

제품 소개



PES-2401

4포트 릴레이 출력보드인 PES-2401은 PHPoC 보드형 제품 전용 스마트 확장보드중 하나입니다. 이 보드를 이용하면 접점 제어를 통해 외부기기를 ON/OFF 할 때 사용할 수 있습니다.

PES-2401의 주요 특징

- 4개의 NO(Normal Open) 릴레이
- 출력포트 인터페이스: 8핀 터미널블록(S type, T type)
- 출력포트 최대 허용 전압: DC 30[V]
- 출력포트 최대 허용 전류: 2[A]
- 소비 전류: 약 160[mA]

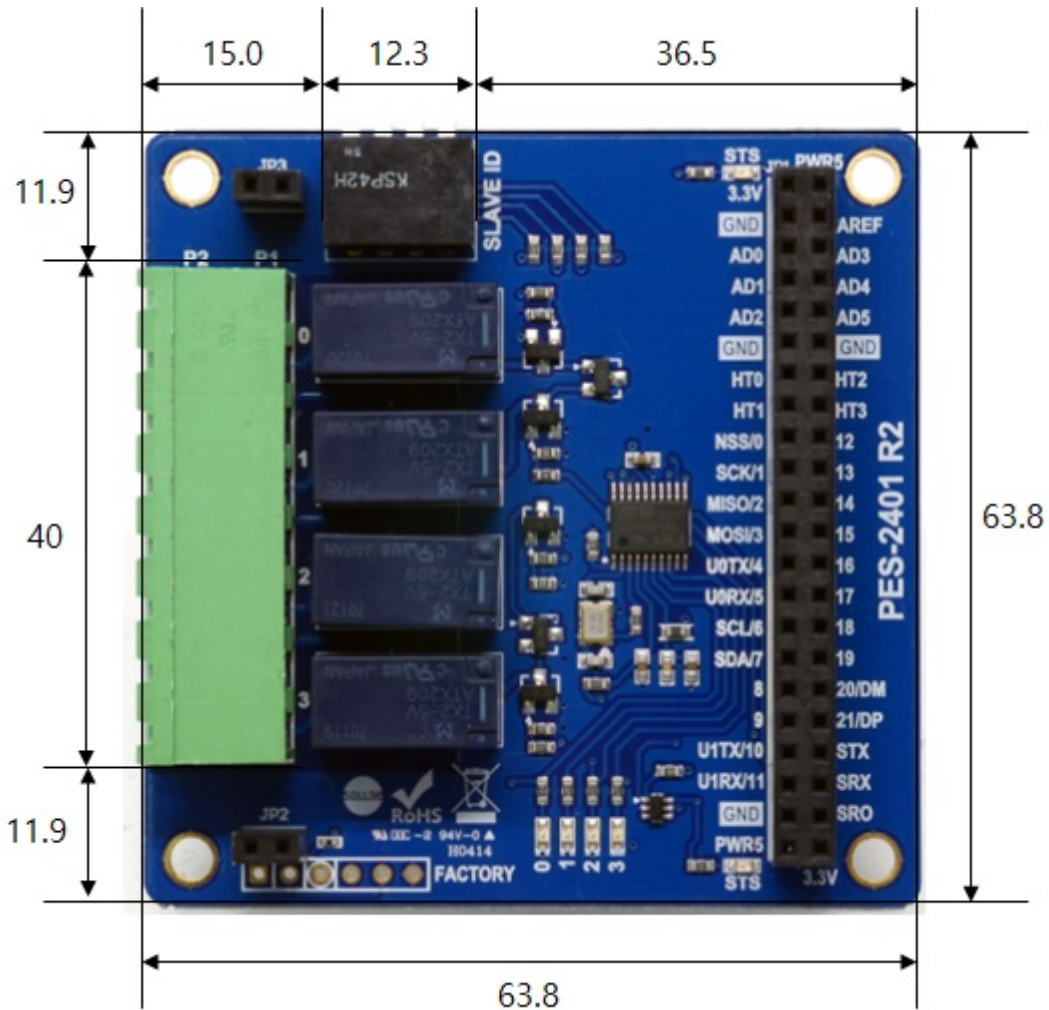
※ 주의: PES-2401을 사용하기 위해서는 반드시 PHPoC 보드가 필요합니다!

스마트 확장보드란?

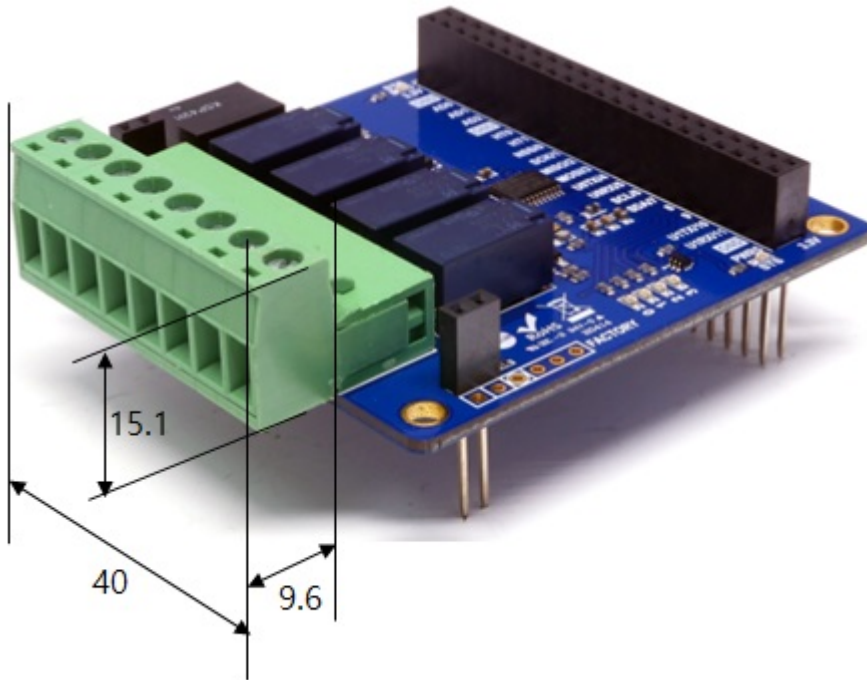
스마트 확장보드는 일반 확장보드와는 달리 PHPoC보드의 디바이스 및 펌웨어와는 독립적인 자체 디바이스와 전용 펌웨어를 내장하고 있습니다. 이 보드는 PHPoC 보드와 전용 통신 포트를 이용해 마스터-슬레이브 방식으로 통신합니다. 하나의 PHPoC 보드에 여러개의 스마트 확장보드를 연결할 수 있으며 각각의 스마트 확장보드에는 반드시 슬레이브 아이디를 설정해야 합니다.

치수

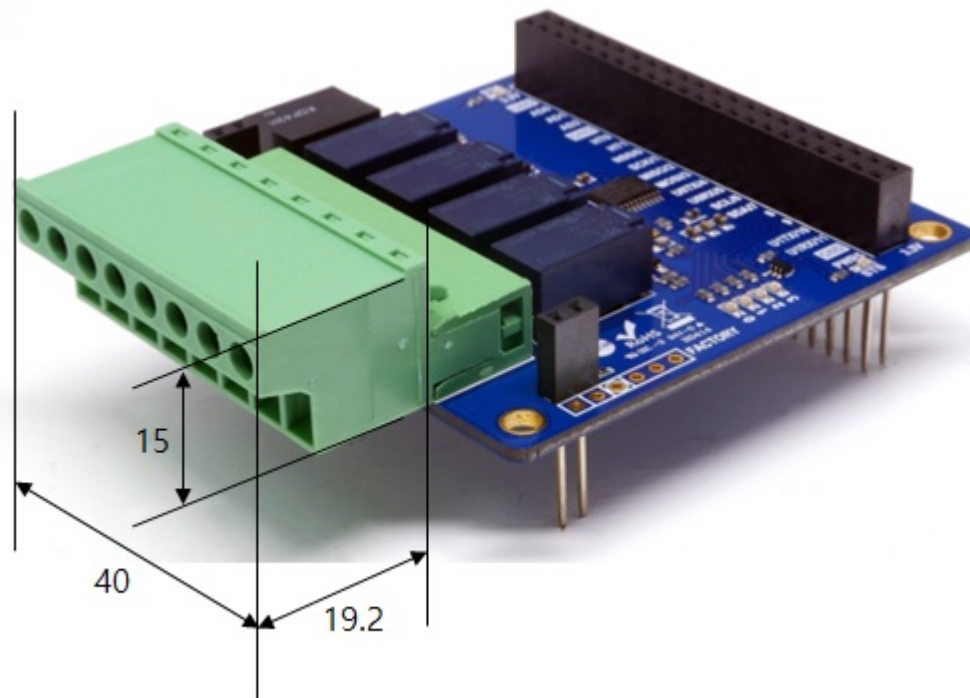
제품 본체



터미널블록(T형)



터미널블록(S형)



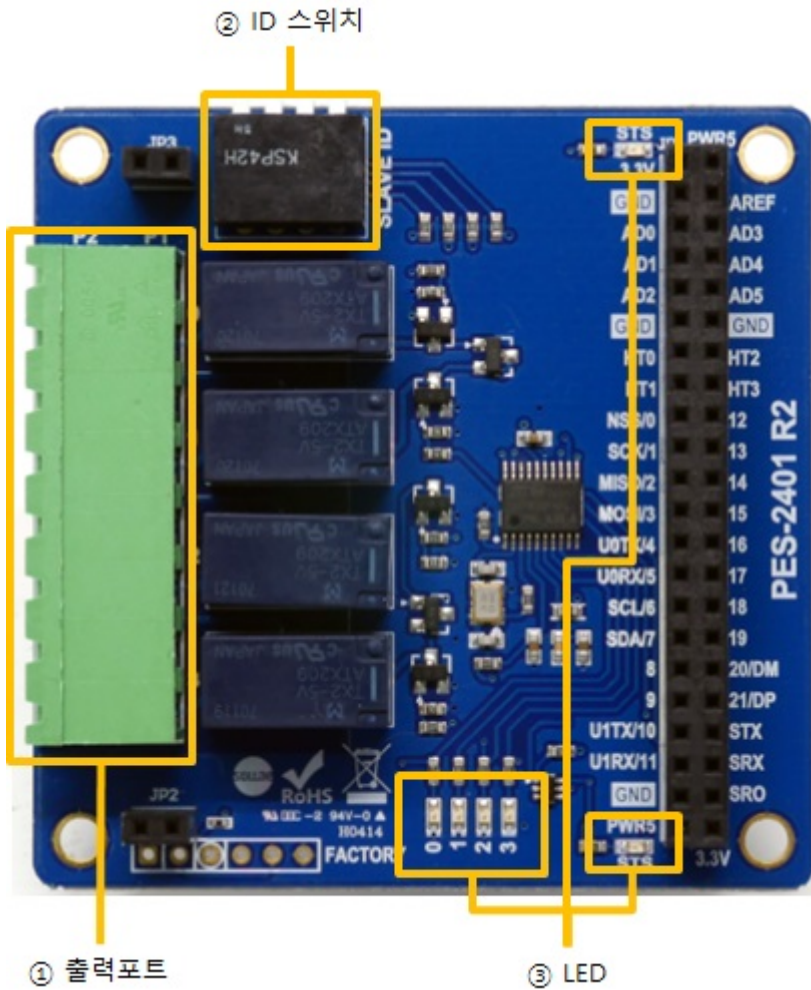
※ 치수(단위 : mm)는 제품 상태 및 재는 각도 등에 따라 약간의 오차가 있을 수 있습니다.

회로도

PES-2401의 회로도입니다.

- [PES-2401-R2-PO.pdf](#)

레이아웃



1. 출력포트

출력포트는 5mm간격의 1 by 8 터미널 블록으로 되어 있습니다. 각 포트는 NO(Normal Open) 형태의 릴레이로 되어 있습니다.

※ Normal Open: 출력포트의 기본 상태가 OFF인 것을 의미합니다.

출력포트의 전압조건에 따른 허용 전류는 다음과 같습니다.

전압조건(DC)	최대허용전류
30V	2A

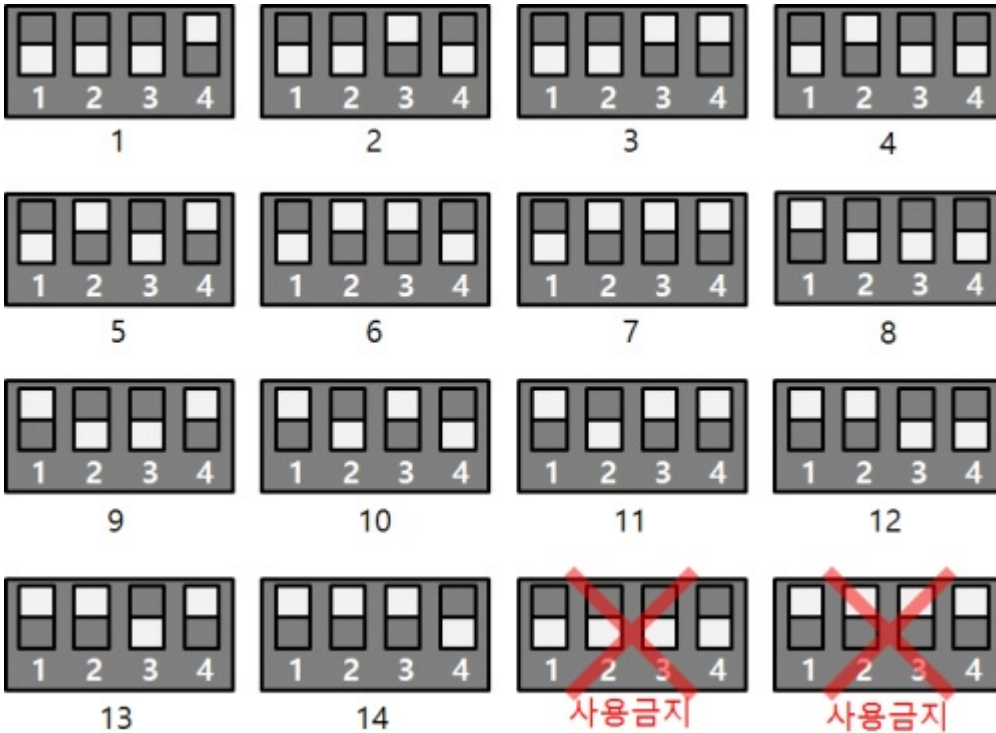
※ 주의: 최대허용전류의 범위를 벗어난 사용은 제품 고장을 초래할 수 있습니다. 반드시 연결되는 장비의 순간전류를 고려하여 사용하시기 바랍니다.

※ 참고: 하나의 PHPoC 보드에 연결할 수 있는 PES-2401의 최대 개수는 4개입니다.

2. 슬레이브 아이디 스위치

슬레이브 아이디는 PHPoC 보드가 스마트 확장보드 각각을 구분하는데 사용됩니다. 따라서 PHPoC 보드

에 연결되는 각 스마트 확장보드는 고유한 슬레이브 아이디를 사용해야 합니다. 슬레이브 아이디는 1부터 14까지 14개 중 하나로 설정할 수 있으며 다음과 같이 4개의 DIP스위치를 조정하여 설정합니다.



3. LED

STS LED

PES-2401 보드에는 2개의 STS LED가 있습니다. JP1 상단의 STS LED는 3.3V에, 하단의 STS LED는 5V와 연결되어 있습니다. 두 LED의 동작은 서로 같으며 다음과 같습니다.

상태	LED 동작
정상	1초마다 켜짐/꺼짐 반복
슬레이브 아이디 설정 이상	매우 빠르게 깜빡임
PHPoC보드와 통신 불가능	꺼짐

출력포트 LED

PES-2401 보드에는 4개의 출력포트 LED가 있으며 동작은 다음과 같습니다.

LED	설명
0	0번 릴레이 출력이 ON일 때 켜짐
1	1번 릴레이 출력이 ON일 때 켜짐
2	2번 릴레이 출력이 ON일 때 켜짐
3	3번 릴레이 출력이 ON일 때 켜짐

사용하기

PES-2401을 사용하는 방법은 다음과 같습니다.

1. PHPoC 보드에 연결

PES-2401은 단독으로 사용할 수 없습니다. 반드시 PHPoC 보드에 연결하여 사용하기 바랍니다.

2. 소프트웨어(IDE) 설치

PHPoC 디버거는 PHPoC 제품의 설정 및 개발에 사용되는 소프트웨어입니다. PES-2401은 PHPoC 보드형 제품을 통해 제어할 수 있으므로 이 보드를 사용하기 위해서는 PC에 PHPoC 디버거를 반드시 설치해야 합니다.

- [PHPoC 디버거 다운로드 페이지](#)
- [PHPoC 디버거 매뉴얼 페이지](#)

3. SPC라이브러리 및 예제코드 활용

SPC라이브러리는 PES-2401을 비롯한 스마트 확장보드 라이브러리입니다. 이 라이브러리를 사용하면 비교적 간단하게 PES-2401을 사용할 수 있습니다. 라이브러리와 함수에 대한 자세한 내용은 다음 문서를 참조하시기 바랍니다.

- [SPC라이브러리 매뉴얼 페이지](#)

명령어

스마트 확장보드를 사용하기 위해서는 SPC 라이브러리의 spc_request_dev함수 또는 spc_request_sys함수를 사용합니다.

```
spc_request_dev($sid, $cmd)
spc_request_sys($sid, $cmd)
```

- \$sid: 슬레이브 아이디
- \$cmd: 명령어 문자열

스마트 확장보드 공통 명령어

모든 스마트 확장보드가 공통으로 지원하는 명령어는 spc_request_sys함수를 사용합니다. 다음은 스마트 확장보드 공통 명령어 목록입니다.

명령어	옵션	설명
get	did	디바이스 아이디 확인
get	uid	유니크 아이디 확인

PES-2401 명령어

스마트 확장보드별로 적용되는 명령어는 spc_request_dev함수를 사용합니다. 다음은 PES-2401이 지원하는 명령어 목록입니다.

명령어	옵션	설명
set	\$port output \$level	선택한 포트를 ON(high) 또는 OFF(low)
set	\$port delay \$time	선택한 포트의 출력을 지연
get	\$port output	선택한 포트의 상태를 확인

- \$port : 제어할 포트 번호(0 ~ 3)
- \$level : 출력할 신호 레벨(high 또는 low)
- \$time : 지연시간(1 ~ 30,000, 단위 : 밀리초)

출력포트 제어

출력포트 제어를 위한 spc_request_dev함수 호출

```
spc_request_dev($sid, $cmd);
```

- \$sid : 슬레이브 아이디
- \$cmd : 명령어 문자열

명령어 구조는 다음과 같습니다.

```
"set $port output $value"
```

- \$port : 출력포트의 핀번호, 0 ~ 3까지 4개의 핀번호를 사용 가능
- \$value : 출력포트를 ON하려면 "high"을, OFF하려면 "low"를 사용

사용 예

```
<?php
include "/lib/sd_spc.php";
spc_reset();
spc_sync_baud();
$sid = 1;

echo "turn all output ports onWrWn";
spc_request_dev($sid, "set 0 output high");
spc_request_dev($sid, "set 1 output high");
spc_request_dev($sid, "set 2 output high");
spc_request_dev($sid, "set 3 output high");

sleep(1);

echo "turn all output ports offWrWn";
spc_request_dev($sid, "set 0 output low");
spc_request_dev($sid, "set 1 output low");
spc_request_dev($sid, "set 2 output low");
spc_request_dev($sid, "set 3 output low");
?>
```

출력 결과

```
turn all output ports on
```

turn all output ports off

출력포트 상태 확인

출력포트 상태 확인을 위한 spc_request_dev함수 호출

```
spc_request_dev($sid, $cmd);
```

- \$sid : 슬레이브 아이디
- \$cmd : 명령어 문자열

명령어 구조는 다음과 같습니다.

```
"get $port output"
```

- \$port : 출력포트의 포트번호, 0 ~ 3까지 4개의 포트번호를 사용 가능

반환 값

정상적인 반환 값은 문자열 형태로서 다음과 같습니다.

반환 값	설명
0	OFF
1	ON

사용 예

```
<?php
include "/lib/sd_spc.php";
spc_reset();
spc_sync_baud();
$sid = 1;

echo "turn all output ports onWrWn";
spc_request_dev($sid, "set 0 output high");
spc_request_dev($sid, "set 1 output high");
spc_request_dev($sid, "set 2 output high");
spc_request_dev($sid, "set 3 output high");

// get status of output ports
echo "Port 0: ", spc_request_dev($sid, "get 0 output"), "WrWn";
echo "Port 1: ", spc_request_dev($sid, "get 1 output"), "WrWn";
echo "Port 2: ", spc_request_dev($sid, "get 2 output"), "WrWn";
echo "Port 3: ", spc_request_dev($sid, "get 3 output"), "WrWn";

sleep(1);

echo "turn all output ports offWrWn";
```

```
spc_request_dev($sid, "set 0 output low");
spc_request_dev($sid, "set 1 output low");
spc_request_dev($sid, "set 2 output low");
spc_request_dev($sid, "set 3 output low");

// get status of output ports
echo "Port 0: ", spc_request_dev($sid, "get 0 output"), "WrWn";
echo "Port 1: ", spc_request_dev($sid, "get 1 output"), "WrWn";
echo "Port 2: ", spc_request_dev($sid, "get 2 output"), "WrWn";
echo "Port 3: ", spc_request_dev($sid, "get 3 output"), "WrWn"
?>
```

출력 결과

```
turn all output ports on
Port 0: 1
Port 1: 1
Port 2: 1
Port 3: 1
turn all output ports off
Port 0: 0
Port 1: 0
Port 2: 0
Port 3: 0
```