

# 제품 소개



## PES-2601

4포트 릴레이 출력보드인 PES-2601은 아두이노용 PHPoC 쉘드 제품 전용 스마트 확장보드입니다. 이 보드를 이용하면 아두이노 스케치를 통해 원격으로 릴레이를 ON/OFF 할 수 있습니다.

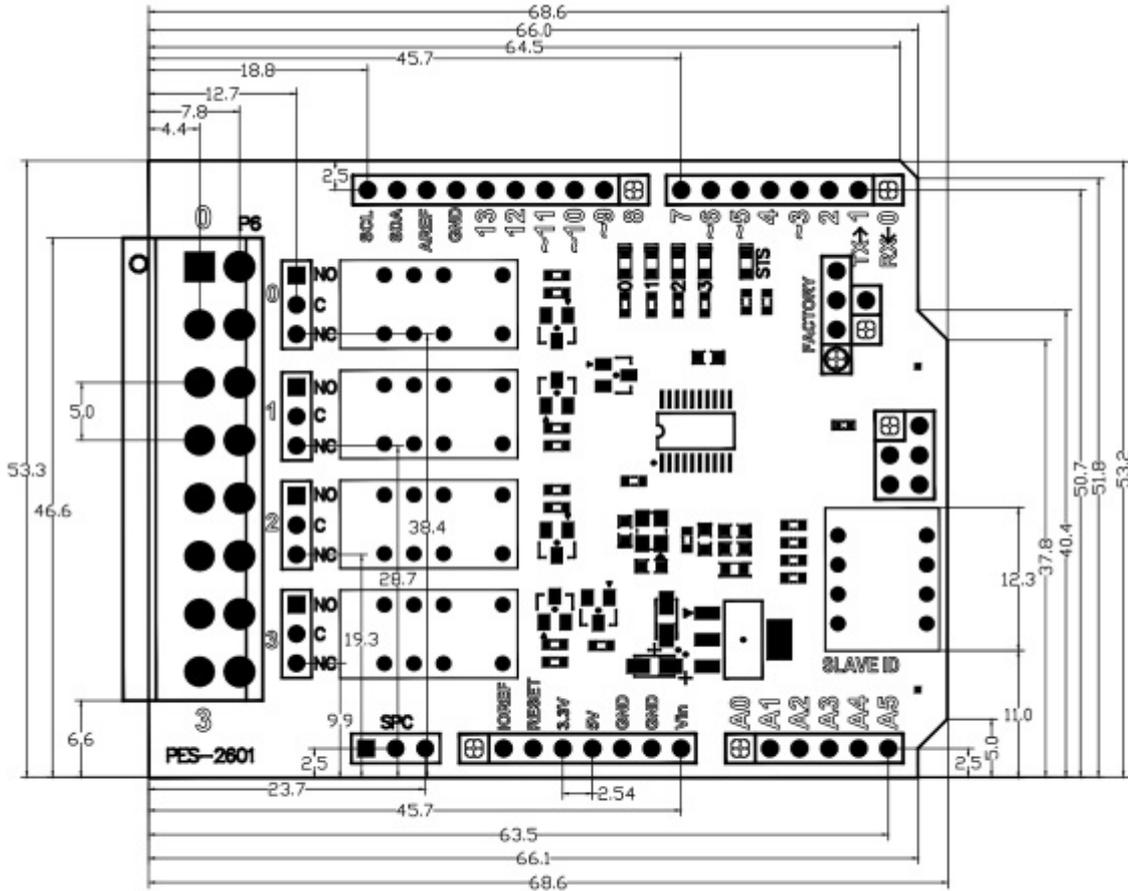
※ 주의 : 이 확장보드를 사용하기 위해서는 반드시 아두이노와 R2 이상 버전의 PHPoC 쉘드가 필요합니다!

### PHPoC 쉘드용 스마트 확장보드란?

PHPoC 쉘드용 스마트 확장보드는 자체 디바이스와 전용 펌웨어를 내장하고 있습니다. 이 보드는 PHPoC 쉘드와 전용 통신 포트를 이용해 마스터-슬레이브 방식으로 통신합니다. 하나의 PHPoC 쉘드에 여러 개의 스마트 확장보드를 연결할 수 있으며 각각의 스마트 확장보드에는 반드시 슬레이브 아이디를 설정해야 합니다.

# 치수

## 제품 본체



PES-2601 Dimension (mm)

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

## 터미널블록

이 보드는 8폴 터미널블록을 사용합니다. 치수는 각 터미널블록의 데이터시트를 참조하시기 바랍니다.

- T형 터미널블록 데이터시트
- S형 터미널블록 데이터시트

# 회로도

---

PES-2601의 회로도입니다.

- [PES-2601-V11-PO.pdf](#)





# 사용하기

---

이 보드를 사용하는 방법은 다음과 같습니다.

## 1. PHPoC 쉴드와 아두이노에 연결

이 보드는 단독으로 사용할 수 없습니다. 반드시 아두이노와 아두이노용 PHPoC 쉴드에 연결하여 사용하시기 바랍니다.

## 2. 아두이노용 라이브러리 설치

아두이노 IDE의 라이브러리 매니저를 통해 Phpoc 라이브러리와 PhpocExpansion 라이브러리를 설치합니다. 아두이노용 PHPoC 쉴드와 스마트 확장보드를 사용하려면 반드시 두 라이브러리를 모두 설치해야 합니다. 라이브러리에 대한 자세한 내용은 다음 문서를 참조하시기 바랍니다.

- [PHPoC 쉴드 라이브러리 레퍼런스](#)

## 3. 예제코드 활용

본 매뉴얼과 라이브러리에 포함된 예제코드를 활용하여 프로그래밍하시기 바랍니다.

# 클래스 및 함수

## 클래스

이 확장보드를 사용하기 위해서는 아두이노 PHPoC Expansion 라이브러리의 ExpansionRelayOutput 클래스를 사용합니다.

## 멤버 함수

ExpansionRelayOutput 클래스의 사용 가능한 멤버함수는 다음과 같습니다.

멤버 함수	설명
int getPID(void)	제품 아이디 읽기
char *getName(void)	제품명 읽기
ExpansionRelayOutput(int sid, int port)	특정 출력포트의 인스턴스 생성
void on(void)	출력포트 켜기(ON)
void off(void)	출력포트 끄기(OFF)
bool isOn(void)	출력포트 상태가 ON인지 확인
bool isOff(void)	출력포트 상태가 OFF인지 확인
void setDelay(int ms)	출력 지연(밀리 초)

# 출력포트 제어

## 예제 (출력포트 제어)

- 아두이노 소스코드

```
#include <PhpocExpansion.h>
#include <Phpoc.h>

byte spcId = 1;

ExpansionRelayOutput output0(spcId, 0);
ExpansionRelayOutput output1(spcId, 1);
ExpansionRelayOutput output2(spcId, 2);
ExpansionRelayOutput output3(spcId, 3);

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
  Expansion.begin();
}

void loop() {
  Serial.println("turn all output ports on");
  output0.on();
  output1.on();
  output2.on();
  output3.on();
  delay(1000);

  Serial.println("turn all output ports off");
  output0.off();
  output1.off();
  output2.off();
  output3.off();
  delay(1000);
}
```

- 출력 결과

```
turn all output ports on
turn all output ports off
...
```

## 예제 (출력 지연)

- 아두이노 소스코드

```
#include <PhpocExpansion.h>
#include <Phpoc.h>

byte spcId = 1;

ExpansionRelayOutput output0(spcId, 0);
ExpansionRelayOutput output1(spcId, 1);
ExpansionRelayOutput output2(spcId, 2);
ExpansionRelayOutput output3(spcId, 3);

void getOutputState() {
  if(output0.isOn())
    Serial.print("Port 0: 1, ");
  else
    Serial.print("Port 0: 0, ");

  if(output1.isOn())
    Serial.print("Port 1: 1, ");
  else
    Serial.print("Port 1: 0, ");

  if(output2.isOff())
    Serial.print("Port 2: 0, ");
  else
    Serial.print("Port 2: 1, ");

  if(output3.isOff())
    Serial.println("Port 3: 0");
  else
    Serial.println("Port 3: 1");
}

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
  Expansion.begin();

  output0.setDelay(100);
  output1.setDelay(100);
  output2.setDelay(100);
  output3.setDelay(100);
}

void loop() {
  Serial.println("turn all output ports on");
  output0.on();
  output1.on();
  output2.on();
}
```

```
output3.on();

// get status of output ports
getOutputState();
delay(110);
getOutputState();

delay(1000);

Serial.println("turn all output ports off");
output0.off();
output1.off();
output2.off();
output3.off();

// get status of output ports
getOutputState();
delay(110);
getOutputState();

delay(1000);
}
```

- 출력 결과

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

# 출력포트 상태 확인

## 예제

- 아두이노 소스코드

```
#include <Phpoc.h>
#include <PhpocExpansion.h>

byte spcId = 1;

ExpansionRelayOutput output0(spcId, 0);
ExpansionRelayOutput output1(spcId, 1);
ExpansionRelayOutput output2(spcId, 2);
ExpansionRelayOutput output3(spcId, 3);

void getOutputState() {
    if(output0.isOn())
        Serial.println("Port 0: 1");
    else
        Serial.println("Port 0: 0");

    if(output1.isOn())
        Serial.println("Port 1: 1");
    else
        Serial.println("Port 1: 0");

    if(output2.isOff())
        Serial.println("Port 2: 0");
    else
        Serial.println("Port 2: 1");

    if(output3.isOff())
        Serial.println("Port 3: 0");
    else
        Serial.println("Port 3: 1");
}

void setup() {
    Serial.begin(9600);
    while(!Serial)
        ;

    Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
    Expansion.begin();
}

void loop() {
    Serial.println("turn all output ports on");
    output0.on();
    output1.on();
    output2.on();
}
```

```
output3.on();

// get status of output ports
getOutputState();

delay(1000);

Serial.println("turn all output ports off");
output0.off();
output1.off();
output2.off();
output3.off();

// get status of output ports
getOutputState();

delay(1000);
}
```

- 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
...
```