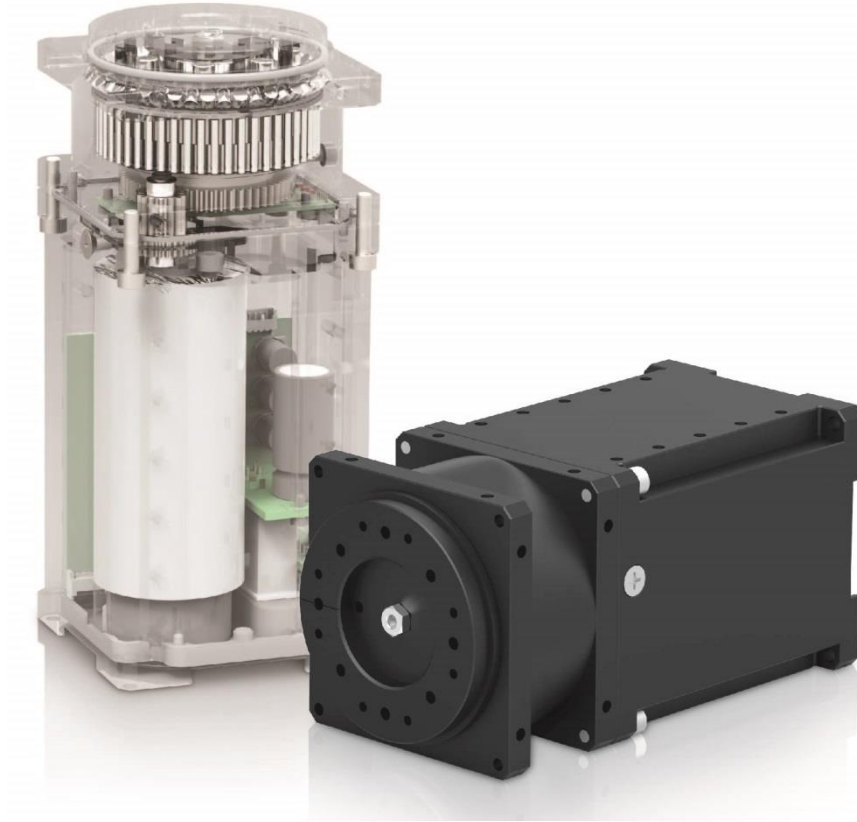


Quick Start for Dynamixel Pro



ROBOTIS

CONTENTS

CONTENTS.....	2
1 Tutorial.....	5
1.1 Roboplus.....	5
1.1.1 Preparation.....	5
i. Roboplus 설치.....	5
ii. Wiring 54 Series.....	5
iii. Wiring 42 Series.....	6
iv. Wiring 54 Series & 42 Series.....	6
v. USB2Dynamixel (RS485 설정).....	7
vi. COM Port Latency Time 설정.....	7
1.1.2 Dynamixel Wizard.....	9
i. Dynamixel Pro를 동작해봅시다.....	9
ii. Dynamixel Pro를 Wheel Mode로 바꾼 뒤 동작시켜 봅시다.....	15
iii. Dynamixel Pro의 LED를 On/Off 해 봅시다.....	18
iv. ID를 3으로 변경해 봅시다.....	19
v. 통신속도를 변경하여 더 빠르게 Dynamixel Pro를 이용해 봅시다.....	21
vi. Joint Mode로 변경한 후 가속도를 설정하고 움직여 봅시다.....	22
vii. 작동 범위를 줄인 후 이를 확인해 봅시다.....	24
viii. 작동 범위를 늘린 후 이를 확인해 봅시다.....	26
ix. Wheel Mode로 변경한 후 가속도를 설정하고 움직여 봅시다.....	28
x. Torque Mode로 변경한 후 동작해 봅시다.....	30
xi. Dynamixel Pro의 Firmware Update를 수행해 봅시다.....	32
xii. Dynamixel Wizard를 통한 Dynamixel Pro의 Firmware를 복구 해 봅시다.....	35
1.2 PC(VS2010).....	40
1.2.1 Preparation.....	40
i. Dynamixel Pro를 C언어로 제어하기 위한 설정을 해봅시다.....	40
ii. C언어로 USB2Dynamixel 과의 연결 및 연결종료를 해 봅시다.....	43
1.2.2 기본적인 Dynamixel Pro 의 기능들을 사용해 봅시다.....	45
i. Torque를 On/Off해 봅시다.....	45
ii. 이제 C언어로 Dynamixel Pro를 작동해 봅시다.....	47

- iii. C언어로 ID가 1인 Dynamixel Pro의 ID를 3으로 변경해 봅시다..... 49
- iv. Dynamixel Pro의 통신속도를 변경해 봅시다..... 52
- v. Dynamixel Pro의 LED를 On/Off해 봅시다..... 55
- vi. Position P gain 값을 변경한 후의 움직임을 확인해 봅시다..... 58
- vii. 천천히 혹은 빠르게 움직이도록 해봅시다..... 61
- viii. Dynamixel Pro의 내부 온도를 확인해 봅시다..... 65
- x. 바퀴 모드로 변경한 후 속도를 변경해봅시다..... 69
- xi. 현재 위치와 현재 속도를 한번에 확인 해 봅시다..... 72
- 1.2.3 Dynamixel Pro 에서 추가된 기능들을 C 언어를 이용하여 사용해 봅시다..... 76
 - i. Dynamixel Pro의 0점을 바꾼 후 이를 확인해 봅시다..... 76
 - ii. Dynamixel Pro의 작동 범위를 줄인 후 이를 확인해 봅시다..... 79
 - iii. Dynamixel Pro의 작동 범위를 늘린 후 이를 확인해 봅시다..... 83
- 1.2.4 Dynamixel Pro 의 Indirect Addressing 기능을 사용해 봅시다..... 86
 - i. Dynamixel PRO의 위치, 속도, 가속도를 한번에 변경해 봅시다..... 86
 - ii. Dynamixel PRO의 온도와 현재 위치 값을 한번에 읽어 봅시다..... 92
- 1.2.5 여러 개의 Dynamixel Pro 를 사용해 봅시다..... 96
 - i. 3개의 Dynamixel Pro의 LED를 On/Off해 봅시다..... 96
 - ii. 3개의 Dynamixel Pro의 목표 위치 바꾸어 봅시다..... 98
 - iii. 3개의 Dynamixel Pro의 현재 위치 값을 확인해 봅시다..... 101
 - iv. 3개의 Dynamixel Pro에서 첫 번째 Dynamixel Pro는 온도를, 두 번째 Dynamixel Pro는 위치 를, 세 번째는 Dynamixel Pro는 현재 토크를 확인해 봅시다..... 104
- 2 Reference 106
 - 2.1 모델별 주요 초기값 106
 - 2.1.1 H Series 106
 - i. H54-200-S500-R..... 106
 - ii. H54-100-S500-R..... 106
 - iii. H42-20-S300-R 106
 - 2.1.2 M Series 107
 - i. M54-60-S250-R 107
 - ii. M54-40-S250-R 107
 - 2.1.3 L Series 107
 - i. L54-50-S290-R 107

- ii. L54-30-S400-R108
- 2.2 Control Table of Dynamixel Pro..... 109
- 2.3 각 부 명칭 111
 - 2.3.1 H54, M54, L54 시리즈 111
 - 2.3.2 H42, L42 시리즈..... 111
- 2.4 Dimension 112
 - 2.4.1 H Series 112
 - i. H54-200-S500-R.....112
 - ii. H54-100-S500-R.....112
 - iii. H42-20-S300-R113
 - 2.4.2 M Series 114
 - i. M54-60-S250-R114
 - ii. M54-40-S250-R114
 - 2.4.3 L Series 115
 - i. L54-50-S290-R115
 - ii. L54-30-S400-R115
 - iii. L42-10-S300-R116
- 2.5 형식 표기 방식 117

1 Tutorial

본 매뉴얼은 Dynamixel Pro 를 처음 사용하는 사용자를 위해 작성되었으며, C 와 C++에 어느 정도 익숙한 사용자를 위한 내용이 포함되어 있습니다.

1.1 Roboplus

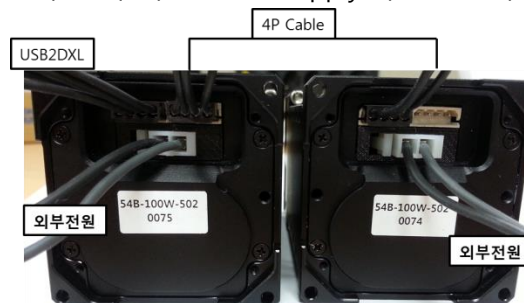
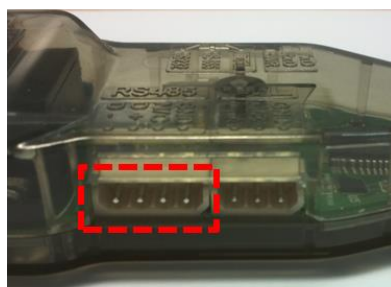
1.1.1 Preparation

i. Roboplus 설치

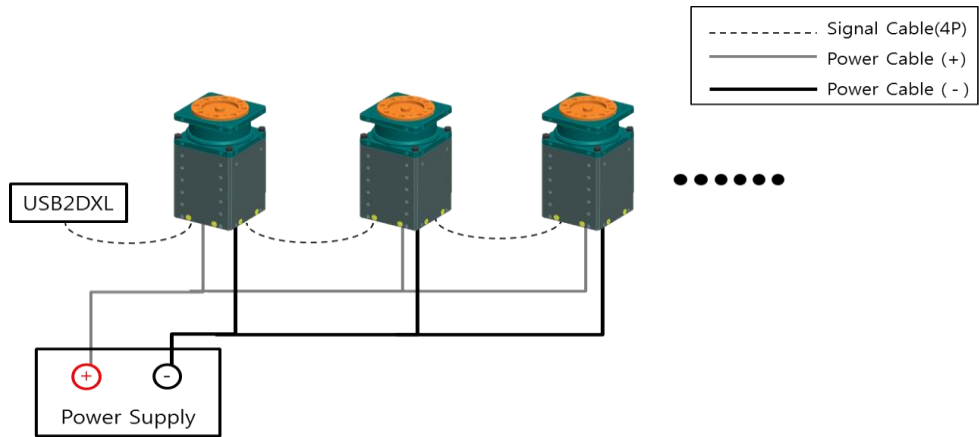
- Roboplus 는 로보티즈의 모든 제품들을 사용 및 프로그래밍 할 수 있는 통합 소프트웨어 입니다.
- 본 챕터에서는 Roboplus 에 포함되어 있는 Dynamixel Wizard 를 사용하여 Dynamixel Pro 를 테스트 하는 내용으로 작성되어 있습니다.
- Roboplus 는 <http://www.robotis.com/xe/download> 에서 최신버전을 다운받아 설치 합니다.

ii. Wiring 54 Series

- Dynamixel Pro 를 사용하기 위해서는 USB2Dynamixel 과 24V 이상의 전원을 공급할 수 있는 PowerSupply 나 SMPS 가 필요합니다.
- Dynamixel Pro 54 시리즈는 외부전원 케이블과 4P 케이블 모두로부터 전원을 공급받을 수 있으나, 안정적인 사용을 위해서 외부전원 케이블로 전원을 공급하는 것을 추천합니다.
- 아래의 사진과 같이 USB2 Dynamixel 의 오른쪽에 4P 케이블을 연결하고 4P 케이블의 다른 쪽을 Dynamixel Pro 에 연결합니다.
- PowerCable 도 Dynamixel Pro 에 꽂은 뒤 반대 쪽을 PowerSupply 에 연결합니다.



- 개략도는 아래와 같습니다.



- 위의 그림과 같이 데이터 체인으로 4P 케이블을 연결하고 전원은 각 Dynamixel Pro 당 하나씩 연결하면 됩니다.

iii. Wiring 42 Series

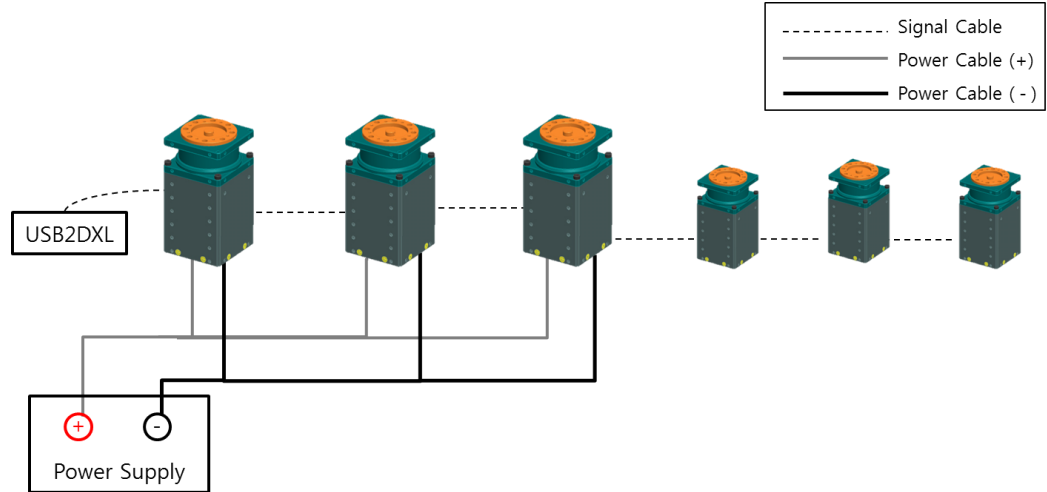
- Dynamixel Pro 42 시리즈는 4P 케이블을 통하여 전원을 공급받습니다.
- 54 시리즈와 마찬가지로 24V 가 필요하며, 이를 위해 SMPS2 Dynamixel 과 같은 장치가 필요합니다.
- USB2Dynamixel 과 SMPS2Dynamixel 을 연결하고 SMPS2Dynamixel 과 Dynamixel Pro 를 연결합니다.



iv. Wiring 54 Series & 42 Series

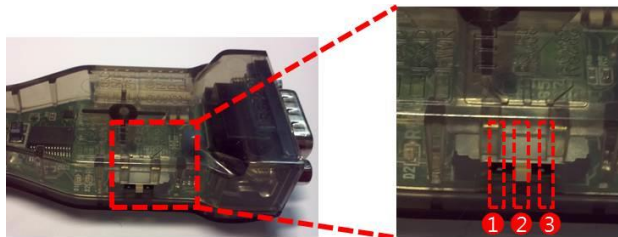
- 먼저 USB2 Dynamixel 과 Dynamixel Pro 를 연결합니다.
- PowerSupply 와 연결되어 있는 외부 전원 케이블을 Dynamixel Pro 에 연결합니다.

- Dynamixel Pro 54 시리즈는 외부전원 케이블과 4P 케이블 모두로부터 전원을 공급받을 수 있으나, 안정적인 사용을 위해서 외부전원 케이블로 전원을 공급하는 것을 추천합니다.
- 그리고 데이터 체인으로 모든 Dynamixel 을 연결합니다.
- 아래의 개략도를 참고 하시기 바랍니다.



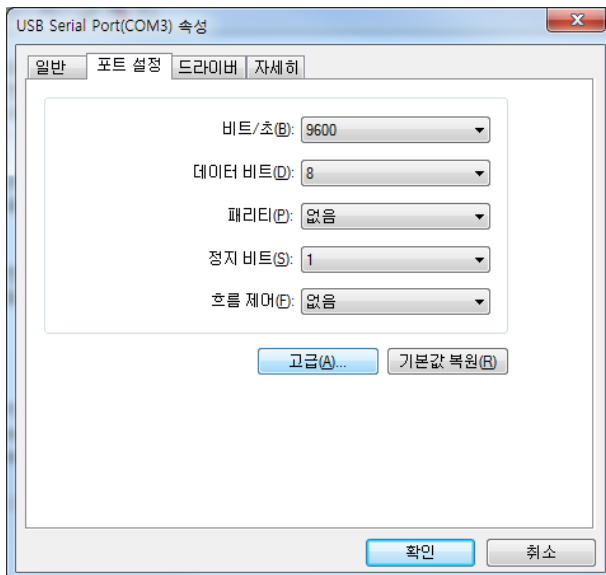
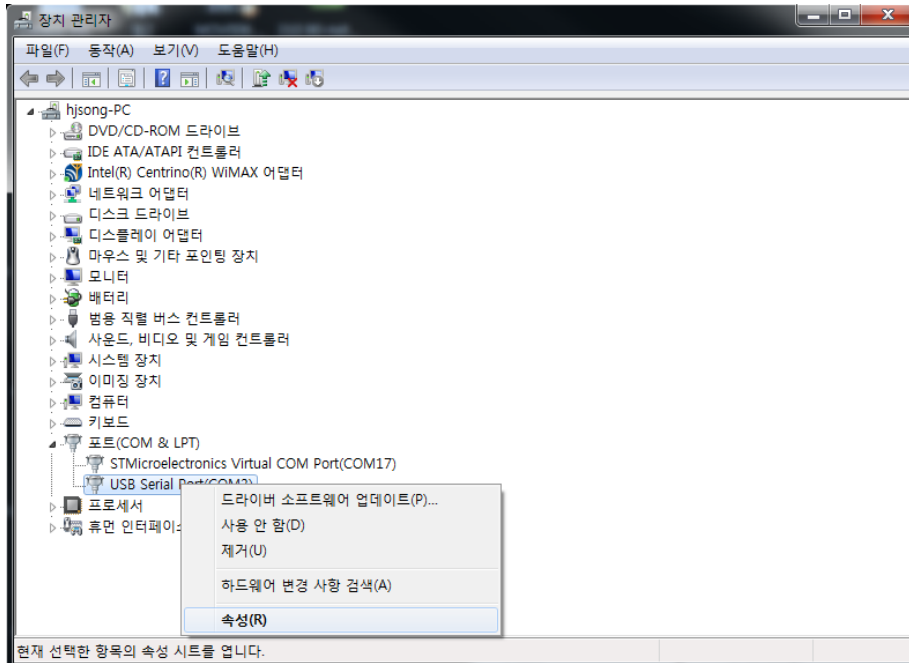
v. USB2Dynamixel (RS485 설정)

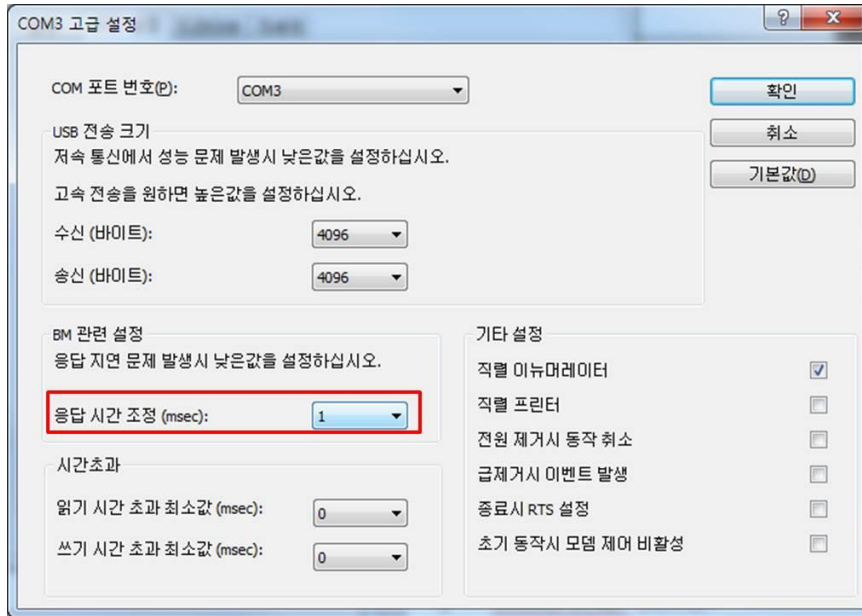
- Dynamixel PRO 는 RS485 방식을 사용합니다.
- 아래 그림과 같이 USB2Dynamixel 의 왼편에 있는 스위치를 2 번 위치에 오도록 하여 통신모드를 RS485 로 변경합니다.



vi. COM Port Latency Time 설정

- USB2Dynamixel 을 이용하여 Dynamixel Pro 를 테스트 및 사용하기 위해서는 포트의 Latency Time 을 조정하는 것이 좋습니다. 아래의 그림대로 Latency Time 을 조정해 주십시오.
- 장치관리자 -> 포트 -> USB Serial Port(오른쪽 마우스 버튼)
-> 속성 -> 포트설정 -> 고급 -> 응답시간조정 -> 1msec 로 설정




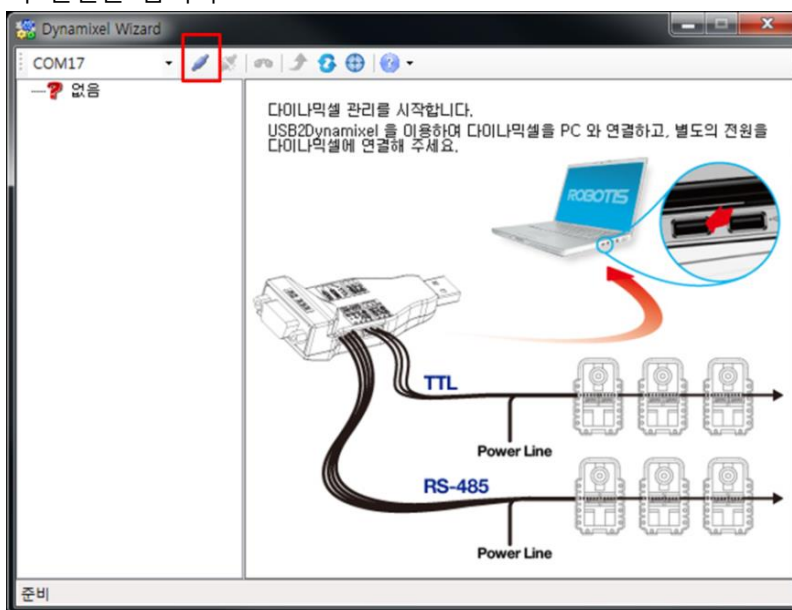


- 확인 버튼을 눌러서 설정을 완료합니다.

1.1.2 Dynamixel Wizard

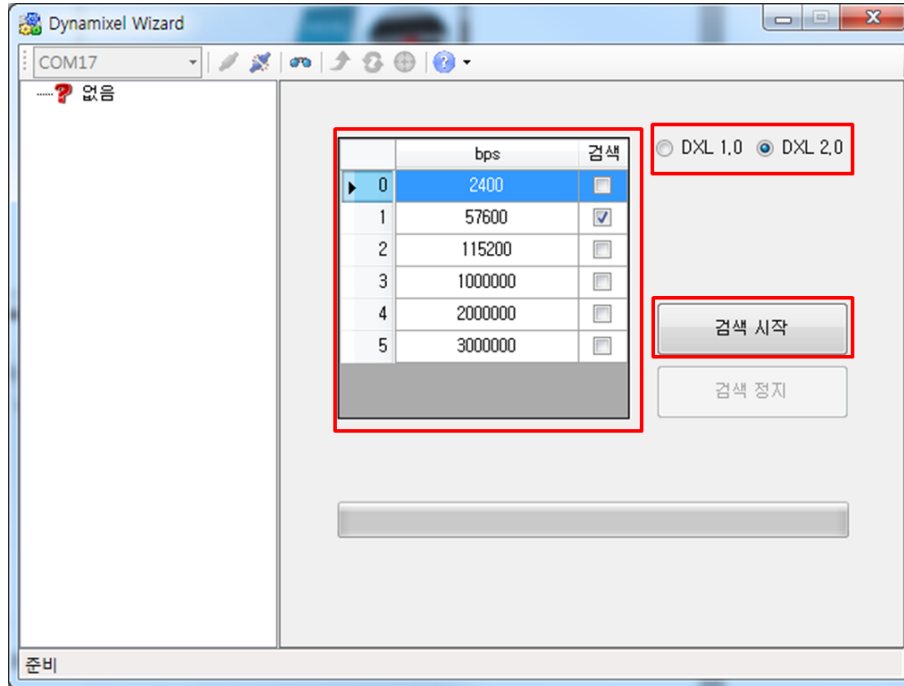
i. Dynamixel Pro 를 동작해봅시다.

- 배선이 완료된 상태에서 Dynamixel Pro 에 24V 의 전원을 공급합니다.
(안전한 사용을 위해 전원이 인가되어 있는 케이블을 Dynamixel Pro 에 연결하지 마시고, 연결이 완료된 상태에서 전원을 인가하십시오.)
- Roboplus 를 실행한 뒤 Dynamixel Wizard 를 실행합니다.
- USB2Dynamixel 에 할당된 COM Port 를 선택한 후  버튼을 눌러서 USB2 Dynamixel 과 연결을 합니다.

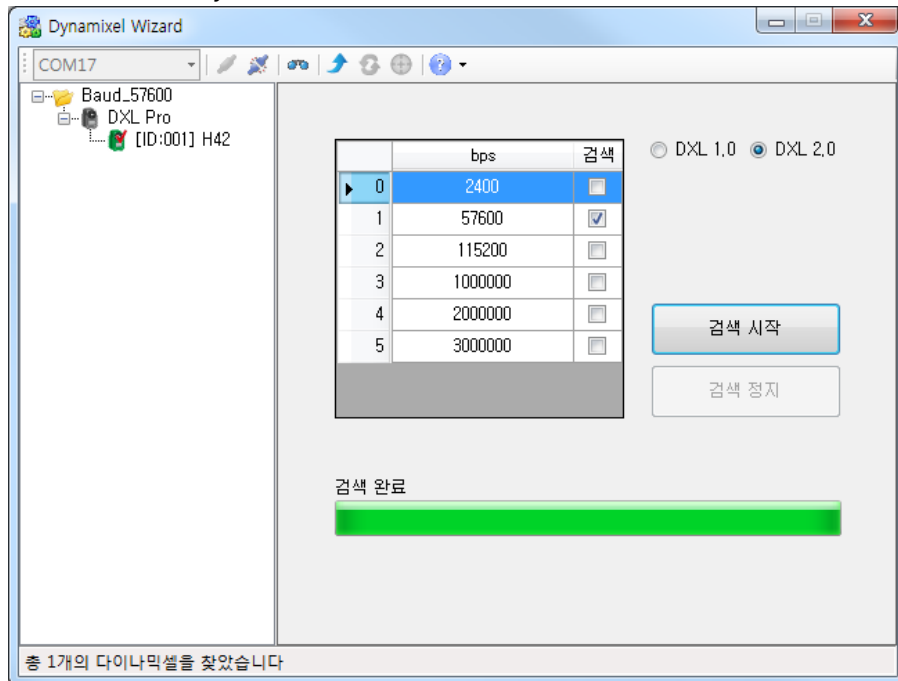


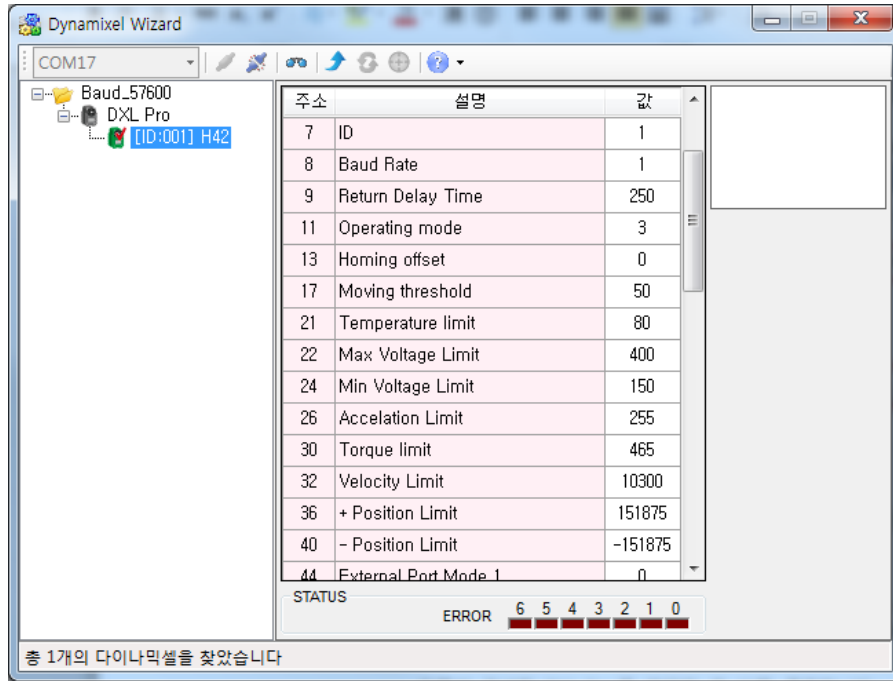
- 포트 연결이 완료된 후 DXL 2.0 으로 선택 그리고 57600 을 선택한 뒤 '검색 시작' 버튼을 누릅니다.

(Dynamixel Pro 의 Default 설정은 ID 는 1, 통신속도는 57600bps 입니다.)

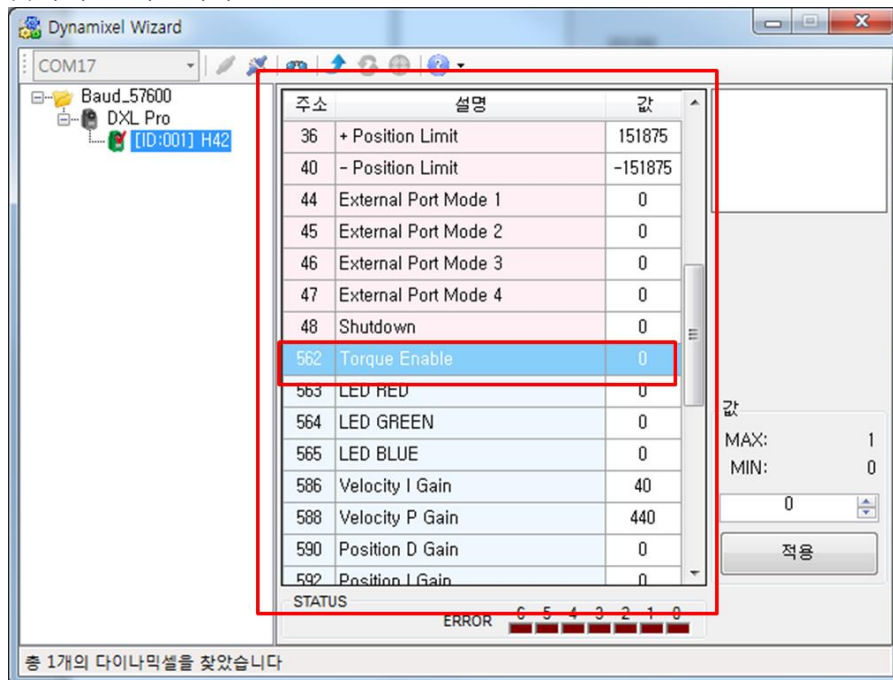


- 왼쪽에 검색된 Dynamixel Pro 를 확인한 후 이를 클릭합니다.

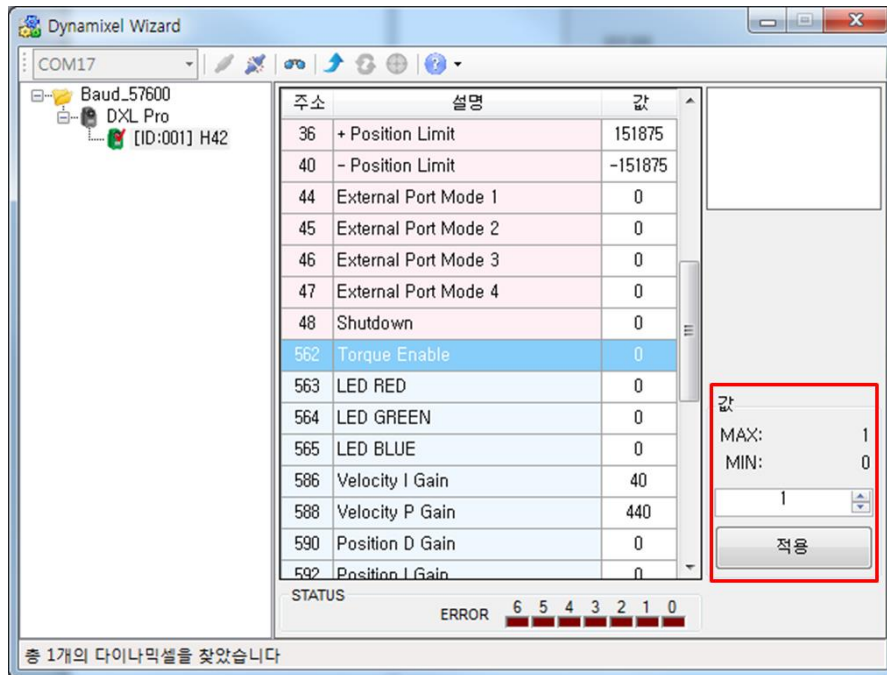




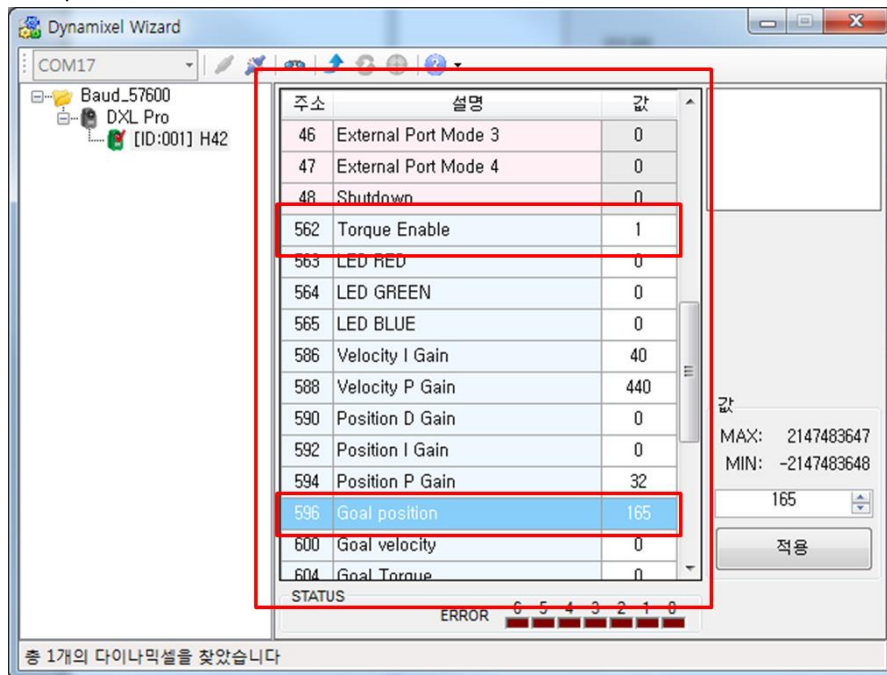
- Dynamixel Pro 는 기존 Dynamixel 과 달리 Torque 가 On 되어 있을 때만 동작할 수 있습니다.
- 따라서 Dynamixel Pro 를 동작시키기 위해서는 먼저 Torque 를 On 시켜야 합니다.
- Dynamixel Wizard 에서 가운데에 표시되어 있는 Table 에서 562 번인 'Torque Enable'을 찾아서 클릭합니다.



- 오른쪽 아래의 값에 1을 적은 뒤 '적용'버튼을 누릅니다.

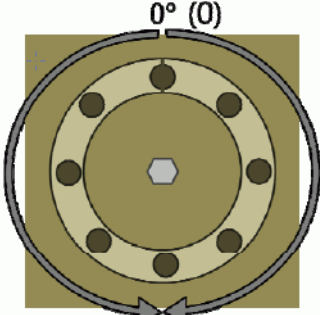
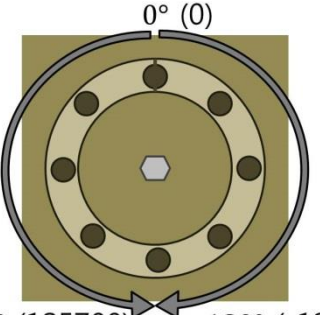
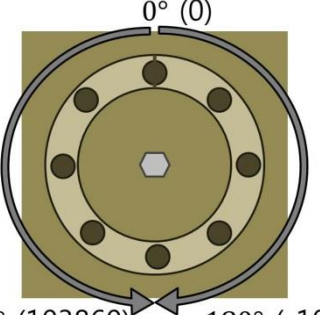


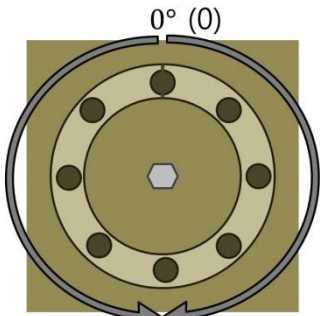
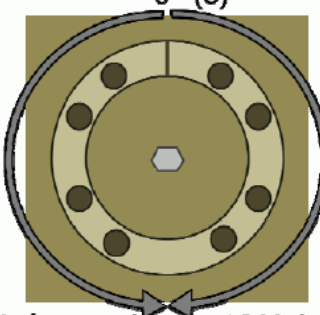
- 'Torque Enable'이 1로 바뀐 것을 확인한 뒤에 Goal Position을 찾은 뒤 클릭합니다.



- 오른쪽 아래의 값에 적당한 값을 입력한 뒤 적용 버튼을 누릅니다.
(Default 상태에서의 Dynamixel Pro 사용시에 입력할 수 있는 값은 2.1 모델별 초기값에서 Max Position Limit과 Min Position Limit을 참고해 주십시오.)
- 적용 버튼을 누르면 Dynamixel Pro가 동작하는 것을 확인할 수 있습니다.
- 만약, 움직이지 않는다면 Torque Enable의 값이 1로 되어 있는지 확인한 뒤, 1이 아닐 경우에는 1로 바꾸어 다시 시도를 합니다.

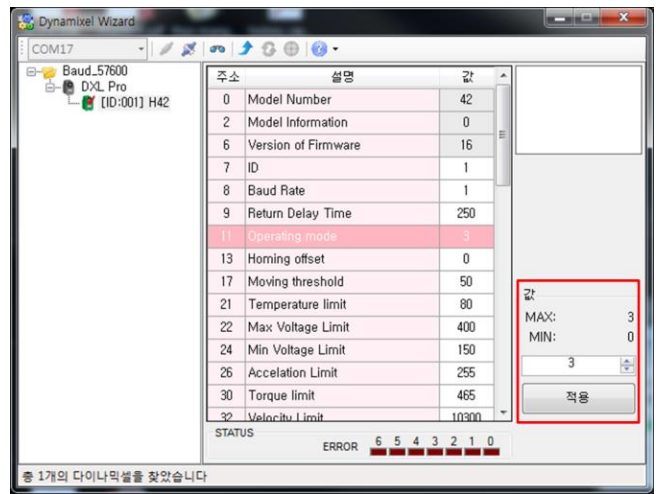
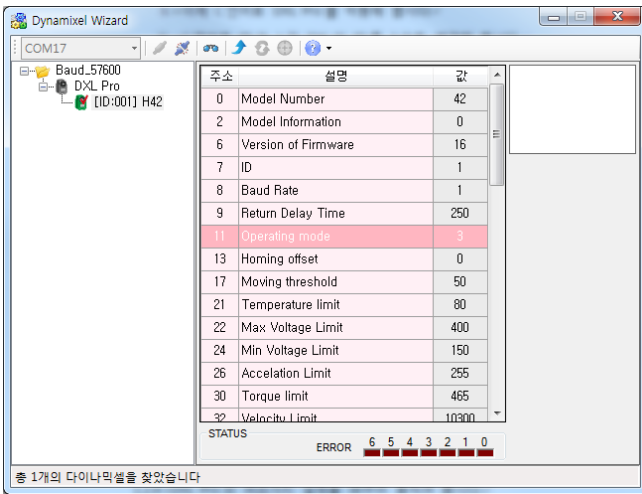
- Goal Position 에 다른 값을 적은 뒤 '적용' 버튼을 눌러 Dynamixel Pro 가 동작하는 지를 확인합니다.
- Goal Position 에 입력한 값과 Dynamixel Pro 의 회전 각 사이의 관계는 아래와 같습니다.

Model	Relationship between angle(deg) and position value
H54-200-S500-R H54-100-S500-R	<p style="text-align: center;">-180 ~ 180 (deg) → -250950 ~ 250950</p> <div style="text-align: center;">  <p>0° (0) 180° (250950) -180° (-250950)</p> </div> <p style="text-align: center;">$Goal\ Angle(deg) = Goal\ Position\ Value \times \frac{180^\circ}{250950}$</p>
M54-60-S250-R M54-40-S250-R	<p style="text-align: center;">-180 ~ 180 (deg) → -125700 ~ 125700</p> <div style="text-align: center;">  <p>0° (0) 180° (125700) -180° (-125700)</p> </div> <p style="text-align: center;">$Goal\ Angle(deg) = Goal\ Position\ Value \times \frac{180^\circ}{125700}$</p>
L54-50-S290-R	<p style="text-align: center;">-180 ~ 180 (deg) → -103860 ~ 103860</p> <div style="text-align: center;">  <p>0° (0) 180° (103860) -180° (-103860)</p> </div> <p style="text-align: center;">$Goal\ Angle(deg) = Goal\ Position\ Value \times \frac{180^\circ}{103860}$</p>

<p>L54-30-S400-R</p>	<p>$-180 \sim 180 \text{ (deg)} \rightarrow -144180 \sim 144180$</p>  <p>$Goal \text{ Angle(deg)} = Goal \text{ Position Value} \times \frac{180^\circ}{144180}$</p>
<p>H42-20-S300-R</p>	<p>$-180 \sim 180 \text{ (deg)} \rightarrow -151900 \sim 151900$</p>  <p>$Goal \text{ Angle(deg)} = Goal \text{ Position Value} \times \frac{180^\circ}{151900}$</p>

ii. Dynamixel Pro 를 Wheel Mode 로 바꾼 뒤 동작시켜 봅시다.

- Dynamixel Pro 는 총 세가지 Mode 로 사용할 수 있습니다.
(Joint Mode, Wheel Mode, Torque Mode)
- Dynamixel Wizard 에 가운데에 표시되어 있는 Table 에서 11 번인 'Operating Mode'를 찾아 클릭합니다.
- 562 번인 Torque Enable 에 1 이라는 값이 적혀 있어 Torque 가 On 된 상태일 때는 아래의 왼쪽 그림과 같이 Mode 변경이 되지 않으며, 만약 Torque Enable 에 0 이라는 값이 적혀 있어 Torque 가 Off 된 상태일 때는 아래의 오른쪽 그림과 같이 Mode 변경을 할 수 있습니다.

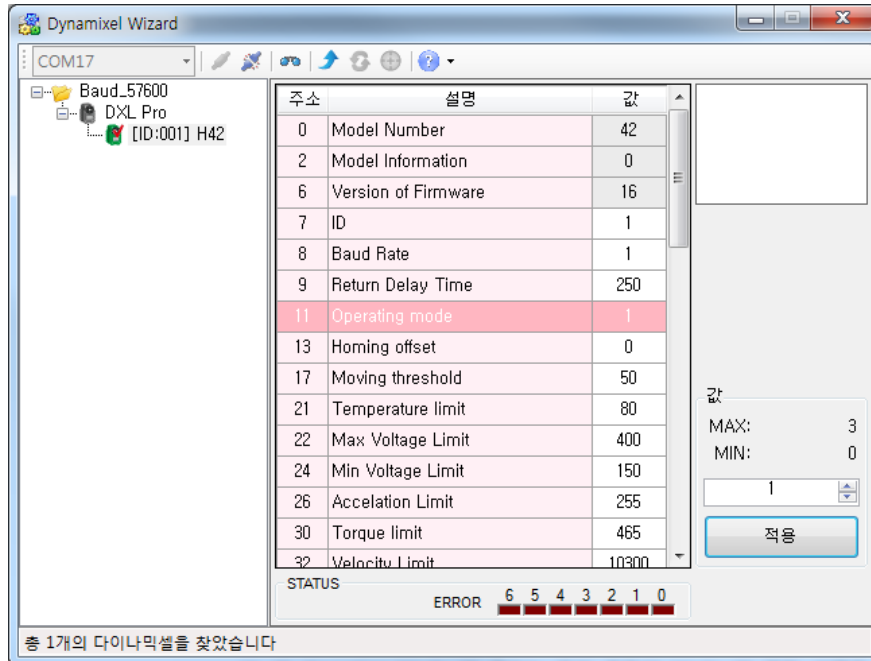


- 11 번 Operating Mode 에 적히는 값과 각각의 Mode 에 관한 설명은 아래와 같습니다.

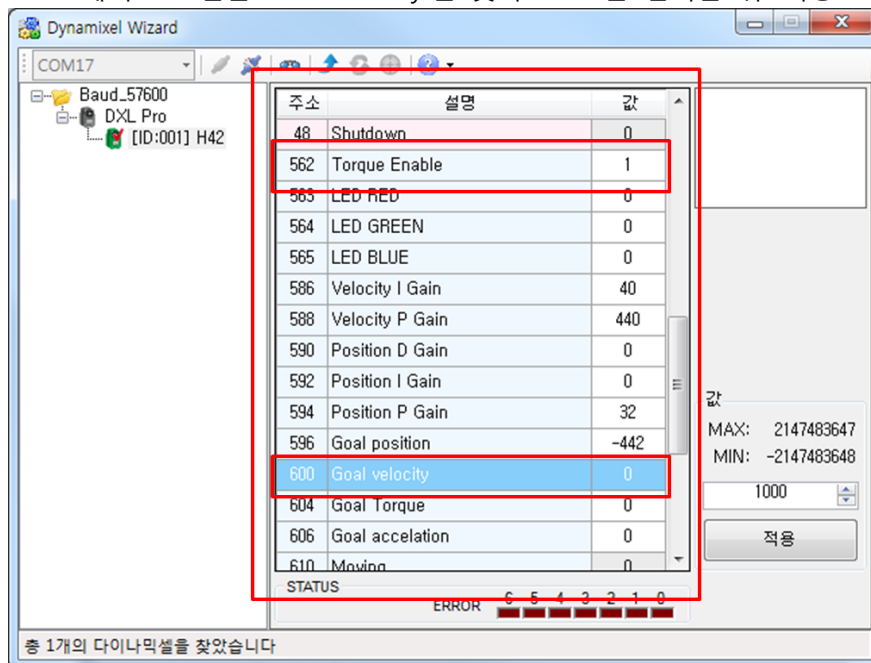
Operating Mode	Value	설명
Joint Mode	3	관절 모드로 위치 및 속도 제어가 가능합니다. 1. 원하는 위치까지 원하는 속도로 회전시킬 수 있습니다. 2. 내부적으로 Velocity Profile 을 생성하여 회전하도록 할 수 있습니다.
Wheel Mode	1	바퀴 모드로 속도 제어가 가능합니다. 1. 원하는 속도로 회전 시킬 수 있습니다. 2. 내부적으로 Velocity Profile 을 생성하여 회전하도록 할 수 있습니다. 3. 바퀴모드에서 위치 제어는 불가능 합니다.
Torque Mode	0	토크 모드로 토크 제어가 가능합니다. 1. 위치 및 속도 제어를 하지 않습니다.. 2. 위치 및 속도 제어가 되지 않고 오직 출력 토크만을 제어할 수 있습니다 3. 마치 바퀴모드처럼 행동합니다.

- 위의 내용을 참고하여, 사용하려는 목적에 따라서 모드를 선택한 뒤에 사용하시면 됩니다.

- Torque Enable 에 0 을 적은 뒤 적용 버튼을 누르고, Operating Mode 에 1 을 적은 뒤 적용 버튼을 누릅니다.



- 그 뒤 562 번인 Torque Enable 에 1 을 적은 뒤 적용 버튼을 누릅니다.
※ Dynamixel Pro 시리즈는 움직이기 전에 반드시 Torque Enable 에 1 을 적어 넣어 Torque On 상태가 되도록 해야 합니다.
- Table 에서 600 번인 Goal Velocity 를 찾아 1000 을 입력한 뒤 '적용' 버튼을 누릅니다.



- Dynamixel Pro 가 시계 반대 방향으로 천천히 움직이는 것을 확인 합니다.
- 만약, 동작하지 않는다면 'Torque On'이 되어 있는지를 확인합니다.

- 이번에는 600 번인 Goal Velocity 에 -1000 을 적은 뒤 적용 버튼을 누릅니다.
- 시계방향으로 회전하는 것을 확인합니다.
- Goal Velocity 에 0 을 입력한 뒤 동작이 멈추는 것을 확인합니다.
- 3000, 10000, 15000, -3000, -10000, -15000, 0 등의 값을 넣은 뒤 적용 버튼을 눌러 어떻게 움직이는지 확인을 합니다.
- 600 번 Goal Velocity 에 적히는 값과 출력 rpm 과의 상관 관계는 다음과 같습니다.

Sign of Value	Rotating Direction	Rpm(20V 이상)
+	CCW	$\frac{\text{Magnitude of Value}}{\text{Gear Reduction Ratio}}$
-	CW	

※ Notice 1

- Table 에서 562 번 Torque Enable 에 1 이 적혀 있으면 Dynamixel Pro 를 동작시킬 수 있는 상태가 되며 이를 **Torque On** 상태라 합니다.
- Table 에서 562 번 Torque Enable 에 0 이 적혀 있으면 Dynamixel Pro 를 동작시킬 수 없으며 이를 **Torque Off** 상태라 합니다.

※ Notice 2

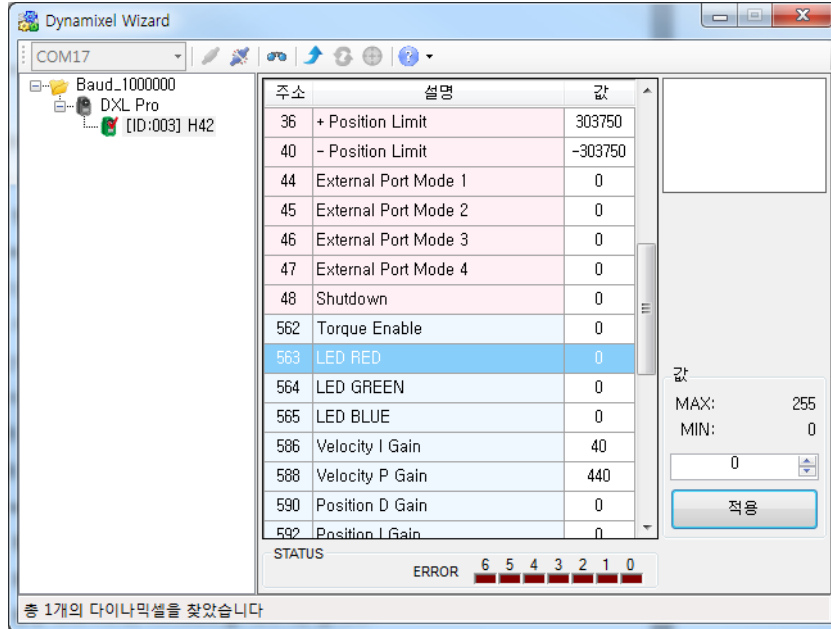
- Dynamixel Wizard 의 가운데에 표시되어 있는 Table 을 Control Table 이라 합니다.
- Control Table 은 Dynamixel 에서 Control 할 수 있는 목록이 망라 되어 있는 Table 입니다.
- Control Table 에서 어떤 주소 값에 적절한 값을 입력하면 Dynamixel 을 동작하거나 설정을 바꿀 수 있습니다.
- 예를 들어, Control Table 에서 562 번(Torque Enable 주소)에 0 또는 1 을 적어 넣으면 Torque 를 On/Off 할 수 있으며, 600 번(Goal Velocity)에 적절한 값을 적어 넣으면 원하는 속도로 Dynamixel 을 회전 시킬 수 있습니다.

※ Notice 3

- Control Table 에서 Dynamixel Pro 의 Operating Mode 와 같이 Torque On 상태에서 변경할 수 없는 Control Table 의 영역을 EEPROM 영역이라 합니다.
- EEPROM 영역에 적혀있는 값은 Dynamixel Pro 를 꺾다 켜도 초기화 되지 않으며, 앞에서 언급 했듯이 Torque On 상태에서는 값을 변경할 수 없으며, EEPROM 영역의 값을 변경시키기 위해서는 반드시 Torque Off 상태가 되어야 합니다.
- EEPROM 영역은 Dynamixel Wizard 에서 분홍색으로 표시됩니다.
- Control Table 에서 Dynamixel Pro 의 Goal Velocity 나 Goal Position 과 같이 Dynamixel Wizard 에서 하늘색으로 표시된 부분을 Control Table 의 RAM 영역이라 합니다.
- RAM 영역에 적혀있는 값은 Dynamixel Pro 를 꺾다 켜면 값이 초기화 되며, RAM 영역은 Torque On/Off 상태와 상관없이 변경할 수 있습니다.

iii. Dynamixel Pro 의 LED 를 On/Off 해 봅시다.

- Dynamixel Pro 의 LED 는 기존 Dynamixel 과 달리 3 색 LED 입니다.
- 따라서 LED 를 여러색깔로 바꿀 수 있습니다.
- Control Table 에서 563 번인 LED RED 를 클릭해 봅시다.

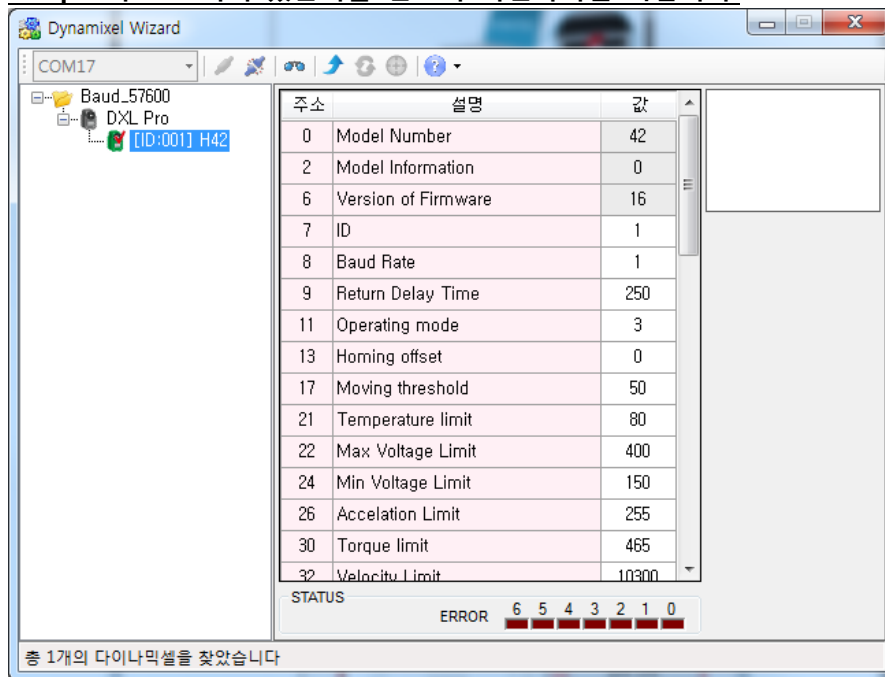


- 255 를 적은 뒤 적용 버튼을 누른 후 빨간색 LED 가 켜지는지를 확인합니다.
- 0 에서 255 사이의 값을 적은 뒤 적용 버튼을 눌러 밝기 변화를 확인합니다.
- LED RED, LED GREEN, LED BLUE 에도 적당한 값을 입력하여 LED 의 색깔과 밝기의 변화를 확인합니다.

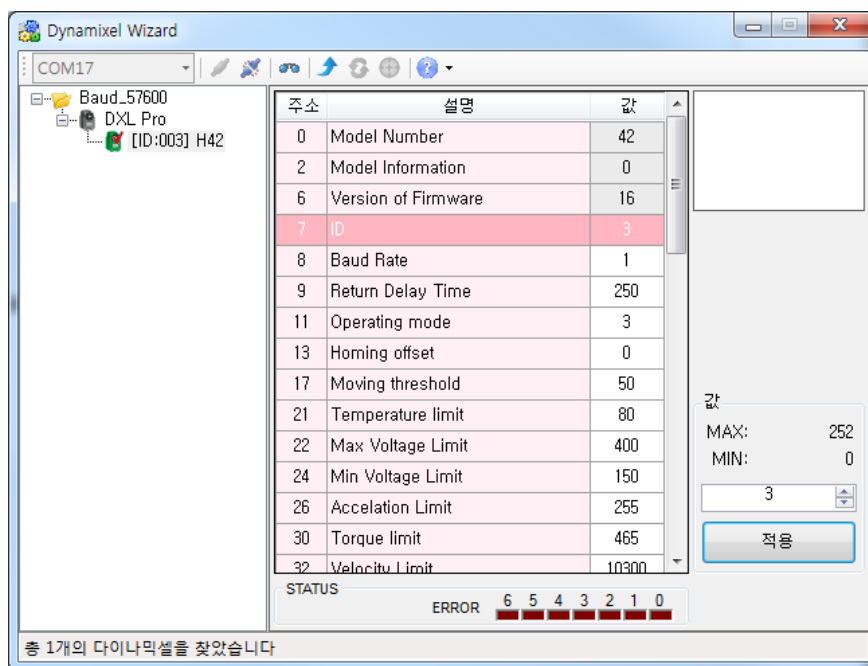


iv. ID 를 3 으로 변경해 봅시다.

- 여러 개의 Dynamixel Pro 가 연결되어 있을 때, Dynamixel Pro 는 ID 로 구분이 되어 통신이 됩니다.
- 예를 들어, 3 개의 Dynamixel Pro 가 연결되어 있을 때, 하나의 Dynamixel Pro 만을 동작 시키려 한다면, 동작시키려는 Dynamixel Pro 의 ID 를 이용하여 명령을 내리면 하나의 Dynamixel Pro 만이 동작하게 됩니다.
- 만약, 연결되어 있는 Dynamixel Pro 중에서 같은 ID 를 가진 Dynamixel Pro 가 연결되어 있을 경우 오작동을 일으킬 수 있습니다.
- Dynamixel Wizard 에 표시된 Control Table 중 7 번이 Dynamixel Pro 의 ID 입니다.
- **Control Table 에서 ID 는 EEPROM 영역 이므로 ID 를 변경하기 전에 Dynamixel Pro 의 Torque 가 Off 되어 있는지를 반드시 확인하시길 바랍니다.**

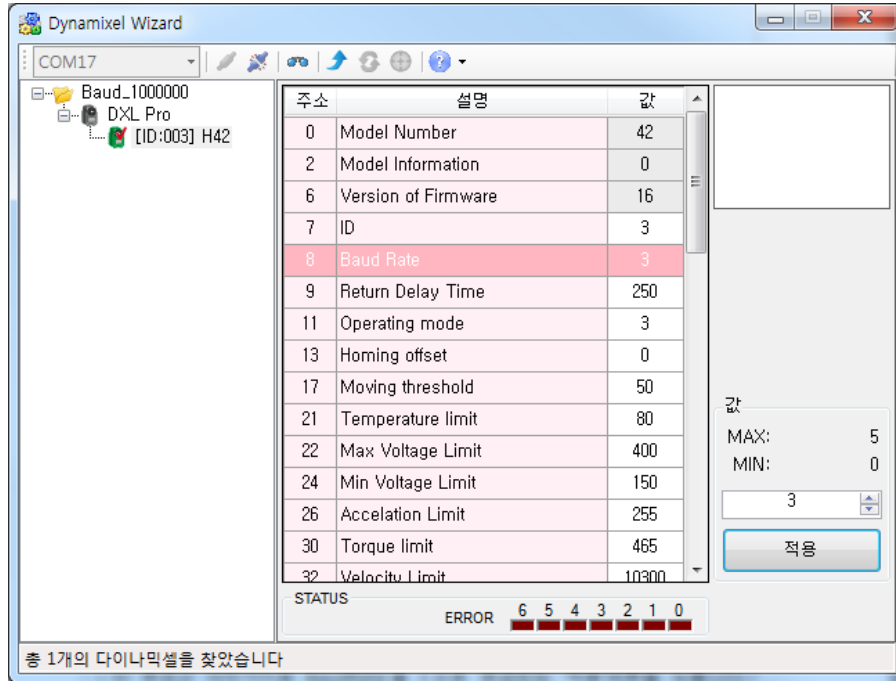


- Control Table 에서 7 번을 클릭한 후 ID 를 3 으로 바꾼 뒤 '적용' 버튼을 누르면, 왼쪽의 Dynamixel Pro 의 목록에서 ID 가 바뀐 것을 확인할 수 있습니다.



- 만약, Dynamixel Pro 의 ID 를 변경할 수가 없을 때에는 Torque Off 상태인지 확인하고 다시 시도해 보시길 바랍니다.

- v. 통신속도를 변경하여 더 빠르게 Dynamixel Pro 를 이용해 봅시다.
 - Control Table 에서 8 번이 Dynamixel Pro 의 Baudrate(통신속도) 입니다.
 - **Control Table 에서 Baudrate(통신속도)는 EEPROM 영역 이므로 ID 를 변경하는 것과 마찬가지로 반드시 Torque 가 Off 상태에서만 바꿀 수 있습니다.**
 - ID 변경과 마찬가지로 Baudrate 를 3 으로 변경한 뒤 '적용'버튼을 누릅니다.



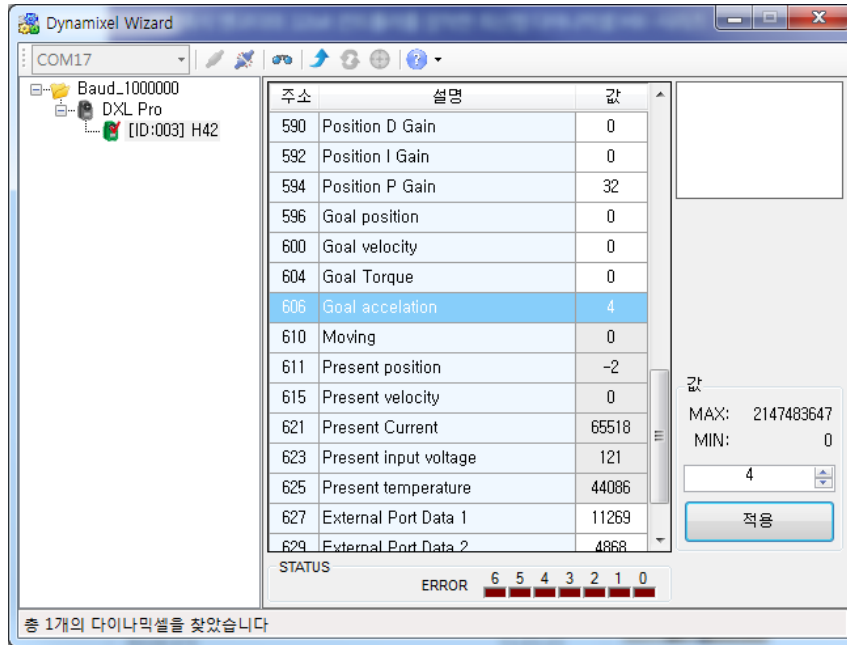
- 통신속도가 1Mbps 로 변경된 것을 확인할 수 있습니다.
- 만약, Dynamixel Pro 의 Baudrate(통신속도)를 변경할 수가 없을 때에는 Torque Off 상태인지를 확인하고 다시 시도해 보시길 바랍니다.
- Dynamixel Pro 의 통신속도와 Control Table 에 입력되는 값과의 관계는 다음과 같습니다.

Value of Control Table	Baud Rate(bps: bit for seconds)
0	2400 bps
1	57600 bps
2	115200 bps
3	1 Mbps
4	2Mbps
5	3Mbps
6	4Mbps
7	4.5Mbps
8	10.5Mbps

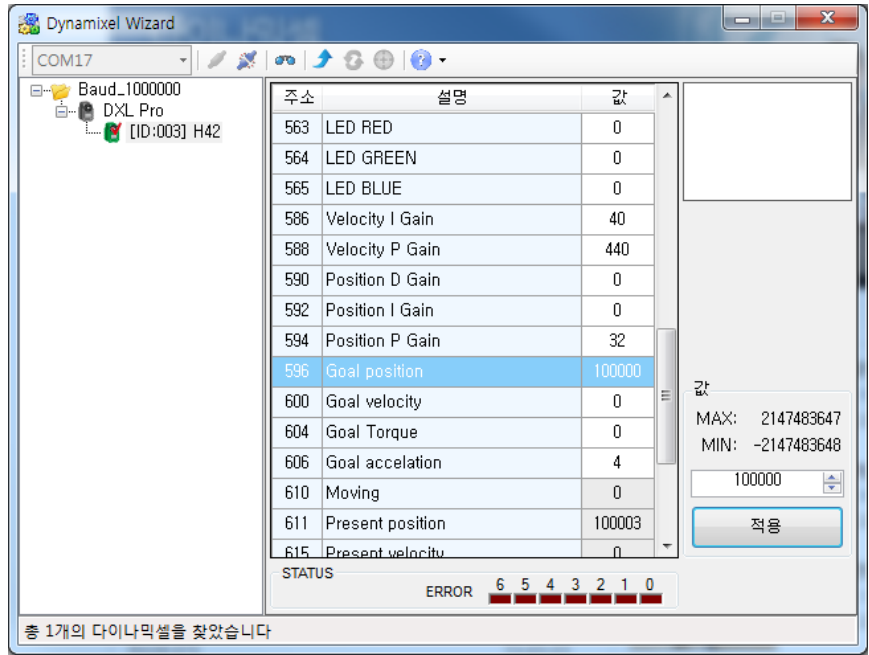
- **※ USB2Dynamixel 에 따라 지원되는 통신속도가 다릅니다. 따라서, 사용하고 있는 USB2Dynamixel 이 지원하는 통신속도 이하로만 셋팅을 해야합니다.**

vi. Joint Mode 로 변경한 후 가속도를 설정하고 움직여 봅시다.

- Dynamixel Pro 가 Torque Off 상태인지를 확인한 후 Torque On 상태라면 Control Table 에서 562 번인 Torque Enable 을 클릭한 뒤 0 을 입력한 뒤 '적용'버튼을 눌러 Torque Off 상태로 만듭니다.
- Control Table 인 11 번인 Operating Mode 에 3 을 입력한 뒤 '적용'버튼을 눌러 Dynamixel Pro 를 Joint Mode 로 변경합니다.
- 606 번인 Goal Acceleration 에 4 를 입력한 뒤 '적용'버튼을 누릅니다.



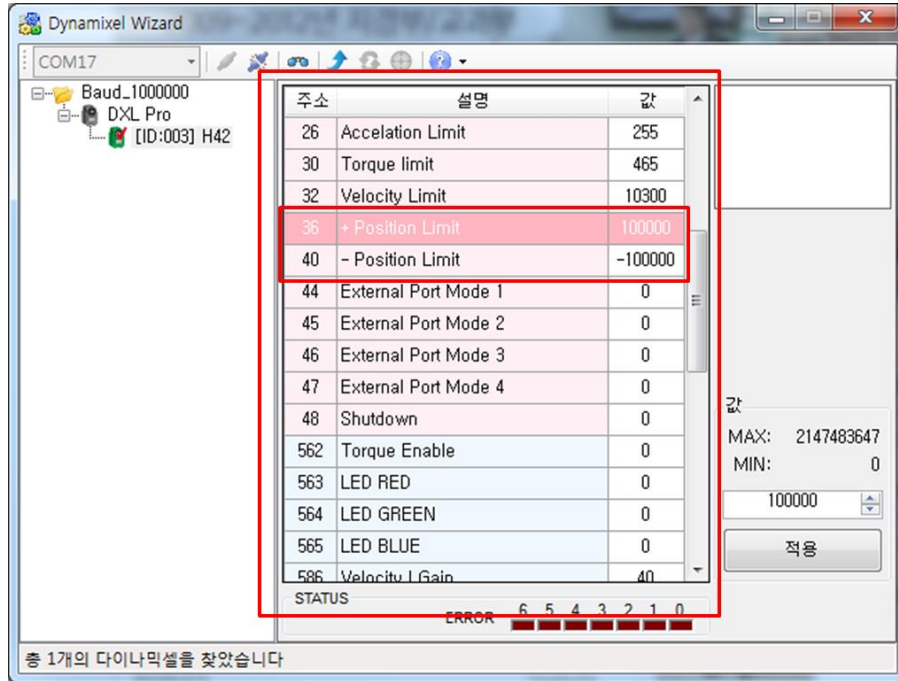
- 562 번인 Torque Enable 에 1 을 입력한 뒤 '적용'버튼을 눌러 Dynamixel Pro 가 'Torque On' 상태가 되도록 합니다.
- 596 번인 Goal Position 에 100,000 혹은 -100,000 값을 입력한 뒤 '적용' 버튼을 눌러서 Goal Acceleration 이 0 으로 설정되어 있을 때와의 움직임의 차이를 살펴 봅니다.



- 만약 동작하지 않는다면, Torque On 상태인지를 확인하시기 바랍니다.
- 또한, Goal Acceleration 이 적히지 않았을 때와 움직임이 다르지 않다면 Goal Acceleration 값이 4로 적혀있는지 확인하기 바랍니다.
- Goal Acceleration 에 0이 아닌 값이 적히면 내부적으로 'Trapezoidal Velocity Profile' 을 생성하여 움직이게 되어 더 부드러운 움직임이 가능해 집니다.
- Goal Acceleration 과 Goal Position 에 임의의 값을 적어 넣은 뒤 Dynamixel Pro 가 어떻게 움직이는지 살펴 봅니다.

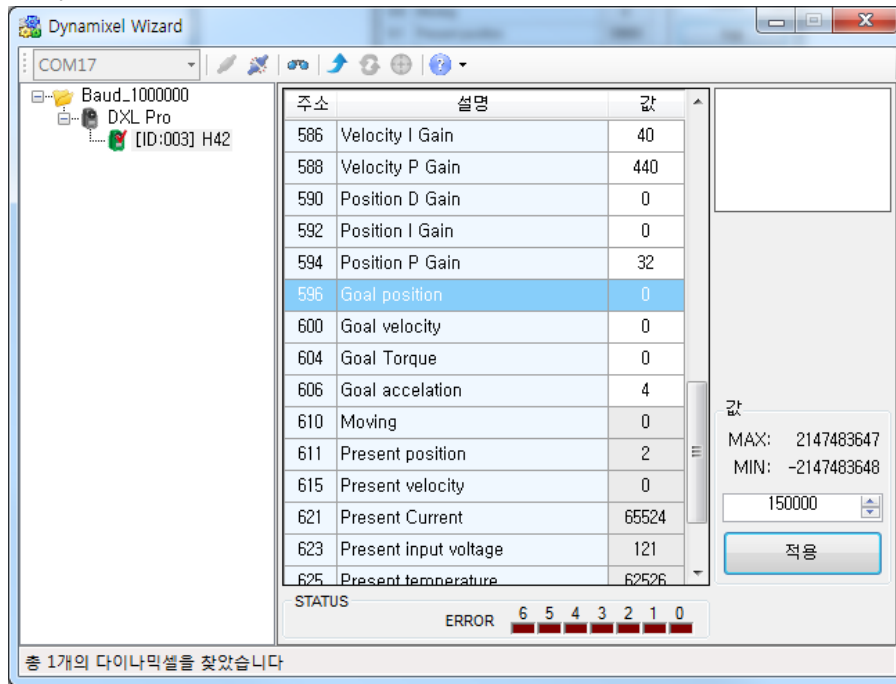
vii. 작동 범위를 줄인 후 이를 확인해 봅시다.

- Dynamixel Pro 를 Torque Off 상태로 만듭니다.
- Control Table 에서 36 번인 Max Position Limit 에 100,000 을 입력한 뒤 '적용' 버튼을 누릅니다.
- 40 번인 Min Position Limit 에 -100,000 을 입력한 뒤 '적용' 버튼을 누릅니다.



- **36 번인 Max Position Limit 과 40 번인 Min Position Limit 은 EEPROM 영역입니다.**
- **그러므로, 만약 값을 바꿀 수가 없다면 Torque On 상태가 아닌지 확인하시기 바랍니다.**
- Max Position Limit 과 Min Position Limit 의 값이 바뀌었다면 Torque On 이 되도록 합니다.

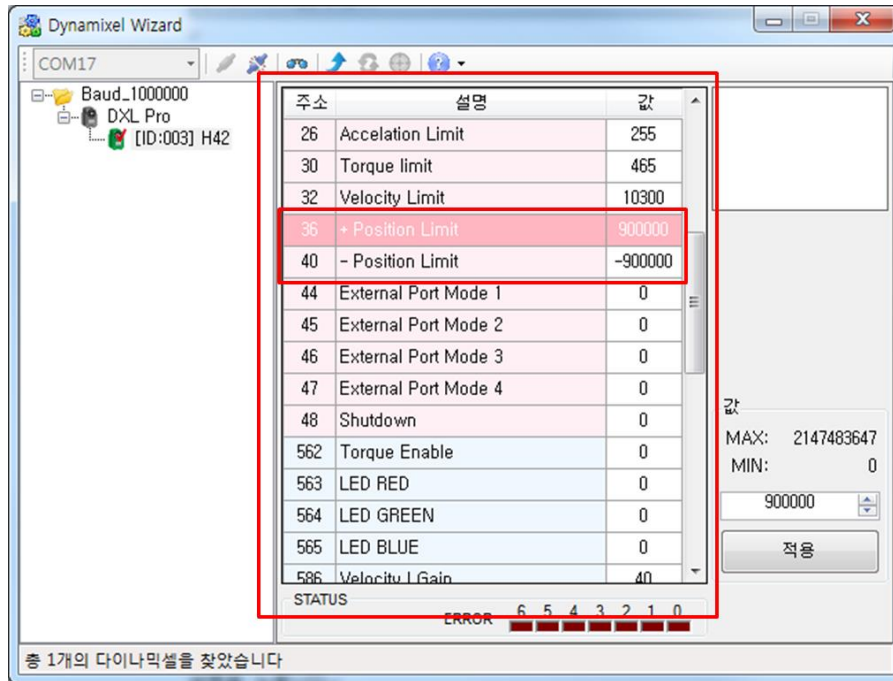
- Torque On 이 되면 596 번인 Goal Position 에 150,000 을 입력합니다.



- Position Limit 값을 넘어선 값을 입력했기 때문에 Torque On 상태임에도 불구하고 반응을 하지 않습니다.
- 만약 Dynamixel Pro 가 동작을 한다면 Max Position Limit 값을 확인하고 값이 100,000 이 아니라면 Torque Off 상태로 만든 뒤 Max Position Limit 에 100,000 을 입력한 뒤 '적용'버튼을 누르고 Torque On 상태로 만듭니다.
- 이번에는 Goal Position 에 100,000 을 입력합니다.
- 100,000 에 해당하는 위치로 이동하는 것을 확인할 수 있습니다.
- -150,000 을 입력하여 동작하지 않음을 확인하고 -100,000 을 입력하여 동작함을 확인합니다.
- Goal Position 에 임의의 값을 입력한 뒤 '적용'버튼을 눌러 Position Limit 안에서만 움직이는지를 확인합니다.

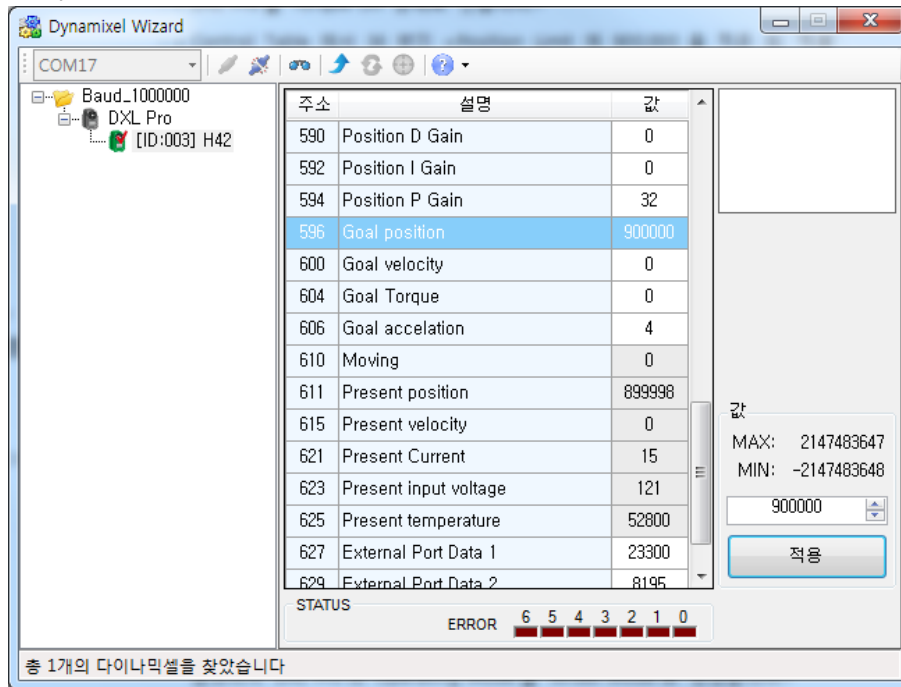
viii. 작동 범위를 늘린 후 이를 확인해 봅시다.

- Max Position Limit 과 Min Position Limit 은 Dynamixel Pro 의 동작범위를 줄일 때도 사용되지만, 동작범위를 늘릴 때에도 사용됩니다.
- Dynamixel Pro 를 Torque Off 상태로 만듭니다.
- Control Table 에서 36 번인 Max Position Limit 에 900,000 을 입력한 뒤 '적용' 버튼을 누릅니다.
- 40 번인 -Position Limit 에 Min 900,000 을 입력한 뒤 '적용' 버튼을 누릅니다.



- 36 번인 Max Position Limit 과 40 번인 Min Position Limit 은 EEPROM 영역입니다.
- 만약 값을 바꿀 수가 없다면 Torque On 상태가 아닌지 확인하시기 바랍니다.
- Max Position Limit 과 Min Position Limit 값이 바뀌었다면 Torque On 이 되도록 합니다.

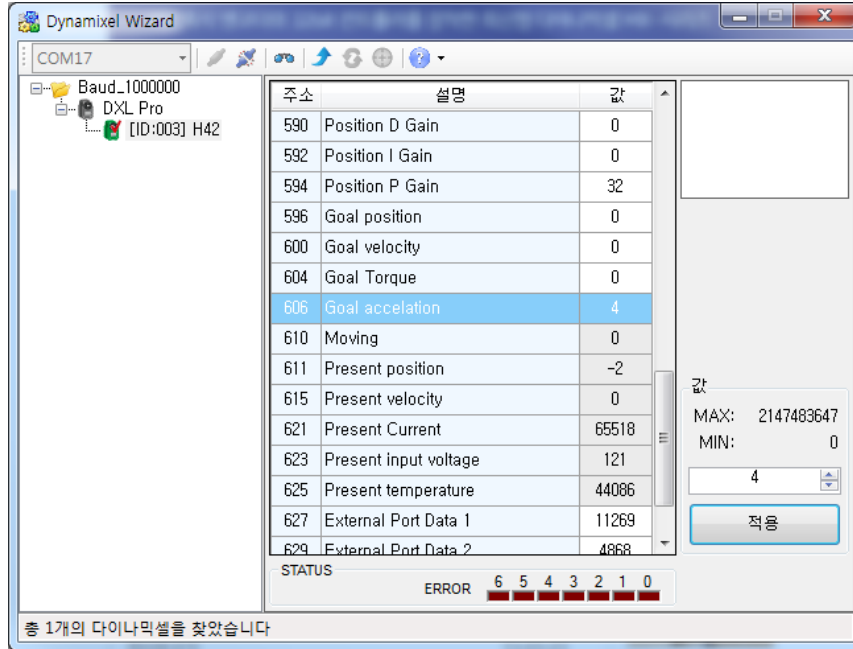
- Torque On 이 되면 596 번인 Goal Position 에 900,000 을 입력합니다.



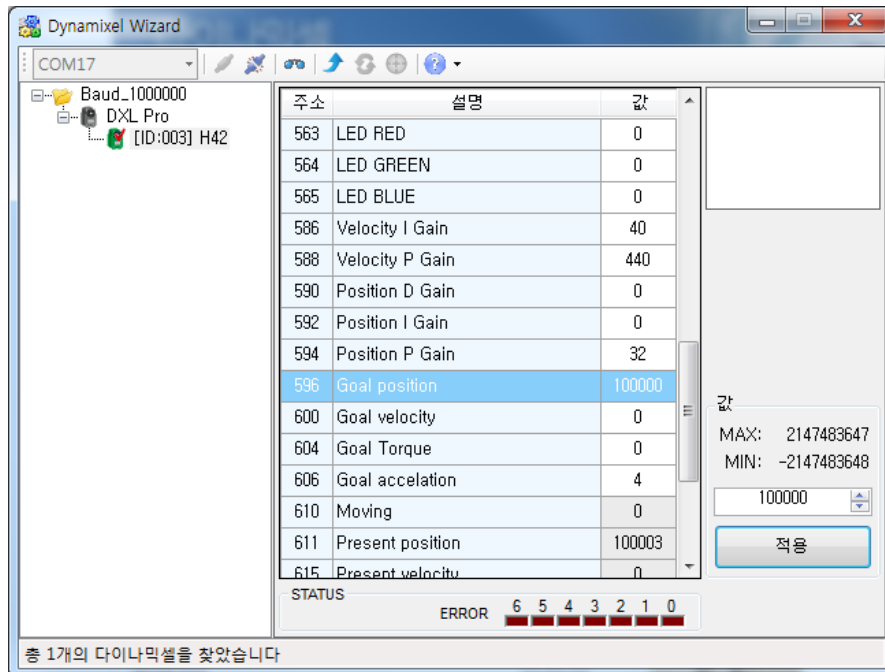
- Dynamixel Pro 가 360 도 이상을 돌아 해당 위치로 가는 것을 확인할 수 있습니다.
- 만약 Dynamixel Pro 가 반응을 하지 않는다면 Max Position Limit 값을 확인하고 값이 900,000 이 아니라면 Torque Off 상태로 만든 뒤 Max Position Limit 에 900,000 을 입력한 뒤 '적용'버튼을 누르고 Torque On 상태로 만듭니다.
- 이번에는 Goal Position 에 -900,000 을 입력합니다.
- 시계 반대방향으로 몇 바퀴를 돌아서 해당하는 위치로 이동함을 알 수 있습니다.
- Goal Position 에 임의의 값을 입력한 뒤 '적용'버튼을 눌러 해당하는 위치로 이동함을 확인합니다.
- Goal Position 에 입력한 값과 각도와와의 관계는 1.1.2. i 의 표를 참고합니다.

ix. Wheel Mode 로 변경한 후 가속도를 설정하고 움직여 봅시다.

- '1.1.2.ii Wheel Mode 로 바꾼 뒤 Dynamixel Pro 를 움직여 봅시다.' 를 참고하여 Dynamixel Pro 의 Operating Mode 를 Wheel Mode 로 변경합니다.
- Control Table 에서 606 번인 Goal Acceleration 에 4 를 입력한 뒤 '적용' 버튼을 누릅니다.



- 562 번인 Torque Enable 에 1 을 입력한 뒤 '적용' 버튼을 눌러 'Torque On' 상태가 되도록 합니다.
- 600 번인 Goal Velocity 에 5,000 혹은 -5,000 값을 적어 넣은 뒤 '적용' 버튼을 눌러서 Goal Acceleration 이 0 으로 설정되어 있을 때와의 움직임의 차이를 살펴 봅니다.

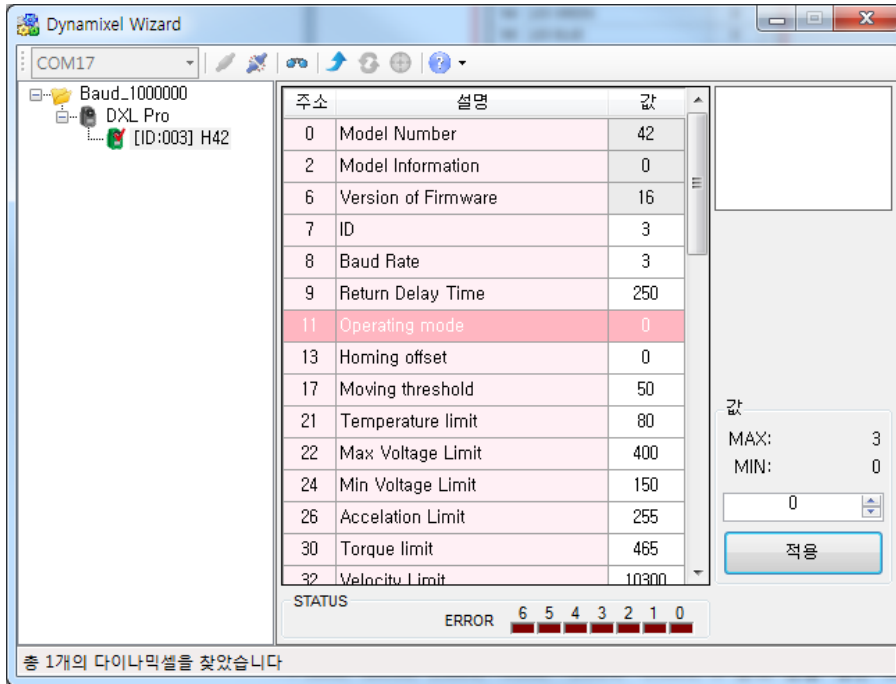


- 동작하지 않는다면, Torque On 상태인지를 확인하기 바랍니다.

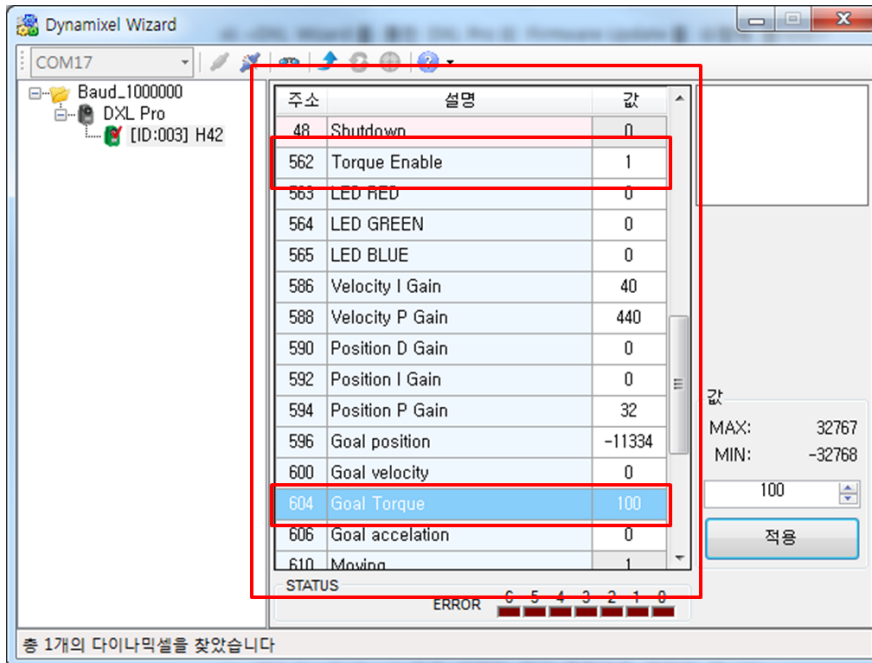
- 또한, Goal Acceleration 이 0 이 입력되어 있을 때와 움직임이 다르지 않다면 Goal Acceleration 값이 4 로 적혀있는지 확인하기 바랍니다.
- Goal Acceleration 에 0 이 아닌 값이 적히면 내부적으로 'Trapezoidal Velocity Profile' 을 생성하여 움직이게 되어 더 부드러운 움직임이 가능해 집니다.
- Goal Acceleration 과 Goal Velocity 에 임의의 값을 적어 넣은 뒤 Dynamixel Pro 가 어떻게 움직이는지 살펴 봅니다.

x. Torque Mode 로 변경한 후 동작해 봅시다.

- Torque Off 상태로 만든 뒤, Operating Mode 에 0 을 입력하고 '적용' 버튼을 누릅니다.



- 다시 Torque On 상태가 되도록 합니다.
- Table 에서 604 번인 Goal Torque 를 찾아 100 을 입력한 뒤 '적용' 버튼을 누릅니다.



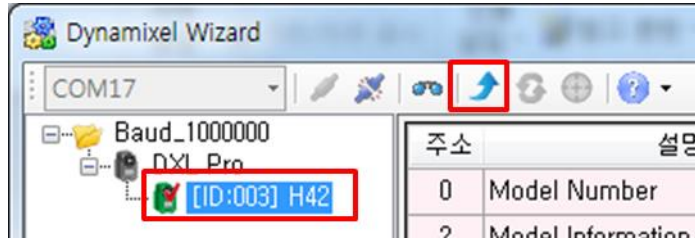
- 시계 반대 방향으로 점점 빨리 회전하는 것을 확인합니다.
- 만약, Dynamixel Pro 가 움직이지 않는다면 Torque Off 된 상태인지를 확인합니다.
- 이번에는 604 번인 Goal Torque 에 -100 을 적은 뒤 적용 버튼을 누릅니다.

- 시계방향으로 점점 빨리 회전하는 것을 확인합니다.
- Goal Velocity 에 0 을 입력한 뒤 Dynamixel Pro 의 동작이 멈추는 것을 확인합니다.
- 30, 100, 400, -400, -100, -30, 0 등의 값을 입력한 뒤 '적용' 버튼을 눌러 Dynamixel Pro 가 어떻게 움직이는지 확인을 합니다.
- 604 번 Goal Torque 에 적히는 값은 Dynamixel Pro 에 흐르는 전류를 제어하는 것입니다.
- Goal Torque 에 적히는 값과 전류의 상관관계는 다음과 같으며, 흐르는 전류와 출력 Torque 간의 상관관계는 Dynamixel Pro Datasheet 를 참고하십시오.

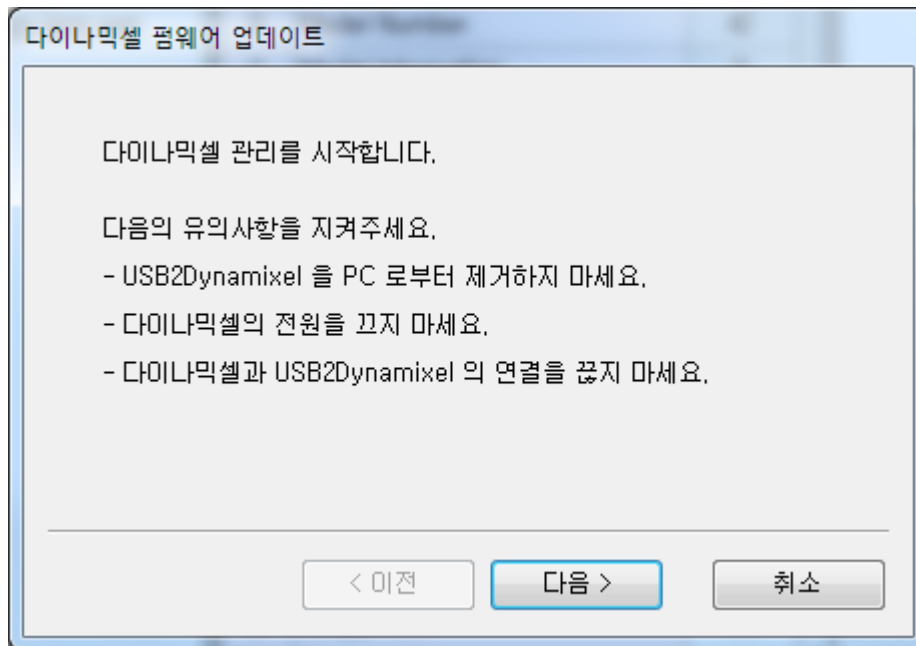
Model	Relationship between goal torque and current
H54 Series	$Current (mA) = goal\ torque\ vlaue \times \frac{33000}{2048}$
H42 Series	$Current (mA) = goal\ torque\ vlaue \times \frac{8250}{2048}$

xi. Dynamixel Pro 의 Firmware Update 를 수행해 봅시다.

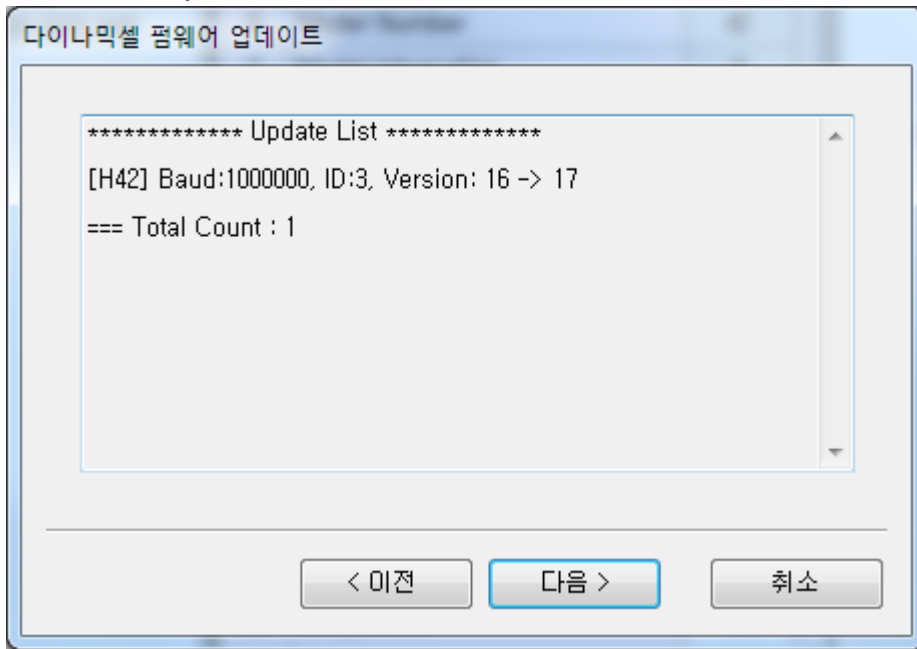
- Firmware 는 Dynamixel 에 설치되어 있는 프로그램을 말하며, Dynamixel 을 제어하는 역할을 담당하고 있습니다.
- Dynamixel Wizard 는 인터넷을 통해 새 버전의 Firmware 를 자동으로 감지하여 항상 최신으로 유지시킵니다.
- 연결되어 있는 Dynamixel Pro 의 Firmware 버전보다 새로운 버전의 Firmware 가 배포되면, 아래의 그림과 같이 Dynamixel Pro 를 Dynamixel Wizard 로 검색했을 때, Dynamixel Pro 의 아이콘이 체크되며 Firmware Update 가 가능함을 알립니다.



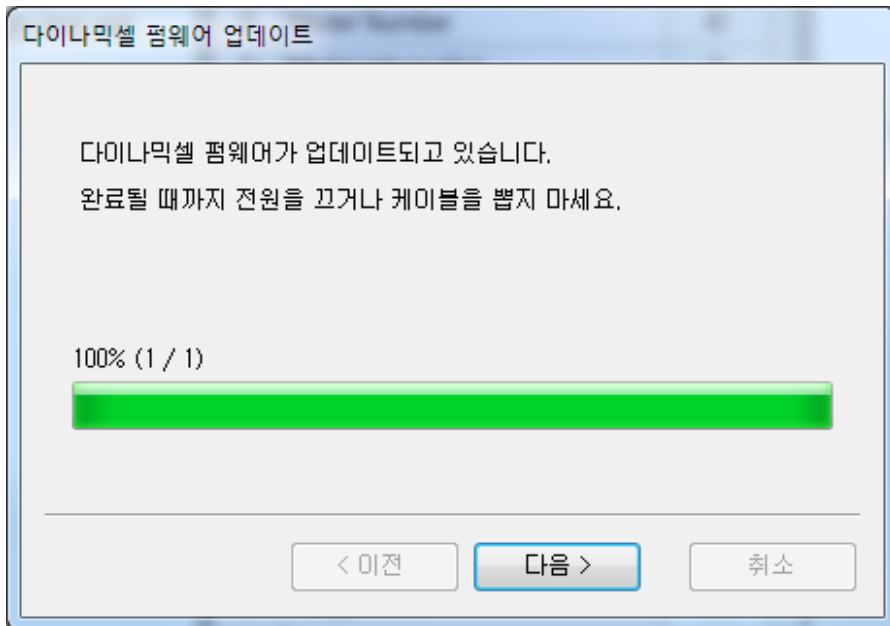
- **Firmware Update 를 할 때에는 반드시 Dynamixel 을 하나만 연결해야 합니다.**
- Firmware Update 를 위해서는 해당 Dynamixel Pro 를 클릭하고 Firmware 업데이트 버튼을 클릭합니다.
- 간단한 도움말과 함께 Firmware 업데이트를 시작합니다.
- 특히, 업데이트 도중 연결이 끊기거나 전원이 꺼지지 않도록 유의해야 합니다.



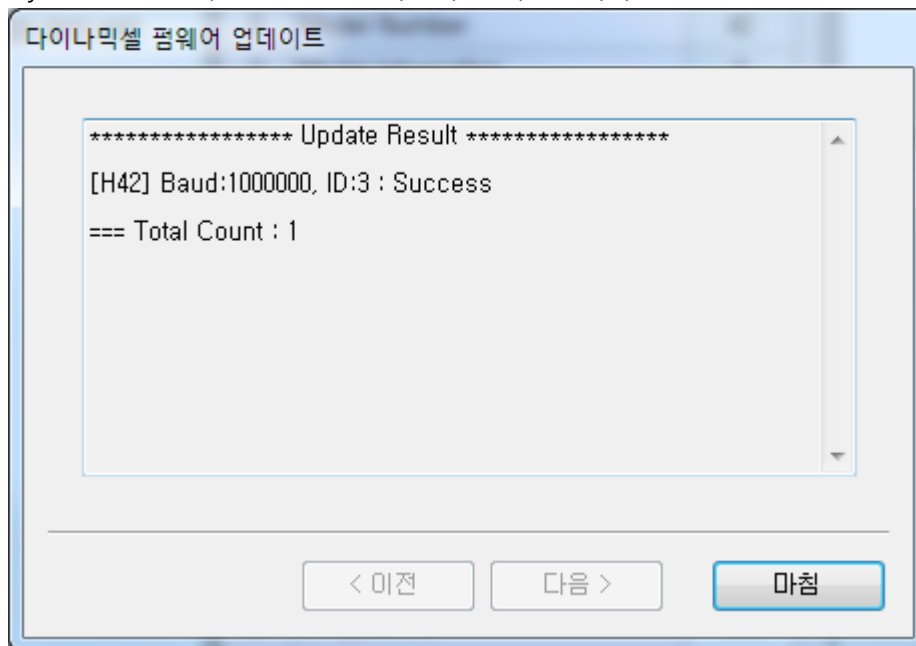
- 현재 연결된 Dynamixel 의 모델명과 Firmware 정보를 확인할 수 있습니다.



- 다음을 누르면 Firmware Update 를 시작합니다. 완료될 때까지 전원이 꺼지거나 케이블이 빠지지 않도록 유의합니다.



- Dynamixel Pro 의 Firmware 설치결과를 확인합니다.

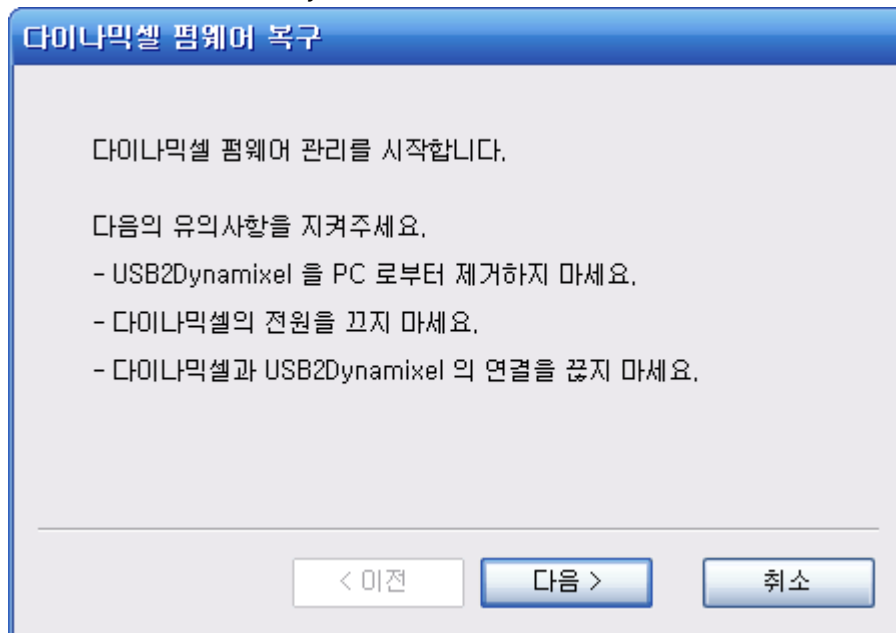


xii. Dynamixel Wizard 를 통한 Dynamixel Pro 의 Firmware 를 복구 해 봅시다.

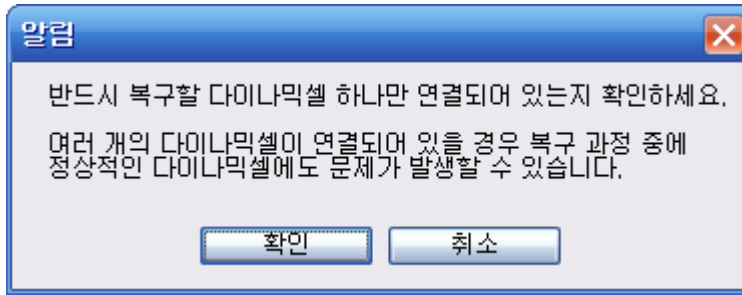
- Dynamixel Wizard 는 Dynamixel Pro 의 Firmware 에 문제가 있는 경우 이를 복구할 수 있습니다.
- **※ Firmware 복구 후에는 모든 값이 초기화 되므로 복구 후 ID 와 Baudrate 값을 다시 확인해야 합니다. USB2Dynamixel 의 모드 위치를 확인해야 하며, 3 핀 케이블은 TTL 로 4 핀 케이블은 RS485 로 스위치를 바꿔주세요.**
- **※ Firmware 복구 시에는 반드시 하나의 Dynamixel Pro 만을 연결해야 하며, 두 개 이상의 Dynamixel Pro 가 데이지체인으로 연결된 상태에서 복구를 시도했을 때에는 오동작이 발생할 수 있습니다.**
- 도구 모음의 Firmware 복구 버튼을 눌러 Dynamixel Firmware 복구 Wizard 를 실행합니다.
- 다른 프로그램에서 USB2Dynamixel 에 할당된 COM Port 가 연결되어 있는 경우에는 해당 버튼이 비활성화 됩니다. Port 연결을 끊고 Firmware 복구 Wizard 를 실행해 주세요.



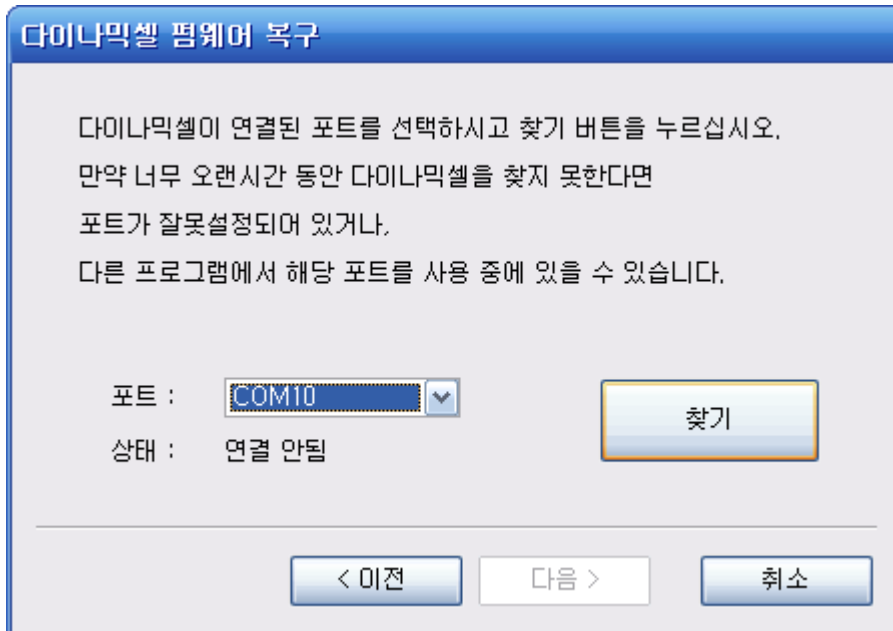
- 간단한 도움말과 함께 Dynamixel Firmware 복구 Wizard 를 시작합니다.



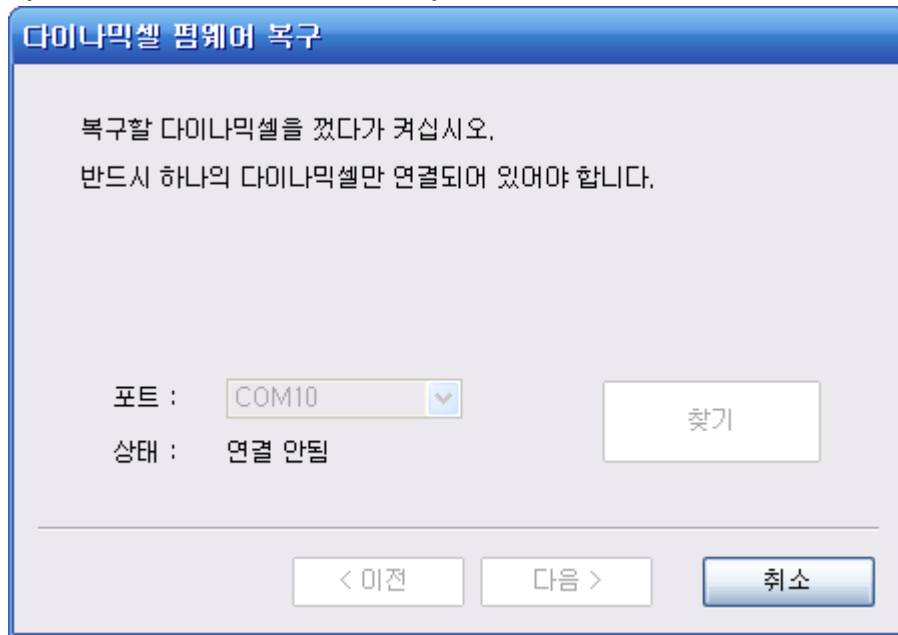
- Dynamixel Firmware 복구 시에는 반드시 한 개의 Dynamixel 만 연결되어 있어야 합니다.



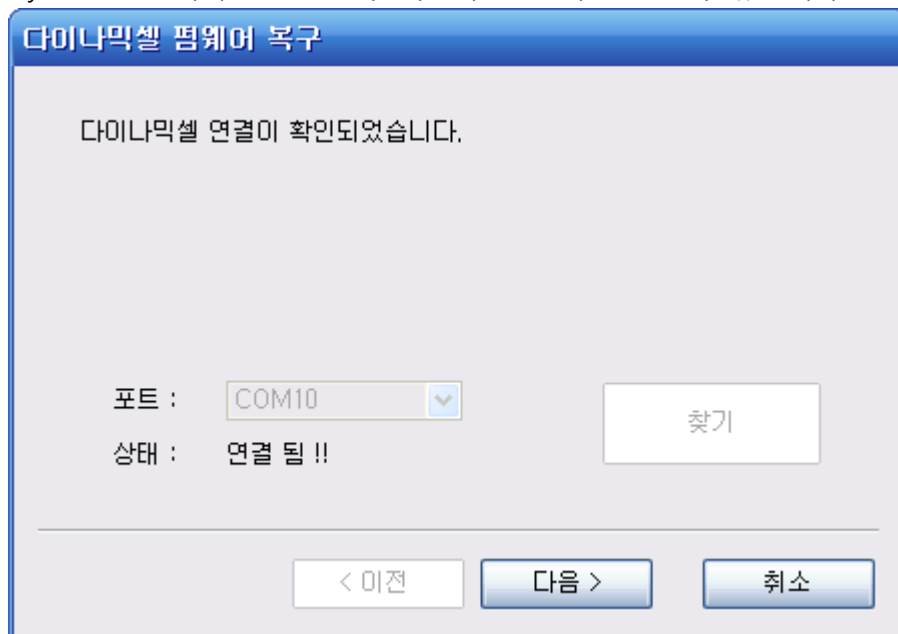
- Firmware 를 인식하지 못하므로 Dynamixel Pro 를 자동 검색할 수 없습니다. 따라서 사용자가 Dynamixel 이 연결된 USB2Dynamixel 의 Port 를 수동으로 설정해 주어야 합니다. Port 가 사용 중이면 Dynamixel 을 인식할 수 없으니 다른 프로그램을 종료하고 진행하십시오.
- USB2Dynamixel 이 연결된 포트를 선택하고 찾기 버튼을 누르십시오.



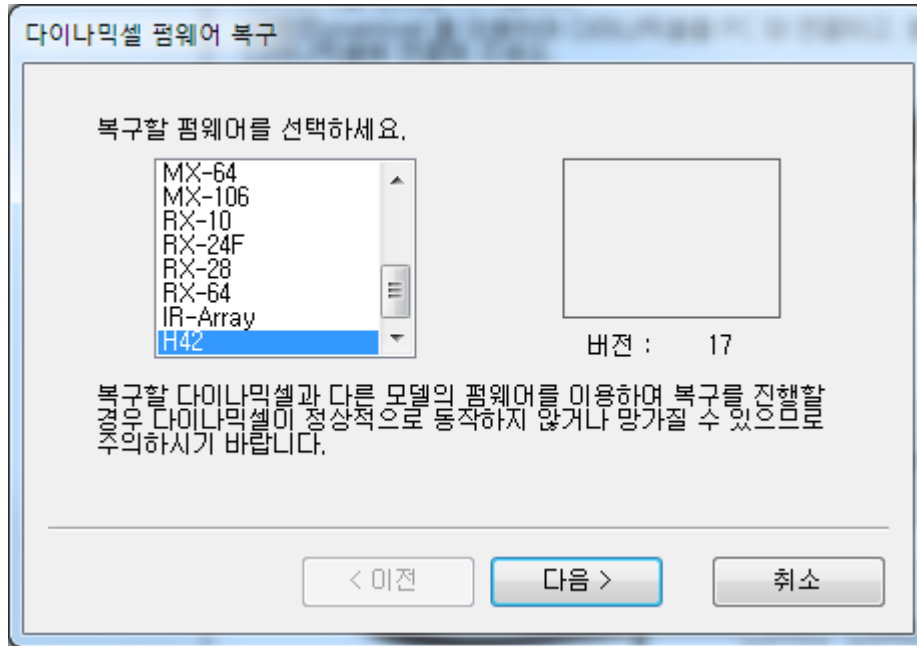
- Dynamixel 을 찾기 위해 복구할 Dynamixel 의 전원을 껐다가 켜십시오.



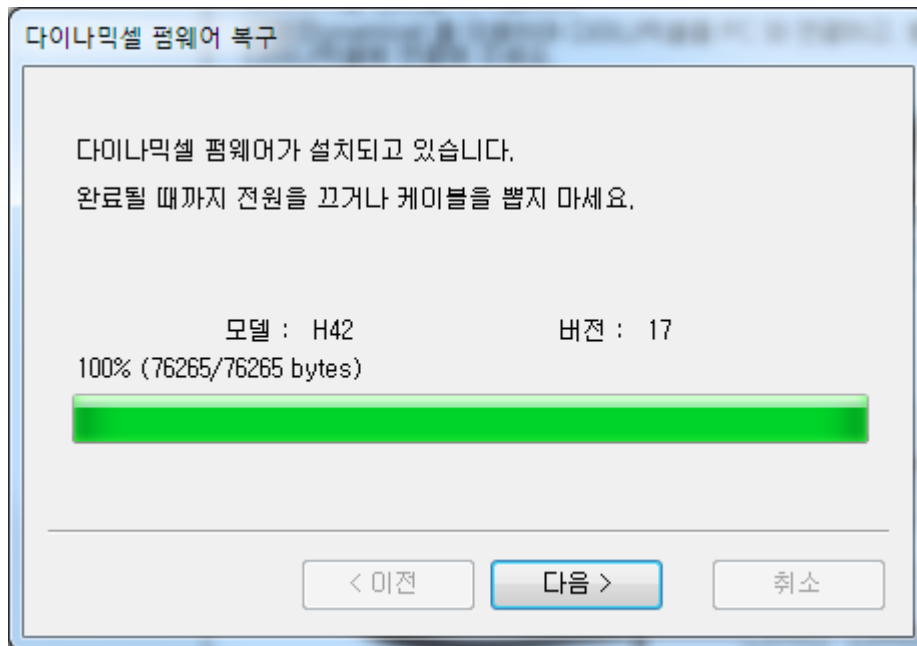
- 만약, 껐다 켜는데도 Dynamixel 이 검색되지 않는다면 몇 번 더 껐다가 켜시길 바랍니다.
- Dynamixel 검색에 성공한 경우 다음과 같은 화면을 볼 수 있습니다.



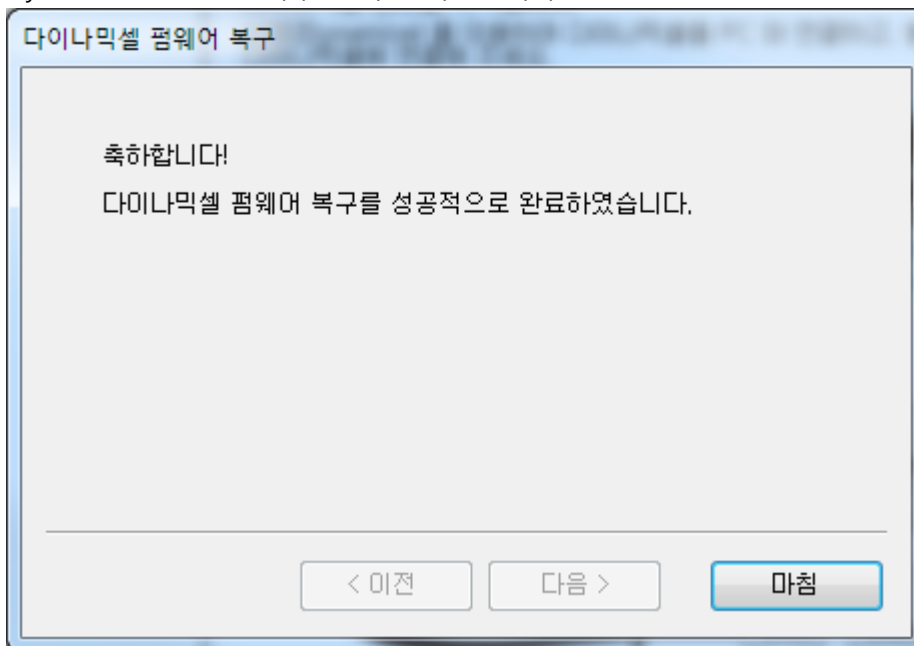
- Dynamixel 을 찾으면 다운로드 할 수 있는 Firmware 정보가 나옵니다. Dynamixel Firmware 복구 모드에서는 모델명을 자동으로 인식할 수 없기 때문에, 사용자가 정확한 모델을 선택해 주어야 합니다. 잘못된 모델을 선택하면 정상적으로 동작하지 않습니다.



- 다음 버튼을 클릭하면 Firmware 복구를 시작합니다. 완료될 때까지 전원이 꺼지거나 케이블이 빠지지 않도록 주의하십시오.



- Dynamixel Firmware 복구 결과를 확인합니다.

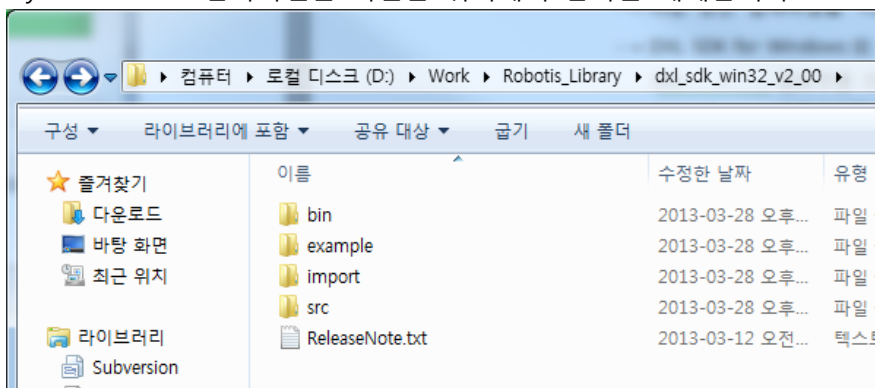


1.2 PC(VS2010)

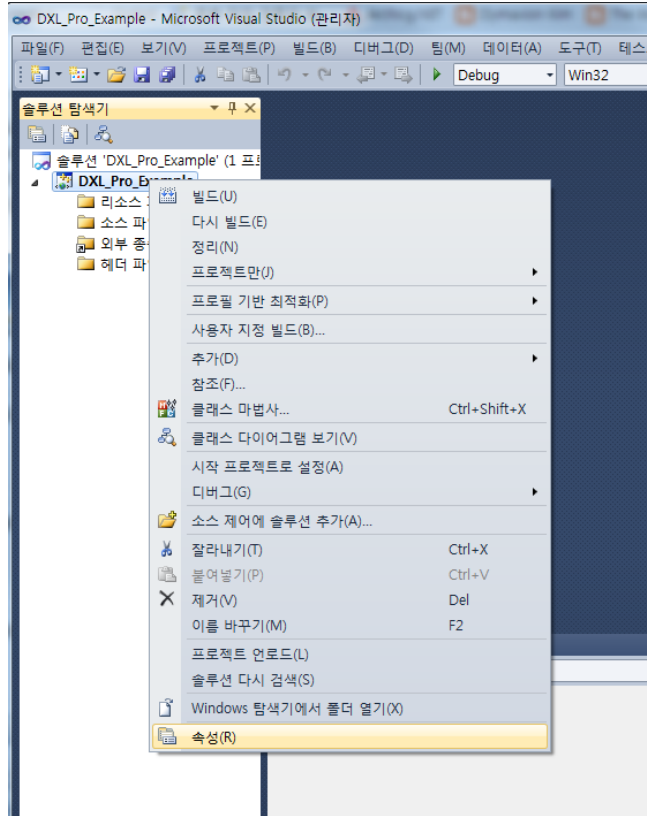
1.2.1 Preparation

i. Dynamixel Pro 를 C 언어로 제어하기 위한 설정을 해봅시다.

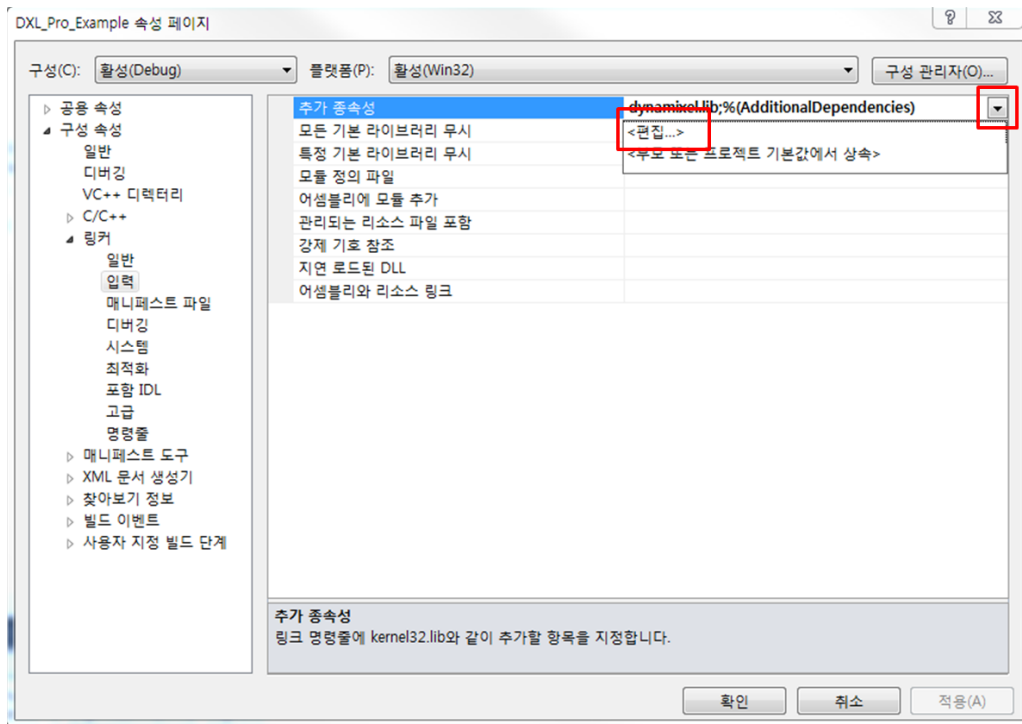
- 앞으로의 Tutorial 을 위해 ID 는 1, Baudrate 는 57600 으로 설정되어 있는 DXL Pro 를 하나 준비하십시오.
- 또한, Tutorial 에 나오는 용어들에 익숙하지 않을 수 있으니, 이전 Chapter 의 Tutorial 을 따라한 뒤 이번 Chapter 의 Tutorial 들을 따라하십시오.
- 먼저 Dynamixel SDK 2.0 for Windows(dx1_sdk_win32_v2_00.zip) 을 동봉된 USB Memory 에서 복사합니다.
- Dynamixel SDK 압축파일을 적절한 위치에서 압축을 해제합니다.



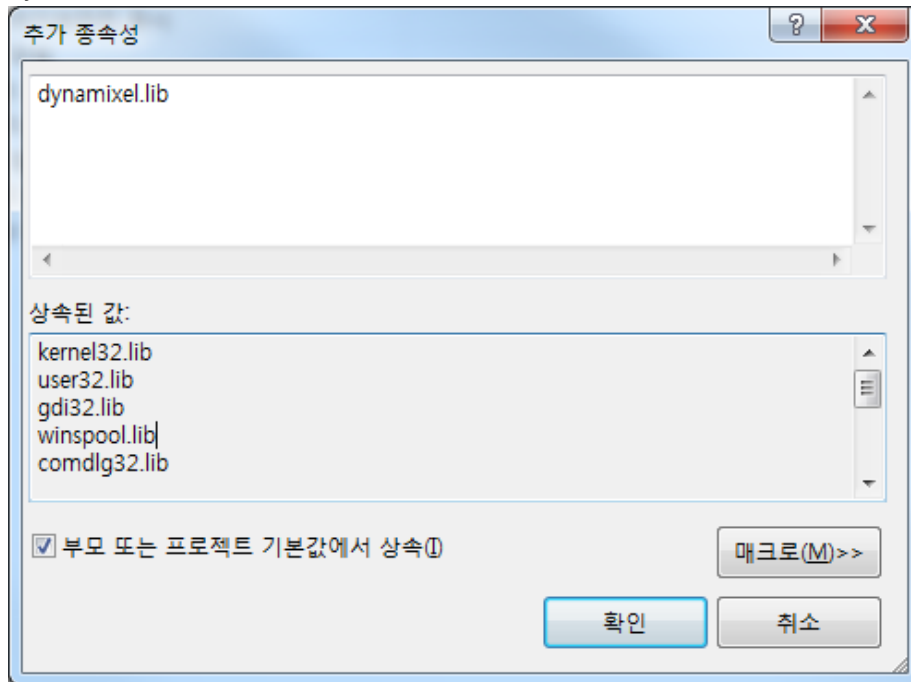
- Dynamixel SDK for Windows 의 폴더 구조는 다음과 같습니다.
 - /bin : 윈도우용 DLL 파일이 들어있습니다.
 - /import : 프로그래밍할때 필요한 파일들이 있습니다.(lib, h)
 - /src : DLL 의 소스가 들어있습니다.
 - /example : 사용자의 프로그래밍 환경에 맞게 Dynamixel 을 제어하는 예제들이 들어있는 곳입니다.
- VS2010 에서 Dynamixel SDK 를 사용하여 프로그래밍을 하기 위해서는 먼저 Dynamixel SDK 를 사용하기 위한 설정을 해야합니다.
- 먼저 VS2010 에서 프로젝트를 생성합니다.
- 프로젝트 폴더에 Dynamixel SDK 의 압축을 풀어놓은 폴더에서 bin/dynamixel.dll, /import/dynamixel.h, /import/dynamixel.lib 파일들을 프로젝트 폴더에 복사해 넣습니다.
- 생성한 프로젝트의 Solution 파일을 실행하고 솔루션 탐색기에서 해당 프로젝트를 오른쪽 마우스 버튼으로 클릭한 뒤 속성페이지에 들어 갑니다.



- 속성 페이지에서 '링커 -> 입력'으로 들어갑니다.
- '추가 종속성'을 클릭한 뒤 오른쪽에 있는 삼각형 버튼을 클릭한 후 편집페이지로 들어갑니다.



- dynamixel.lib 을 입력한 뒤 '확인' 버튼을 눌러 설정을 저장합니다.



- 이로써 Dynamixel SDK 를 사용하여 프로그래밍 하기 위한 준비가 끝났습니다.

ii. C 언어로 USB2Dynamixel 과의 연결 및 연결종료를 해 봅시다.

- 먼저, cpp 파일을 하나 생성하고 아래의 코드를 추가합니다. COM_PORT 와 BAUD_RATE 는 사용하는 USB2Dynamixel 에 할당된 COM Port 와 사용하고 있는 Dynamixel Pro 의 통신속도에 맞추어서 선언하시면 됩니다.

```
#include "dynamixel.h"

#define COM_PORT_NUM 17 //Comport Number of USB2DXL
#define BAUD_RATE_NUM 1 //Baudrate Number of DXL Pro
```

- Baudrate Number 와 통신속도사이의 관계는 아래와 같습니다.

Baudrate Number	Bps(= bit per seconds)
0	2400 bps
1	57600 bps
2	115200 bps
3	1 Mbps
4	2 Mbps
5	3 Mbps
6	4 Mbps
7	4.5 Mbps
8	10.5 Mbps

- 먼저 SerialPort 형의 변수를 선언하고 SerialPort 구조체 안의 모든 값을 0 으로 초기화 합니다.
- 그 뒤에 SerialPort 의 주소값을 저장할 SerialPort 구조체의 포인터 형 변수를 선언하고 먼저 선언한 SerialPort 형 변수의 주소값으로 초기화 합니다.

```
SerialPort sp = {0,0,0,0,0};
SerialPort *Port = &sp;
```

- 다음으로 dxl_initialize 함수와 dxl_terminate 함수를 사용하여 USB2Dynamixel 과의 연결 및 종료를 수행합니다.
- 아래는 프로그램의 전체 Source 입니다..

```
main.cpp
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"

#define COM_PORT_NUM 17 //Comport Number of USB2DXL
#define BAUD_RATE_NUM 1 //Baudrate Number of DXL Pro. 1 is 57600 bps
int main(void)
{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;
```

```
//Open the port of USB2DXL
if(dxl_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )
    printf("Succeed to open USB2Dynamixel!\n");
else
{
    printf( "Failed to open USB2Dynamixel!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}

printf( "Press any key to terminate...\n" );
getch();

//Close the port of USB2DXL
dxl_terminate(Port);

return 0;
}
```

- COMM_RXSUCCESS 는 USB2Dynamixel 을 이용한 Dynamixel Pro 와의 통신에 성공했을 때 return 되는 값입니다.
- Dynamixel SDK 를 통해서 Dynamixel Pro 와 통신을 할 때에는 사용한 함수에서 반환되는 값으로 통신성공여부를 알 수 있습니다.

1.2.2 기본적인 Dynamixel Pro 의 기능들을 사용해 봅시다.

i. Torque 를 On/Off 해 봅시다.

- 1.1 에서 설명했듯이 Dynamixel Pro 의 Control Table 에서 Torque On/Off 를 할 수 있는 Torque Enable 의 주소값은 562 번입니다. 또한 Torque Enable 에션 1Byte 가 할당되어 있습니다.
- 따라서 Control Table 에서 1Byte 의 값만을 변경할 수 있는 `dxl_write_byte` 함수를 통해서 이를 수행할 수 있습니다.
- 아래는 프로그램의 전체 Source 입니다.

```

main.cpp
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"

#define COM_PORT_NUM    17        //Comport Number of USB2DXL
#define BAUD_RATE_NUM   1         //Baudrate Number of DXL Pro. 1 is 57600 bps

#define P_TORQUE_ENABLE 562       //Address of Torque Enable in Control Table
#define ID               1        //ID of DXL Pro you use

int main(void)
{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;

    //Open the port of USB2DXL
    if(dxl_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )
        printf("Succeed to open USB2Dynamixel!!\n");
    else
    {
        printf( "Failed to open USB2Dynamixel!!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    int ErrorStatus;
    int Result;

    //Torque ON
    printf( "Press any key to turn on the torque...\n" );
    _getch();
    Result = dxl_write_byte(Port, ID, P_TORQUE_ENABLE, 1, &ErrorStatus);

    //Torque OFF
    printf( "Press any key to turn off the torque...\n" );
    _getch();
    Result = dxl_write_byte(Port, ID, P_TORQUE_ENABLE, 0, &ErrorStatus);

    //Close the port of USB2DXL
    printf( "Press any key to terminate...\n" );
    _getch();
}

```

```
dxl_terminate(Port);  
  
return 0;  
}
```

- dxl_initialize, dxl_write_byte 등 Dynamixel Pro 와 통신하는 함수들의 return 값은 통신 결과를 나타냅니다.
- 이를 통하여 명령의 성공 여부를 알 수 있습니다.

ii. 이제 C 언어로 Dynamixel Pro 를 작동해 봅시다.

- 1.1 에서 설명했듯이 Dynamixel Pro 의 Control Table 에서 Goal Position 의 주소값은 562 번입니다. 또한 Goal Position 에는 4Byte 가 할당되어 있습니다.
- 따라서 Control Table 에서 4 Byte 의 값을 변경할 수 있는 dxl_write_dword 함수를 통해서 이를 수행할 수 있습니다.
- 아래는 프로그램의 전체 Source 입니다.

```

main.cpp
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"

#define COM_PORT_NUM 17 //Comport Number of USB2DXL
#define BAUD_RATE_NUM 1 //Baudrate Number of DXL Pro. 1 is 57600 bps

#define P_TORQUE_ENABLE 562 //Address of Torque Enable in Control Table
#define P_GOAL_POSITION 596 //Address of Goal Position in Control Table
#define ID 1 //ID of DXL Pro you use

int main(void)
{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;

    //Open the port of USB2DXL
    if(dxl_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )
        printf("Succeed to open USB2Dynamixel!\n");
    else
    {
        printf( "Failed to open USB2Dynamixel!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    int Result, ErrorStatus;

    //Torque ON
    printf( "Torque On...\n" );
    Result = dxl_write_byte(Port, ID, P_TORQUE_ENABLE, 1, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    int Goal_Pos1 = 100000, Goal_Pos2 = -100000;

    //Change the vlaue of goal position
    printf("Press any key to roatate the DXL Pro to position 1\n");

```

```
_getch();
Result = dxl_write_dword(Port, ID, P_GOAL_POSITION, Goal_Pos1, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( " Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}

//Change the vlaue of goal position
printf("Press any key to roatate the DXL Pro to position 2\n");
_getch();
Result = dxl_write_dword(Port, ID, P_GOAL_POSITION, Goal_Pos2, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( " Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}

//Torque OFF
printf( "Press any key to turn off the torque...\n" );
_getch();
Result = dxl_write_dword(Port, ID, P_TORQUE_ENABLE, 0, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( " Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}

//Close the port of USB2DXL
printf( "Press any key to terminate...\n" );
_getch();
dxl_terminate(Port);

return 0;
}
```


iii. C 언어로 ID 가 1 인 Dynamixel Pro 의 ID 를 3 으로 변경해 봅시다.

- 1.1 에서 설명했듯이 Dynamixel Pro 의 Control Table 에서 ID 의 주소값은 7 번입니다. 또한 ID 는 1Byte 가 할당되어 있으므로 dxl_write_byte 로 ID 를 바꿀 수 있습니다.
- 하지만, ID 는 Control Table 에서 Torque On 상태에서는 바꿀 수 없는 EEPROM 영역이므로 ID 를 바꾸기 전에는 반드시 Torque Off 상태로 만들어야 합니다.
- 아래는 프로그램의 전체 Source 입니다.

```

main.cpp
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"

#define COM_PORT_NUM    17    //Comport Number of USB2DXL
#define BAUD_RATE_NUM   1     //Baudrate Number of DXL Pro

#define P_ID            7     //Address of ID in Contorl Table
#define P_TORQUE_ENABLE 562   //Address of Torque Enable in Control Table

#define ID              1     //ID of DXL Pro you use

// Print error bit of status packet
void PrintErrorCode(int ErrorCode)
{
    if(ErrorCode & ERBBIT_VOLTAGE)
        printf("Input voltage error!\n");

    if(ErrorCode & ERBBIT_ANGLE)
        printf("Angle limit error!\n");

    if(ErrorCode & ERBBIT_OVERHEAT)
        printf("Overheat error!\n");

    if(ErrorCode & ERBBIT_RANGE)
        printf("Out of range error!\n");

    if(ErrorCode & ERBBIT_CHECKSUM)
        printf("Checksum error!\n");

    if(ErrorCode & ERBBIT_OVERLOAD)
        printf("Overload error!\n");

    if(ErrorCode & ERBBIT_INSTRUCTION)
        printf("Instruction code error!\n");
}

int main(void)
{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;

    //Open the port of USB2DXL
    if(dxl_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )
        printf("Succeed to open USB2Dynamixel!\n");
}

```

```

else
{
    printf( "Failed to open USB2Dynamixel!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}

int Result, ErrorStatus;

//Torque Off
printf( "Torque Off...\n" );
Result = dxl_write_byte(Port, ID, P_TORQUE_ENABLE, 0, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}

int desired_ID = 3;

//Change the ID of DXL Pro you use
printf("Press any key to change the ID of DXL Pro you use\n");
_getch();
Result = dxl_write_byte(Port, ID, P_ID, desired_ID, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}
else
{
    if(ErrorStatus != 0 )
        PrintErrorCode(ErrorStatus);
    else
        printf("Succeed to chagne the ID of DXL Pro you use!\n");
}

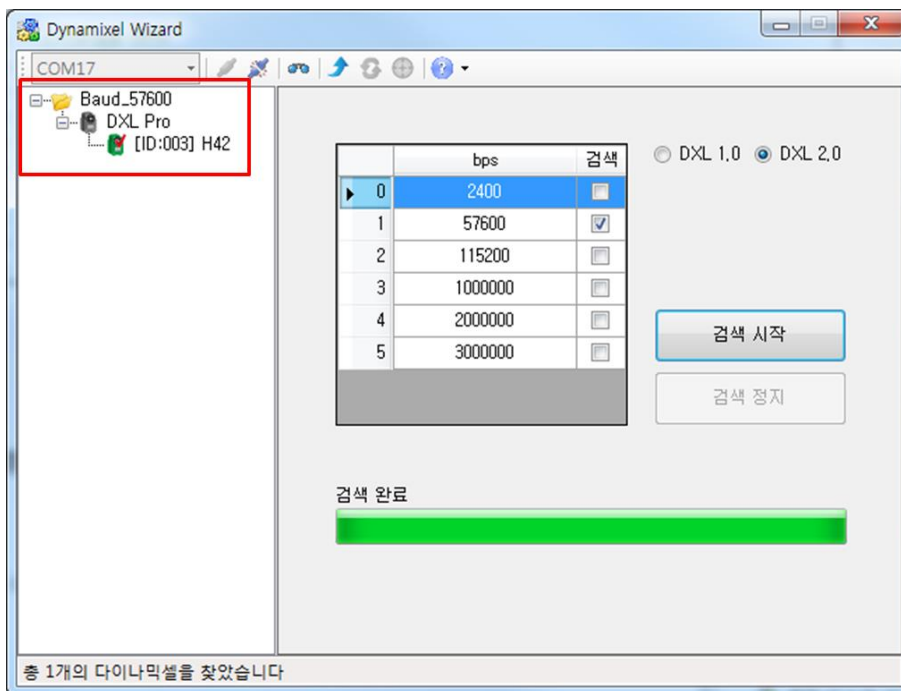
//Close the port of USB2DXL
printf( "Press any key to terminate...\n" );
_getch();
dxl_terminate(Port);

return 0;
}

```

- dxl_write_byte, dxl_write_dword 등 Dynamixel Pro 의 Control Table 에 값을 쓰거나 읽는 API 들의 인자로 들어간 ErrorStatus 는 Dynamixel Pro 에서 보내주는 Error 값을 받아 옵니다.

- Dynamixel Wizard 를 이용하여 변경된 ID 를 확인해 봅니다.



iv. Dynamixel Pro 의 통신속도를 변경해 봅시다.

- 앞의 'C 언어로 ID 가 1 인 Dynamixel Pro 의 ID 를 3 으로 변경해 봅시다.'를 따라한 후라면, Dynamixel Wizard 혹은 앞의 Tutorial 을 수정하여 ID 를 다시 1 로 변경하여 주십시오.
- 1.1 에서 설명했듯이 Dynamixel Pro 의 Control Table 에서 Baudrate(통신속도)의 주소값은 8 번입니다. 또한 Baudrate 는 1Byte 가 할당되어 있으므로 dxl_write_byte 로 Baudrate 를 바꿀 수 있습니다.
- 하지만, ID 와 마찬가지로 Baudrate 는 Control Table 에서 Torque On 상태에서는 바꿀 수 없는 EEPROM 영역이므로 Baudrate 를 바꾸기 전에는 반드시 Torque Off 상태로 만들어야 합니다.
- 아래는 프로그램의 전체 Source 입니다.

```

main.cpp
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"

#define COM_PORT_NUM    17    //Comport Number of USB2DXL
#define BAUD_RATE_NUM  1    //Baudrate Number of DXL Pro

#define P_BAUD_RATE     8    //Address of ID in Contorl Table
#define P_TORQUE_ENABLE 562  //Address of Torque Enable in Control Table

#define ID              1    //ID of DXL Pro you use

// Print error bit of status packet
void PrintErrorCode(int ErrorCode)
{
    if(ErrorCode & ERRCODE_VOLTAGE)
        printf("Input voltage error!\n");

    if(ErrorCode & ERRCODE_ANGLE)
        printf("Angle limit error!\n");

    if(ErrorCode & ERRCODE_OVERHEAT)
        printf("Overheat error!\n");

    if(ErrorCode & ERRCODE_RANGE)
        printf("Out of range error!\n");

    if(ErrorCode & ERRCODE_CHECKSUM)
        printf("Checksum error!\n");

    if(ErrorCode & ERRCODE_OVERLOAD)
        printf("Overload error!\n");

    if(ErrorCode & ERRCODE_INSTRUCTION)
        printf("Instruction code error!\n");
}

int main(void)

```

```

{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;

    //Open the port of USB2DXL
    if(dxl_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )
        printf("Succeed to open USB2Dynamixel!\n");
    else
    {
        printf( "Failed to open USB2Dynamixel!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    int Result, ErrorStatus;

    //Torque Off
    printf( "Torque Off...\n" );
    Result = dxl_write_byte(Port, ID, P_TORQUE_ENABLE, 0, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    int desired_Baudrate = 3; //1 Mbps

    //Change the ID of DXL Pro you use
    printf("Press any key to change the baudrate of DXL Pro you use\n");
    _getch();
    Result = dxl_write_byte(Port, ID, P_BAUD_RATE, 3, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }
    else
    {
        if(ErrorStatus != 0 )
            PrintErrorCode(ErrorStatus);
        else
            printf("Succeed to chage the baudrate of DXL Pro you use!\n");
    }

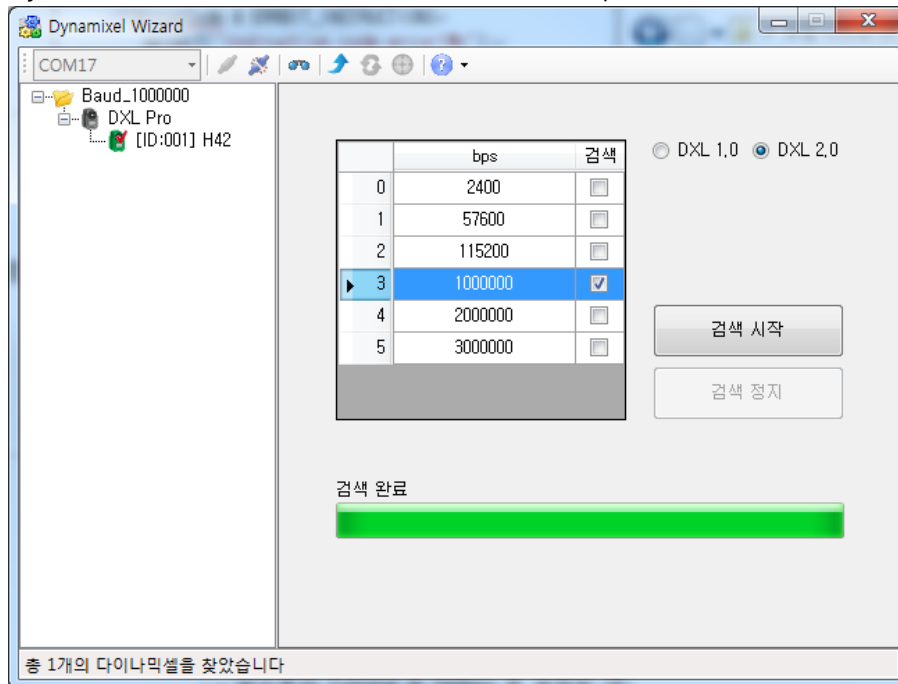
    //Close the port of USB2DXL
    printf( "Press any key to terminate...\n" );
    _getch();
    dxl_terminate(Port);

    return 0;
}

```

```
}
```

- Dynamixel Wizard 를 이용하여 Baudrate 가 1Mbps 로 바뀌었는지를 확인합니다.



- 바뀌지 않았다면 Dynamixel Pro 가 Torque Off 상태가 되어 있는 지를 확인합니다.
- 앞으로의 Tutorial 은 1Mbps 로 설정된 Dynamixel Pro 로 진행됩니다.

v. Dynamixel Pro 의 LED 를 On/Off 해 봅시다.

- Baudrate 가 1Mbps 로 변경되었으므로 아래와 같이 코드를 수정합니다.

```
#define COM_PORT_NUM 17 //Comport Number of USB2DXL
#define BAUD_RATE_NUM 3 //Baudrate Number of DXL Pro
```

- 1.1 에서 설명했듯이 Dynamixel Pro 의 Control Table 에서 LED 관련 주소값은 총 3 개이며 각각 1 Byte 가 할당되어 있습니다. 빨간색은 563 번, 초록색은 564 번, 파란색은 565 번이며, 각각 1Byte 가 할당되어 있으므로 dxl_write_byte 함수를 통해서 각각을 제어할 수 있습니다.
- 1Byte 가 할당되어 있기 때문에 0~255 의 값으로 설정할 수 있으며, 값이 클수록 LED 의 밝기가 밝아집니다.
- 아래는 프로그램의 전체 Source 입니다.

```
main.cpp
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"

#define COM_PORT_NUM 17 //Comport Number of USB2DXL
#define BAUD_RATE_NUM 3 //Baudrate Number of DXL Pro

#define P_LED_RED 563 //Address of LED_RED in Contorl Table
#define P_LED_GREEN 564 //Address of LED_RED in Contorl Table
#define P_LED_BLUE 565 //Address of LED_RED in Contorl Table

#define ID 1 //ID of DXL Pro you use

// Print error bit of status packet
void PrintErrorCode(int ErrorCode)
{
    if(ErrorCode & ERRBIT_VOLTAGE)
        printf("Input voltage error!\Wn");

    if(ErrorCode & ERRBIT_ANGLE)
        printf("Angle limit error!\Wn");

    if(ErrorCode & ERRBIT_OVERHEAT)
        printf("Overheat error!\Wn");

    if(ErrorCode & ERRBIT_RANGE)
        printf("Out of range error!\Wn");

    if(ErrorCode & ERRBIT_CHECKSUM)
        printf("Checksum error!\Wn");

    if(ErrorCode & ERRBIT_OVERLOAD)
        printf("Overload error!\Wn");

    if(ErrorCode & ERRBIT_INSTRUCTION)
        printf("Instruction code error!\Wn");
}
```

```

int main(void)
{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;

    //Open the port of USB2DXL
    if(dxl_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )
        printf("Succeed to open USB2Dynamixel!\n");
    else
    {
        printf( "Failed to open USB2Dynamixel!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    int Result, ErrorStatus;

    //Turn off the LED
    Result = dxl_write_byte(Port, ID, P_LED_RED, 0, &ErrorStatus);
    Result = dxl_write_byte(Port, ID, P_LED_GREEN, 0, &ErrorStatus);
    Result = dxl_write_byte(Port, ID, P_LED_BLUE, 0, &ErrorStatus);

    //Turn on and change the color of LED in DXL Pro
    printf("Press any key to change the color of LED in DXL Pro\n");
    _getch();
    Result = dxl_write_byte(Port, ID, P_LED_RED, 255, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }
    else
    {
        if(ErrorStatus != 0 )
            PrintErrorCode(ErrorStatus);
        else
            printf("Succeed to change the color of LED in DXL Pro!\n");
    }

    printf("Press any key to change the color of LED in DXL Pro\n");
    _getch();
    Result = dxl_write_byte(Port, ID, P_LED_RED, 0, &ErrorStatus);
    Result = dxl_write_byte(Port, ID, P_LED_GREEN, 255, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }
    else
    {

```



```

        if(ErrorStatus != 0 )
            PrintErrorCode(ErrorStatus);
        else
            printf("Succeed to change the color of LED in DXL Pro!\n");
    }

    printf("Press any key to change the color of LED in DXL Pro\n");
    _getch();
    Result = dxl_write_byte(Port, ID, P_LED_GREEN, 0, &ErrorStatus);
    Result = dxl_write_byte(Port, ID, P_LED_BLUE, 255, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }
    else
    {
        if(ErrorStatus != 0 )
            PrintErrorCode(ErrorStatus);
        else
            printf("Succeed to change the color of LED in DXL Pro!\n");
    }

    //Close the port of USB2DXL
    printf( "Press any key to terminate...\n" );
    _getch();
    dxl_terminate(Port);
    return 0;
}

```

- LED의 밝기를 조절해 가면서 여러 번 실습해 봅니다.

vi. Position P gain 값을 변경한 후의 움직임을 확인해 봅시다.

- Dynamixel Pro 는 기본적으로 PID 제어 방식으로 제어가 됩니다. 따라서 Gain 값에 따라 움직임을 달라집니다.
- Dynamixel Pro 의 Control Table 에서 찾아보면, Position_P_Gain 의 주소값은 594 번이며, 2Byte 가 할당되어있음을 알 수 있습니다.
- 따라서, Control Table 에서 2 Byte 값을 한번에 변경할 수 있는 dxl_write_word 함수로 이를 수행할 수 있습니다.
- 아래는 프로그램의 전체 Source 입니다.

```
main.cpp
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"

#define COM_PORT_NUM      17      //Comport Number of USB2DXL
#define BAUD_RATE_NUM     3       //Baudrate Number of DXL Pro
#define P_TORQUE_ENABLE   562     //Address of Torque Enable in Control Table
#define P_POSITION_P_GAIN 594     //Address of Position P Gain in Contorl Table
#define P_GOAL_POSITION   596     //Address of Goal Position in Contorl Table

#define ID                 1      //ID of DXL Pro you use

// Print error bit of status packet
void PrintErrorCode(int ErrorCode)
{
    if(ErrorCode & ERRCODE_VOLTAGE)
        printf("Input voltage error!\n");

    if(ErrorCode & ERRCODE_ANGLE)
        printf("Angle limit error!\n");

    if(ErrorCode & ERRCODE_OVERHEAT)
        printf("Overheat error!\n");

    if(ErrorCode & ERRCODE_RANGE)
        printf("Out of range error!\n");

    if(ErrorCode & ERRCODE_CHECKSUM)
        printf("Checksum error!\n");

    if(ErrorCode & ERRCODE_OVERLOAD)
        printf("Overload error!\n");

    if(ErrorCode & ERRCODE_INSTRUCTION)
        printf("Instruction code error!\n");
}

int main(void)
{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;
```

```

//Open the port of USB2DXL
if(dxl_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )
    printf("Succeed to open USB2Dynamixel!\n");
else
{
    printf( "Failed to open USB2Dynamixel!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}

int Result, ErrorStatus;

//Torque on
printf( "Torque on...\n" );
Result = dxl_write_byte(Port, ID, P_TORQUE_ENABLE, 1, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}

//change the position p gain
printf("Press any key to change the position p gain\n");
_getch();
Result = dxl_write_word(Port, ID, P_POSITION_P_GAIN, 8, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}
else
{
    if(ErrorStatus != 0 )
        PrintErrorCode(ErrorStatus);
}

//change the goal position value
printf("Press any key to change the goal position\n");
_getch();
Result = dxl_write_dword(Port, ID, P_GOAL_POSITION, 100000, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}
else
{
    if(ErrorStatus != 0 )
        PrintErrorCode(ErrorStatus);
}

```

```

}

//change the position p gain
printf("Press any key to change the position p gain\n");
_getch();
Result = dxl_write_word(Port, ID, P_POSITION_P_GAIN, 256, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}
else
{
    if(ErrorStatus != 0 )
        PrintErrorCode(ErrorStatus);
}

//change the goal position value
printf("Press any key to change the goal position\n");
_getch();
Result = dxl_write_dword(Port, ID, P_GOAL_POSITION, -100000, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}
else
{
    if(ErrorStatus != 0 )
        PrintErrorCode(ErrorStatus);
}

//Close the port of USB2DXL
printf( "Press any key to terminate...\n" );
_getch();
dxl_terminate(Port);
return 0;
}

```

- P Gain 을 조절함에 따라 Dynamixel Pro 의 움직이는 속도가 달라짐을 확인합니다.

vii. 천천히 혹은 빠르게 움직이도록 해봅시다.

- 관절모드의 Dynamixel Pro 의 속도를 조절하기 위해서는 바퀴모드와 마찬가지로 Goal Velocity 를 조절하면 됩니다.
- Dynamixel Pro 의 Goal Position 과 Goal Velocity 를 바꾸는 방법은 두 가지가 있습니다.
- 첫째는 dxl_write_dword 함수를 두 번 사용하여 바꾸는 방법이고 둘째는 dxl_write 함수를 사용하여 한번의 통신으로 Goal Position 과 Goal Velocity 를 바꾸는 방법 입니다.
- 지금까지의 Tutorial 로 dxl_write_byte, dxl_write_word, dxl_write_dword 를 이용하여 Contol Table 에서 1 Byte, 2 Byte, 4 Byte 에 해당하는 값을 바꾸는 법은 익숙해 졌을 테니 두번째 방법으로 이를 구현해 보겠습니다.
- Goal Position 은 4Byte, Goal Velocity 도 4Byte 입니다.
- 따라서, Goal Position 과 Goal Velocity 를 한번에 바꾸기 위해서는 8Byte 의 Data 를 보내야 하며, 이때의 Data 는 dxl_write_byte, dxl_write_word, dxl_write_dword 등과 다르게 사용자가 직접 구성하여야 합니다.
- Data 는 아래와 같이 만들 수 있습니다.

```
int position, velocity;
position = 100000;
velocity = 10000;

//Make a tx data
unsigned char data[8];
data[0] = DXL_LOBYTE(DXL_LOWORD(position));
data[1] = DXL_HIBYTE(DXL_LOWORD(position));
data[2] = DXL_LOBYTE(DXL_HIWORD(position));
data[3] = DXL_HIBYTE(DXL_HIWORD(position));
data[4] = DXL_LOBYTE(DXL_LOWORD(velocity));
data[5] = DXL_HIBYTE(DXL_LOWORD(velocity));
data[6] = DXL_LOBYTE(DXL_HIWORD(velocity));
data[7] = DXL_HIBYTE(DXL_HIWORD(velocity));
```

- DXL_LOWORD, DXL_HIWORD, DXL_LOBYTE, DXL_HIBYTE 들을 이용하여 쓰고자 하는 위치값과 속도값을 4Byte 로 나누어서 unsigned char 형의 배열에 순서대로 대입합니다.
- 그 뒤에 dxl_write 함수를 이용하여 위에서 구성한 Data 를 Dynamixel Pro 에 보냅니다.

```
dxl_write(Port, ID, P_GOAL_POSITION, 8, data, &ErrorStatus);
```

- Goal Position 과 Goal Velocity 를 바꾸기 위해서는 dxl_write 함수를 이용하여 Goal Position 의 주소부터 8 Byte 를 바꿔야 하므로, 주소값에는 Goal Position, 길이는 8 을 넣었습니다.
- 아래는 프로그램의 전체 Source 입니다.

```
main.cpp
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"
```

```

#define COM_PORT_NUM      17      //Comport Number of USB2DXL
#define BAUD_RATE_NUM     3       //Baudrate Number of DXL Pro
#define P_TORQUE_ENABLE   562     //Address of Torque Enable in Control Table
#define P_GOAL_POSITION   596     //Address of Goal Position in Control Table

#define ID                1       //ID of DXL Pro you use

// Print error bit of status packet
void PrintErrorCode( int ErrorCode)
{
    if(ErrorCode & ERRCODE_VOLTAGE)
        printf("Input voltage error!\n");

    if(ErrorCode & ERRCODE_ANGLE)
        printf("Angle limit error!\n");

    if(ErrorCode & ERRCODE_OVERHEAT)
        printf("Overheat error!\n");

    if(ErrorCode & ERRCODE_RANGE)
        printf("Out of range error!\n");

    if(ErrorCode & ERRCODE_CHECKSUM)
        printf("Checksum error!\n");

    if(ErrorCode & ERRCODE_OVERLOAD)
        printf("Overload error!\n");

    if(ErrorCode & ERRCODE_INSTRUCTION)
        printf("Instruction code error!\n");
}

int main(void)
{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;

    //Open the port of USB2DXL
    if(dxl_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )
        printf("Succeed to open USB2Dynamixel!\n");
    else
    {
        printf( "Failed to open USB2Dynamixel!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    int Result, ErrorStatus;

    //Torque on
    printf( "Torque on...\n" );
    Result = dxl_write_byte(Port, ID, P_TORQUE_ENABLE, 1, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {

```

```

        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    int position, velocity;
    position = 100000;
    velocity = 10000;

    //Make a tx data
    unsigned char data[8];
    data[0] = DXL_LOBYTE(DXL_LOWORD(position));
    data[1] = DXL_HI BYTE(DXL_LOWORD(position));
    data[2] = DXL_LOBYTE(DXL_HIWORD(position));
    data[3] = DXL_HI BYTE(DXL_HIWORD(position));
    data[4] = DXL_LOBYTE(DXL_LOWORD(velocity));
    data[5] = DXL_HI BYTE(DXL_LOWORD(velocity));
    data[6] = DXL_LOBYTE(DXL_HIWORD(velocity));
    data[7] = DXL_HI BYTE(DXL_HIWORD(velocity));

    //change the position value and moving speed
    printf( "Press any key to change the position value and moving speed...\n" );
    _getch();
    Result = dxl_write(Port, ID, P_GOAL_POSITION, 8, data, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }
    else
    {
        if(ErrorStatus != 0 )
            PrintErrorCode(ErrorStatus);
    }
}

position = -100000;
velocity = 2000;
//Make a tx data
data[0] = DXL_LOBYTE(DXL_LOWORD(position));
data[1] = DXL_HI BYTE(DXL_LOWORD(position));
data[2] = DXL_LOBYTE(DXL_HIWORD(position));
data[3] = DXL_HI BYTE(DXL_HIWORD(position));
data[4] = DXL_LOBYTE(DXL_LOWORD(velocity));
data[5] = DXL_HI BYTE(DXL_LOWORD(velocity));
data[6] = DXL_LOBYTE(DXL_HIWORD(velocity));
data[7] = DXL_HI BYTE(DXL_HIWORD(velocity));

//change the position value and moving speed
printf( "Press any key to change the position value and moving speed...\n" );
_getch();
Result = dxl_write(Port, ID, P_GOAL_POSITION, 8, data, &ErrorStatus);
if( Result != COMM_RXSUCCESS )

```

```
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}
else
{
    if(ErrorStatus != 0 )
        PrintErrorCode(ErrorStatus);
}

//Close the port of USB2DXL
printf( "Press any key to terminate...\n" );
_getch();
dxl_terminate(Port);
return 0;
}
```

- 처음 위치에는 빠른속도로 도달하고, 다음 위치에는 느린속도로 도달함을 확인할 수 있습니다.

viii. Dynamixel Pro 의 내부 온도를 확인해 봅시다.

- 지금까지, Dynamixel Pro 에 Write 명령을 쓰는 법을 익혔습니다.
- 지금부터는 Dynamixel Pro 로부터 정보를 받아오기 위한 Read 명령에 대한 예제에 관한 Tutorial 입니다.
- Dynamixel Pro 의 Control Table 에서 현재 온도를 나타내는 Present temperature 의 주소는 625 번이며 1 Byte 가 할당되어 있습니다.
- 따라서 dxl_read_byte 함수로 현재의 Dynamixel Pro 의 내부 온도를 알 수 있습니다.
- 아래는 프로그램의 전체 Source 입니다.

```

main.cpp
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"

#define COM_PORT_NUM          17    //Comport Number of USB2DXL
#define BAUD_RATE_NUM        3     //Baudrate Number of DXL Pro
#define P_PRESENT_TEMPERATURE 625   //Address of Present Temperature in Contorl Table

#define ID                    1     //ID of DXL Pro you use

// Print error bit of status packet
void PrintErrorCode(int ErrorCode)
{
    if(ErrorCode & ERRBIT_VOLTAGE)
        printf("Input voltage error!\n");

    if(ErrorCode & ERRBIT_ANGLE)
        printf("Angle limit error!\n");

    if(ErrorCode & ERRBIT_OVERHEAT)
        printf("Overheat error!\n");

    if(ErrorCode & ERRBIT_RANGE)
        printf("Out of range error!\n");

    if(ErrorCode & ERRBIT_CHECKSUM)
        printf("Checksum error!\n");

    if(ErrorCode & ERRBIT_OVERLOAD)
        printf("Overload error!\n");

    if(ErrorCode & ERRBIT_INSTRUCTION)
        printf("Instruction code error!\n");
}

int main(void)
{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;

    //Open the port of USB2DXL
    if(dxl_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )

```

```

        printf("Succeed to open USB2Dynamixel!\n");
    else
    {
        printf( "Failed to open USB2Dynamixel!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    int Result, ErrorStatus;

    printf("Press any key to terminate...\n" );
    printf("\n");
    int temp;
    while(true)
    {
        if(_kbhit())
            break;

        //Read the present temperature
        Result = dxl_read_byte(Port, ID,
                               P_PRESENT_TEMPERATURE, &temp, &ErrorStatus);
        if( Result != COMM_RXSUCCESS )
        {
            printf( "Failed to write!\n" );
            printf( "Press any key to terminate...\n" );
            _getch();
            return 0;
        }
        else
        {
            PrintErrorCode(ErrorStatus);
        }

        printf("Wr ");
        printf("current temperature : %d", temp);
    }

    //Close the port of USB2DXL
    dxl_terminate(Port);
    return 0;
}

```

- 프로그램을 실행하면 현재의 내부 온도가 표시됩니다.

ix. Dynamixel Pro 의 현재 위치 값을 확인해 봅시다.

- Dynamixel Pro 의 Control Table 에서 현재 위치를 나타내는 Present Position 의 주소는 611 번이며 4 Byte 가 할당되어 있습니다.
- 따라서 dxl_read_dword 함수로 현재의 Dynamixel Pro 의 현재 위치를 알 수 있습니다.
- 아래는 프로그램의 전체 Source 입니다.

```

main.cpp
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"

#define COM_PORT_NUM      17    //Comport Number of USB2DXL
#define BAUD_RATE_NUM    3     //Baudrate Number of DXL Pro
#define P_PRESENT_POSITION 611 //Address of Present Position in Contorl Table

#define ID                1     //ID of DXL Pro you use

// Print error bit of status packet
void PrintErrorCode(int ErrorCode)
{
    if(ErrorCode & ERBIT_VOLTAGE)
        printf("Input voltage error!\n");

    if(ErrorCode & ERBIT_ANGLE)
        printf("Angle limit error!\n");

    if(ErrorCode & ERBIT_OVERHEAT)
        printf("Overheat error!\n");

    if(ErrorCode & ERBIT_RANGE)
        printf("Out of range error!\n");

    if(ErrorCode & ERBIT_CHECKSUM)
        printf("Checksum error!\n");

    if(ErrorCode & ERBIT_OVERLOAD)
        printf("Overload error!\n");

    if(ErrorCode & ERBIT_INSTRUCTION)
        printf("Instruction code error!\n");
}

int main(void)
{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;

    //Open the port of USB2DXL
    if(dxl_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )
        printf("Succeed to open USB2Dynamixel!\n");
    else
    {
        printf( "Failed to open USB2Dynamixel!\n" );
        printf( "Press any key to terminate...\n" );
    }
}

```

```

        _getch();
        return 0;
    }

    int Result, ErrorStatus;

    printf( "Press any key to terminate...\Wn" );
    printf( "\Wn" );
    int temp;
    while(true)
    {
        if(_kbhit())
            break;

        //Read the present position
        Result = dxl_read_dword(Port, ID, P_PRESENT_POSITION, (unsigned*)&temp, &ErrorStatus);
        if( Result != COMM_RXSUCCESS )
        {
            printf( "Failed to write!\Wn" );
            printf( "Press any key to terminate...\Wn" );
            _getch();
            return 0;
        }
        else
        {
            PrintErrorCode(ErrorStatus);
        }

        printf("\Wr");
        printf("present position : %d", temp);
    }

    //Close the port of USB2DXL
    dxl_terminate(Port);
    return 0;
}

```

- 프로그램을 실행하면 현재 각도가 표시됩니다.

x. 바퀴 모드로 변경한 후 속도를 변경해봅시다.

- Dynamixel Pro 의 Control Table 에서 Operating Mode 의 주소값은 11 번이며, 1Byte 가 할당되어 있습니다. 이 Operating Mode 의 값을 바꿈으로써 Dynamixel Pro 를 Joint Mode, Wheel Mode, Torque Mode 로 변경할 수 있습니다,
- Operating Mode 의 경우 1Byte 가 할당되어 있기 때문에 dxl_write_byte 를 사용하여 바꿀 수 있고, EEPROM 영역이기 때문에 반드시 Torque 가 Off 된 상태에서 바꾸어야 합니다.
- 또한, 1.1 에서 확인 했듯이 Wheel Mode 에서도 Dynamixel Pro 를 동작 시키기 위해서는 Torque On 을 해야하며, Goal_Velocity 값을 변경함으로써 Dynamixel Pro 의 회전속도를 제어할 수 있습니다.
- 아래는 프로그램의 전체 Source 입니다.

```
main.cpp
#include <Windows.h>
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"

#define COM_PORT_NUM          17    //Comport Number of USB2DXL
#define BAUD_RATE_NUM        3     //Baudrate Number of DXL Pro

#define P_OPERATING_MODE     11    //Address of Operation Mode in Control Table
#define P_TORQUE_ENABLE     562   //Address of Torque Enable in Control Table
#define P_GOAL_VELOCITY     600   //Address of Goal Velocity in Control Table

#define ID                    1    //ID of DXL Pro you use

// Print error bit of status packet
void PrintErrorCode(int ErrorCode)
{
    if(ErrorCode & ERRBIT_VOLTAGE)
        printf("Input voltage error!\n");

    if(ErrorCode & ERRBIT_ANGLE)
        printf("Angle limit error!\n");

    if(ErrorCode & ERRBIT_OVERHEAT)
        printf("Overheat error!\n");

    if(ErrorCode & ERRBIT_RANGE)
        printf("Out of range error!\n");

    if(ErrorCode & ERRBIT_CHECKSUM)
        printf("Checksum error!\n");

    if(ErrorCode & ERRBIT_OVERLOAD)
        printf("Overload error!\n");

    if(ErrorCode & ERRBIT_INSTRUCTION)
        printf("Instruction code error!\n");
}
```

```

int main(void)
{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;

    //Open the port of USB2DXL
    if(dx1_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )
        printf("Succeed to open USB2Dynamixel!\n");
    else
    {
        printf( "Failed to open USB2Dynamixel!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    int Result, ErrorStatus;

    //Torque Off
    printf( "Torque Off...\n" );
    Result = dx1_write_byte(Port, ID, P_TORQUE_ENABLE, 0, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    printf("Press any key to change the operating mode...\n");
    _getch();
    Result = dx1_write_byte(Port, ID, P_OPERATING_MODE, 1, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }
    else
    {
        if(ErrorStatus != 0 )
            PrintErrorCode(ErrorStatus);
        else
            printf("Succeed to chage the operationg mode!\n");
    }

    //Torque On
    printf( "Torque On...\n" );
    Result = dx1_write_byte(Port, ID, P_TORQUE_ENABLE, 1, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
    }
}

```

```
        return 0;
    }

    //Change the Goal Velocity
    printf("Press any key to change goal velocity...\n");
    _getch();
    Result = dxl_write_dword(Port, ID, P_GOAL_VELOCITY, 5000, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }
    else
    {
        if(ErrorStatus != 0 )
            PrintErrorCode(ErrorStatus);
        else
            printf("Succeed to chage the operationg mode!\n");
    }

    //Close the port of USB2DXL
    printf( "Press any key to terminate...\n" );
    _getch();

    //Torque Off
    printf( "Torque Off...\n" );
    Result = dxl_write_byte(Port, ID, P_TORQUE_ENABLE, 0, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    dxl_terminate(Port);
    return 0;
}
```

- xi. 현재 위치와 현재 속도를 한번에 확인 해 봅시다.
 - Dynamixel Pro 의 현재 속도와 현재 위치를 확인하는 방법은 두가지가 있습니다.
 - 첫째는 dxl_read_dword 함수를 두 번 사용하여 바꾸는 방법이고 둘째는 dxl_read 함수를 사용하여 한번의 통신으로 Present Position 과 Present Velocity 를 바꾸는 방법 입니다.
 - 지금까지의 Tutorial 로 dxl_read_byte, dxl_read_word, dxl_read_dword 를 이용하여 Control Table 에서 1 Byte, 2 Byte, 4 Byte 에 해당하는 값을 읽는 법은 익숙해 졌을 테니 두 번째 방법으로 이를 구현해 보겠습니다.
 - Present Position 은 4Byte, Present Velocity 도 4Byte 입니다.
 - 따라서, Present Position 과 Present Velocity 를 한번에 읽기 위해서는 8Byte 의 Data 를 한번에 받아야 하며, 이때 받은 Data 는 dxl_read_byte, dxl_read_word, dxl_read_dword 등과 다르게 사용자가 직접 Byte 수에 맞게 변경해야 합니다.
 - Data 는 아래와 같이 변경할 수 있습니다.

```

unsigned char data[8];
dxl_read(Port, ID, P_PRESENT_POSITION, 8, data, &ErrorStatus);

int present_position, present_velocity;
present_position=DXL_MAKEDWORD(DXL_MAKEWORD(data[0],data[1]),DXL_MAKEWORD(data[2],data[3]));
present_velocity=DXL_MAKEDWORD(DXL_MAKEWORD(data[4],data[5]),DXL_MAKEWORD(data[6],data[7]) );
    
```

- DXL_MAKEWORD, DXL_MAKEDWORD 들을 이용하여 unsigned char 형의 배열로 읽어온 값들을 위치값과 속도값을 나타내는 4Byte 값으로 바꿀 수 있습니다.
- 아래는 프로그램의 전체 Source 입니다.

```

main.cpp
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"

#define COM_PORT_NUM          17      //Comport Number of USB2DXL
#define BAUD_RATE_NUM        3       //Baudrate Number of DXL Pro

#define P_OPERATING_MODE      11      //Address of Operation Mode in Control Table
#define P_TORQUE_ENABLE       562     //Address of Torque Enable in Control Table
#define P_GOAL_POSITION       596     //Address of Goal Position in Control Table
#define P_GOAL_VELOCITY       600     //Address of Goal Velocity in Control Table
#define P_PRESENT_POSITION    611     //Address of Present Position in Control Table
#define P_PRESENT_VELOCITY    615     //Address of Present Velocity in Control Table

#define ID                    1       //ID of DXL Pro you use

// Print error bit of status packet
void PrintErrorCode( int ErrorCode)
{
    if(ErrorCode & ERRBIT_VOLTAGE)
        printf("Input voltage error!\n");

    if(ErrorCode & ERRBIT_ANGLE)
    
```



```

        printf("Angle limit error!\n");

    if(ErrorCode & ERRBIT_OVERHEAT)
        printf("Overheat error!\n");

    if(ErrorCode & ERRBIT_RANGE)
        printf("Out of range error!\n");

    if(ErrorCode & ERRBIT_CHECKSUM)
        printf("Checksum error!\n");

    if(ErrorCode & ERRBIT_OVERLOAD)
        printf("Overload error!\n");

    if(ErrorCode & ERRBIT_INSTRUCTION)
        printf("Instruction code error!\n");
}

int main(void)
{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;

    //Open the port of USB2DXL
    if(dx1_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )
        printf("Succeed to open USB2Dynamixel!\n");
    else
    {
        printf( "Failed to open USB2Dynamixel!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    int Result, ErrorStatus;

    //Torque Off
    printf( "Torque Off...\n" );
    Result = dx1_write_byte(Port, ID, P_TORQUE_ENABLE, 0, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    printf("Press any key to change the operating mode...\n");
    _getch();
    Result = dx1_write_byte(Port, ID, P_OPERATING_MODE, 1, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }
}

```

```

}
else
{
    if(ErrorStatus != 0 )
        PrintErrorCode(ErrorStatus);
    else
        printf("Succeed to chage the operationg mode!\n");
}

//Torque On
printf( "Torque On...\n" );
Result = dxl_write_byte(Port, ID, P_TORQUE_ENABLE, 1, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}

//Change the Goal Velocity
printf("Press any key to change goal velocity...\n");
_getch();
Result = dxl_write_dword(Port, ID, P_GOAL_VELOCITY, 5000, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}
else
{
    if(ErrorStatus != 0 )
        PrintErrorCode(ErrorStatus);
    else
        printf("Succeed to chage the operationg mode!\n");
}

printf("Press any key to terminate...\n" );
printf("\n");
while(true)
{
    if(_kbhit())
        break;

    unsigned char data[8];
    dxl_read(Port, ID, P_PRESENT_POSITION, 8, data, &ErrorStatus);

    int present_position, present_velocity;
    present_position = DXL_MAKEDWORD( DXL_MAKEWORD(data[0], data[1]),
                                     DXL_MAKEWORD(data[2], data[3]) );
    present_velocity = DXL_MAKEDWORD( DXL_MAKEWORD(data[4], data[5]),
                                     DXL_MAKEWORD(data[6], data[7]) );
}

```

```
        printf("Wr");
        printf("present position : %d, presen velocity : %d"
              , present_position, present_velocity);
    }
    printf("Wn");

    //Close the port of USB2DXL
    printf( "Press any key to terminate...Wn" );
    _getch();

    //Torque Off
    printf( "Torque Off...Wn" );
    Result = dxl_write_byte(Port, ID, P_TORQUE_ENABLE, 0, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!Wn" );
        printf( "Press any key to terminate...Wn" );
        _getch();
        return 0;
    }

    dxl_terminate(Port);
    return 0;
}
```

1.2.3 Dynamixel Pro 에서 추가된 기능들을 C 언어를 이용하여 사용해 봅시다.

- i. Dynamixel Pro 의 0 점을 바꾼 후 이를 확인해 봅시다.
 - Dynamixel Pro 는 0 점 위치를 사용자가 마음대로 바꿀 수 있습니다.
 - Dynamixel Pro 에서 0 점 위치를 바꿀 수 있는 방법은 Control Table 에서 Homing Offset 값을 바꾸는 방법입니다.
 - Control Table 에서 Homing Offset 의 주소값은 13 번입니다. 또한 Homing Offset 은 위치값으로 지정하게 되므로 4Byte 가 할당되어 있습니다. 따라서, dxl_write_dword 로 Homing Offset 을 바꿀 수 있습니다.
 - 하지만, Homing Offset 은 Torque On 상태에서는 바꿀 수 없는 EEPROM 영역이므로 Homing Offset 을 바꾸기 전에는 반드시 Torque Off 상태로 만들어야 합니다.
 - Homing Offset 값을 바꿀 때에는 바꾸고자 하는 0 점 위치에 -를 곱해서 지정하면 됩니다.
 - 예를 들어 Position Value 로 50000 의 위치를 0 점으로 하고 싶다면 Homing Offset 에는 -50000 을 넣어야 합니다.
 - 아래는 프로그램의 전체 Source 입니다.
 - 아래의 프로그램을 실행하기 전에 Operation Mode 가 Joint Mode 로 되어 있는 지를 확인하십시오.

```
main.cpp
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"

#define COM_PORT_NUM      17      //Comport Number of USB2DXL
#define BAUD_RATE_NUM    3       //Baudrate Number of DXL Pro

#define P_HOMING_OFFSET   13      //Address of Homing Offset in Control Table
#define P_TORQUE_ENABLE   562     //Address of Torque Enable in Control Table
#define P_GOAL_POSITION   596     //Address of Goal Position in Control Table

#define ID                1      //ID of DXL Pro you use

// Print error bit of status packet
void PrintErrorCode( int ErrorCode)
{
    if(ErrorCode & ERRBIT_VOLTAGE)
        printf("Input voltage error!\n");

    if(ErrorCode & ERRBIT_ANGLE)
        printf("Angle limit error!\n");

    if(ErrorCode & ERRBIT_OVERHEAT)
        printf("Overheat error!\n");

    if(ErrorCode & ERRBIT_RANGE)
        printf("Out of range error!\n");
}
```

```

    if(ErrorCode & ERRBIT_CHECKSUM)
        printf("Checksum error!\n");

    if(ErrorCode & ERRBIT_OVERLOAD)
        printf("Overload error!\n");

    if(ErrorCode & ERRBIT_INSTRUCTION)
        printf("Instruction code error!\n");
}

int main(void)
{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;

    //Open the port of USB2DXL
    if(dx1_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )
        printf("Succeed to open USB2Dynamixel!\n");
    else
    {
        printf( "Failed to open USB2Dynamixel!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    int Result, ErrorStatus;

    //Torque Off
    printf( "Torque Off...\n" );
    Result = dx1_write_byte(Port, ID, P_TORQUE_ENABLE, 0, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    //Change the zero point
    printf("Press any key to change the zero point...\n");
    _getch();
    Result = dx1_write_dword(Port, ID, P_HOMING_OFFSET, -50000, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }
    else
    {
        if(ErrorStatus != 0 )
            PrintErrorCode(ErrorStatus);
        else

```

```

        printf("Succeed to chage the homing offset!\n");
    }

    //Torque On
    printf( "Torque On...\n" );
    Result = dxl_write_byte(Port, ID, P_TORQUE_ENABLE, 1, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    //Change the Goal position
    printf("Press any key to change the Goal Position...\n");
    _getch();
    Result = dxl_write_dword(Port, ID, P_GOAL_POSITION, 0, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }
    else
        PrintErrorCode(ErrorStatus);

    //Close the port of USB2DXL
    printf( "Press any key to terminate...\n" );
    _getch();
    dxl_terminate(Port);
    return 0;
}

```

- 만약, 프로그램이 제대로 실행이 되지 않는다면 Operating Mode 가 Joint Mode 로 세팅되어 있는 지를 확인하십시오.

ii. Dynamixel Pro 의 작동 범위를 줄인 후 이를 확인해 봅시다.

- +, - Position Limit 에 대한 설명은 1.1 을 참고합니다.
- 아래는 프로그램의 전체 Source 입니다.

```
main.cpp
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"

#define COM_PORT_NUM          17      //Comport Number of USB2DXL
#define BAUD_RATE_NUM        3       //Baudrate Number of DXL Pro

#define P_PLUS_POSITION_LIMIT 36      //Address of Plus Position Limit in Control Table
#define P_MINUS_POSITION_LIMIT 40     //Address of Minus Position Limit in Control Table
#define P_TORQUE_ENABLE       562     //Address of Torque Enable in Control Table
#define P_GOAL_POSITION       596     //Address of Goal Position in Control Table

#define ID                    1       //ID of DXL Pro you use

// Print errorbit of status packet
void PrintErrorCode( int ErrorCode)
{
    if(ErrorCode & ERRBIT_VOLTAGE)
        printf("Input voltage error!\n");

    if(ErrorCode & ERRBIT_ANGLE)
        printf("Angle limit error!\n");

    if(ErrorCode & ERRBIT_OVERHEAT)
        printf("Overheat error!\n");

    if(ErrorCode & ERRBIT_RANGE)
        printf("Out of range error!\n");

    if(ErrorCode & ERRBIT_CHECKSUM)
        printf("Checksum error!\n");

    if(ErrorCode & ERRBIT_OVERLOAD)
        printf("Overload error!\n");

    if(ErrorCode & ERRBIT_INSTRUCTION)
        printf("Instruction code error!\n");
}

int main(void)
{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;

    //Open the port of USB2DXL
    if(dx1_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )
        printf("Succeed to open USB2Dynamixel!\n");
    else
    {
```

```

        printf( "Failed to open USB2Dynamixel!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    int Result, ErrorStatus;

    //Torque Off
    printf( "Torque Off...\n" );
    Result = dxl_write_byte(Port, ID, P_TORQUE_ENABLE, 0, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    //Change the Plus Position Limit
    printf("Press any key to change the plus position limit...\n");
    _getch();
    Result = dxl_write_dword(Port, ID, P_PLUS_POSITION_LIMIT, 100000, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }
    else
    {
        if(ErrorStatus != 0 )
            PrintErrorCode(ErrorStatus);
        else
            printf("Succeed to chage the plus position limit!\n");
    }

    //Change the Minus Position Limit
    printf("Press any key to change the minus position limit...\n");
    _getch();
    Result = dxl_write_dword(Port, ID, P_MINUS_POSITION_LIMIT, -100000, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }
    else
    {
        if(ErrorStatus != 0 )
            PrintErrorCode(ErrorStatus);
        else
            printf("Succeed to chage the minus position limit!\n");
    }
}

```



```
//Torque On
printf( "Torque On...\n" );
Result = dxl_write_byte(Port, ID, P_TORQUE_ENABLE, 1, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}

//Change the Goal position
printf("Press any key to change the Goal Position to 120000...\n");
_getch();
Result = dxl_write_dword(Port, ID, P_GOAL_POSITION, 120000, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}
else
    PrintErrorCode(ErrorStatus);

//Change the Goal position
printf("Press any key to change the Goal Position to 100000\n");
_getch();
Result = dxl_write_dword(Port, ID, P_GOAL_POSITION, 100000, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}
else
    PrintErrorCode(ErrorStatus);

//Change the Goal position
printf("Press any key to change the Goal Position to -120000\n");
_getch();
Result = dxl_write_dword(Port, ID, P_GOAL_POSITION, -120000, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}
else
    PrintErrorCode(ErrorStatus);
```

```
//Change the Goal position
printf( "Press any key to change the Goal Position to -100000\n" );
_getch();
Result = dxl_write_dword(Port, ID, P_GOAL_POSITION, -100000, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}
else
    PrintErrorCode(ErrorStatus);

//Close the port of USB2DXL
printf( "Press any key to terminate...\n" );
_getch();
dxl_terminate(Port);
return 0;
}
```

- 만약, 프로그램이 제대로 실행이 되지 않는다면 Operating Mode 가 Joint Mode 로 세팅되어 있는 지를 확인하십시오.

iii. Dynamixel Pro 의 작동 범위를 늘린 후 이를 확인해 봅시다.

- +, - Position Limit 에 대한 설명은 1.1 을 참고합니다.
- 아래는 프로그램의 전체 Source 입니다.

```
main.cpp
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"

#define COM_PORT_NUM          17      //Comport Number of USB2DXL
#define BAUD_RATE_NUM        3       //Baudrate Number of DXL Pro

#define P_PLUS_POSITION_LIMIT 36      //Address of Plus Position Limit in Control Table
#define P_MINUS_POSITION_LIMIT 40     //Address of Minus Position Limit in Control Table
#define P_TORQUE_ENABLE       562     //Address of Torque Enable in Control Table
#define P_GOAL_POSITION        596    //Address of Goal Position in Control Table

#define ID                    1       //ID of DXL Pro you use

// Print error bit of status packet
void PrintErrorCode( int ErrorCode)
{
    if(ErrorCode & ERRBIT_VOLTAGE)
        printf("Input voltage error!\n");

    if(ErrorCode & ERRBIT_ANGLE)
        printf("Angle limit error!\n");

    if(ErrorCode & ERRBIT_OVERHEAT)
        printf("Overheat error!\n");

    if(ErrorCode & ERRBIT_RANGE)
        printf("Out of range error!\n");

    if(ErrorCode & ERRBIT_CHECKSUM)
        printf("Checksum error!\n");

    if(ErrorCode & ERRBIT_OVERLOAD)
        printf("Overload error!\n");

    if(ErrorCode & ERRBIT_INSTRUCTION)
        printf("Instruction code error!\n");
}

int main(void)
{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;

    //Open the port of USB2DXL
    if(dx1_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )
        printf("Succeed to open USB2Dynamixel!\n");
    else
    {
        printf( "Failed to open USB2Dynamixel!\n" );
    }
}
```

```

        printf( "Press any key to terminate...\Wn" );
        _getch();
        return 0;
    }

    int Result, ErrorStatus;

    //Torque Off
    printf( "Torque Off...\Wn" );
    Result = dxl_write_byte(Port, ID, P_TORQUE_ENABLE, 0, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\Wn" );
        printf( "Press any key to terminate...\Wn" );
        _getch();
        return 0;
    }

    //Change the Plus Position Limit
    printf("Press any key to change the plus position limit...\Wn");
    _getch();
    Result = dxl_write_dword(Port, ID, P_PLUS_POSITION_LIMIT, 500000, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\Wn" );
        printf( "Press any key to terminate...\Wn" );
        _getch();
        return 0;
    }
    else
    {
        if(ErrorStatus != 0 )
            PrintErrorCode(ErrorStatus);
        else
            printf("Succeed to chage the plus position limit!\Wn");
    }

    //Change the Minus Position Limit
    printf("Press any key to change the minus position limit...\Wn");
    _getch();
    Result = dxl_write_dword(Port, ID, P_MINUS_POSITION_LIMIT, -500000, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\Wn" );
        printf( "Press any key to terminate...\Wn" );
        _getch();
        return 0;
    }
    else
    {
        if(ErrorStatus != 0 )
            PrintErrorCode(ErrorStatus);
        else
            printf("Succeed to chage the minus position limit!\Wn");
    }
}

```

```

//Torque On
printf( "Torque On...\n" );
Result = dxl_write_byte(Port, ID, P_TORQUE_ENABLE, 1, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}

//Change the Goal position
printf("Press any key to change the Goal Position to 500000...\n");
_getch();
Result = dxl_write_dword(Port, ID, P_GOAL_POSITION, 500000, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}
else
    PrintErrorCode(ErrorStatus);

//Change the Goal position
printf("Press any key to change the Goal Position to -500000\n");
_getch();
Result = dxl_write_dword(Port, ID, P_GOAL_POSITION, -500000, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}
else
    PrintErrorCode(ErrorStatus);

//Close the port of USB2DXL
printf( "Press any key to terminate...\n" );
_getch();
dxl_terminate(Port);
return 0;
}

```

- 만약, 프로그램이 제대로 실행이 되지 않는다면 Operating Mode 가 Joint Mode 로 세팅되어 있는지를 확인하십시오.

1.2.4 Dynamixel Pro 의 Indirect Addressing 기능을 사용해 봅시다.

- i. Dynamixel PRO 의 위치, 속도, 가속도를 한번에 변경해 봅시다.
 - Dynamixel Pro 의 Control Table 을 보면, Goal Position, Goal Velocity, Goal Acceleration 이 연속적으로 배열되어 있지 않기 때문에 이 3 가지 값을 한번에 바꾸기 위해서는 dxl_write 와 dxl_write_dword 를 통하여 두 번 통신하여야 합니다.
 - 이런 상황을 해결하기 위해 Dynamixel Pro 에서는 Indirect Address 기능이 구현되어 있습니다.
 - Indirect Address 는 기본 주소값 외에 하나의 주소값을 더 부여하는 기능입니다.
 - 예를 들어 596 번을 주소값으로 갖는 Goal Position 에 634 번의 주소값을 하나 더 갖도록 하는 것입니다.
 - Indirect Address 는 먼저 EEPROM 영역의 Indirect Address 에 설정하고자 하는 주소값을 쓰는 것부터 시작합니다.
 - Indirect Address 에 지정하는 값은 기본 주소값이며, 2 Byte 씩 할당되어 있으므로 아래와 같이 dxl_write_word 를 사용하여 지정합니다.

```
dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_0, P_GOAL_POSITION, &ErrorStatus);
dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_1, P_GOAL_POSITION+1, &ErrorStatus);
dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_2, P_GOAL_POSITION+2, &ErrorStatus);
dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_3, P_GOAL_POSITION+3, &ErrorStatus);
dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_4, P_GOAL_VELOCITY, &ErrorStatus);
dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_5, P_GOAL_VELOCITY+1, &ErrorStatus);
dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_6, P_GOAL_VELOCITY+2, &ErrorStatus);
dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_7, P_GOAL_VELOCITY+3, &ErrorStatus);
dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_8, P_GOAL_ACCELERATION, &ErrorStatus);
dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_9, P_GOAL_ACCELERATION+1, &ErrorStatus);
dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_10, P_GOAL_ACCELERATION+2, &ErrorStatus);
dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_11, P_GOAL_ACCELERATION+3, &ErrorStatus);
```

- 아래는 프로그램의 전체 Source 입니다.

```
main.cpp
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"

#define COM_PORT_NUM 17 //Comport Number of USB2DXL
#define BAUD_RATE_NUM 3 //Baudrate Number of DXL Pro

#define P_INDIRECT_ADDRESS_0 49 //Address of 1st Indirect Address in Control Table
#define P_INDIRECT_ADDRESS_1 51 //Address of 2nd Indirect Address in Control Table
#define P_INDIRECT_ADDRESS_2 53 //Address of 3rd Indirect Address in Control Table
#define P_INDIRECT_ADDRESS_3 55 //Address of 4th Indirect Address in Control Table
#define P_INDIRECT_ADDRESS_4 57 //Address of 5th Indirect Address in Control Table
#define P_INDIRECT_ADDRESS_5 59 //Address of 6th Indirect Address in Control Table
#define P_INDIRECT_ADDRESS_6 61 //Address of 7th Indirect Address in Control Table
#define P_INDIRECT_ADDRESS_7 63 //Address of 8th Indirect Address in Control Table
#define P_INDIRECT_ADDRESS_8 65 //Address of 9th Indirect Address in Control Table
#define P_INDIRECT_ADDRESS_9 67 //Address of 10th Indirect Address in Control Table
```

```

#define P_INDIRECT_ADDRESS_10 69 //Address of 11th Indirect Address in Control Table
#define P_INDIRECT_ADDRESS_11 71 //Address of 12th Indirect Address in Control Table

#define P_TORQUE_ENABLE 562 //Address of Torque Enable in Control Table
#define P_GOAL_POSITION 596 //Address of Goal Position in Control Table
#define P_GOAL_VELOCITY 600 //Address of Goal Velocity in Control Table
#define P_GOAL_ACCELERATION 606 //Address of Goal Acceleration in Control Table
#define P_INDIRECT_DATA_0 634 //Address of Goal Indirect Data in Control Table

#define ID 1 //ID of DXL Pro you use

// Print error bit of status packet
void PrintErrorCode( int ErrorCode)
{
    if(ErrorCode & ERRBIT_VOLTAGE)
        printf("Input voltage error!\n");

    if(ErrorCode & ERRBIT_ANGLE)
        printf("Angle limit error!\n");

    if(ErrorCode & ERRBIT_OVERHEAT)
        printf("Overheat error!\n");

    if(ErrorCode & ERRBIT_RANGE)
        printf("Out of range error!\n");

    if(ErrorCode & ERRBIT_CHECKSUM)
        printf("Checksum error!\n");

    if(ErrorCode & ERRBIT_OVERLOAD)
        printf("Overload error!\n");

    if(ErrorCode & ERRBIT_INSTRUCTION)
        printf("Instruction code error!\n");
}

int main(void)
{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;

    //Open the port of USB2DXL
    if(dxl_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )
        printf("Succeed to open USB2Dynamixel!\n");
    else
    {
        printf( "Failed to open USB2Dynamixel!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    int Result, ErrorStatus;

    //Torque Off
    printf( "Torque Off...\n" );

```

```

Result = dxl_write_byte(Port, ID, P_TORQUE_ENABLE, 0, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}

//Set the Indirect Address for Goal Position
printf("Press any key to set the indirect address for goal position...\n");
_getch();
Result = dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_0, P_GOAL_POSITION, &ErrorStatus);
Result = dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_1, P_GOAL_POSITION+1, &ErrorStatus);
Result = dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_2, P_GOAL_POSITION+2, &ErrorStatus);
Result = dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_3, P_GOAL_POSITION+3, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}
else
{
    if(ErrorStatus != 0 )
        PrintErrorCode(ErrorStatus);
    else
        printf("Succeed to set the indirect address for goal position!\n");
}

//Set the Indirect Address for Goal Velocity
printf("Press any key to set the indirect address for goal velocity...\n");
_getch();
Result = dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_4, P_GOAL_VELOCITY, &ErrorStatus);
Result = dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_5, P_GOAL_VELOCITY+1, &ErrorStatus);
Result = dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_6, P_GOAL_VELOCITY+2, &ErrorStatus);
Result = dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_7, P_GOAL_VELOCITY+3, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}
else
{
    if(ErrorStatus != 0 )
        PrintErrorCode(ErrorStatus);
    else
        printf("Succeed to set the indirect address for goal velocity!\n");
}

//Set the Indirect Address for Goal Acceleration
printf("Press any key to set the indirect address for goal acceleration...\n");
_getch();

```



```

Result = dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_8, P_GOAL_ACCELERATION, &ErrorStatus);
Result = dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_9, P_GOAL_ACCELERATION+1, &ErrorStatus);
Result = dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_10, P_GOAL_ACCELERATION+2, &ErrorStatus);
Result = dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_11, P_GOAL_ACCELERATION+3, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}
else
{
    if(ErrorStatus != 0 )
        PrintErrorCode(ErrorStatus);
    else
        printf("Succeed to set the indirect address for goal acceleration!\n");
}

//Torque On
printf( "Torque On...\n" );
Result = dxl_write_byte(Port, ID, P_TORQUE_ENABLE, 1, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}

int position, velocity, acceleration;
position = 100000;
velocity = 10000;
acceleration = 16;

//Make a tx data
unsigned char data[12];
data[0] = DXL_LOBYTE(DXL_LOWORD(position));
data[1] = DXL_HIBYTE(DXL_LOWORD(position));
data[2] = DXL_LOBYTE(DXL_HIWORD(position));
data[3] = DXL_HIBYTE(DXL_HIWORD(position));
data[4] = DXL_LOBYTE(DXL_LOWORD(velocity));
data[5] = DXL_HIBYTE(DXL_LOWORD(velocity));
data[6] = DXL_LOBYTE(DXL_HIWORD(velocity));
data[7] = DXL_HIBYTE(DXL_HIWORD(velocity));
data[8] = DXL_LOBYTE(DXL_LOWORD(acceleration));
data[9] = DXL_HIBYTE(DXL_LOWORD(acceleration));
data[10] = DXL_LOBYTE(DXL_HIWORD(acceleration));
data[11] = DXL_HIBYTE(DXL_HIWORD(acceleration));

//change the position value, moving speed, acceleration using indirect addr
printf( "Press any key to change the position, speed, acceleration...\n" );
_getch();
Result = dxl_write(Port, ID, P_INDIRECT_DATA_0, 12, data, &ErrorStatus);
if( Result != COMM_RXSUCCESS )

```

```

{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}
else
{
    if(ErrorStatus != 0 )
        PrintErrorCode(ErrorStatus);
}

position = -100000;
velocity = 1000;
acceleration = 2;

//Make a tx data
data[0] = DXL_LOBYTE(DXL_LOWORD(position));
data[1] = DXL_HIBYTE(DXL_LOWORD(position));
data[2] = DXL_LOBYTE(DXL_HIWORD(position));
data[3] = DXL_HIBYTE(DXL_HIWORD(position));
data[4] = DXL_LOBYTE(DXL_LOWORD(velocity));
data[5] = DXL_HIBYTE(DXL_LOWORD(velocity));
data[6] = DXL_LOBYTE(DXL_HIWORD(velocity));
data[7] = DXL_HIBYTE(DXL_HIWORD(velocity));
data[8] = DXL_LOBYTE(DXL_LOWORD(acceleration));
data[9] = DXL_HIBYTE(DXL_LOWORD(acceleration));
data[10] = DXL_LOBYTE(DXL_HIWORD(acceleration));
data[11] = DXL_HIBYTE(DXL_HIWORD(acceleration));

//change the position value, moving speed, acceleration using indirect addr
printf( "Press any key to change the position, speed, acceleration...\n" );
_getch();
Result = dxl_write(Port, ID, P_INDIRECT_DATA_0, 12, data, &ErrorStatus);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}
else
{
    if(ErrorStatus != 0 )
        PrintErrorCode(ErrorStatus);
}

//Close the port of USB2DXL
printf( "Press any key to terminate...\n" );
_getch();
dxl_terminate(Port);
return 0;
}

```


ii. Dynamixel PRO 의 온도와 현재 위치 값을 한번에 읽어 봅시다.

- Indirect Address 를 이용하면 Write 명령뿐만 아니라 연이어 배열되어 있지 않은 값을 읽을 때에도 쓸 수 있습니다.
- Indirect Address 를 이용하여 현재 위치와 온도 값을 한번에 읽어 봅시다.
- 아래는 프로그램의 전체 Source 입니다.

```
main.cpp
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"

#define COM_PORT_NUM          17    //Comport Number of USB2DXL
#define BAUD_RATE_NUM        3     //Baudrate Number of DXL Pro

#define P_OPERATING_MODE     11    //Address of Operating Mode in Control Table
#define P_INDIRECT_ADDRESS_0 49    //Address of 1st Indirect Address in Control Table
#define P_INDIRECT_ADDRESS_1 51    //Address of 2nd Indirect Address in Control Table
#define P_INDIRECT_ADDRESS_2 53    //Address of 3rd Indirect Address in Control Table
#define P_INDIRECT_ADDRESS_3 55    //Address of 4th Indirect Address in Control Table
#define P_INDIRECT_ADDRESS_4 57    //Address of 5th Indirect Address in Control Table

#define P_TORQUE_ENABLE      562   //Address of Torque Enable in Control Table
#define P_PRESENT_POSITION  611   //Address of Present Position in Control Table
#define P_PRESENT_TEMPERATURE 625  //Address of Present Temperature in Control Table

#define P_INDIRECT_DATA_0    634   //Address of Goal Indirect Data in Control Table

#define ID                    1    //ID of DXL Pro you use

// Print error bit of status packet
void PrintErrorCode( int ErrorCode)
{
    if(ErrorCode & ERRBIT_VOLTAGE)
        printf("Input voltage error!\n");

    if(ErrorCode & ERRBIT_ANGLE)
        printf("Angle limit error!\n");

    if(ErrorCode & ERRBIT_OVERHEAT)
        printf("Overheat error!\n");

    if(ErrorCode & ERRBIT_RANGE)
        printf("Out of range error!\n");

    if(ErrorCode & ERRBIT_CHECKSUM)
        printf("Checksum error!\n");

    if(ErrorCode & ERRBIT_OVERLOAD)
        printf("Overload error!\n");

    if(ErrorCode & ERRBIT_INSTRUCTION)
        printf("Instruction code error!\n");
}
```

```

int main(void)
{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;

    //Open the port of USB2DXL
    if(dxl_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )
        printf("Succeed to open USB2Dynamixel!\n");
    else
    {
        printf( "Failed to open USB2Dynamixel!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    int Result, ErrorStatus;

    //Torque Off
    printf( "Torque Off...\n" );
    Result = dxl_write_byte(Port, ID, P_TORQUE_ENABLE, 0, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    //change the operating mode to wheel mode
    printf( "Make the Mode of DXL Pro to Wheel Mode...\n" );
    Result = dxl_write_byte(Port, ID, P_OPERATING_MODE, 1, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    //Set the Indirect Address for Present Temperature
    printf("Press any key to set the indirect address for temperature...\n");
    _getch();
    Result = dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_0, P_PRESENT_TEMPERATURE, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }
    else
    {
        if(ErrorStatus != 0 )
            PrintErrorCode(ErrorStatus);
    }
}

```

```

        else
            printf("Succeed to set the indirect address for temperature!\n");
    }

    //Set the Indirect Address for Goal Velocity
    printf("Press any key to set the indirect address for present position...\n");
    _getch();
    Result = dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_1, P_PRESENT_POSITION, &ErrorStatus);
    Result = dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_2, P_PRESENT_POSITION+1, &ErrorStatus);
    Result = dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_3, P_PRESENT_POSITION+2, &ErrorStatus);
    Result = dxl_write_word(Port, ID, P_INDIRECT_ADDRESS_4, P_PRESENT_POSITION+3, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }
    else
    {
        if(ErrorStatus != 0 )
            PrintErrorCode(ErrorStatus);
        else
            printf("Succeed to set the indirect address for present position!\n");
    }

    //Torque On
    printf( "Torque On...\n" );
    Result = dxl_write_byte(Port, ID, P_TORQUE_ENABLE, 1, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    //Rotating Start
    printf("Rotating Start\n");
    Result = dxl_write_dword(Port, ID, 600, 5000, &ErrorStatus);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    printf("Press any key to terminate...\n" );
    printf("\n");
    while(true)
    {
        if(_kbhit())
            break;

        unsigned char data[5];

```

```
        dxl_read(Port, ID, P_INDIRECT_DATA_0, 5, data, &ErrorStatus);

        int temp, present_position;
        temp = data[0];
        present_position = DXL_MAKEDWORD( DXL_MAKEWORD(data[1], data[2]), DXL_MAKEWORD(data[3],
data[4]) );

        printf("Wr ");
        printf("present temperature : %d, presen position : %d", temp, present_position);
    }
    printf("Wn");

    //Close the port of USB2DXL
    dxl_terminate(Port);
    return 0;
}
```

1.2.5 여러 개의 Dynamixel Pro 를 사용해 봅시다.

i. 3 개의 Dynamixel Pro 의 LED 를 On/Off 해 봅시다.

- 여러 개의 Dynamixel Pro 에 동시에 명령을 내릴때에는 dxl_sync_write 함수를 사용합니다.
- 아래는 프로그램의 전체 Source 입니다.

```

main.cpp
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"

#define COM_PORT_NUM      17      //Comport Number of USB2DXL
#define BAUD_RATE_NUM    3       //Baudrate Number of DXL Pro

#define P_LED_RED        563     //Address of LED RED in Control Table

#define ID_1             1       //ID of DXL Pro you use
#define ID_2             2
#define ID_3             3

// Print error bit of status packet
void PrintErrorCode(int ErrorCode)
{
    if(ErrorCode & ERRBIT_VOLTAGE)
        printf("Input voltage error!\n");

    if(ErrorCode & ERRBIT_ANGLE)
        printf("Angle limit error!\n");

    if(ErrorCode & ERRBIT_OVERHEAT)
        printf("Overheat error!\n");

    if(ErrorCode & ERRBIT_RANGE)
        printf("Out of range error!\n");

    if(ErrorCode & ERRBIT_CHECKSUM)
        printf("Checksum error!\n");

    if(ErrorCode & ERRBIT_OVERLOAD)
        printf("Overload error!\n");

    if(ErrorCode & ERRBIT_INSTRUCTION)
        printf("Instruction code error!\n");
}

int main(void)
{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;

    //Open the port of USB2DXL
    if(dxl_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )
        printf("Succeed to open USB2Dynamixel!\n");
    else

```



```

    {
        printf( "Failed to open USB2Dynamixel!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    int Result;

    unsigned char param[6];
    param[0] = ID_1;
    param[1] = 255;
    param[2] = ID_2;
    param[3] = 255;
    param[4] = ID_3;
    param[5] = 255;

    //LED On
    printf( "Press any to turn on the LED...\n" );
    _getch();
    Result = dxl_sync_write(Port, P_LED_RED, 1, param, 6);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    param[0] = ID_1;
    param[1] = 0;
    param[2] = ID_2;
    param[3] = 0;
    param[4] = ID_3;
    param[5] = 0;

    //LED Off
    printf( "Press any to turn off the LED...\n" );
    _getch();
    Result = dxl_sync_write(Port, P_LED_RED, 1, param, 6);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    //Close the port of USB2DXL
    printf( "Press any key to terminate...\n" );
    _getch();
    dxl_terminate(Port);
    return 0;
}

```

ii. 3 개의 Dynamixel Pro 의 목표 위치 바꾸어 봅시다.

- 아래는 프로그램의 전체 Source 입니다.

```
main.cpp
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"

#define COM_PORT_NUM          17      //Comport Number of USB2DXL
#define BAUD_RATE_NUM        3       //Baudrate Number of DXL Pro

#define P_TORQUE_ENABLE       562     //Address of Torque Enable in Control Table
#define P_GOAL_POSITION       596     //Address of Goal Position in Control Table

#define ID_1                   1      //ID of DXL Pro you use
#define ID_2                   2
#define ID_3                   3

// Print error bit of status packet
void PrintErrorCode(int ErrorCode)
{
    if(ErrorCode & ERRBIT_VOLTAGE)
        printf("Input voltage error!\n");

    if(ErrorCode & ERRBIT_ANGLE)
        printf("Angle limit error!\n");

    if(ErrorCode & ERRBIT_OVERHEAT)
        printf("Overheat error!\n");

    if(ErrorCode & ERRBIT_RANGE)
        printf("Out of range error!\n");

    if(ErrorCode & ERRBIT_CHECKSUM)
        printf("Checksum error!\n");

    if(ErrorCode & ERRBIT_OVERLOAD)
        printf("Overload error!\n");

    if(ErrorCode & ERRBIT_INSTRUCTION)
        printf("Instruction code error!\n");
}

int main(void)
{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;

    //Open the port of USB2DXL
    if(dxl_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )
        printf("Succeed to open USB2Dynamixel!\n");
    else
    {
        printf("Failed to open USB2Dynamixel!\n");
        printf("Press any key to terminate...\n");
    }
}
```

```

        _getch();
        return 0;
    }

    int Result;
    unsigned char param[15];
    //Torque ON
    printf("Torque On\n");
    param[0] = ID_1;
    param[1] = 1;
    param[2] = ID_2;
    param[3] = 1;
    param[4] = ID_3;
    param[5] = 1;
    Result = dxl_sync_write(Port, P_TORQUE_ENABLE, 1, param, 6);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    param[0] = ID_1;
    param[1] = DXL_LOBYTE(DXL_LOWORD(10000));
    param[2] = DXL_HIBYTE(DXL_LOWORD(10000));
    param[3] = DXL_LOBYTE(DXL_HIWORD(10000));
    param[4] = DXL_HIBYTE(DXL_HIWORD(10000));
    param[5] = ID_2;
    param[6] = DXL_LOBYTE(DXL_LOWORD(50000));
    param[7] = DXL_HIBYTE(DXL_LOWORD(50000));
    param[8] = DXL_LOBYTE(DXL_HIWORD(50000));
    param[9] = DXL_HIBYTE(DXL_HIWORD(50000));
    param[10]= ID_3;
    param[11]= DXL_LOBYTE(DXL_LOWORD(-80000));
    param[12]= DXL_HIBYTE(DXL_LOWORD(-80000));
    param[13]= DXL_LOBYTE(DXL_HIWORD(-80000));
    param[14]= DXL_HIBYTE(DXL_HIWORD(-80000));

    printf( "Press any to change goal position...\n" );
    _getch();
    Result = dxl_sync_write(Port, P_GOAL_POSITION, 4, param, 15);
    if( Result != COMM_RXSUCCESS )
    {
        printf( "Failed to write!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    param[0] = ID_1;
    param[1] = DXL_LOBYTE(DXL_LOWORD(0));
    param[2] = DXL_HIBYTE(DXL_LOWORD(0));
    param[3] = DXL_LOBYTE(DXL_HIWORD(0));
    param[4] = DXL_HIBYTE(DXL_HIWORD(0));

```

```
param[5] = ID_2;
param[6] = DXL_LOBYTE(DXL_LOWORD(0));
param[7] = DXL_HIBYTE(DXL_LOWORD(0));
param[8] = DXL_LOBYTE(DXL_HIWORD(0));
param[9] = DXL_HIBYTE(DXL_HIWORD(0));
param[10]= ID_3;
param[11]= DXL_LOBYTE(DXL_LOWORD(0));
param[12]= DXL_HIBYTE(DXL_LOWORD(0));
param[13]= DXL_LOBYTE(DXL_HIWORD(0));
param[14]= DXL_HIBYTE(DXL_HIWORD(0));

printf( "Press any to change goal position...\n" );
_getch();
Result = dxl_sync_write(Port, P_GOAL_POSITION, 4, param, 15);
if( Result != COMM_RXSUCCESS )
{
    printf( "Failed to write!\n" );
    printf( "Press any key to terminate...\n" );
    _getch();
    return 0;
}

//Close the port of USB2DXL
printf( "Press any key to terminate...\n" );
_getch();
dxl_terminate(Port);
return 0;
}
```

iii. 3 개의 Dynamixel Pro 의 현재 위치 값을 확인해 봅시다.

- 여러 개의 Dynamixel Pro 에서 동시에 값을 읽어올 때에는 dxl_bulk_read 함수를 사용합니다.
- 아래는 프로그램의 전체 Source 입니다.

```
main.cpp
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"

#define COM_PORT_NUM      17      //Comport Number of USB2DXL
#define BAUD_RATE_NUM    3       //Baudrate Number of DXL Pro

#define P_PRESENT_POSITION 611   //Address of Goal Position in Control Table

#define ID_1              1       //ID of DXL Pro you use
#define ID_2              2
#define ID_3              3

// Print error bit of status packet
void PrintErrorCode(int ErrorCode)
{
    if(ErrorCode & ERRBIT_VOLTAGE)
        printf("Input voltage error!\n");

    if(ErrorCode & ERRBIT_ANGLE)
        printf("Angle limit error!\n");

    if(ErrorCode & ERRBIT_OVERHEAT)
        printf("Overheat error!\n");

    if(ErrorCode & ERRBIT_RANGE)
        printf("Out of range error!\n");

    if(ErrorCode & ERRBIT_CHECKSUM)
        printf("Checksum error!\n");

    if(ErrorCode & ERRBIT_OVERLOAD)
        printf("Overload error!\n");

    if(ErrorCode & ERRBIT_INSTRUCTION)
        printf("Instruction code error!\n");
}

int main(void)
{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;

    //Open the port of USB2DXL
    if(dxl_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )
        printf("Succeed to open USB2Dynamixel!\n");
    else
    {
```

```

        printf( "Failed to open USB2Dynamixel!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    int Result;
    int position_length=4;
    unsigned char param[15];
    param[0] = ID_1;
    param[1] = DXL_LOBYTE(P_PRESENT_POSITION);
    param[2] = DXL_HIBYTE(P_PRESENT_POSITION);
    param[3] = DXL_LOBYTE(position_length);
    param[4] = DXL_HIBYTE(position_length);
    param[5] = ID_2;
    param[6] = DXL_LOBYTE(P_PRESENT_POSITION);
    param[7] = DXL_HIBYTE(P_PRESENT_POSITION);
    param[8] = DXL_LOBYTE(position_length);
    param[9] = DXL_HIBYTE(position_length);
    param[10] = ID_3;
    param[11] = DXL_LOBYTE(P_PRESENT_POSITION);
    param[12] = DXL_HIBYTE(P_PRESENT_POSITION);
    param[13] = DXL_LOBYTE(position_length);
    param[14] = DXL_HIBYTE(position_length);

    BulkData bd[256];
    BulkData *pbd[256];
    for(unsigned int i = 0 ; i <256 ; i++)
    {
        pbd[i] = &bd[i];
    }

    dxl_bulk_read(Port, param, 15, pbd);

    int present_position1;
    int present_position2;
    int present_position3;

    dxl_get_bulk_dword(pbd, ID_1, P_PRESENT_POSITION, (unsigned*)&present_position1);
    dxl_get_bulk_dword(pbd, ID_2, P_PRESENT_POSITION, (unsigned*)&present_position2);
    dxl_get_bulk_dword(pbd, ID_3, P_PRESENT_POSITION, (unsigned*)&present_position3);

    printf( "Present Position\n" );
    printf( "ID_1 : %d\n", present_position1);
    printf( "ID_2 : %d\n", present_position2);
    printf( "ID_3 : %d\n", present_position3);

    //Close the port of USB2DXL
    printf( "Press any key to terminate...\n" );
    _getch();
    dxl_terminate(Port);
    return 0;
}

```


- iv. 3 개의 Dynamixel Pro 에서 첫 번째 Dynamixel Pro 는 온도를, 두 번째 Dynamixel Pro 는 위치를, 세 번째는 Dynamixel Pro 는 현재 토크를 확인해 봅시다.
- dxl_bulk_read 를 사용하면, 각각의 Dynamixel Pro 로부터 다른 정보를 얻을 수 있습니다.
 - 아래는 전체 소스코드 입니다.

```

main.cpp
#include <stdio.h>
#include <conio.h>
#include "dynamixel.h"

#define COM_PORT_NUM          17    //Comport Number of USB2DXL
#define BAUD_RATE_NUM        3     //Baudrate Number of DXL Pro

#define P_PRESENT_POSITION   611   //Address of Goal Position in Control Table
#define P_PRESENT_CURRENT    621   //Address of Present Current in Control Table
#define P_PRESENT_TEMPERATURE 625  //Address of Present Temperature in Control Table

#define ID_1                  1     //ID of DXL Pro you use
#define ID_2                  2
#define ID_3                  3

// Print error bit of status packet
void PrintErrorCode( int ErrorCode)
{
    if(ErrorCode & ERRBIT_VOLTAGE)
        printf("Input voltage error!\n");

    if(ErrorCode & ERRBIT_ANGLE)
        printf("Angle limit error!\n");

    if(ErrorCode & ERRBIT_OVERHEAT)
        printf("Overheat error!\n");

    if(ErrorCode & ERRBIT_RANGE)
        printf("Out of range error!\n");

    if(ErrorCode & ERRBIT_CHECKSUM)
        printf("Checksum error!\n");

    if(ErrorCode & ERRBIT_OVERLOAD)
        printf("Overload error!\n");

    if(ErrorCode & ERRBIT_INSTRUCTION)
        printf("Instruction code error!\n");
}

int main(void)
{
    SerialPort sp = {0,0,0,0,0};
    SerialPort *Port = &sp;

    //Open the port of USB2DXL
    if(dxl_initialize(Port, COM_PORT_NUM, BAUD_RATE_NUM) == COMM_RXSUCCESS )

```



```

        printf("Succeed to open USB2Dynamixel!\n");
    else
    {
        printf( "Failed to open USB2Dynamixel!\n" );
        printf( "Press any key to terminate...\n" );
        _getch();
        return 0;
    }

    int Result;
    int current_length = 2, position_length = 4, temperature_length = 1;
    unsigned char param[15];
    param[0] = ID_1;
    param[1] = DXL_LOBYTE(P_PRESENT_CURRENT);
    param[2] = DXL_HIBYTE(P_PRESENT_CURRENT);
    param[3] = DXL_LOBYTE(current_length);
    param[4] = DXL_HIBYTE(current_length);
    param[5] = ID_2;
    param[6] = DXL_LOBYTE(P_PRESENT_POSITION);
    param[7] = DXL_HIBYTE(P_PRESENT_POSITION);
    param[8] = DXL_LOBYTE(position_length);
    param[9] = DXL_HIBYTE(position_length);
    param[10] = ID_3;
    param[11] = DXL_LOBYTE(P_PRESENT_TEMPERATURE);
    param[12] = DXL_HIBYTE(P_PRESENT_TEMPERATURE);
    param[13] = DXL_LOBYTE(temperature_length);
    param[14] = DXL_HIBYTE(temperature_length);

    BulkData bd[256];
    BulkData *pbd[256];
    for(unsigned int i = 0 ; i <256 ; i++)
        pbd[i] = &bd[i];

    dxl_bulk_read(Port, param, 15, pbd);

    short present_current;
    int present_position;
    int present_temperature;

    dxl_get_bulk_word( pbd, ID_1, P_PRESENT_CURRENT, (int*)&present_current);
    dxl_get_bulk_dword(pbd, ID_2, P_PRESENT_POSITION, (unsigned*)&present_position);
    dxl_get_bulk_byte( pbd, ID_3, P_PRESENT_TEMPERATURE, &present_temperature);

    printf("Present Position\n");
    printf("ID_1 : %d\n", present_current);
    printf("ID_2 : %d\n", present_position);
    printf("ID_3 : %d\n", present_temperature);

    //Close the port of USB2DXL
    printf( "Press any key to terminate...\n" );
    _getch();
    dxl_terminate(Port);
    return 0;
}

```

2 Reference

2.1 모델별 주요 초기값

2.1.1 H Series

i. H54-200-S500-R

Name	Default Value
ID	1
BaudRate	1 (57600bps)
Max Position Limit	251000
Min Position Limit	-251000
Velocity Limit	16600
Current Limit	620
Velocity I Gain	14
Velocity P Gain	399
Position P Gain	32

ii. H54-100-S500-R

Name	Default Value
ID	1
BaudRate	1 (57600bps)
Max Position Limit	251000
Min Position Limit	-251000
Velocity Limit	17000
Current Limit	310
Velocity I Gain	16
Velocity P Gain	256
Position P Gain	32

iii. H42-20-S300-R

Name	Default Value
ID	1
BaudRate	1 (57600bps)
Max Position Limit	151875
Min Position Limit	-151875
Velocity Limit	10300
Current Limit	465
Velocity I Gain	40
Velocity P Gain	440

Position P Gain	32
-----------------	----

2.1.2 M Series

i. M54-60-S250-R

Name	Default Value
ID	1
BaudRate	1 (57600bps)
Max Position Limit	125700
Min Position Limit	-125700
Velocity Limit	8000
Current Limit	180
Velocity I Gain	16
Velocity P Gain	256
Position P Gain	32

ii. M54-40-S250-R

Name	Default Value
ID	1
BaudRate	1 (57600bps)
Max Position Limit	125700
Min Position Limit	-125700
Velocity Limit	8000
Current Limit	180
Velocity I Gain	16
Velocity P Gain	256
Position P Gain	32

2.1.3 L Series

i. L54-50-S290-R

Name	Default Value
ID	1
BaudRate	1 (57600bps)
Max Position Limit	103860
Min Position Limit	-103860
Velocity Limit	8000
Current Limit	180
Velocity I Gain	16

Velocity P Gain	256
Position P Gain	32

ii. L54-30-S400-R

Name	Default Value
ID	1
BaudRate	1 (57600bps)
Max Position Limit	144180
Min Position Limit	-144180
Velocity Limit	13000
Current Limit	180
Velocity I Gain	16
Velocity P Gain	256
Position P Gain	32

2.2 Control Table of Dynamixel Pro

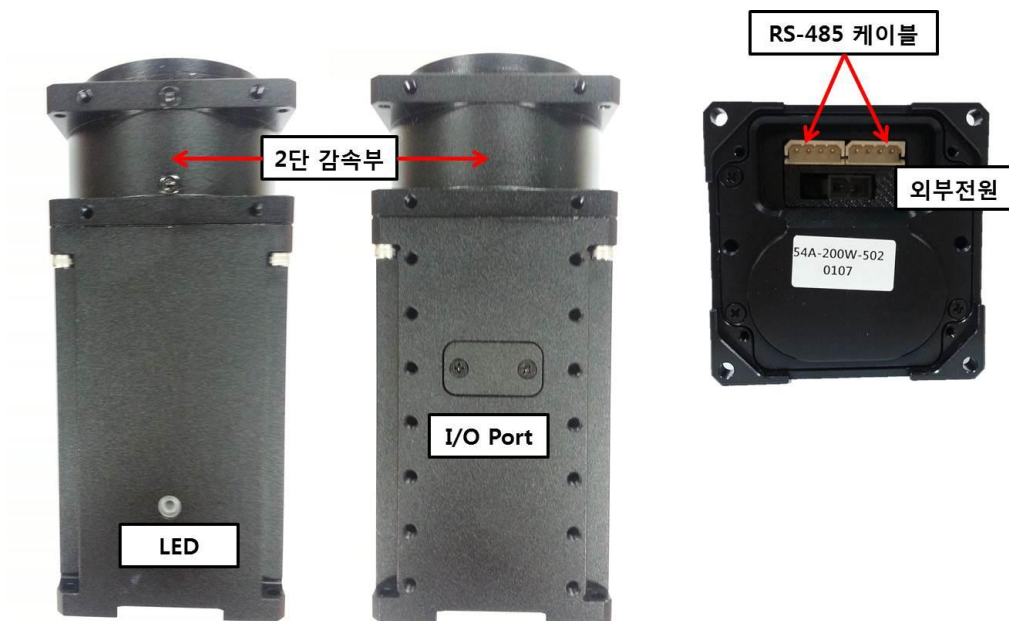
(R : Read, RW : Read and Write)

AREA	주소	크기 (byte)	명칭	의미	접근	기본값
EEPROM	0	2	Model Number	모델번호	R	-
	6	1	Version of Firmware	펌웨어 버전 정보	R	-
	7	1	ID	다이나믹셀 프로 ID	RW	1
	8	1	Baud Rate	다이나믹셀 통신속도	RW	1
	9	1	Return Delay Time	응답지연시간	RW	250
	11	1	Operating mode	동작모드	RW	3
	13	4	Homing offset	0 점 위치 조정값	RW	0
	17	4	Moving threshold	움직임 유무를 결정하는 속도 기준값	RW	50
	21	1	Temperature limit	내부 한계 온도	RW	80
	22	2	Max Voltage Limit	최저 한계 전압	RW	400
	24	2	Min Voltage Limit	최고 한계 전압	RW	150
	26	4	acceleration Limit	가속도 한계값	RW	-
	30	2	Torque limit	토크 한계값	RW	-
	32	4	Velocity Limit	속도 한계값	RW	-
	36	4	Max Position Limit	최대 위치 제한값	RW	-
	40	4	Min Position Limit	최소 위치 한계값	RW	-
	44	1	External Port Mode 1	외부 포트 모드 1	RW	0
	45	1	External Port Mode 2	외부 포트 모드 2	RW	0
	46	1	External Port Mode 3	외부 포트 모드 3	RW	0
	47	1	External Port Mode 4	외부 포트 모드 4	RW	0
	48	1	Shutdown	셧다운	RW	26
	49	2	Indirect Address 1	간접 주소 값 1	RW	634
51	2	Indirect Address 2	간접 주소 값 2	RW	635	
53	2	Indirect Address 3	간접 주소 값 3	RW	636	
...	간접 주소 값 N	RW	-	
569	2	Indirect Address 256	간접 주소 값 256	RW	889	
RAM	562	1	Torque Enable	토크 On/Off	RW	0
	563	1	LED RED	RED LED 밝기 값	RW	0
	564	1	LED GREEN	GREEN LED 밝기 값	RW	0
	565	1	LED BLUE	BLUE LED 밝기 값	RW	0
	586	2	Velocity I Gain	속도 I Gain	RW	-
	588	2	Velocity P Gain	속도 P Gain	RW	-
	594	2	Position P Gain	위치 P Gain	RW	-

596	4	Goal position	목표 위치 값	RW	-
600	4	Goal velocity	목표 속도 값	RW	0
604	2	Goal Torque	목표 토크 값	RW	0
606	4	Goal acceleration	목표 가속도 값	RW	0
610	1	Moving	움직임 유무	R	-
611	4	Present position	현재 위치 값	R	-
615	4	Present velocity	현재 속도 값	R	-
621	2	Present Current	현재 전류 값	R	-
623	2	Present input voltage	현재 전압	R	-
625	1	Present temperature	현재 온도	R	-
626	2	External Port Data 1	외부 포트 데이터 1	R / RW	0
628	2	External Port Data 2	외부 포트 데이터 2	R / RW	0
630	2	External Port Data 3	외부 포트 데이터 3	R / RW	0
632	2	External Port Data 4	외부 포트 데이터 4	R / RW	0
634	1	Indirect Data 1	간접 주소 데이터 1	RW	0
635	1	Indirect Data 2	간접 주소 데이터 2	RW	0
636	1	Indirect Data 3	간접 주소 데이터 3	RW	0
...	...	Indirect Data N	...	RW	0
889	1	Indirect Data 256	간접 주소 데이터 256	RW	0
890	1	Registered Instruction	Instruction 의 등록 여부	R	0
891	1	Status Return Level	응답 레벨	RW	2
892	2	Hardware error status	하드웨어 에러 상태	R	0

2.3 각 부 명칭

2.3.1 H54, M54, L54 시리즈



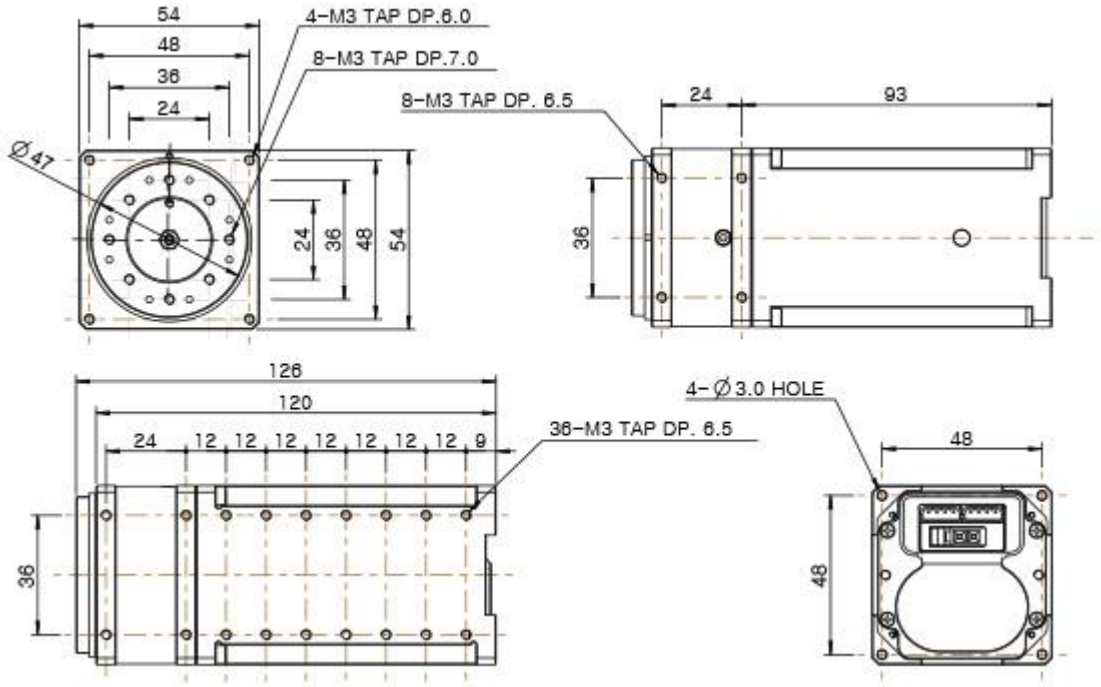
2.3.2 H42, L42 시리즈.



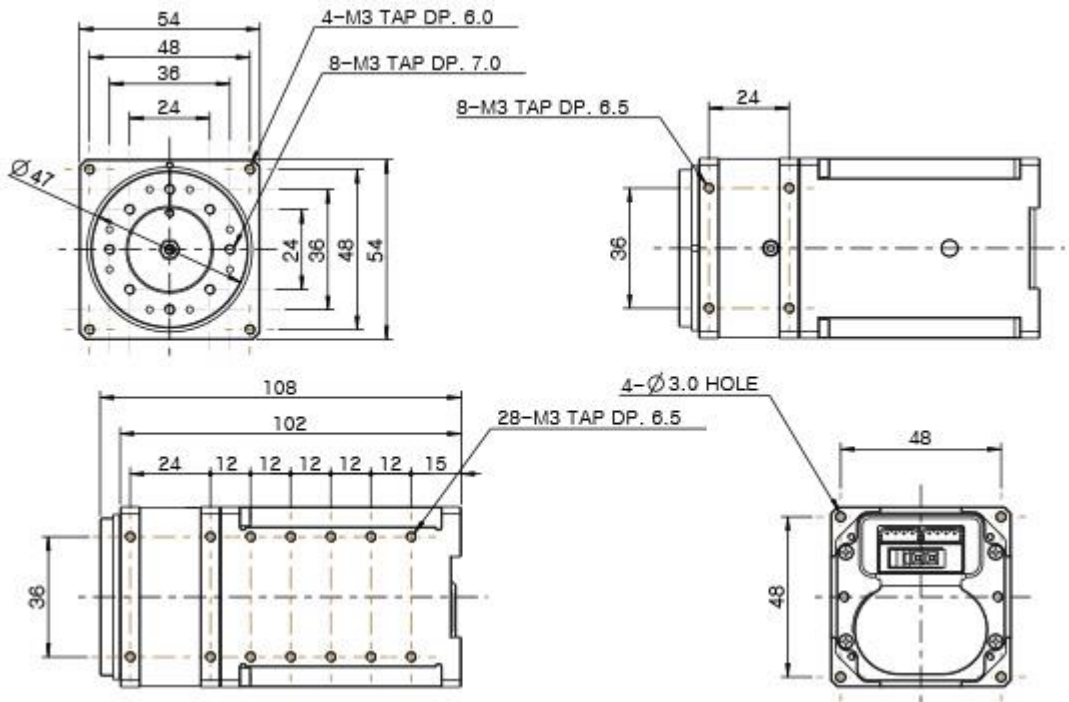
2.4 Dimension

2.4.1 H Series

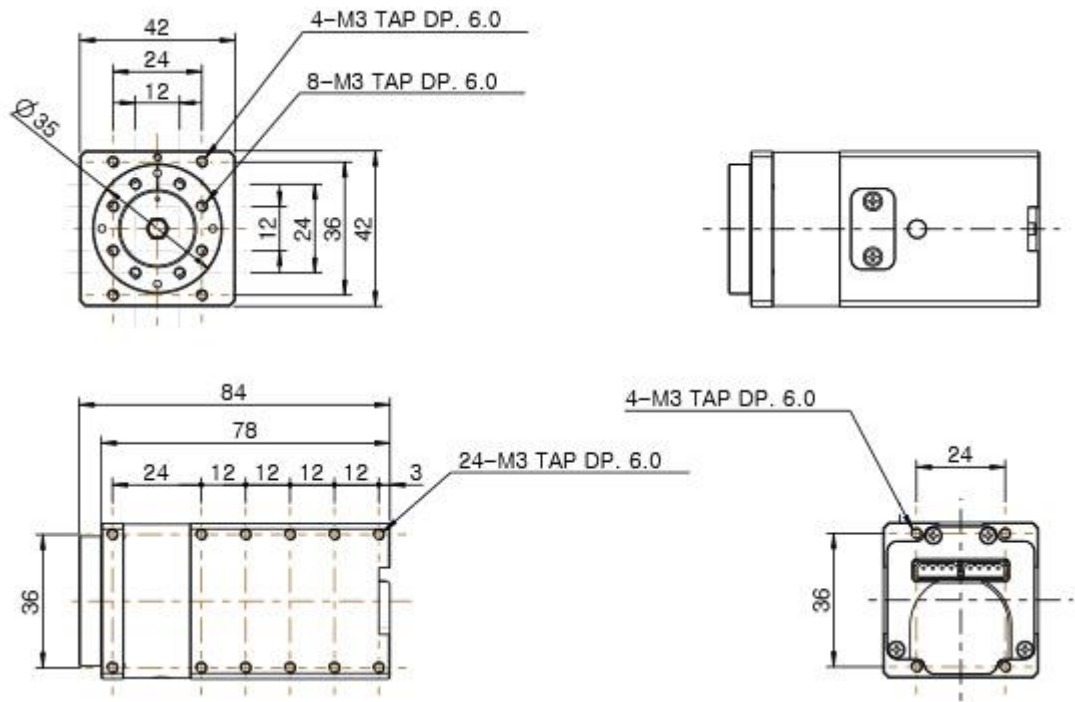
i. H54-200-S500-R



ii. H54-100-S500-R

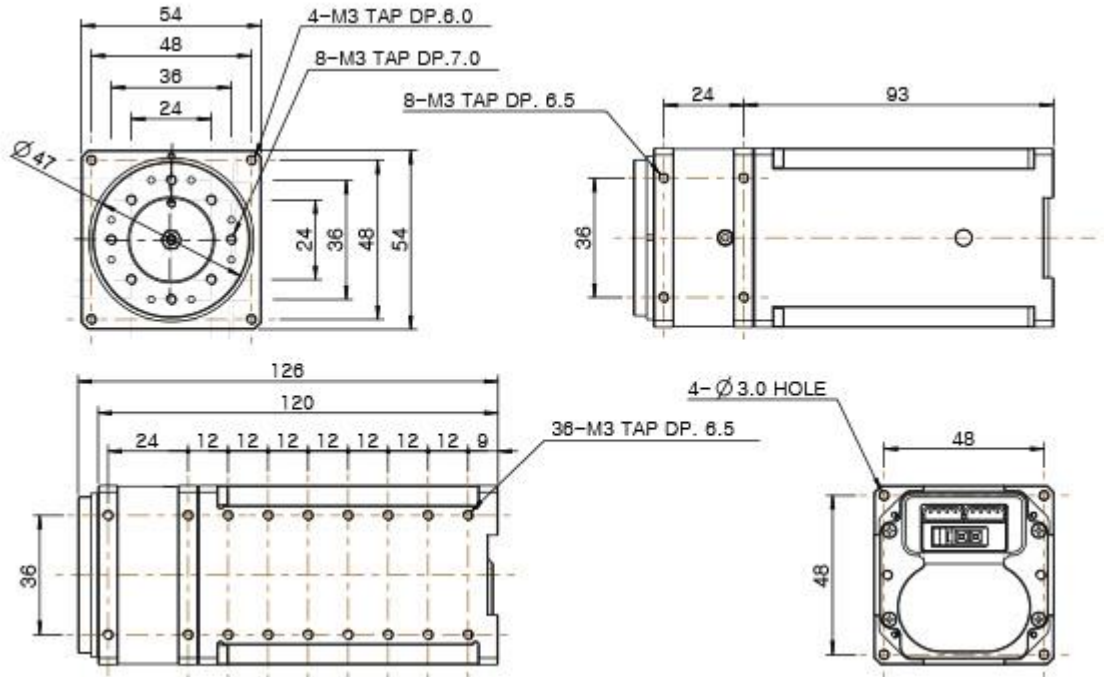


iii. H42-20-S300-R

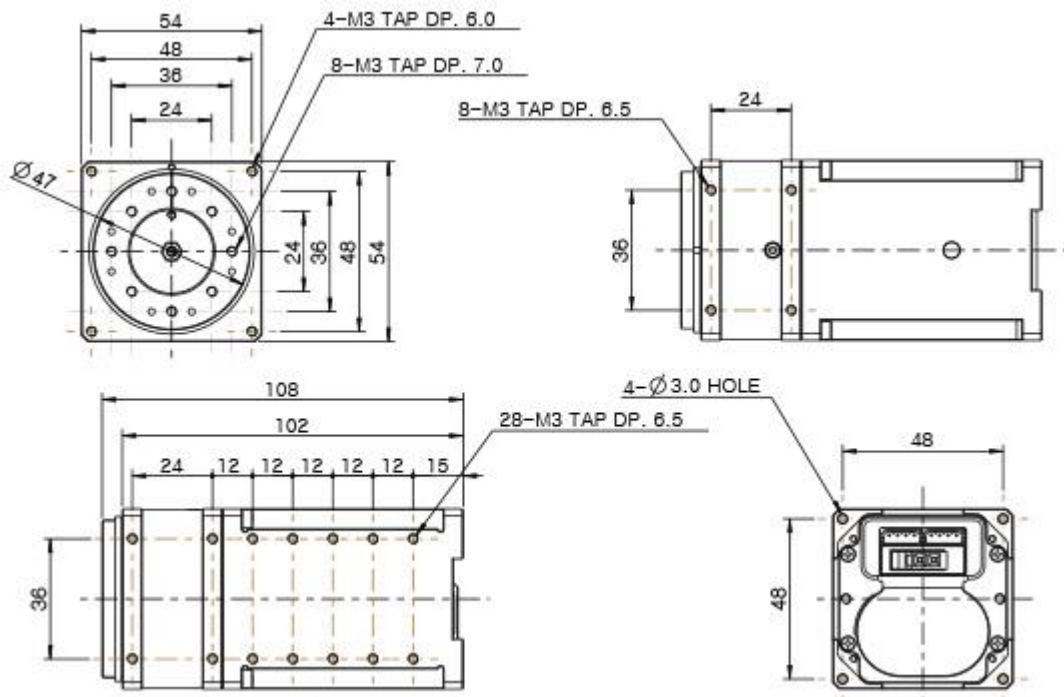


2.4.2 M Series

i. M54-60-S250-R

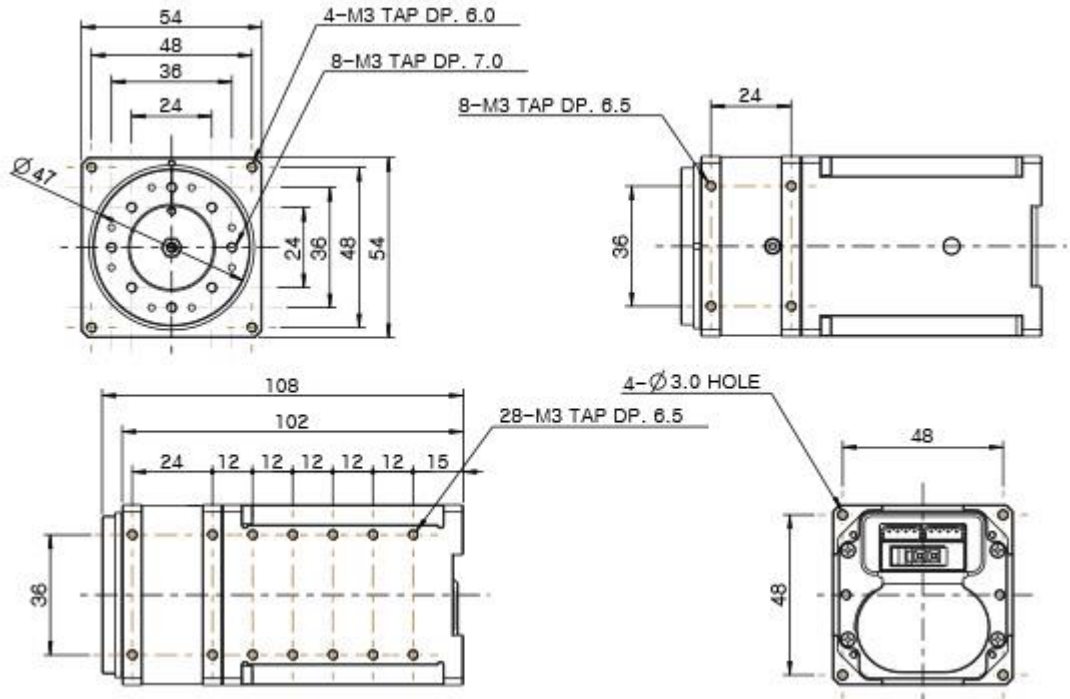


ii. M54-40-S250-R

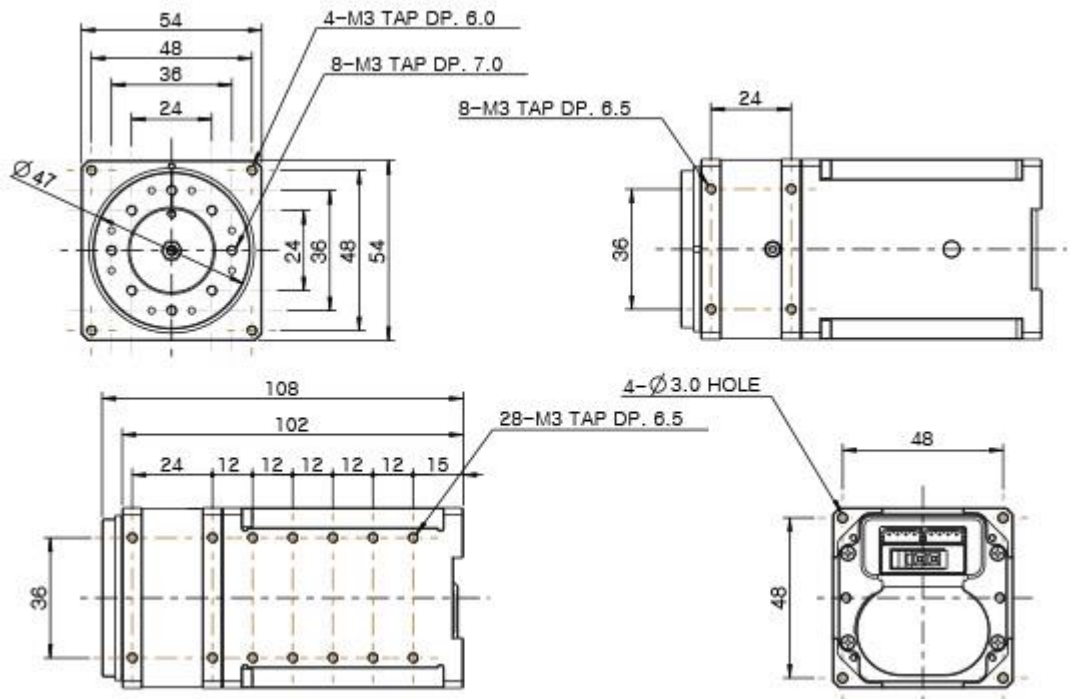


2.4.3 L Series

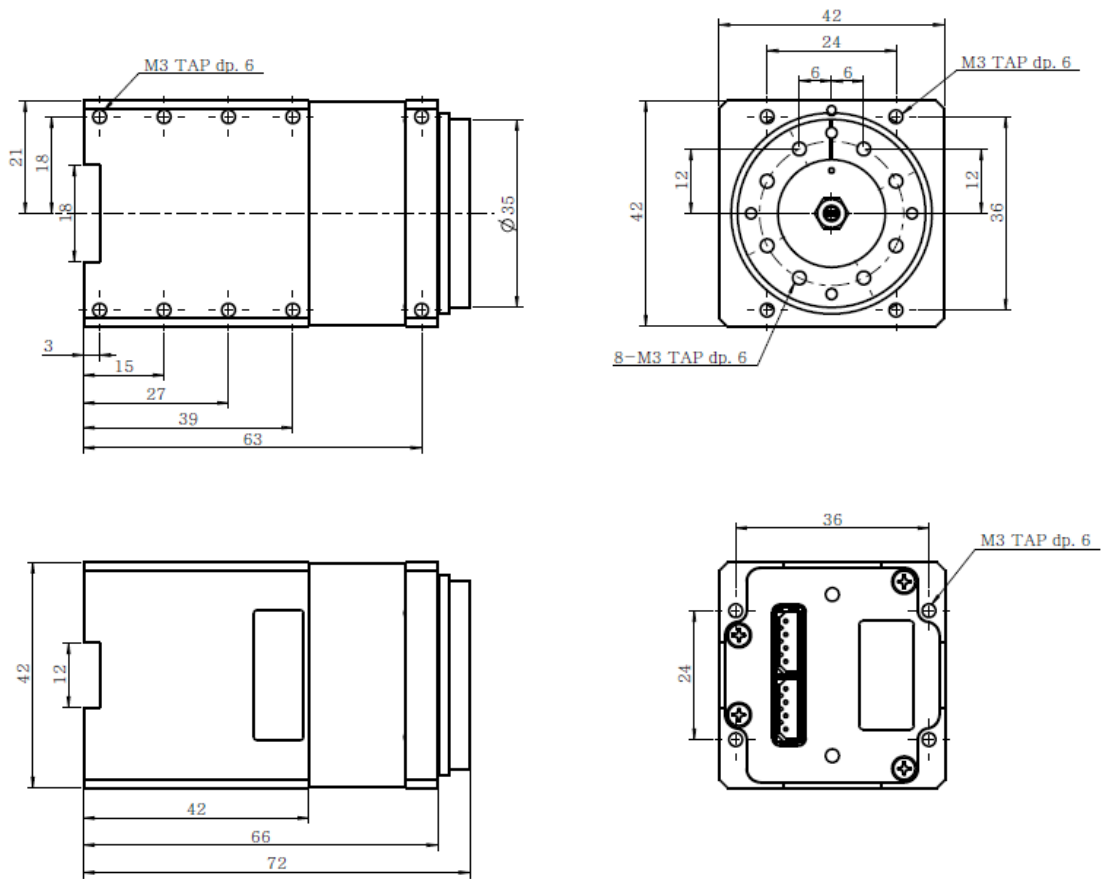
i. L54-50-S290-R



ii. L54-30-S400-R



iii. L42-10-S300-R



2.5 형식 표기 방식

