

제품 매뉴얼

Smart I/O-Series

(Smart I/O-II, III)

Industrial Embedded Computer

(주)에이치엔에스 홈페이지: www.hnsts.co.kr
(주)에이치엔에스 쇼핑몰: www.hnsstore.co.kr
SmartX Framework 홈페이지: www.smartx.co.kr

업데이트 : 2023. 3. 9

Smart I/O-Series

사용 방법 요약 및 필수 숙지 사항



Smart I/O-Series의 Base Board 선택, IEC-Series와 연결, 전원 연결 등 Smart I/O-Series를 처음 사용할 때 확인하여 개발 환경을 구축 할 수 있는 내용이 요약되어 있으니 참고하시기 바랍니다.

STEP-1

Smart I/O Base Board 선택

[요약 설명]

- Smart I/O의 Base Board에는 Smart I/O-II, Smart I/O-III 두 가지 모델이 있습니다. 해당 모델들은 서로 다른 차이점이 있으며, 사용 할 구성 및 환경에 따라 Base Board를 선택하시기 바랍니다.

[주요 확인 내용 참조]

- PART-I. Smart I/O-Series 시작하기 → 3. Smart I/O-II, III의 차이점 비교
- PART-I. Smart I/O-Series 시작하기 → 5. Smart I/O-II, III 공통사항

STEP-2

IEC-Series별 Extension Port 특성에 따른 입/출력 Port 선택

[요약 설명]

- IEC-Series별 Extension Port의 각 Port마다 초기 상태 값(High/Low)이 다르기 때문에 입력 Port와 출력 Port를 선택하여 사용해야 합니다.

[주요 확인 내용 참조]

- PART-I. Smart I/O-Series 시작하기 → 5-4. IEC-Series별 Extension Port 특성에 따른 입/출력 Port 선택

STEP-3

Smart I/O-Block Series 장착 위치 및 방법

[요약 설명]

- Smart I/O-II와 Smart I/O-III는 각각 Block의 장착 위치가 정해져 있습니다. 원하는 기능의 Block을 알맞은 위치에 올바른 과정으로 장착합니다.

[주요 확인 내용 참조]

- PART-I. Smart I/O-Series 시작하기 → 5-2. Block-Series 장착 방법
- PART-I. Smart I/O-Series 시작하기 → 4. Smart I/O Base Board 각부 명칭

STEP-4

IEC-Series와 Smart I/O-Series의 연결 및 전원 연결

[요약 설명]

- Block 장착이 완료된 Smart I/O-Series와 IEC-Series를 IDC Cable을 사용하여 Extension Port를 연결합니다.
- Smart I/O-II, III 모두 기본적으로 AC 전원(AC 220V)을 사용합니다. 하지만 DC 전원만 사용할 수 있는 환경인 경우 DC 전원(DC 12V)을 사용할 수 있는 개조방법 또한 설명하고 있습니다.

[주요 확인 내용 참조]

- PART-II. Smart I/O-II Base Board → 2. IEC-Series와 Smart I/O-II 제품 연결하기(케이블 연결)
- PART-III. Smart I/O-III Base Board → 2. IEC-Series와 Smart I/O-III 제품 연결하기(케이블 연결)
- PART-I. Smart I/O-Series 시작하기 → 5-1. 전원 연결하기

STEP-5

외부 인터페이스 연결 예시 및 예제 프로그램

[요약 설명]

- 외부 인터페이스 장비를 Block에 연결하고 관련 예제 프로그램을 사용하여 동작하는지 확인합니다.

[주요 확인 내용 참조]

- PART-IV. Smart I/O-Block Series 호환성 → 각 Block의 Block 응용 방법, 언어별 주요소스 코드 및 응용 예제

STEP-6

Smart I/O-Series 사용 시 주요문제 해결 방법

[요약 설명]

- Smart I/O-Series를 사용하면서 발생할 수 있는 주요 문제들에 대하여 해결 방법을 제시하고 있습니다.

[주요 확인 내용 참조]

- PART-I. Smart I/O-Series 시작하기 → 6. Smart I/O-Series 사용 시 주요문제 해결 방법

주의 사항 - I

본 사용 설명서의 저작권은 (주)에이치앤에스에 있습니다.

본 사용 설명서의 내용 중 일부 또는 전부를 다른 목적으로 복제 또는 복사를 할 수 없습니다.

본 사용 설명서의 내용은 품질 향상을 위해서 사전 통보 없이 변경될 수 있습니다. 변경된 사용 설명서는 당사 홈페이지 (www.hnsts.co.kr)에서 확인하시기 바랍니다.

안전을 위한 주의 사항

제품을 사용하기 전에 반드시 본 사용 설명서를 충분히 읽어 본 후 사용하시기 바랍니다. 본 사용 설명서를 숙지하지 않은 상태에서 발생된 모든 피해에 대해 당사에서는 일체의 책임을 지지 않으므로 주의하십시오.

지정된 규격품 이외의 시스템을 사용하여 발생한 손상 및 본 사용 설명서의 사용 방법과 주의 사항을 따르지 않아 시스템을 손상시켰을 때는 당사에서 책임지지 않으므로 주의하십시오.

특히 안전에 대한 주의 사항은 제품을 올바르게 사용하여 사고나 위험을 예방하기 위한 내용이오니 반드시 지켜주시기 바랍니다.

1. 제품의 규격은 품질 개선을 위하여 임의로 사양이 변동될 수 있습니다.
2. 전원이 인가된(전류가 흐르는) 상태에서는 단자대를 만지지 마십시오. 감전의 원인이 됩니다.
3. 배선 작업 전 시스템 관련 모든 전원이 차단된(꺼진) 상태에서 배선 작업을 하시기 바랍니다. 감전 또는 제품 손상의 원인이 됩니다. 또한 잘못된 배선은 제품이 파손되거나 오작동의 원인이 될 수 있습니다.
4. 외부 전원 또는 제품의 이상 발생시에 전체 제어 시스템을 보호하기 위해 IEC-Series의 외부에 보호 회로를 구성하여 사용하시기 바랍니다.
5. 오동작으로 인해 전체 시스템의 안전성 또는 인체에 심각한 문제를 초래할 수 있으며 잘못 취급하였을 경우 사용자가 사망하거나 중상을 입는 위험 상태가 발생할 수 있습니다.
6. 시스템의 손상 및 오작동으로 발생하는 피해를 보호할 수 있도록 외부에 비상 정지 스위치, 안전 회로, 보호 회로 등과 같은 안전 장치를 설치하여 사용하시기 바랍니다. 잘못 취급하여 사용자가 상해를 입거나 물적 손해 등 위험 상황이 발생할 것으로 예상되는 경우 비상 정지, 인터록 회로를 외부 회로에서 구성하시기 바랍니다.
7. 인체 사고나 중대한 손해로 확대될 것으로 예측되는 용도로 사용할 경우에는 이종 안전장치 등 안전 대책을 세워주시기 바랍니다.
8. 전선은 단자 나사로 확실히 조여주십시오. 접촉이 불량일 경우 이상 발열이나 고장의 원인이 됩니다.
9. 정격 사양 범위 이외에서는 사용하지 마십시오. 이상 발열이나 고장의 원인이 됩니다.
10. 분해나 개조하지 마십시오. 감전이나 고장의 원인이 되어 시스템에 문제가 발생할 경우 당사는 책임을 지지 않으므로 주의하십시오.
11. 진동이 많은 환경에서 사용하는 경우는 브래킷을 나사로 확실하게 조여주시기 바랍니다. 해당 조치 없이 사용하는 경우 제품에 직접 진동이 가해져 오동작, 단락, 낙하 등의 원인이 됩니다.

주의 사항 - II

쇼트 / 정전기 주의

- 쇼트의 가능성이 존재하므로 작업 환경 주변에 전도성이 높은 물체를 두지 않습니다.
- 작업 환경이 건조한 경우 정전기가 발생할 수 있으므로 작업 전 절연 장갑을 착용합니다.
- 전기가 흐르는 전선의 피복이 벗겨져 있는 경우 절연체로 감아줍니다.

전기 사용 시 주의

- AC(220V) 또는 DC(12V) 미 입력 시 각 기능들이 정상 동작 안 함(ADC, DAC, RS232, Relay)
- Smart I/O 전원 연결은 DC인 경우 12V(+,- 극성 주의), AC인 경우 220V을 사용하시기 바랍니다.
- DC 전원 사용 시 극성이 맞게 되었는지 확인합니다. (+, -)

점퍼, 결선

- IEC-Series의 非Lite/Lite 종류에 따른 Smart I/O의 점퍼 연결은 올바른지 확인합니다.
- IEC-Series와 Smart I/O 확장 케이블(Extension Port) 연결 시 포트와 맞는 케이블이 바뀌지 않도록 주의합니다.
- 각종 센서 사용 시 전기를 인가하기 전에 최초 계획한 대로 결선이 맞는지 확인합니다.

Block 사용

- Block 사용 시 타사 제품과 함께 사용할 수 없습니다. (I/O 보드와 호환되도록 설계됨)
- Block 사용 시 전기적인 사양을 유의하여 사용해야 합니다. (ex: ADC는 0~5V, 0~10V 입력)

제품 매뉴얼의 내용 중간에 주의, 중요, 참고, 참조, SmartX Framework, 권장과 같이 매뉴얼을 쉽게 이해할 수 있도록 중요한 내용들을 따로 정리하여 표시하고 있으니 반드시 읽어보시기 바랍니다.

주의 제품 사용에 있어 주의해야 하는 내용을 설명합니다. 해당 내용을 필독하여 지시문에 따라 개발 환경에 적용해야 하며, 주의 사항을 따르지 않아 발생하는 모든 피해는 당사에서 책임 지지 않으므로 주의하시기 바랍니다.

중요 매뉴얼의 내용 중 반드시 확인하여 적용해야 하는 내용을 설명합니다. 해당 내용을 필독하여 지시문에 따라 개발 환경에 적용하시기 바랍니다.

참고 매뉴얼 내용의 이해를 돋고 강조하는 내용을 설명합니다. 해당 내용은 매뉴얼을 이해할 때 참고하시기 바랍니다.

참조 현재 위치가 아닌 다른 위치의 내용 혹은 해당 매뉴얼 이외의 내용을 참조하여 같이 봐야하는 내용을 설명합니다. 해당 내용은 현재 위치의 내용만으로 이해가 어려운 경우 참조하시기 바랍니다.

SmartX Framework SmartX Framework의 기능을 SmartX Framework 매뉴얼에서 찾아볼 수 있도록 설명합니다. 해당 내용은 SmartX Framework의 기능을 요약 설명한 것으로 SmartX 홈페이지(www.smartx.co.kr)에서 참조하시기 바랍니다.

권장 매뉴얼의 내용 중 권장하는 사용 방법에 대해서 설명합니다. 해당 내용을 읽어본 후 개발 환경에 적용 할 때 적극적으로 반영해 보시기 바랍니다.

목차

1

Smart I/O-Series 시작하기

1. Smart I/O-Series 소개 및 구성	010
2. 인터페이스의 전원 분리(Interface Isolation)	011
3. Smart I/O-II, III의 차이점 비교	012
4. Smart I/O Base Board 각부 명칭	013
5. Smart I/O-II, III 공통사항	015
5-1. 전원 연결하기	015
5-2. Block-Series 장착 방법	020
5-3. Smart I/O-II, III 기구 디자인 가이드	022
5-4. IEC-Series별 Extension Port 특성에 따른 입/출력 Port 선택	024
6. Smart I/O-Series 사용 시 주요문제 해결 방법	026
6-1. Smart I/O Base Board에 전원이 입력되었는지 확인	026
6-2. Smart I/O Base Board에 체결된 Block의 위치가 올바른지 확인	027
6-3. Extension Port 연결에 문제가 없는지 확인	027
6-4. 결선에 문제가 없는지 확인	028
6-5. Block의 LED등으로 입출력 신호를 확인	028
6-6. 관련 프로그램 코드가 올바른지 확인	029
7. Smart I/O-I 제품 단종에 따른 Smart I/O-II 제품으로 Smart I/O-I과 같이 구성하는 방법	029

2

Smart I/O-II Base Board

1. 외형 치수	032
2. IEC-Series와 Smart I/O-II 제품 연결하기(케이블 연결)	033
3. Smart I/O-II Base Board의 핀맵 안내 및 Header 커넥터 연결 상태	034
3-1. Smart I/O-II Base Board 호환성	034
3-2. Block 커넥터와 Header 커넥터의 연결 상태 및 명칭	034
3-3. Smart I/O-II의 Block 장착 위치	036
3-4. Smart I/O-II Base Board 연결 정보(설명) 및 전기적 사양	036
3-5. Smart I/O-II Base Board 기본 내부 기능	037
4. Smart I/O-II 커넥터 사양 안내	039

3

Smart I/O-III Base Board

1. 외형 치수	042
2. IEC-Series와 Smart I/O-III 제품 연결하기(케이블 연결)	043

3. Smart I/O-III Base Board의 핀맵 안내 및 Header 커넥터 연결 상태	044
3-1. Smart I/O-III Base Board 호환성	044
3-2. Block 커넥터와 Header 커넥터의 연결 상태 및 명칭	044
3-3. Smart I/O-III의 Block 장착 위치	046
3-4. Smart I/O-III Base Board 연결 정보(설명) 및 전기적 사양	047
3-5. Smart I/O-III Base Board 기본 내부 기능	049
4. Smart I/O-III 커넥터 사양 안내	049

4

Smart I/O-Block Series

1. Smart I/O-Block Series 호환성	052
2. Smart I/O-Block Series 사양	052
3. Smart I/O-Block Series 커넥터 사양 안내	053
4. Input Block	054
4-1. Input Block 소개	054
4-2. Input Block 연결단자 설명	054
4-3. Input Block 장착위치	056
4-4. Input Block 외형 및 치수	057
4-5. Input Block 응용 방법(외부 인터페이스 결선)	058
4-6. 언어별 주요 소스코드 및 응용 예제	063
5. ADC Block	066
5-1. ADC Block 소개	066
5-2. ADC Block 연결단자 설명	066
5-3. ADC Block 입력신호를 전압에서 전류로 변환하여 입력 받는 방법	068
5-4. ADC Block 장착위치	068
5-5. ADC Block 외형 및 치수	069
5-6. ADC Block 응용 방법(외부 인터페이스 결선)	070
5-7. 언어별 주요 소스 코드 및 응용 예제	072
6. InputCounter Block	076
6-1. InputCounter Block 소개	076
6-2. InputCounter Block 연결단자 설명	077
6-3. InputCounter Block 장착위치	078
6-4. InputCounter Block 외형 및 치수	079
6-5. InputCounter Block 응용 방법(외부 인터페이스 결선)	080
6-6. 언어별 주요소스 코드 및 응용 예제	081
7. Relay Block	086
7-1. Relay Block 소개	086
7-2. Relay Block 연결단자 설명	086
7-3. Relay Block 장착위치	088
7-4. Relay Block 외형 및 치수	089
7-5. Relay Block 응용 방법(외부 인터페이스 결선)	090
7-6. 언어별 주요소스 코드 및 응용 예제	091
8. FET Block	093
8-1. FET Block 소개	093
8-2. FET Block 연결단자 설명	093

8-3. FET Block 장착위치	095
8-4. FET Block 외형 및 치수	096
8-5. FET Block 응용 방법(외부 인터페이스 결선)	096
8-6. 언어별 주요소스 코드 및 응용 예제	099
9. DAC Block	102
9-1. DAC Block 소개	102
9-2. DAC Block 연결단자 설명	102
9-3. DAC Block 출력신호를 전압에서 전류로 변환하여 출력하는 방법	103
9-4. DAC Block 장착위치	104
9-5. DAC Block 외형 및 치수	105
9-6. DAC Block 응용 방법(외부 인터페이스 결선)	105
9-7. 언어별 주요소스 코드 및 응용 예제	106
10. RS232 Isolation Block	108
10-1. RS232 Isolation Block 소개	108
10-2. RS232 Isolation Block 연결단자 설명	108
10-3. RS232 Isolation Block 장착위치	109
10-4. RS232 Isolation Block 외형 및 치수	110
10-5. RS232 Isolation Block 응용 방법(외부 인터페이스 결선)	111
10-6. 언어별 주요 소스 코드 및 응용 예제	112

MEMO

1

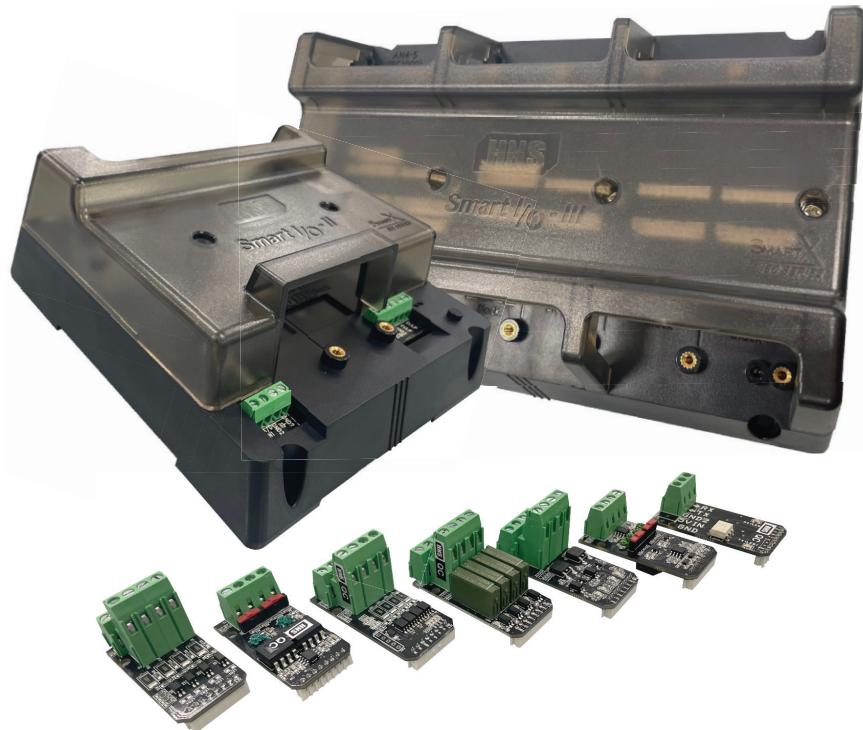
Part

Smart I/O-Series 시작하기

Part-I. Smart I/O-Series 시작하기

1. Smart I/O-Series 소개 및 구성

Smart I/O-Series는 IEC-Series와 연동하여 외부 디바이스와 인터페이스를 지원하는 제품으로 쉽고 편리하게 산업 환경에 적용할 수 있는 솔루션이며 기존의 HMI(Human Machine Interface)와 PLC(Programmable Logic Controller)의 기능을 단품으로 구성할 수 있는 제품입니다.



※ Smart I/O-Series 특징

- 1** I/O 구성을 Flexible하게 구성할 수 있도록 각각의 I/O Module을 Block 형태로 제작하였습니다.
- 2** 외부 인터페이스와 IEC-Series 사이에 Ground가 분리되어(Isolation) IEC-Series가 전기적으로 보호됩니다.
- 3** I/O 제어는 SmartX Framework를 통해 아주 쉽게 처리 가능합니다.
- 4** 외부 기기(장치)들과 별도의 인터페이스 회로 없이 바로 연결하여 사용할 수 있습니다.
- 5** 다양한 기능의 I/O Block을 지원합니다.
- 6** 적은 비용으로 HMI와 PLC 기능의 제품으로 구성할 수 있습니다.

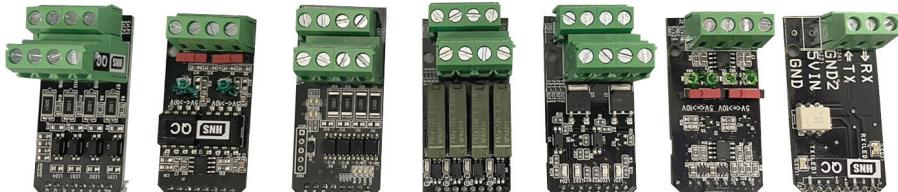
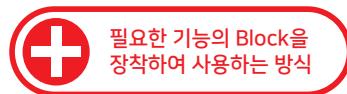
Smart I/O-Series에는 Smart I/O-II, Smart I/O-III Base Board가 있으며, 해당 Board에 각종 입출력 기능을 Module화하여 원하는 인터페이스를 구성할 수 있는 Block Series 제품이 있어 각 기능의 Block Series를 Smart I/O에 장착함으로써 다양한 입출력 기능을 사용할 수 있도록 구성되어 있습니다.



Smart I/O-II Base Board



Smart I/O-III Base Board

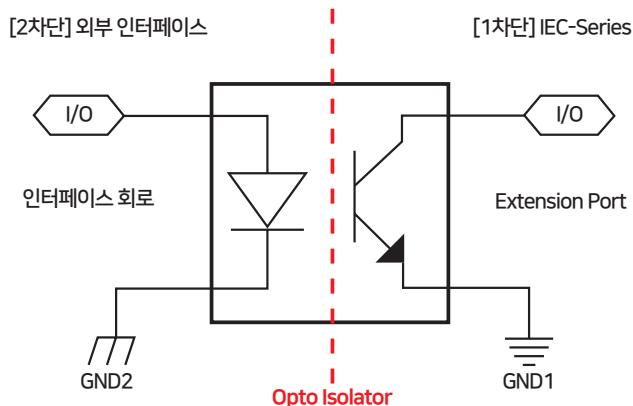


Smart I/O-Block Series

2. 인터페이스의 전원 분리(Interface Isolation)

Smart I/O-Series의 전원 Isolation 시스템은 외부 인터페이스와 내부 제어부의 전기적인 분리(Isolation)를 통하여 외부에 전기적인 데미지가 발생하더라도 IEC-Series에는 전기적인 데미지가 전달이 되지 않게 구성되어 전기적으로 안정적인 동작을 할 수 있도록 최소의 보호기능을 가지고 있습니다.

Smart I/O Base Board and Blocks



3. Smart I/O-II, III의 차이점 비교

IEC-Series 제품과 I/O의 구성에 따라 Smart I/O Base Board를 선택하여 사용해야 합니다. 각 Board별 지원되는 IEC-Series 제품과 Block별 최대 장착 수량 등을 확인하여 제품을 선택하시기 바랍니다.

1) Smart I/O Base Board 기본 사양 비교

제품군	Smart I/O-II	Smart I/O-III
사용가능 IEC-Series 제품	IEC-Series 전제품	IEC 非Lite-Series 전제품
Extension Port	Extension Port-I	Extension Port-II
기능 변경	Block으로 자유롭게 구성 가능	
GPIO 채널수	16Ch	48Ch
PWM 채널수	2Ch	
ADC 채널수	4Ch (내부 2Ch + Block 2Ch)	4Ch (IEC1000-Series는 6Ch)
Input Block 최대 장착 개수	4개 (16Ch)	12개 (48Ch)
InputCounter Block 최대 장착 개수	4개 (8Ch)	조합(합산)하여 최대 4개 (16Ch)를 장착할 수 있습니다. 4개 (8Ch)
Relay Block 최대 장착 개수	4개 (16Ch)	12개 (48Ch)
FET Block 최대 장착 개수	4개 (16Ch)	12개 (48Ch)
ADC Block 최대 장착 개수	1개 (2Ch)	2개 (4Ch) (IEC1000-Series는 3개, 6Ch)
RS232 Block 최대 장착 개수	1개 (1Ch)	1개 (1Ch)
DAC Block 최대 장착 개수	1개 (2Ch)	1개 (2Ch)

참고

장착 가능한 Block의 총 개수는 Smart I/O-II는 5개, Smart I/O-III는 12개로 제한됩니다. 따라서 Base Board 별 최대 사용할 Block의 합산된 개수는 5개, 12개를 넘지 않도록 조합하시기 바랍니다.

2) Smart I/O Base Board 기능

① Smart I/O-II Base Board에 포함(내장)된 기능

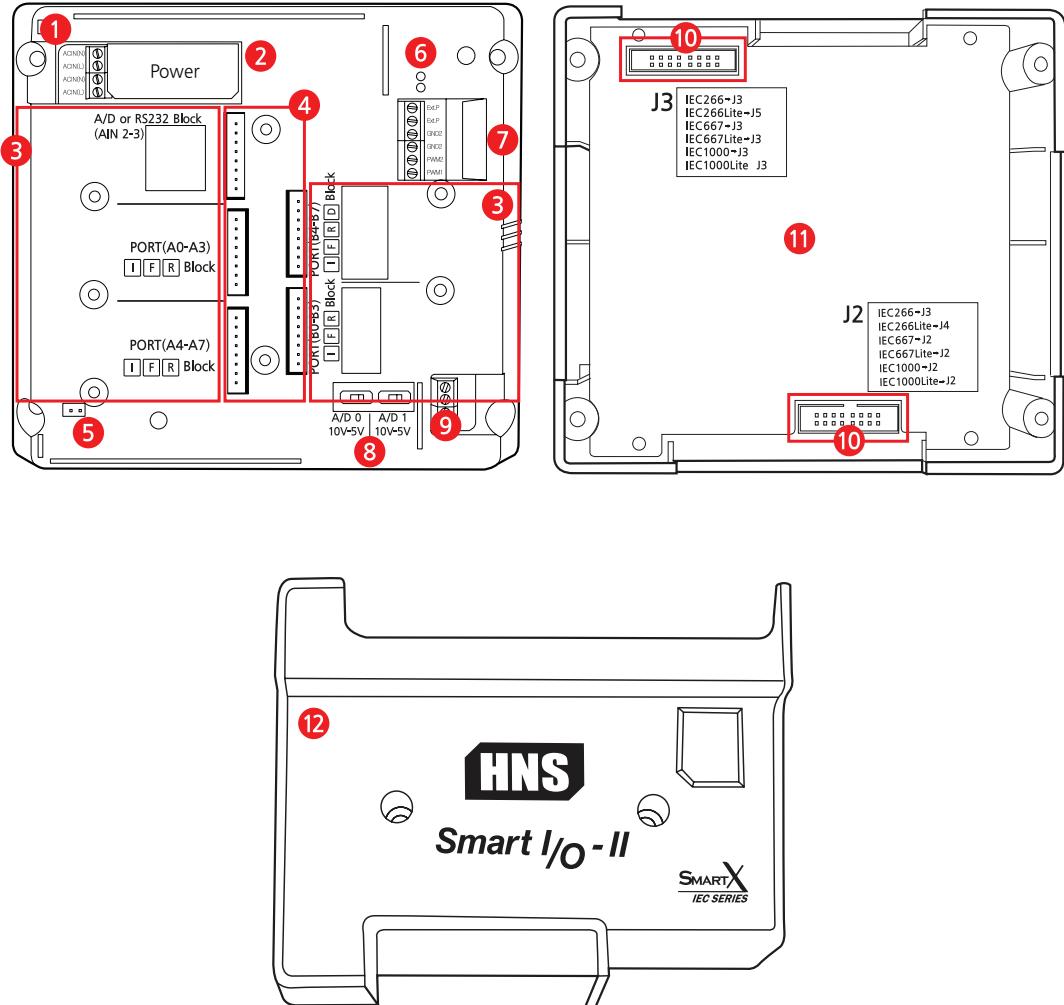
장치기능	Pin Name	Channel	전기적 사양
ADC 입력	A/D IN	2Ch	0 ~ 5V / 0 ~ 10V (12Bit)
PWM 출력	PWM1, PWM2	2Ch	VSS = 55V, ID = 17A

② Smart I/O-III Base Board에 포함(내장)된 기능

Smart I/O-III Base Board에는 외부 인터페이스를 구성할 수 있는 별도의 내부 기능이 없습니다. 따라서 필요한 기능의 I/O Block을 장착하여 사용하시기 바랍니다.

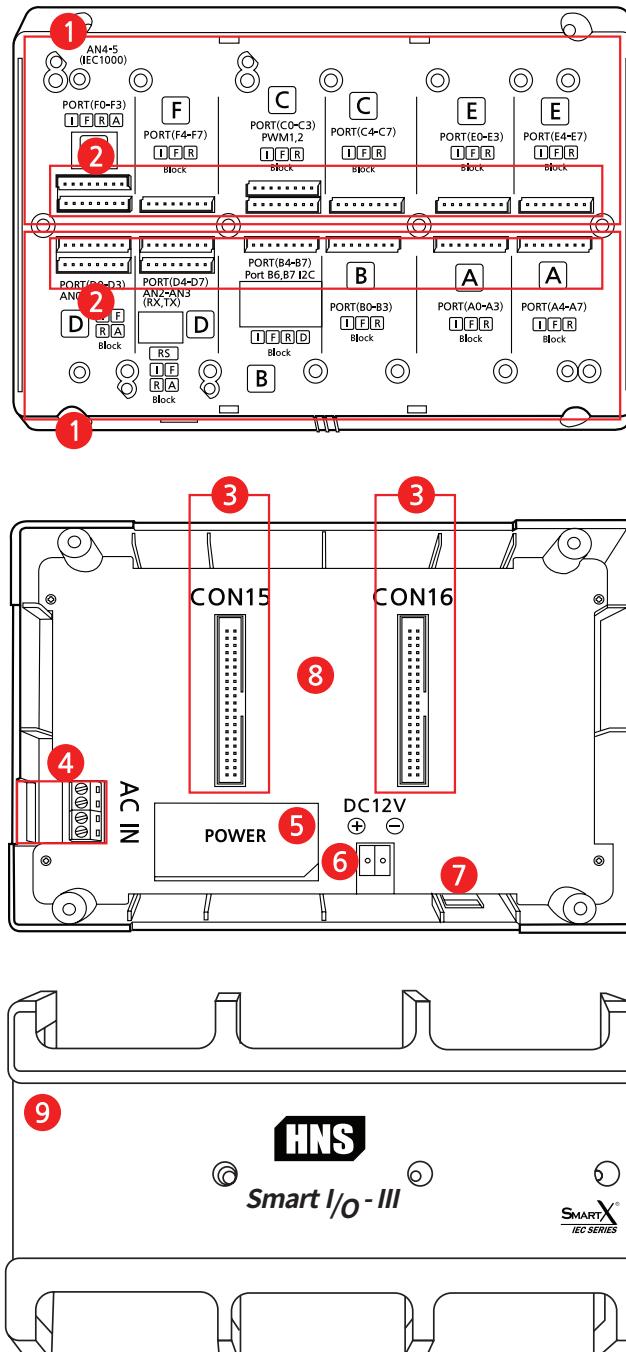
4. Smart I/O Base Board 각부 명칭

1) Smart I/O-II Base Board의 각부 명칭



- | | | | | |
|---------------|--------------------------|-------------------------|-------------------|-------------------------|
| ① AC 전원 입력 단자 | ② AC/DC Converter (SMPS) | ③ Block 장착 위치 | ④ Block Connector | ⑤ Lite-Series 점퍼 |
| ⑥ 전원 상태 LED | ⑦ 내부 장착 PWM 단자 | ⑧ ADC 입력전압 변경 Slide S/W | ⑨ 내부 장착 ADC 단자 | ⑩ Extension Port-I 연결단자 |
| ⑪ 하부 보호판 | ⑫ 상부 케이스 | | | |

2) Smart I/O-III Base Board의 각부 명칭



- | | | | | |
|---------------|-------------------|-----------------------------|---------------|-----------------------------|
| ① Block 장착 위치 | ② Block Connector | ③ Extension Port-II
연결단자 | ④ AC 전원 입력 단자 | ⑤ AC/DC Converter
(SMPS) |
| ⑥ DC 전원 입력 단자 | ⑦ 전원 상태 LED | ⑧ 하부 보호판 | ⑨ 상부 케이스 | |

5. Smart I/O-II, III 공통사항

5-1. 전원 연결하기

Smart I/O-Series 기능의 정상 동작을 위해 반드시 AC 외부 입력 전원이 필요하며, 시스템 보호를 위하여 전원 절연(Isolation)을 목적으로 기본적으로는 AC 전원을 입력하도록 설계되어 있습니다.

주의

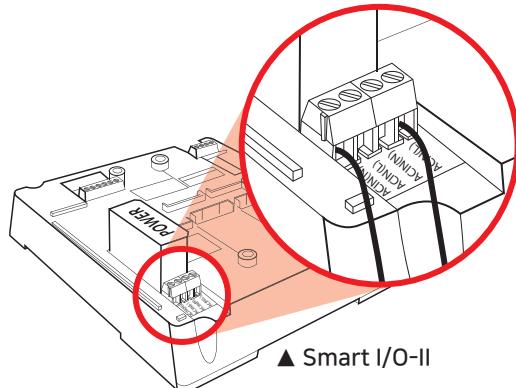
1. AC 전원을 사용하는 경우 제품에 전원(SMPS)이 반드시 장착되어 있어야 합니다.
2. 잘못된 배선으로 연결 시 폭발/상해/제품 고장등의 위험이 있습니다.
3. AC 전원 연결 시 위험하므로 반드시 OFF된 상태에서 연결합니다.
4. 극성이 없으므로 N, L이 뒤바뀌어도 무관합니다.
5. 전선은 단자 나사로 확실히 조여 접촉불량이 발생되지 않도록 합니다.

1) AC 전원 연결하기

< Smart I/O-II Base Board의 AC 전원 연결 >

Smart I/O-II 전원 연결 방법

Smart I/O-II의 상부 케이스를 제거한 후 터미널 단자에 전원을 연결합니다.
(반드시 AC 전원이 차단된 상태에서 연결하시기 바랍니다.)



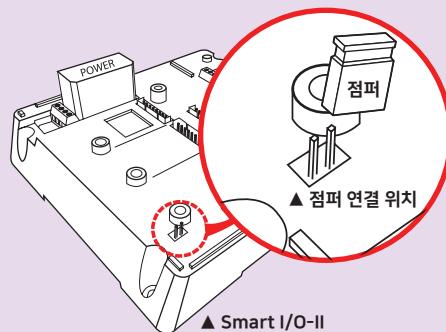
중요

IEC Lite-Series 제품 사용 시 점퍼 유무에 대한 중요사항

IEC Lite-Series를 사용하는 경우 Smart I/O-II Base Board에 점퍼를 연결하여 사용하셔야 하며, IEC 非Lite-Series를 사용하는 경우에는 점퍼 연결을 해체하여 사용하셔야 합니다.

점퍼 장착	Lite-Series	IEC 1000 Series	IEC 1000Lite-43 [B1/B2], IEC 1000Lite-56 [B1/B2], IEC 1000Lite-07N [B1/B2], IEC 1000Lite-07 [B1/B2], IEC 1000Lite-08 [B1/B2], IEC 1000Lite-102 [B1/B2], IEC 1000Lite-104 [B1/B2]
		IEC 667 Series	IEC 667Lite-56 [B1/B2], IEC 667Lite-07 [B1/B2], IEC 667Lite-08 [B1/B2], IEC 667Lite-102 [B1/B2], IEC 667Lite-104 [B1/B2]
점퍼 해제	非 Lite-Series	IEC 1000 Series	IEC 1000-07N [B1/B2], IEC 1000-07 [B1/B2], IEC 1000-08 [B1/B2], IEC 1000-102 [B1/B2], IEC 1000-104 [B1/B2], IEC 1000-104H [B1/B2], IEC 1000-150 [B]
		IEC 667 Series	IEC 667-07 [B1/B2], IEC 667-08 [B1/B2], IEC 667-102 [B1/B2], IEC 667-104 [B1/B2],

- 다음 페이지 계속 →

**주의**

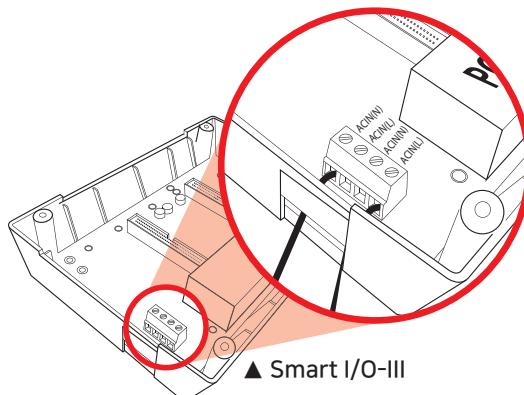
점퍼가 연결된 상태에서 IEC Lite-Series 제품을 연결할 경우 장비에 심각한 데미지를 줄 수 있습니다.

< Smart I/O-III Base Board의 AC 전원 연결 >

Smart I/O-III**전원 연결 방법**

Smart I/O-III의 제품 뒷면을 위로 향하게 하여 하단 터미널 단자에 전원을 연결합니다.

(반드시 AC 전원이 차단된 상태에서 연결하시기 바랍니다.)



2) DC 전원 연결하기(별도 개조작업 필요)

주의

- DC 전원으로 변경(개조)하여 사용할 경우 DC 전원은 IEC-Series 전원의 Ground(GND)와 분리된 전원을 사용하시기 바랍니다. 분리된(Isolation) 전원을 사용하지 않을 경우 입력 측과 IEC-Series 인터페이스 측의 전기적인 보호가 되지 않습니다.
- DC 전원 연결의 경우 반드시 Board에 표시되어 있는 극성을 참고하여 연결하시기 바랍니다.
- AC와 DC 전원을 동시에 입력할 수 없습니다.

AC 전원을 인가하기 어려운 환경에서 DC 전원을 인가하기 위한 변경 방법 설명입니다. DC 전원을 사용할 경우 입력 전압은 반드시 아래의 표를 참고하시기 바랍니다.

제품명	입력전원 사양
Smart I/O-II	DC 12V, 300mA 이상 사용 권장
Smart I/O-III	DC 12V, 800mA 이상 사용 권장

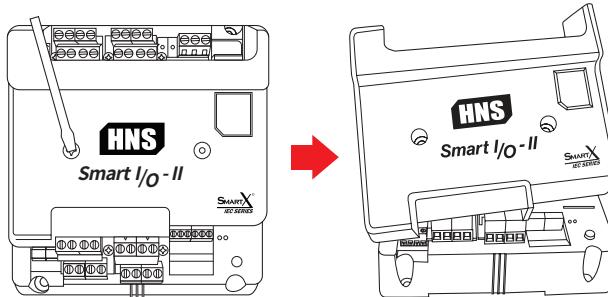
참고

제품 구입시 DC 전원으로 변경 요청을 하시면 DC 전원으로 바로 사용할 수 있는 상태의 제품을 받으실 수 있습니다.

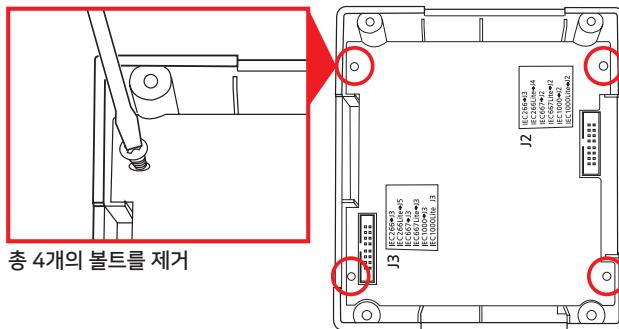
< Smart I/O-II Base Board의 DC 전원 개조작업 >

STEP-1 Smart I/O-II의 상부 및 하부 케이스를 모두 해제하여 Board를 분리

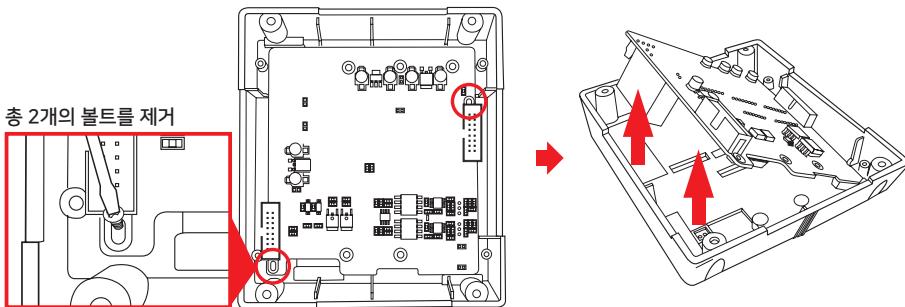
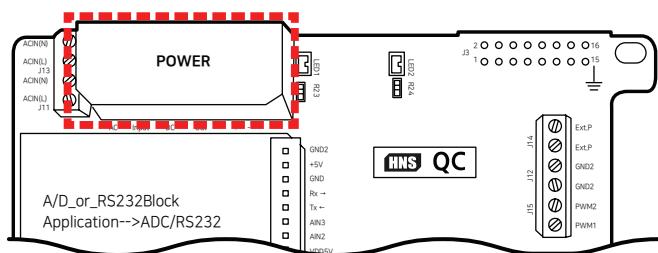
< 상부 케이스 해제 >



< 하부 보호판 분리 >

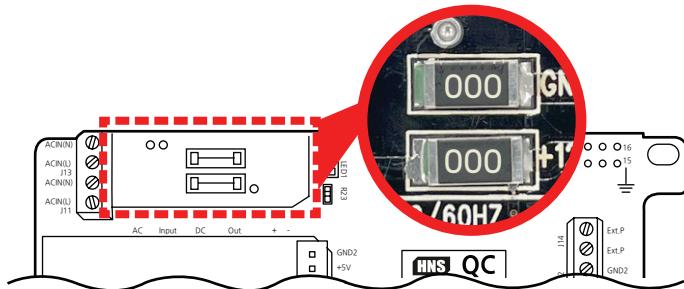


< Board 분리 >

**STEP-2** SMPS(Power)를 제거 (사출케이스 제거한 상태)

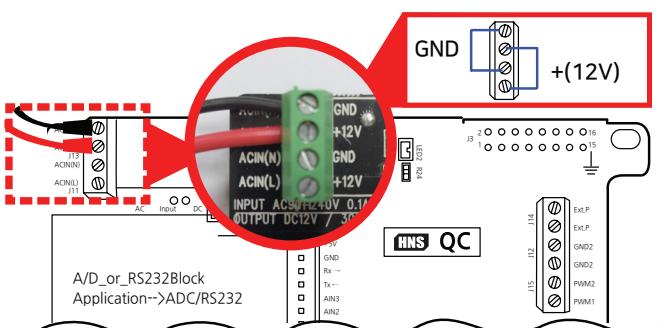
STEP-3

SMPS를 제거한 부분에 사진과 같이 Shunt 저항을 장착할 수 있도록 되어 있으며, 이곳에 Shunt(0Ω)저항 또는 점퍼선을 연결



STEP-4

전원 연결은 AC와 달리 DC 전원을 사용할 경우 반드시 Board에 표시되어 있는 극성을 참고하여 연결



< Smart I/O-III Base Board의 DC 전원 개조작업 >

참고

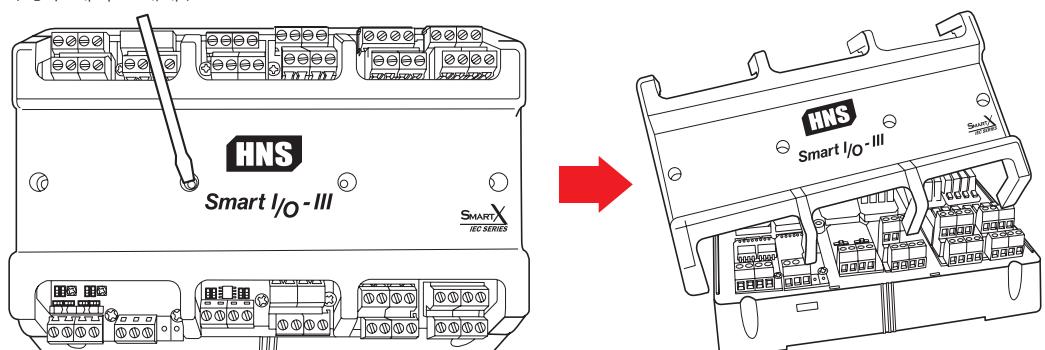
Smart I/O – III Base Board의 경우 DC 전원을 사용할 때 SMPS를 제거할 필요가 없습니다.

Smart I/O – III Base Board의 경우 2EDGV-5.0-02P 핀을 장착하여 DC 전원을 인가하면 AC 전원과는 별도의 회로로 전원이 공급되므로 SMPS를 제거하지 않아도 DC 전원을 사용할 수 있습니다.

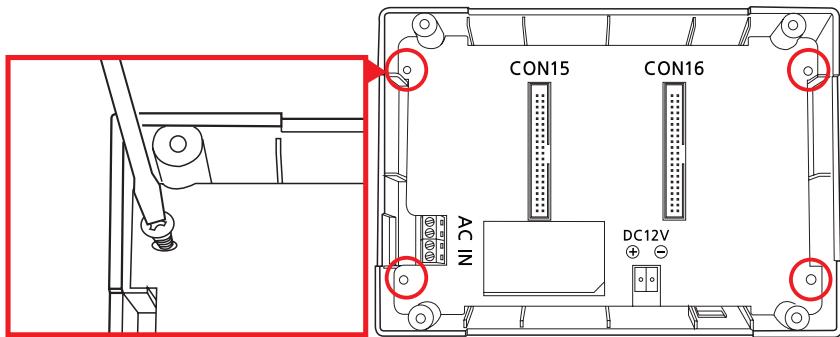
STEP-1

Smart I/O-III의 상부 및 하부 케이스를 모두 해제하여 Board를 분리

< 상부 케이스 해제 >

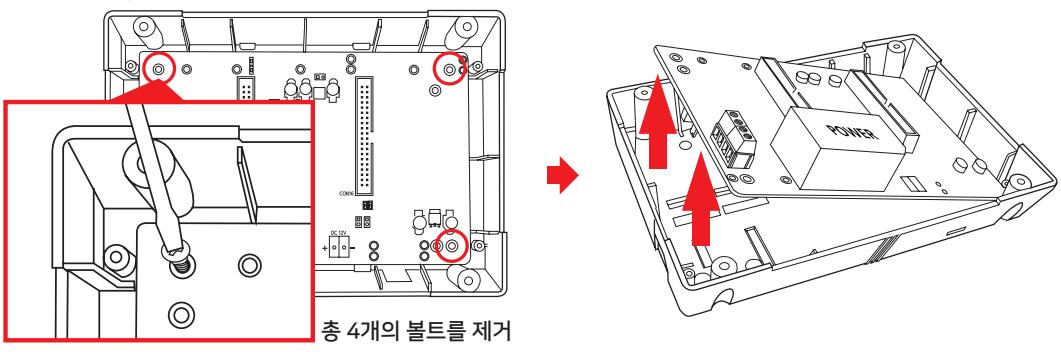


< 하부 보호판 분리 >

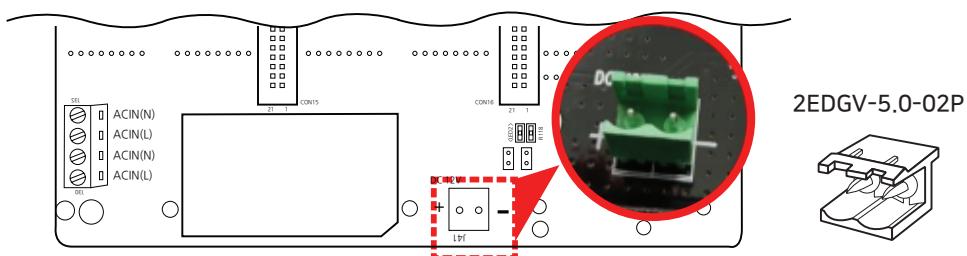


총 4개의 볼트를 제거

< Board 분리 >

**STEP-2**

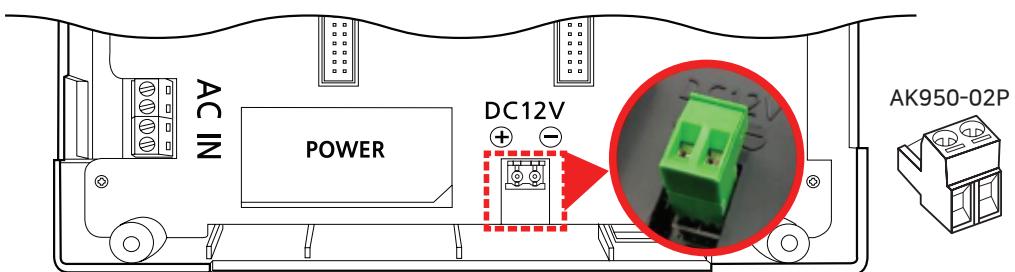
Smart I/O-III 구매 시 동봉된 커넥터 2EDGV-5.0-02P 핀을 준비하여 J18 단자에 장착



2EDGV-5.0-02P

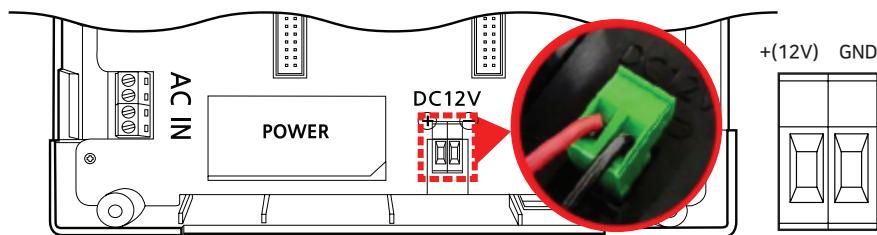
STEP-3

사출케이스 하부보호판을 조립한 후 조립한 상태에서 나머지 콘넥터 AK950-02P를 연결



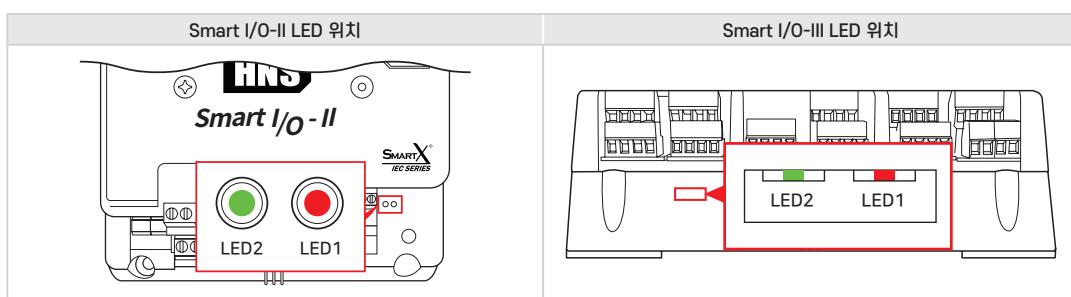
STEP-4

전원연결은 AC와 달리 DC 전원을 사용할 경우 반드시 Board에 표시되어 있는 극성을 참고하여 연결



3) 전원 입력 시 LED 상태

Smart I/O - Base Board에 전원을 입력 시 LED를 통해 상태를 확인할 수 있습니다. 아래에서 Smart I/O-II, III의 LED 위치를 참고 하시기 바랍니다.



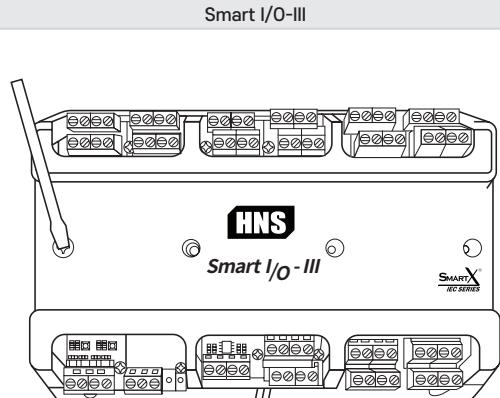
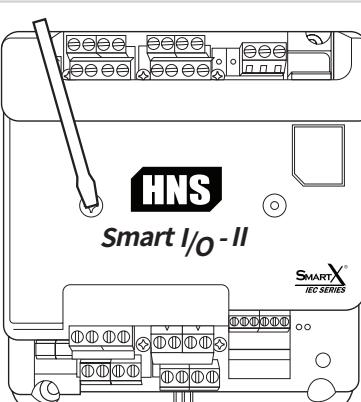
< Smart I/O-II, III Base Board의 전원 LED 상태 >

LED	설명	상태 설명	
		LED-ON	LED-OFF
LED1 [LED : 빨간색]	IEC-Series와 Extension Port 연결에 따른 전원 인가 상태 표시	LED-ON DC3.3V와 5V가 정상 출력	LED-OFF DC3.3V와 5V가 미 출력
LED2 [LED : 초록색]	Smart I/O - II, III의 AC 전원 인가 상태에 따른 출력 상태 표시	LED-ON DC5V와 12V가 정상 출력	LED-OFF DC5V와 12V가 미 출력

5-2. Block-Series 장착 방법

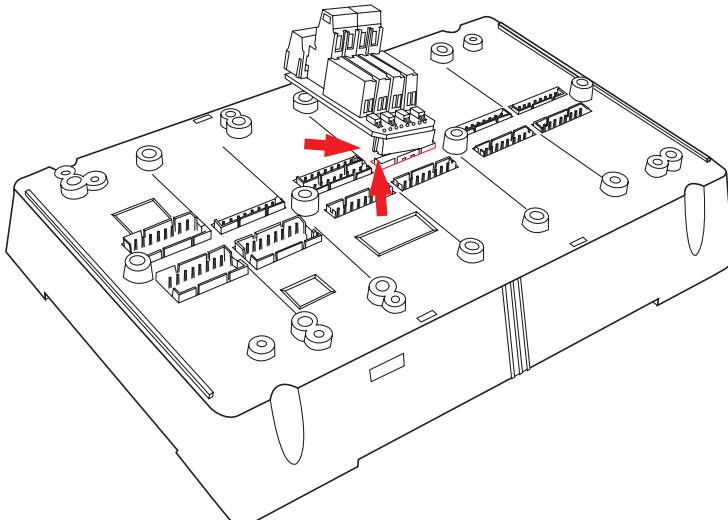
STEP-1

Smart I/O-Base Board의 전면 커버(상부 케이스)의 나사를 풀어 케이스를 해제합니다



STEP-2

Smart I/O-Block 장착 위치를 참고하여 Smart I/O와 Base Board의 홈을 맞춰 Block을 장착합니다.

**STEP-3**

동봉된 볼트를 사용하여 장착된 Block을 고정합니다.

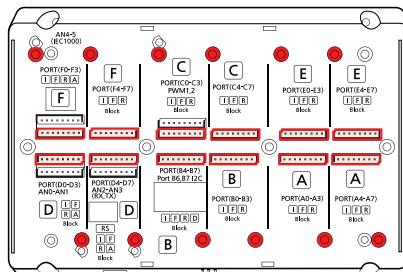
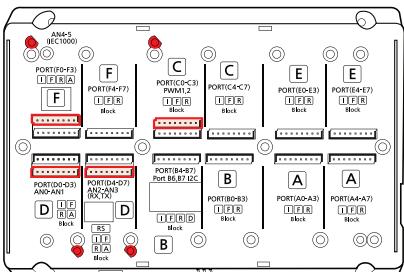
Smart I/O-Base Board에 Block 장착 시 사용되는 볼트

볼트 명	M3X6(PH) 동근머리	Tapping(PH) 3X6
사용 Board	Smart I/O - II, III	Smart I/O - III
수량	12개 (Smart I/O - II는 5개)	4개
그림		

< Smart I/O-II Base Board의 경우 >

사용 볼트	장착 Block	볼트 체결 위치
	Input Block ADC Block RS232 Isolation Block FET Block Relay Block DAC Block InputCounter Block	<p>나사(M3X6 PH)</p> <p>모든 Block류 고정 시 M3X6(PH) 볼트가 사용됨.</p>

< Smart I/O-III Base Board의 경우 >

사용 볼트		
장착 Block	Input Block, FET Block, Relay Block, DAC Block, InputCounter Block	ADC Block, RS232 Isolation Block, FET Block(PWM Output일 경우)
Block 장착 및 볼트 체결 위치		
볼트 체결 위치에 따른 차이점	 구리 Insert가 있는 부분에는 M3X6(PH)를 사용	 모서리가 깎이고 사출물 그대로인 부분에는 Tapping(PH)를 사용

STEP-4

Block 장착이 완료되었다면 상부 케이스를 다시 조립하여 마무리합니다.

5-3. Smart I/O-II, III 기구 디자인 가이드

Smart I/O-II, III를 기구에 고정할 수 있도록 기구 케이스를 제작한 후 장착 방법을 참고하시기 바랍니다.

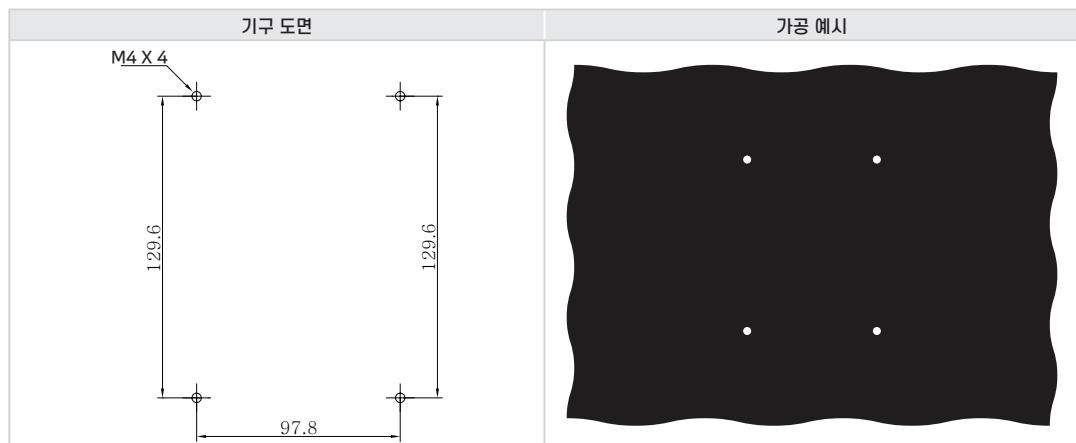
< 기구 케이스 제작 방법 >

판금 및 케이스측의 탭(Tap hole) 기구 도면 자료를 다운로드 받아 훌 가공처리 하여 케이스 부분을 제작하시기 바랍니다.

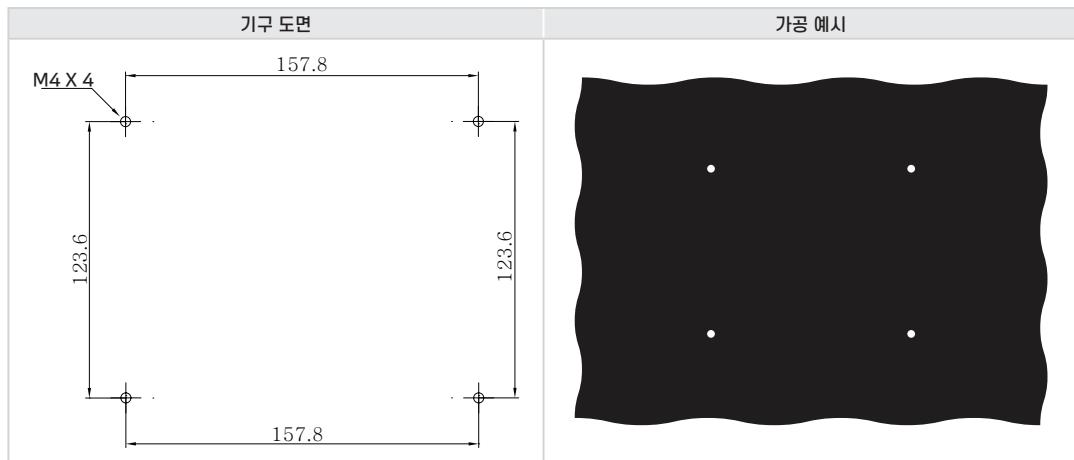
참고

- 고정 철판은 1T 이상을 사용하는 것을 권장합니다.
- 기구 도면은 [홈페이지(www.hnsts.co.kr)] - [자료실] - [도면 및 승인원] - [Smart I/O-II, III 기구 디자인 및 가이드]에서 다운로드 할 수 있습니다.

1) Smart I/O-II의 기구 도면



2) Smart I/O-III의 기구 도면



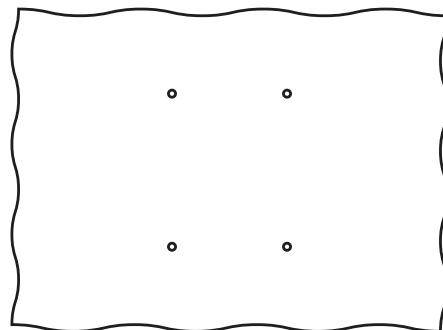
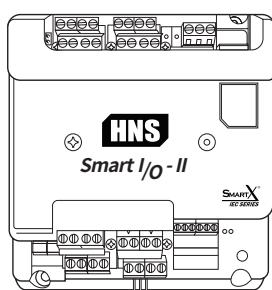
<기구 장착(고정) 방법 >

주의

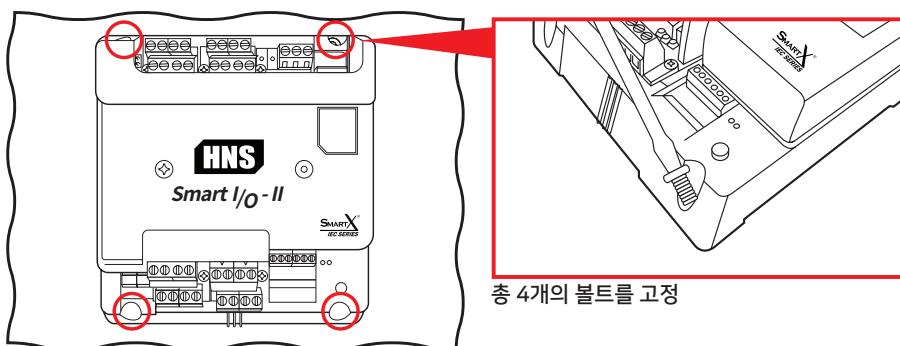
본 그림은 Smart I/O-II 기준으로 제시된 그림이며, Smart I/O-III에서도 그림과 같이 적용하여 장착하시기 바랍니다.

STEP-1

제품과 기구케이스(판금)를 준비합니다.

**STEP-2**

제품을 케이스의 방향에 맞게 올려놓은 후 상하좌우의 고정홀을 볼트로 고정합니다.

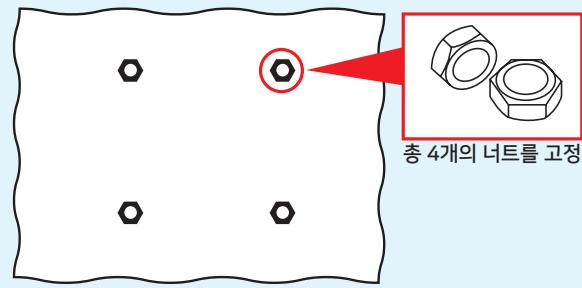
**참고**

제품 구매 시 동봉된 M4X10 볼트 4개를 사용하여 제품을 기구에 채결하시기 바랍니다.

참고

고정 판넬의 홀 가공 시 Tapping 작업이 안되어 있는 경우 참고사항

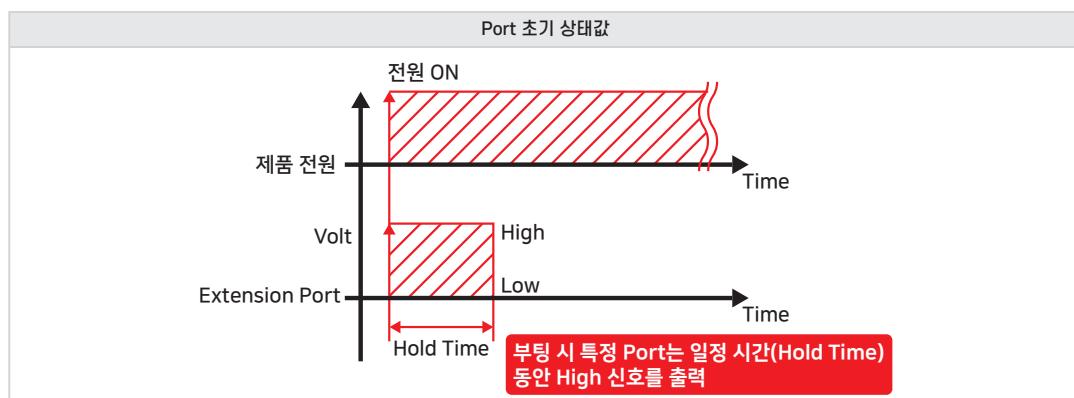
고정 판넬의 홀 가공 시 Tapping 작업이 안되어 있는 경우 M4너트를 구입하여 고정하시기 바랍니다.



5-4. IEC-Series별 Extension Port 특성에 따른 Input/Output Port 선택

IEC-Series별 Extension Port의 각 Port마다 초기 상태 값(High/Low)이 다르기 때문에 아래의 내용들을 참고하여 입력 Port, 출력 Port를 선택하여 사용하시기 바랍니다.

IEC-Series	GPIO General Purpose Input Output
IEC667-Series	<ul style="list-style-type: none"> Port-A ~ H: 각 핀마다 Input/Output 선택 가능 [3.3V Level] Port-B의 6, 7Pin은 오픈 드레인 출력 Port-B(3, 4, 5)와 Port-D(3, 4, 6)의 경우 IEC667의 전원 인가 시 약 260ms ~ 330ms 동안 High로 출력되기 때문에 입력으로 사용하는 것을 권장합니다.
IEC1000-Series	<ul style="list-style-type: none"> Port-A ~ H: 각 핀마다 Input/Output 선택 가능 [3.3V Level] Port-B의 6, 7Pin은 오픈 드레인 출력



주의

각 포트별 Pull-Down 저항을 장착한 후 측정하였습니다. IEC-Series에서 특정 포트는 전원 ON 후 일정 시간 동안(Hold Time) High Level 신호가 출력됩니다.

주의

OS 버전별(OPT, STD, PRO) 부팅시간의 차이로 측정값이 다소 차이가 발생할 수 있습니다.
(측정 OS는 STANDARD(STD)버전 기준입니다.)
하기의 데이터는 대략적인 데이터로서 약간의 시간 오차가 발생할 수 있습니다.

참고

초기 값이 Floating 되거나 High로 표시되는 경우에는 Pull Down/Pull Up 저항을 걸어서 사용하면 됩니다.

<IEC 667 B23(Build Version) OS-Port 초기 상태값(※ Pull Down 저항 장착 후 테스트)>

Port-Pin	0	1	2	3	4	5	6	7
PORT A	Hold Time 구간	LOW	LOW	LOW	LOW	LOW	LOW	LOW
	Hold Time 이후	LOW	LOW	LOW	LOW	LOW	LOW	LOW
	※Hold Time 구간	LOW	LOW	LOW	LOW	LOW	LOW	LOW
	※Hold Time 이후	LOW	LOW	LOW	LOW	LOW	LOW	LOW
PORT B	Hold Time 구간	LOW (8.04s)	LOW (8.04s)	LOW (8.04s)	LOW (240ms) LOW (8.04)	LOW (240ms) LOW (8.04)	LOW (240ms) LOW (8.04)	LOW LOW
	Hold Time 이후	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH	LOW LOW
	※Hold Time 구간	LOW	LOW	LOW	HIGH (315ms)	HIGH (315ms)	HIGH (315ms)	LOW LOW
	※Hold Time 이후	LOW	LOW	LOW	LOW	LOW	LOW	LOW
PORT C	Hold Time 구간	LOW	LOW	LOW	LOW	LOW	LOW	LOW
	Hold Time 이후	LOW	LOW	LOW	LOW	LOW	LOW	LOW
	※Hold Time 구간	LOW	LOW	LOW	LOW	LOW	LOW	LOW
	※Hold Time 이후	LOW	LOW	LOW	LOW	LOW	LOW	LOW
PORT D	Hold Time 구간	LOW (1.4s)	LOW (1.4s)	LOW (1.6ms)	LOW (210ms) HIGH (210ms) LOW (1.14s)	HIGH (210ms) LOW (1.14s)	HIGH (210ms) LOW (1.14s)	HIGH (210ms) LOW (1.14s)
	Hold Time 이후	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH HIGH
	※Hold Time 구간	LOW	LOW	LOW (1.13s)	LOW (210ms) HIGH (1.6ms) LOW	HIGH (250ms)	HIGH (250ms)	HIGH (250ms)
	※Hold Time 이후	LOW	LOW	LOW	LOW	LOW	LOW	LOW
PORT E	Hold Time 구간	LOW	LOW	LOW	LOW	LOW	LOW	LOW
	Hold Time 이후	LOW	LOW	LOW	LOW	LOW	LOW	LOW
	※Hold Time 구간	LOW	LOW	LOW	LOW	LOW	LOW	LOW
	※Hold Time 이후	LOW	LOW	LOW	LOW	LOW	LOW	LOW
PORT F	Hold Time 구간	LOW (1.4s)	LOW (1.4s)	LOW (1.4s)	LOW (1.4s)	LOW (1.4s)	LOW (1.4s)	LOW (1.4s)
	Hold Time 이후	LOW (1.4s)	LOW (1.4s)	LOW (1.4s)	LOW (1.4s)	LOW (1.4s)	LOW (1.4s)	LOW (1.4s)
	※Hold Time 구간	LOW	LOW	LOW	LOW	LOW	LOW	LOW
	※Hold Time 이후	LOW	LOW	LOW	LOW	LOW	LOW	LOW
PORT G	Hold Time 구간	LOW	LOW	LOW	LOW	LOW	LOW	HIGH (1.4s) LOW (0.3s)
	Hold Time 이후	LOW	LOW	LOW	LOW	LOW	LOW	LOW

PORT G	※Hold Time 구간	LOW	LOW (1.7s)						
	※Hold Time 이후	LOW							
PORT H	Hold Time 구간	LOW	LOW	LOW	LOW	-	-	-	-
	Hold Time 이후	LOW	LOW	LOW	LOW	-	-	-	-
	※Hold Time 구간	LOW	LOW	LOW	LOW	-	-	-	-
	※Hold Time 이후	LOW	LOW	LOW	LOW	-	-	-	-

주의**IEC667-Series Port 초기 상태값**

IEC667-Series에서 Port-B, Port-D, Port-F, Port-G는 입력으로 사용하는 것을 권장합니다.

Port-B 0, 1, 2, 3, 4, 5 Pin은 전원 인가 후 9.3s 경과하여 HIGH로 출력되며, Port-D 0 ~ 7Pin, Port-F 0 ~ 7Pin, Port-G 7Pin은 전원 인가 후 1.4s 경과하여 High로 출력됩니다. 제시한 Port는 Pull Down 저항을 걸어서 사용하는 경우에는 문제가 되지 않습니다.

< IEC1000 B1(Build Version) OS(Windows Embedded CE 6.0)-Port 초기 상태값 (※ : Pull Down 저항 장착 후 테스트) >

Port-Pin	0	1	2	3	4	5	6	7
PORT A~H	Hold Time 구간	LOW						
	Hold Time 이후	LOW						
	※Hold Time 구간	LOW						
	※Hold Time 이후	LOW						

6. Smart I/O-Series 사용 시 주요문제 해결 방법

Smart I/O-Series를 사용하면서 발생할 수 있는 주요 문제들에 대하여 해결 방법을 제시하고 있습니다. 제품 사용에 문제가 발생할 경우 아래의 내용들을 참고하여 접검 및 해결하시기 바랍니다.

6-1. Smart I/O Base Board에 전원이 입력되었는지 확인

Smart I/O Base Board에는 AC 전원을 입력하게 되어있습니다. Smart I/O Base Board에 전원이 입력되지 않은 경우 특정 Block이 정상 동작하지 않을 수 있습니다. 그러므로 반드시 “[5-1. 전원 연결하기] – [3] 전원 입력 시 LED 상태]” 내용을 참고하여 Smart I/O Base Board에 전원이 인가되었는지 확인합니다.

주의**Smart I/O – Base Board 사용 중 입력전원(AC/DC)이 입력되지 않은 경우 Block의 기능 정상 동작 여부**

※ 전원 입력에 따른 Block-Series 동작 여부

AC 전원 입력없음	Input Block 동작 함	Input.C Block 동작 안 함	FET Block	Relay Block	ADC Block	DAC Block	RS232 Block
---------------	---------------------	-------------------------	-----------	-------------	-----------	-----------	-------------

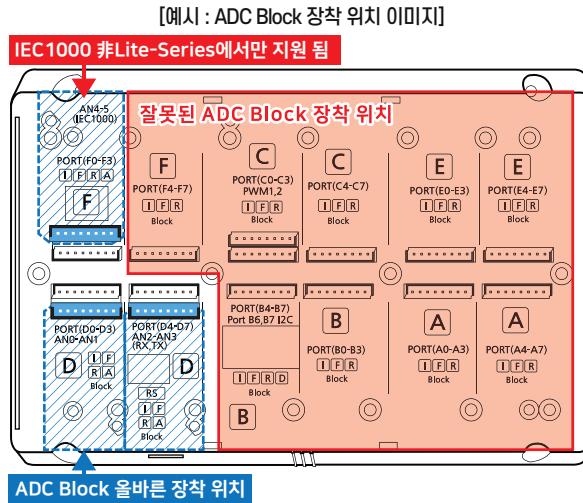
Smart I/O – Base Board에 전원이 인가되지 않은 상태에서 특정 Block이 정상 동작하지 않을 수 있으니 반드시 위의 표를 참고하여 전원을 인가해주시기 바랍니다.

참고

Smart I/O Base Board의 전원 입력에 대한 자세한 사항은 “[Part-I. Smart I/O – Series 시작하기] – [5. Smart I/O – II, III 공통사항] – [5-1. 전원 연결하기]” 내용을 참고하시기 바랍니다.

6-2. Smart I/O Base Board에 장착된 Block의 위치가 올바른지 확인

Smart I/O의 Block-Series는 각 Block마다 장착하는 위치가 정해져 있습니다. 정해진 장착 위치가 아닌 다른 위치에 Block을 장착하는 경우 정상으로 동작하지 않으므로 정해진 위치에 올바르게 장착되었는지 확인하시기 바랍니다.



위 예시 그림과 같이 잘못된 위치에 Block을 장착하는 경우 데이터 입출력이 정상 동작하지 않습니다. 아래의 참고 박스를 확인하여 Block별 정확한 장착 위치를 확인하여 Block을 장착하시기 바랍니다.

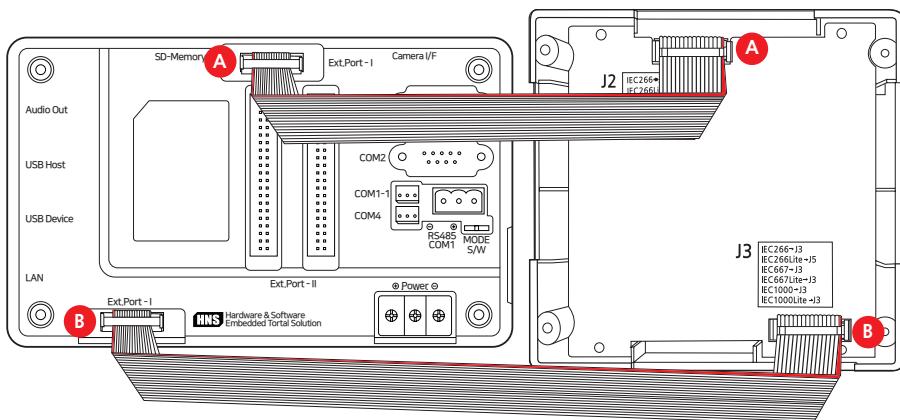
참고

Block별 장착 위치는 각 Smart I/O Base Board의 [3-3. Smart I/O-II, III의 Block 장착 위치] 또는 Block별 설명 [Block 장착 위치] 내용을 참고하시기 바랍니다.

6-3. Extension Port 연결에 문제가 없는지 확인

Smart I/O Base Board와 IEC-Series 제품은 Extension Port를 통해 연결하여 사용합니다. Extension Port를 연결할 때 Smart I/O Base Board와 IEC-Series의 동일한 Extension Port에 IDC Cable을 연결해야 합니다. 만약 다른 Port에 연결할 경우 정상적인 동작을 하지 않는 문제가 발생합니다.

[예시 : Smart I/O-II Base Board와 IEC-Series 제품을 IDC Cable로 연결]



위 그림과 같이 Extension Port - A는 A에 B는 B에 연결하여 동일한 Port에 연결되도록 합니다.

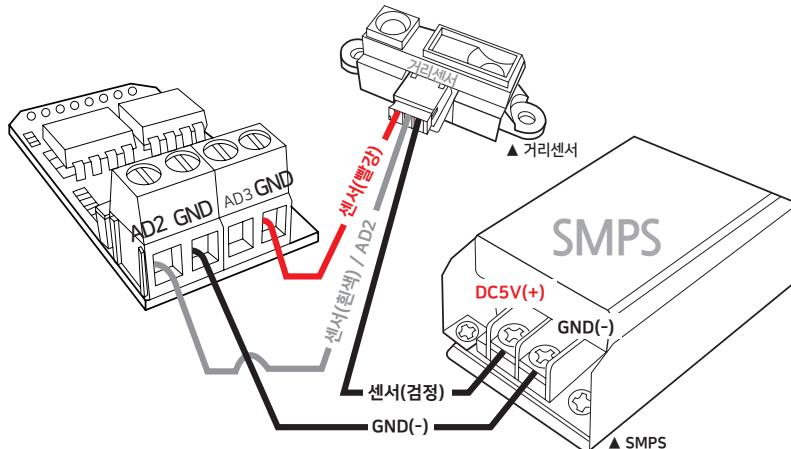
참고

Smart I/O Base Board별 Extension Port 연결 방법은 각 Smart I/O Base Board의 [2. IEC-Series와 Smart I/O-II, III 제품 연결하기(케이블 연결)] 내용을 참고하시기 바랍니다.

6-4. 결선에 문제가 없는지 확인

Smart I/O의 경우 외부 인터페이스 장치(센서, 모터 등)와 함께 사용하게 됩니다. 해당 외부 인터페이스 장치를 정확하게 결선하지 않는 경우 Smart I/O Board 및 외부 인터페이스 장치의 고장으로 이어질 수 있으며, 정상 동작을 하지 않는 문제가 발생할 수 있습니다.

[예시 : ADC Block 외부 인터페이스 장치 결선 이미지]

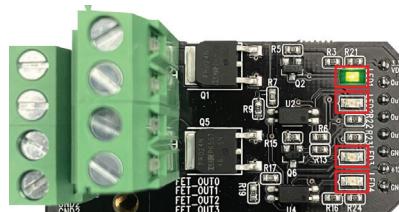


위 그림과 같이 각 Block의 내용에서 외부 인터페이스 장치 결선 이미지를 예시로 확인할 수 있습니다. 해당 디바이스의 데이터시트를 확인하여 결선이 정확하게 되었는지 확인 후 전원을 인가하시기 바랍니다.

6-5. Block의 LED 등으로 입출력 신호를 확인

Smart I/O Block-Series의 FET Block, Relay Block에는 LED가 포함되어 있어 출력 신호에 따라 LED가 점등됩니다. 해당 LED를 통해 실제로 출력 신호가 출력되고 있는지 확인이 가능합니다. 따라서 프로그램에서 출력 신호를 보냈지만 Block에서 LED가 점등되지 않는 경우 Block의 장착 위치가 맞는지 혹은 Base Board의 전원이 정상적으로 인가되었는지 확인이 필요합니다.

[FET Block LED 위치]



[Relay Block LED 위치]



참고

Input Module의 경우 입력 신호 간섭을 방지하기 위해 LED가 포함되어 있지 않습니다. Input은 LED로 확인할 수 없으므로 테스트기 등을 사용하여 전압의 변화로 Block의 상태를 확인합니다.

6-6. 관련 프로그램 코드가 올바른지 확인

Smart I/O Base Board의 전원 인가, Block 장착 위치, 결선, LED 점등 등에 문제가 없는 경우에는 프로그램 코드의 문제가 아닌지 점검이 필요합니다. Debug 모드를 통해 코드를 트레이스하여 코드에 입력되는 값과 출력되는 값을 확인하여 코드가 정상 동작하는지 확인합니다.

[예시 : TechNote 51. [C#, VB.NET] Smart I/O-I, II, III_통합 예제]

```
// PORT-A 감지 이벤트 처리 함수
private void GPIO_PortA_OnPortDataChange( int iPortDatas )
{
    //PORT-A의 각 핀 방향이 INPUT일 때 iPortDatas 값에 따른 Lamp On/Off
    if ( btnPortDirA0.ButtonStatus == SmartX.SmartButton.BUTSTATUS.UP )
    {
        pbLampA0.Image = (( iPortDatas & 1 ) == 1 ) ? Properties.Resources.on_led : Properties.Resources.off_led;
    }
}
```

코드에서 입력 및 출력이 잘 되는지 혹은 입력 및 출력된 값이 올바른지 확인할 수 있습니다.

참조

Smart I/O 예시 프로그램은 [홈페이지](www.hnsts.co.kr) – [자료실] – [Tech Note]에서 확인 및 다운로드 받을 수 있습니다.

※ Block별 사용되는 SmartX Framework 예제

아래 표를 확인하여 Block에 사용되는 SmartX Framework Component를 확인하여 예제 프로그램을 통해 간단한 테스트를 진행할 수 있습니다.

[예제 프로그램 경로] : SmartX 홈페이지(www.smartx.co.kr) → 예제 프로그램(사용중인 개발 언어에 맞게 다운로드)

Block-Series	SmartX Framework 관련 예제
Input Block	SmartGPIO 예제
ADC Block	SmartADC 예제
RS232 Isolation Block	SmartSerialPort 예제
FET Block	SmartGPIO, SmartPWM 예제
Relay Block	SmartGPIO 예제
DAC Block	SmartDAC 예제
InputCounter Block	SmartInputCounter 예제

7. Smart I/O-I 제품 단종에 따른 Smart I/O-II 제품으로 Smart I/O-I과 같이 구성하는 방법

Smart I/O-I 제품이 단종되어 더 이상 구입 및 사용할 수 없을 때 Smart I/O-II 제품으로 Smart I/O-I의 환경을 구성할 수 있는 방법입니다. Smart I/O-I를 기존에 사용 중이거나 Smart I/O-I를 구입하고자 하는 경우에는 아래 STEP을 참고하여 Smart I/O-II로 환경을 구성하시기 바랍니다.

STEP-1

아래의 Block들을 사용하여 구성합니다.

사용 Block 이름	Relay Block	FET Block	Input Block
사용될 개수	1개	1개	2개
Block 이미지	A relay block with two sets of contacts labeled A1/A2 and B1/B2.	An FET driver block with four channels labeled A1/A2, B1/B2, C1/C2, and D1/D2.	An input block with two sets of terminals labeled A1/A2 and B1/B2.

참고

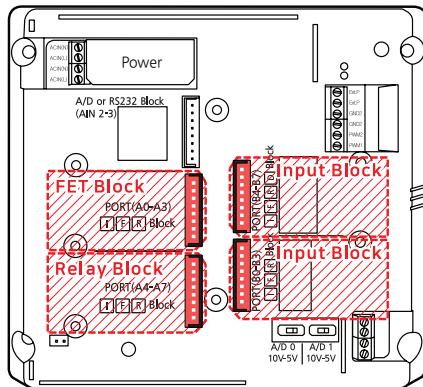
ADC와 PWM의 경우 Smart I/O-II Base Board에 단자가 내장되어 있기 때문에 Block을 장착하지 않고 사용할 수 있습니다.

STEP-2

Smart I/O-II Base Board의 상부 케이스 나사를 풀어 케이스를 해제합니다.

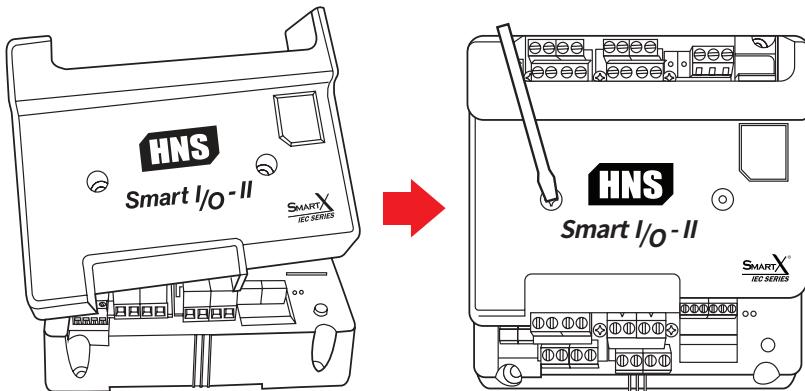
STEP-3

준비한 Block을 아래 이미지와 같이 올바른 위치에 장착한 후 Block을 M3X6(PH) 볼트로 고정합니다.



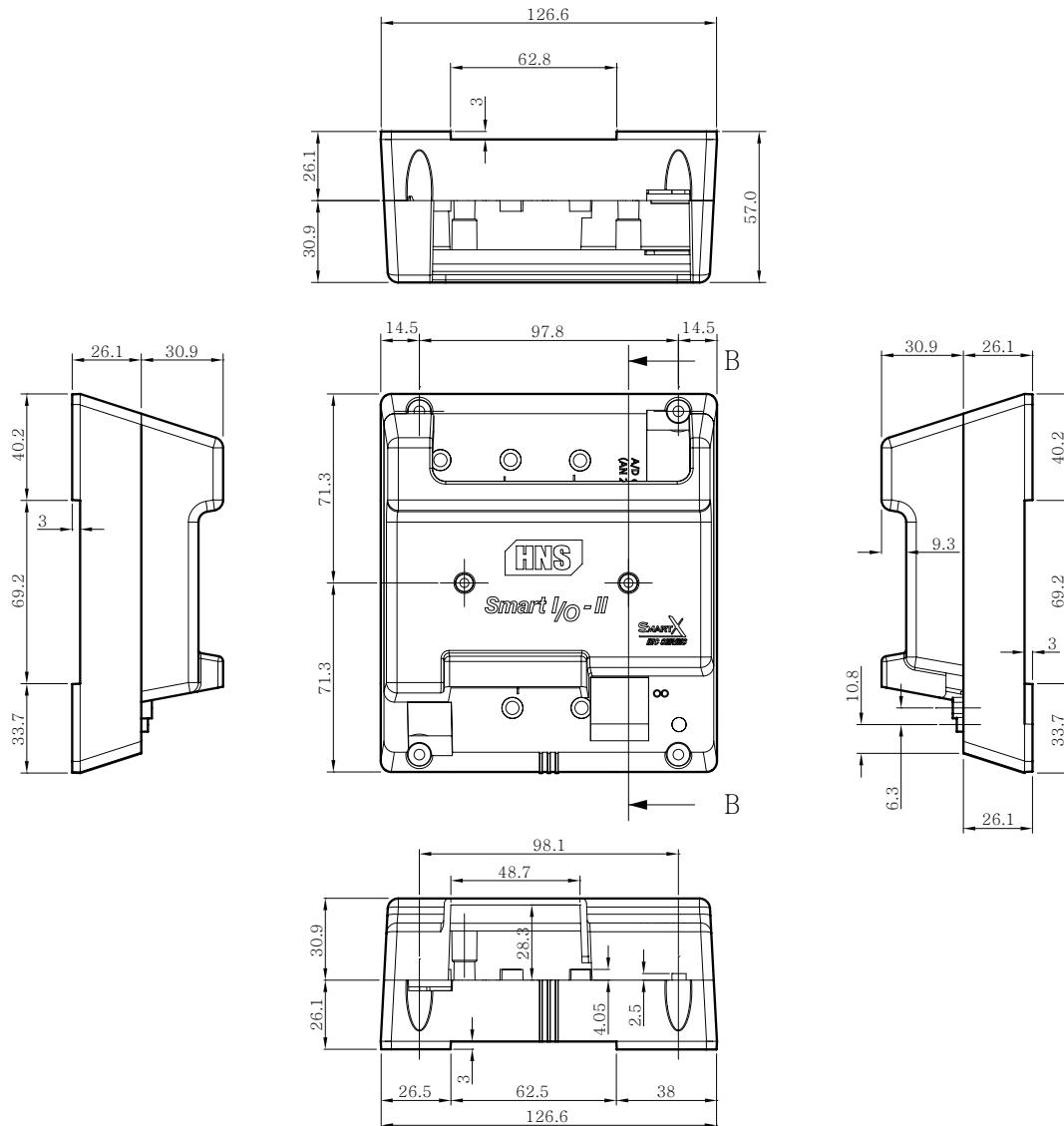
STEP-3

Block 장착이 완료된 후 상부 케이스를 조립하여 마무리합니다.



Part 2

Smart I/O-II Base Board

Part-II. Smart I/O-II Base Board**1. 외형 치수****참고**

본 제품의 도면은 “[홈페이지](www.hnsts.co.kr)” – [자료실] – [도면 및 승인원] – [3. 도면 및 승인원]”에서 다운로드 할 수 있습니다.

참고

Smart I/O-II Base Board를 기구 케이스에 장착할 때는 4개의 고정홀을 기구 케이스에 탭(Tap hole) 처리한 뒤 볼트로 고정해야 합니다. 관련된 자세한 사항은 [Part-I. Smart I/O-Series 시작하기] – [5-3. Smart I/O-II, III 기구 디자인 가이드] 을 참고하시기 바랍니다.

2. IEC-Series와 Smart I/O-II 제품 연결하기 (케이블 연결)

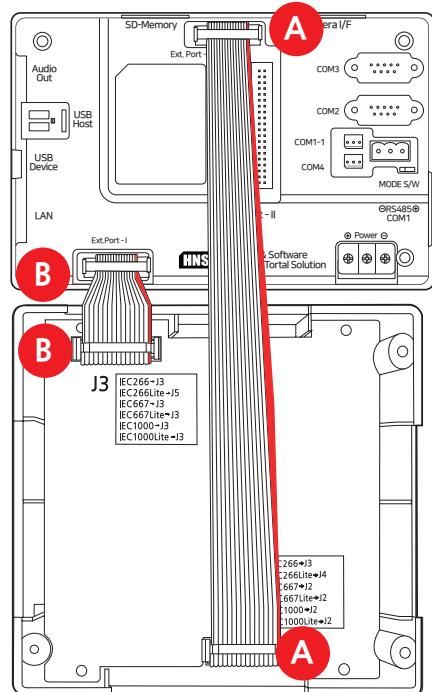


주의

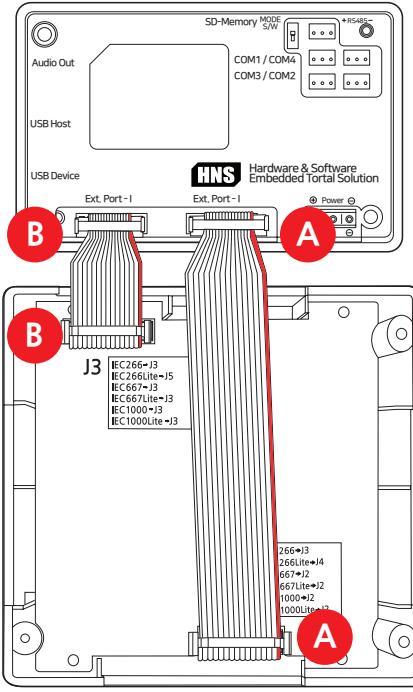
- 연결할 때 반드시 IEC-Series 전원을 OFF한 후 연결하여 사용하시기 바랍니다. (전원 ON 상태에서 연결 시 고장의 원인이 됩니다.)
- Extension Port - I, II의 IDC Cable Set의 경우 50cm 이하의 길이로 사용하는 것을 권장합니다. 50cm 이상으로 길이를 연장하는 경우 제품의 정상적인 동작을 보장하지 못합니다.

[연결사용 안내]

IEC667/1000 非Lite-Series



IEC667/1000 Lite-Series

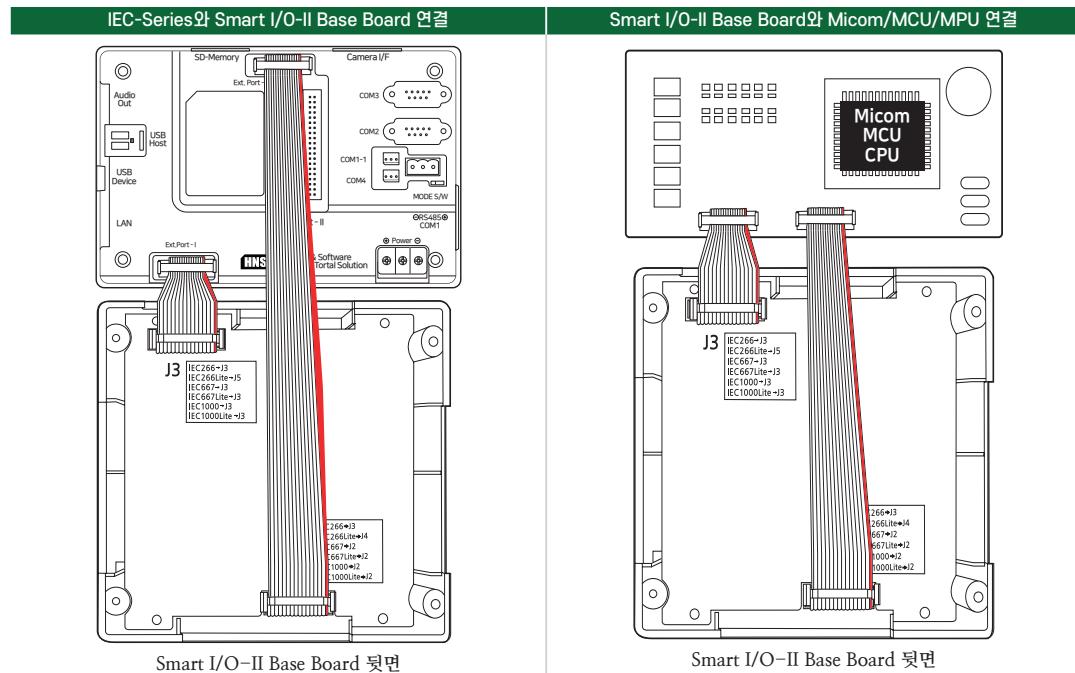


IEC 非Lite-Series	연결	Smart I/O-II	IEC Lite-Series	연결	Smart I/O-II
A	↔	A	A	↔	A
B	↔	B	B	↔	B

3. Smart I/O-II Base Board의 판맵 안내 및 Header 커넥터 연결 상태

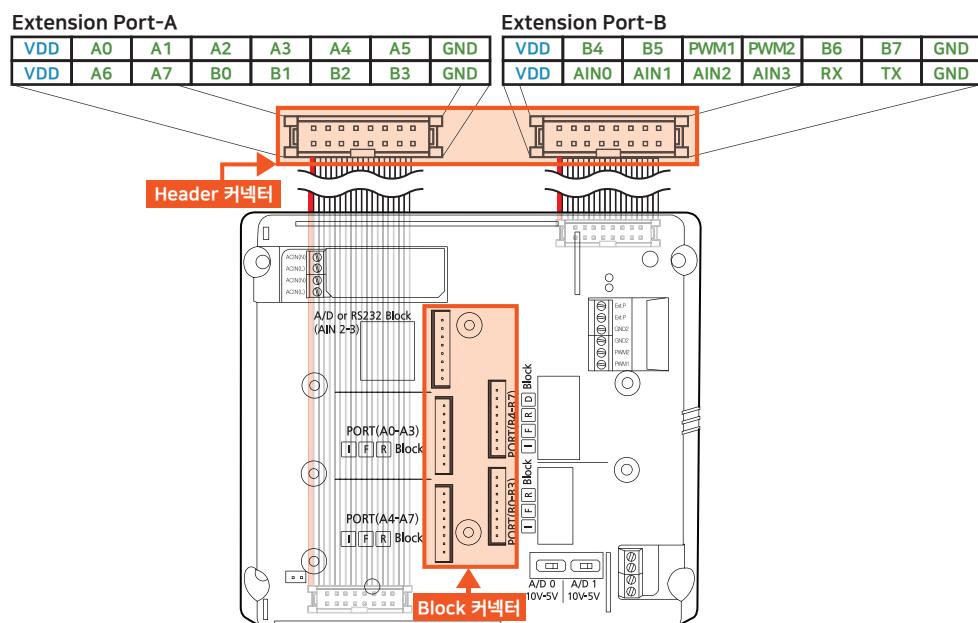
3-1. Smart I/O-II Base Board 호환성

IEC-Series에서만 지원되는 것을 다른 시스템(MICOM, MCU, MPU, 아두이노, 라즈베리파이 등)에서도 연동이 가능하도록 연결 및 제어에 필요한 정보를 제공하여 Smart I/O-II Base Board의 활용도를 높였습니다.



3-2. Block 커넥터와 Header 커넥터의 연결 상태 및 명칭

Block 커넥터의 핀과 Header 커넥터의 핀이 직접적으로 연결되어 있는 경우 핀의 색상으로 구분되어 있으니 참고하시기 바랍니다. (※ 초록색 : 직접 연결)

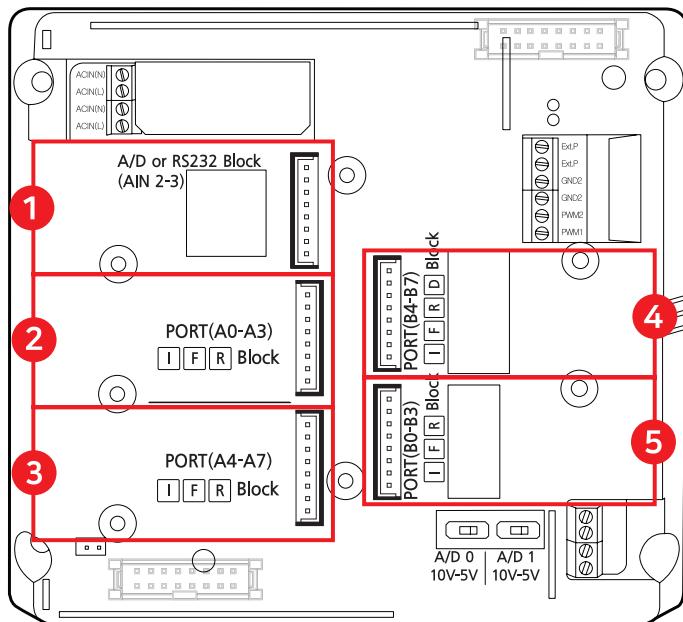


주의

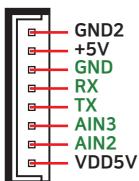
케이블 방향에 따라 핀의 위치가 변경될 수 있으며 잘못된 결선 시 Smart I/O-II Base Board의 고장이 발생할 수 있으므로 케이블 방향과 핀의 위치를 정확히 확인하고 결선하시기 바랍니다.

참고

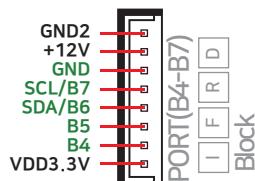
자세한 핀의 전기적 사항은 “3-4. Smart I/O-II Base Board 연결 정보(설명) 및 전기적 사양” 내용을 참고하시기 바랍니다.



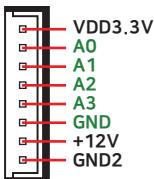
1 A/D or RS232 Block
(AIN 2-3)



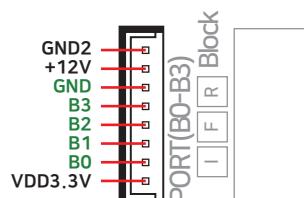
4



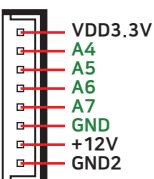
2 PORT(A0-A3)
I F R Block



5



3 PORT(A4-A7)
I F R Block



3-3. Smart I/O-II의 Block 장착 위치

Block별 장착 가능 위치를 표를 참고하여 확인하시기 바랍니다

No.	각부 명칭	Input Block	FET Block	Relay Block	ADC Block	DAC Block	Input.C Block	RS232 Block
1	A/D(AIN2~3), RS232 Isolation	-	-	-	장착 가능	-	-	장착 가능
2	PORT A(A0~A3)	장착 가능	장착 가능	장착 가능	-	-	장착 가능	-
3	PORT A(A4~A7)	장착 가능	장착 가능	장착 가능	-	-	장착 가능	-
4	PORT B(B4~B7) [Port B6, B7 → I2C]	장착 가능	장착 가능	장착 가능	-	장착 가능	장착 가능	-
5	PORT B(B0~B3)	장착 가능	장착 가능	장착 가능	-	-	장착 가능	-

3-4. Smart I/O-II Base Board 연결 정보(설명) 및 전기적 사양

VDD(전원 입력)의 DC 12V@300mA의 전류 값은 Relay Block을 모든 포트에 연결했을 때 기준이며, Smart I/O Base Board에 인가되는 용량이기 때문에 실제로 사용할 때는 IEC-Series 입력 전원 용량을 IEC-Series 제품별 전원 용량 + Smart I/O Base Board 용량(300mA) + 30%로 사용하시기 바랍니다. IEC-Series의 제품별 전원 용량은 “[IEC-Series 제품 매뉴얼] – [Part-I. IEC-Series 기본 사양] – [8. 제품별 사용 용량]”을 참고하시기 바랍니다.

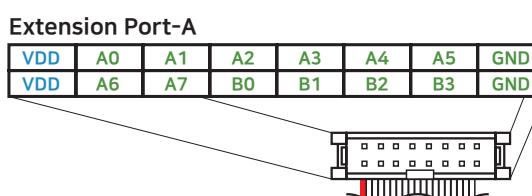
중요



Power = IEC-Series 전원 용량 + Smart I/O Base Board에서 사용되는 VDD 용량(300mA) + 30%

* 따라서 IEC-Series의 전원 용량은 기존에 사용하던 용량보다 더 높게 설정하여 사용하시기 바랍니다.

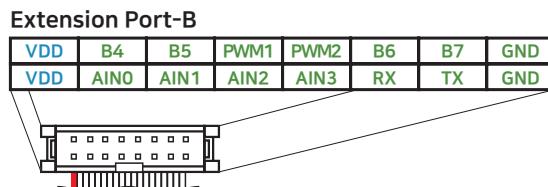
① 하단 커넥터(Extension Port-A)



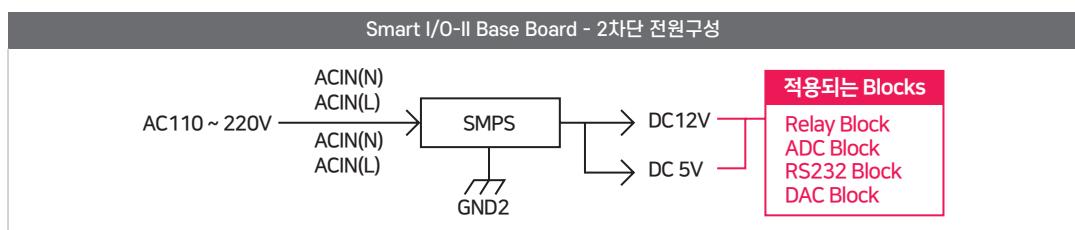
명칭	설명	전기적 사양
VDD	전원 입력(공급)	DC 12 ~ 24V (DC 12V@300mA) ※ 상단 참고 박스 확인
A0	Port A0 입/출력	
A1	Port A1 입/출력	
A2	Port A2 입/출력	DC 0V, 3.3V
A3	Port A3 입/출력	10mA(핀당 전류)
A4	Port A4 입/출력	
A5	Port A5 입/출력	
GND	Ground (1차단)	GND
VDD	전원 입력(공급)	DC 12 ~ 24V (DC 12V@300mA) ※ 상단 참고 박스 확인

A6	Port A6 입/출력	
A7	Port A7 입/출력	
B0	Port B0 입/출력	DC 0V, 3.3V
B1	Port B1 입/출력	10mA(핀당 전류)
B2	Port B2 입/출력	
B3	Port B3 입/출력	
GND	Ground (1차단)	GND

② 상단 커넥터(Extension Port-B)



명칭	설명	전기적 사양
VDD	전원 입력(공급)	DC 12 ~ 24V (DC 12V@300mA) ※ 상단 참고 박스 확인
B4	Port B4 입/출력	
B5	Port B5 입/출력	
PWM1	PWM1 입력	DC 0V, 3.3V
PWM2	PWM2 입력	10mA(핀당 전류)
B6	Port B6 입/출력	
B	Port B7 입/출력	
GND	Ground (1차단)	GND
VDD	전원 입력(공급)	DC 12 ~ 24V (DC 12V@300mA) ※ 상단 참고 박스 확인
ADIN0		
ADIN1		DC 0V, 3.3V
ADIN2		
ADIN3		
RX	통신 수신(RX) TTL-5V, COM	DC 0V, 5V
TX	통신 수신(TX) TTL-5V, COM	
GND	Ground (1차단)	GND



3-5. Smart I/O-II Base Board 기본 내부 기능

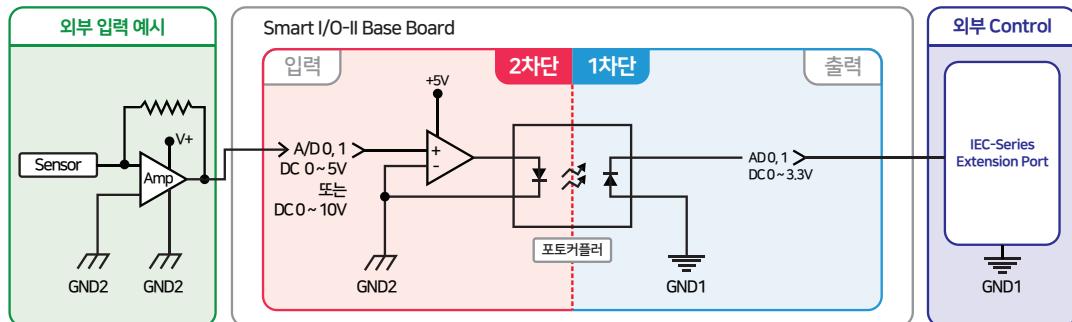
Smart I/O-II Base Board는 Smart I/O-III Base Board와 다르게 ADC Block 2채널, PWM(FET) Block 2채널이 기본적으로 내장되어 있습니다.

※ I/O 구성

기능	FET-PWM	ADC - IN
내장 기능 채널 수	2Ch	4Ch

1) Analog Input(ADC - IN0, 1)

입력 측의 전압 범위는 전압 범위 설정 스위치에 따라서 DC 0 ~ 5V or DC 0 ~ 10V 입니다. 각각의 전압 입력은 내부 OP Amp를 통하여 DC 0 ~ 3.3V 비례하여 변환되어 출력됩니다.

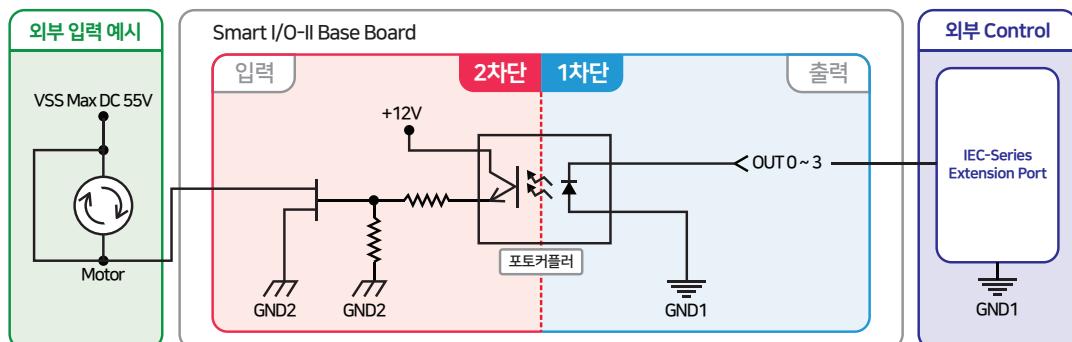


참고

Analog Input(ADC - IN0, 1)에 대한 자세한 사항은 “[Part-IV. Smart I/O – Block Series] → [5. ADC Block]”의 내용을 참고하시기 바랍니다.

2) PWM(FET) Output(PWM 0, 1)

PWM 출력은 최대 2채널을 사용할 수 있으며 FET Block으로 연결되어 있습니다. FET은 Open Drain으로 구성되어 있으며 Gate 단의 신호에 따라서 FET는 Turn On/Off 상태가 됩니다.



참고

PWM(FET) Output(PWM 0, 1)에 대한 자세한 사항은 “[Part-IV. Smart I/O – Block Series] → [8. FET Block]”의 내용을 참고하시기 바랍니다.

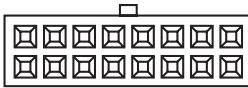
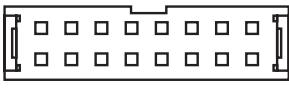
주의

1. 전력량에 따라 FET 소자에 발열이 발생할 수 있으므로, 발열이 적게 발생하도록 발열량을 검토하고 전력량을 줄여 사용하시기 바랍니다.
2. IRFR024N은 VSS = 55V, $I_d = 17A$ 정격의 FET입니다.
3. Smart I/O-Series에서는 PWM 파형이 반전되어 출력됩니다.

참조

PWM(FET) Output(PWM0, 1)의 응용 예제 프로그램은 “[홈페이지(www.hnsts.co.kr)] → [자료실] → [Tech Note] → [54. [C#, VB.NET] Smart I/O – Series PWM과 FET – Block 사용 예제]”를 참조하시기 바랍니다.

4. Smart I/O-II 커넥터 사양 안내

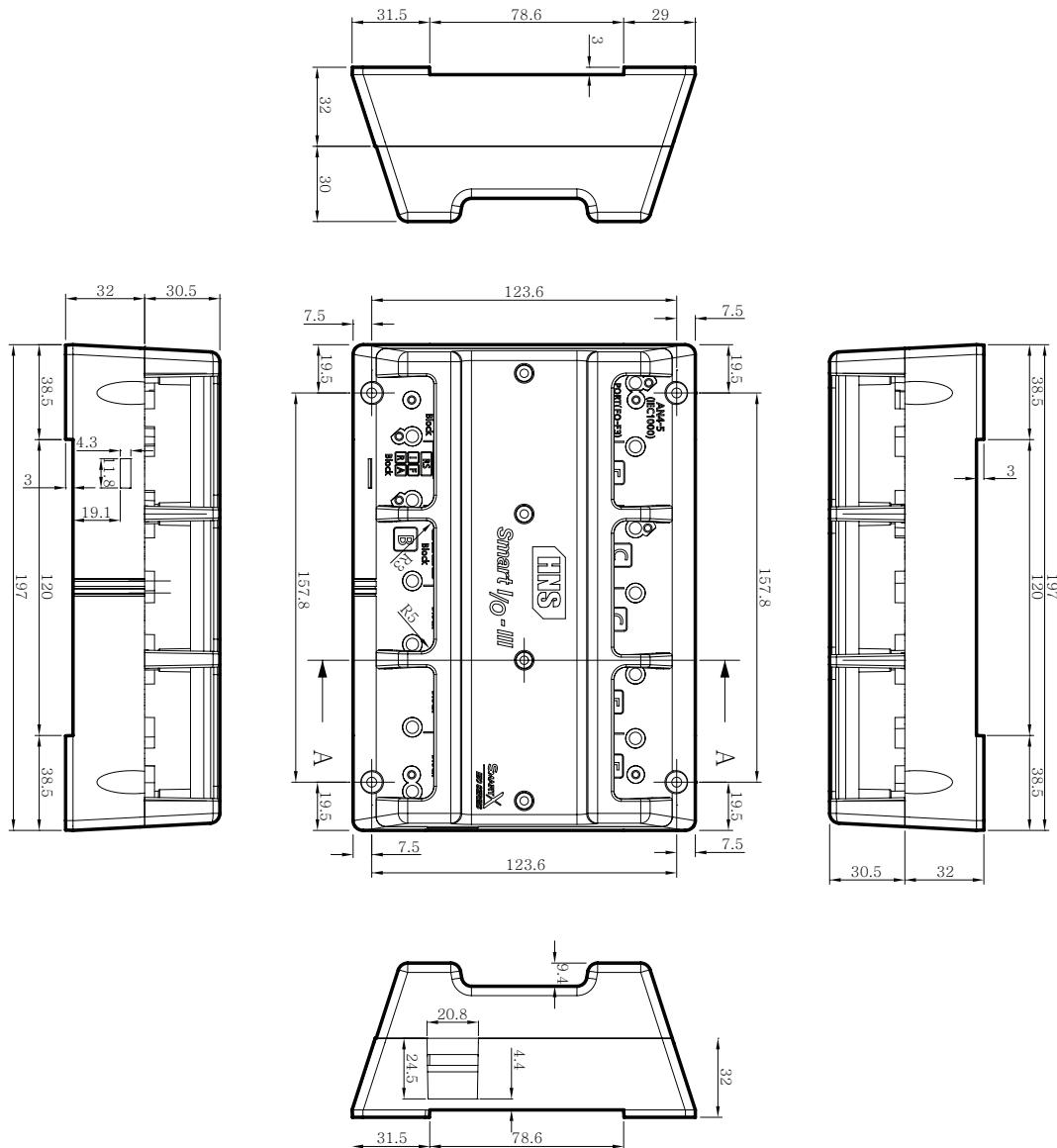
	16핀 보드 방향	16핀 케이블 맞은편
제품사진		
위치번호/명칭 Part Number Vendor	FH14-16DS-1 FH14-16DS-1 Any Vendor(범용적)	FL12-16 FL12-16A-S-B-2

MEMO

3

Part 3

Smart I/O-III Base Board

Part-III. Smart I/O-III Base Board**1. 외형 치수****참조**

본 제품의 도면은 “[홈페이지](www.hnsts.co.kr) – [자료실] – [도면 및 승인월] – [3. 도면 및 승인월]”에서 다운로드할 수 있습니다.

참고

Smart I/O-III Base Board를 기구 케이스에 장착할 때는 4개의 고정홀을 기구 케이스에 탭(Tap hole) 처리한 뒤 볼트로 고정해야 합니다. 관련된 자세한 사항은 [Part-I. Smart I/O-Series 시작하기] – [5-3. Smart I/O-II, III 기구 디자인 가이드] 내용을 참고하시기 바랍니다.

2. IEC-Series와 Smart I/O-III 제품 연결하기(케이블 연결)

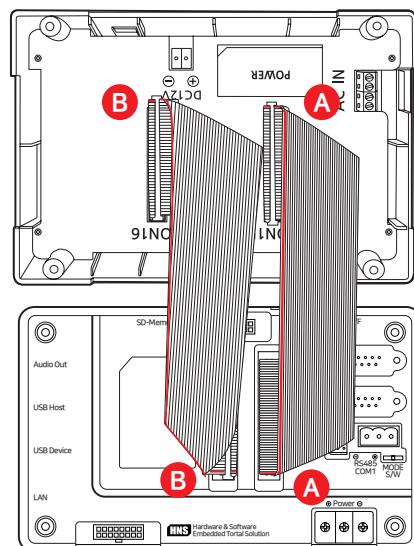


주의

- 연결할 때 반드시 IEC-Series 전원을 OFF한 후 연결하여 사용하시기 바랍니다. (전원 ON 상태에서 연결 시 고장의 원인이 됩니다.)
- Extension Port – I, II의 IDC Cable Set의 경우 50cm 이하의 길이로 사용하는 것을 권장합니다. 50cm 이상으로 길이를 연장하는 경우 제품의 정상적인 동작을 보장하지 못합니다.

[연결사용 안내]

IEC667/1000 非Lite-Series

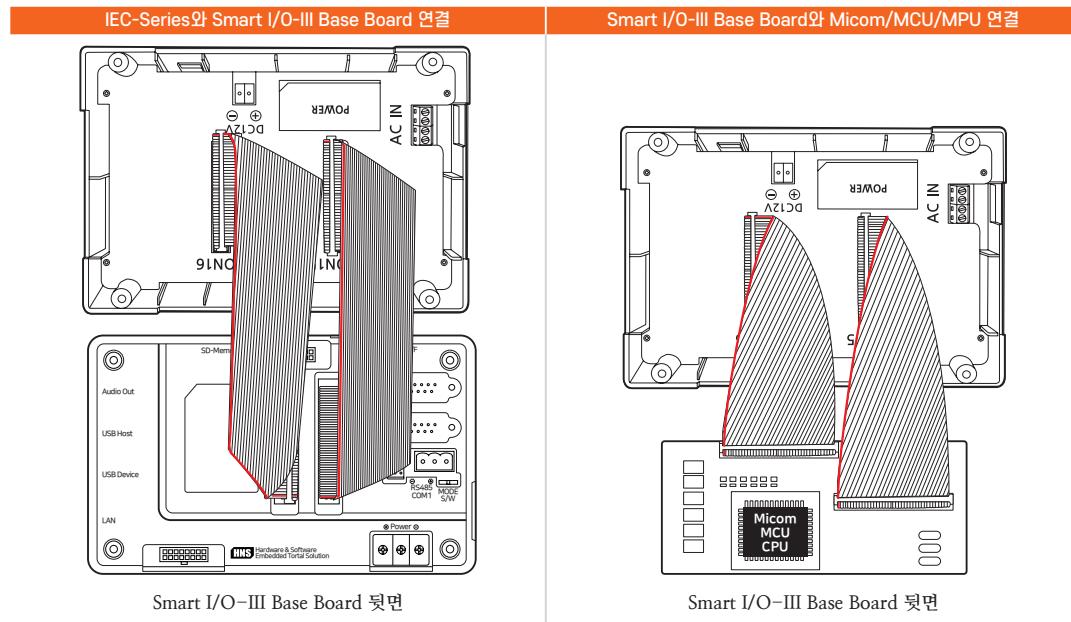


IEC 非Lite-Series	연결	Smart I/O-III
A	↔	A
B	↔	B

3. Smart I/O-III Base Board의 핀맵 안내 및 Header 커넥터 연결 상태

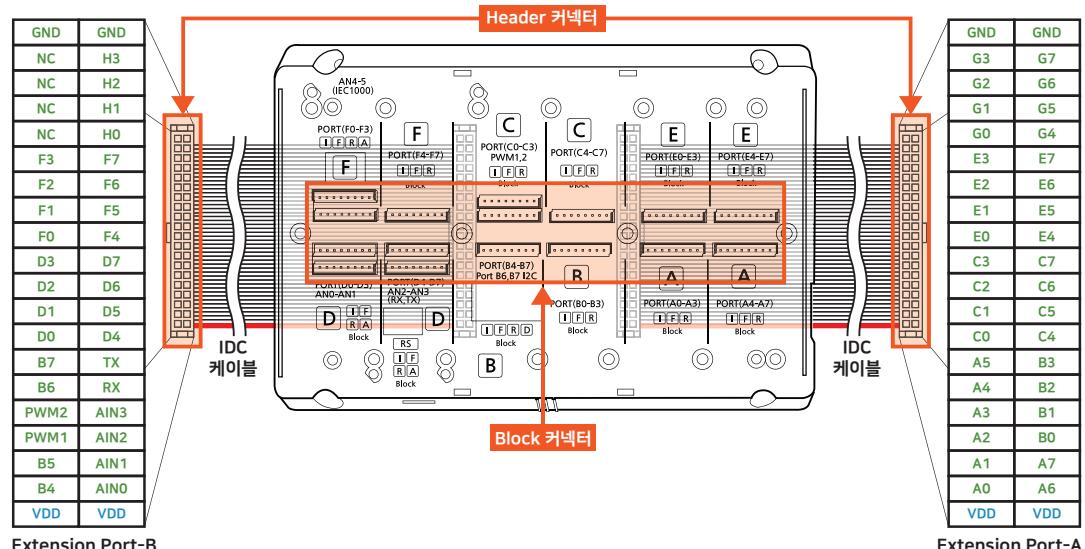
3-1. Smart I/O-III Base Board 호환성

IEC-Series에서만 지원되는 것을 다른 시스템(MICOM, MCU, MPU, 아두이노, 라즈베리파이 등)에서도 연동이 가능하도록 연결 및 제어에 필요한 정보를 제공하여 Smart I/O - III Base Board의 활용도를 높였습니다.



3-2. Block 커넥터와 Header 커넥터의 연결 상태 및 명칭

Block 커넥터의 핀과 Header 커넥터의 핀이 직접적으로 연결되어 있는 경우 핀의 색상으로 구분되어 있으니 참고하시기 바랍니다. (※ 초록색 : 직접 연결)



Extension Port-B

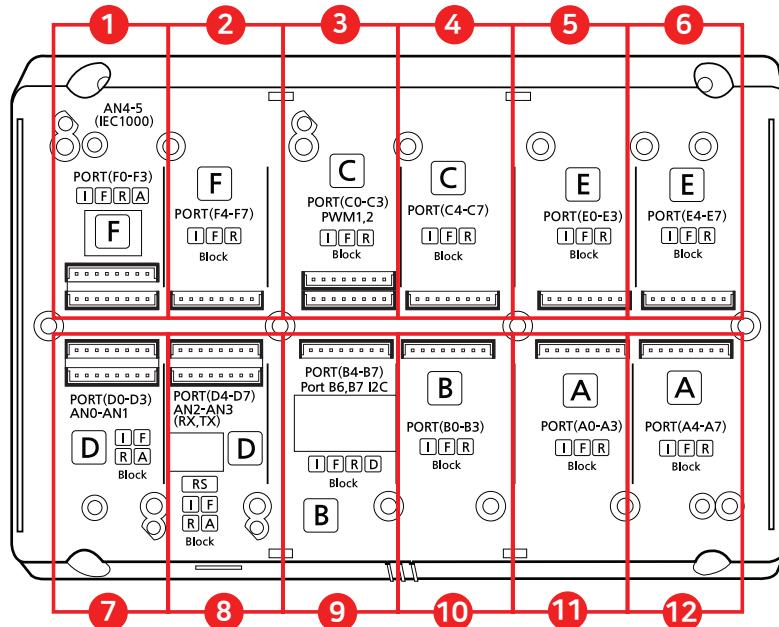
Extension Port-A

주의

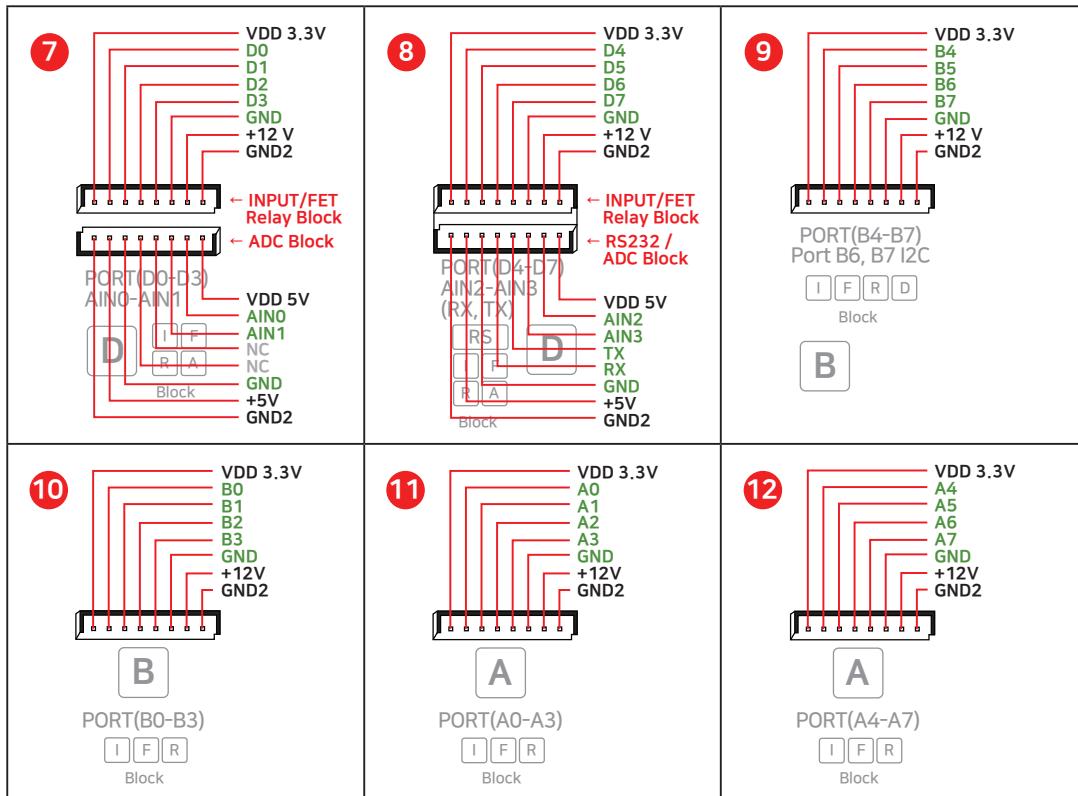
케이블 방향에 따라 핀의 위치가 변경될 수 있으며 잘못된 결선 시 Smart I/O-II Base Board의 고장이 발생할 수 있으므로 케이블 방향과 핀의 위치를 정확히 확인하고 결선하시기 바랍니다.

참고

자세한 핀의 전기적 사항은 “3-4. Smart I/O-III Base Board 연결 정보(설명) 및 전기적 사양” 내용을 참고하시기 바랍니다.



1	PORT(F0-F3) AIN4 AIN5 NC NC GND +5V GND2	2	F PORT(F4-F7) I F R Block	3	GND2 +12V GND NC NC PORT(C0-C3) PWM2 PWM1 VDD 3.3V
	← ADC Block ← INPUT/FET Relay Block	VDD 3.3V F4 F5 F6 F7 GND +12V GND2		← FET(PWM) Block ← INPUT/FET Relay Block	VDD 3.3V C0 C1 C2 C3 GND +12V GND2
4	C PORT(C4-C7) I F R Block	5	E PORT(E0-E3) I F R Block	6	E PORT(E4-E7) I F R Block
	VDD 3.3V C4 C5 C6 C7 GND +12V GND2		VDD 3.3 E0 E1 E2 E3 GND +12V GND2		VDD 3.3V E4 E5 E6 E7 GND +12V GND2



3-3. Smart I/O-III의 Block 장착 위치

Block별 장착 가능 위치를 표를 참고하여 확인하시기 바랍니다.

No.	각부 명칭	Input Block	FET Block	Relay Block	ADC Block	DAC Block	Input.C Block	RS232 Block
1	PORT F0 ~ F3 A/D(AIN 4~5)	장착 가능	장착 가능	장착 가능	장착 가능	-	-	-
2	PORT F4 ~ F7	장착 가능	장착 가능	장착 가능	-	-	-	-
3	PORT C0 ~ C3 PWM1, 2	장착 가능	장착 가능	장착 가능	-	-	-	-
4	PORT C4 ~ C7	장착 가능	장착 가능	장착 가능	-	-	-	-
5	PORT E0 ~ E3	장착 가능	장착 가능	장착 가능	-	-	-	-
6	PORT E4 ~ E7	장착 가능	장착 가능	장착 가능	-	-	-	-
7	PORT D0 ~ D3 A/D(AIN 0~1)	장착 가능	장착 가능	장착 가능	장착 가능	-	-	-
8	PORT D4 ~ D7 A/D(AIN 2~3), RS232	장착 가능	장착 가능	장착 가능	장착 가능	-	-	장착 가능
9	PORT B4 ~ B7	장착 가능	장착 가능	장착 가능	-	장착 가능	장착 가능	-
10	PORT B0 ~ B3	장착 가능	장착 가능	장착 가능	-	-	장착 가능	-
11	PORT A0 ~ A3	장착 가능	장착 가능	장착 가능	-	-	장착 가능	-
12	PORT A4 ~ A7	장착 가능	장착 가능	장착 가능	-	-	장착 가능	-

3-4. Smart I/O-III Base Board 연결 정보(설명) 및 전기적 사양

VDD(전원 입력)의 DC 12V@720mA의 전류 값은 Relay Block을 모든 포트에 연결했을 때 기준이며, Smart I/O Base Board에 인가되는 용량이기 때문에 실제로 사용할 때는 IEC-Series 입력 전원 용량을 IEC-Series 제품별 전원 용량 + Smart I/O Base Board 용량(720mA) + 30%로 사용하시기 바랍니다. IEC-Series의 제품별 전원 용량은 “[IEC-Series 제품 매뉴얼] – [Part-I. IEC-Series 기본 사양] – [8. 제품별 사용 용량]”을 참고하시기 바랍니다.

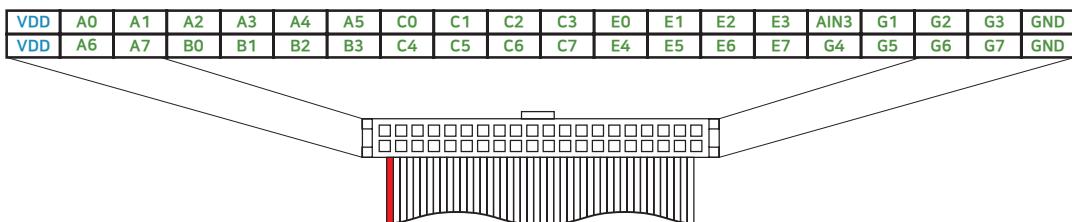
중요



Power = IEC-Series 전원 용량 + Smart I/O Base Board에서 사용되는 VDD 용량(720mA) + 30%

※ 따라서 IEC-Series의 전원 용량은 기존에 사용하던 용량보다 더 높게 설정하여 사용하시기 바랍니다.

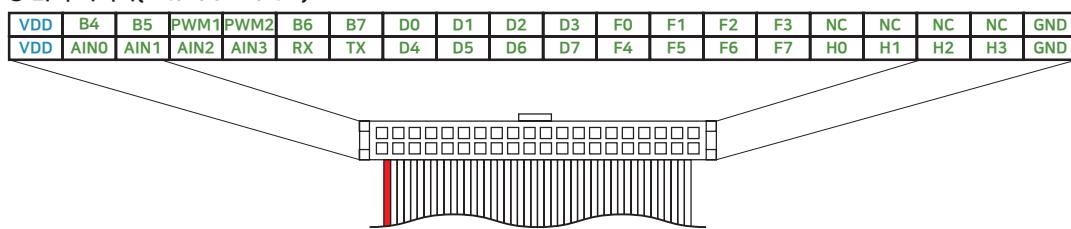
① 우측 커넥터(Extension Port-A)



명칭	설명	전기적 사양
VDD	전원 입력(공급)	DC 12 ~ 24V (DC 12V@720mA) ※ 상단 참조 박스 확인
A0	Port A0 입/출력	
A1	Port A1 입/출력	
A2	Port A2 입/출력	
A3	Port A3 입/출력	
A4	Port A4 입/출력	
A5	Port A5 입/출력	
C0	Port C0 입/출력	
C1	Port C1 입/출력	
C2	Port C2 입/출력	DC 0V, 3.3V 10mA(핀당 전류)
C3	Port C3 입/출력	
E0	Port E0 입/출력	
E1	Port E1 입/출력	
E2	Port E2 입/출력	
E3	Port E3 입/출력	
G0	Port G0 입/출력	
G1	Port G1 입/출력	
G2	Port G2 입/출력	
G3	Port G3 입/출력	
GND	Ground (1차단)	GND
VDD	전원 입력(공급)	DC 12 ~ 24V (DC 12V@720mA) ※ 상단 참조 박스 확인
A6	Port A6 입/출력	
A7	Port A7 입/출력	
B0	Port B0 입/출력	DC 0V, 3.3V 10mA(핀당 전류)
B1	Port B1 입/출력	
B2	Port B2 입/출력	

B3	Port B3 입/출력
C4	Port C4 입/출력
C5	Port C5 입/출력
C6	Port C6 입/출력
C7	Port C7 입/출력
E4	Port E4 입/출력
E5	Port E5 입/출력
E6	Port E6 입/출력
E7	Port E7 입/출력
G4	Port G4 입/출력
G5	Port G5 입/출력
G6	Port G6 입/출력
G7	Port G7 입/출력
GND	Ground (1차단)
	DC 0V, 3.3V 10mA(핀당 전류) GND

② 좌측 커넥터(Extension Port-B)

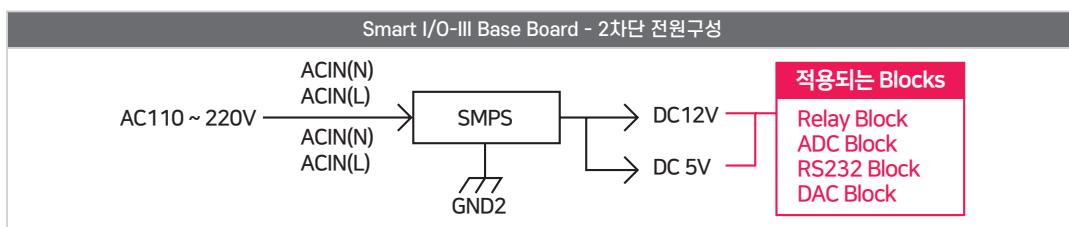


명칭	설명	전기적 사양
VDD	전원 입력(공급)	DC 12 ~ 24V (DC 12V@720mA) ※ 상단 참고 박스 확인
B4	Port B4 입/출력	
B5	Port B5 입/출력	
PWM1	PWM1 입력	
PWM2	PWM2 입력	
B6	Port B6 입/출력	
B7	Port B7 입/출력	
D0	Port D0 입/출력	DC 0V, 3.3V
D1	Port D1 입/출력	10mA(핀당 전류)
D2	Port D2 입/출력	
D3	Port D3 입/출력	
F0	Port F0 입/출력	
F1	Port F1 입/출력	
F2	Port F2 입/출력	
F3	Port F3 입/출력	
NC	공핀	-
NC	공핀	-
AIN4	ADC 출력	
AIN5	※ IEC1000 非Lite-Series Only	DC 0 ~ 3.3V
GND	Ground (1차단)	GND
VDD	전원 입력(공급)	DC 12 ~ 24V (DC 12V@720mA) ※ 상단 참고 박스 확인
AIN0		
AIN1		
AIN2	ADC 출력	DC 0 ~ 3.3V
AIN3		
Rx	통신 핀 Rx COM4 TTL-5V	DC 0V, 5V
Tx	통신 핀 Tx COM4 TTL-5V	
D4	Port D4 입/출력	DC 0V, 3.3V
D5	Port D5 입/출력	10mA(핀당 전류)

D6	Port D6 입/출력	
D7	Port D7 입/출력	
F4	Port F4 입/출력	
F5	Port F5 입/출력	
F6	Port F6 입/출력	DC 0V, 3.3V
F7	Port F7 입/출력	10mA(핀당 전류)
H0	Port H0 입/출력	
H1	Port H1 입/출력	
H2	Port H2 입/출력	
H3	Port H3 입/출력	
GND	Ground (1차단)	GND

참고

※ VDD 전원 입력(공급)은 Smart I/O-III Base Board에 입력되는 전원으로 DC 12 ~ 24V의 구동 전압이 필요합니다. VDD의 Block들의 구동용 공급 전압으로 1차단의 다양한 전압들(DC 5V, DC 3.3V)을 내부적으로 만드는 소스로 사용되는 전원입니다.

**3-5. Smart I/O-III Base Board 기본 내부 기능**

Smart I/O-III Base Board는 내부 기능 Block을 포함하고 있지 않습니다.

※ I/O 구성

기능	FET-PWM	ADC - IN
내장 기능 채널 수	-	-

4. Smart I/O-III 커넥터 사양 안내

	40핀 보드 방향	40핀 케이블 맞은편
제품사진		
위치번호/명칭 Part Number	BH01-XXSG BH01-40SG	2.54 IDC Y-TYPE XXPIN FL01-40DR
Vendor	Any Vendor(범용적)	

MEMO

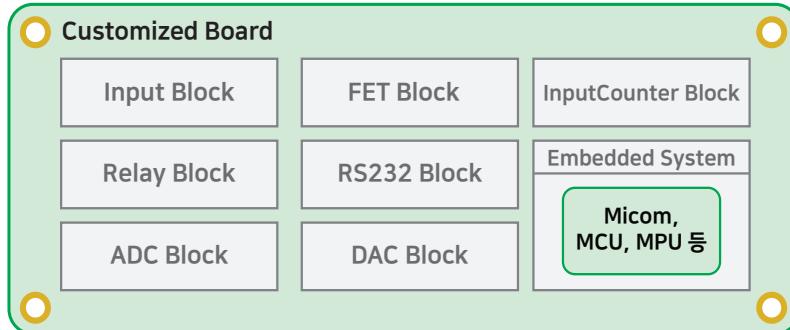
Part 4

Smart I/O-Block Series

Part - IV. Smart I/O-Block Series

1. Smart I/O-Block Series 호환성

IEC-Series에서만 지원되는 것을 다른 시스템(MICOM, MCU, MPU, U-Process, One-Chip 등)에서도 연동이 가능하도록 연결 및 제어에 필요한 정보를 제공하여 Smart I/O – Block Series의 활용도를 높였습니다.



참고

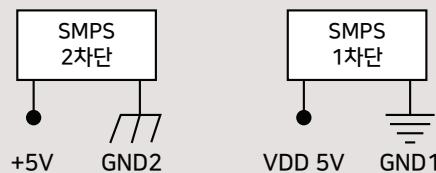
Smart I/O – II Base Board 또는 Smart I/O – III Base Board를 사용하는 경우 Block의 전원은 모두 Base Board에서 공급되므로 별도의 전원을 구성할 필요가 없습니다.

권장

Smart I/O – II, III Base Board를 사용하지 않는 경우 Block의 전원 공급 권장 사항

포토 커플러를 통해 전원이 분리(Isolation)되어 있으며 1차단 DC 5V와 2차단 DC 5V 공급이 필요하며 이때 전원의 GND는 각각 분리되어 있는 것을 권장합니다.

※ 입력 전압 DC 5V는 예시이며 Block마다 3.3V, 5V, 12V 등 입력 전압이 상이하므로 각 Block의 연결단자 설명에 있는 전기적 사양을 참고하시기 바랍니다.



2. Smart I/O-Block Series 사양

제품군	출력기능		
	Relay Block	FET Block	DAC Block
사진			
기능	Relay 접점 출력	FET 출력	12Bit 아날로그 출력
채널수	4	4	2
전기적 사양	Max 5A/250VAC 5A/0~30VDC	VSS(Max 55V / 49A) 주파수 Max 70kHz	0~5V, 0~10V(12Bit)
Smart I/O-II 최대 허용개수	4(16Ch)	4(16Ch)	1(2Ch)
Smart I/O-III 최대 허용개수	12(48Ch)	12(48Ch)	1(2Ch)

제품군	입력기능		
	ADC Block	Input Block	InputCounter Block
사진			
기능	아날로그 신호 입력	입력(무극성 포토 커플러)	입력(고속 입력 카운터) Rotary Encoder 입력
채널수	2	4	2 (Rotary Encoder 1Ch)
전기적 사양	0 ~ 5V(12Bit) 0 ~ 10V(12Bit)	DC12 ~ 24V	DC12 ~ 24V
Smart I/O-II 최대 허용개수	1(2Ch)	4(16Ch)	4(8Ch) PulseCounter 8Ch, RotaryEncoder 4Ch
Smart I/O-III 최대 허용개수	2(4Ch) ※IEC1000 非Lite는 3(6Ch)	12(48Ch)	4(8Ch) PulseCounter 8Ch, RotaryEncoder 4Ch

제품군	통신기능	
	RS232 Isolation Block	
사진		
기능	RS232 통신 절연	
채널수	1	
전기적 사양	최대 19200bps 지원	
Smart I/O-II 최대 허용개수	1	
Smart I/O-III 최대 허용개수	1	

3. Smart I/O-Block Series 커넥터 사양 안내

	Block 방향	Block 맞은편
제품사진		
위치번호/명칭	BMH250	BMW250
Part Number	BMH250-08	BMW250-08
Vendor	YEONHO [YEONHOELECTRONICS]	

4. Input Block

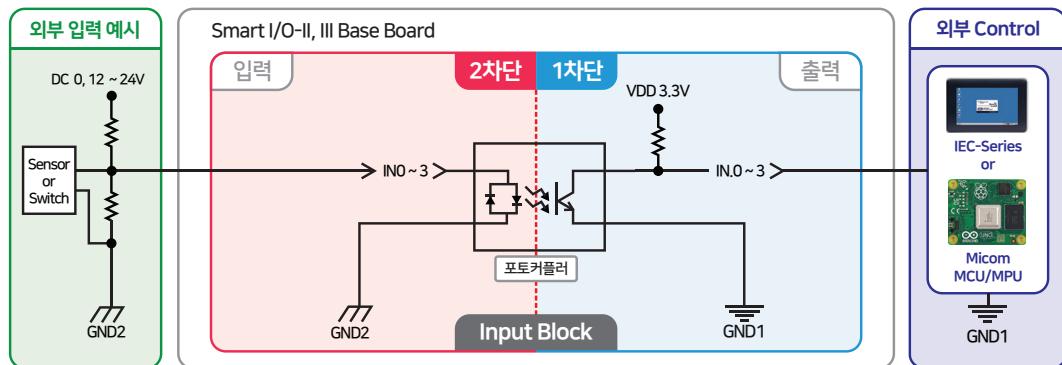
4-1. Input Block 소개

Input Block 1개로 4채널의 입력을 처리할 수 있으며 극성 없이 입력이 가능합니다. High Logic은 DC 12 ~ 24V를 사용할 수 있습니다. Input Block의 경우 SmartGPIO와 연동하여 사용할 수 있습니다.

참조

Input Block 사용 시 입력 전압 범위를 5 ~ 12V로 사용하는 방법 안내

Input Block의 입력 전압 범위를 5~12V로 사용하는 방법은 “[홈페이지(www.hnsts.co.kr)] – [자료실] – [Tech Note] – [25. InputBlock의 입력 전압을 5~12V로 변경하여 사용하는 방법]” 내용을 참조하시기 바랍니다.


주의

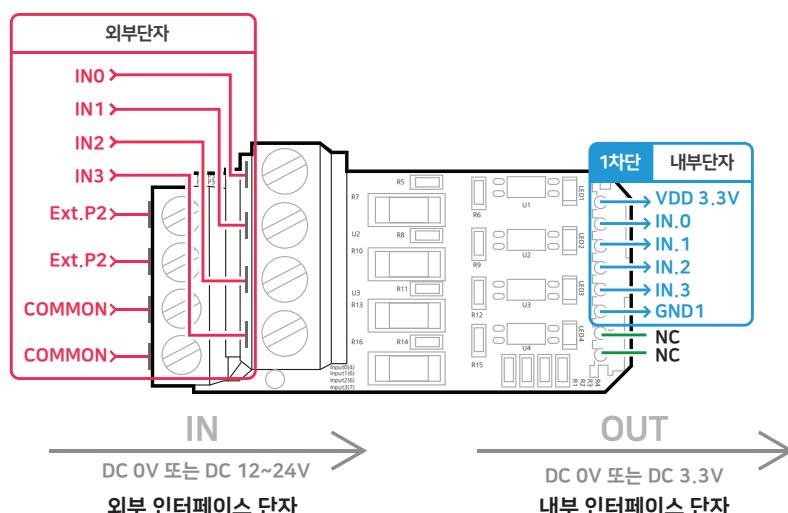
1. 포토 커플러 입력은 무극성이며 COMMON 내부는 공통으로 묶여 있습니다.
2. 전원은 DC 12 ~ 24V 까지 사용 가능하며, COMMON / IN0 ~ 3 전원은 (+/-) 상관없이 반대로만 연결하면 됩니다.

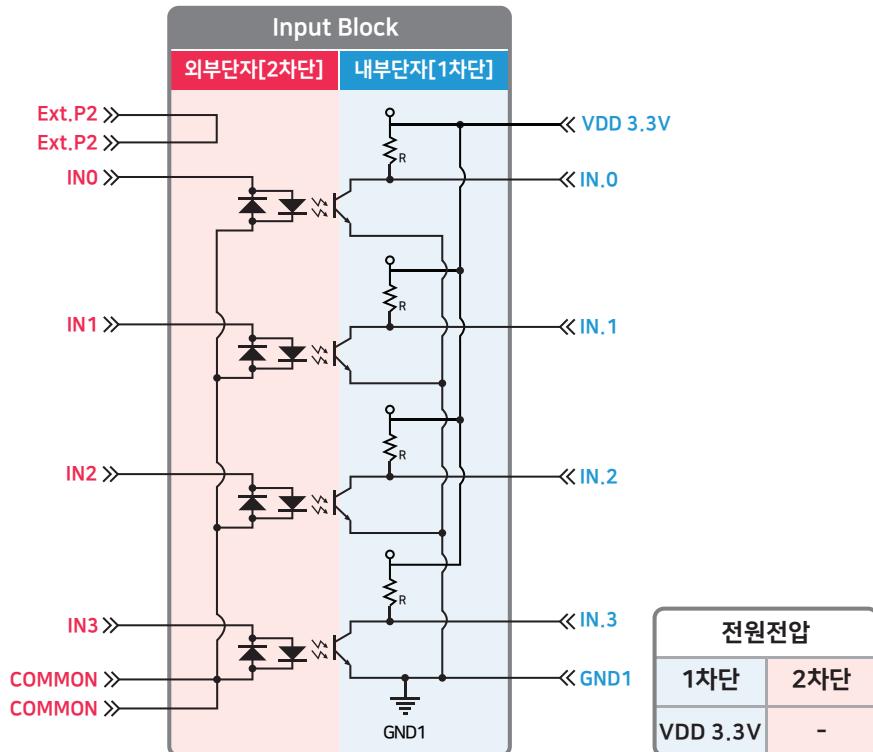
4-2. Input Block 연결단자 설명

외부단자(2차단, INx)에 DC 12 ~ 24V 입력 시 내부단자(1차단, IN.x)에 DC 0V를 출력하고 DC 0V 입력 시 DC 3.3V를 출력합니다.

참고

IEC-Series와 Smart I/O-Series를 연동하여 사용할 경우 내부단자(1차단) 정보는 숙지할 필요 없습니다. Base Board의 제작 및 별도의 I/O 제어 Board를 제작할 경우에 참고하시기 바랍니다.



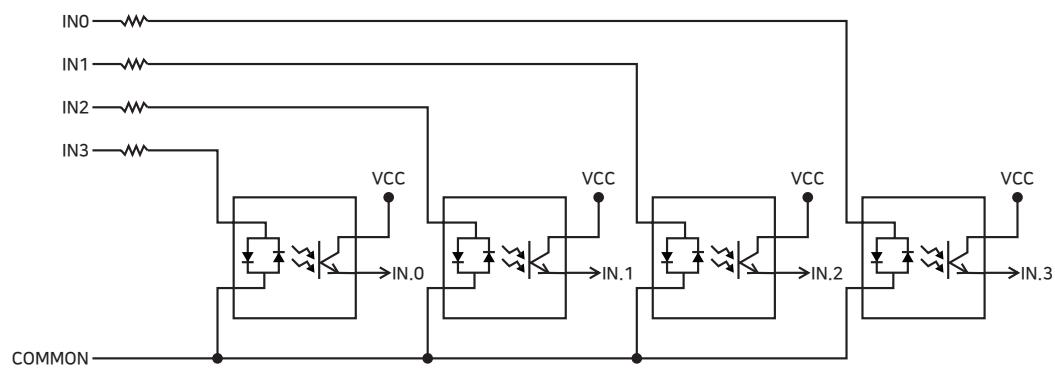


※ 외부 인터페이스 단자

핀 명칭	설명	전기적 사양
Ext.P2	공핀(Ext.P2는 서로 묶임)	-
COMMON	입력 INx에 대한 공통 Ground 단자	GND
IN0		
IN1	DC 입력	DC 0V or 12V@6mA
IN2		DC 0V or 24V@12mA
IN3		

※ COMMON

4채널(IN0 ~ IN3)중 한쪽 단자는 공통으로 묶여 있다.



※ 내부 인터페이스 단자

핀 명칭	설명	전기적 사양
VDD 3.3V	전원 입력 공급(1차단)	DC 3.3V@4mA
IN.0		
IN.1		
IN.2	외부 입력에 따른 출력 신호	DC 0V or 3.3V@1mA
IN.3		
GND1	VDD 3.3V에 대한 GROUND (내부단자[1차단]의 GND1)	GND
NC	-	-

※ SmartGPIO와 연동 시 로직 테이블

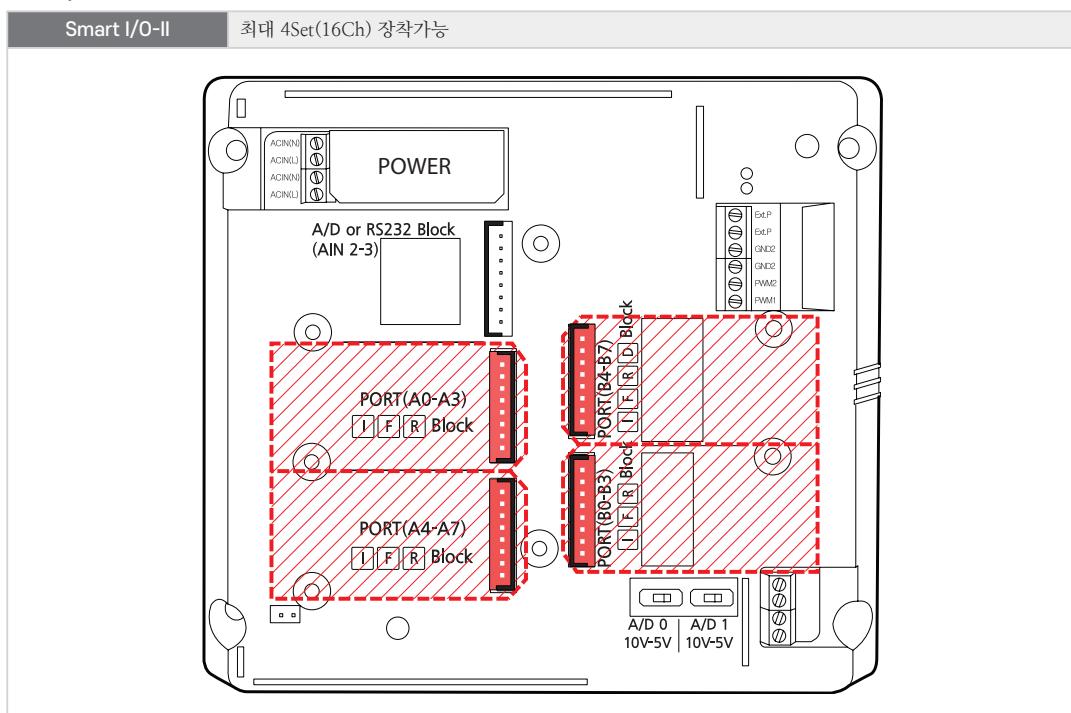
Direction	외부 인터페이스 입력 값	IN.x 출력 신호	Port Data 값(High/Low)
입력	High(12~24V)	0V	Low(0)
	Low(0V)	3.3V	High(1)

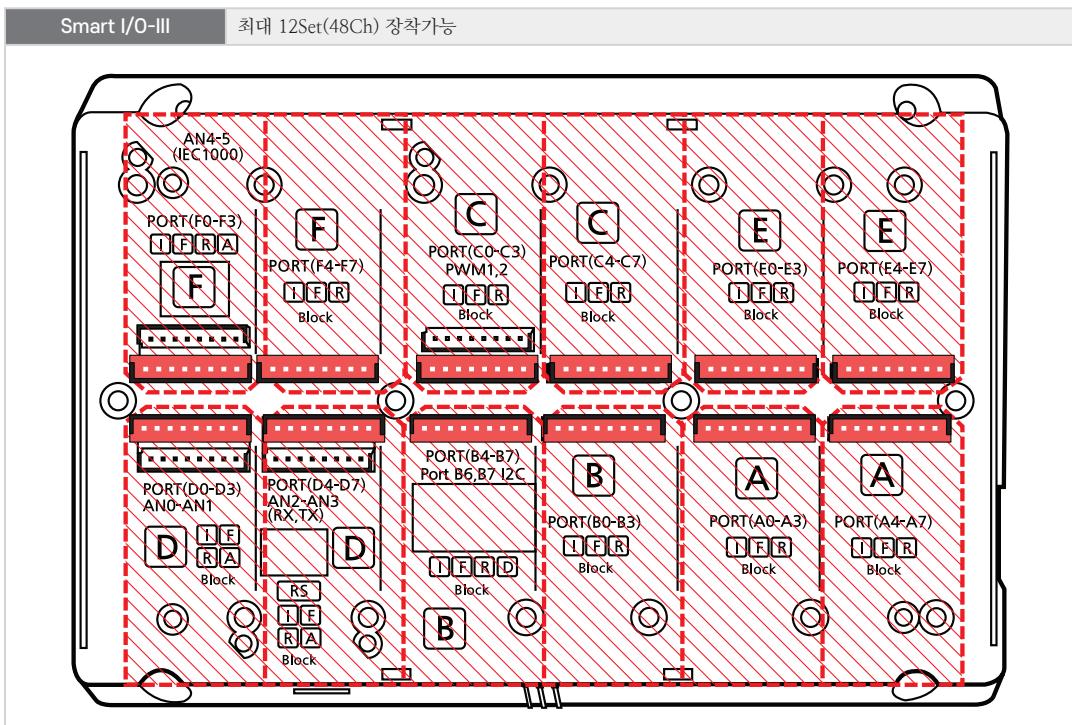
참조

해당 로직 테이블은 SmartX Framework의 SmartGPIO 컴포넌트와 연동하여 Input Block을 사용하였을 때를 기준으로 작성된 로직 테이블입니다.

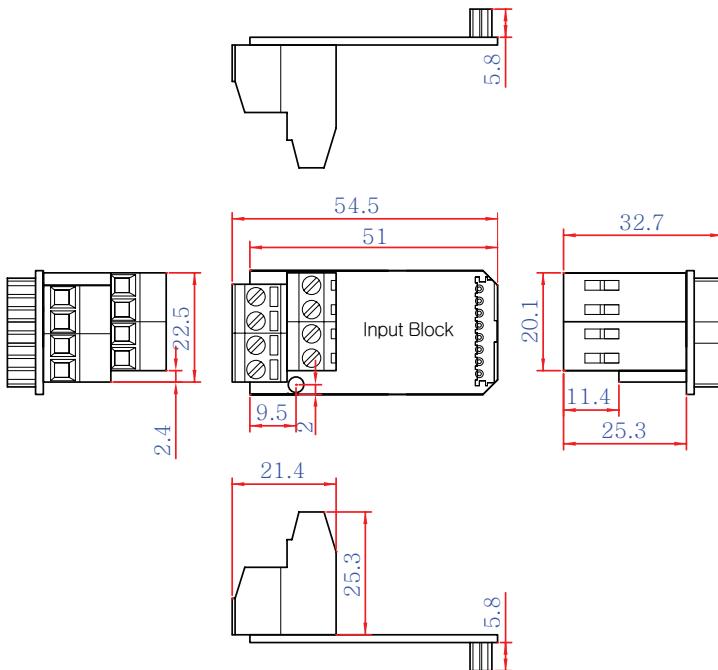
SmartGPIO의 자세한 사항은 “[SmartX 홈페이지](http://www.smartx.co.kr)” → [Reference Guide] → [Part 3. 하드웨어 장치 제어 컴포넌트] → [2. SmartGPIO]” 내용을 참조하시기 바랍니다.

4-3. Input Block 장착위치





4-4. Input Block 외형 및 치수



참조

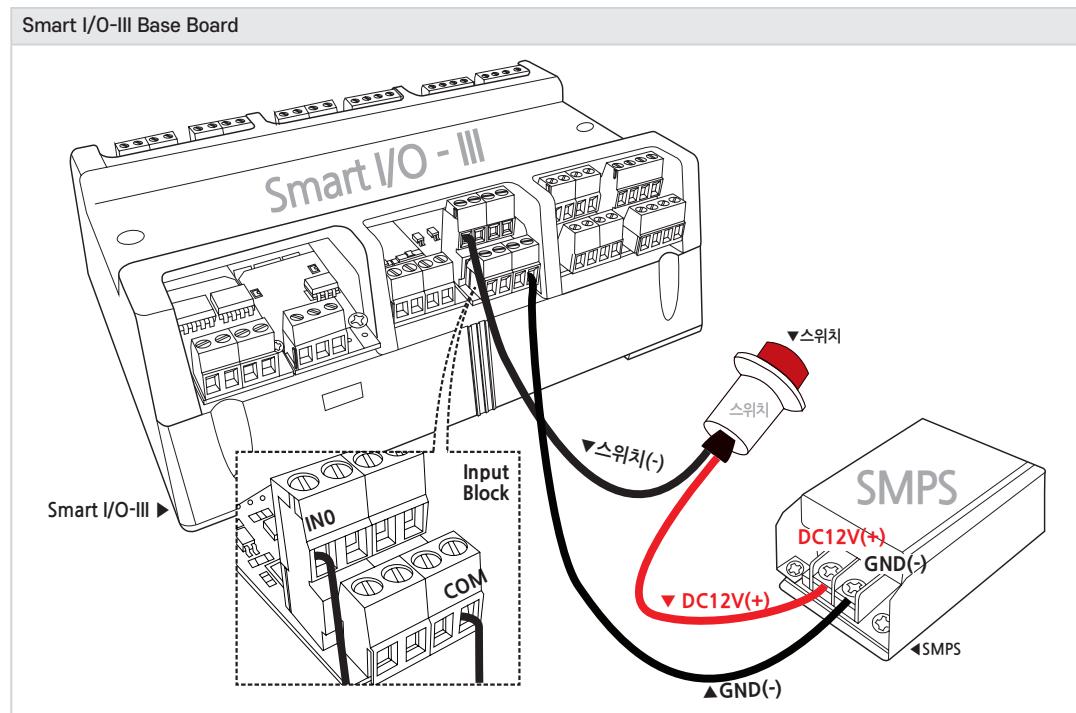
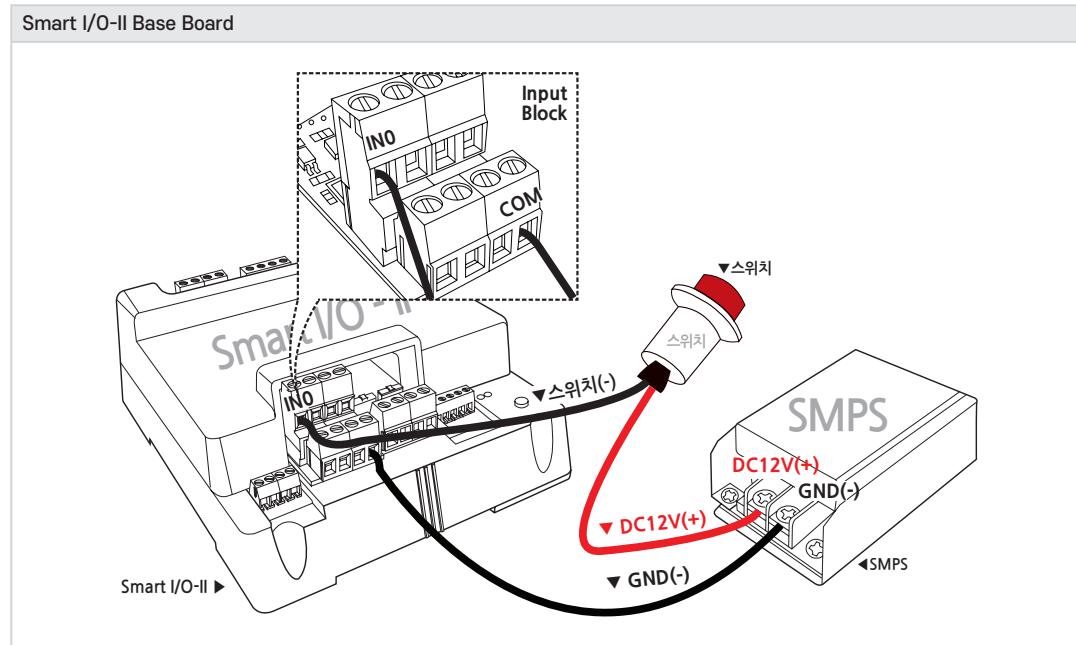
본 제품의 도면은 “[홈페이지](www.hnsts.co.kr)” – [자료실] – [도면 및 승인원]”에서 다운로드할 수 있습니다.

4-5. Input Block 등용 방법(외부 인터페이스 결선)

1) 스위치 연결방법

아래와 같이 스위치를 결선하시기 바랍니다.

< Smart I/O-II, III Base Board 결선도 예시 >

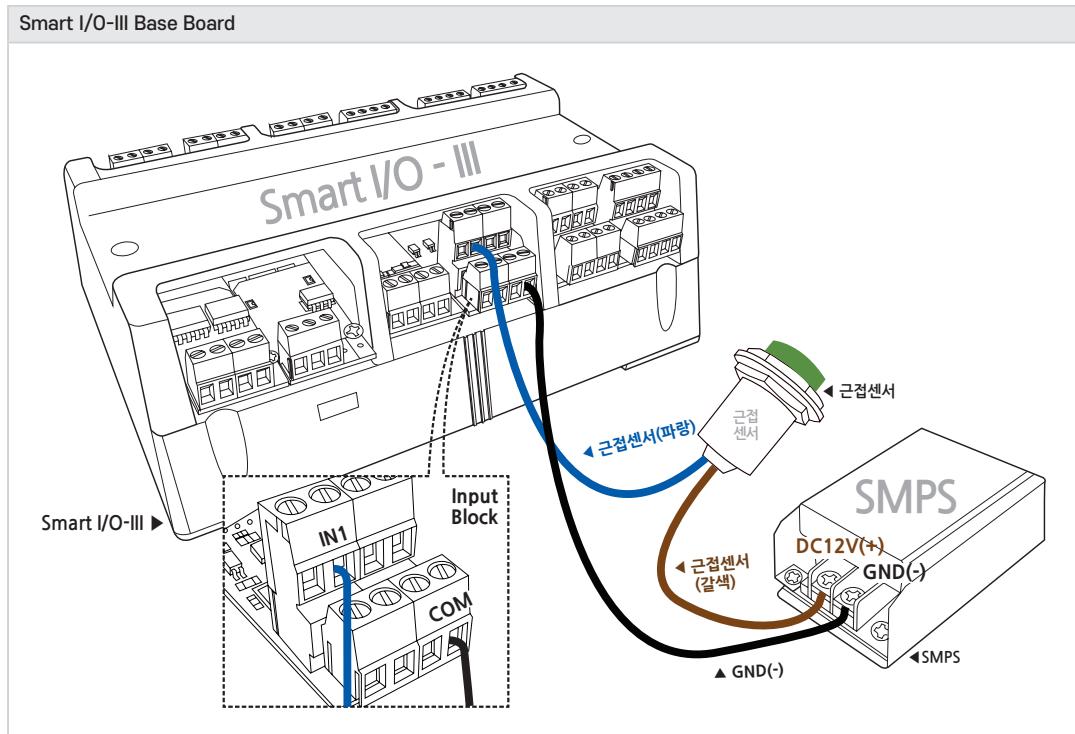
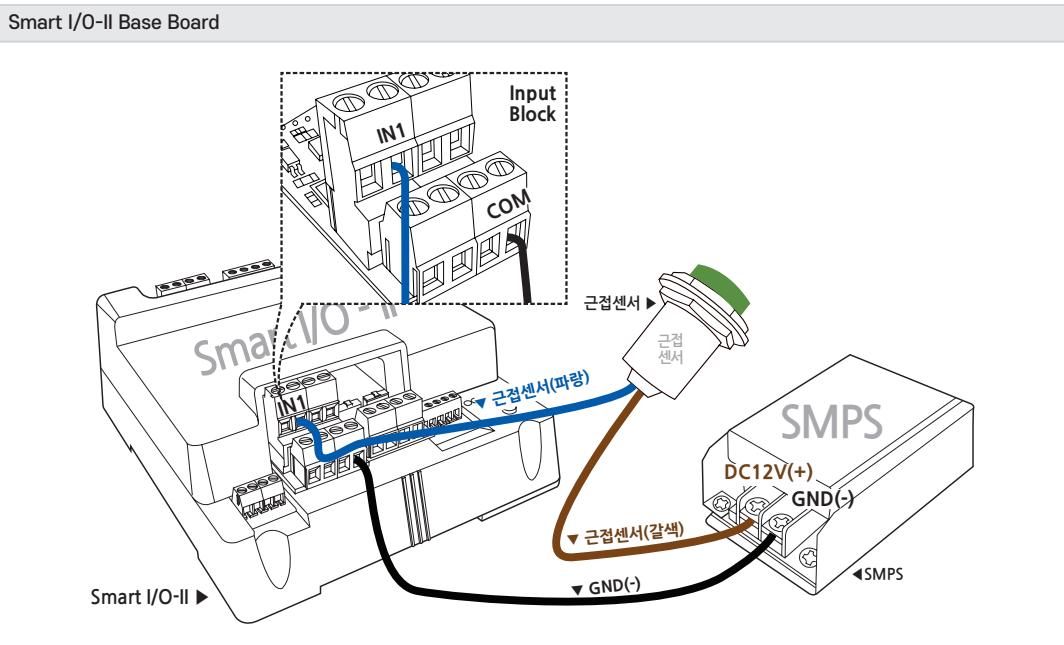


2) 2선식 센서 연결방법(-COM)

아래와 같이 결선을 하면 되지만 센서 제품에 따라 레이터 시트를 참고하여 결선하시기 바랍니다.

- 사용 센서 : 오토닉스(PRT18-8DO)

< Smart I/O-II, III Base Board 결선도 예시 >

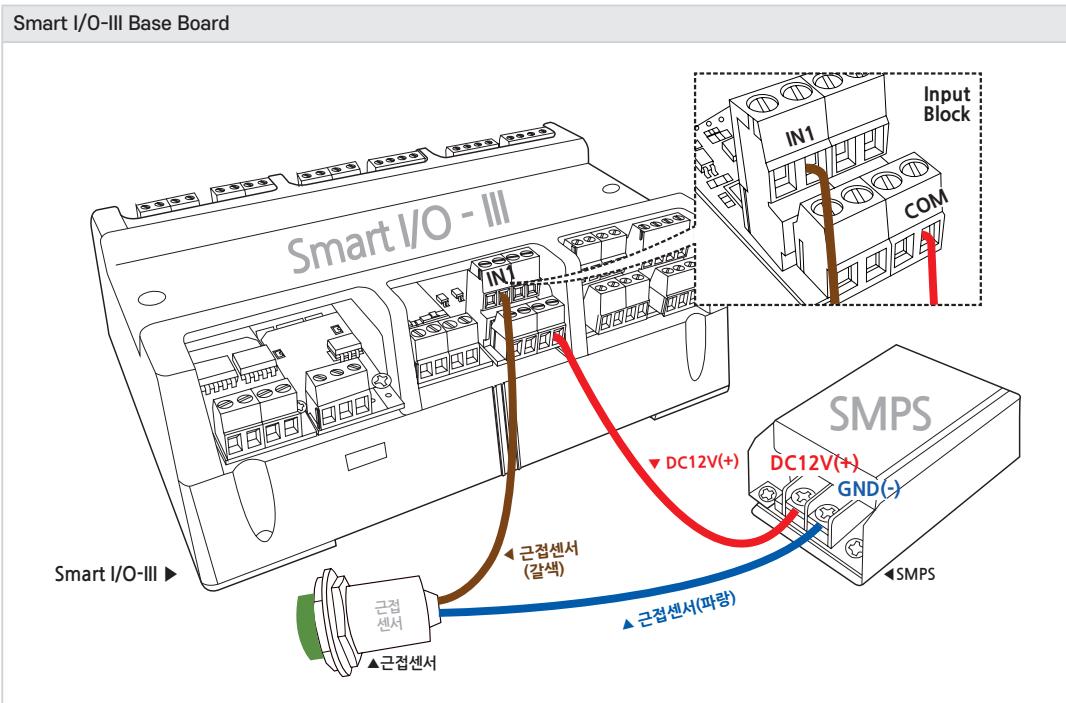
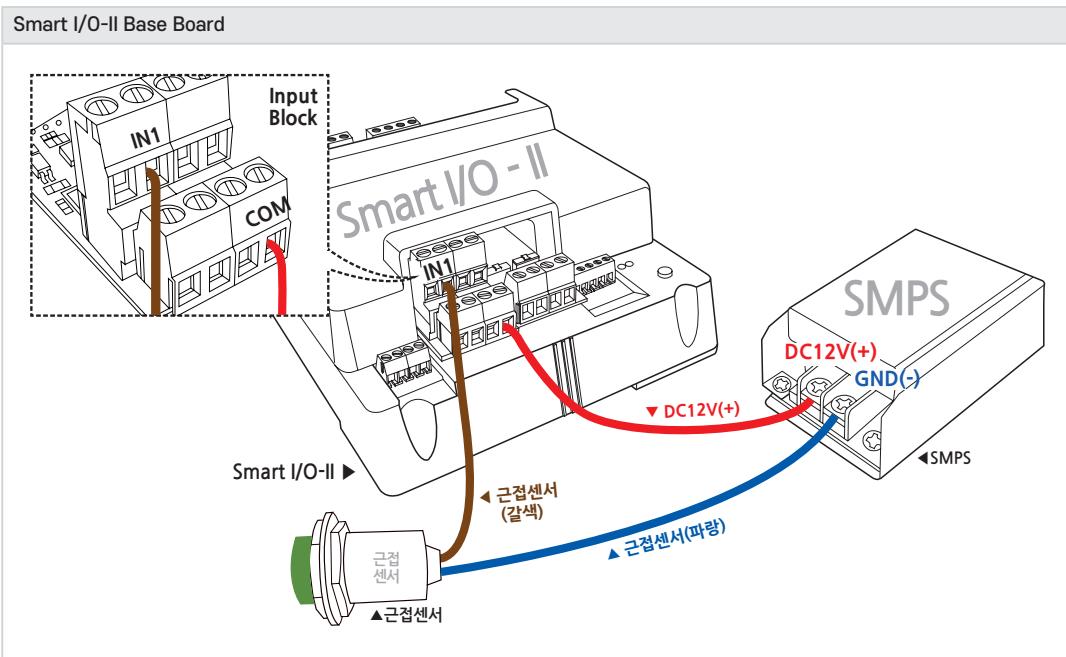


3) 2선식 센서 연결방법(+COM)

아래와 같이 결선을 하면 되지만 센서 제품에 따라 데이터 시트를 참고하여 결선하시기 바랍니다.

- 사용 센서 : 오토닉스(PRT18-8DO)

< Smart I/O-II, III Base Board 결선도 예시 >

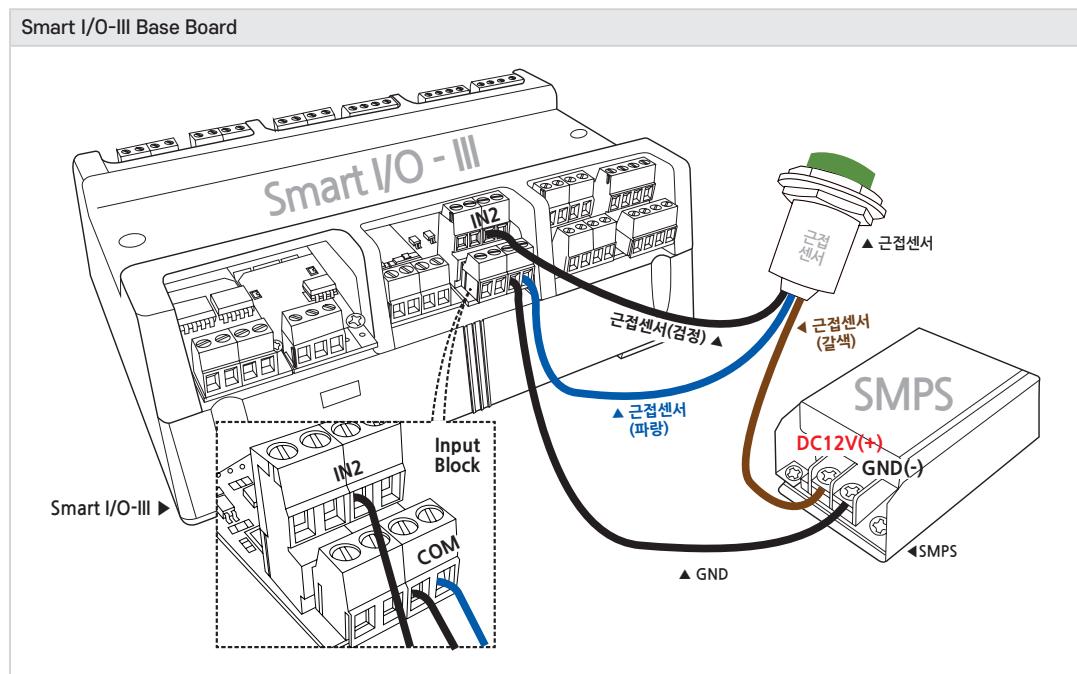
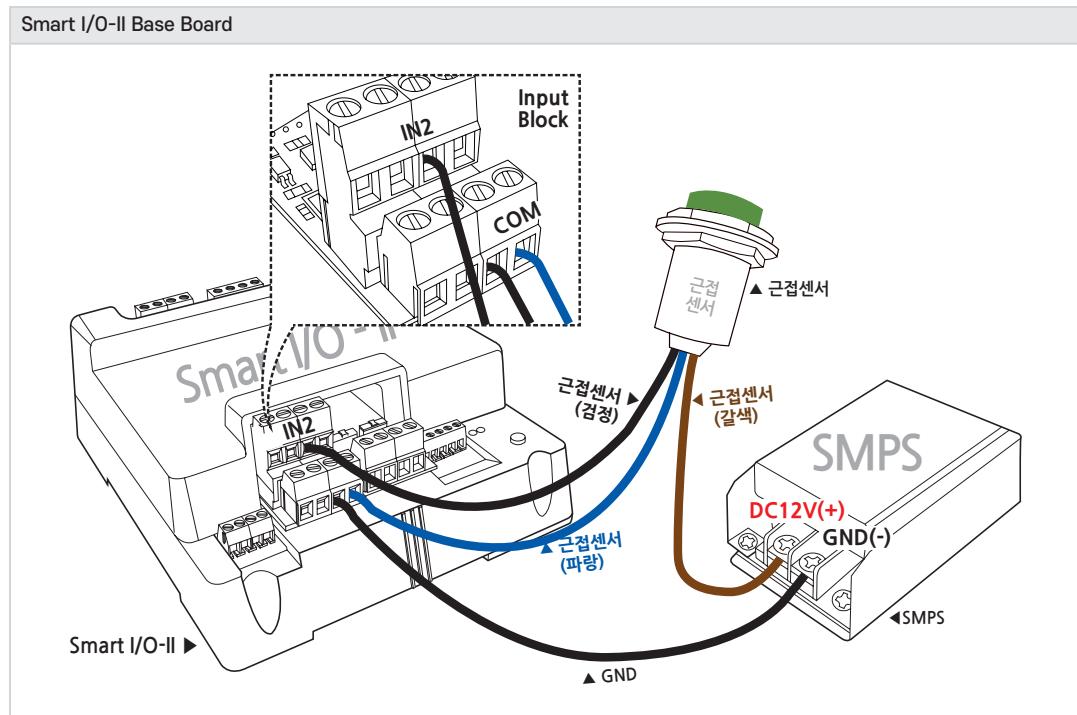


4) 3선식 센서 연결방법(PNP형)

아래와 같이 결선을 하면 되지만 센서 제품에 따라 레이터 시트를 참고하여 결선하시기 바랍니다.

- 사용 센서 : 오토닉스(PRT18-8DP)

< Smart I/O-II, III Base Board 결선도 예시 >

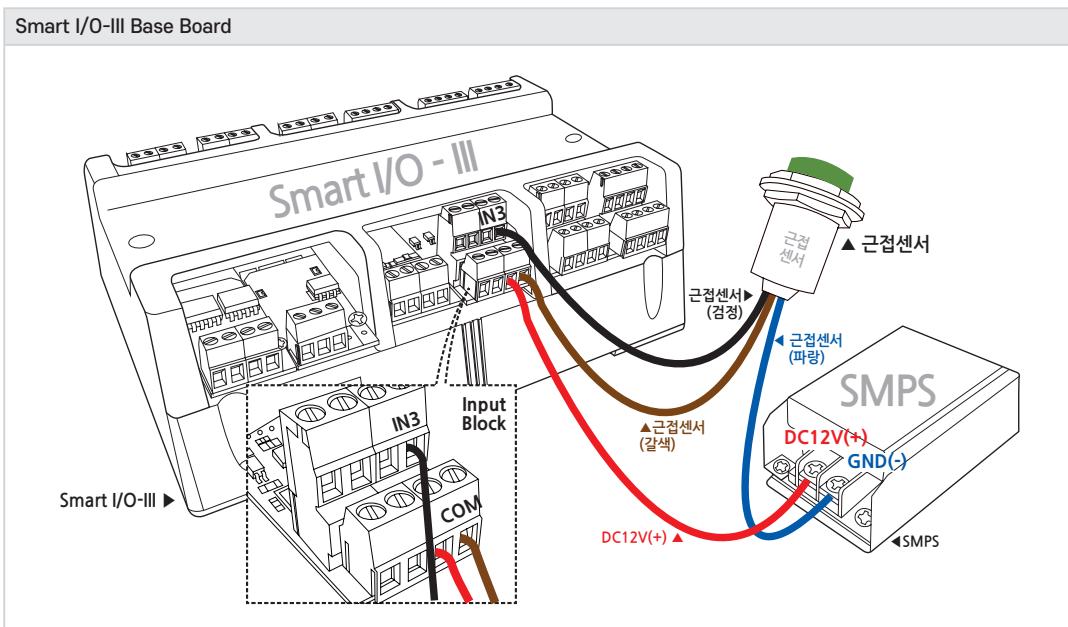
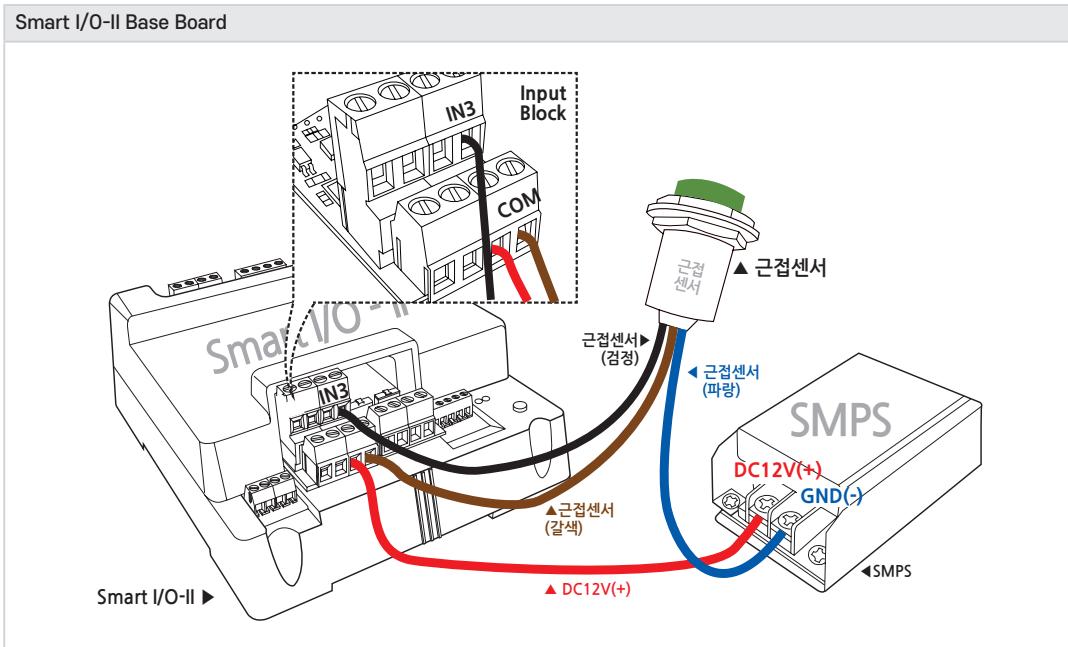


5) 3선식 센서 연결방법(NPN형)

아래와 같이 결선을 하면 되지만 센서 제품에 따라 데이터 시트를 참고하여 결선하시기 바랍니다.

- 사용 센서 : 오토닉스(PRT18-8DN)

< Smart I/O-II, III Base Board 결선도 예시 >



4-6. 언어별 주요 소스코드 및 응용 예제

SmartX Framework	SmartGPIO
소스파일	TechNote53_SmartIO_INPUT
소스경로	[홈페이지](www.hnsts.co.kr) → [자료실] → [53. [C#, VB.NET] Smart I/O – Series Input-Block 사용예제]
기능	스위치, 근접센서 등 입력기능
응용분야	스위치, 외부입력장치
준비사항	<p>[PIN0] 스위치</p> <p>[PIN1] 근접2선식(Normal Open)센서 : (주)오토닉스(PRT18-8DO)</p> <p>[PIN2] 근접3선식(PNP)센서 : (주)오토닉스(PRT18-8DP)</p> <p>[PIN3] 근접3선식(NPN)센서 : (주)오토닉스(PRT18-8DN)</p>



[동작설명]

- 프로그램을 시작한 후 외부에서 각각의 포트별로 입력신호(High → Low)가 발생하면 ② 이미지가 변경되며 ③ 입력 카운트가 1단위로 증가합니다.
- 만약 프로그램 실행 도중에 ③의 카운트를 0으로 초기화하고 싶은 경우 ①의 “Initial” 버튼을 클릭하면 초기화 됩니다.

SmartX Framework SmartGPIO와 관련된 자세한 사항은 “[SmartX 홈페이지(www.smartx.co.kr)] → [Reference Guide] → [Part 3. 하드웨어 장치 제어 컴포넌트] → [2. SmartGPIO]” 내용을 참조하시기 바랍니다.

< C# 예제 주요소스 코드 >

C# 예시 코드 Form1_Load 이벤트에서 SmartGPIO 초기화

```
private void Form1_Load(object sender, EventArgs e)
{
    // 현재 Form을 MainForm으로 설정합니다.
    smartForm1.MainForm = this;
    // SmartGPIO의 Port를 B로 설정
    smartGPIO1.PortSelect = SmartX.SmartGPIO.PORTID.PORTB;
    // PortB의 상태를 ALL Low로 설정
    smartGPIO1.PortDatas = 0;
    // PortB의 방향을 ALL Input으로 설정
    smartGPIO1.PortDirs = 0;
    // PortB의 상태 변화를 체크하는 이벤트 시작
    smartGPIO1.PortWatchStart();
}
```

C# 예시 코드

PortB의 상태가 변경되면 OnPortDatasChange 이벤트에서 입력 처리

```
// PortB의 상태 변경이 감지되었을 때 발생되는 이벤트
private void smartGPIO1_OnPortDatasChange(int iPortDatas)
{
    SmartX.SmartGPIO.PORTPIN iBit;
    iBit = smartGPIO1.PortDetection(SmartX.SmartGPIO.PORTID.PORTB, SmartX.SmartGPIO.TRIGGERMODE.LowActive,
    iPortDatas);

    switch (iBit)
    {
        case SmartX.SmartGPIO.PORTPIN.PIN0:
            labCount0.Text = (++_iInputCount[0]).ToString();
            smartPictureBox1.DrawNext(1);
            break;
        case SmartX.SmartGPIO.PORTPIN.PIN1:
            labCount1.Text = (++_iInputCount[1]).ToString();
            smartPictureBox2.DrawNext(1);
            break;
        case SmartX.SmartGPIO.PORTPIN.PIN2:
            labCount2.Text = (++_iInputCount[2]).ToString();
            smartPictureBox3.DrawNext(1);
            break;
        case SmartX.SmartGPIO.PORTPIN.PIN3:
            labCount3.Text = (++_iInputCount[3]).ToString();
            smartPictureBox4.DrawNext(1);
            break;
        case SmartX.SmartGPIO.PORTPIN.ZERO:
            smartPictureBox1.DrawNext(0);
            smartPictureBox2.DrawNext(0);
            smartPictureBox3.DrawNext(0);
            smartPictureBox4.DrawNext(0);
            break;
    }
}
```

<VB 예제 주요소스 코드 >

VB 예시 코드

Form1_Load 이벤트에서 SmartGPIO 초기화

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As EventArgs) Handles MyBase.Load
    ' 현재 Form을 MainForm으로 설정합니다.
    SmartForm1.MainForm = Me
    ' SmartGPIO의 Port를 B로 설정
    SmartGPIO1.PortSelect = SmartX.SmartGPIO.PORTID.PORTB
    ' PortB의 방향을 ALL Input으로 설정
    SmartGPIO1.PortDirs = 0
    ' PortB의 상태를 ALL Low로 설정
    SmartGPIO1.PortDatas = 0
    ' PortB의 상태 변경을 체크하는 이벤트 시작
    SmartGPIO1.PortWatchStart()
End Sub
```

VB 예시 코드

PortB의 상태가 변경되면 OnPortDataChange 이벤트에서 입력 처리

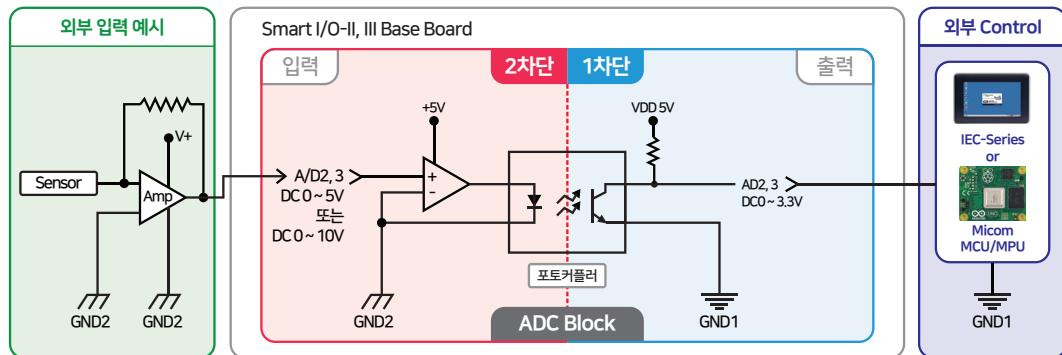
```
' PortB의 상태 변경이 감지되었을 때 발생되는 이벤트
Private Sub smartGPIO1_OnPortDataChange(ByVal iPortData As Integer) Handles SmartGPIO1.OnPortDataChange
    Dim iBit As SmartX.SmartGPIO.PORTPIN
    iBit = SmartGPIO1.PortDetection(SmartX.SmartGPIO.PORTID.PORTB, SmartX.SmartGPIO.TRIGGERMODE.HighActive, iPortData)

    Select Case iBit
        Case SmartX.SmartGPIO.PORTPIN.PIN0
            m_iInputCount(0) += 1
            labCount0.Text = m_iInputCount(0).ToString()
            smartPictureBox1.DrawNext(1)
        Case SmartX.SmartGPIO.PORTPIN.PIN1
            m_iInputCount(1) += 1
            labCount1.Text = m_iInputCount(1).ToString()
            smartPictureBox2.DrawNext(1)
        Case SmartX.SmartGPIO.PORTPIN.PIN2
            m_iInputCount(2) += 1
            labCount2.Text = m_iInputCount(2).ToString()
            smartPictureBox3.DrawNext(1)
        Case SmartX.SmartGPIO.PORTPIN.PIN3
            m_iInputCount(3) += 1
            labCount3.Text = m_iInputCount(3).ToString()
            smartPictureBox4.DrawNext(1)
        Case SmartX.SmartGPIO.PORTPIN.ZERO
            smartPictureBox1.DrawNext(0)
            smartPictureBox2.DrawNext(0)
            smartPictureBox3.DrawNext(0)
            smartPictureBox4.DrawNext(0)
    End Select
End Sub
```

5. ADC Block

5-1. ADC Block 소개

ADC Input은 Analog OptoCoupler를 사용한 것이 특징입니다. 따라서 입력측(ADC Input)과 IEC-Series의 AIN은 서로 전원이 분리되어 있습니다. A/D 입력전압은 DC 0 ~ 5V 또는 DC 0 ~ 10V를 선택하여 사용할 수 있습니다. IEC-Series와 연동하여 12Bit(4096)의 해상도를 지원합니다. 또한 IEC667-Series의 경우 ADC 4채널, IEC1000 非Lite-Series의 경우 ADC 6채널을 지원합니다. ADC Block의 경우 SmartADC와 연동하여 사용할 수 있습니다.



주의

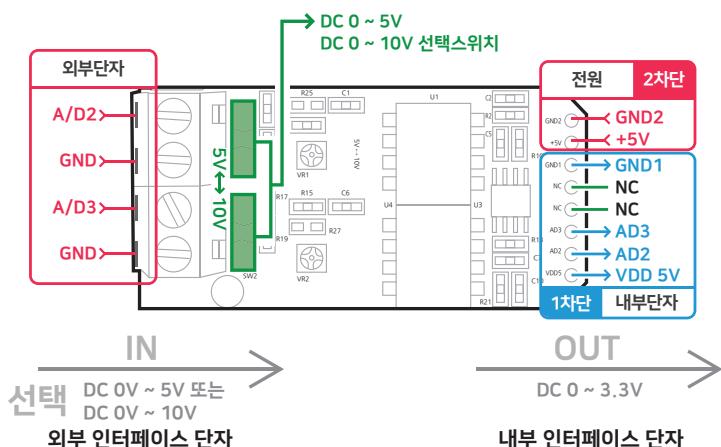
- 외부입력 전원(멀티 템) 사용 시 접지가 있는 것을 사용해야 합니다. 접지가 없는 경우 ADC의 입력 값이 많이 흔들릴 수 있습니다. Extension-II 케이블의 길이를 연장하는 경우 노이즈의 영향으로 신호가 왜곡될 수 있습니다.
- 非Lite-Series에서 ADC Block, DAC Block 적용시 제품의 전원을 12V 이하로 사용하는 것을 권장합니다. 만약 12V이상 사용시 ADC Block, DAC Block의 전원에서 발열이 발생할 수 있습니다.

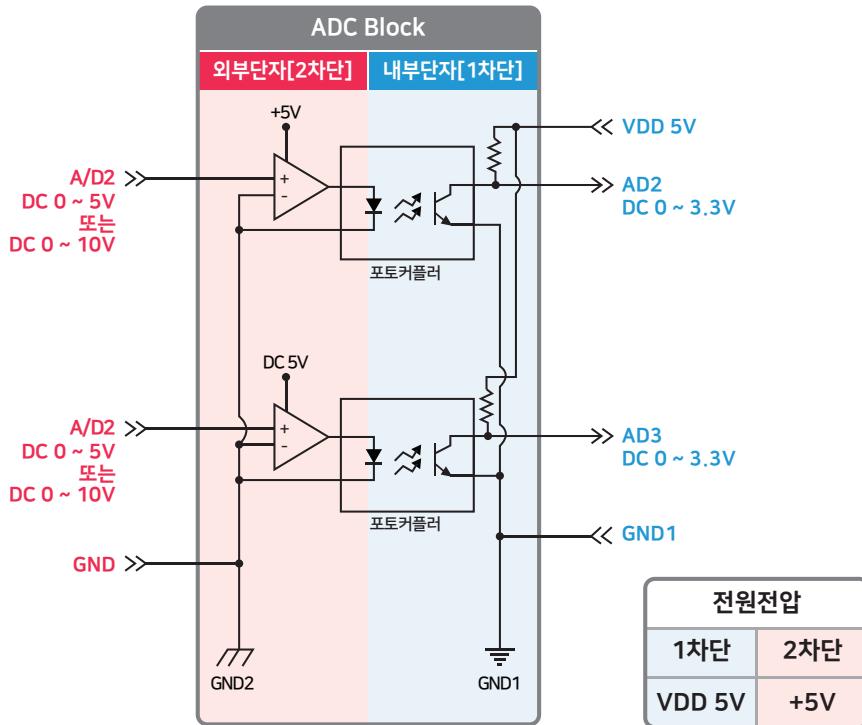
5-2. ADC Block 연결단자 설명

외부단자(2차단, A/Dx)에 DC 0 ~ 5V 또는 DC 0 ~ 10V 입력 시 내부단자(1차단, ADx)에 DC 0 ~ 3.3V로 변환하여 출력합니다.

참고

IEC-Series와 Smart I/O-Series를 연동하여 사용할 경우 내부단자(1차단) 정보는 숙지할 필요 없습니다. Base Board의 제작 및 별도의 I/O 제어 Board를 제작할 경우에 참고하시기 바랍니다.





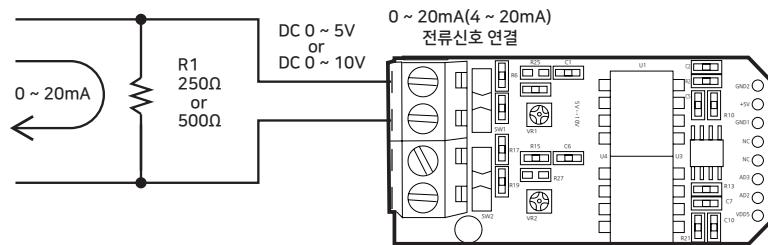
※ 외부 인터페이스 단자

핀 명칭	설명	전기적 사양
A/D2	ADC 입력 신호	DC 0 ~ 5V or DC 0 ~ 10V
GND	A/Dx에 대한 GROUND (외부단자[2차단]의 GND2)	GND
A/D3	ADC 입력 신호	DC 0 ~ 5V or DC 0 ~ 10V
GND	A/Dx에 대한 GROUND (외부단자[2차단]의 GND2)	GND

※ 내부 인터페이스 단자

핀 명칭	설명	전기적 사양
+5V	전원 입력 공급(2차단)	DC 5V@40mA
GND2	+5V에 대한 GROUND (전원[2차단]의 GND2)	GND
VDD 5V	전원 입력 공급(1차단)	DC 5V@3mA
GND1	VDD 5V에 대한 GROUND (내부단자[1차단]의 GND1)	GND
NC	-	-
AD3	ADC 출력 신호	DC 0 ~ 3.3V
AD2		

5-3. ADC Block 입력 신호를 전압에서 전류로 변환하여 입력받는 방법



R_1 저항 값	출력전압 S/W 범위	계산값
$R_1 = 250\Omega$	DC 0 ~ 5V	$V = I \times R = 20mA \times 250\Omega = DC 5V$
$R_1 = 500\Omega$	DC 0 ~ 10V	$V = I \times R = 20mA \times 500\Omega = DC 10V$

R_1 을 별도로 결선하여 전류 신호를 전압 신호로 변환하여 사용합니다.

만약 20mA보다 높은 전류를 입력해야 한다면 R_1 의 저항 값을 변환하여 사용합니다.

[예시]

50mA 전류를 입력 시 $R = V / I$ 공식을 이용하여 DC 0 ~ 5V 사용 시 $R = 5V / 50mA = 100\Omega$, DC 0 ~ 10V 사용 시 $R = 10V / 50mA = 200\Omega$ 의 저항을 별도로 결선하여 사용합니다.

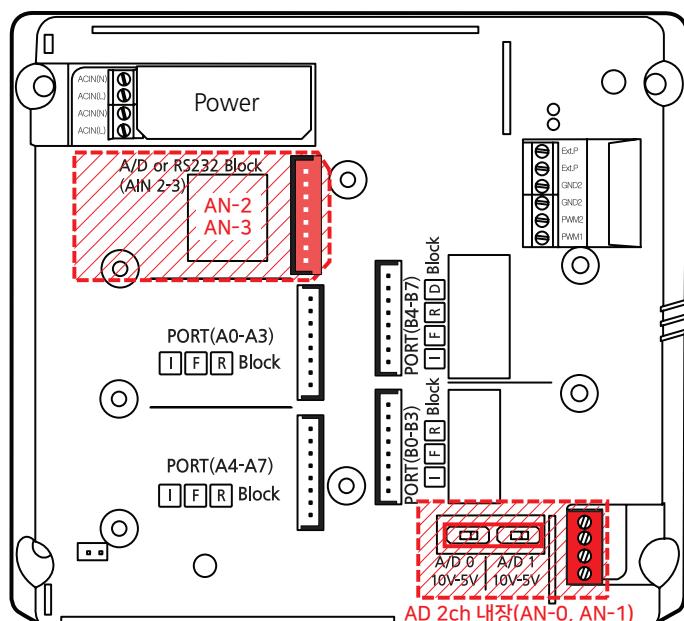
주의

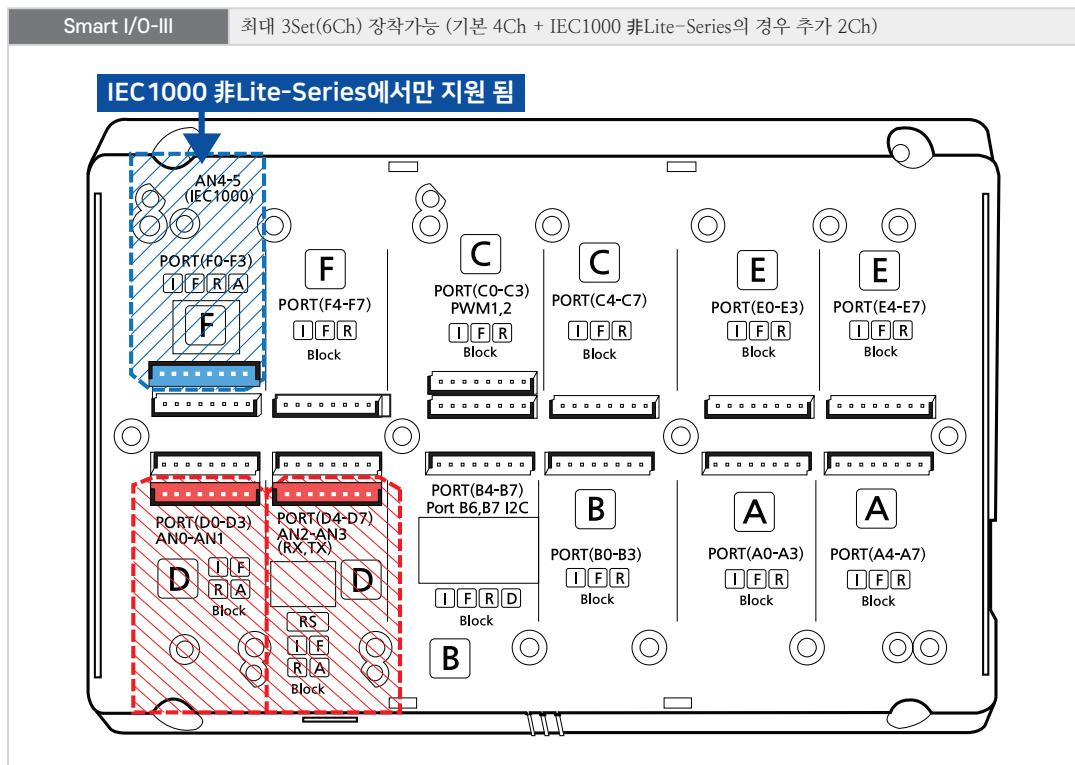
입력 전류에 따라 저항 값의 설정이 잘못될 경우 허용되는 전압 범위보다 높은 전압이 인가되어 제품에 고장의 원인이 될 수 있습니다.

5-4. ADC Block 장착위치

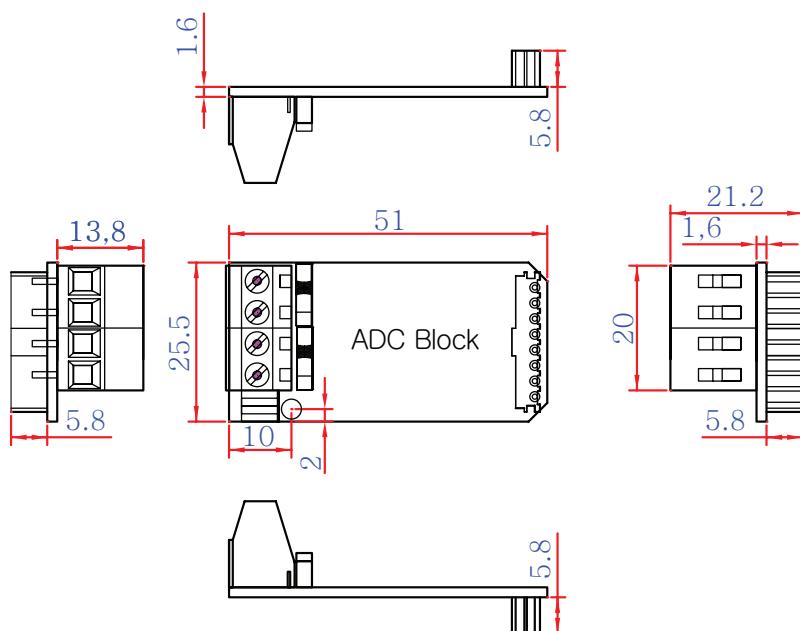
Smart I/O-II

최대 1Set(2Ch) 장착가능 / Base Board 2Ch 내장





5-5. ADC Block 외형 및 치수



참조

본 제품의 도면은 “[홈페이지](www.hnsts.co.kr)” – [자료실] – [도면 및 승인원]”에서 다운로드할 수 있습니다.

5-6. ADC Block 응용 방법(외부 인터페이스 결선)

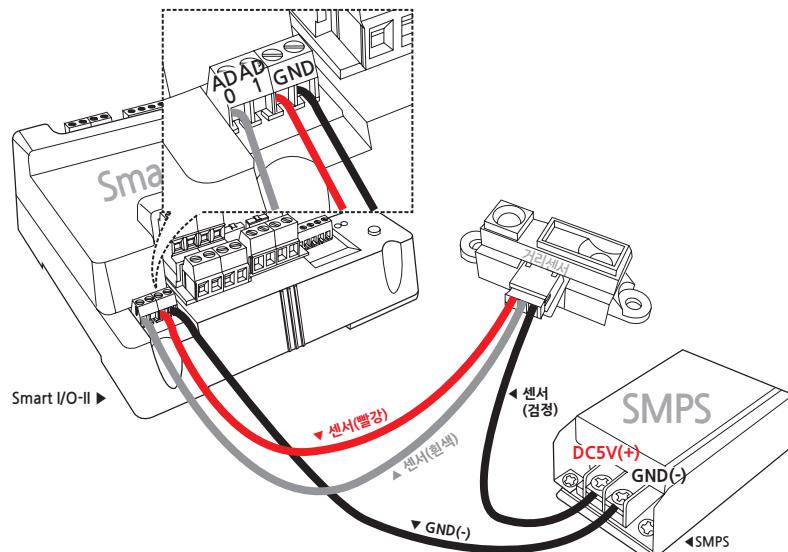
1) 거리센서 연결방법

아래와 같이 결선을 하면 되지만 센서 제품에 따라 데이터 시트를 참고하여 결선하시기 바랍니다.

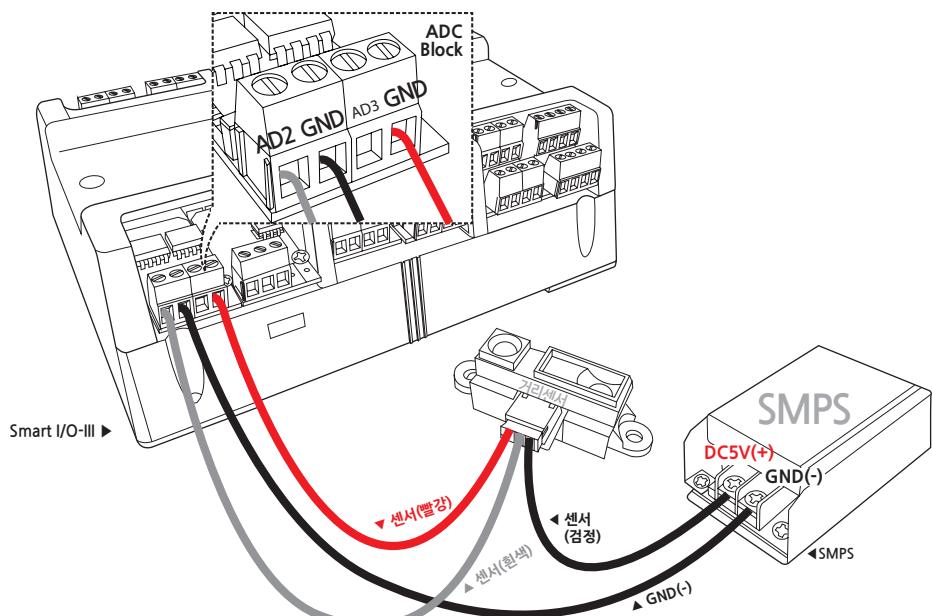
- 사용 센서 : SHARP(GP2Y0A41SK0F)

< Smart I/O-II, III Base Board 결선도 예시 >

Smart I/O-II Base Board



Smart I/O-III Base Board

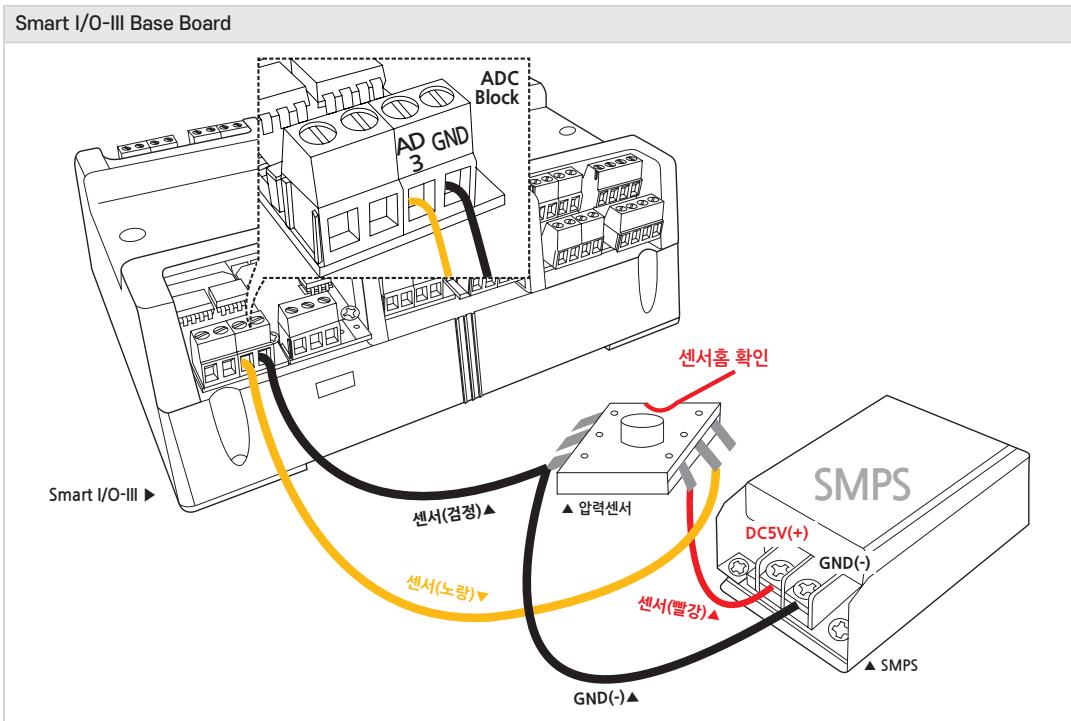
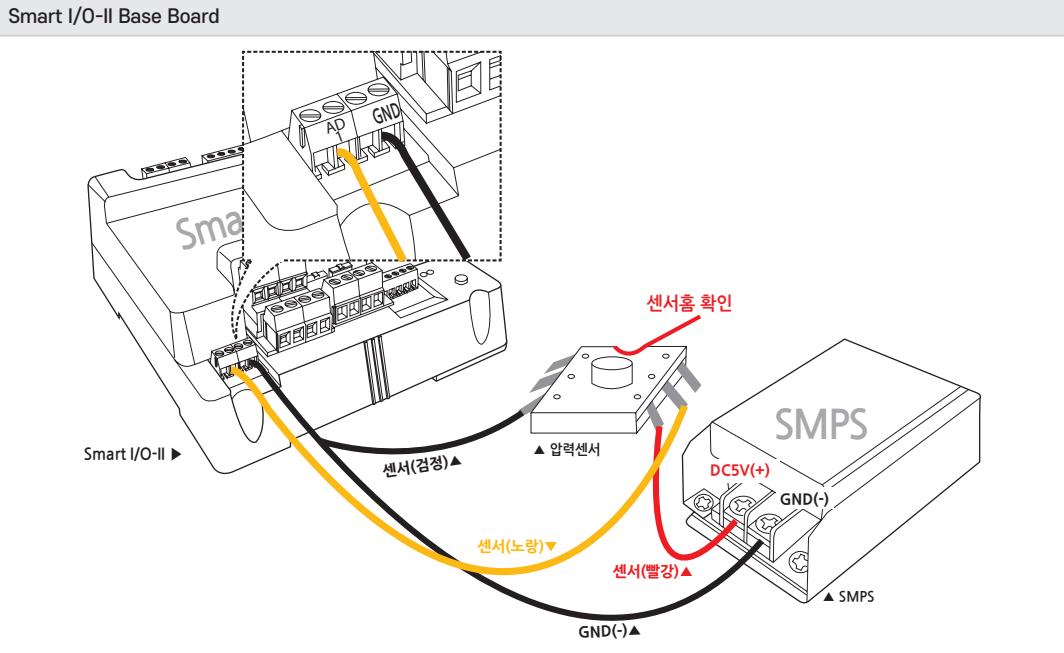


2) 압력센서 연결방법

아래와 같이 결선을 하면 되지만 센서 제품에 따라 레이터 시트를 참고하여 결선하시기 바랍니다.

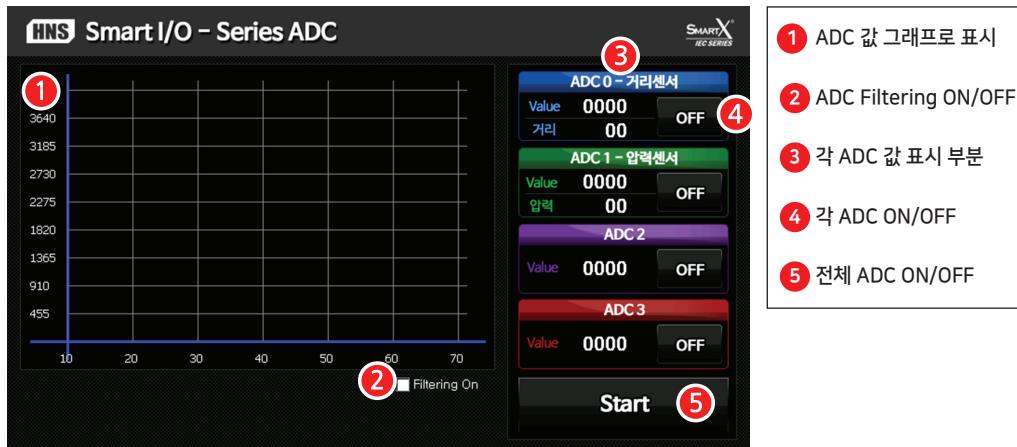
- 사용 센서 : Smate(33A-030G-2210)

< Smart I/O-II, III Base Board 결선도 예시 >



5-7. 언어별 주요 소스코드 및 응용 예제

SmartX Framework	SmartADC
소스파일	TechNote52_SmartIO_ADC
소스경로	[홈페이지](www.hnsts.co.kr) → [자료실] → [Tech Note] → [52. [C#, VB.NET] Smart I/O - I, II, III_ADC 예제]
기능	온습도, 근접, 압력, 가속도, 자이로센서 등 Analog 센서 값 처리
응용분야	센서사용 측정분야
준비사항	[ADC 0] 거리센서 : SHARP(GP2Y0A41SK0F) [ADC 1] 압력센서 : Smate(33A-030G-2210)



[동작설명]

- ② Filtering On CheckBox를 Check하면 ReadDataDetailFiltering() 메소드를 사용하고 일정 범위 이외의 데이터 값을 무시한 데이터의 평균을 구합니다.
- ⑤ Start 버튼을 클릭하면 SmartTimer를 Start합니다. SmartTimer_Tick 이벤트에서는 ADC 0채널 ~ 3채널 까지 를 읽는 코드가 실행됩니다.
- ④ OFF 버튼을 클릭하면 해당 ADC 채널의 smartADC1.ReadData() 혹은 smartADC1.ReadDataDetailFiltering() 코드가 활성화(Active) 됩니다. OFF 버튼은 해당 채널 별로 존재하며 각 버튼의 ON/OFF는 해당 채널만 활성화 / 비활성화 시킵니다. 즉 ADC0과 ADC1을 ON하는 경우 ADC0과 ADC1채널에서 값을 읽어옵니다. ADC2와 ADC3은 OFF되어 값을 읽지 않습니다.
- ③ Start 버튼을 클릭한 후 ④ OFF 버튼을 클릭하여 해당 채널을 ON하여 아날로그 데이터가 입력되는 경우 ① Value 값에 디지털(Digital) 값으로 표시되며, 해당 디지털 값은 SmartDraw의 Chart 기능을 사용하여 ① 차트를 그려줍니다.

SmartX Framework

SmartADC와 관련된 자세한 사항은 “[SmartX 홈페이지](www.smartx.co.kr) → [Reference Guide] → [Part 3. 하드웨어 장치 제어 컴포넌트] → [7. SmartADC]” 내용을 참조하시기 바랍니다.

< C# 예제 주요소스 코드 >

참고

C#의 주요 소스 코드의 일부 코드가 생략되어 있습니다. 상세한 코드를 확인하시기 위해서는 상단 표에 나와있는 “소스경로”를 참고하여 예제 프로그램을 참고하시기 바랍니다.

C# 예시 코드

모든 채널(4Ch)의 ADC Read Start / Stop 설정

```

private void btnADCStart_Click(object sender, EventArgs e)
{
    // …DetailFiltering 선택 관련 코드 중략…

    // ADC Start
    if (btnADCStart.ButtonStatus == SmartX.SmartButton.BUTSTATUS.DOWN)
    {
        smartTimer1.Start();
        chkFiltering.Enabled = false;
    }

    // ADC Stop
    else if (btnADCStart.ButtonStatus == SmartX.SmartButton.BUTSTATUS.UP)
    {
        smartTimer1.Stop();
        chkFiltering.Enabled = true;
    }
}

```

C# 예시 코드

ADC 값을 읽어서 Value 및 Chart에 표시하는 Timer 이벤트

```

// ADC 값 읽어서 표시하는 Timer
private void smartTimer1_Tick(object sender, EventArgs e)
{
    int iVal = 0, iVal1 = 0, iVal2 = 0, iVal3 = 0;
    double iADCVal = 0, iADCVal1 = 0, iADCVal2 = 0, iADCVal3 = 0;

    // ADC 0채널의 Read가 ON일 경우
    if (btnADC0.ButtonStatus == SmartX.SmartButton.BUTSTATUS.DOWN)
    {
        if (chkFiltering.Checked == true)
        {
            // ADC0 데이터를 얻어온다.
            iVal = smartADC1.ReadDataDetailFiltering(0, m_iRawData, m_iAvailable);
        }
        else
        {
            // ADC0 데이터를 얻어온다.
            iVal = smartADC1.ReadData(0);
        }

        // 거리측정계산 함수
        ADC0_DataParsing(iVal);
    }
    else
    {
        labADCValue0.Text = " ";
        labADCDistance0.Text = " ";
    }

    // …ADC 1 ~ 3 채널의 Read관련 코드 중략…

    // ADC 값 표시
    labADCValue0.Text = iVal.ToString();
    labADCValue1.Text = iVal1.ToString();
    labADCValue2.Text = iVal2.ToString();
    labADCValue3.Text = iVal3.ToString();
}

```

```

// 그래프에 4채널의 데이터를 표시하기 위한 계산
iADCVal = ((double)iVal / 4096);
iADCVal1 = ((double)iVal1 / 4096);
iADCVal2 = ((double)iVal2 / 4096);
iADCVal3 = ((double)iVal3 / 4096);

iVal = (int)((double)270.0 * iADCVal);
iVal1 = (int)((double)270.0 * iADCVal1);
iVal2 = (int)((double)270.0 * iADCVal2);
iVal3 = (int)((double)270.0 * iADCVal3);

// 그래프의 데이터를 입력합니다. (인자의 수에 맞추어 그래프의 각각의 계열에 적용됨)
smartDraw1.Chart.PutData(iVal, iVal1, iVal2, iVal3);
}

```

C# 예시 코드

ADC0 센서값을 거리로 계산해주는 메소드(센서의 구간이 선형적일 때 적용)

```

// 센서 특성상 선형적이지 않을 경우 구간별로 측정해야 합니다.
// ADC0 센서값을 거리로 계산해주는 공식. 20cm ~ 50cm 구간 사이 변환 공식
// 본 코드는 AD값으로 거리를 계산하는 공식이며 AD값에 따라 거리값의 증/감이 비교적
// 비례하는 구간을 계산한 공식입니다.
// AD값의 구간별로 정확하게 거리를 계산하기 위해서는 AD구간별로 계산공식이 필요합니다.

private void ADC0_DataParsing(double val)
{
    // 거리센서의 ADC값 = 1064는 거리20cm, ADC값 = 491은 거리50cm입니다.
    // ADC값 1064 - 491 = 573(ADC구간)이고 20cm ~ 50cm(ADC의 거리구간)
    // 계산공식은 573 : 30 = (1064 - val) : x 의 결과값에 20(출발점)을 더합니다.
    // ADC값
    labADCValue0.Text = val.ToString();
    // 거리값
    labADCDistance0.Text = Math.Round(((1064 - val) * 30) / 573 + 20, 0).ToString() + " cm";
}

```

< VB 예제 주요소스 코드 >**참고**

VB.NET 주요 소스 코드에서는 일부 코드가 생략되어 있습니다. 상세한 코드를 확인하기 위해서는 상단 표에 나와있는 “소스경로”에서 예제 프로그램을 참고하시기 바랍니다.

VB 예시 코드

모든 채널(4Ch)의 ADC Read Start / Stop 설정

```

Private Sub btnADCStart_Click(ByVal sender As Object, ByVal e As EventArgs) Handles btnADCStart.Click
    ' ...DetailFiltering 선택 관련 코드 중략...
    If btnADCStart.ButtonStatus = SmartX.SmartButton.BUTSTATUS.DOWN Then
        ' ADC Start
        SmartTimer1.Start()
        chkFiltering.Enabled = False
    ElseIf btnADCStart.ButtonStatus = SmartX.SmartButton.BUTSTATUS.UP Then
        ' ADC Stop
        SmartTimer1.Stop()
        chkFiltering.Enabled = True
    End If
End Sub

```

VB 예시 코드

ADC 값을 읽어서 Value 및 Chart에 표시하는 Timer 이벤트

```

‘ ADC 값 읽어서 표시하는 Timer
Private Sub smartTimer1_Tick(ByVal sender As Object, ByVal e As EventArgs) Handles SmartTimer1.Tick
    Dim iVal As Integer = 0, iVal1 As Integer = 0, iVal2 As Integer = 0, iVal3 As Integer = 0
    Dim iADCVal As Double = 0, iADCVal1 As Double = 0, iADCVal2 As Double = 0, iADCVal3 As Double = 0

    ‘ ADC 0채널의 Read가 ON일 경우
    If btnADC0.ButtonStatus = SmartX.SmartButton.BUTSTATUS.DOWN Then
        If chkFiltering.Checked = True Then
            ‘ ADC0 데이터를 얻어온다.
            iVal = SmartADC1.ReadDataDetailFiltering(0, m_iRawData, m_iAvailable)
        Else
            ‘ ADC0 데이터를 얻어온다.
            iVal = SmartADC1.ReadData(0)
        End If
        ‘ 거리측정계산 함수
        ADC0_DataParsing(iVal)
    Else
        labADCValue0.Text = “ ”
        labADCDistance0.Text = “ ”
    End If
    ‘ …ADC 1 ~ 3 채널의 Read관련 코드 중략 및 ADC 값 표시 및 계싼 코드 VB생략…

    ‘ 그래프의 데이터를 입력 합니다. (인자의 수에 맞추어 그래프의 각각의 계열에 적용됨)
    smartDraw1.Chart.PutData(iVal, iVal1, iVal2, iVal3)
End Sub

```

VB 예시 코드

ADC0 센서값을 거리로 계산해주는 메소드(센서의 구간이 선형적이지 때 적용)

- 센서 특성상 선형적이지 않을경우 구간별로 측정해야 합니다.
- ADC0 센서값을 거리로 계산해주는 공식. 20cm ~ 50cm 구간 사이 변환 공식
- 본 코드는 AD값으로 거리를 계산하는 공식이며 AD값에 따라 거리값의 증/감이 비교적
- 비례하는 구간을 계산한 공식입니다.
- AD값의 구간별로 정확하게 거리를 계산하기위해서는 AD구간별로 계산공식이 필요합니다.

```

Private Sub ADC0_DataParsing(ByVal val As Double)
    ‘ 거리센서의 ADC값 = 1064는 거리20cm, ADC값 = 491은 거리50cm 입니다.
    ‘ ADC값 1064 - 491 = 573(ADC구간) 이고 20cm ~ 50cm(ADC의 거리구간)
    ‘ 계산공식은 573 : 30 = (1064 - val) : x 의 결과값에 20(출발점)을 더합니다.

    ‘ ADC값
    labADCValue0.Text = val.ToString()
    ‘ 거리값
    labADCDistance0.Text = Math.Round((((1064 - val) * 30) / 573) + 20, 0).ToString() & “ cm ”
End Sub

```

※ ADC0채널 거리센서

센서 특성상 선형적이지 않을 경우 구간별로 측정해야 합니다. 본 예제 소스에서 사용한 거리센서는 일부 구간만이 선형적인 관계로 해당 부분을 계산공식으로 표현하였으며 거리센서의 전체구간을 표현하려면 데이터시트를 참고하여 구간별로 측정해야 합니다.

※ ADC1채널 압력센서

센서를 불거나 흡입하면 센서 값이 바뀝니다.

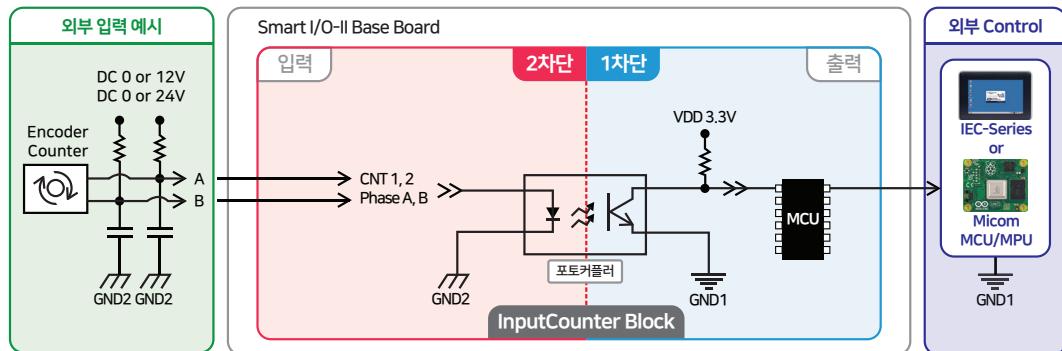
참고

6. InputCounter Block

6-1. InputCounter Block 소개

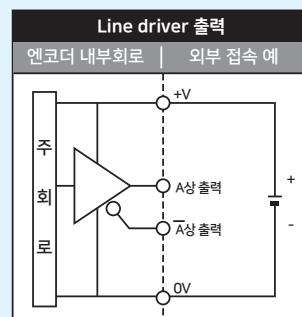
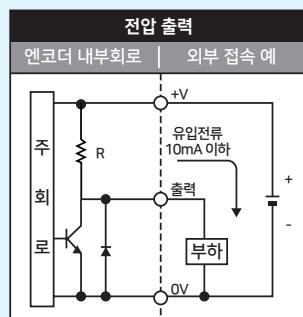
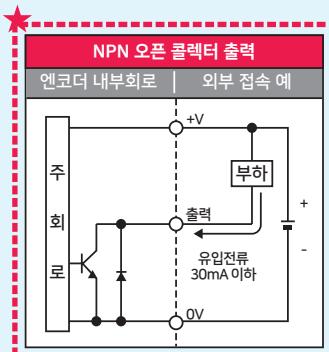
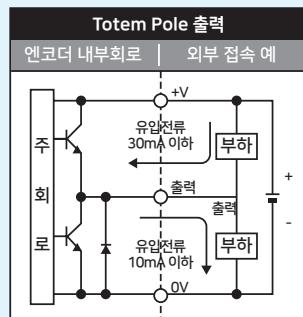
InputCounter Block은 4개를 동시에 사용할 수 있으며 Block 1개당 2채널의 Input Counter와 1채널의 Rotary Encoder Counter를 지원합니다. 카운팅 데이터 범위가 최대 64Bit(기본 32Bit)이며, CheckSum 예외 처리 기능을 제공하여 데이터 신뢰성을 보장합니다.

또한 내부에 독립적인 MCU가 탑재되어 있어 기존 SmartGPIO의 주파수 22Hz(45ms)보다 높은 680KHz(0.00147ms)의 속도로 입력 신호를 처리할 수 있습니다. InputCounter Block의 경우 SmartInputCounter와 연동하여 사용할 수 있습니다.



참고 Rotary Encoder 출력 Type은 NPN 오픈 컬렉터 출력 모델을 사용

[제어출력 회로도]



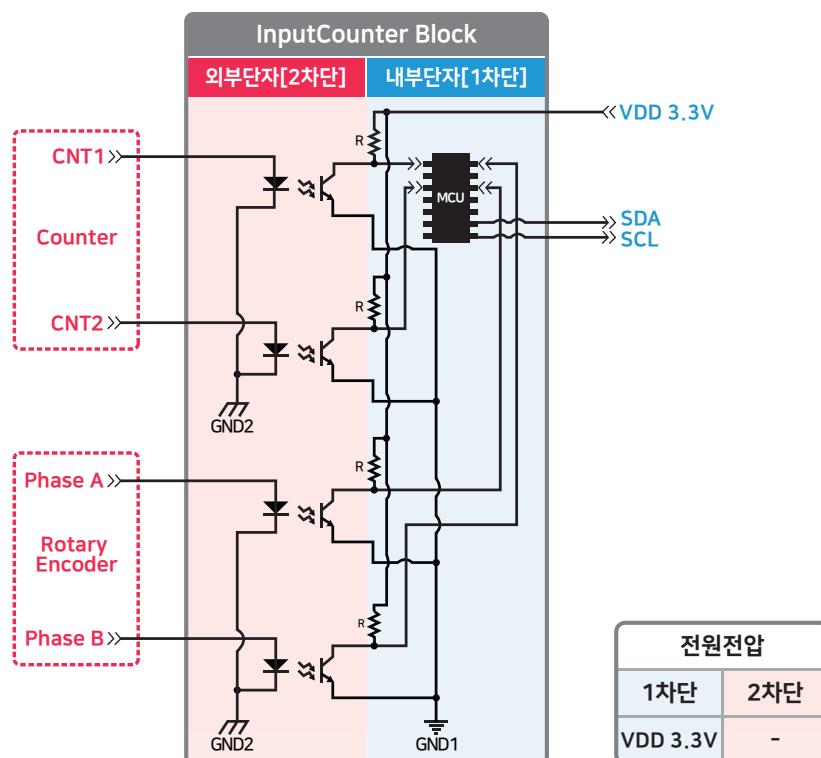
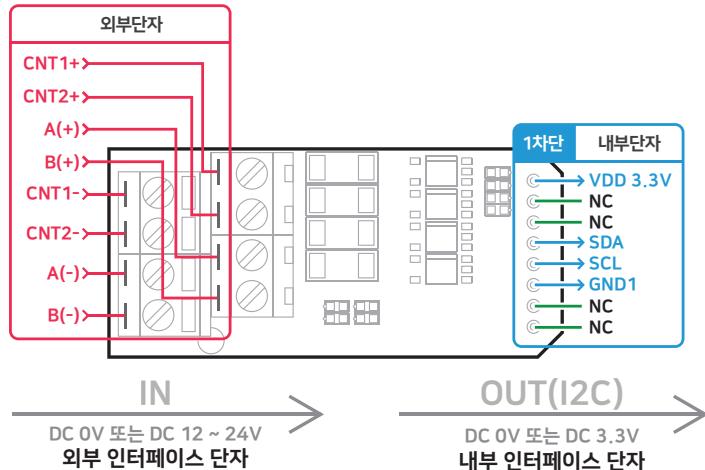
※ Rotary Encoder 연결 시 A, B상만 연결하여 사용합니다.

자세한 Rotary Encoder 데이터 시트는 제조사에서 참고하시기 바랍니다.

6-2. InputCounter Block 연결단자 설명

참고

IEC-Series와 Smart I/O-Series를 연동하여 사용할 경우 내부단자(1차단) 정보는 숙지할 필요 없습니다.
Base Board의 제작 및 별도의 I/O 제어 Board를 제작할 경우에 참고하시기 바랍니다.



※ 외부 인터페이스 단자

핀 명칭	설명	전기적 사양
CNT1+	펄스 입력(카운터) 단자(+극성)	DC 0V or 12V
CNT2+		DC 0V or 24V
CNT1-	펄스 입력(카운터) 단자(+극성)에 대한 GROUND(2차단)	GND
CNT2-		
A(+)	Rotary Encoder A, B상의 입력 단자(+극성)	DC 0V or 12V
B(+)		DC 0V or 24V
A(-)	Rotary Encoder A, B상의 입력 단자(+극성)에 대한 GROUND(2차단)	GND
B(-)		

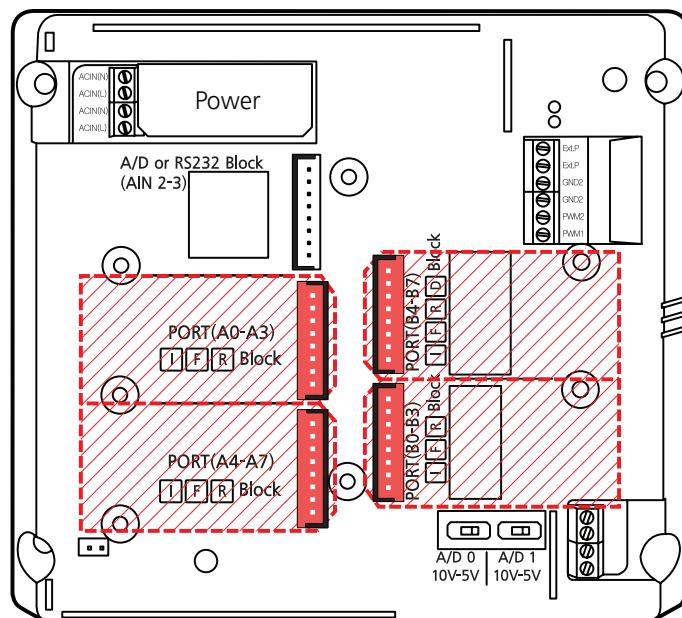
※ 내부 인터페이스 단자

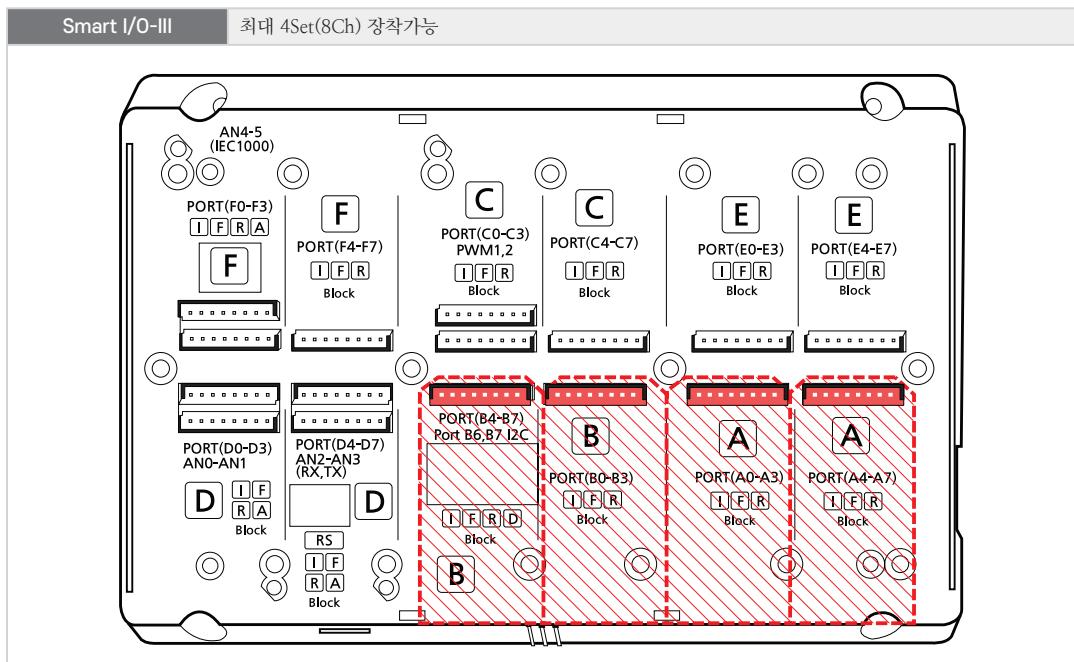
핀 명칭	설명	전기적 사양
VDD 3.3V	전원 입력 공급	DC 3.3V
SDA	출력 신호	DC 0V or 3.3V
SCL		
GND1	VDD 3.3V에 대한 GROUND (내부단자[1차단]의 GND1)	GND
NC	-	-

6-3. InputCounter Block 장착위치

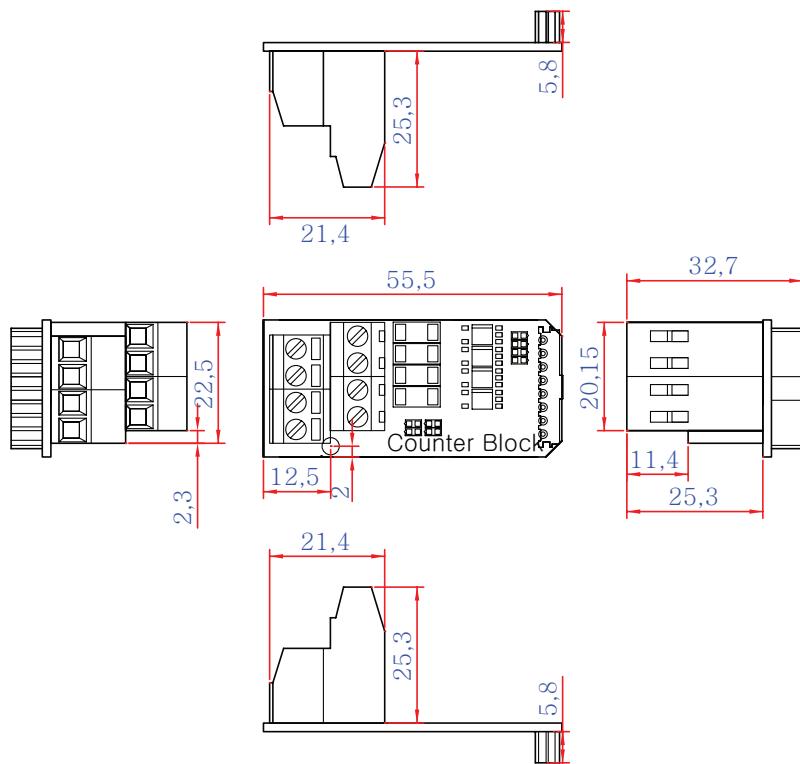
Smart I/O-II

최대 4Set(8Ch) 장착가능





6-4. InputCounter Block 외형 및 치수

**참조**

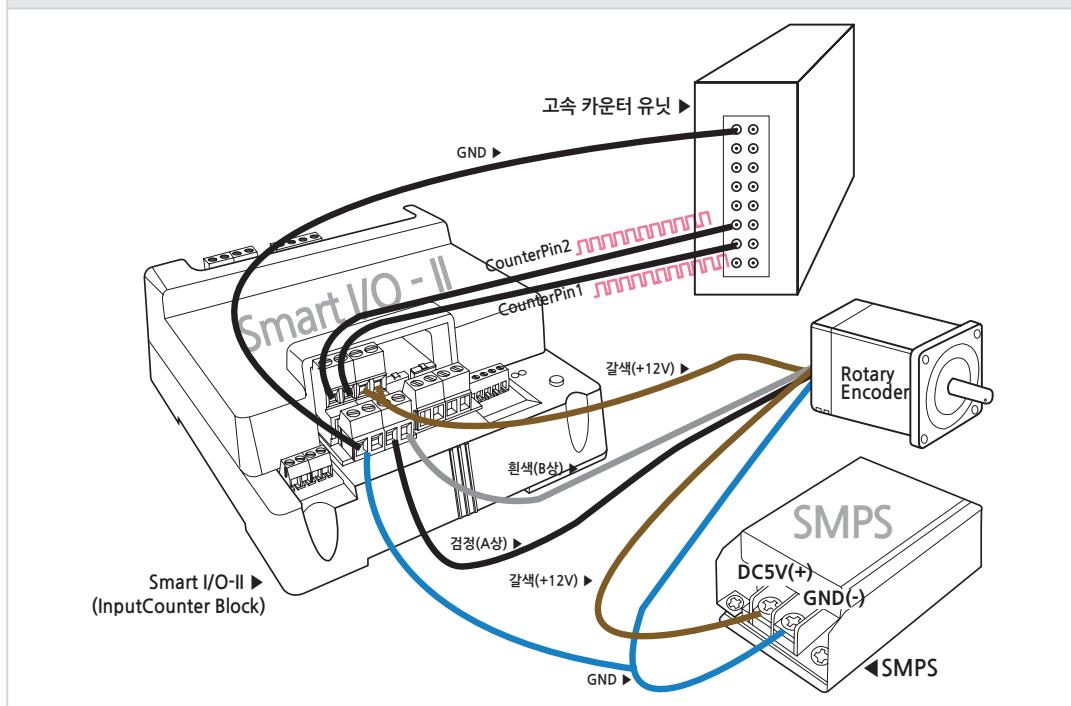
본 제품의 도면은 “[홈페이지](www.hnsts.co.kr) – [자료실] – [도면 및 승인원]”에서 다운로드할 수 있습니다.

6-5. InputCounter Block 응용 방법(외부 인터페이스 결선)

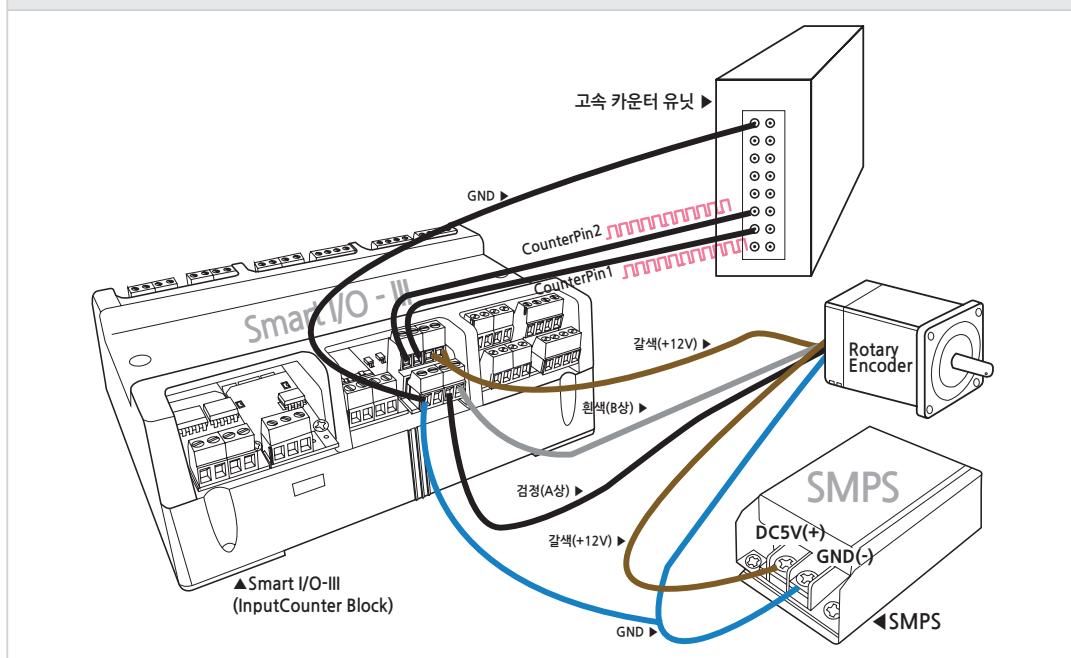
1) 고속 펄스 출력, Rotary Encoder 연결방법

< Smart I/O-II, III Base Board 결선도 예시 >

Smart I/O-II Base Board

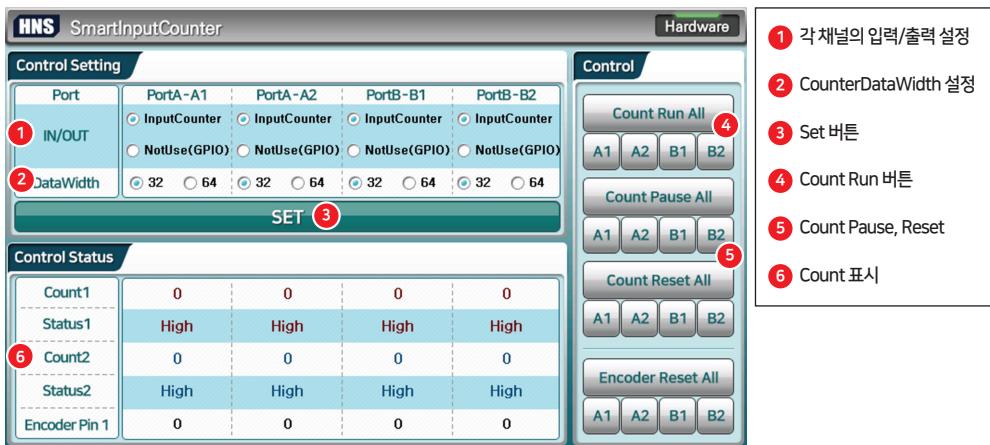


Smart I/O-III Base Board



6-6. 언어별 주요 소스코드 및 응용 예제

SmartX Framework	SmartInputCounter
소스파일	SmartInputCounter 예제
소스경로	[SmartX 홈페이지(www.smarty.co.kr)] → [예제 프로그램] → [개발 언어에 맞는 예제 프로그램 다운로드] → [Hardware Control Component] – [SmartInputCounter]
기능	Pulse Counter, Rotary Encoder 등 입력 기능
응용분야	Rotary Encoder, 외부 고속 입력장치
준비사항	• Rotary Encoder : Autonics(E3OS4-3000-3-N-24)



[동작설명]

- ① RadioButton을 선택하여 Extension Port – A의 A1과 A2채널 그리고 Extension Port – B의 B1과 B2채널의 In/Out 설정을 합니다. [InputCounter / NoUse(GPIO)]
- ② RadioButton을 선택하여 입력 데이터 크기에 따라 32Bit, 64Bit로 CounterDataWidth를 설정합니다.
- ③ Set 버튼을 클릭하여 환경을 설정합니다.
- ④ Count Run All을 클릭하여 고속 입력 Count 또는 Rotary Encoder Count를 시작합니다. 또는 InputCounter로 설정한 블록을 별별적으로 시작이 가능합니다.
- ⑤ Count Run 이후 데이터가 입력되면 ⑥에 입력 Count가 출력됩니다.
- ⑥ Counting을 하는 도중에 Pause All(중지), Reset All(카운팅 초기화)의 동작수행이 가능합니다.

SmartX Framework

SmartInputCounter와 관련된 자세한 사항은 “[SmartX 홈페이지(www.smarty.co.kr)] → [Reference Guide] → [Part 3. 하드웨어 장치 제어 컴포넌트] → [10. SmartInputCounter]” 내용을 참조하시기 바랍니다.

< C# 예제 주요소스 코드 >

참고

C#의 주요 소스 코드의 일부 코드가 생략되어 있습니다. 상세한 코드를 확인하기 위해서는 상단 표에 나와 있는 “소스경로”에서 예제 프로그램을 참고하시기 바랍니다.

C# 예시 코드

사용 방식 선택에 따라 SmartInputCounter의 속성 및 SmartGPIO의 속성 설정

```
// **중요** InputCounter 블럭보다 반드시 먼저 설정 해야합니다.
```

```
// A1이 NotUsed인 경우
```

```
if (radA1_NotUsed.Checked == true)
```

```

{
    // PORT A1을 출력으로 설정
    GPIO_PortA.DirPin0 = SmartX.SmartGPIO.PINDIR.OUTPUT;
    GPIO_PortA.DirPin1 = SmartX.SmartGPIO.PINDIR.OUTPUT;
    GPIO_PortA.DirPin2 = SmartX.SmartGPIO.PINDIR.OUTPUT;
    GPIO_PortA.DirPin3 = SmartX.SmartGPIO.PINDIR.OUTPUT;
    // Port-A(A1)는 Pin의 0번~3번까지 입출력(SmartGPIO)으로 사용할 수 있다. PORT A 출력을 Low
    GPIO_PortA.PortDatas = 0x00;
}

// **중요** 반드시 PortMode 속성을 Initialize()보다 먼저 설정해야 합니다. A1이 Used인 경우
if (radA1_Input.Checked == true)
{
    smartInputCounter1.InputCounterA1.PortMode = SmartX.SmartInputCounter.PORTMODES.Used;
    if (smartInputCounter1.InputCounterA1.Initialize() == true)
    {
        // smartInputCounter1의 InputCounterA1의 데이터 모드를 32Bit로 설정
        // 32Bit의 경우 최대 4,294,967,295를 표현 가능하며 64Bit의 경우 최대 1.8E308까지 표현 가능합니다.
        if (radA1_DW32.Checked == true)
        {
            smartInputCounter1.InputCounterA1.CounterDataWidth = SmartX.SmartInputCounter.DATAWIDTH._32BIT;
        }
        else
        {
            smartInputCounter1.InputCounterA1.CounterDataWidth = SmartX.SmartInputCounter.DATAWIDTH._64BIT;
        }
    }
}
else
{
    smartInputCounter1.InputCounterA1.PortMode = SmartX.SmartInputCounter.PORTMODES.NotUsed;
}

```

C# 예시 코드

SmartInputCounter의 시작(Run) / 중지(Pause) / 재시작(Reset)

```

// InputCounter or Output 시작(A1, A2, B1, B2)
private void btnRun_Click(object sender, EventArgs e)
{
    if (radA1_Input.Checked == true && sender == btnRunA1)
    {
        smartInputCounter1.InputCounterA1.RunCounterPin1();
        smartInputCounter1.InputCounterA1.RunCounterPin2();
    }
    else if (radA2_Input.Checked == true && sender == btnRunA2)
    {
        smartInputCounter1.InputCounterA2.RunCounterPin1();
        smartInputCounter1.InputCounterA2.RunCounterPin2();
    }
    else if (radB1_Input.Checked == true && sender == btnRunB1)
    {
        smartInputCounter1.InputCounterB1.RunCounterPin1();
        smartInputCounter1.InputCounterB1.RunCounterPin2();
    }
    else if (radB2_Input.Checked == true && sender == btnRunB2)
    {

```

```

        smartInputCounter1.InputCounterB2.RunCounterPin1();
        smartInputCounter1.InputCounterB2.RunCounterPin2();
    }
    if (DataCountingTimer.IsStart == false)
    {
        DataCountingTimer.Interval = 1;
        DataCountingTimer.Start();
    }
}
// InputCounter or Output 중지, 재시작 코드 생략...

```

C# 예시 코드

현재 카운팅 데이터 값을 확인

```

private void DataCountingTimer_Tick(object sender, EventArgs e)
{
    // A1이 출력
    if (radA1_NotUsed.Checked == true)
    {
        GPIO_OutputTest(GPIO_PortA, 0x0F);
    }
    // A1이 입력
    else if (radA1_Input.Checked == true)
    {
        try
        {
            // Pin의 Counter 값, EncoderCounter값을 Label에 출력합니다.
            labCounterPinA10.Text = smartInputCounter1.InputCounterA1.CounterPin1.ToString();
            labCounterPinA11.Text = smartInputCounter1.InputCounterA1.CounterPin2.ToString();
            labCounterPinA12.Text = smartInputCounter1.InputCounterA1.EncoderCounterCh1.ToString();

            if (smartInputCounter1.InputCounterA1.StatusCounterPin1 == true)
            {
                labStatusA10.Text = " High ";
            }
            else
            {
                labStatusA10.Text = " Low ";
            }

            if (smartInputCounter1.InputCounterA1.StatusCounterPin2 == true)
            {
                labStatusA11.Text = " High ";
            }
            else
            {
                labStatusA11.Text = " Low ";
            }
        }
        catch (SmartX.SmartInputCounter.CheckSumErrorException)
        {
            DataCountingTimer.Stop();
            SmartX.SmartMessageBox.Show(" Data Error ");
        }
    }
}

```

< VB 예제 주요소스 코드 >

참고

VB.NET 주요 소스 코드에서는 일부 코드가 생략되어 있습니다. 상세한 코드를 확인하기 위해서는 상단 표에 나와있는 “소스경로”에서 예제 프로그램을 참고하시기 바랍니다.

VB 예시 코드

사용 방식 선택에 따라 SmartInputCounter의 속성 및 SmartGPIO의 속성 설정

```
' **중요** InputCounter 블럭보다 반드시 먼저 설정 해야합니다. A1이 NotUsed인 경우
If radA1_NotUsed.Checked = True Then
    ' …PORT A1을 출력으로 설정 코드는 VB 예시 코드에서 생략되었습니다.…


    ' Port-A(A1)는 Pin의 0번~3번까지 입출력(SmartGPIO)으로 사용할 수 있다. PORT A 출력을 Low
    GPIO_Ports.PortA.PortDatas = &H0
End If

' **중요** 반드시 PortMode 속성을 Initialize()보다 먼저 설정해야 합니다. A1이 Used인 경우
If radA1_Input.Checked = True Then
    SmartInputCounter1.InputCounterA1.PortMode = SmartX.SmartInputCounter.PORTMODES.Used

    If SmartInputCounter1.InputCounterA1.Initialize() = True Then
        ' smartInputCounter1의 InputCounterA1의 데이터 모드를 32Bit로 설정
        ' 32Bit의 경우 최대 4,294,967,295를 표현 가능하며 64Bit의 경우 최대 1.8E308까지 표현 가능합니다.
        If radA1_DW32.Checked = True Then
            SmartInputCounter1.InputCounterA1.CounterDataWidth = SmartX.SmartInputCounter.DATAWIDTH._32BIT
        Else
            SmartInputCounter1.InputCounterA1.CounterDataWidth = SmartX.SmartInputCounter.DATAWIDTH._64BIT
        End If
    End If
Else
    SmartInputCounter1.InputCounterA1.PortMode = SmartX.SmartInputCounter.PORTMODES.NotUsed
End If
```

VB 예시 코드

SmartInputCounter의 시작(Run) / 중지(Pause) / 재시작(Reset)

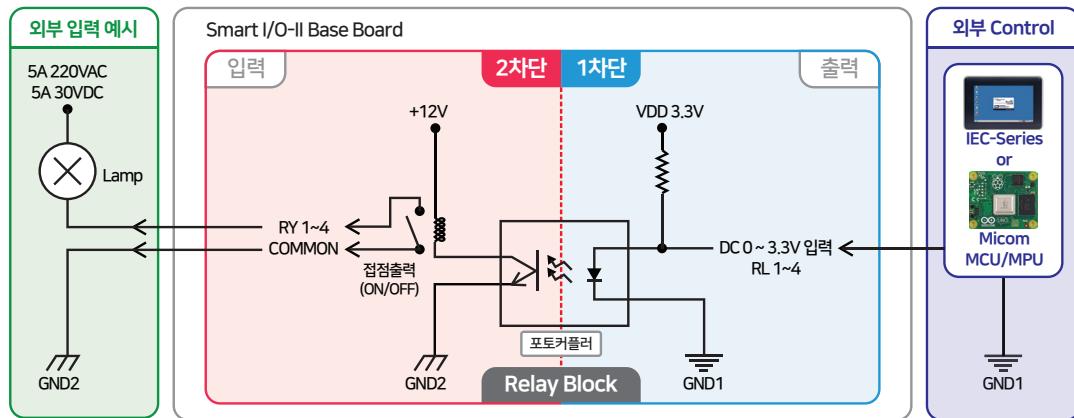
```
' InputCounter or Output 시작(A1, A2, B1, B2)
Private Sub btnRun_Click(ByVal sender As Object, ByVal e As EventArgs) Handles btnRunB2.Click, btnRunB1.
Click, btnRunA2.Click, btnRunA1.Click
    If radA1_Input.Checked = True AndAlso sender Is btnRunA1 Then
        SmartInputCounter1.InputCounterA1.RunCounterPin1()
        SmartInputCounter1.InputCounterA1.RunCounterPin2()
    ElseIf radA2_Input.Checked = True AndAlso sender Is btnRunA2 Then
        SmartInputCounter1.InputCounterA2.RunCounterPin1()
        SmartInputCounter1.InputCounterA2.RunCounterPin2()
    ElseIf radB1_Input.Checked = True AndAlso sender Is btnRunB1 Then
        SmartInputCounter1.InputCounterB1.RunCounterPin1()
        SmartInputCounter1.InputCounterB1.RunCounterPin2()
    ElseIf radB2_Input.Checked = True AndAlso sender Is btnRunB2 Then
        SmartInputCounter1.InputCounterB2.RunCounterPin1()
        SmartInputCounter1.InputCounterB2.RunCounterPin2()
    End If
    If DataCountingTimer.IsStart = False Then
        DataCountingTimer.Interval = 1
        DataCountingTimer.Start()
    End If
End Sub
' InputCounter or Output 중지, 재시작 코드 생략…
```

VB 예시 코드	현재 카운팅 데이터 값을 확인
<pre>' Input Counter의 Count 값, Status 값 출력 Private Sub DataCountingTimer_Tick(ByVal sender As Object, ByVal e As EventArgs) Handles DataCountingTimer.Tick If radA1_NotUsed.Checked = True Then ' A1이 출력 GPIO_OutputTest(GPIO_PortA, &HF) ElseIf radA1_Input.Checked = True Then ' A1이 입력 Try ' Pin의 Counter 값, EncoderCounter값을 Label에 출력합니다. labCounterPinA10.Text = SmartInputCounter1.InputCounterA1.CounterPin1.ToString() labCounterPinA11.Text = SmartInputCounter1.InputCounterA1.CounterPin2.ToString() labCounterPinA12.Text = SmartInputCounter1.InputCounterA1.EncoderCounterCh1.ToString() If SmartInputCounter1.InputCounterA1.StatusCounterPin1 = True Then labStatusA10.Text = "High" Else labStatusA10.Text = "Low" End If If SmartInputCounter1.InputCounterA1.StatusCounterPin2 = True Then labStatusA11.Text = "High" Else labStatusA11.Text = "Low" End If Catch CheckSumErrorException As SmartX.SmartInputCounter.CheckSumErrorException DataCountingTimer.Stop() SmartX.SmartMessageBox.Show("Data Error") End Try End If End Sub</pre>	

7. Relay Block

7-1. Relay Block 소개

Relay 접점 Output은 Relay 접점을 이용하여 간단한 접점 제어를 할 수 있습니다. (5A 250VAC/5A 30VDC) Relay Block의 경우 SmartGPIO와 연동하여 사용할 수 있습니다.

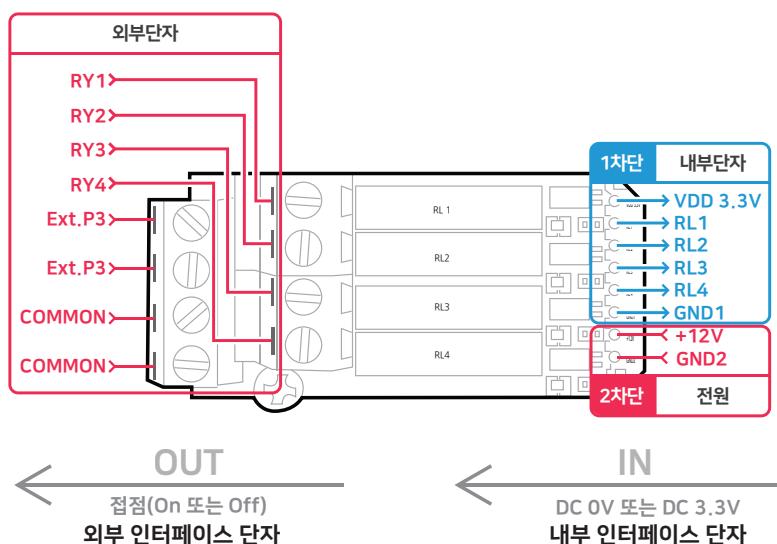

주의

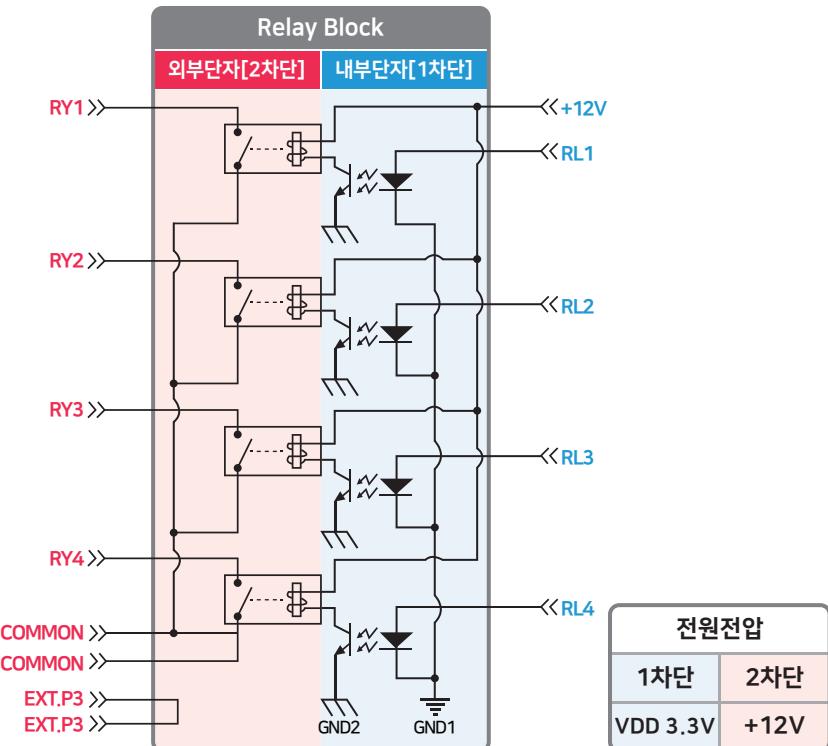
Relay COM단자는 그림과 같이 COMMON 단자에 공통으로 묶여 있습니다.

7-2. Relay Block 연결단자 설명

참고

IEC-Series와 Smart I/O-Series를 연동하여 사용할 경우 내부단자(1차단) 정보는 숙지할 필요 없습니다. Base Board의 제작 및 별도의 I/O 제어 Board를 제작할 경우에 참고하시기 바랍니다.

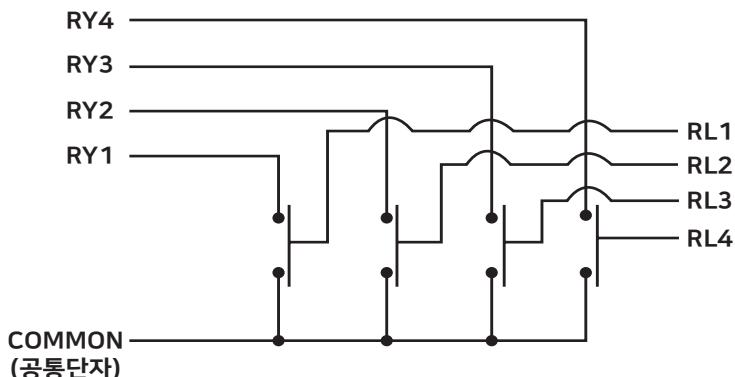




※ 외부 인터페이스 단자

핀 명칭	설명	전기적 사양
Ext.P3	공핀(Ext.P3는 서로 묶임)	-
COMMON	공통단자(Relay Contact Common)	-
RY1		
RY2	Relay 접점(출력접점)	Max 5A/250VAC,
RY3	Relay Contact & Common On/Off	5A/30VDC
RY4		

※ COMMON



4채널 Relay SW 접점 중 한쪽 단자는 공통으로 묶여 있습니다.

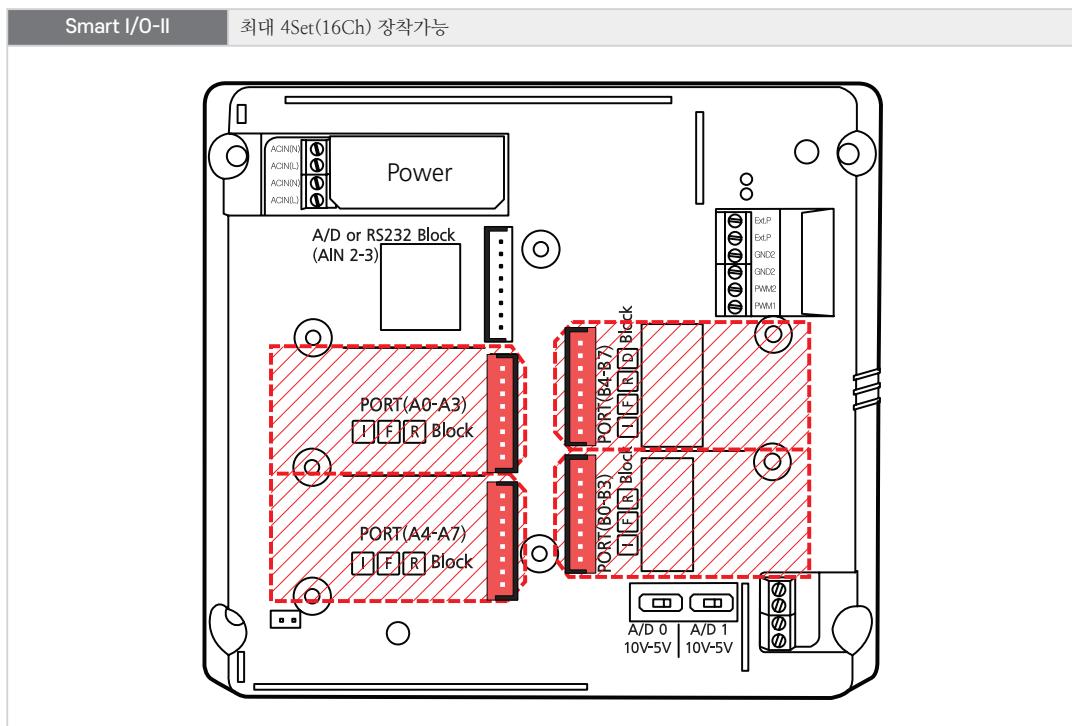
※ 내부 인터페이스 단자

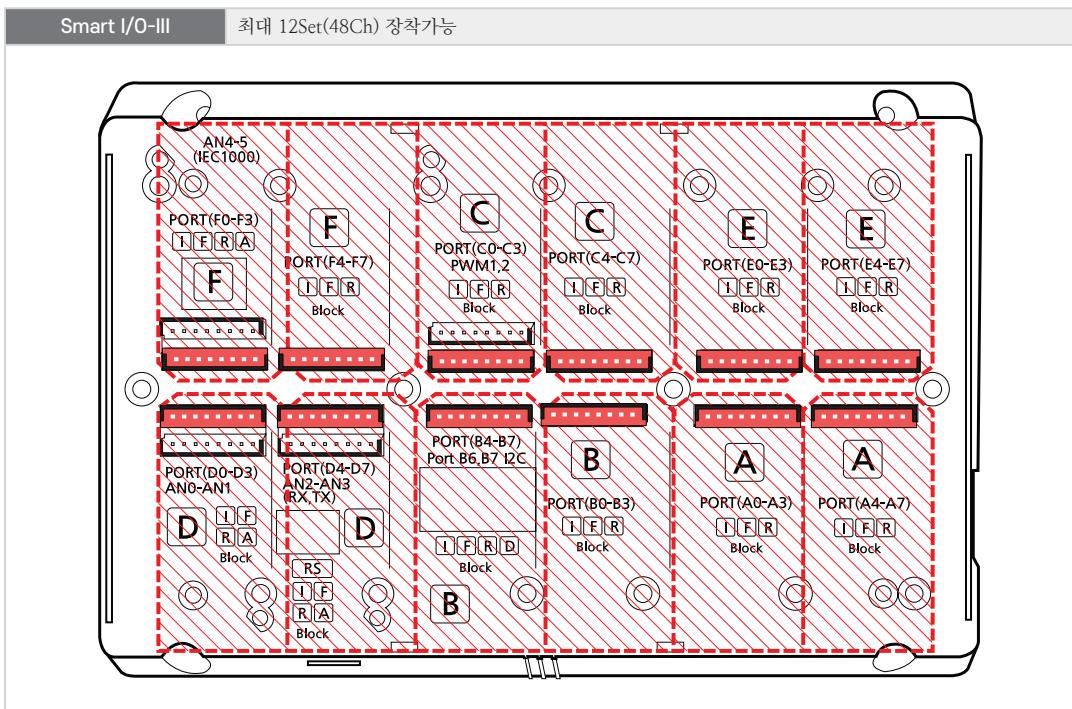
핀 명칭	설명	전기적 사양
+12V	전원 입력 공급	DC 12V@50mA
GND2	+12V에 대한 GROUND (전원[2차단]의 GND2)	GND
VDD 3.3V	전원 입력 공급	DC 3.3V@60mA
GND1	VDD 3.3V에 대한 GROUND (내부단자[1차단]의 GND1)	GND
RL1		
RL2		
RL3	입력 신호	DC 0V or 3.3V@7mA
RL4		

※ SmartGPIO와 연동 시 로직 테이블

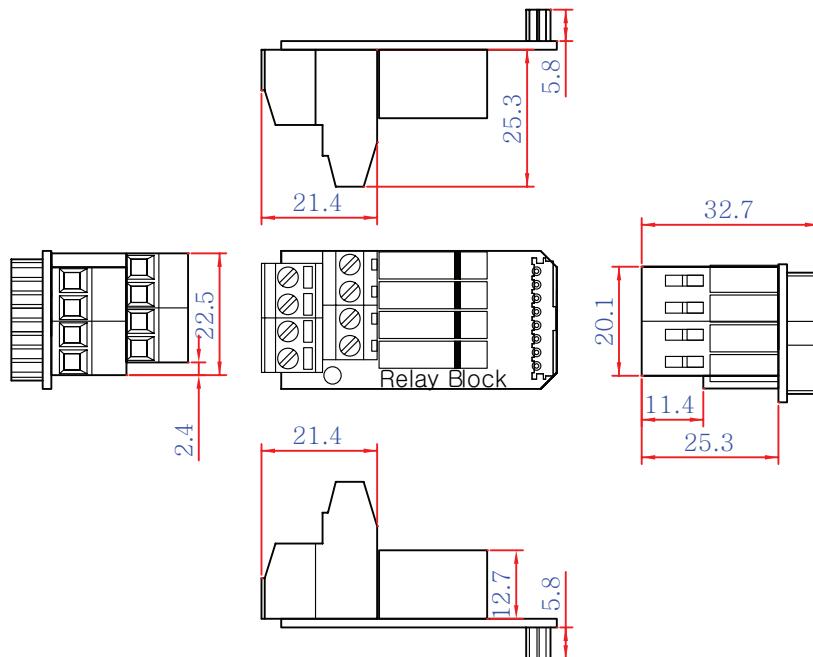
Direction	Port Data 값(High/Low)	내부 인터페이스 RLx 입력 신호	외부 인터페이스 출력 상태
입력	High(1)	3.3V	ON
	Low(0)	0V	OFF

7-3. Relay Block 장착위치





7-4. Relay Block 외형 및 치수



참조

본 제품의 도면은 “[홈페이지](www.hnsts.co.kr) – [자료실] – [도면 및 승인원]”에서 다운로드할 수 있습니다.

7-5. Relay Block 응용 방법(외부 인터페이스 결선)

1) 할로겐램프 연결방법

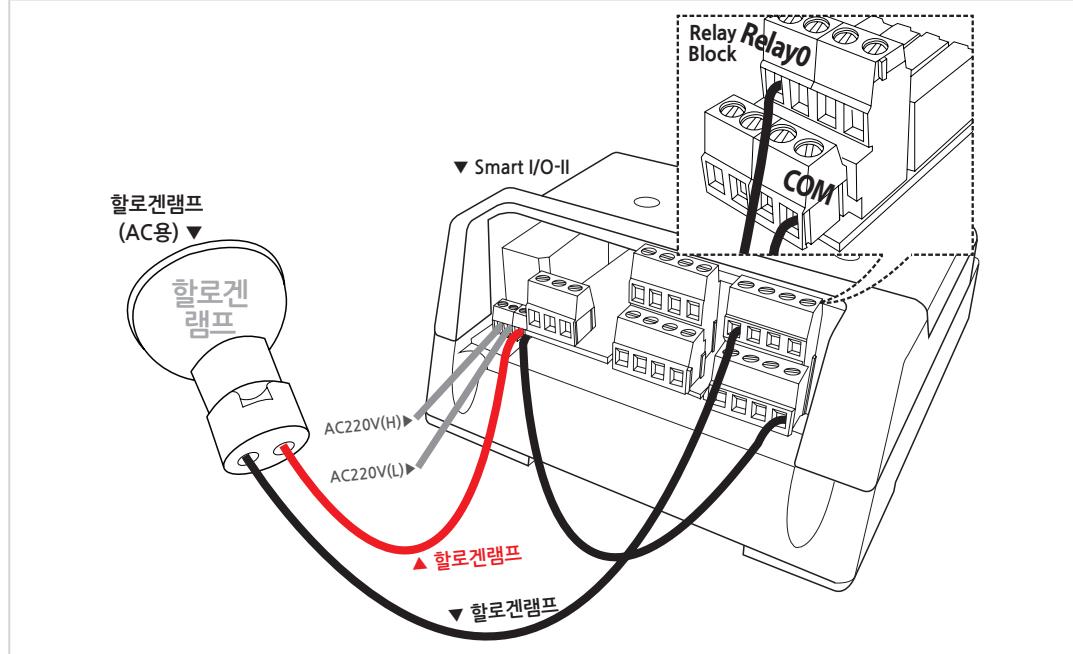
아래와 같이 할로겐램프를 결선하시기 바랍니다.

참고

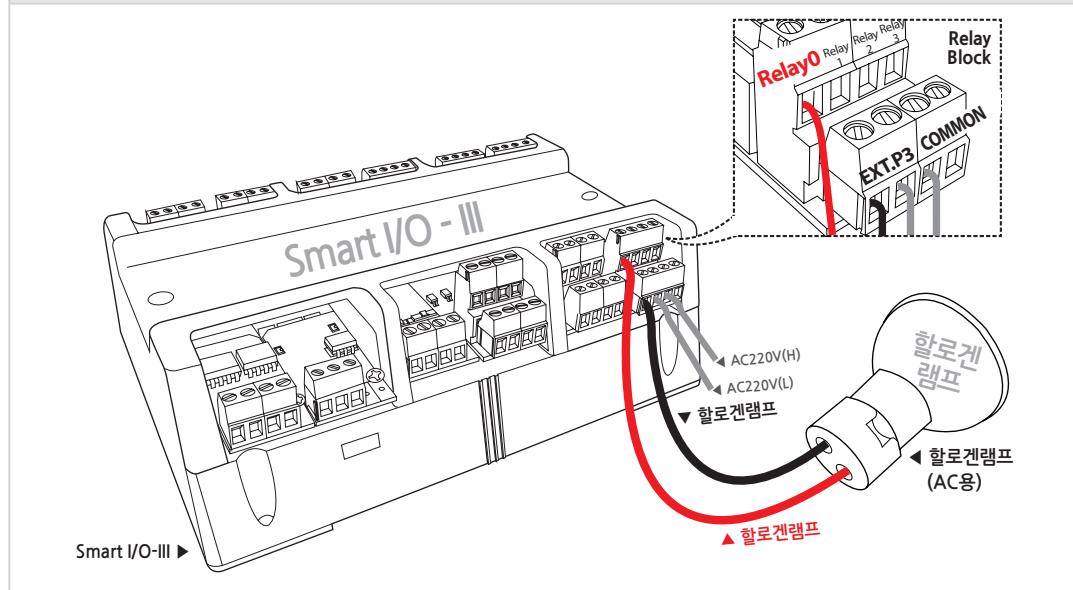
할로겐램프는 (+/-) 구분이 없습니다.

< Smart I/O-II, III Base Board 결선도 예시 >

Smart I/O-II Base Board

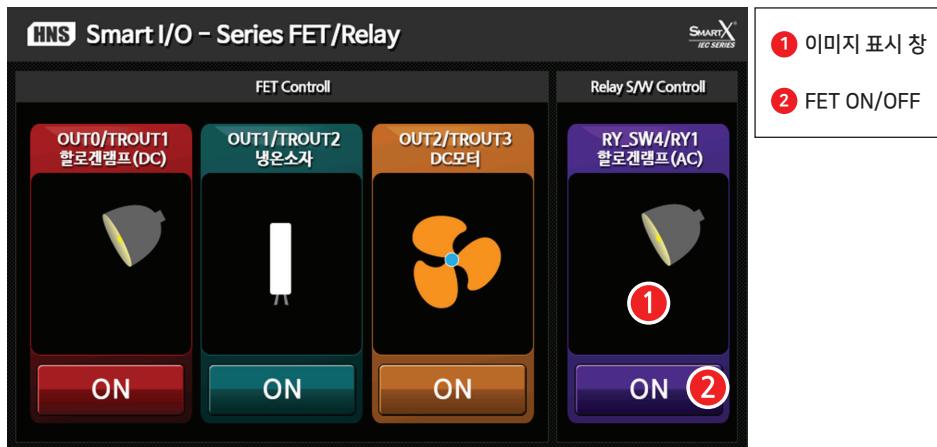


Smart I/O-III Base Board



7-6. 언어별 주요 소스코드 및 응용 예제

SmartX Framework	SmartGPIO
소스파일	TechNote55_SmartIO_Output
소스경로	[홈페이지](www.hnsts.co.kr) → [자료실] → [Tech Note] → [55. [C#, VB.NET] Smart I/O-Series FET/Relay 출력 사용 예제]
기능	AC/DC 전원으로 구동하는 장비 연동 시 ON/OFF 조작
응용분야	AC/DC 스위치 기능
준비사항	[PIN4] 할로겐램프 : (AC 220V, 50W)



[동작설명]

- 프로그램을 시작한 후 ② 각 포트의 ON 버튼을 클릭하면 해당 포트 별로 출력신호(LOW→HIGH)가 발생하면서 ① 이미지가 변경됩니다. 4번째 RY_SW4/RY1 할로겐램프(AC)의 경우에는 Relay 기능의 테스트가 가능합니다.

SmartX Framework

SmartGPIO와 관련된 자세한 내용은 “[SmartX 홈페이지](www.smartx.co.kr) → [Reference Guide] → [Part 3. 하드웨어 장치 제어 컴포넌트] → [2. SmartGPIO]”내용을 참조하시기 바랍니다.

< C# 예제 주요소스 코드 >

참고

C#의 주요 소스 코드의 일부 코드가 생략되어 있습니다. 상세한 코드를 확인하기 위해서는 상단 표에 나와 있는 “소스경로”에서 예제 프로그램을 참고하시기 바랍니다.

C# 예시 코드

Form1_Load 이벤트에서 SmartGPIO 초기화

```
private void Form1_Load(object sender, EventArgs e)
{
    // GPIO Port를 A로 설정
    smartGPIO1.PortSelect = SmartX.SmartGPIO.PORTID.PORTA;
    // PortA의 방향을 ALL_OUTPUT으로 설정
    smartGPIO1.PortDirs = 255;
    // PortA의 상태를 ALL_Low로 설정
    smartGPIO1.PortDatas = 0;
    // ...이미지 설정 코드 생략...
}
```

C# 예시 코드 btnAClampCtrl 버튼이 클릭되면 smartGPIO1.Data4의 상태 값(High/Low)을 변경

```
// Relay S/W AC할로겐램프 ON/OFF 제어
private void btnAClampCtrl_Click(object sender, EventArgs e)
{
    // AC 할로겐램프 ON
    if (btnAClampCtrl.ButtonStatus == SmartX.SmartButton.BUTSTATUS.DOWN)
    {
        smartGPIO1.Data4 = SmartX.SmartGPIO.PinStatus.High;
        pbAClamp.DrawNext(1);
    }
    // AC 할로겐램프 OFF
    else
    {
        smartGPIO1.Data4 = SmartX.SmartGPIO.PinStatus.Low;
        pbAClamp.DrawNext(0);
    }
}
```

< VB 예제 주요소스 코드 >

참고

VB.NET 주요 소스 코드에서는 일부 코드가 생략되어 있습니다. 상세한 코드를 확인하기 위해서는 상단 표에 나와있는 “소스경로”에서 예제 프로그램을 참고하시기 바랍니다.

VB 예시 코드 Form1_Load 이벤트에서 SmartGPIO 초기화

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As EventArgs) Handles MyBase.Load
    ' GPIO Port를 A로 설정
    SmartGPIO1.PortSelect = SmartX.SmartGPIO.PORTID.PORTA
    ' PortA의 방향을 ALL_OUTPUT으로 설정
    SmartGPIO1.PortDirs = 255
    ' PortA의 상태를 ALL_Low로 설정
    SmartGPIO1.PortDatas = 0
    ' ...이미지 설정 코드 생략...
End Sub
```

VB 예시 코드 btnAClampCtrl 버튼이 클릭되면 smartGPIO1.Data4의 상태 값(High/Low)을 변경

```
' Relay S/W AC할로겐램프 ON/OFF 제어
Private Sub btnAClampCtrl_Click(ByVal sender As Object, ByVal e As EventArgs) Handles btnAClampCtrl.Click
    If btnAClampCtrl.ButtonStatus = SmartX.SmartButton.BUTSTATUS.DOWN Then
        ' AC 할로겐램프 ON
        SmartGPIO1.Data4 = SmartX.SmartGPIO.PinStatus.High
        pbAClamp.DrawNext(1)
    Else
        ' AC 할로겐램프 OFF
        SmartGPIO1.Data4 = SmartX.SmartGPIO.PinStatus.Low
        pbAClamp.DrawNext(0)
    End If
End Sub
```

8. FET Block

8-1. FET Block 소개

N-Channel FET을 사용하였으며 회로와 같이 드레인(Open Drain)에 연결되어 있습니다. 구동하고자 하는 부하에 따라 별도의 전원이 필요합니다. FET Block의 경우 SmartGPIO, SmartPWM과 연동하여 사용할 수 있습니다.

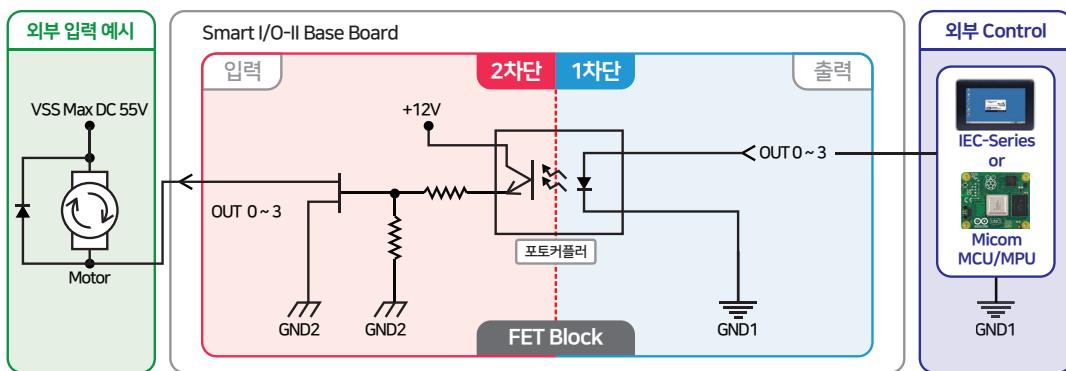
주의

전력량에 따라 FET 소자에 발열이 발생할 수 있으므로 발열이 적게 발생하도록 발열량을 검토하고 전력량을 줄여 사용하시기 바랍니다.

참고

FET Block 출력속도

70kHz 이하 사용을 권장하며 70kHz보다 높은 주파수에서 동작할 경우 슬루 레이트(Slew Rate)에 의해 Duty Rate가 변경될 수 있습니다.

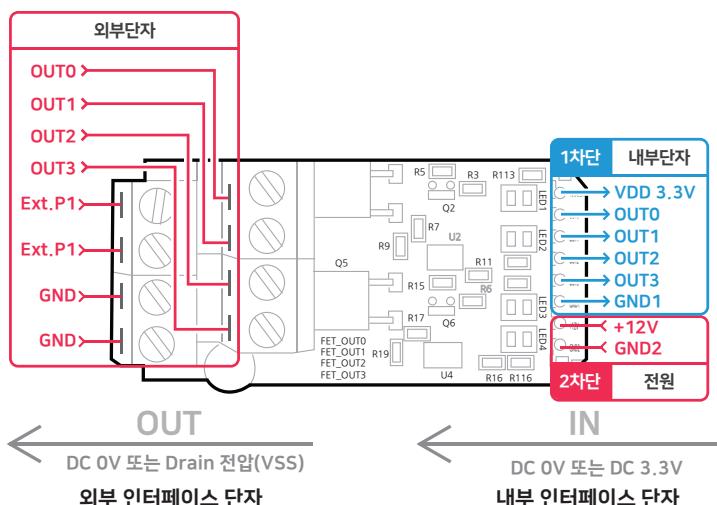

주의

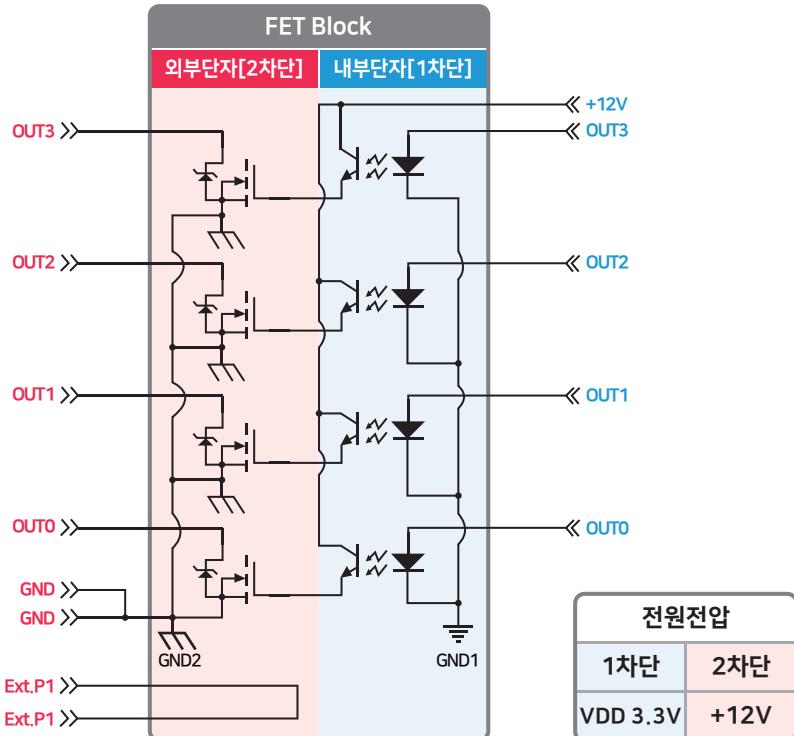
IRFR024N은 VSS = 55V, $I_d = 17A$ 정격의 FET입니다. I_d (Drain Current)는 17A로 지속적인 출력으로 사용할 경우 심한 발열이 발생할 수 있으므로 전류를 낮게 사용하시기 바랍니다.

8-2. FET Block 연결단자 설명

참고

IEC-Series와 Smart I/O-Series를 연동하여 사용할 경우 내부단자(1차단) 정보는 숙지할 필요 없습니다. Base Board의 제작 및 별도의 I/O 제어 Board를 제작할 경우에 참고하시기 바랍니다.





※ 외부 인터페이스 단자

핀 명칭	설명	전기적 사양
Ext.P1	공핀(Ext.P1은 서로 묶임)	-
GND	출력 신호에 대한 GROUND (외부단자[2차단]의 GND2)	GND
OUT0		
OUT1	출력 신호(FET Turn-On/Off)	VSS(Max 55V/포트당 17A)
OUT2	FET Open Drain 출력 단자로 외부 VSS 전원이 인가된 상태의 Load(회로) 구성이 필요합니다.	(주의) 별열에 따른 전류 제한 필요
OUT3		

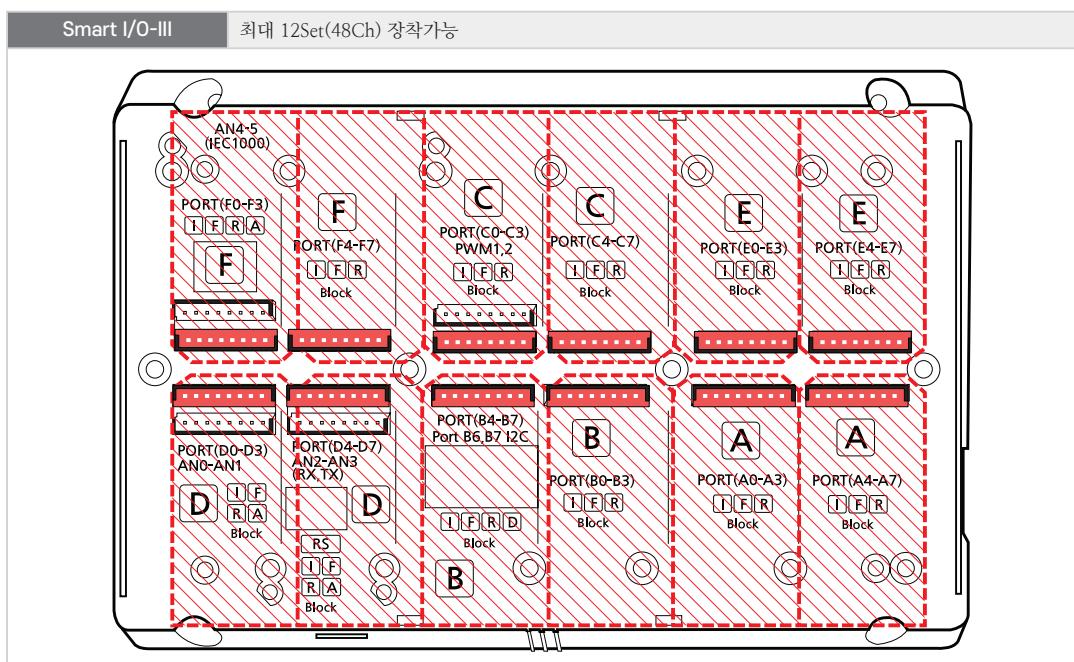
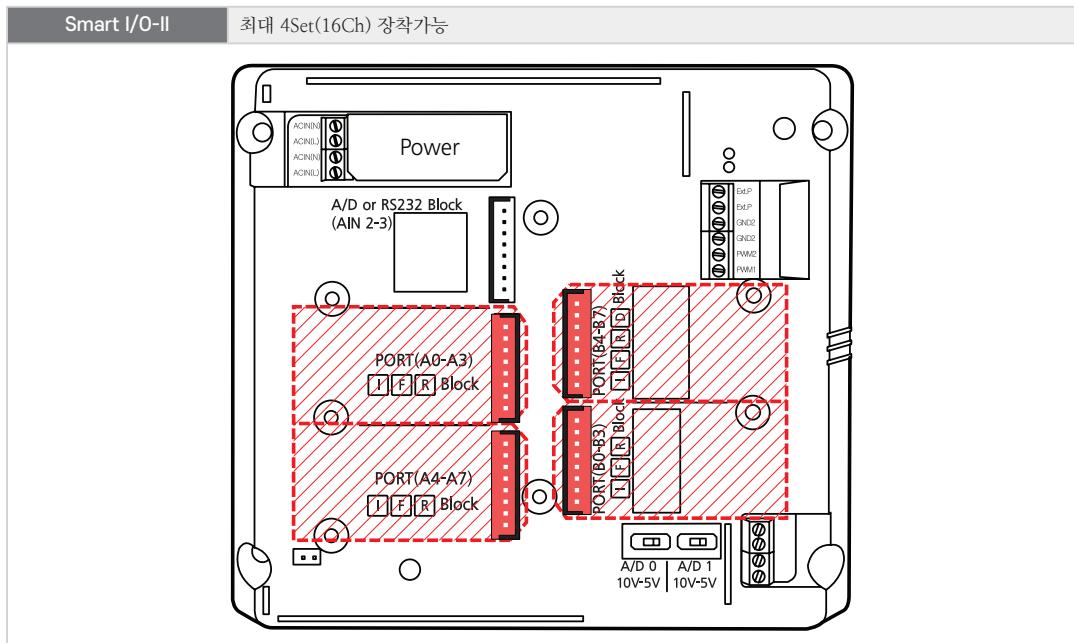
※ 내부 인터페이스 단자

핀 명칭	설명	전기적 사양
+12V	전원 입력 공급	DC 12V@8mA
GND2	+12V에 대한 GROUND (전원[2차단]의 GND2)	GND
VDD 3.3V	전원 입력 공급	DC 3.3V@40mA
GND1	VDD 3.3V에 대한 GROUND (내부단자[1차단]의 GND1)	GND
OUT0		
OUT1		
OUT2	입력 신호	DC 0V or 3.3V@7mA
OUT3		

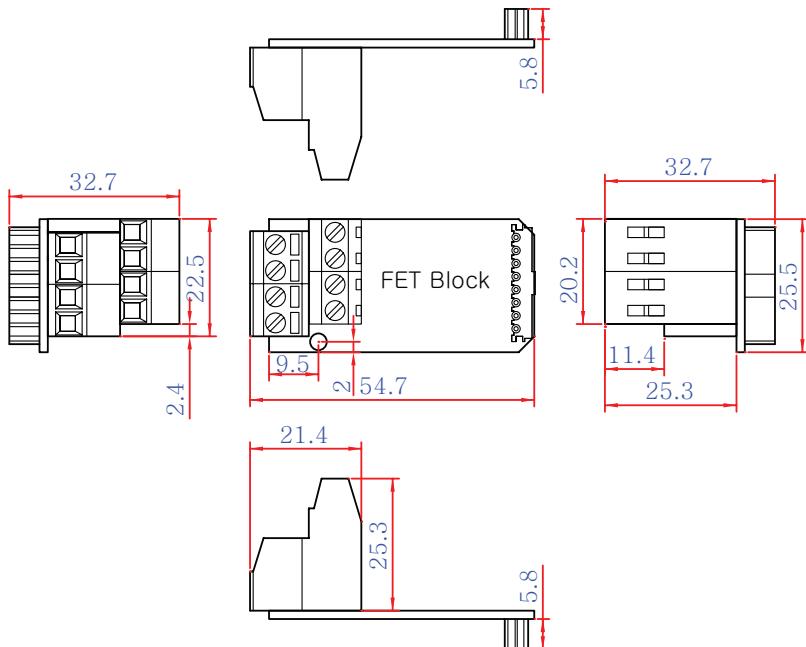
※ SmartGPIO와 연동 시 로직 테이블

Direction	Port Data 값(High/Low)	내부 인터페이스 OUT 입력 신호	외부 인터페이스 출력상태
입력	High(1)	3.3V	ON
	Low(0)	0V	OFF

8-3. FET Block 장착위치



8-4. FET Block 외형 및 치수



참조

본 제품의 도면은 “[홈페이지](www.hnsts.co.kr) – [자료실] – [도면 및 승인원]”에서 다운로드할 수 있습니다.

8-5. FET Block 응용 방법(외부 인터페이스 결선)

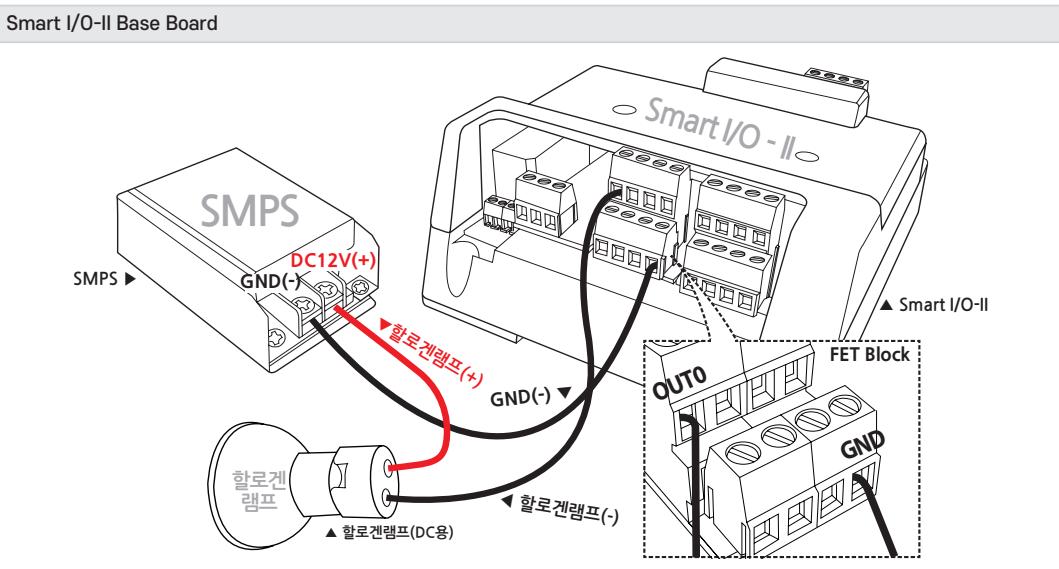
1) 할로겐램프 연결방법

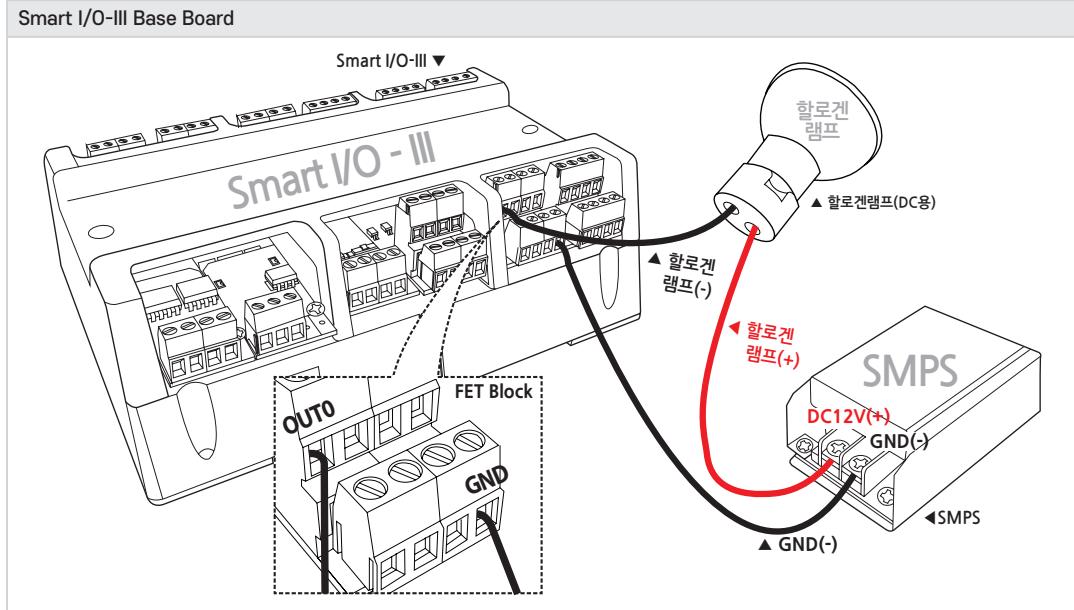
아래와 같이 할로겐램프를 결선하시기 바랍니다.

참고

할로겐램프는 (+/-) 구분이 없습니다.

< Smart I/O-II, III Base Board 결선도 예시 >





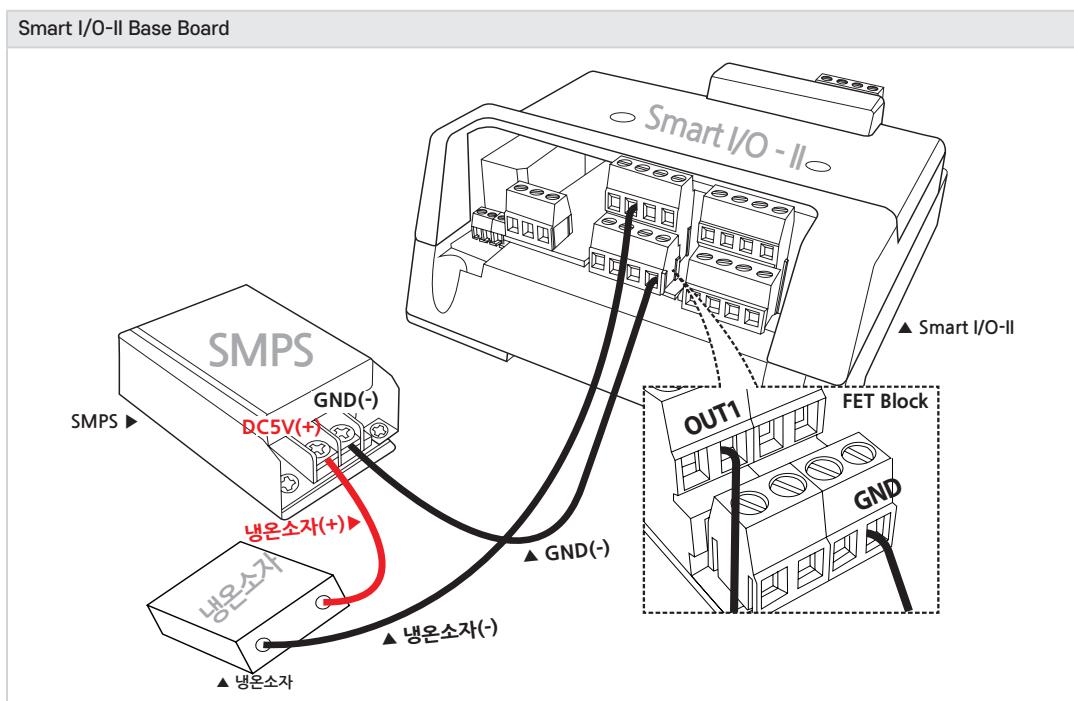
2) 냉온소자 연결방법

아래와 같이 냉온소자를 결선하시기 바랍니다.

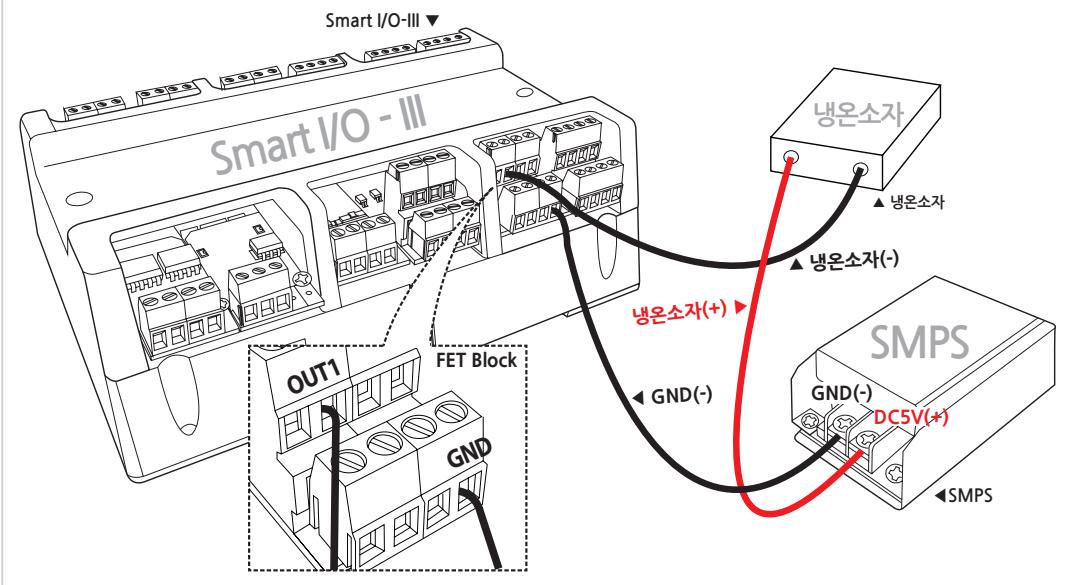
참고

1. 전원을 반대로 연결하는 경우 차가운 면과 뜨거운 면이 바뀝니다.
2. 발열판으로 뜨거운 면의 열을 식혀줘야 뜨거운 면이 차가운 면으로 열전도 되는 현상을 막을 수 있습니다.

< Smart I/O-II, III Base Board 결선도 예시 >



Smart I/O-III Base Board



3) DC모터 연결방법

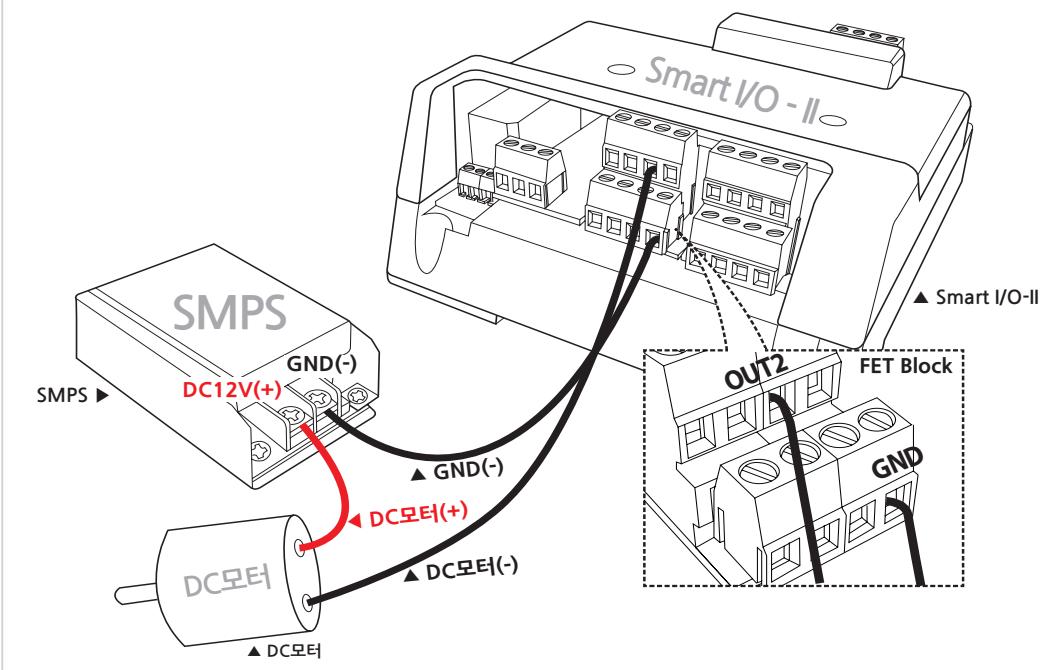
아래와 같이 DC모터를 결선하시기 바랍니다.

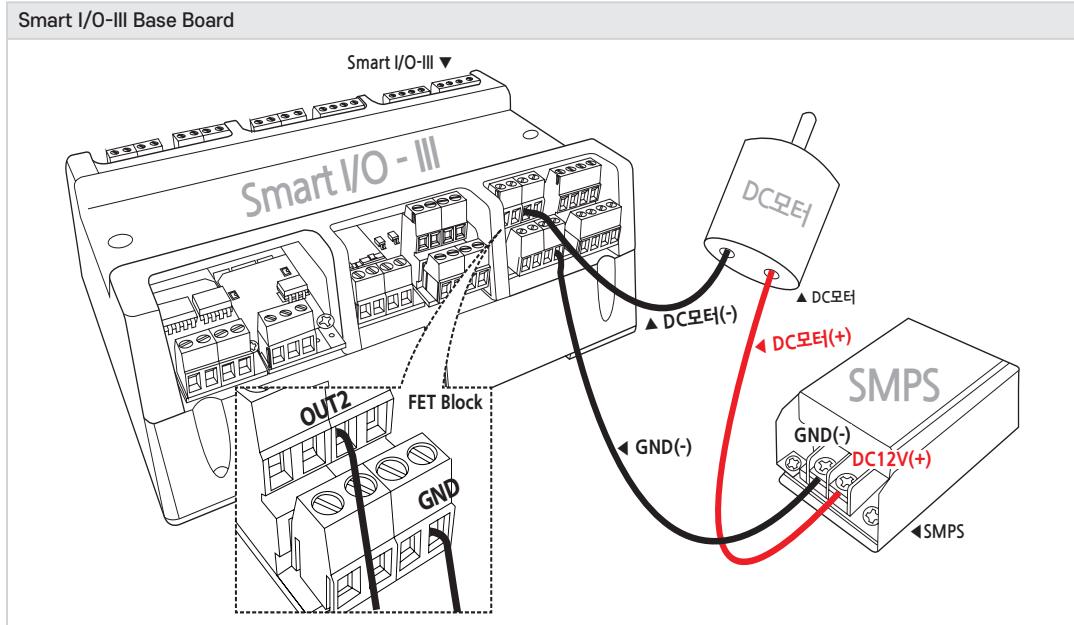
참고

1. DC모터는 (+/-)를 반대로 연결하는 경우 역방향으로 회전합니다.

< Smart I/O-II, III Base Board 결선도 예시 >

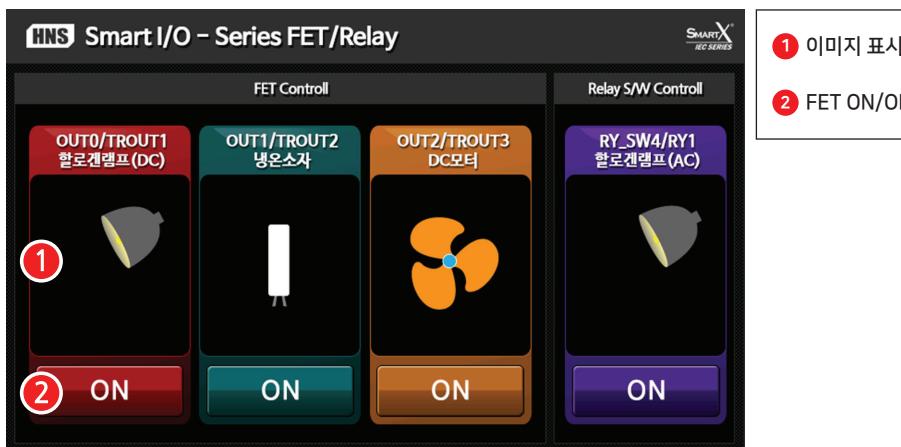
Smart I/O-II Base Board





8-6. 언어별 주요 소스코드 및 응용 예제

SmartX Framework	SmartGPIO
소스파일	TechNote55_SmartIO_Output
소스경로	[홈페이지](http://www.hnsts.co.kr) → [자료실] → [Tech Note] → [55. [C#, VB.NET] Smart I/O-Series FET/Relay 출력 사용 예제]
기능	AC/DC 전원으로 구동하는 장비 연동 시 온/오프 조작
응용분야	DC 할로겐램프, 냉온소자, DC 모터 등 제어
준비사항	<ul style="list-style-type: none"> [PIN0] 할로겐램프 : (DC 12V, 10W) [PIN1] 냉온소자 : (FALC1-00705T150) [PIN2] DC모터 : (주금일모터(KDG37-3429A-050))



[동작설명]

- 프로그램을 시작한 후 ② 각 포트의 ON 버튼을 클릭하면 해당 포트별로 출력신호(LOW→HIGH)가 발생하면 서 ① 이미지가 변경됩니다. 4번째 RY_SW4/RY1 할로겐램프(AC)의 경우에는 Relay 기능의 테스트가 가능합니다.

**SmartX
Framework**

SmartGPIO와 관련된 자세한 사항은 “[SmartX 홈페이지(www.smartx.co.kr)] → [Reference Guide] → [Part 3. 하드웨어 장치 제어 컴포넌트] → [2. SmartGPIO]” 내용을 참조하시기 바랍니다.

< C# 예제 주요소스 코드 >

참고

C#의 주요 소스 코드의 일부 코드가 생략되어 있습니다. 상세한 코드를 확인하기 위해서는 상단 표에 나와 있는 “소스경로”에서 예제 프로그램을 참고하시기 바랍니다.

C# 예시 코드

Form1_Load 이벤트에서 SmartGPIO 초기화

```
private void Form1_Load(object sender, EventArgs e)
{
    // GPIO Port를 A로 설정
    smartGPIO1.PortSelect = SmartX.SmartGPIO.PORTID.PORTA;
    // PortA의 방향을 ALL_OUTPUT으로 설정
    smartGPIO1.PortDirs = 255;
    // PortA의 상태를 ALL_Low로 설정
    smartGPIO1.PortData = 0;
    // …이미지 설정 코드 생략…
}
```

C# 예시 코드

btnDCLampCtrl 버튼이 클릭되면 smartGPIO1.Data0의 상태 값(High/Low)을 변경

```
// FET(TROUT1) DC 할로겐램프 ON/OFF 제어
private void btnDCLampCtrl_Click(object sender, EventArgs e)
{
    // DC 할로겐램프 ON
    if (btnDCLampCtrl.ButtonStatus == SmartX.SmartButton.BUTSTATUS.DOWN)
    {
        smartGPIO1.Data0 = SmartX.SmartGPIO.PinStatus.High;
        pbDCLamp.DrawNext(1);
    }
    // DC 할로겐램프 OFF
    else
    {
        smartGPIO1.Data0 = SmartX.SmartGPIO.PinStatus.Low;
        pbDCLamp.DrawNext(0);
    }
}
```

< VB 예제 주요소스 코드 >

참고

VB.NET 주요 소스 코드에서는 일부 코드가 생략되어 있습니다. 상세한 코드를 확인하기 위해서는 상단 표에 나와 있는 “소스경로”에서 예제 프로그램을 참고하시기 바랍니다.

VB 예시 코드

Form1_Load 이벤트에서 SmartGPIO 초기화

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As EventArgs) Handles MyBase.Load
    ' GPIO Port를 A로 설정
    SmartGPIO1.PortSelect = SmartX.SmartGPIO.PORTID.PORTA
    ' PortA의 방향을 ALL_OUTPUT으로 설정
    SmartGPIO1.PortDirs = 255
    ' PortA의 상태를 ALL_Low로 설정
    SmartGPIO1.PortData = 0
    ' …이미지 설정 코드 생략…
End Sub
```

VB 예시 코드

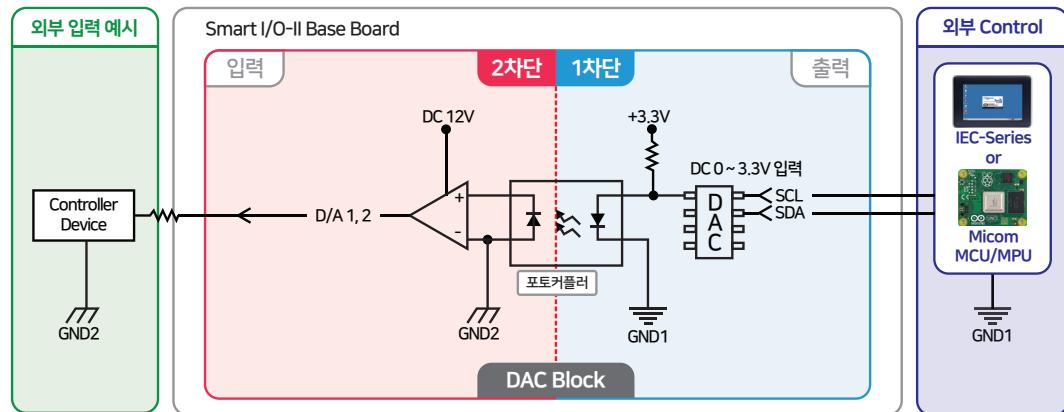
btnDCLampCtrl 버튼이 클릭되면 smartGPIO1.Data0의 상태 값(High/Low)을 변경

```
' FET(TROUT1) DC 할로겐램프 ON/OFF 제어
Private Sub btnDCLampCtrl_Click(ByVal sender As Object, ByVal e As EventArgs) Handles btnDCLampCtrl.Click
    If btnDCLampCtrl.ButtonStatus = SmartX.SmartButton.BUTSTATUS.DOWN Then
        ' DC 할로겐램프 ON
        SmartGPIO1.Data0 = SmartX.SmartGPIO.PinStatus.High
        pbDCLamp.DrawNext(1)
    Else
        ' DC 할로겐램프 OFF
        SmartGPIO1.Data0 = SmartX.SmartGPIO.PinStatus.Low
        pbDCLamp.DrawNext(0)
    End If
End Sub
```

9. DAC Block

9-1. DAC Block 소개

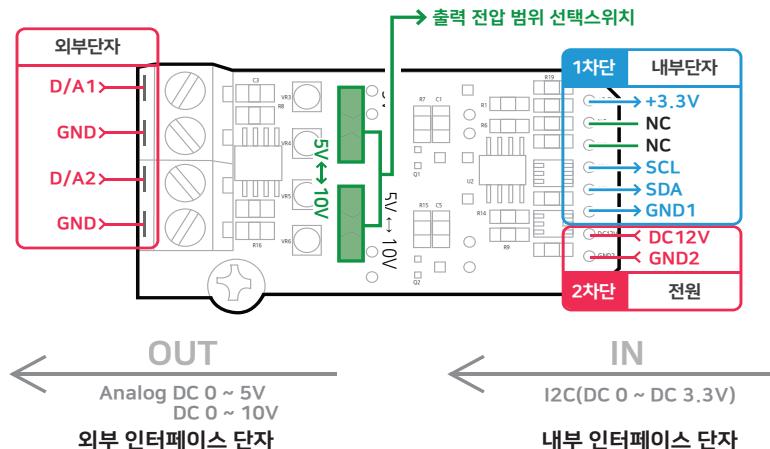
DAC Output은 Analog OptoCoupler를 사용한 것이 특징이며 IIC(I2C)통신을 이용하였습니다. DAC 출력전압은 DC 0~5V 또는 DC 0~10V를 사용할 수 있으며, 12Bit 해상도를 가지고 있습니다. DAC Block의 경우 SmartDAC와 연동하여 사용할 수 있습니다.

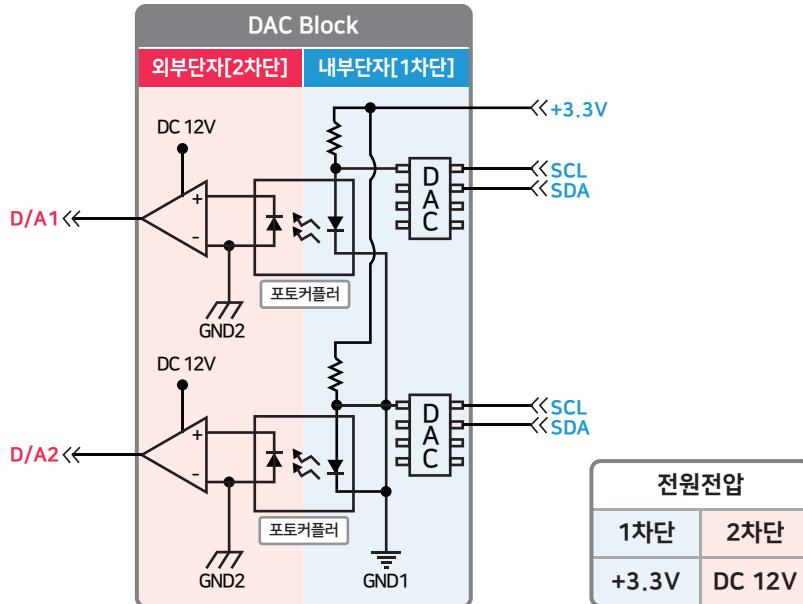


9-2. DAC Block 연결단자 설명

참고

IEC-Series와 Smart I/O-Series를 연동하여 사용할 경우 내부단자(1차단) 정보는 숙지할 필요 없습니다. Base Board의 제작 및 별도의 I/O 제어 Board를 제작할 경우에 참고하시기 바랍니다.





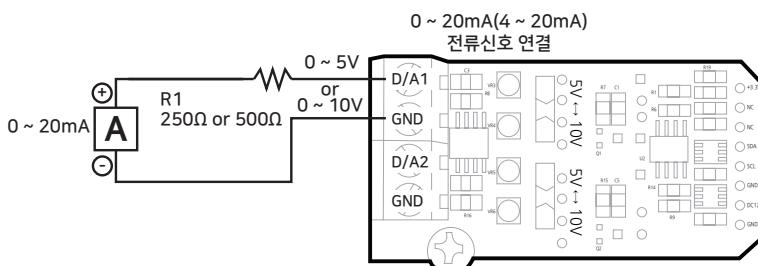
※ 외부 인터페이스 단자

핀 명칭	설명	전기적 사양
D/A1	DAC 전압 출력	DC 0 ~ 5V, DC 0 ~ 10V (12Bit)
GND	D/Ax에 대한 GROUND (외부단자[2차단]의 GND2)	GND
D/A2	DAC 전압 출력	DC 0 ~ 5V, DC 0 ~ 10V (12Bit)
GND	D/Ax에 대한 GROUND (외부단자[2차단]의 GND2)	GND

※ 내부 인터페이스 단자

핀 명칭	설명	전기적 사양
+3.3V	전원 입력 공급(1차단)	DC 3.3V@30mA
GND1	+3.3V에 대한 GROUND (내부단자[1차단]의 GND1)	GND
DC12V	전원 입력 공급(2차단)	DC 12V@5mA
GND2	DC12V에 대한 GROUND (전원[2차단]의 GND2)	GND
NC	-	-
SDA	입력 신호 SDA(Serial Data Line)	DC 0V, DC 3.3V
SCL	입력 신호 SCL(Serial Clock Line)	DC 0V, DC 3.3V

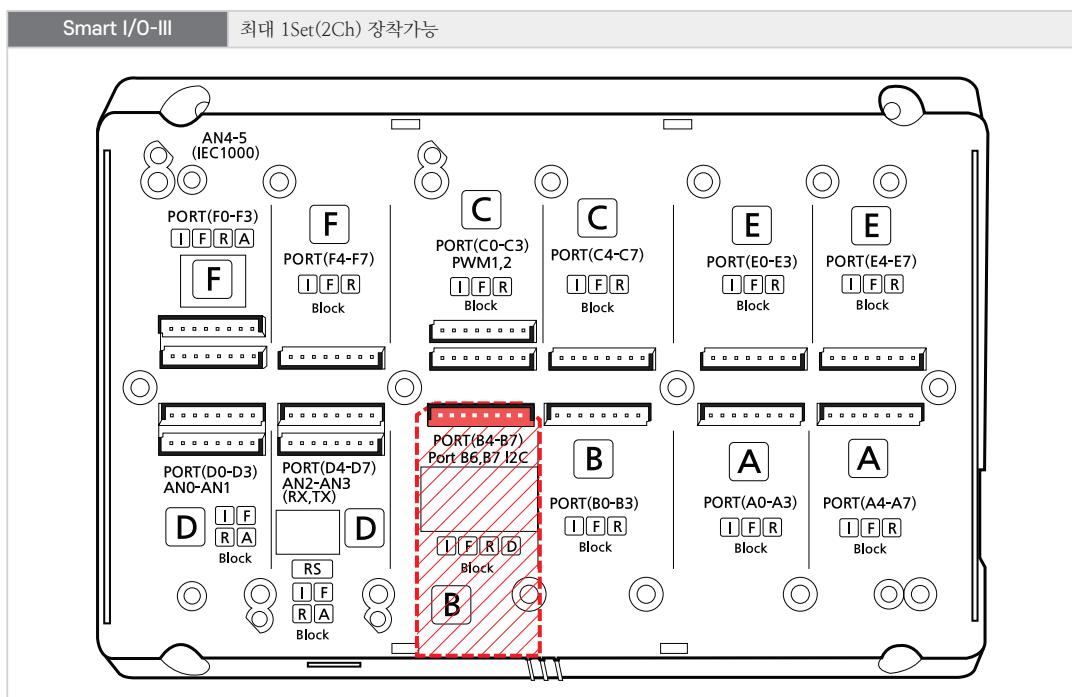
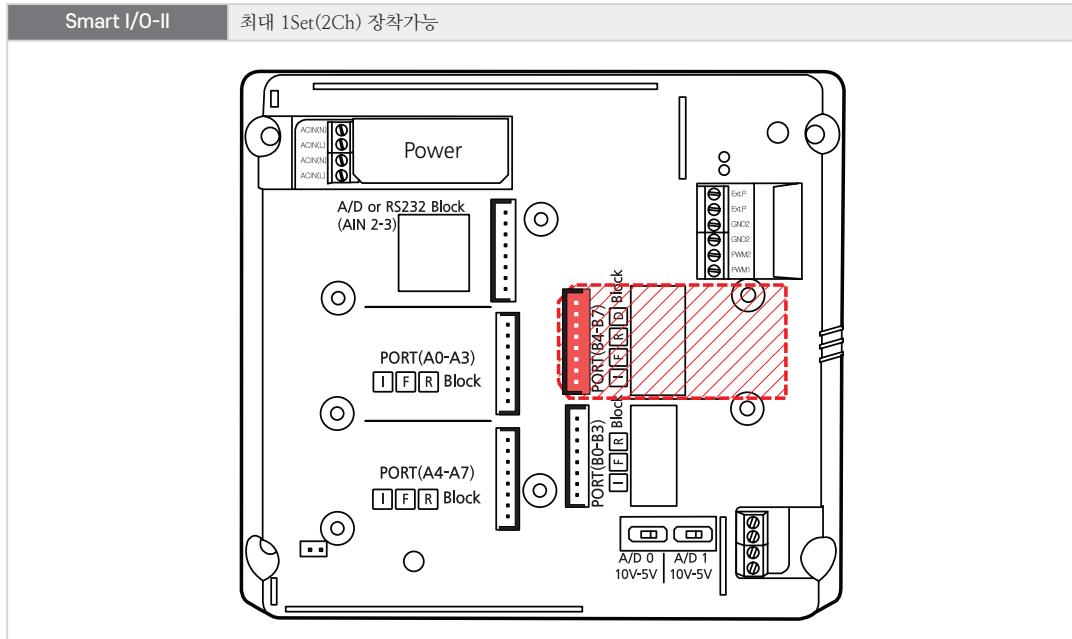
9-3. DAC Block 출력 신호를 전압에서 전류로 변환하여 출력하는 방법



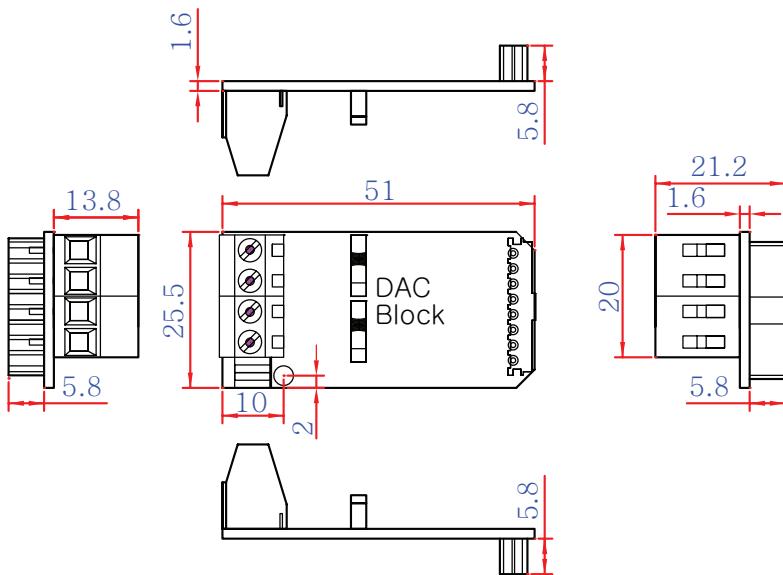
R ₁ 저항 값	출력전압 S/W 범위	계산 값 (I = V / R)
250 Ω	DC 0 ~ 5V	5V / 250 Ω = 20mA
500 Ω	DC 0 ~ 10V	10V / 500 Ω = 20mA

R₁을 직렬로 결선하여 전압 출력을 전류 신호로 변환하여 사용한다.

9-4. DAC Block 장착위치



9-5. DAC Block 외형 및 치수



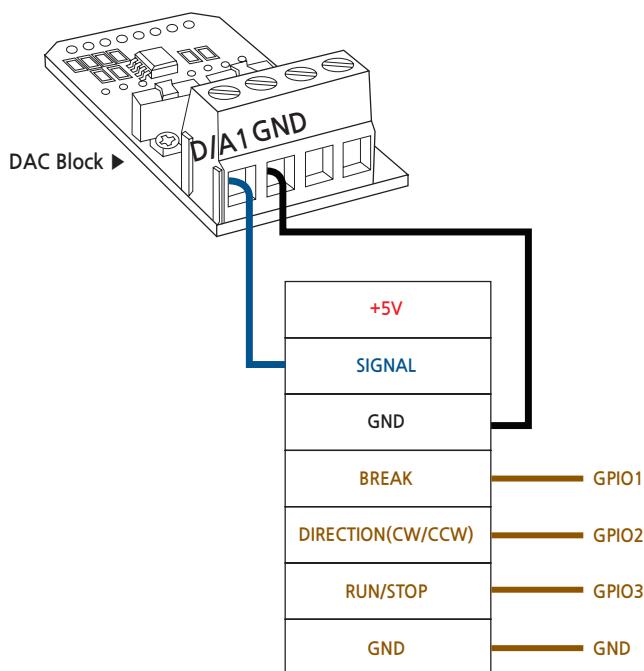
참조

본 제품의 도면은 “[홈페이지(www.hnsts.co.kr)] – [자료실] – [도면 및 승인원]”에서 다운로드할 수 있습니다.

9-6. DAC Block 응용 방법(외부 인터페이스 결선)

1) BLDC 모터 드라이버 연결방법

< BLDC 모터 드라이버 결선도 예시 >



9-7. 언어별 주요 소스코드 및 응용 예제

SmartX Framework	SmartDAC
소스파일	TechNote51_SmartIO123
소스경로	[홈페이지(www.hnsts.co.kr)] → [자료실] → [Tech Note] → [51. [C#, VB.NET] Smart IO - I, II, III_통합 예제]
기능	모터 드라이버 속도 제어
응용분야	전압을 조절하여 모터 드라이버 속도 제어
준비사항	<ul style="list-style-type: none"> BLDC모터드라이버 : (NT-S-BLSM2410) DC모터드라이버 : (TACHYON_60DX_100A) BLDC모터드라이버 : (LBD-V Driver)



- ① 채널 설정
- ② 출력 전압 설정

[동작설명]

- 프로그램이 시작되고 ① 채널을 선택하는 RadioButton을 클릭하여 사용할 채널을 선택
- 채널을 선택한 뒤 ② Up/Down 버튼을 사용하여 출력 전압을 변경

SmartX Framework

SmartDAC와 관련된 자세한 사항은 “[SmartX 홈페이지(www.smartx.co.kr)] → [Reference Guide] → [Part 3. 하드웨어 장치 제어 컴포넌트] → [8. SmartDAC]” 내용을 참조하시기 바랍니다.

< C# 예제 주요소스 코드 >

C# 예제 코드

UpDownButton을 클릭하여 채널에 따라 DAC 전압을 변경 및 출력

```
// DAC 선택된 채널로 현재 설정된 값으로 출력 변경
private void DACValueChange(object sender, EventArgs e)
{
    if (_m_iDACChannel == 1)
    {
        smartDAC1.DAC_Output(SmartX.SmartDAC.DACOUTCHANNEL.DAC_CHANNEL1, (int)udDAC.Value);
    }
    else
    {
        smartDAC1.DAC_Output(SmartX.SmartDAC.DACOUTCHANNEL.DAC_CHANNEL2, (int)udDAC.Value);
    }
}
```

<VB 예제 주요소스 코드>

VB 예시 코드

UpDownButton을 클릭하여 채널에 따라 DAC 전압을 변경 및 출력

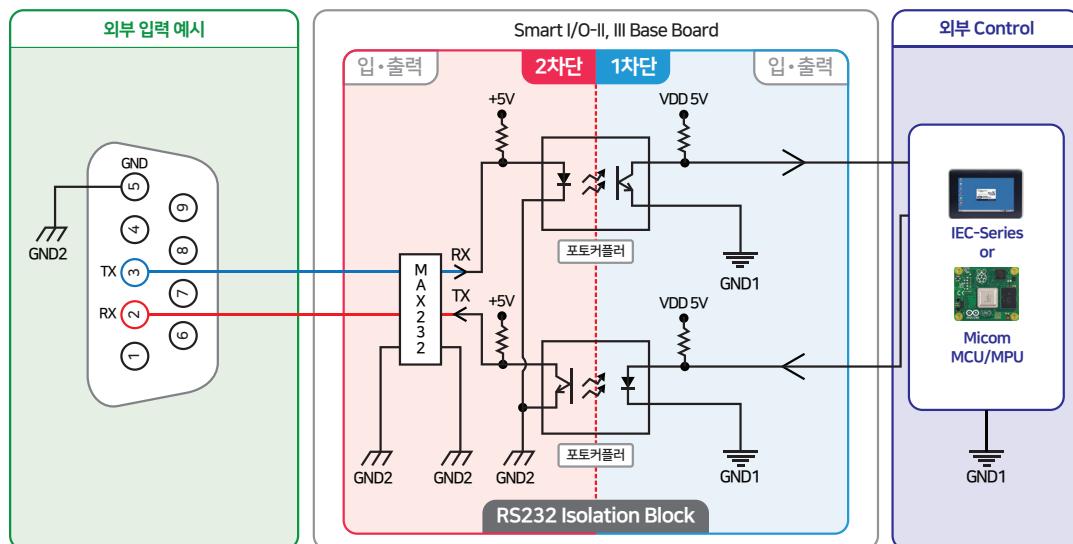
' DAC 선택된 채널로 현재 설정된 값으로 출력 변경

```
Private Sub DACValueChange(ByVal sender As Object, ByVal e As EventArgs) Handles udDAC.OnUpButtonClick,  
udDAC.OnDownButtonClick  
    If m_iDACChannel = 1 Then  
        SmartDAC1.DAC_Output(SmartX.SmartDAC.DACOUTCHANNEL.DAC_CHANNEL1, CInt(udDAC.Value))  
    Else  
        SmartDAC1.DAC_Output(SmartX.SmartDAC.DACOUTCHANNEL.DAC_CHANNEL2, CInt(udDAC.Value))  
    End If  
End Sub
```

10. RS232 Isolation Block

10-1. RS232 Isolation Block 소개

RS232 Isolation Block으로 통신할 경우 IEC667/1000-Series 모두 COM4(TTL Level)로 연결되며, 최대 19200bps까지 지원됩니다. RS232 Isolation Block의 경우 SmartSerialPort와 연동하여 사용할 수 있습니다.

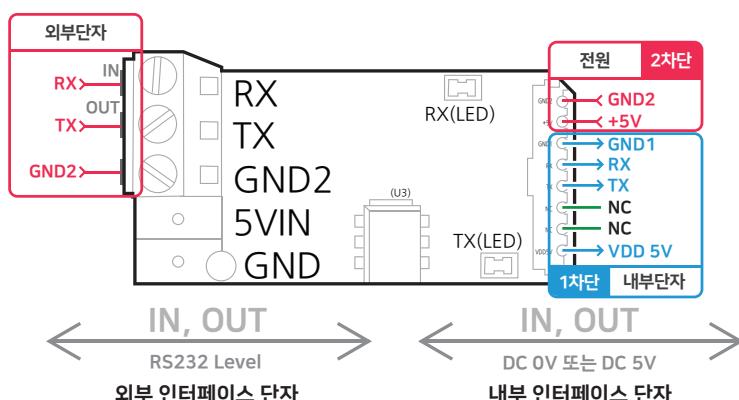

주의

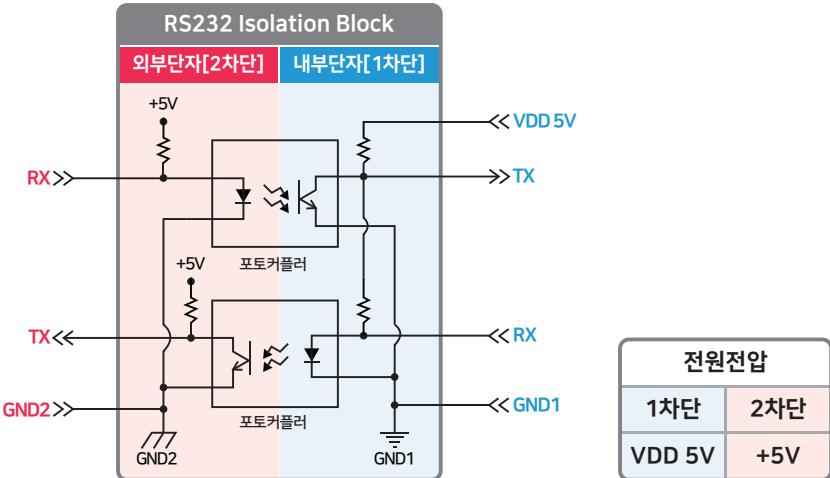
Serial 통신에서 COM포트 사용 시 흐름제어 신호선은 지원하지 않습니다.

10-2. RS232 Isolation Block 연결단자 설명

참고

IEC-Series와 Smart I/O-Series를 연동하여 사용할 경우 내부단자(1차단) 정보는 숙지할 필요 없습니다. Base Board의 제작 및 별도의 I/O 제어 Board를 제작할 경우에 참고하시기 바랍니다.





※ 외부 인터페이스 단자

핀 명칭	설명	전기적 사양
RX	통신의 RX(수신)	
TX	통신의 TX(송신)	DC $\pm 15V$
GND2	통신에 대한 GROUND (외부단자[2차단]의 GND2)	GND

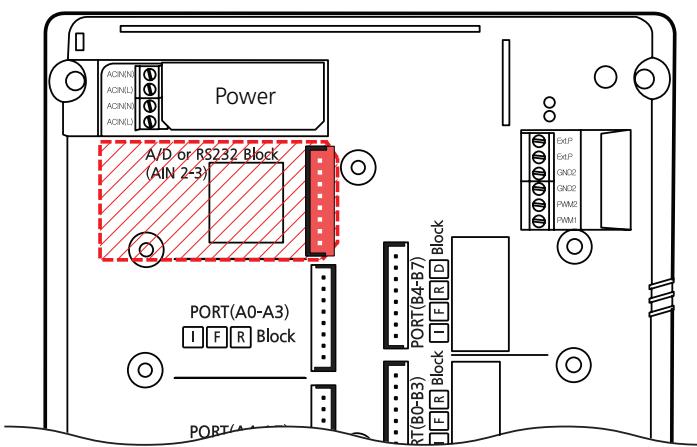
※ 내부 인터페이스 단자

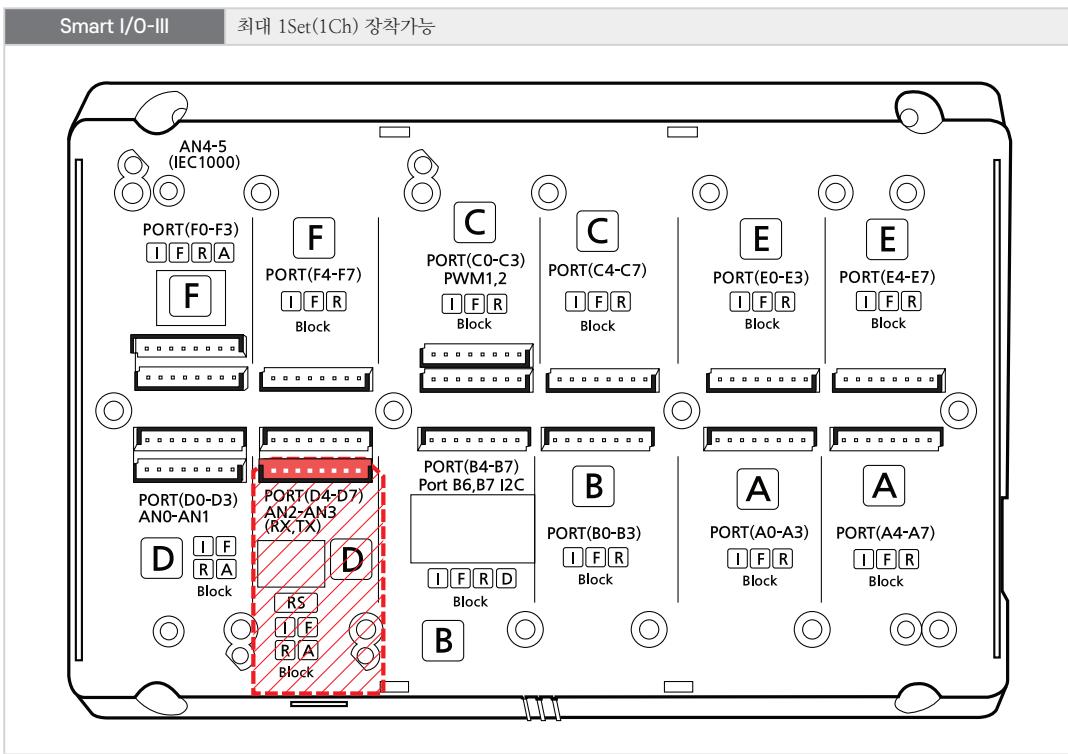
핀 명칭	설명	전기적 사양
+5V	DC 5V 전원 입력(2차단)	DC 5V@40mA
GND2	+5V에 대한 GROUND (전원[2차단]의 GND2)	GND
VDD 5V	전원 입력 공급(1차단)	DC 5V@30mA
GND1	VDD 5V에 대한 GROUND (내부단자[1차단]의 GND1)	GND
RX	통신의 RX(수신) TTL 5V	
TX	통신의 TX(송신) TTL 5V	DC 0V or 5V
NC	-	-

10-3. RS232 Isolation Block 장착위치

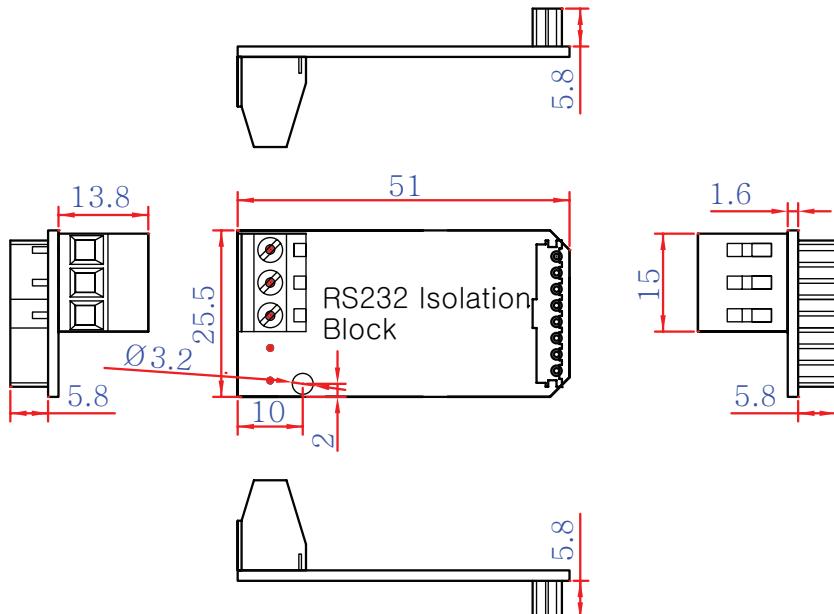
Smart I/O-II

최대 1Set(1Ch) 장착가능





10-4. RS232 Isolation Block 외형 및 치수

**참조**

본 제품의 도면은 “[홈페이지](www.hnsts.co.kr)] – [자료실] – [도면 및 승인원]”에서 다운로드할 수 있습니다.

10-5. RS232 Isolation Block 응용 방법(외부 인터페이스 결선)

1) RS232(SerialPort) 연결방법

SerialPort를 아래와 같이 결선하시기 바랍니다.

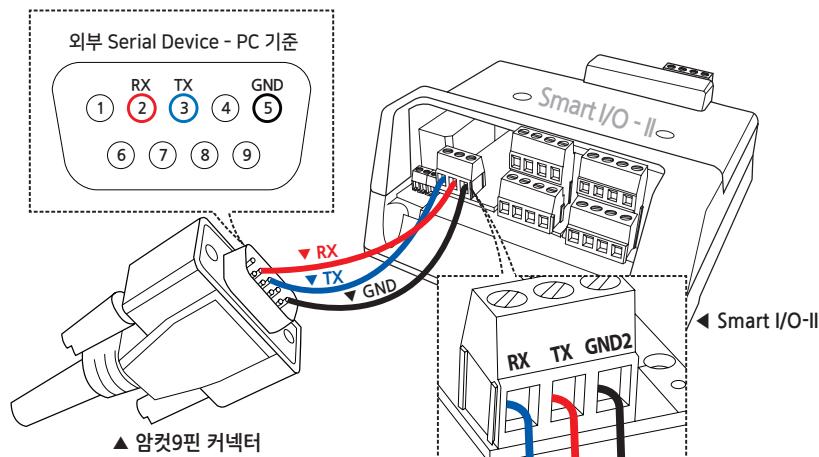
참고

아래의 RS232 결선도의 경우 “외부 Serial Device”는 PC를 기준으로 만들었습니다. 사용하는 Device에 따라 Rx, Tx, GND의 위치가 다를 수 있으므로 각 Device의 핀맵을 참고하시기 바랍니다.

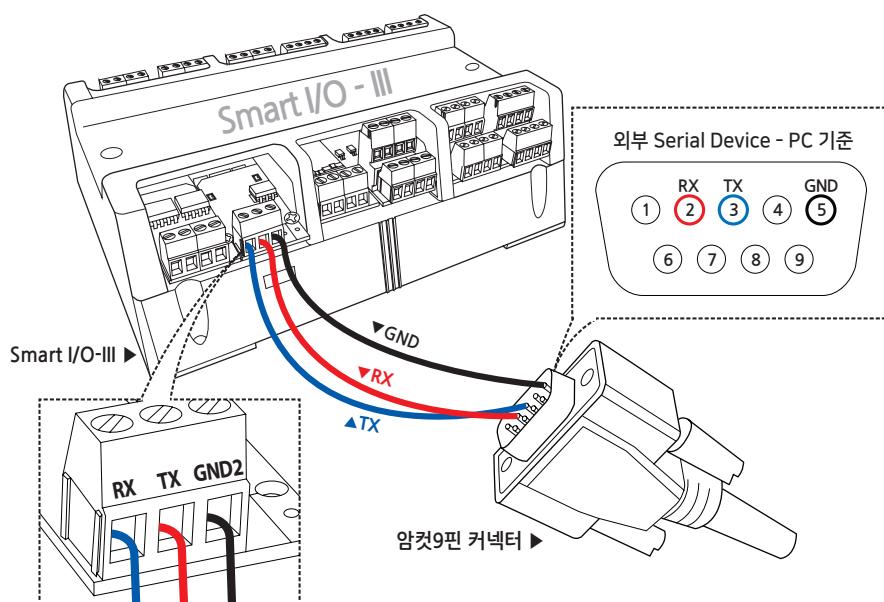
※ 각 Device의 Rx와 Tx는 서로 교차되도록(Rx ← Tx, Tx → Rx) 연결해야 합니다.

< Smart I/O-II, III Base Board 결선도 예시 >

Smart I/O-II Base Board

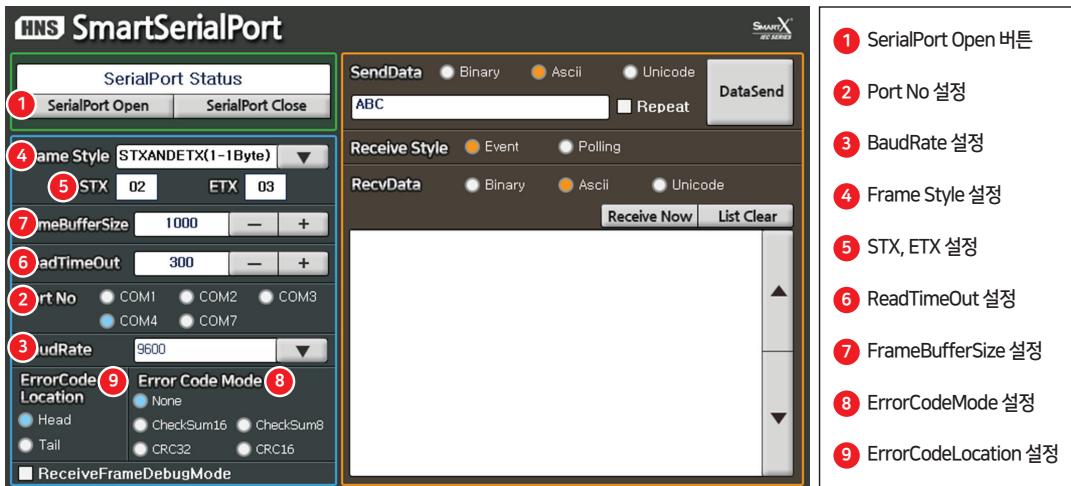


Smart I/O-III Base Board



10-6. 언어별 주요 소스코드 및 응용 예제

SmartX Framework	SmartSerialPort
소스파일	TechNote56_SmartIO_RS232
소스경로	[홈페이지](www.hnsts.co.kr) → [자료실] → [Tech Note] → [56. [C#, VB.NET] Smart I/O – Series RS232 – Block 사용 예제]
기능	시리얼(232, 485) 통신 기능
응용분야	시리얼 통신 분야
준비사항	<ul style="list-style-type: none"> RS232 연장 케이블 시리얼 9핀 커넥터(암놈)



[동작설명]

- 프로그램을 시작한 후 아래의 표와 같이 필요한 부분의 값을 설정합니다.

②	Port No	포트번호 COM1 ~ COM4 설정 가능
③	BaudRate	통신속도 110bps ~ 115200bps 설정 가능(RS232 Block의 경우 최대 19200bps까지 지원)
④	Frame Style	STX, ETX, ReadTimeOut 여부에 따라 Frame Style을 선택
⑤	STX, ETX	STX, ETX를 사용하는 Frame Style을 사용하는 경우 설정
⑥	ReadTimeOut	ReadTimeOut을 사용하는 Frame Style을 사용하는 경우 설정
⑦	FrameBufferSize	한 Frame의 최대 크기를 설정, 통신 시 발생할 수 있는 최대 크기로 설정
⑧	Error Code Mode	CheckSum, CRC Error Check 기능을 사용하고자 하는 경우 설정
⑨	Error Code Location	Error Check 기능을 사용하는 경우 Error Check Code의 위치를 설정

- 설정을 마쳤다면 ① “SerialPort Open” 버튼을 클릭하여 포트를 오픈합니다.
- Data 형식(Binary, Ascii, Unicode)에 맞춰 Send 혹은 Receive를 진행합니다.

**SmartX
Framework**

SmartSerialPort와 관련된 자세한 사항은 “[SmartX 홈페이지](www.smartx.co.kr) → [Reference Guide] → [Part 3. 하드웨어 장치 제어 컴포넌트] → [3. SmartSerialPort]” 내용을 참조하시기 바랍니다.

< C# 예제 주요소스 코드 >

참고

C#의 주요 소스 코드의 일부 코드가 생략되어 있습니다. 상세한 코드를 확인하시기 위해서는 상단 표에 나와있는 “소스경로”를 참고하여 예제 프로그램을 참고하시기 바랍니다.

C# 예시 코드

SerialPort 환경 설정 및 포트 Open

```
// 송/수신 데이터의 구조가 Frame과 NoneFrame에 따라 FrameSeparationType을 구분하여 설정
private void btnOpen_Click(object sender, EventArgs e)
{
    // 선택한 인덱스를 저장
    int iFrameSelect = cbFrameStyle.SelectedItemIndex;

    // smartSerialPort가 Open되지 않은 경우
    if (smartSerialPort1.IsOpen == false)
    {
        // STX, ETX를 저장할 바이트 배열
        byte[] bSTX = new byte[2];
        byte[] bETX = new byte[2];

        // SmartSerialPort STX, ETX 초기화
        smartSerialPort1.STXCodes = null;
        smartSerialPort1.ETXCodes = null;

        // STX, ETX를 변환하여 배열에 저장
        bSTX[0] = Convert.ToByte(labSTX1.Text);
        bSTX[1] = Convert.ToByte(labSTX2.Text);
        bETX[0] = Convert.ToByte(labETX1.Text);
        bETX[1] = Convert.ToByte(labETX2.Text);
        switch (iFrameSelect)
        {
            case 0:
            case 4:
                // STX(1Byte), ETX(1Byte)를 사용하는 경우
                smartSerialPort1.STXCode = bSTX[0];
                smartSerialPort1.ETXCode = bETX[0];
                break;
            case 1:
            case 5:
                // STX(2Byte), ETX(1Byte)를 사용하는 경우
                smartSerialPort1.STXCodes = bSTX;
                smartSerialPort1.ETXCode = bETX[0];
                break;
            case 6:
                // ... 중략 ...
        }

        // SmartComboBox로 선택한 Item에 따라 Baud_Rate 설정
        smartSerialPort1.Baud_Rate = (SmartX.SmartSerialPort.BAUDRATE)(Convert.ToInt32(cbBaudRate.Text));

        // Port번호 선택. RS232블럭을 사용하는 경우 IEC667/1000-Series은 COM4 포트만 사용 가능합니다.
        if (radCOM1.Checked == true)
        {
            smartSerialPort1.PortNo = SmartX.SmartSerialPort.COMPORTNO.COM1;
        }
        else if (radCOM2.Checked == true)
        {
            smartSerialPort1.PortNo = SmartX.SmartSerialPort.COMPORTNO.COM2;
        }
    }
}
```

```
{  
    smartSerialPort1.PortNo = SmartX.SmartSerialPort.COMPORTNO.COM2;  
}  
// … 중략 …  
if (radCOM4.Checked == false)  
{  
    SmartX.SmartMessageBox.Show("RS232 Block을 사용하는 경우 IEC667/1000은 COM4 포트만 사용할 수 있습니다.");  
}  
  
// Error Check Mode  
if (radNone.Checked == true)  
{  
    smartSerialPort1.ErrorCheckMode = SmartX.SmartSerialPort.ERRORCHECK.NONE;  
}  
else if (radCheckSum8.Checked == true)  
{  
    smartSerialPort1.ErrorCheckMode = SmartX.SmartSerialPort.ERRORCHECK.CHECKSUM8;  
}  
// … 중략 …  
// ErrorCodeLocation  
if (radHead.Checked == true)  
{  
    smartSerialPort1.ErrorCode_Location = SmartX.SmartSerialPort.ERRORCODELOCATION.HEADER;  
}  
else  
{  
    smartSerialPort1.ErrorCode_Location = SmartX.SmartSerialPort.ERRORCODELOCATION.TAIL;  
}  
if (radEvent.Checked == true)  
{  
    smartSerialPort1.ReceiveDetect = SmartX.SmartSerialPort.RECEIVEDETECTTYPE.EVENT_QUEUE;  
}  
else  
{  
    smartSerialPort1.ReceiveDetect = SmartX.SmartSerialPort.RECEIVEDETECTTYPE.POLLING_NONEQUEUE;  
}  
  
// ReadTimeout 설정 (ReadTimeout을 사용하는 FrameType의 경우 반드시 설정해야 합니다.)  
smartSerialPort1.ReadTimeout = Convert.ToInt32(udReadTimeOut.Value);  
// Frame의 BufferSize를 설정  
smartSerialPort1.FrameBufferSize = Convert.ToInt32(udBufferSize.Value);  
// ReceiveFrameDebugMode 설정  
// (True는 FrameData[STX+Data+ErrorCheckCode+ETX] 송/수신, False는 Data만 송/수신)  
smartSerialPort1.ReceiveFrameDebugMode = chkDebugMode.Checked;  
smartSerialPort1.RS485SoftwareDetection = false;  
// Port가 열려있지 않다면 Port를 Open합니다.  
if (smartSerialPort1.IsOpen == false)  
{  
    smartSerialPort1.Open();  
    labStatus.Text = "SerialPort Opened";  
}  
}  
}
```

C# 예시 코드

데이터 송신

```
// Data를 Send하기 위한 함수
private void SendData()
{
    // Frame Style에 따른 Send Method 분류 [1. Frame-Event / 2. Frame-Polling]
    if (smartSerialPort1.FrameSeparationType == SmartX.SmartSerialPort.FRAMESEPARATIONTYPES.NONEFRAME_READ_TIMEOUT)
    {
        if (radSendAscii.Checked == true)
        {
            smartSerialPort1.WriteNoneFrame(labSendData.Text, SmartX.SmartSerialPort.CODETYPES.ASCIICODE);
        }
        else if (radSendUnicode.Checked == true)
        {
            smartSerialPort1.WriteNoneFrame(labSendData.Text, SmartX.SmartSerialPort.CODETYPES.UNICODE);
        }
        else
        {
            byte[] bByteArray = Encoding.UTF8.GetBytes(labSendData.Text);
            smartSerialPort1.WriteNoneFrame(bByteArray);
        }
    }
    else
    {
        if (radSendAscii.Checked == true)
        {
            smartSerialPort1.WriteFrame(labSendData.Text, SmartX.SmartSerialPort.CODETYPES.ASCIICODE);
        }
        else if (radSendUnicode.Checked == true)
        {
            smartSerialPort1.WriteFrame(labSendData.Text, SmartX.SmartSerialPort.CODETYPES.UNICODE);
        }
        else
        {
            // 문자열 --> Byte[] 변환후 전송
            byte[] bByteArray = Encoding.UTF8.GetBytes(labSendData.Text);
            smartSerialPort1.WriteFrame(bByteArray);
        }
    }
}
}
```

C# 예시 코드

데이터 수신

```
// Event 수신 방식
private void smartSerialPort1_OnReadQueueEvent()
{
    string strData = " ";
    byte[] readByte;
    SmartX.SmartSerialPort.FRAMEDATAREADSTATUS eStatus;

    // 1. Frmae, 2. NoneFrame
    eStatus = smartSerialPort1.ReadQueue(out readByte);
    // ASCII, UniCode, Binary 방식에 따라 구분
    if (radRecvAscii.Checked == true)
```

```

{
    strData = SmartX.SmartSerialPort.ConvertAsciiByteToString(readByte);
}
else if (radRecvUnicode.Checked == true)
{
    strData = SmartX.SmartSerialPort.ConvertUnicodeByteToString(readByte);
}
else
{
    // 수신된 프레임 데이터를 각각의 바이트를 16진수로 표시 합니다
    for (int i = 0; i < readByte.Length; i++)
    {
        strData += string.Format(" {0:X2} ", readByte[i]);
    }
}
// ListBox에 데이터 삽입
if (eStatus == SmartX.SmartSerialPort.FRAMEDATAREADSTATUS.EMPTY)
{
    lstRecvData.AddItem(" EMPTY ");
}
else if (eStatus == SmartX.SmartSerialPort.FRAMEDATAREADSTATUS.FAILDATA)
{
    lstRecvData.AddItem(" Fail ");
}
else
{
    lstRecvData.AddItem(strData);
}
}
}

```

< VB 예제 주요소스 코드 >

참고

VB.NET 주요 소스 코드에서는 일부 코드가 생략되어 있습니다. 상세한 코드를 확인하기 위해서는 상단 표에 나와있는 “소스경로”에서 예제 프로그램을 참고하시기 바랍니다.

VB 예시 코드

SerialPort 환경 설정 및 포트 Open

```

' 송/수신 데이터의 구조가 Frame과 NoneFrame에 따라 FrameSeparationType을 구분하여 설정
Private Sub btnOpen_Click(ByVal sender As Object, ByVal e As EventArgs) Handles btnOpen.Click
    ' 선택한 인덱스를 저장
    Dim iFrameSelect As Integer = cbFrameStyle.SelectedIndex
    ' smartSerialPort가 Open되지 않은 경우
    If SmartSerialPort1.IsOpen = False Then
        ' …SmartSerialPort의 통신 설정 관련 속성중 일부는 VB에서 생략되었습니다. …
        ' Port번호 선택. RS232블럭을 사용하는 경우 IEC667/1000-Series은 COM4 포트만 사용 가능합니다.
        If radCOM1.Checked = True Then
            SmartSerialPort1.PortNo = SmartX.SmartSerialPort.COMPORTNO.COM1
        End If

        If radEvent.Checked = True Then
            SmartSerialPort1.ReceiveDetect = SmartX.SmartSerialPort.RECEIVEDECTTYPE.EVENT_QUEUE
        Else
            SmartSerialPort1.ReceiveDetect = SmartX.SmartSerialPort.RECEIVEDECTTYPE.POLLING_NONEQUEUE
        End If
    End If

```

```

‘ ReadTimeout 설정 (ReadTimeout을 사용하는 FrameType의 경우 반드시 설정해야 합니다.)  

SmartSerialPort1.ReadTimeout = Convert.ToInt32(udReadTimeOut.Value)

‘ Port가 열려있지 않다면 Port를 Open합니다.  

If SmartSerialPort1.IsOpen = False Then  

    SmartSerialPort1.Open()  

    labStatus.Text = “SerialPort Opened”  

End If  

End If  

End Sub

```

VB 예시 코드 데이터 송신

```

Private Sub SendData()  

    ‘ Frame Style에 따른 Send Method 분류  

    ‘ 1. Frame-Event / 2. Frame-Polling  

If SmartSerialPort1.FrameSeparationType = SmartX.SmartSerialPort.FRAMESEPARATIONTYPES.NONEFRAME_READ  

TIMEOUT Then  

    If radSendAscii.Checked = True Then  

        SmartSerialPort1.WriteNoneFrame(labSendData.Text, SmartX.SmartSerialPort.CODETYPES.ASCIICODE)  

    ElseIf radSendUnicode.Checked = True Then  

        SmartSerialPort1.WriteNoneFrame(labSendData.Text, SmartX.SmartSerialPort.CODETYPES.UNICODE)  

    Else  

        Dim bByteArray As Byte() = Encoding.UTF8.GetBytes(labSendData.Text)  

        SmartSerialPort1.WriteNoneFrame(bByteArray)  

    End If  

Else  

    ‘ …SmartSerialPort의 Frame Style의 송신 방식의 코드는 VB에서 생략되었습니다. …  

End If  

End Sub

```

VB 예시 코드 데이터 수신

```

Private Sub smartSerialPort1_OnReadQueueEvent() Handles SmartSerialPort1.OnReadQueueEvent  

Dim strData As String = “ ”  

Dim readByte As Byte()  

Dim eStatus As SmartX.SmartSerialPort.FRAMEDATAREADSTATUS

eStatus = SmartSerialPort1.ReadQueue(readByte)

‘ ASCII, UniCode, Binary 방식에 따라 구분  

If radRecvAscii.Checked = True Then  

    strData = SmartX.SmartSerialPort.ConvertAsciiByteToString(readByte)  

ElseIf radRecvUnicode.Checked = True Then  

    strData = SmartX.SmartSerialPort.ConvertUnicodeByteToString(readByte)  

Else  

    ‘ 수신된 프레임 데이터를 각각의 바이트를 16진수로 표시 합니다  

    For i As Integer = 0 To readByte.Length - 1  

        strData += String.Format(“ {0:X2} ”, readByte(i))  

    Next  

End If

If eStatus = SmartX.SmartSerialPort.FRAMEDATAREADSTATUS.VALIDDATA Then  

    lstRecvData.AddItem(strData)
    ‘ …수신 실패 처리 코드는 VB에서 생략되었습니다. …  

End If  

End Sub

```

Smart I/O-Series 제품 매뉴얼 (Smart I/O-II, III)

본 저작권은 (주)에이치앤에스에 있습니다.
제품 및 자세한 문의 사항은 아래의 연락처 및 메일로 문의해 주시기 바랍니다.

(주)에이치앤에스

서울특별시 금천구 가산디지털1로 181, 1505호 (가산동, 가산더블유센터)

대표전화: 02-6402-8001 / 팩스: 02-6442-9775

부서안내	연락처	직통 전화	이메일
제품 구매 및 견적문의	02-6402-8001(내선 1번)	070-7094-5770	sales@hnsts.co.kr
하드웨어 기술문의	02-6402-8001(내선 2번)	070-7094-5001	app@smartx.co.kr
소프트웨어 기술문의	02-6402-8001(내선 3번)	070-7094-5002	app@smartx.co.kr
제품서비스 기술문의	02-6402-8001(내선 4번)	070-7094-5003	tech@smartx.co.kr

홈페이지: www.hnsts.co.kr / 쇼핑몰(제품구매): www.hnsstore.co.kr

SmartX Framework 사이트: www.smartx.co.kr