional Gain

Stepping Motor Control 에서는 아래의 식과 같이 l7m03 에 의해 PFM_CLK Divider 가 정해지고 그 값에 따라 PFM_Clock Frequency 이 정해집니다. 정해진 PFM_Clock Frequency 와 lxx08 값에 의해 lxx30 이 정해집니다.

$$lxx30 = 660000 / (lxx08 * PFMClock(MHz))$$

$$PFM_Clock\ Frequency = 39.3216MHz / 2^{(PFM_Clock\ Divider)}$$

$$l7m03 = Encoder_SCLK\ Divider + 8 * PFM_CLK\ Divider + 64 * DAC_CLK\ Divider + 512 * ADC_CLK\ Divider$$

- lxx31 : Motor xx Derivative Gain

Stepping Motor Control 에서는 0 으로 설정합니다.

- lxx32 : Motor xx Velocity Feedforward Gain

Stepping Motor Control 에서는 아래의 식과 같이 Servo Frequency 에 의해 lxx32 가 정해집니다.

$$lxx32 = 6660 * Servo\ Frequency\ (kHz)$$

- lxx33 : Motor xx Integral Gain

Stepping Motor Control 에서는 0 으로 설정합니다.

- lxx34 : Motor xx Integration Mode

Stepping Motor Control 에서는 0 으로 설정합니다.

- lxx35 : Motor xx Acceleration Feedforward Gain

Stepping Motor Control 에서는 0 으로 설정합니다.

*Note:*lxx30 ~ lxx35 : 메뉴얼 Acc-24E2S 27 ~ 28 페이지

5. Stepping Motor Control 변수 설정 예(1 번 모터)

- I100=1 // 1 번 모터 activation
- I7216=3 // Output Mode = A,B(DAC), C(PFM)
- I7210=8 // Encoder/Timer Decoder Control : Internal Pulse and Direction
- I102=* // Command Output Address 초기화(output A : \$78202)
- I102=I102+2 // Command Output Address = output C(\$78204)
- I119=1 // 최대 허용 조그 가속도 = 1counts/msec²
- I120=100 // 가속시간 = 100msec
- I121=20 // S 커브 시간 = 20*2 = 40msec
- I122=32 // 조그 속도 = 32counts/msec
- I130=700 // P gain = 700(Default)
- I131=0 // D gain = 0
- I132=15050 // Vff = 15050(Default)
- I133=0 // I gain = 0
- I134=0 // IM = 0
- I135=0 // Aff = 0

모터 구동 전 확인 및 주의 사항

1. Servo Driver 의 제어모드(속도모드 또는 토크모드) 설정을 합니다.

2. (+,-) Limit 확인을 합니다.

오타로 인한 잘못된 명령이나 정상적이지 않은 설정으로 인해 모터가 갑작스럽게 빠른 속도로 구동될 경우 모터의 움직임을 제어할 수 있는 +방향 Limit 와 - 방향 Limit 가 제대로 동작하는지 확인해야 합니다.

- (+,-) Limit Sensor, Home Sensor 의 배선을 확인합니다.
- Ixx24 변수의 bit17 을 0 으로 설정해서 하드웨어 Limit 를 enable 시킵니다.
Limit 센서는 반드시 Normal Close Type 이어야 합니다.
Limit 센서를 사용하지 않는 경우에는 Ixx24=\$20001 로 설정합니다.
- Ixx15 변수에서 Abort 명령과 Limit 를 만났을때의 감속률을 확인합니다.
(Ixx15 = 5 로 설정해서 감속률을 크게 하여 Abort 명령과 Limit 를 만났을 때 빠른 시간에 정지할 수 있도록 합니다.)
- 가능하다면 손으로 모터를 (+,-) Limit Sensor 까지 이동시키거나 Sensor 에 다른 물체를 접촉해봐서 Sensor 의 동작여부를 확인합니다.

Download Suggested M-variables 기능을 사용한 경우에 Mxx20 (Home Sensor), Mxx21(+ Limit), Mxx22(-Limit) 변수로 확인할 수 있습니다. (xx : 모터 번호)

Note:

lxx24 : 메뉴얼 TURBO SRM 144 페이지

lxx15 : 메뉴얼 TURBO SRM 140 페이지

3. Encoder 방향 확인

Encoder 방향이 모터 방향과 다른 경우에 모터는 제대로 동작하지 않습니다.

- 손으로 모터를 돌리면서 pewin32pro 의 position 창에서 position(cts)값이 변하는지 확인합니다. 시스템에서 +방향으로 이동할 때 position(cts)값이 증가하고 -방향으로 이동할 때 position(cts)값이 감소해야 합니다.
- 손으로 모터를 돌리기 어려운 경우에는 아래에 나와있는 Analog DAC 출력확인 부분을 확인하고 open loop 명령으로 모터를 구동해서 방향을 확인합니다.
- position(cts)값이 반대로 나올 경우 l7mn0 변수의 현재 값을 확인하고 이에 맞는 옵션의 값으로 바꿔줍니다. (0 ↔ 4, 1 ↔ 5, 2 ↔ 6, 3 ↔ 7)

Note:

l7mn0 : 메뉴얼 TURBO SRM 255 페이지

4. Analog DAC 출력확인

open loop 명령으로 모터가 잘 동작하는지 확인하고 0%출력일 때 모터의 동작여부를 확인해서 offset 전압을 보정해줍니다.

- Terminal 창에 #xxo0 명령을 실행해서 모터 출력을 확인합니다.
(#xxo0 명령은 xx 번 모터의 Amp 를 Enable 시키고(open loop 상태) 0%출력으로 구동하는 명령입니다.)

*** 주의사항 ***

- #xxo0 명령을 실행할 때는 출력을 5%(#xxo5)이하로 하는 것이 좋습니다.
10%정도의 출력으로도 모터가 굉장히 빠르게 움직입니다.
- Open loop 명령으로 모터를 구동하는 경우에 모터가 폭주할 수 있기 때문에 Open loop 명령 실행 후 모터를 정지시킬 준비를 해야 합니다.

모터를 정지시킬때는 Ctrl + K(모든 모터 죽이기) 명령을 실행합니다.

- 정상적인 경우 0% 출력이기 때문에 모터가 움직이지 않아야 하지만 offset 전압으로 인해 움직임이 발생할 수 있습니다. 움직임이 발생하는 경우에 모터가 움직이지 않도록 lxx29 변수 값을 조정합니다.
- + 방향으로 움직일 경우 lxx29 값을 -값으로 설정합니다.
- 방향으로 움직일 경우 lxx29 값을 +값으로 설정합니다.
- Analog 출력의 기준전압은 lxx69 로 설정합니다. 기본값은 20480 으로 6.25V 입니다. 최대값인 10V 로 설정하려면 lxx69 를 32767 로 설정합니다.

Note:

lxx29: 메뉴얼 TURBO SRM 153 페이지

lxx69: 메뉴얼 TURBO SRM 169 페이지

5. Encoder 방향 재확인

- Analog DAC 출력 조정을 하면서 Encoder 방향이 다시 바뀌었을 경우에 조정 방법입니다.
- H/W 수정 : 결선수정 → DAC+를 DAC-로 결선 수정(권장)
- S/W 수정 : lxx30(P gain) 값을 -값으로 설정합니다.(비권장)

6. 가능하면 모터와 부하의 연결을 해제한 상태에서 모터를 구동해봅니다.

7. 부하가 연결된 모터를 구동할때는 모터와 부하 주위에 사람이 없게 하고 동작합니다.

8. 모터의 직경, 감속비, 엔코더의 체배율 등을 확인해서 counts 당 이동거리가 얼마나 되는지 확인해서 모터에 부하를 연결했을 때 안전하게 구동할 수 있는 거리가 몇 counts 인지 확인해야 합니다.

예) 모터가 한 바퀴 이동할 때 128 Pulse 발생

감속비 = 19:1, 엔코더 체배율 = 4 체배, 지름 : 11.64mm

한 바퀴 이동 : $128 \times 19 \times 4 = 9728 \text{ counts}$

한 바퀴 이동 : $11.64 \times \text{PI}(3.141592) = 36.5681 \text{ mm}$

1mm 이동을 위한 counts 수 : 266.02393 counts

안전하게 구동할 수 있는 거리가 10cm(100mm)정도이면 26600 counts 의 범위 안에서 모터를 구동해야 합니다.

JOG 명령

PEWIN 32Pro 의 Terminal 창에서 모터를 즉시 구동시키는 온라인 명령어입니다.

1. #xxJ+ : xx 모터를 + 방향으로 이동합니다.
2. #xxJ- : xx 모터를 - 방향으로 이동합니다.
3. #xxJ/ : xx 모터를 정지합니다..
4. #xxK : xx 모터를 정지시키고 Open Loop mode 로 합니다.
(이 경우에 모터를 손으로 돌릴 수 있습니다.)
5. #xxJ = 숫자 : xx 모터를 숫자위치로 이동합니다.
예) #1J = 2000 : 1 번 모터를 2000cts 위치로 이동합니다.
#1J = -2000 : 1 번 모터를 -2000cts 위치로 이동합니다.
6. #xxJ : 숫자 : xx 모터를 현재 위치에서 숫자만큼 이동합니다.
예) #1J:2000 : 1 번 모터를 2000cts 만큼 이동합니다.
현재 위치가 2000cts 인 경우 4000cts 로 이동합니다.
#1J:-2000 : 1 번 모터를 -2000cts 만큼 이동합니다.
현재 위치가 2000cts 인 경우 0cts 로 이동합니다.

Note:

Jogging commands: 메뉴얼 TURBO SRM 77 페이지

모터 구동

가속 시간은 시스템에 맞게 충분히 설정하고 조그 속도는 저속으로 설정하고 모터를 구동 해서 Following Error 를 확인합니다. Following Error 가 큰 경우에는 PID Gain 튜닝을 다시 해서 Following Error 를 감소시킵니다. Following Error 가 감소하면 조그속도를 증가시키면서 모터와 부하의 상태를 확인합니다. 속도가 증가함에 따라서 Following Error 가 커지거나 모터에서 소음, 진동이 발생하는 경우에 PID Gain 튜닝을 다시 합니다.

Homing Motors(Motor 의 Home 위치 이동)

Motor 가 Home(원점)을 찾는 작업 및 프로그램을 말합니다.

Motor 를 구동하여 작업을 할 때, 작업을 완료하고 전원을 off 한 후 다음 날 전원을 on 했을 때 Motor 는 Home(원점)으로 이동해서 작업을 시작해야 그 전에 하던 작업과 똑같은 결과를 발생할 수 있습니다. 만약 전원을 ON 시키고 Home 을 찾지 않고 그 자리에서 작업을 바로 실행한다면 시작 지점이 어제와 다르기 때문에 어제와 같은 결과를 발생할 수 없습니다.

Home 의 위치 결정 방법(I7mn3)

1. I7mn3=0 : Home 의 위치를 Home Flag 로 결정합니다.
2. I7mn3=1 : Home 의 위치를 Positive End Limit Flag 로 결정합니다.
(Ixx24 의 bit17 을 1 로 수정합니다. Ixx24=\$20001)
3. I7mn3=1 : Home 의 위치를 Negative End Limit Flag 로 결정합니다.
(Ixx24 의 bit17 을 1 로 수정합니다. Ixx24=\$20001)
4. I7mn3=3 : Home 의 위치를 User Flag 로 결정합니다.

Note:

I7mn3 : 메뉴얼 TURBO SRM 257 페이지

Ixx24 : 메뉴얼 TURBO SRM 144 페이지

Home Search 방법

1. Home Sensor, (+,-) Limit Sensor 를 사용하는 경우
 - 전원을 ON 했을 때 모터의 위치를 기준으로 Home 의 위치가 +방향에 있을 수도 있고 -방향에도 있을 수 있기 때문에 먼저 Motor 를 -방향으로 이동시켜서 Negative End Limit 까지 이동시킵니다.
 - Negative End Limit 에 도착한 후 Home 명령(#xxHM)으로 모터를 +방향으로 이동 시키면서 Home 을 찾습니다.
2. Home Sensor 없이 (+,-) Limit Sensor 만 사용하는 경우
 - (+) Limit Sensor 와 (-) Limit Sensor 중에서 Home Sensor 로 사용할 것을 정합니다.
 - 시스템의 전원을 ON 했을 때 모터가 Home Sensor 로 정한 Limit Sensor 에 있을 수 있기 때문에 모터가 Home Sensor 로 정한 Limit Sensor 를 만나고 지나가는 시점을 Home 으로 결정합니다.
3. 시스템에 따라서 Home 을 찾는 방법은 변경 될 수 있습니다.

Homing 관련 I variables 와 명령어

1. Ixx19 : Motor xx Maximum Jog/Home Acceleration(counts/msec²)
Jog 명령(#xxJ-)과 Home 명령(#xxHM)으로 이동할 때 가속도를 결정합니다.
2. Ixx20 : Motor xx jog/Home Acceleration Time(msec)
Jog 명령(#xxJ-)과 Home 명령(#xxHM)으로 이동할 때 가속도 시간을 결정합니다.

3. lxx21 : Motor xx Jog/Home S-curve Time(msec)

Jog 명령(#xxJ-)과 Home 명령(#xxHM)으로 이동할 때 S-curve 시간을 결정합니다.

★ 주의사항 ★

lxx20 과 lxx21 은 둘 다 0 으로 설정하면 안 됩니다.

4. lxx22 : Motor xx Jog Speed(counts / msec)

Jog 명령(#xxJ-)으로 이동할 때 속도를 결정합니다.

5. lxx23 : Motor xx Home Speed and Direction(counts/msec)

- Home 명령(#xxHM)으로 이동할 때 속도와 방향을 결정합니다.
- 부호가 -인 경우에 -방향으로 이동하고 +인 경우에 +방향으로 이동합니다.

6. lxx24 : Motor xx Flag Mode Control

Positive End limit Flag 와 Negative End Limit Flag 로 Home 의 위치를 결정할 때 lxx24 의 Bit17 을 1 로 설정해서 Hardware Limit 를 Disable 합니다.

7. lxx26 : Motor xx Home Offset

실제 원점의 위치에서 일정 거리만큼 떨어진 곳을 원점으로 잡고 싶을 때 사용합니다. 원하는 거리 값을 lxx26 에 넣습니다. (단위 : 1/16 counts)

8. l7mn2 : Servo IC m Channel n Capture Control

위치 Capture 방법을 Flag 로 할지 Index(Encoder 의 C 상)로 할지 또는 두 개를 같이 사용해서 할지 결정합니다.

예) l7mn2=2 : Flag 가 High 가 되었을 때 capture 합니다.

l7mn2=10 : Flag 가 Low 가 되었을 때 capture 합니다.

l7mn2=1 : Index(Encoder 의 C 상)이 high 가 되었을 때 capture 합니다.

l7mn2=5 : Index(Encoder 의 C 상)이 Low 가 되었을 때 capture 합니다.

9. 온라인 명령어 : HM

- #xxHM : xx 번 모터가 Home 을 찾는 이동을 시작하는 명령입니다.

Note:

lxx19 ~ lxx24 : 메뉴얼 TURBO SRM 142~144 페이지

lxx26 : 메뉴얼 TURBO SRM 150 페이지

l7mn2 : 메뉴얼 TURBO SRM 256 페이지

HM(On-Line Command) : 메뉴얼 TURBO SRM 353 페이지

Homing 관련 Motor 2 의 M-variables

1. M220 : HMFL2 flag input status

motor2 의 Home Flag 가 High 가 되었을 때 1 이 되고, Low 일 때 0 이 됩니다.

2. M221 : PLIM2 flag input status

motor2 의 Positive end limit 를 만나거나 Positive end limit flag 가 high 가 되면 1 이 됩니다. 1 이 되면 motor status 창에서 Positive end limit set(soft of hard) 부분에 불이 들어옵니다.

3. M222 : MLIM2 flag input status

motor2 의 Negative end limit 를 만나거나 Negative end limit flag 가 high 가 되면 1 이 됩니다. 1 이 되면 Motor Status 창에서 Negative end limit set(soft of hard) 부분에 불이 들어옵니다.

4. M245 : #2 Home complete bit

Home 명령을 통해 Home 을 찾게 되면 값이 1 이 되고, Motor Status 창에서 Home Complete 부분에 불이 들어옵니다.

5. M240 : #2 Background in-position bit

Home 을 찾으면 값이 1 이 되고 Motor Status 창에서 In-position true 부분에 불이 들어옵니다.

6. M233 : #2 Desired-velocity-zero bit

Home 을 찾으면 값이 1 이 되고 Motor Status 창에서 Desired velocity 0 부분에 불이 들어옵니다.

7. M238 : #2 Open-loop-mode bit

모터 2 가 Open Loop 가 되었을 때(#2K 실행) 1 이 됩니다.

Note:

M220, M221, M222, M233, M238, M240, M245 : 메뉴얼 TURBO SRM 699, 700 페이지

Homing Motors 예제(Home Sensor, (+,-) Limit Sensor 를 사용하는 경우)

close

end gat

del gat

open plc 15 clear ; buffer open

P1237=0 ; 에러코드의 발생여부를 확인할 수 있는 모니터링용 변수

M245=0 ; #2 Home complete
 i220=300 ; Acceleration Time
 i221=50 ; S-Curve Time
 l222=20 ; Jog Speed
 l223=20 ; Home Speed and direction

COMMAND"#2j-" ; motor 2 를 Negative 방향으로 이동시켜서 Negative Limit 에
 ; 도달하도록 합니다.

I5311=500*8388608/I10 ; 500msec 동안 지연

While (I5311 > 0)

EndWhile

I5311=(50000)*8388608/I10 ; 50000msec(50 초) 동안 Timeout 설정
 ; Negative limit, in position, desired vel. zero 확인

While (M232 = 0OrM240 = 0OrM233 = 0)

AND(m238=0) ; open loop mode 확인

AND(I5311>0) ; timeout 확인

EndWhile

/*

m230, m240, m233, m238 을 확인하면서 50 초의 시간제한을 둡니다. Negative Limit End Flag 가 High 가 되면(Negative Limit 에 도달하면) m232, m240, m233 이 모두 1 로 변하면서 while 문을 빠져나옵니다.

50 초 내에 Negative Limit End Flag 가 High 로 바뀌지 않을 경우(Negative Limit 에 도달하지 못한 경우)에는 아래와 같은 에러 코드가 실행됩니다.

*/

If (M238 = 1OrI5311 != 0) ; open loop mode 확인 , Timeout 확인하여 에러코드 실행

P1237=1 ; 모니터링 변수, 에러코드가 발생하면 P1237 변수가 0 에서 1 로 바뀌는

; 것을 통해 에러코드 발생여부를 확인할 수 있습니다.

COMMAND"#2k" ; #2 Kill, 모터를 죽여서 구동을 멈춥니다.

I5311=100*8388608/I10 ; 100msec 동안 지연

While (I5311 > 0)

EndWhile

```

DISABLEPLC15          ; disable plc 15, plc15 프로그램 종료
RETURN                ; 바로 프로그램의 끝인 close 로 가서 종료합니다.
EndIf

I5311=(1000)*8388608/I10      ; 1000msec 동안 지연
While (I5311 > 0)
EndWhile

COMMAND"#2hm"           ; #2 home start, Home 찾기 시작
I5311=500*8388608/I10      ; 500msec 동안 지연
While (I5311 > 0)
EndWhile

I5311=(50000)*8388608/I10    ; 50000msec(50 초) 동안 Timeout 설정
While (M245 = 0OrM240 = 0OrM233 = 0)
                                ; home complete, in position, Desired Vel. Zero 확인
    AND(m238=0)                ; open loop mode 확인
    AND(I5311>0)               ; timeout 확인
EndWhile
/*
m245, m240, m233, m238 을 확인하면서 50 초의 시간제한을 둔다. Home Flag 가
High 가 되면(Home 을 찾았을 때) m245, m240, m233 이 모두 1 로 변하면서 while 문을
빠져나옵니다.
50 초 내에 Negative Limit End Flag 가 High 로 바뀌지 않을 경우(Negative Limit 에
도달하지 못한 경우)에는 아래와 같은 에러 코드가 실행됩니다.
*/

If (M238 = 1OrI5311 !=> 0)      ; open loop mode, Timeout 확인
    P1237=2      ; 모니터링 변수, 에러코드가 발생하면 P1237 변수가 0 에서 2 로 바뀌는
                  ; 것을 통해 에러코드 발생여부를 확인할 수 있습니다.
    COMMAND"#2k"          ; #2 Kill
    I5311=100*8388608/I10   ; 100msec 동안 지연
    While (I5311 > 0)
    EndWhile

```

```
DISABLEPLC15          ; disable plc15
RETURN
EndIf

I5311=(1000)*8388608/I10      ; 1000msec 동안 지연
While (I5311 > 0)
EndWhile

dis plc15              ; disable plc15
close                 ; buffer 닫음
```

Data Gathering (자료 수집)

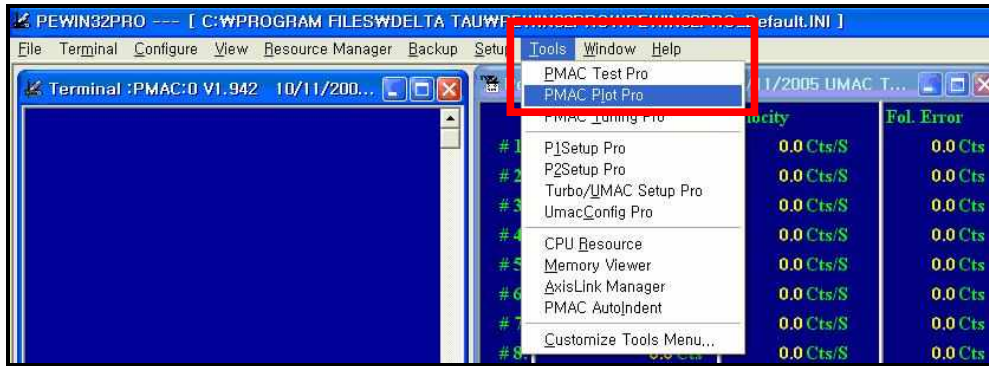
서보루프 튜닝, 프로그램 디버깅 등을 위해서 모터의 속도, 가속도, 위치 등의 값을 실시간으로 수집해서 PMAC Plot Pro라는 프로그램을 통해서 그래프로 나타냅니다. 여기에서는 간단하게 사용할 수 있는 Quick Plot에 대해서만 설명합니다.

Data Gathering 관련 명령어

1. DEFINE GATHER (DEF GAT) : gathering준비. 현재 남은 메모리 영역을 전부 gathering 영역으로 설정합니다.
2. GATHER (GAT) : gathering을 시작합니다.
3. END GATHER (ENDG) : gathering을 종료합니다.
4. DELETE GATHER (DEL GAT) : gathering buffer 삭제. 메모리 영역을 복귀합니다.

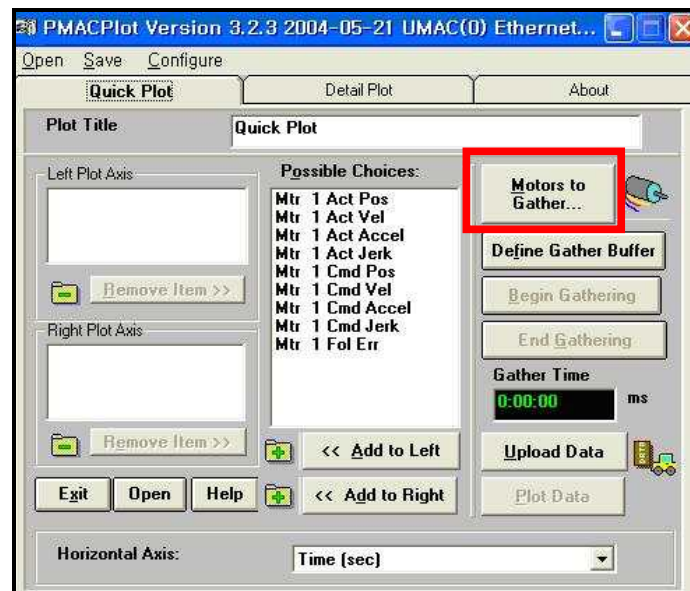
간단한 Data Gathering 절차 예제

1. 자료 수집 내용 : 1 번 모터를 이동시키면서 그 운동에 대한 정보를 수집합니다.
2. 자료 수집 절차
 - Tool 메뉴의 PMAC Plot Pro 를 클릭합니다.



< PMAC Plot Pro 실행 위치>

- Quick Plot 창에서 Motors to Gather... 클릭합니다.

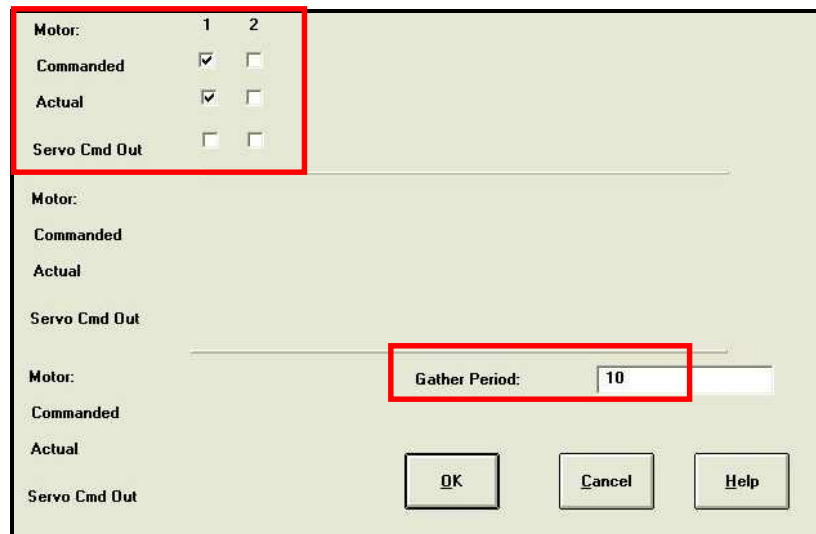


< Motor to Gather... 위치 >

- 자료수집을 하고 싶은 내용을 정합니다. (lxx00=1 로 설정된 모터만 표시됩니다.)
 - Commanded : 명령 값을 확인합니다.
 - Actual : 명령에 의해 실제 발생한 결과값(Motor Feedback 값)을 확인합니다.
 - Servo Cmd Out : Analog $\pm 10V$ 에 대한 출력값을 확인합니다.
-32767(-10V) ~ 32768(10V)의 값으로 표시됩니다.
 - Gather Period : 자료수집 주기를 정합니다.

예) Gater Period = 1 : servo update 시간(442 μ sec)마다 자료 수집을 합니다.

Gater Period = 5 : servo update * 5 시간마다 자료 수집을 합니다.



< 자료 수집 내용 및 Gather Period 설정 >

- 명령어를 실행합니다.

del gat // 이전에 사용하고 있을지도 모를 자료수집 buffer 를 삭제합니다.

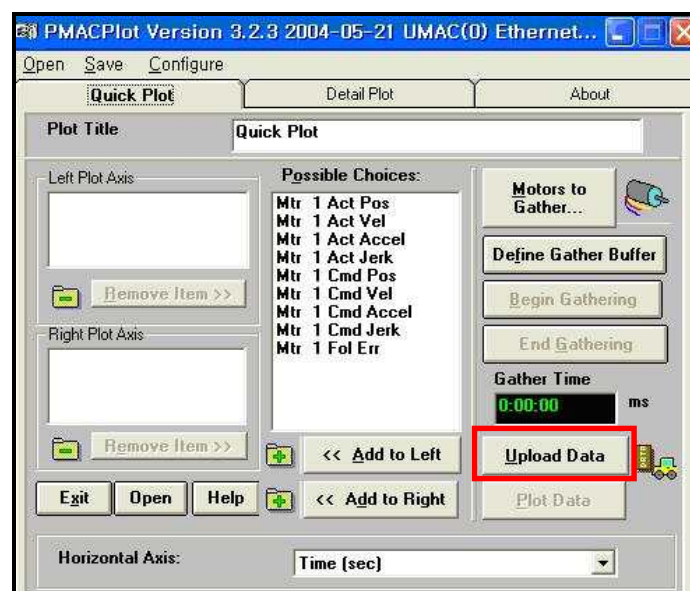
def gat // 자료수집 준비를 합니다.

gat #1j+ // 동작시작과 동시에 자료수집을 하기 위해서 한 블록에
// 자료수집 실행 명령(gat)과 동작실행 명령(#1j+)을 씁니다.

#1j/ // 동작을 정지시킵니다.

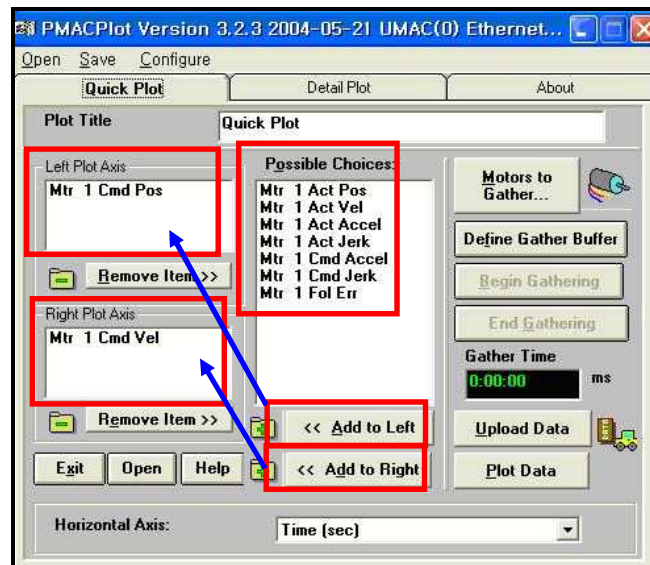
end g // 자료수집을 종료합니다.

- PMAC Plot 창의 Quick Plot 창에서 Upload Data 클릭합니다.



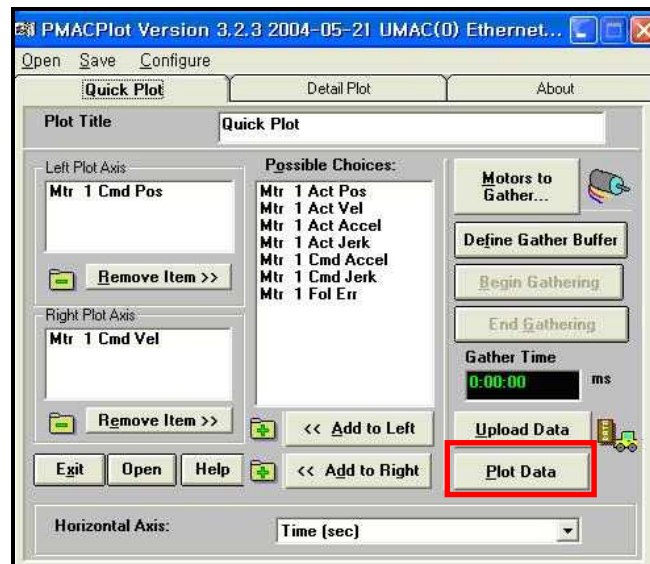
< Upload Data 위치 >

- Possible Choices 부분에서 확인하고 싶은 정보를 클릭해서
Add to Left 나 Add to Right 버튼을 클릭해서 원하는 Plot Axis 로 이동시킵니다.



< Possible Choice 및 Plot Axis 이동 >

- Plot Data 를 클릭해서 수집된 자료의 그래프를 확인합니다.



< Plot Data 위치 >

PLC Program 의 작성과 실행

하드웨어 PLC와 같은 량의 많은 작업을 수행하고 모션 프로그램에 관계없이 반복적이고 빠른 동작을 합니다.

- 용도 : 입력모니터 (monitoring inputs)
- 출력모니터 (setting outputs)
- 각종 게인 값 변경 (changing gains)

PMAC 상태감시 (monitoring card status)

명령실행 (commanding actions, cmd “ ”)

메세지 전송 “ sending messages”

1. Terminal 창에 i5=3 실행(background PLCs on) → 메뉴얼 Turbo SRM 87 페이지
2. File 메뉴에서 New File 클릭 → Editor Pro 창이 열립니다.
3. close 작성 : 열려 있을 수 있는 다른 프로그램의 buffer 를 닫습니다. 다른 buffer 가 열려있을 경우에 현재 프로그램이 오작동 할 수 있습니다.
4. end gat 작성 : 만약 gathering(자료수집)을 하고 있을 경우에 종료시킵니다.
5. del gat 작성 : 종료시킨 gathering(자료수집) buffer 를 삭제해서 메모리를 user 가 사용할 수 있도록 합니다.
- 6.OPEN PLCn CLEAR 작성
→ n 번 plc 프로그램을 작성하기 위해 buffer 를 열고 그 buffer 의 내용을 모두 삭제하는 작업입니다. (n : 0~31)
7. 원하는 내용 작성
8. 마지막 줄에 close 작성 → buffer 를 닫습니다.
9. 아래 그림의 아이콘을 누르거나 Ctrl+s 를 눌러서 작성한 프로그램을 저장할 수 있습니다.



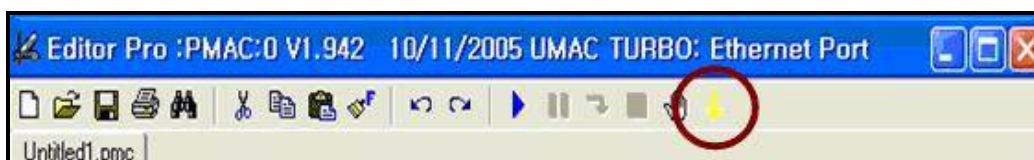
< 저장 아이콘 >

10. 저장한 파일은 원하는 폴더에 보관할 수 있고 아래 그림의 아이콘을 눌러서 열어 볼 수 있습니다.



< 열기 아이콘 >

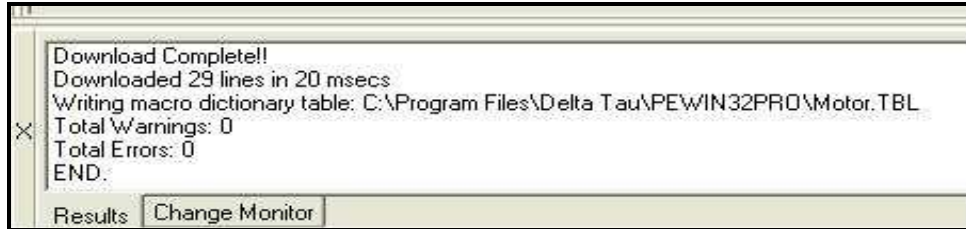
11. 작성한 파일을 실행하기 위해서 아래그림의 아이콘을 눌러서 Download 합니다.
프로그램을 저장하지 않아도 Download 하여 실행시킬 수 있습니다.



< 다운로드 아이콘 >

12. Download 를 완료하면 아래와 같은 창이 Pewin32Pro 의 하단부에 나타납니다.

Error 가 있는지 확인하고 있을 경우에 작성한 프로그램을 수정합니다.



< Error 확인 창 >

13. Error 없이 Download 를 완료하면 Terminal 창에 ENA PLCn 을 실행합니다.

여기서 n 은 작성한 PLC 프로그램의 번호입니다.

14. 작성한 PLC 프로그램이 실행됩니다.

Motion Program 의 작성과 실행

모션 프로그램은 위치결정 및 윤곽이송을 위해 모터를 제어하고 조정하는 역할을 합니다.

특징 : 한번에 한 블록씩 실행하고, 모든 계산은 이송이전에 수행됩니다.

좌표계에서 실행됩니다.

한 프로그램은 복수개의 좌표계에서 동시에 실행될 수 있습니다.

한 개의 좌표계에서는 한 개의 프로그램만이 실행가능합니다.

실행 전에 같은 좌표계에 있는 모든 모터를 servo on(amp enable) 시켜야 합니다.

1. File 메뉴에서 New File 클릭 → Editor Pro 창이 열립니다.
2. close 작성 : 열려 있을 수 있는 다른 프로그램의 buffer 를 닫습니다. 다른 buffer 가 열려있을 경우에 현재 프로그램이 오작동 할 수 있습니다.
3. end gat 작성 : 만약 gathering(자료수집)을 하고 있을 경우에 종료시킵니다.
4. del gat 작성 : 종료시킨 gathering(자료수집) buffer 를 삭제해서 메모리를 user 가 사용할 수 있도록 합니다.
5. &n 작성 : Motion 프로그램은 좌표계에서만 실행되기 때문에 motion 프로그램을 실행할 좌표계를 설정합니다. (n:좌표계 번호(1~16))
6. #xx->X 작성 : 좌표계에 할당될 모터를 설정하고 모터를 어떤 축으로 사용할지 결정합니다. (xx:모터번호, X:축이름(X,Y,Z,U,V,W,A,B,C))

7. OPEN PROGgn CLEAR 작성

→ n 번 Motion 프로그램을 작성하기 위해 buffer 를 열고 그 buffer 의 내용을 모두 삭제하는 작업입니다.

8. 원하는 내용 작성

9. 마지막 줄에 close 작성 → buffer 를 닫습니다.

10. 아래 그림의 아이콘을 눌러서 작성한 프로그램을 저장할 수 있습니다.



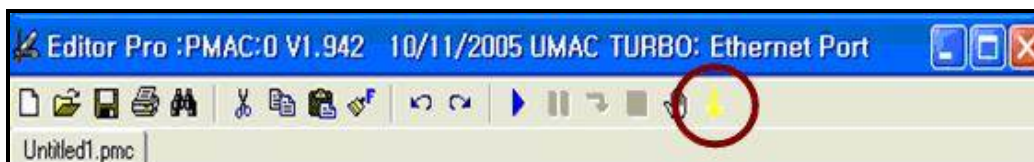
< 저장 아이콘 >

11. 저장한 파일은 원하는 폴더에 보관할 수 있고 아래 그림의 아이콘을 눌러서 열어 볼 수 있습니다.



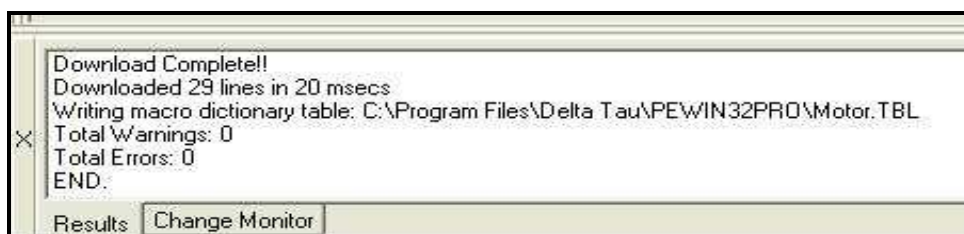
< 열기 아이콘 >

12. 작성한 파일을 실행하기 위해서 아래그림의 아이콘을 눌러서 Download 합니다.
프로그램을 저장하지 않아도 Download 하여 실행시킬 수 있습니다.



< 다운로드 아이콘 >

13. Download 를 완료하면 아래와 같은 창이 Pwin32Pro 의 하단부에 나타납니다.
Error 가 있는지 확인하고 있을 경우에 작성한 프로그램을 수정합니다.



< Error 확인 창 >

14. Error 없이 Download 를 완료하면 Terminal 창에 #xxJ/를 실행하여
Servo On(amplifier enabled)시킵니다.

여기서 xx 는 작성한 프로그램에서 수행되는 모터번호이고 모터가 한 개가 아닌

경우에는 모든 모터를 Servo On(amplifier enabled)합니다.

15. &n Bn R 을 Terminal 창에 실행합니다.

&n : n 번 좌표계라는 뜻으로 어떤 좌표계에서 프로그램을 실행할지 결정합니다.

Bn : 실행할 프로그램 번호를 결정합니다.

R : 실행 명령입니다.

16. 작성한 Motion 프로그램이 실행됩니다.

비상정지 절차

프로그램을 통해 모터를 구동중에 비상 상황이 발생하여 급히 정지시켜야 되는 경우 다음과 같은 절차로 모터를 정지시킵니다.

온라인 명령어 A 실행 → 모터가 정지한 것을 확인 → 온라인 명령어 K 실행

A : 모터를 정지시키고 amp enable 상태입니다.

K : 모터를 정지시키고 amp disable 상태입니다.

- 비상정지 예제 PLC 프로그램

CLOSE // 현재 열려있는 Buffer 를 닫습니다.

END GATHER // 자료수집을 하고 있다면 종료시킵니다.

DELETE GATHER // 자료수집을 했던 메모리를 삭제합니다.

// p7000 : Emergency stop input latch

// m7000 : Emergency stop input

// m7024 : main contact for main AC for BUS Voltage

// i5111 : count down timer

OPEN PLC 5 CLEAR // 5 번 PLC 프로그램의 Buffer 를 열고 내용을 삭제합니다.

If (M7000 = 1 And P7000 = 0) // Latch 가 0 이고 비상정지 스위치가 눌러졌을 때

COMMAND^A // All Motor Abort(모든 모터 정지 / Amp enable 상태)

I5111=500*8388608/I10 // 감속을 위한 500msec 시간지연

While (I5111 > 0)

EndWhile

COMMAND^K // All Motor Kill(모든 모터 정지 / Amp disable 상태)

M7024=0 // main contact for main AC for BUS Voltage 차단

```
P7000=1    // latch 를 1 로 설정해서 명령이 계속 재입력 되는 것을 방지
EndIf
If (M7000 = 0 And P7000 = 1)    // 비상정지가 눌러진 후
    M7024=1                      // main contact for main AC for BUS Voltage 연결
    I5111=5000*8388608/I110      // 5000msec 시간 지연
    While (I5111 > 0)
    EndWhile
    COMMAND^A    // 모든 모터를 Amp Enable 상태로 만듭니다.
    P7000=0       // latch 를 0 으로 설정해서 명령이 계속 재입력 되는 것을 방지
EndIf

CLOSE    // buffer 닫음. 프로그램 종료.
```