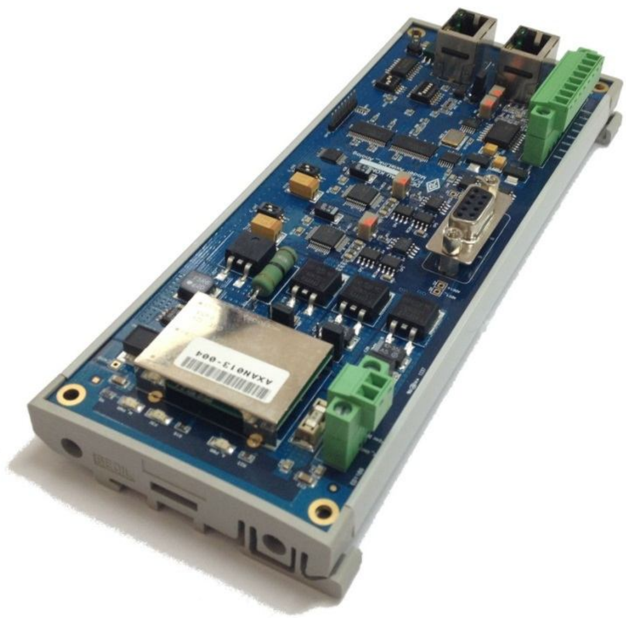
User Manual

AxisLink-Analog



Analog to Digital Converter 2CH
Digital to Analog Converter 2CH

JUL 20, 2015 (REV 1.0.3)

DELTA TAU
Data Systems, Inc.
NEW IDEAS IN MOTION ....

REVISION HISTORY			
VERSION	DESCRIPTION	DATE	APPVD
1.0.0	Manual 초본작성	2012.11.21	이명성
1.0.1	S-CH SW 설명 추가 DAC Data 변환 설명 추가	2014.03.31	이명성
1.0.2	실행 예제 수정	2014.04.15	이명성
1.0.3	AxisLink I/O Explorer 설명 추가	2015.07.20	이명성

※ 본 매뉴얼은 필요에 따라 수시로 업데이트 될 수 있으며, 본사 홈페이지에서 최신항목을 다운 받아 사용하시기 바랍니다. 매뉴얼에 관련된 문의사항이나 요청사항은 델타타우 코리아로 연락바랍니다.

델타타우 코리아 홈페이지 : http://www.deltatau.co.kr
델타타우 코리아 연 락 처 : 031) 813-6156

Table of Contents

1. INTRODUCTION

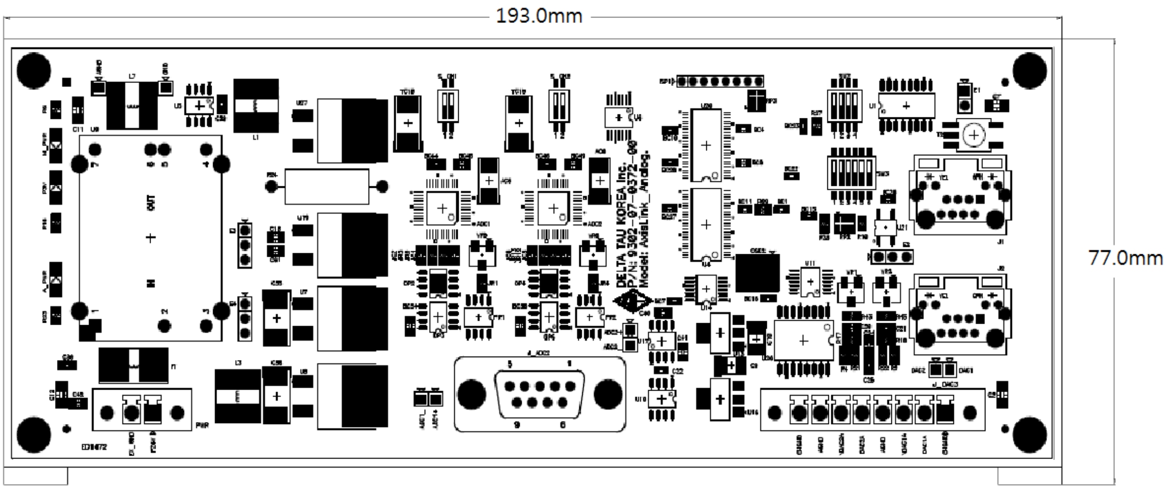
본 매뉴얼은 "AxisLink-Analog" 보드를 사용하는데 필요한 내용을 포함하고 있습니다. "AxisLink-Analog" 보드는 UMAC 에 장착된 "AxisLink-Master" 보드와 함께 사용되는 슬레이브 보드입니다.

"AxisLink-Analog" 보드는 외부의 Analog 신호를 UMAC 으로 전달하거나 UMAC 을 통해 외부로 Analog 신호를 내보낼 때 사용됩니다. "RS422" 통신방식을 이용하여 신호를 전달하며 다수의 AxisLink 보드들과 함께 사용이 가능합니다.

적용 가능한 모델은 UMAC, Clipper, Cruiser 가 있으며 연결되는 모델에는 반드시 AxisLink-Master 보드가 장착되어 있어야 합니다.

2. HARDWARE SETUP

2.1. Dimension



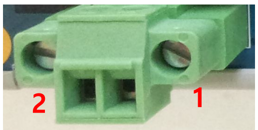
Dimension : 193 x 77 x 40 mm (W * H * D) –Rail Mount 취부 시

2.2. Electronic Specification

전원 : DC 24[v] 1[A]
DAC : Differential -10[v] ~ +10[v]
ADC : Differential -10[v] ~ +10[v]

2.3. Connector

- **PWR (2 Pin Terminal)**



Pin	Symbol	Function	DESCRIPTION
1	P24V	Input	DC 24[V] Power
2	GND	Input	Ground

- **J_DAC (8 Pin Terminal)**

- DAC 출력 CH1과 CH2가 포함됩니다.

Pin	Symbol	Function	DESCRIPTION
1	CHGND	Common	Shield
2	DAC1A	Output	CH1 DAC+
3	/DAC1A	Output	CH1 DAC-
4	AGND	Common	Analog Reference Voltage
5	DAC2A	Output	CH2 DAC+
6	/DAC2A	Output	CH2 DAC-
7	AGND	Common	Analog Reference Voltage
8	CHGND	Common	Shield

- **J_ADC (9 Pin DSUB)**

- ADC 입력 CH1과 CH2가 포함 됩니다.

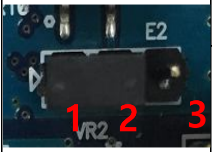
Pin	Symbol	Function	DESCRIPTION
1	-5Vdc	Output	-5V reference output
2	VREF	Output	5Vdc precision reference
3	ADC2-	Input	A-D Conv. Channel 2-
4	AGND	GND	Analog ref GND
5	ADC1+	Input	A-D Conv. Channel 1+
6	+5Vdc	Output	+5V reference output
7	AGND	GND	Analog ref GND
8	ADC2+	Input	A-D Conv. Channel 2+
9	ADC1-	Input	A-D Conv. Channel 1-

2.4. Jumper

- **E1: 종단저항**

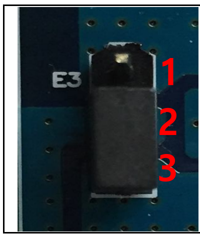
JUMPER	DESCRIPTION	DEFAULT
E1	종단 저항 구성 점퍼 (AxisLink의 마지막 Slave 보드는 Short 처리 할 것)	Short

- **E2: Manual Reset**



JUMPER	DESCRIPTION	DEFAULT
E2	1-2 Short : Not used	1-2 Short
	2-3 Short : Manual Reset	

- **E3: CH1, CH2 OPAMP Bias 전원 선택**
- **E4: CH3, CH4 OPAMP Bias 전원 선택**

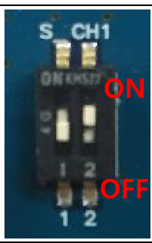


JUMPER	DESCRIPTION	DEFAULT
E3	1-2 Short : -10[V] ~ +10[V]	1-2 Short
	2-3 Short : -5[V] ~ +5[V]	
E4	1-2 Short : -10[V] ~ +10[V]	1-2 Short
	2-3 Short : -5[V] ~ +5[V]	

※ 위 전압 범위는 Differential Input 일 경우 기준입니다.
※ 반드시 "3.3. SW Setup for ADC Input"을 참고하시기 바랍니다.

2.5. Switch

- **S_CH1~4 Bit1: 각 채널의 ADC방식 선택**
- **S_CH1~4 Bit2: 각 채널의 입력범위 선택**

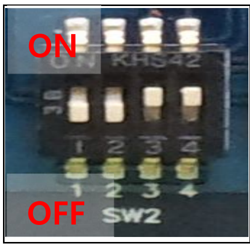


SWITCH	DESCRIPTION	DEFAULT
S_CH1	Bit1: OFF➔Bipolar, ON➔Unipolar	OFF
	Bit2: ON➔±5[V], OFF➔±10[V]	ON
S_CH2	Bit1: OFF➔Bipolar, ON➔Unipolar	OFF
	Bit2: ON➔±5[V], OFF➔±10[V]	ON

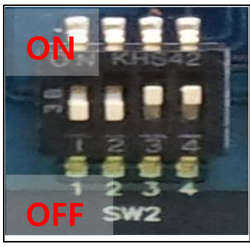
※ 위 전압 범위는 Differential Input 일 경우 기준입니다.
※ 반드시 "3.3. SW Setup for ADC Input"을 참고하시기 바랍니다.

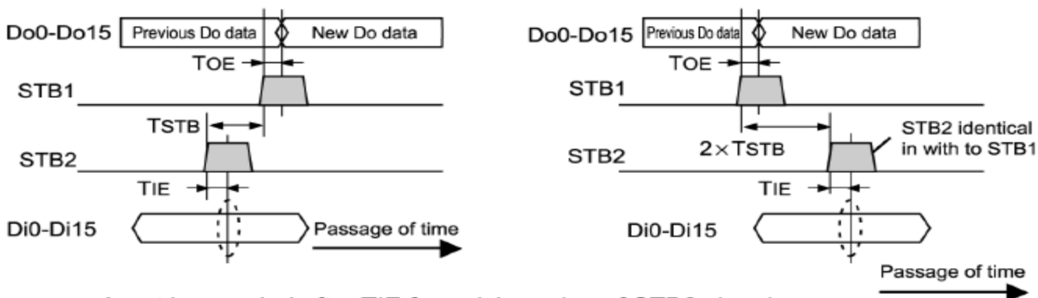
- **SW2: 통신에 관련된 사항을 설정합니다.**

- SW2-1, SW2-2는 통신속도를 설정입니다.

<table>
<tr><td rowspan="5"></td><th>SW2-1</th><th>SW2-2</th><th>Function</th><th>Description</th></tr>
<tr><td>OFF</td><td>OFF</td><td>12Mbps(Default)</td><td rowspan="4">Baud Rate 설정</td></tr>
<tr><td>ON</td><td>OFF</td><td>6Mbps</td></tr>
<tr><td>OFF</td><td>ON</td><td>3Mbps</td></tr>
<tr><td>ON</td><td>ON</td><td>1.5Mbps</td></tr>
</table>

- SW2-3, SW2-4는 DATA 전송에 관련된 설정을 합니다.

<table>
<tr><td rowspan="5"></td><th colspan="2"></th><th>Function</th><th>Description</th></tr>
<tr><th rowspan="2">SW2-3</th><td>ON</td><td>Input Update Before Output</td><td rowspan="2">Input Update 설정</td></tr>
<tr><td>OFF</td><td>Input Update After Output</td></tr>
<tr><th rowspan="2">SW2-4</th><td>ON</td><td>Handshake Enable</td><td rowspan="2">응답대기 설정</td></tr>
<tr><td>OFF</td><td>Handshake Disable</td></tr>
</table>



Input is sampled after TIE from rising edge of STB2 signal

- SW2-3을 ON으로 설정하면 PMAC에서 입력한 DATA가 바로 전달됩니다.
- SW2-3을 OFF로 설정하면 PMAC에서 입력한 DATA가 한 사이클 이후 전달 됩니다.

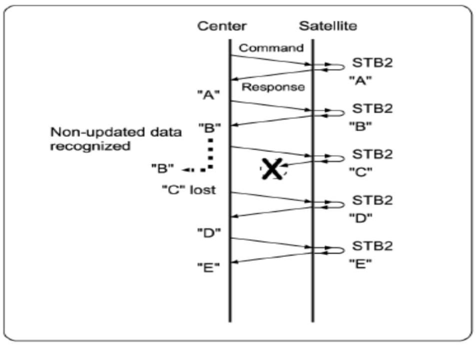


Fig. 4.6B Handshaking Disabled

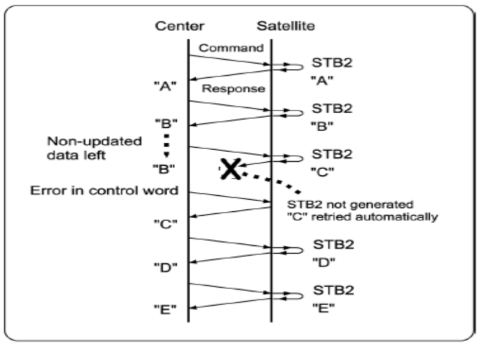
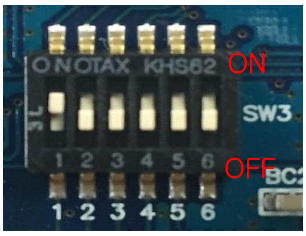


Fig. 4.6C Handshaking Enabled

- SW3-4를 ON으로 설정하면 Handshake가 enable되어 전송되지 않은 Data가 있을 경우 대기 하였다가 Update를 진행합니다.
- SW3-4가 OFF로 되어 있으면 전송되지 않은 Data 가 있어도 Update를 계속 진행 합니다.

- **SW3: AxisLink Slave Adress 설정**



- AxisLink-Analog 는 최대 31개 까지 Daisy Chain으로 연결 가능합니다.
- 각 Slave ID는 SW1에 의해 결정됩니다.
- Master 보드의 Control Register 주소는 AxisLink-Master 보드의 Base Address에 따라 달라집니다.
- AxisLink-Analog의 Data Register의 주소는 Master 보드의 Base Address와 Slave 보드의 ID에 따라 달라집니다.

✓ **Master System Control** Register Address = **Base Address** + **X:$C0**
✓ **Master System Status** Register Address = **Base Address** + **X:$C1**
✓ **Master Basic Control** Register Address = **Base Address** + **X:$C7**

✓ **Slave CH1 DAC Data** Register = **Base Address** + **Slave ID*2** + **Y:$40**
✓ **Slave CH2 DAC Data** Register = **Base Address** + **Slave ID*2** + **Y:$41**
✓ **Slave CH1 ADC Data** Register = **Base Address** + **Slave ID*2** + **Y:$80**
✓ **Slave CH2 ADC Data** Register = **Base Address** + **Slave ID*2** + **Y:$81**

※ AxisLink-Master 보드의 Base Address 설정 및 Register에 관련된 설명은 Master Board User Manual을 참고하시기 바랍니다.

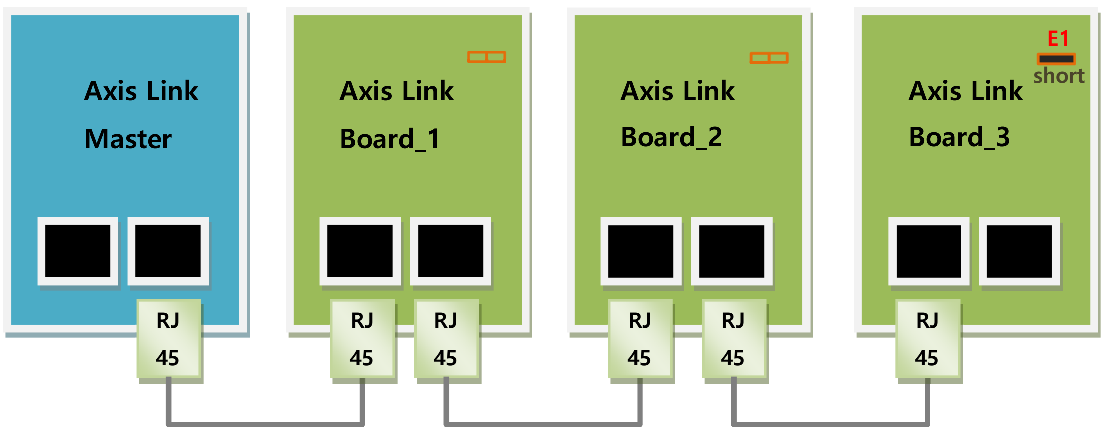
- AxisLink-Analog Data Register Address

NO	Board ID	Bit0	Bit1	Bit2	Bi3	Bit4	Bit5	Slave Address
1	1번 Board	ON	OFF	OFF	OFF	OFF	Don't Care	Base Add + $42,$43
2	2번 Board	OFF	ON	OFF	OFF	OFF		Base Add + $44,$45
3	3번 Board	ON	ON	OFF	OFF	OFF		Base Add + $46,$47
4	4번 Board	OFF	OFF	ON	OFF	OFF		Base Add + $48,$49
5	5번 Board	ON	OFF	ON	OFF	OFF		Base Add + $4A,$4B
6	6번 Board	OFF	ON	ON	OFF	OFF		Base Add + $4C,$4D
7	7번 Board	ON	ON	ON	OFF	OFF		Base Add + $4E,$4F
8	8번 Board	OFF	OFF	OFF	ON	OFF		Base Add + $50,$51
9	9번 Board	ON	OFF	OFF	ON	OFF		Base Add + $52,$53
10	10번 Board	OFF	ON	OFF	ON	OFF		Base Add + $54,$55
11	11번 Board	ON	ON	OFF	ON	OFF		Base Add + $56,$57
12	12번 Board	OFF	OFF	ON	ON	OFF		Base Add + $58,$59
13	13번 Board	ON	OFF	ON	ON	OFF		Base Add + $5A,$5B
14	14번 Board	OFF	ON	ON	ON	OFF		Base Add + $5C,$5D
15	15번 Board	ON	ON	ON	ON	OFF		Base Add + $5E,$5F
16	16번 Board	OFF	OFF	OFF	OFF	ON		Base Add + $60,$61
17	17번 Board	ON	OFF	OFF	OFF	ON		Base Add + $62,$63
18	18번 Board	OFF	ON	OFF	OFF	ON		Base Add + $64,$65
19	19번 Board	ON	ON	OFF	OFF	ON		Base Add + $66,$67
20	20번 Board	OFF	OFF	ON	OFF	ON		Base Add + $68,$69
21	21번 Board	ON	OFF	ON	OFF	ON		Base Add + $6A,$6B
22	22번 Board	OFF	ON	ON	OFF	ON		Base Add + $6C,$6D
23	23번 Board	ON	ON	ON	OFF	ON		Base Add + $6E,$6F
24	24번 Board	OFF	OFF	OFF	ON	ON		Base Add + $70,$71
25	25번 Board	ON	OFF	OFF	ON	ON		Base Add + $72,$73
26	26번 Board	OFF	ON	OFF	ON	ON		Base Add + $74,$75
27	27번 Board	ON	ON	OFF	ON	ON		Base Add + $76,$77
28	28번 Board	OFF	OFF	ON	ON	ON		Base Add + $78,$79
29	29번 Board	ON	OFF	ON	ON	ON		Base Add + $7A,$7B
30	30번 Board	OFF	ON	ON	ON	ON		Base Add + $7C,$7D
31	31번 Board	ON	ON	ON	ON	ON		Base Add + $7E,$7F

3. CONNECTION

3.1. Connection with Master

AxisLink는 Master와 Slave간에 "RJ45" Connector를 통해 "RS422" 통신을 합니다. 여러 개의 Slave 보드를 연결 시 아래와 같이 Daisy Chain을 구성하여야 합니다.



- 여러 개의 Slave는 서로 다른 Address를 가져야 하며 Slave Board의 Address는 **Dip Switch1**을 사용하여 설정합니다. (최대 31개 Slave 연결가능)

- 구성된 Daisy Chain의 마지막 Slave Board(위 예시의 Board3)에는 임피던스를 매칭시키기 위해 **종단저항**을 삽입하여야 합니다. 종단저항은 Slave Board의 점퍼(**E1**)를 Short시키면 삽입됩니다.

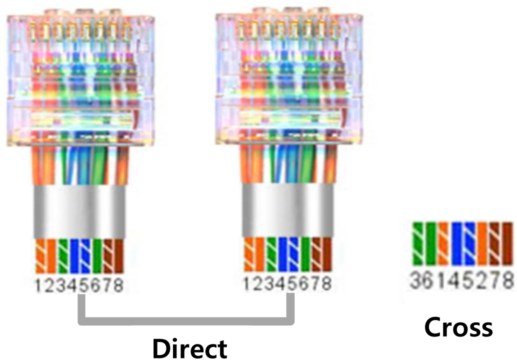
- 연결을 위해 사용되는 Ethernet Cable은 일반적으로 사용되는 Cross Type이 아닌 **Direct Type**을 사용하여야 합니다.

- **Direct Type**의 경우 양단의 RJ45 Connector 선 배열이 일치합니다.

- **Cross Type**의 경우 양단의 배열이 다릅니다.

  1번(주황 줄) ↔ 3번(녹색 줄)

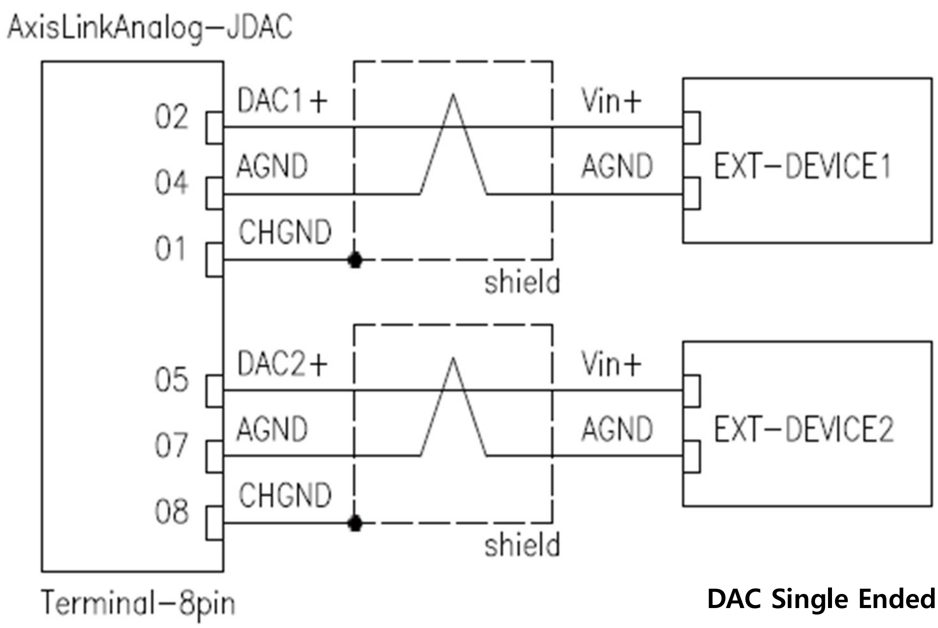
  2번(주황) ↔ 6번(녹색)



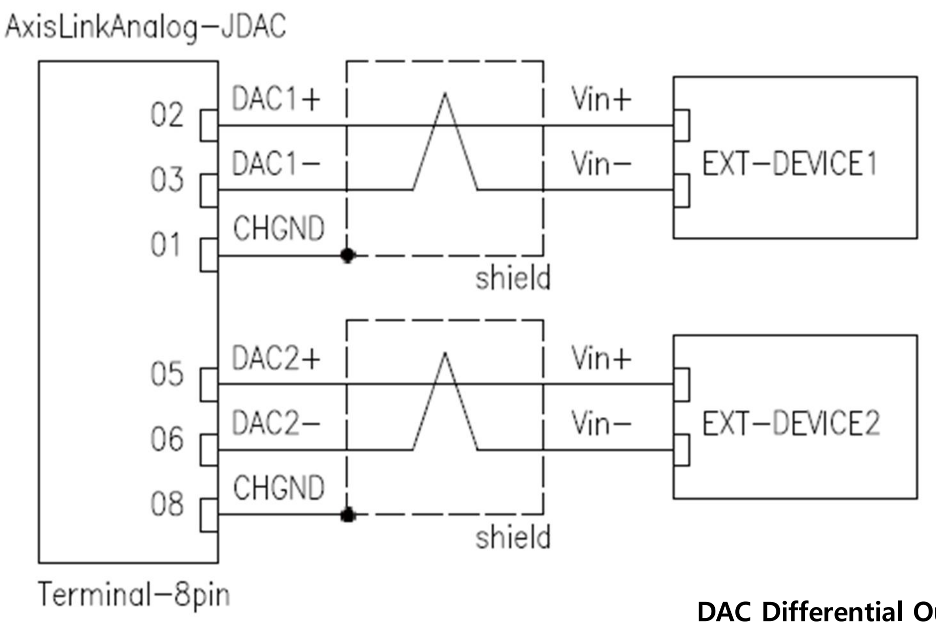
(위의 선 색깔은 일반적으로 사용되는 경우이며 제조사에 따라 색깔은 다를 수 있습니다)

3.2. Connection with Device

3.2.1. J_DAC Connection

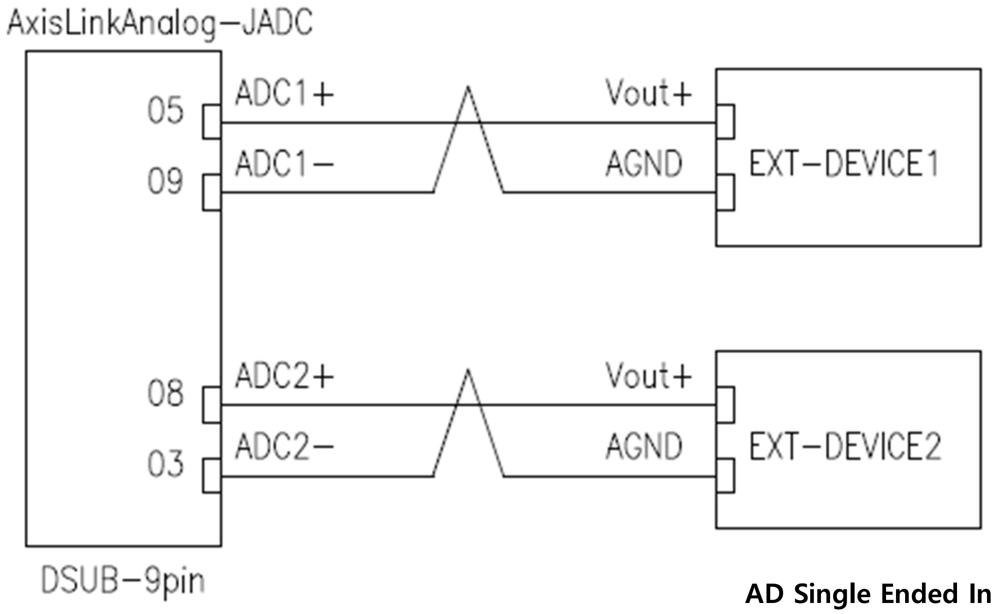


DAC Single Ended Out

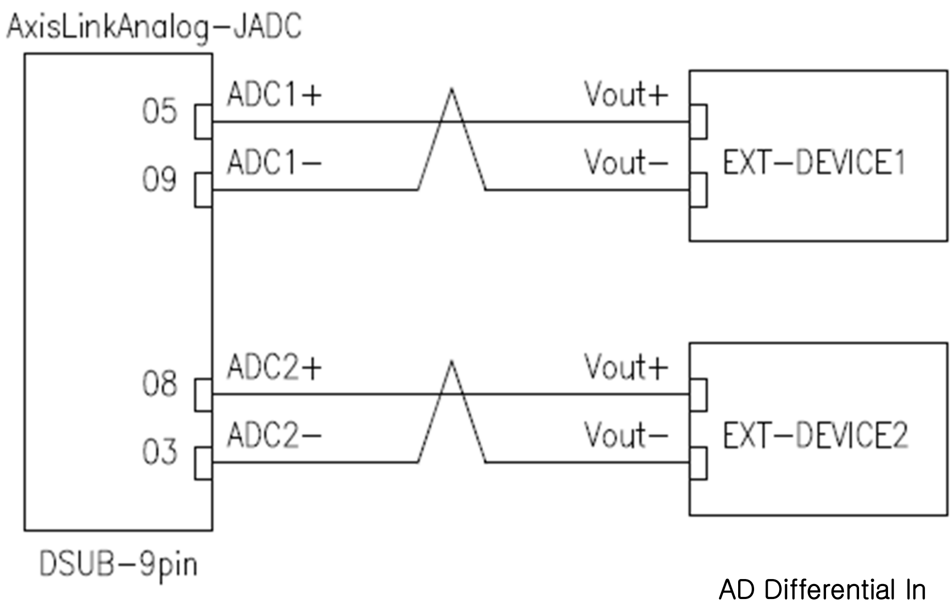


DAC Differential Out

3.2.2. J_ADC Connection



AD Single Ended In

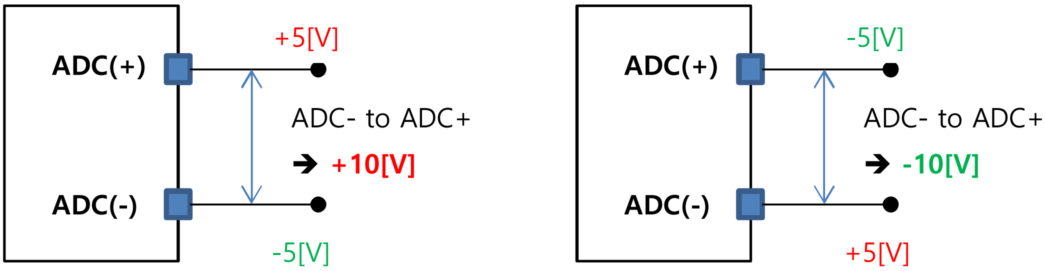


AD Differential In

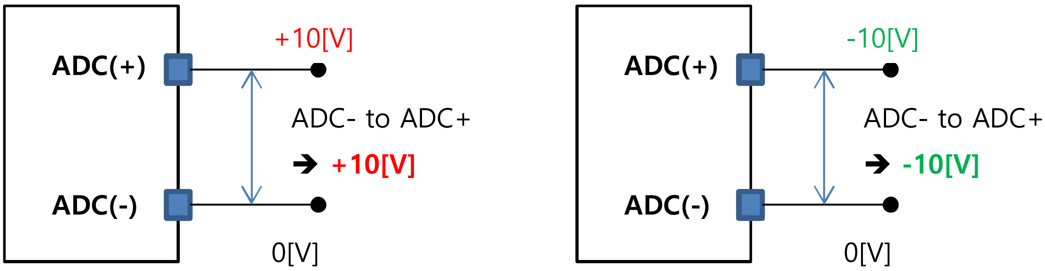
3.3. SW Setup for ADC Input

A. -5~+5[V] Bipolar Input

Range : ±5[V], Pole : Bipolar
→ **ADC(-) to ADC(+) Input Range**
= -10[V] ~ +10[V]



<Differential Ended Input>



<Single Ended Input>

E3(for CH1,2)
E4(for CH3,4)
→2-3 Short(±5[V])

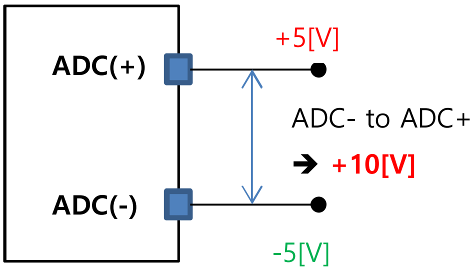
S_CHn-1
→ OFF(Bipolar)
S_CHn-2
→ ON(±5[V])

B. -5~+5[V] Unipolar Input

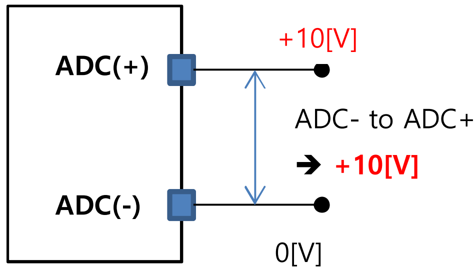
Range : ±5[V], Pole : unipolar

➔ **ADC(-) to ADC(+) Input Range**

= 0[V] ~ +10[V]



<Differential Ended Input>



<Single Ended Input>

E3(for CH1,2)

E4(for CH3,4)

➔2-3 Short(±5[V])

S_CHn-1

➔ ON(Unipolar)

S_CHn-2

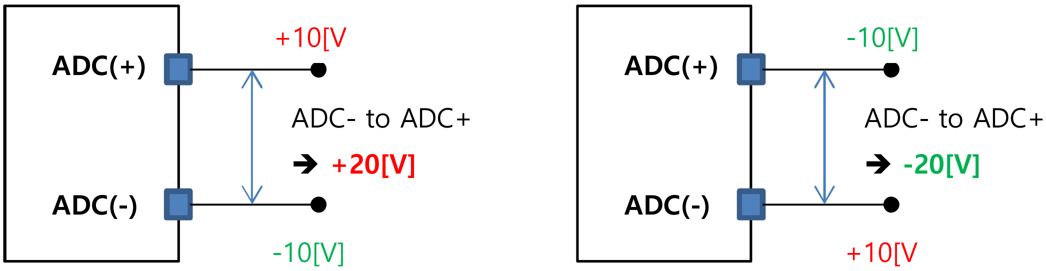
➔ ON(±5[V])

C. -10~+10[V] Bipolar Input

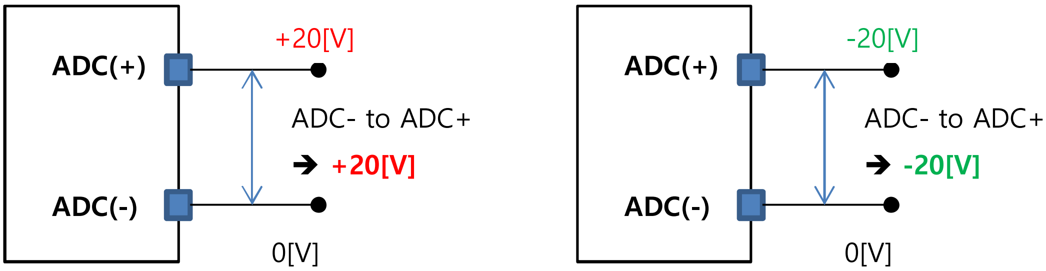
Range : ±10[V], Pole : Bipolar

➔ **ADC(-) to ADC(+) Input Range**

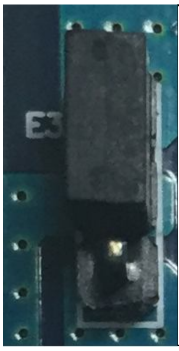
= -20[V] ~ +20[V]



<Differential Ended Input>



<Single Ended Input>



E3(for CH1,2)

E4(for CH3,4)

➔**1-2 Short(±10[V])**

S_CHn-1

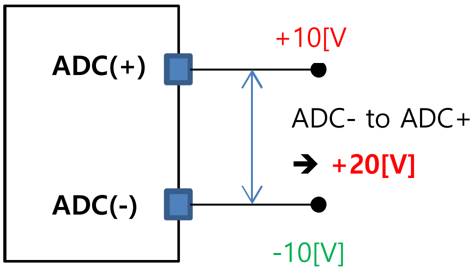
➔ **OFF(Bipolar)**

S_CHn-2

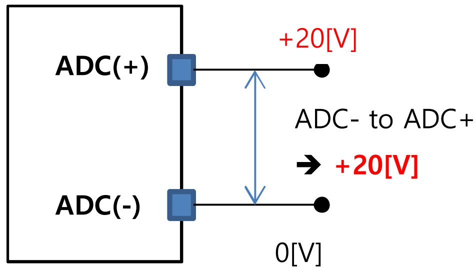
➔ **OFF(±10[V])**

D. -10[V] ~ +10[V] Unipolar Input

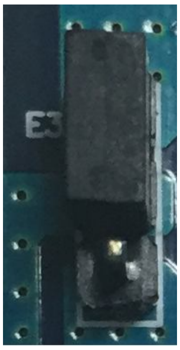
Range : ±10[V], Pole : unipolar
➔ **ADC(-) to ADC(+) Input Range**
= 0[V] ~ +20[V]



<Differential Ended Input>



<Single Ended Input>



E3(for CH1,2)
E4(for CH3,4)
➔1-2 Short(±10[V])

S_CHn-1
➔ ON(Unipolar)
S_CHn-2
➔ OFF(±10[V])

4. ADC CONVERTING VALUE

4.1. E3/E4 : 2-3 Short, S_CH-1 : OFF, S_CH-2 : ON

- **Bipolar Convert**
- **±5[V] Range : ADC- to ADC+ ➔ -10[V] ~ +10[V]**

A. Differential Ended Input

$$\textbf{Input ADC Value} = \frac{(\textbf{input voltage} + 5)[\textbf{v}]}{10[\textbf{v}]} \times (2^{16} - 1)$$

Input Voltage	Calculation	ADC Value
-5[V](Differential)	0/10*65535	0
0[V](Differential)	5/10*65535	32767
5[V](Differential)	10/10*65535	65535

$$\textbf{Input Voltage} = \textbf{Input ADC Value} \div (2^{16} - 1) \times 10 - 5$$

```
//Ex) Base Add : $6C000, CH1 Input Voltage
#define ADC_CH1                 M4000
#define INPUT_VOLTAGE           P100

ADC_CH1->Y:$6C082,8,16,U
INPUT_VOLTAGE = ADC_CH1/$FFFF*10-5
```

B. Single Ended Input

$$\textbf{Input ADC Value} = \frac{(\textbf{input voltage} + 10)[\textbf{v}]}{20[\textbf{v}]} \times (2^{16} - 1)$$

Input Voltage	Calculation	ADC Value
-10[V](Single)	0/20*65535	0
0[V](Single)	10/20*65535	32767
10[V](Single)	20/20*65535	65535

$$\textbf{Input Voltage} = \textbf{Input ADC Value} \div (2^{16} - 1) \times 20 - 10$$

```
//Ex) Base Add : $6C000, CH1 Input Voltage
#define ADC_CH1                 M4000
#define INPUT_VOLTAGE           P100

ADC_CH1->Y:$6C082,8,16,U
INPUT_VOLTAGE = ADC_CH1/$FFFF*20-10
```

4.2. E3/E4 : 2-3 Short, S_CH-1 : ON, S_CH-2 : ON

- **Unipolar Convert**
- **±5[V] Range : ADC- to ADC+ → 0[V] ~ +10[V]**

A. Differential Ended Input

$$\textbf{Input Value} = \frac{\textbf{input votage[v]}}{\textbf{5[v]}} \times (\mathbf{2^{16} - 1})$$

Input Voltage	Calculation	ADC Value
0[V](Differential)	0/5*65535	0
2.5[V](Differential)	2.5/5*65535	32767
5[V](Differential)	5/5*65535	65535

$$\textbf{Input Voltage} = \textbf{Input ADC Value} \div (\mathbf{2^{16} - 1}) \times \mathbf{5}$$

```
//Ex) Base Add : $6C000, CH1 Input Voltage
#define ADC_CH1                 M4000
#define INPUT_VOLTAGE           P100

ADC_CH1->Y:$6C082,8,16,U
INPUT_VOLTAGE = ADC_CH1/$FFFF*5
```

B. Single Ended Input

$$\textbf{Input Value} = \frac{\textbf{input votage[v]}}{\textbf{10[v]}} \times (\mathbf{2^{16} - 1})$$

Input Voltage	Calculation	ADC Value
0[V](Single)	0/10*65535	0
5[V](Single)	5/10*65535	32767
10[V](Single)	10/10*65535	65535

$$\textbf{Input Voltage} = \textbf{Input ADC Value} \div (\mathbf{2^{16} - 1}) \times \mathbf{10}$$

```
//Ex) Base Add : $6C000, CH1 Input Voltage
#define ADC_CH1                 M4000
#define INPUT_VOLTAGE           P100

ADC_CH1->Y:$6C082,8,16,U
INPUT_VOLTAGE = ADC_CH1/$FFFF*10
```

4.3. E3/E4 : 1-2 Short, S_CH-1 : OFF, S_CH-2 : OFF

- **Bipolar Convert**
- **±10[V] Range : ADC- to ADC+ ➔ -20[V] ~ +20[V]**

A. Differential Ended Input

$$\textbf{Input ADC Value} = \frac{(\textbf{input voltage} + \textbf{10})[\textbf{v}]}{\textbf{20}[\textbf{v}]} \times (\textbf{2}^{\textbf{16}} - \textbf{1})$$

Input Voltage	Calculation	ADC Value
-10[V](Differential)	0/20*65535	0
0[V](Differential)	10/20*65535	32767
10[V](Differential)	20/20*65535	65535

$$\textbf{Input Voltage} = \textbf{Input ADC Value} \div (\textbf{2}^{\textbf{16}} - \textbf{1}) \times \textbf{20} - \textbf{10}$$

```
//Ex) Base Add : $6C000, CH1 Input Voltage
#define ADC_CH1                  M4000
#define INPUT_VOLTAGE            P100

ADC_CH1->Y:$6C082,8,16,U
INPUT_VOLTAGE = ADC_CH1/$FFFF*20-10
```

B. Single Ended Input

$$\textbf{Input ADC Value} = \frac{(\textbf{input voltage} + \textbf{20})[\textbf{v}]}{\textbf{40}[\textbf{v}]} \times (\textbf{2}^{\textbf{16}} - \textbf{1})$$

Input Voltage	Calculation	ADC Value
-20[V](Single)	0/40*65535	0
0[V](Single)	20/40*65535	32767
20[V](Single)	40/40*65535	65535

$$\textbf{Input Voltage} = \textbf{Input ADC Value} \div (\textbf{2}^{\textbf{16}} - \textbf{1}) \times \textbf{40} - \textbf{20}$$

```
//Ex) Base Add : $6C000, CH1 Input Voltage
#define ADC_CH1                  M4000
#define INPUT_VOLTAGE            P100

ADC_CH1->Y:$6C082,8,16,U
INPUT_VOLTAGE = ADC_CH1/$FFFF*40-20
```

4.4. E3/E4 : 1-2 Short, S_CH-1 : ON, S_CH-2 : OFF

- **Unipolar Convert**
- **±10[V] Range : ADC- to ADC+ ➔ 0[V] ~ +20[V]**

A. Differential Ended Input

$$\textbf{Input Value} = \frac{\textbf{input votage[v]}}{\textbf{10[v]}} \times (\mathbf{2^{16} - 1})$$

Input Voltage	Calculation	ADC Value
0[V](Differential)	0/10*65535	0
5[V](Differential)	5/10*65535	32767
10[V](Differential)	10/10*65535	65535

$$\textbf{Input Voltage} = \textbf{Input ADC Value} \div (\mathbf{2^{16} - 1}) \times \mathbf{10}$$

```
//Ex) Base Add : $6C000, CH1 Input Voltage
#define ADC_CH1                 M4000
#define INPUT_VOLTAGE           P100

ADC_CH1->Y:$6C082,8,16,U
INPUT_VOLTAGE = ADC_CH1/$FFFF*10
```

B. Single Ended Input

$$\textbf{Input Value} = \frac{\textbf{input votage[v]}}{\textbf{20[v]}} \times (\mathbf{2^{16} - 1})$$

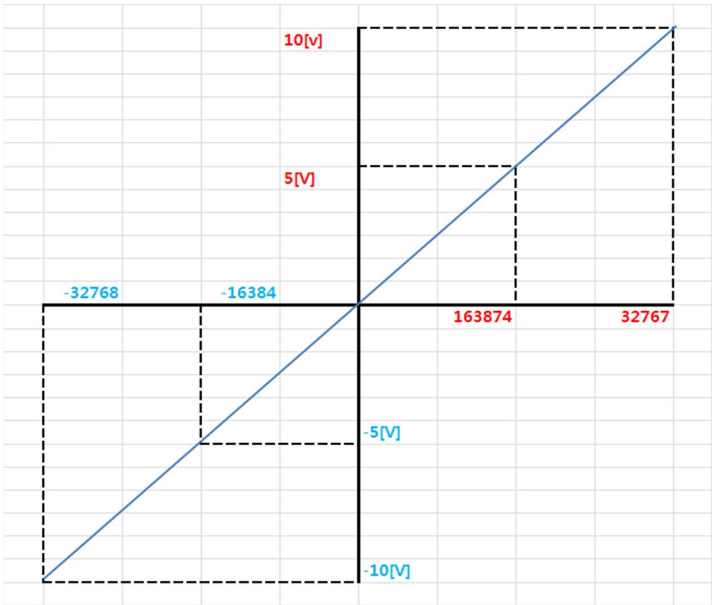
Input Voltage	Calculation	ADC Value
0[V](Single)	0/20*65535	0
10[V](Single)	10/20*65535	32767
20[V](Single)	20/20*65535	65535

$$\textbf{Input Voltage} = \textbf{Input ADC Value} \div (\mathbf{2^{16} - 1}) \times \mathbf{20}$$

```
Ex) Base Add : $6C000, CH1 Input Voltage
#define ADC_CH1                 M4000
#define INPUT_VOLTAGE           P100

ADC_CH1->Y:$6C082,8,16,U
INPUT_VOLTAGE = ADC_CH1/$FFFF*20
```

5. DAC CONVERTING VALUE

- AxisLnik-Analog의 DAC Data는 16bit signed integer 형식입니다.
- DAC 출력 범위는 **Differential -10[V] ~ +10[V]** 입니다.
- 연결되는 장치가 **Differential**일 경우 **DAC(-)** 핀을 연결하시고 **Single Ended**일 경우 **AGND** 핀을 연결하시기 바랍니다.



$$\mathbf{Out\ Voltage}[V] = \frac{\mathbf{DAC\ Data}}{(\mathbf{2^{16}-1})} \times \mathbf{10}[V]$$

DAC Data	DAC(-) to DAC(+)	AGND to DAC(+)
-32768	-20[V]	-10[V]
0	0[V]	0[V]
32768	20[V]	10[V]

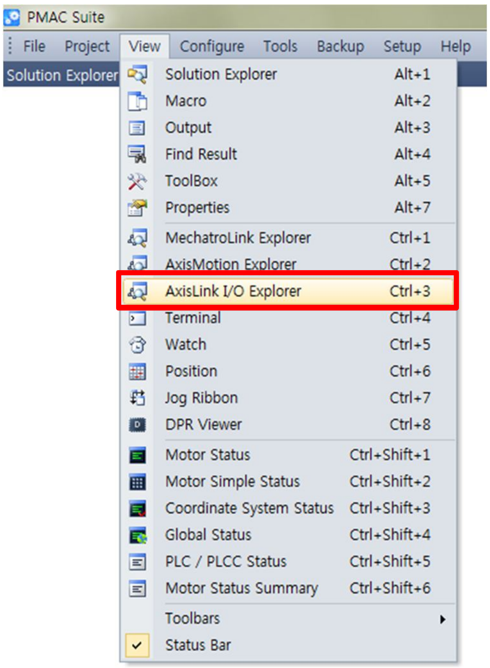
$$\mathbf{DAC\ Data} = \frac{\mathbf{Out\ Voltage}[V]}{\mathbf{10}[V]} \times (\mathbf{2^{16}-1})$$

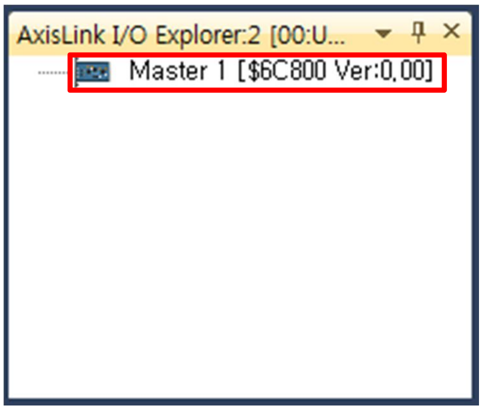
```
Ex) Base Add : $6C000, CH1 DAC Data
#define DAC_CH1                 M7000
#define OUTPUT_VOLTAGE          P200


DAC_CH1->Y:$6C042,8,16,U
DAC_CH1= OUTPUT_VOLTAGE /10*$7FFF
```

6. AXIS LINK I/O EXPLORER

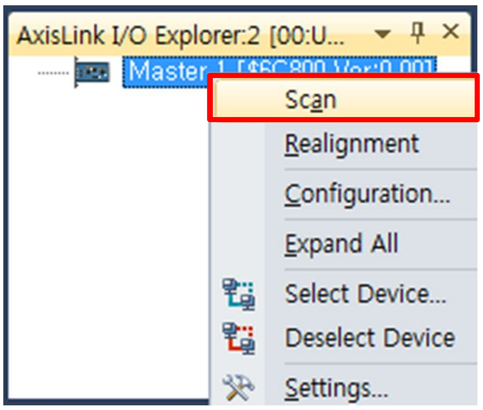
- PMAC Suit에 내장된 AxisLink I/O Explorer를 사용하면 기본적인 동작상태 및 하드웨어 이상 유무를 체크할 수 있습니다.

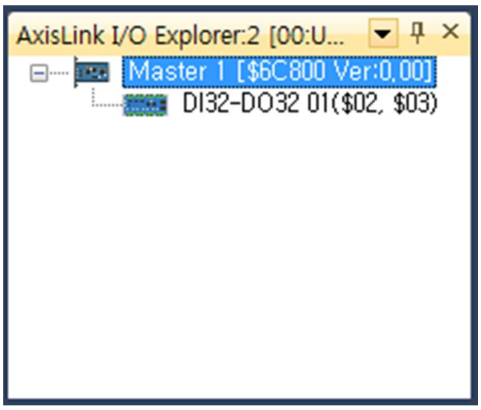
A. AxisLink I/O Explorer를 실행합니다. AxisLink I/O Explorer에서 Master 카드가 검색되면 Base Address를 확인할 수 있습니다.



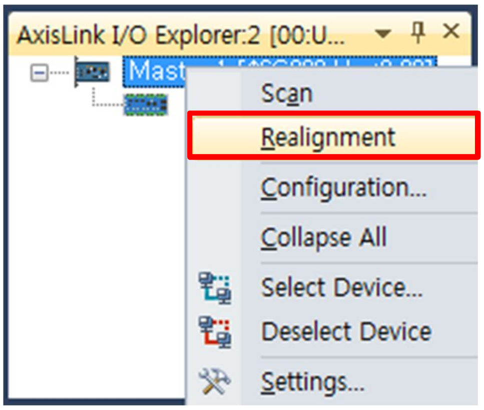


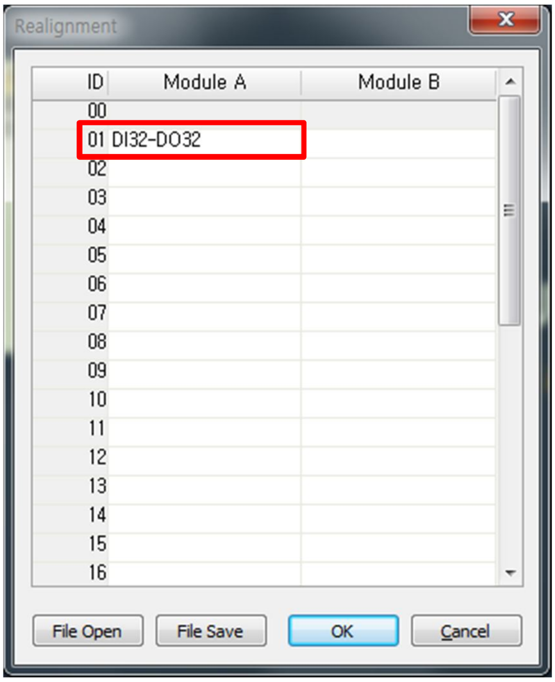
B. Master를 클릭한 후 마우스 오른쪽 버튼을 눌러 Scan을 클릭합니다. AxisLink-Analog가 검색되면 초기에 AxisLink-32 digital I/O 보드로 인식됩니다.



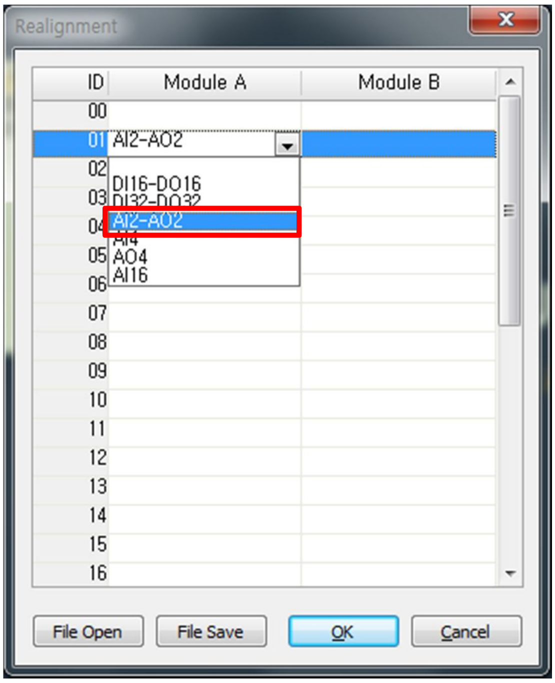


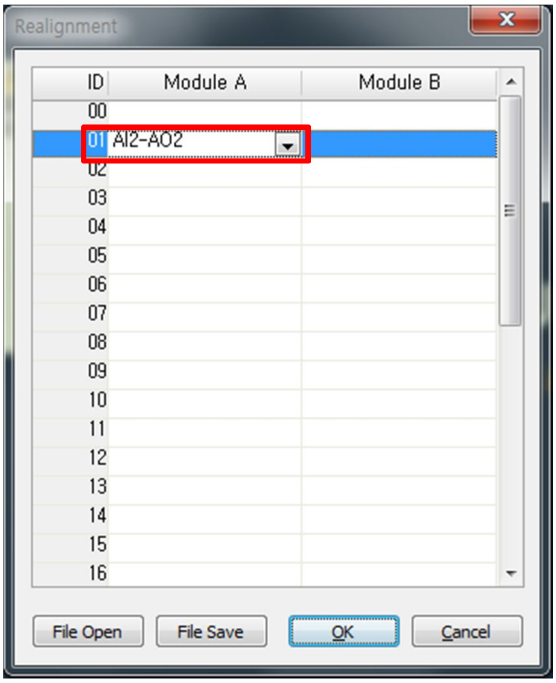
C. Master를 클릭한 후 마우스 오른쪽 버튼을 눌러 Realignment를 클릭합니다.



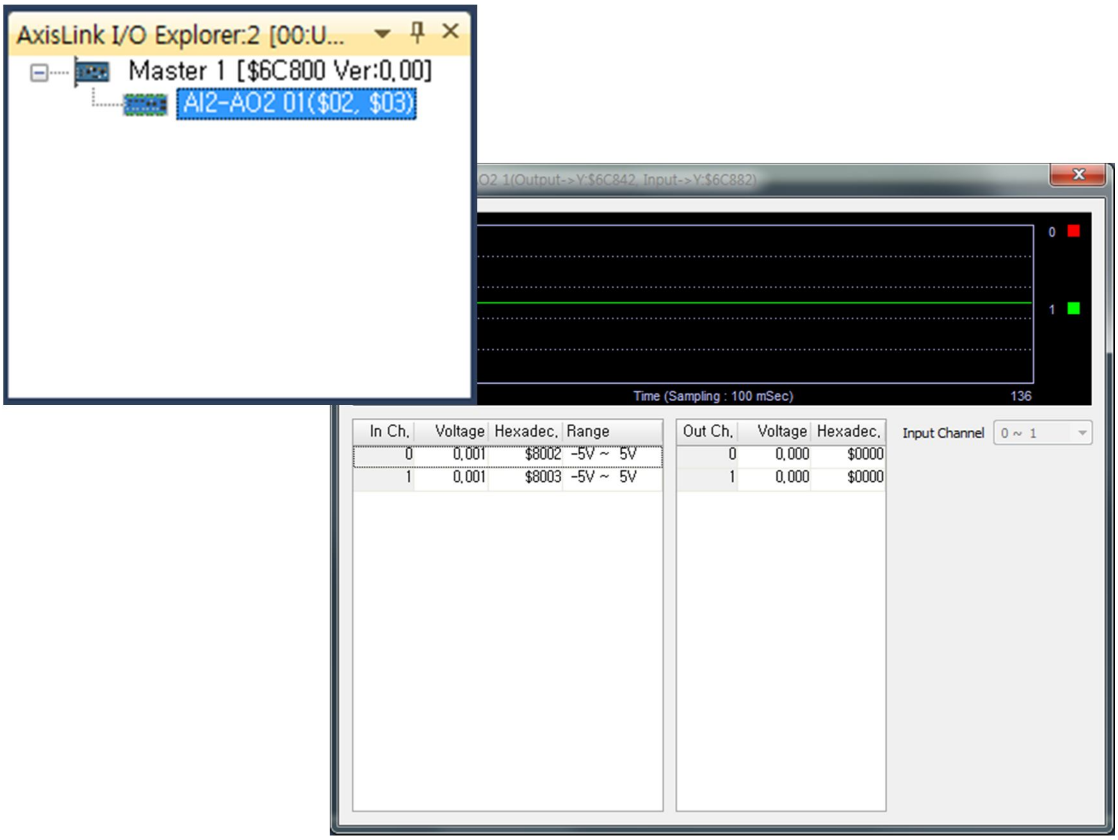


- DI32-DO32(digital I/O 32)로 표시된 부분을 클릭하여 AI2-AO2(Analog In2 Out2)로 변경한 후 OK 버튼을 누릅니다.

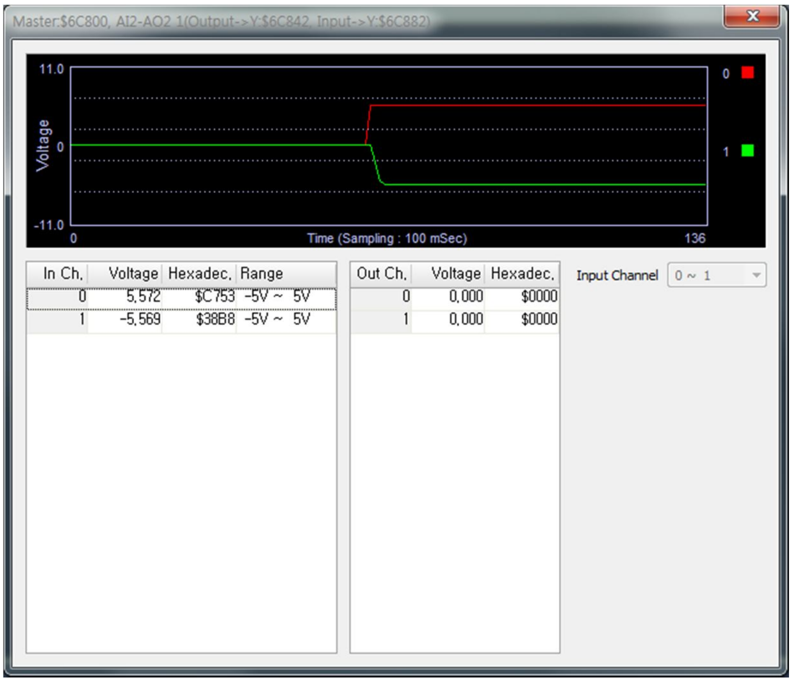




D. Slave를 더블 클릭하면 AxisLink Analog 보드를 Test 하실 수 있습니다.



In Ch와 그래프로 ADC 입력 값이 표시되며 Out Ch에 값을 입력하면 DAC 출력을 Test할 수 있습니다.



7. EXAMPLE

7.1. Initialize PLC

```
//////////////////////////////////////////////////////////////////////////////
// Controller가 Cliper 또는 Cruiser일 경우 Base Address는 $6C0000이며
// UMAC을 사용 할 경우 반드시 Axis Link I/O Explorer를통해
// Base Address를 확인 하시기 바랍니다.
//////////////////////////////////////////////////////////////////////////////
#define UMAC         // Clipper나Cruiser 사용시 주석처리 합니다.

#ifdef UMAC                    // Ex) UMAC Base Add = $6C800
       #define AXISIO_SCR_ADR        X:$6C8C0,8,16
       #define AXISIO_SSR_ADR        X:$6C8C1,8,16
       #define AXISIO_BCR_ADR        X:$6C8C7,8,16
#else                          // Ex) Clipper or Cruiser = $6C000
       #define AXISIO_SCR_ADR        X:$6C0C0,8,16
       #define AXISIO_SSR_ADR        X:$6C0C1,8,16
       #define AXISIO_BCR_ADR        X:$6C0C7,8,16
#endif


#define M_AXISIO_SCR         M5000
#define M_AXISIO_SSR         M5001
#define M_AXISIO_BCR         M5002

M_AXISIO_SCR->AXISIO_SCR_ADR
M_AXISIO_SSR->AXISIO_SSR_ADR
M_AXISIO_BCR->AXISIO_BCR_ADR


OPEN PLC 1 CLEAR
// 메모리를 초기화합니다.

#ifdef UMAC
  CMD"WL:$6c800,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0"
  CMD"WL:$6c840,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0"
  CMD"WL:$6c880,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0"
  CMD"WY:$6c8C0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0"
  CMD"WY:$6c900,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0"
  CMD"WY:$6c940,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0"
  CMD"WY:$6c980,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0"
  CMD"WY:$6c9C0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0"
#else
  CMD"WL:$6c000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0"
  CMD"WL:$6c040,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0"
  CMD"WL:$6c080,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0"
  CMD"WY:$6c0C0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0"
  CMD"WY:$6c100,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0"
  CMD"WY:$6c140,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0"
  CMD"WY:$6c180,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0"
  CMD"WY:$6c1C0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0"
#endif
```

```
//메모리를 안정적으로 초기화하기 위해 일정시간 기다립니다.
  I5111 = 50*8388608/I10
  While (I5111 > 0)   Endwhile

//통신 속도 및 Scna할 Slave 개수를 지정합니다.
  M_AXISIO_BCR = $13  // Full duplex, 12Mbps (Bit4 = 1, Bit1..0 = 11)
  M_AXISIO_SCR = 5    // Slave ID 2번까지연속스캔을사용할경우(Slave ID*2+1)
  DISABLE PLC 1
CLOSE
```

7.2. ADC Voltage Calculate PLC

```
/////////////////////////////////////////////////////////////////////////////
// 아래항목은E3, E4 점퍼및S-CH 설정에따라값을변경해주십시오.
/////////////////////////////////////////////////////////////////////////////
// A. INPUT_TYPE      1 : Single Ended Input
//                    2 : Differential Ended Input
#define CH1_TYPE             1       // CH1의입력방식선택

#define CH2_TYPE             1       // CH2의입력방식선택
/////////////////////////////////////////////////////////////////////////////
// B. INPUT_POLE      1 : Bipolar
//                    2 : Unipolar
#define CH1_POLE             1       // CH1의컨버팅방식선택

#define CH2_POLE             1       // CH2의컨버팅방식선택
/////////////////////////////////////////////////////////////////////////////
// C. INPUT_RANGE     1 : ±[V](ADC(-) to ADC(+) -> -10[V] ~ +10[V])
//                    2 : ±[V](ADC(-) to ADC(+) -> -20[V] ~ +20[V])
#define CH1_RANGE            1       // CH1의전압범위선택

#define CH2_RANGE            1       // CH1의전압범위선택
/////////////////////////////////////////////////////////////////////////////
#define UMAC   // Clipper나Cruiser 사용시주석처리합니다.

#ifdef UMAC           // Ex) UMAC Base Add = $6C800
       #define ADC1_ADR      Y:$6C882,8,16,U
       #define ADC2_ADR      Y:$6C883,8,16,U
#else                 // Ex) Clipper or Cruiser = $6C000
       #define ADC1_ADR      Y:$6C082,8,16,U
       #define ADC2_ADR      Y:$6C083,8,16,U
#endif

#define ADC_DATA1            M4000
#define ADC_DATA2            M4001

ADC_DATA1->ADC1_ADR
ADC_DATA2->ADC2_ADR

#define INPUT_TEMP_1         P1000
#define INPUT_TEMP_2         P1001

#define INPUT_VOLT_1         P2000
#define INPUT_VOLT_2         P2001


Open PLC 2 clear
       INPUT_TEMP_1 = (ADC_DATA1/$FFFF*(20/CH1_POLE)-(2-CH1_POLE)*10)
       INPUT_TEMP_2 = (ADC_DATA2/$FFFF*(20/CH2_POLE)-(2-CH2_POLE)*10)
       INPUT_VOLT_1 = INPUT_TEMP_1/CH1_TYPE*CH1_RANGE
       INPUT_VOLT_2 = INPUT_TEMP_2/CH2_TYPE*CH2_RANGE
Close
```

7.3. DAC Data Calculate PLC

```
#define UMAC   // Clipper나Cruiser 사용시주석처리합니다.

#ifdef UMAC            // Ex) UMAC Base Add = $6C800
       #define DAC1_ADR      Y:$6C842,8,16,S
       #define DAC2_ADR      Y:$6C843,8,16,S
#else                  // Ex) Clipper or Cruiser = $6C000
       #define DAC1_ADR      Y:$6C042,8,16,S
       #define DAC2_ADR      Y:$6C043,8,16,S
#endif

#define DAC_DATA1                  M7000
#define DAC_DATA2                  M7001

DAC_DATA1->DAC1_ADR
DAC_DATA2->DAC2_ADR

#define OUTPUT_VOLT_1        P3000
#define OUTPUT_VOLT_2        P3001

open plc 3 clear
       DAC_DATA1 = (OUTPUT_VOLT_1/10)*$7FFF
       DAC_DATA2 = (OUTPUT_VOLT_2/10)*$7FFF
close
```