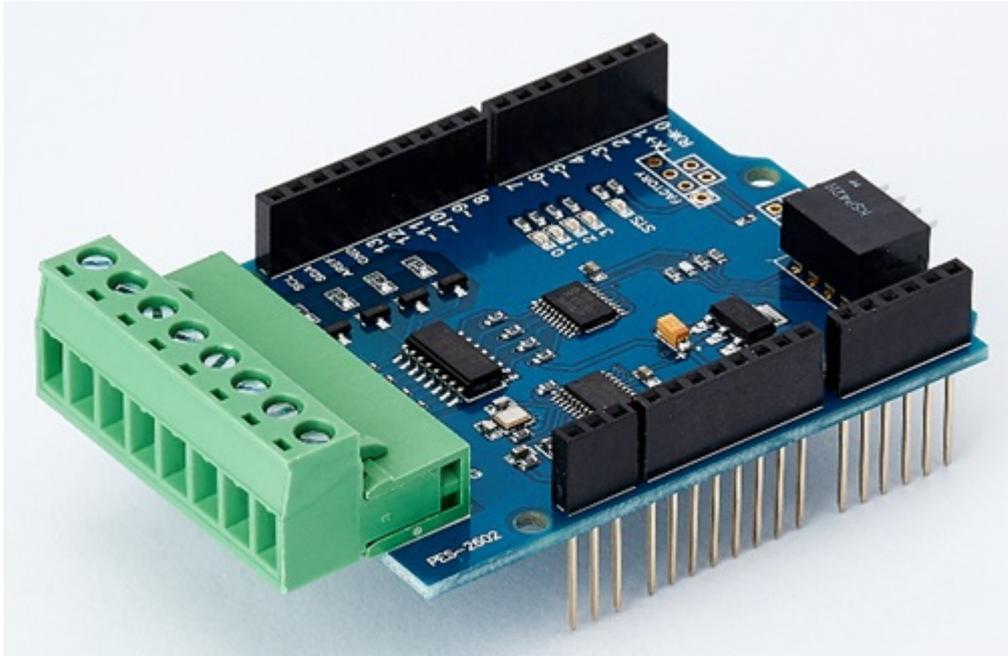


# Introduction



## PES-2602

PES-2602, 4-Port Photocopler Input board, is one of smart expansion boards for PHPoC shield for Arduino. You can monitor both DRY contact and WET contact outputs from some devices via the Arduino sketch.

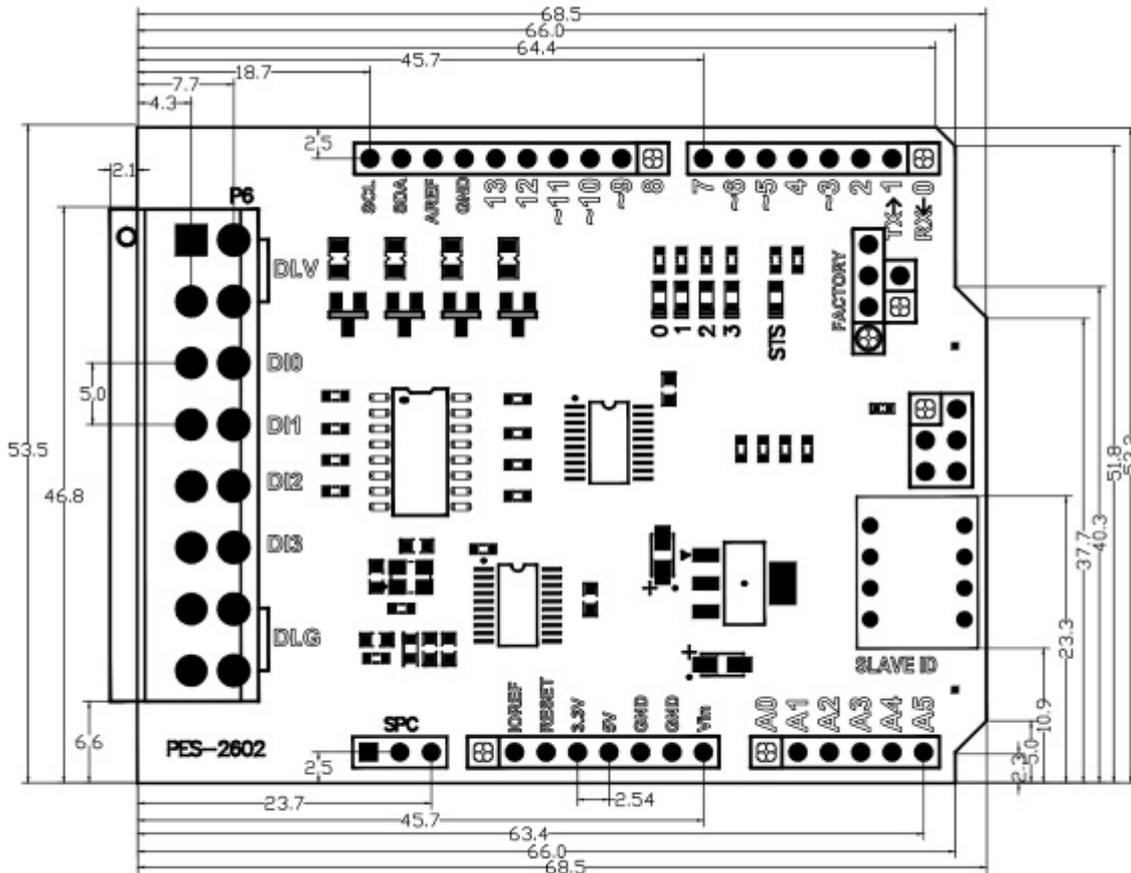
※ Caution : Both a PHPoC shield with R2 or later version and an Arduino board are required to use this board!

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-2602 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](#)

---

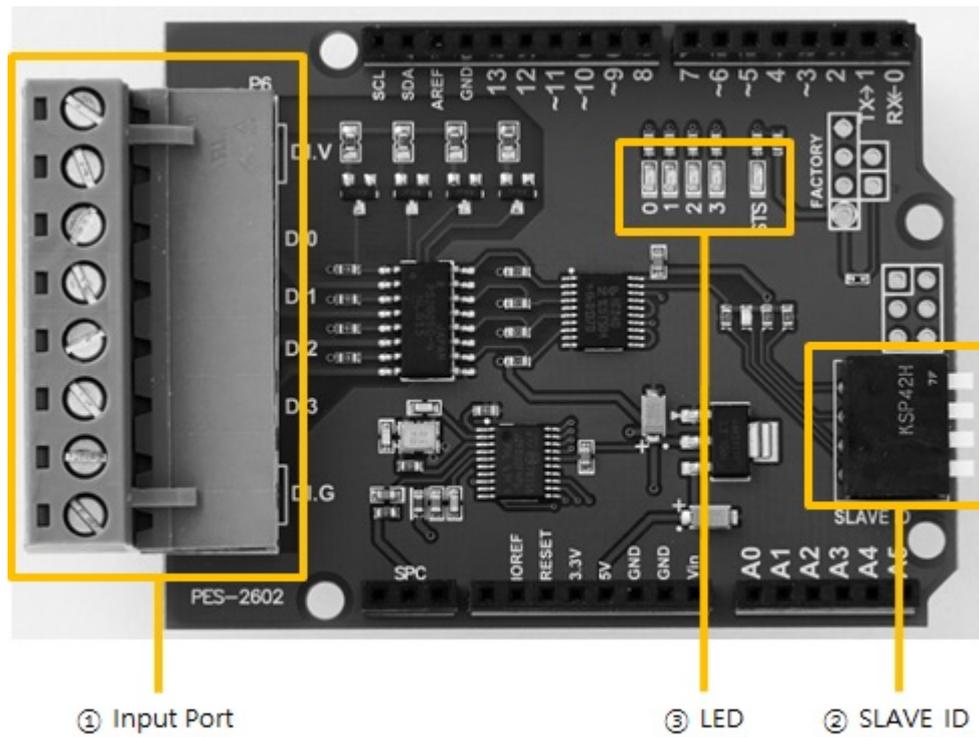
# Schematic

---

This is the schematic of PES-2602.

- [PES-2602-V10-PO.pdf](#)

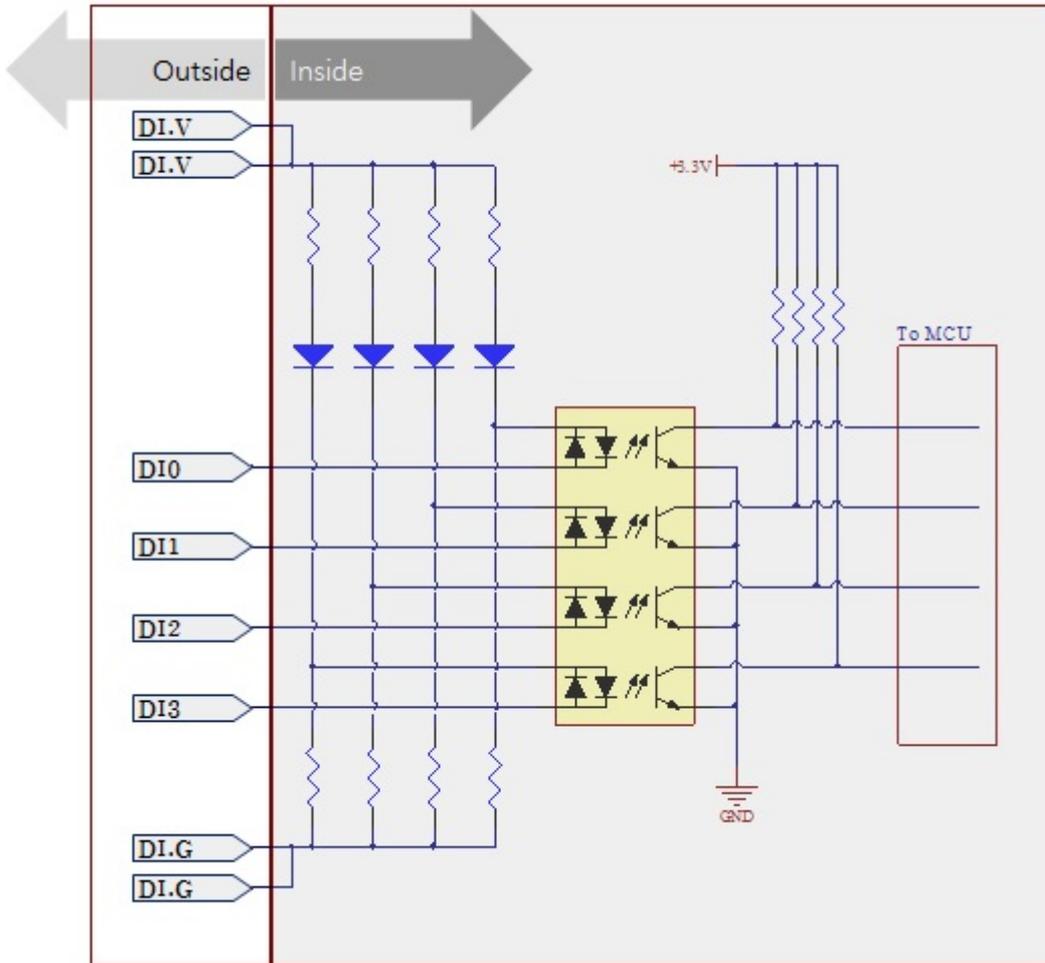
# Layout



## 1. Input Ports

Input ports are interfaced with a 5mm spaced terminal block which has 8 terminals. Every output port is isolated by photocoupler and supports wet contact, dry contact, NPN transistor and PNP transistor input.

Circuit Diagram of Input Port

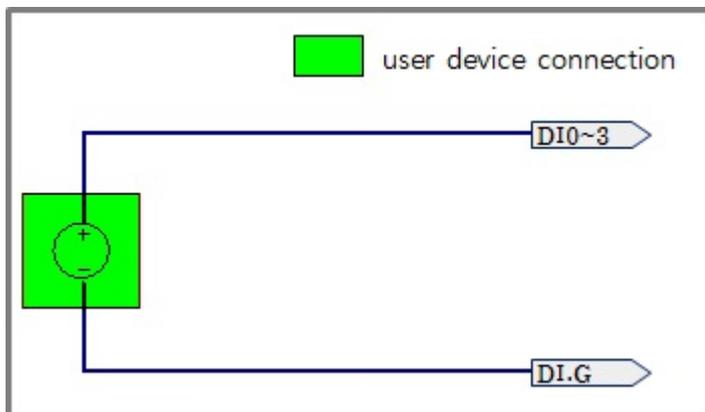


### WET contact

Voltage conditions in wet contact is as follows:

division	condition
maximum DC input	DC 30V
ON	more than DC 3V
OFF	less than DC 1.5V

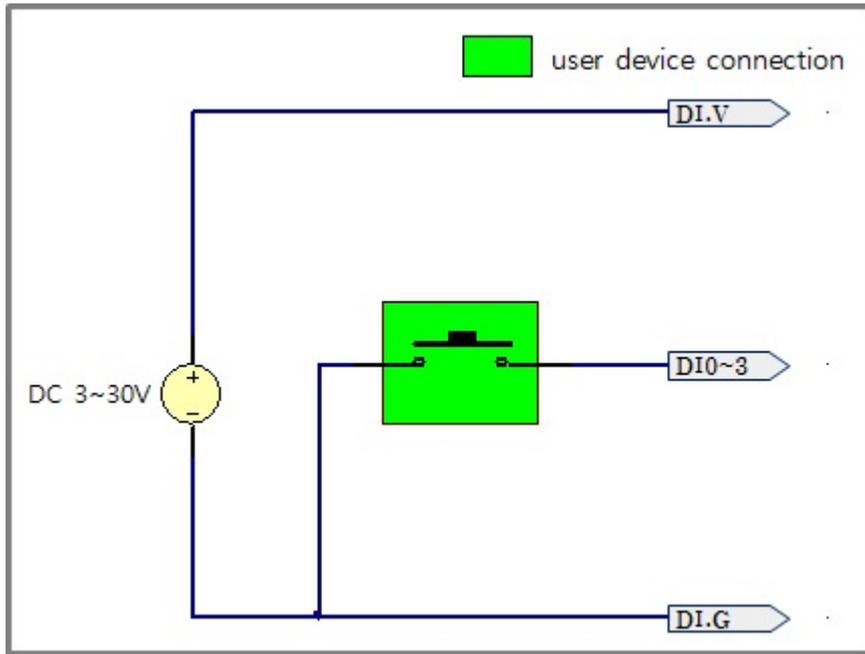
Refer to the following figure for connection with your device.



### Dry contact

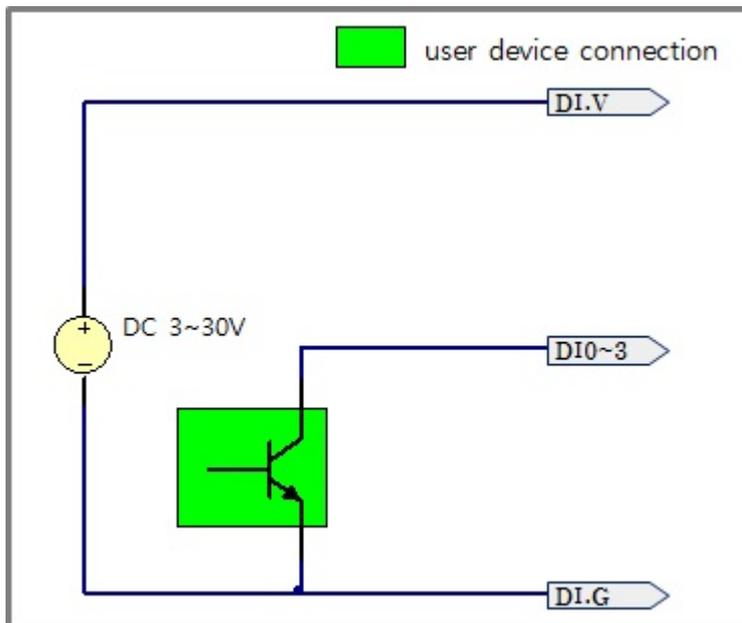
An input port is ON under being short circuit between the port and DI.G port in this type. It means

additional power should be supplied between DI.V and DI.G. Refer to the following figure for connection with your device.



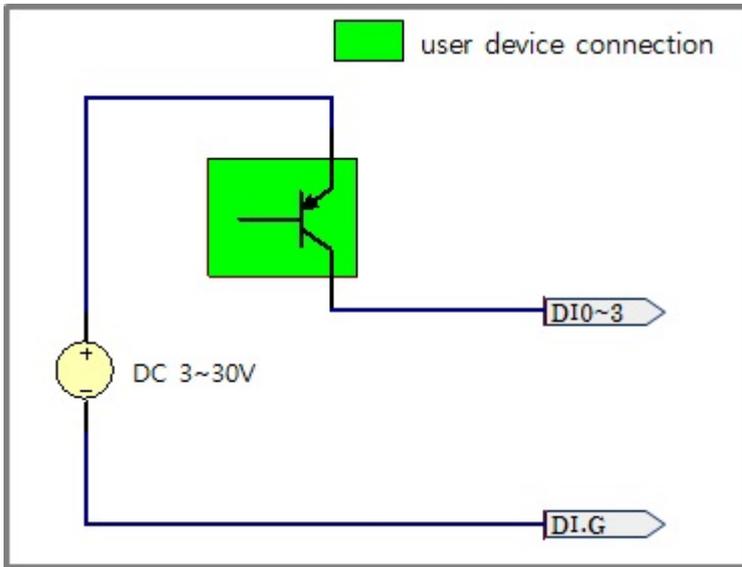
### NPN Transistor Connection

Refer to the following figure for connection with an NPN transistor.



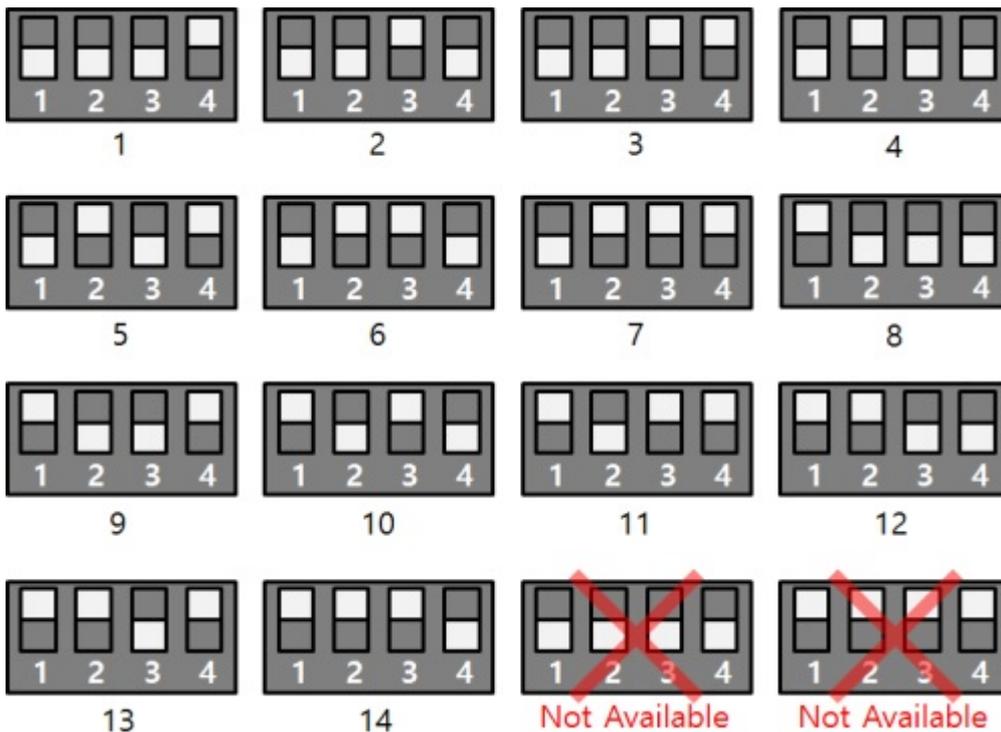
### PNP Transistor Connection

Refer to the following figure for connection with a PNP transistor.



## 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 input 0 is ON
1	turned ON with input 1 is ON

LED	Description
2	turned ON with input 2 is ON
3	turned ON with input 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 ExpansionPhotoInput class of the PHPoC Expansion library.

## Member Functions

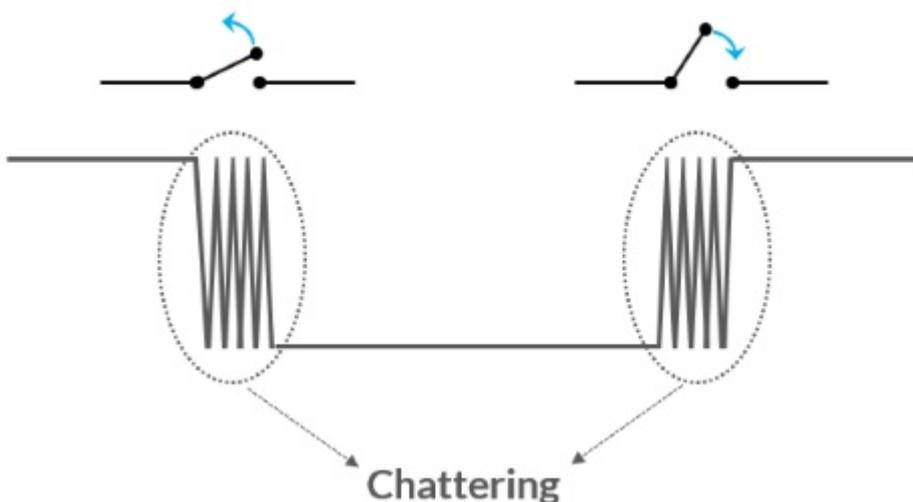
Available member functions of the ExpansionPhotoInput class are as follows:

Member Function	Description
int getPID(void)	get the product's ID
char *getName(void)	get the product's name
ExpansionPhotoInput(int sid, int port)	create an instance of a specific input port
bool isOn(void)	return true when an input port is ON
bool isOff(void)	return true when an input port is OFF
void setDelay(int ms)	set valid time(millisecond)

### setDelay Function

If the input signal level changes but does not be maintained for the specified delay time, this board will ignore it. The unit is milliseconds and the setting range is 1 to 30,000. If set to zero, the minimum value is applied. This delay time takes effect until the another setting for the same port is run.

This function is designed to avoid impact of the chattering phenomenon. This phenomenon happens when a manual switch repeatedly turns on and off in a short time because of vibration of the contact. This may cause malfunction in some kind of application.



# Monitoring Input Ports

## Example

- source code for Arduino

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

byte spcId = 1;

ExpansionPhotoInput input0(spcId, 0);
ExpansionPhotoInput input1(spcId, 1);
ExpansionPhotoInput input2(spcId, 2);
ExpansionPhotoInput input3(spcId, 3);

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

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

  Serial.println(input0.getPID());
  Serial.println(input0.getName());
}

void loop() {
  if(input0.isOn())
    Serial.println("Port 0: 1");
  else
    Serial.println("Port 0: 0");

  if(input1.isOn())
    Serial.println("Port 1: 1");
  else
    Serial.println("Port 1: 0");

  if(input2.isOff())
    Serial.println("Port 2: 0");
  else
    Serial.println("Port 2: 1");

  if(input3.isOff())
    Serial.println("Port 3: 0");
  else
    Serial.println("Port 3: 1");

  Serial.println();

  delay(1000);
}
```

```
}
```

- output result(example)

```
Port 0: 0  
Port 1: 1  
Port 2: 0  
Port 3: 1  
...
```