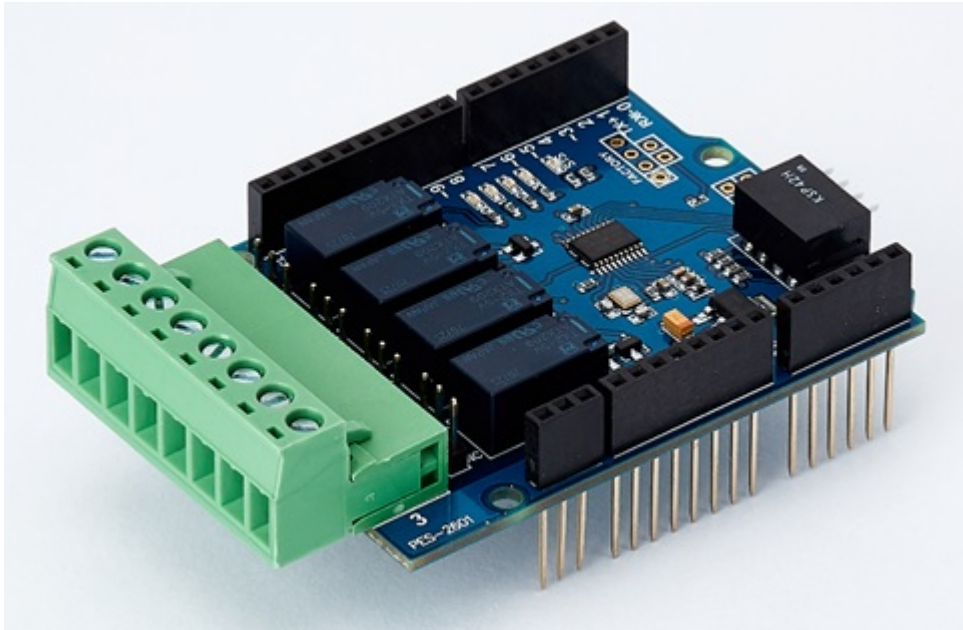


Introduction



PES-2601

PES-2601, 4-Port Relay Output board, is one of smart expansion boards for PHPoC Shields for Arduino. You can remotely turned on / off the relays with this board via the Arduino sketch.

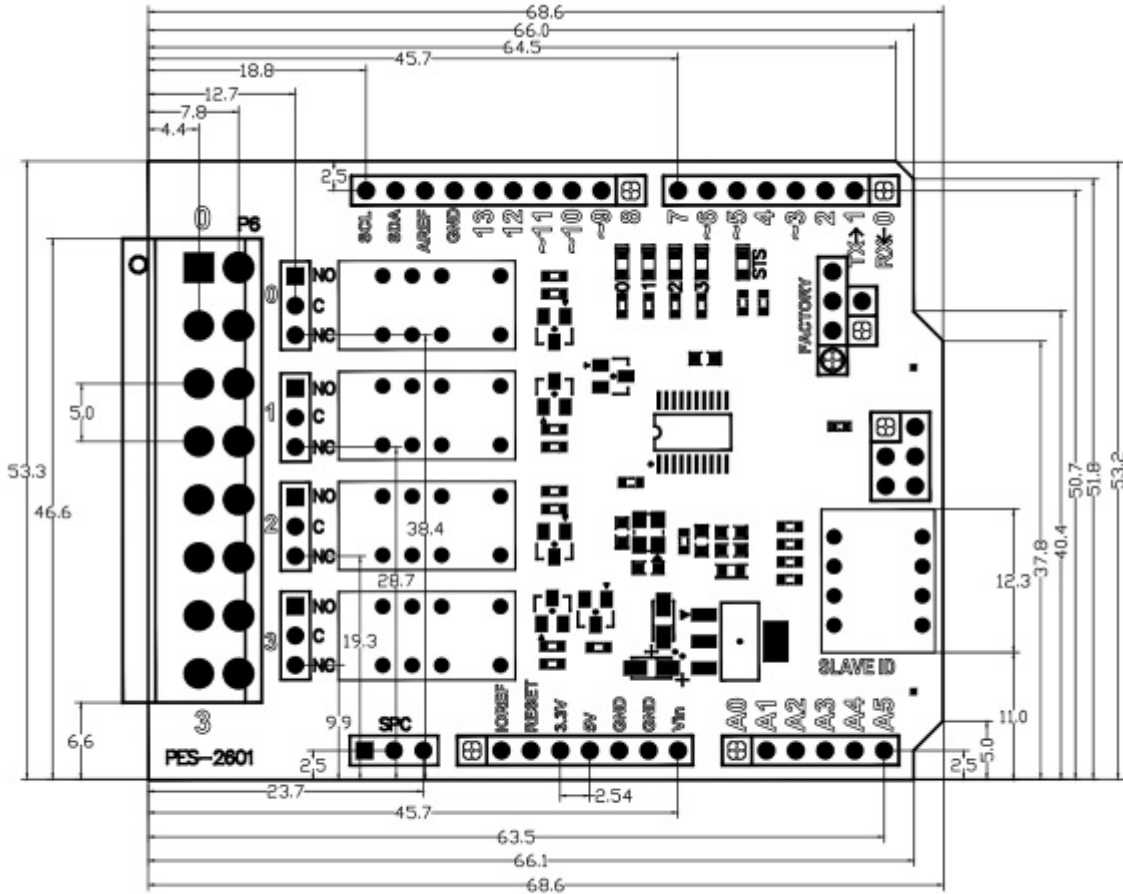
※ Caution : Both a PHPoC shield with R2 or later version and an Arduino board are required to use this board!

What is the Smart Expansion Board for PHPoC shield?

A smart expansion board for PHPoC shield has own devices and firmware. This board communicate with a PHPoC shield in a master-slave protocol through the designated port. Two or more smart expansion boards can be connected to one PHPoC shield and each of them required to be setting a slave id.

Dimension

Body



PES-2601 Dimension (mm)

※ Dimensions(unit : mm) may vary according to a method of measurement.

Terminal Block

This board uses two types of 8-pole terminal block. Refer to each datasheet for dimension.

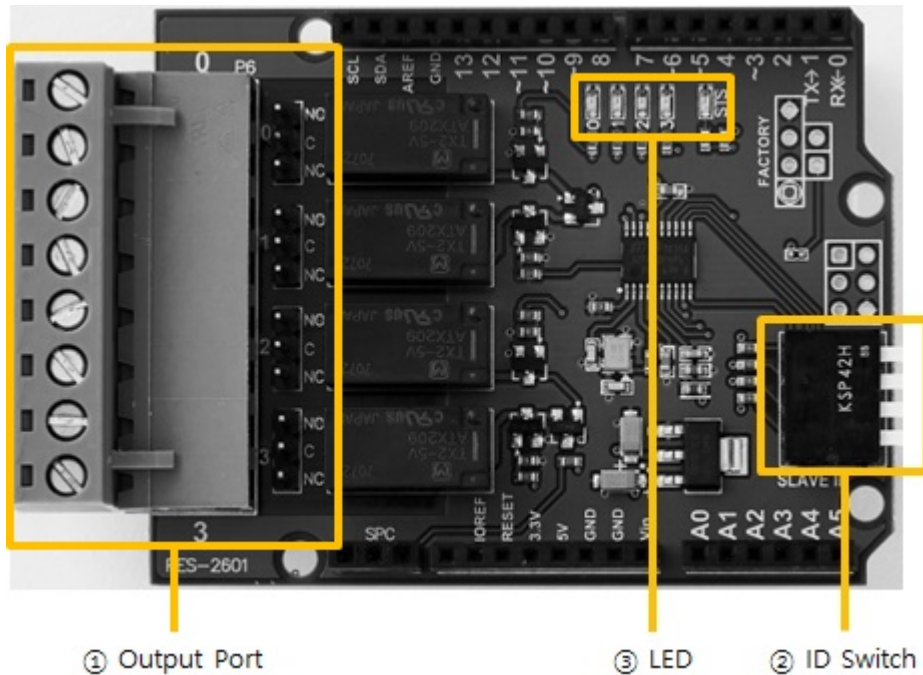
- [Datasheet of T-type Terminal Block](#)
- [Datasheet of S-type Terminal Block](#)

Schematic

This is the schematic of PES-2601.

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

Layout



1. Output Ports

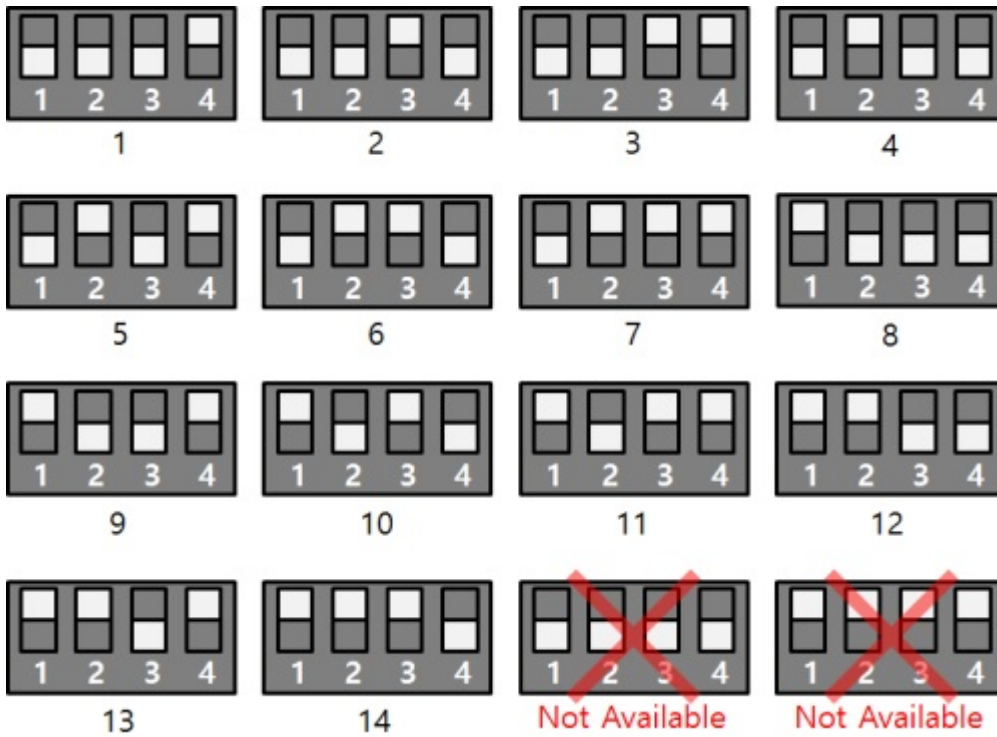
Output ports are interfaced with a 5mm spaced terminal block which has 8 terminals. Every output port is connected to a relay and each port can be selected NO (Normal Open) or NC (Normal Close) by a jumper. Output ports' range of use is as follows:

Voltage (DC)	Max. Permissible Current
30V	2A

※ Caution : It may result in product malfunction to use beyond the maximum permissible current. Be sure to use it considering the peak current of a connected device.

2. SLAVE ID Switch

A slave ID is used when PHPoC shield identifies each smart expansion board. So, each smart expansion board, which is connected to a PHPoC shield, should have a unique slave ID. The slave ID can be set one of the numbers from 1 to 14 by 4 DIP switches as follows:



3. LED

This board has five LEDs.

LED	Description
STS	when slave id is valid > repeat on and off every second when slave id is invalid > blinks very quickly when it failed to communicate with a PHPoC shield > off
0	turned ON with output 0 is ON
1	turned ON with output 1 is ON
2	turned ON with output 2 is ON
3	turned ON with output 3 is ON

How to Use

This board can be used by steps as follows.

1. Connect to a PHPoC Shield and an Arduino

It is not possible to use this board alone. Please be sure that connection to a PHPoC Shield and an Arduino.

2. Install Libraries for Arduino

Install PHPoC and PhpocExpansion library via library manager on Arduino IDE. Both libraries are required to use PHPoC shield and this board. Refer to the manual pages below for detail about the libraries.

- [PHPoC shield library reference](#)

3. Use Sample Codes

Use sample codes in libraries and examples in this manual.

Class and Functions

Class

To use this extension board, use the ExpansionRelayOutput class of the PHPoC Expansion library.

Member Functions

Available member functions of the ExpansionRelayOutput class are as follows:

Member Function	Description
int getPID(void)	get the product's ID
char *getName(void)	get the product's name
ExpansionRelayOutput(int sid, int port)	create an instance of a specific output port
void on(void)	turn a output port ON
void off(void)	turn a output port OFF
bool isOn(void)	return true when a output port is ON
bool isOff(void)	return true when a output port is OFF
void setDelay(int ms)	give delay for output(millisecond)

Controlling Output Ports

Example : On/Off

- source code for Arduino

```
#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);
}
```

- output result

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


Example : Delayed On/Off

- source code for Arduino Sketch

```
#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);
}
```

- output result

```
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
...
```

Monitoring Output Ports

Example

- source code for Arduino Sketch

```
#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);
}
```

- output result

```
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
...
```