Grove - Relay SKU:103020005

The Grove-Relay module is a digital normally-open switch. Through it, you can control circuit of high voltage with low voltage, say 5V on the controller. There is an indicator LED on the board, which will light up when the controlled terminals get closed.

Version

Parameter	V1.1	V1.2
Product Release Date	27th Jan 2013	9th June 2014
Operating Voltage	5V	3.3V~5V
Operating Current	60mA	100mA
Relay Life	100,000 Cycle	100,000 Cycle
Max Switching Voltage	250VAC/30VDC	250VAC/30VDC
Max Switching Current	5A	5A

!!!Tip More details about Grove modules please refer to Grove System

Specifications

Parameter	Value/Range
Size	L:40mm W:20mm H: 18.5mm
Weight	12.6g
Certification	ROHS

Platforms Supported

Arduino	Raspberry Pi	BeagleBone	Wio	LinkIt ONE

!!!Caution The platforms mentioned above as supported is/are an indication of the module's software or theoritical compatibility. We only provide software library or code examples for Arduino platform in most cases. It is not possible to provide software library / demo code for all possible MCU platforms. Hence, users have to write their own software library.

Getting Started

Play With Arduino

!!!Note If this is the first time you work with Arduino, we strongly recommend you to see Getting Started with Arduino before the start.

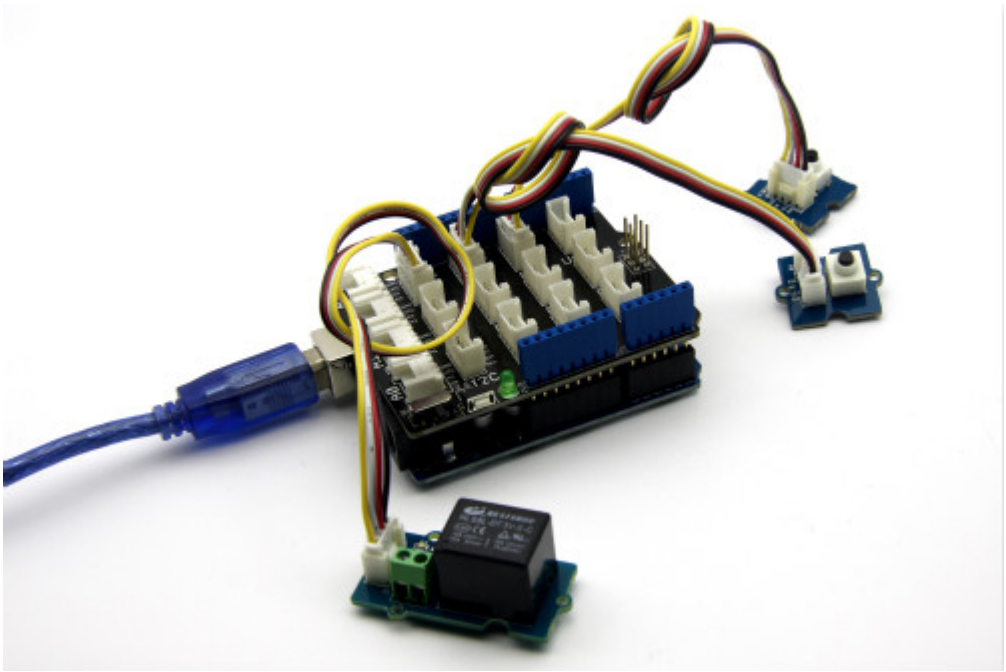
Materials required

Seeeduino V4.2	Base Shield	Grove-Button x2	Grove-Relay
			

!!!note **1** Please plug the USB cable gently, otherwise you may damage the port. Please use the USB cable with 4 wires inside, the 2 wires cable can't transfer data. If you are not sure about the wire you have, you can click here to buy.
2 Each Grove module comes with a Grove cable when you buy. In case you lose the Grove cable, you can click here to buy

Hardware

- **Step 1.** Connect Grove-Relay to port **D4** of Grove-Base Shield.
- **Step 2.** Connect Grove-Button#1 to port **D2** of Grove-Base Shield, Connect Grove-Button#2 to port **D3** of Grove-Base Shield.
- **Step 3.** Plug Grove - Base Shield into Seeeduino.
- **Step 4.** Connect Seeeduino to PC via a Micro-USB cable.



!!!Note If we don't have the base shield, we also can directly connect the Grove-Relay and Grove-Button to Arduino board. Please follow below connection.

Grove-Relay	Arduino	Grove Cable
GND	GND	Black
VCC	5V	Red
SIG	D4	Yellow

Grove-Button#1	Arduino	Grove Cable
GND	GND	Black
VCC	5V	Red
SIG	D2	Yellow

Grove-Button#2	Arduino	Grove Cable
GND	GND	Black
VCC	5V	Red
SIG	D3	Yellow

Software

Here is a demo that shows you how to control a Grove - Relay with a Grove - Button. When one button gets pressed, the relay will close. When the other button gets pressed, the relay will open.

- **Step 1.** Open the Arduino IDE and copy the following code into a new sketch.

```
// Relay Control

void setup()
{
  pinMode(2, INPUT);
  pinMode(3, INPUT);
  pinMode(4, OUTPUT);
}

void loop()
{
  if (digitalRead(2)==HIGH)
  {
    digitalWrite(4, HIGH);
    delay(100);
  }
  if (digitalRead(3)==HIGH)
  {
    digitalWrite(4, LOW);
  }
}
```

- **Step 2.** Upload the demo. If you do not know how to upload the code, please check How to upload code.

Done uploading, if you press the button#1 the relay should be on; and if you press the button#2 the relay should be off.

Play with Codecraft

Hardware

Step 1. Connect a Grove - Relay to port D4, connect two Grove - Button to port D2 and port D3 of a Base Shield.

Step 2. Plug the Base Shield to your Seeeduino/Arduino.

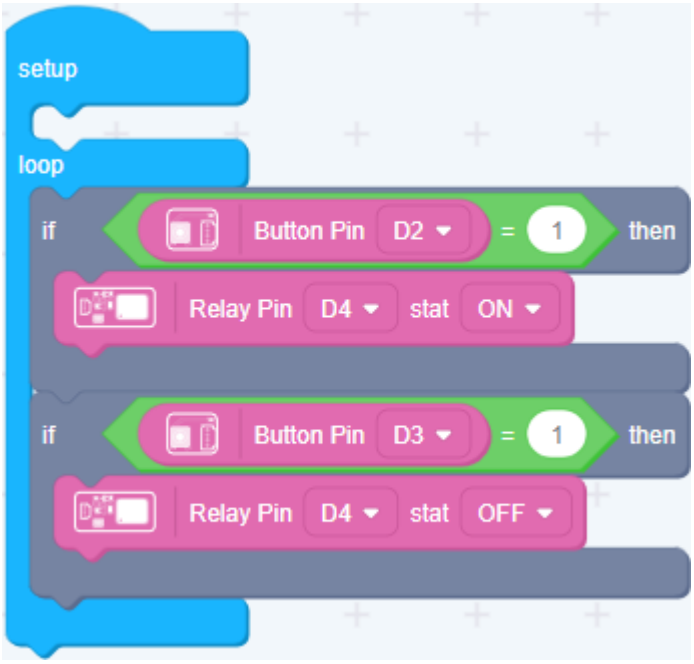
Step 3. Link Seeeduino/Arduino to your PC via an USB cable.

Software

Step 1. Open Codecraft, add Arduino support, and drag a main procedure to working area.

!!!Note If this is your first time using Codecraft, see also Guide for Codecraft using Arduino.

Step 2. Drag blocks as picture below or open the cdc file which can be downloaded at the end of this page.



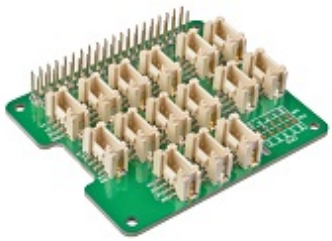
Upload the program to your Arduino/Seeeduino.

!!!Success When the code finishes uploaded. Relay will turns on when you push the button connected to port D2, and it will turns off when you push the button connected to port D3.

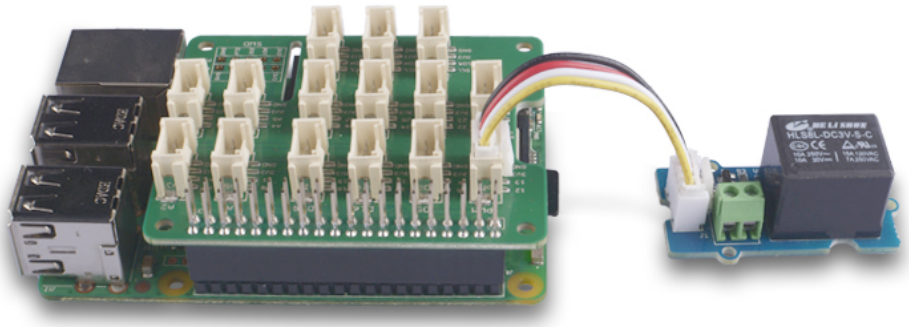
Play With Raspberry Pi (With Grove Base Hat for Raspberry Pi)

Hardware

- **Step 1**. Things used in this project:

Raspberry pi	Grove Base Hat for RasPi	Grove - Relay
		

- **Step 2**. Plug the Grove Base Hat into Raspberry.
- **Step 3**. Connect the Grove - Relay to port 12 of the Base Hat.
- **Step 4**. Connect the Raspberry Pi to PC through USB cable.



!!! Note For step 3 you are able to connect the relay module to **any GPIO Port** but make sure you change the command with the corresponding port number.

Software

- **Step 1**. Follow Setting Software to configure the development environment.
- **Step 2**. Download the source file by cloning the grove.py library.

```
cd ~
git clone https://github.com/Seeed-Studio/grove.py
```

- **Step 3**. Excute below commands to run the code.

```
cd grove.py/grove
python grove_relay.py 12
```

Following is the grove_relay.py code.

```
from grove.gpio import GPIO


class GroveRelay(GPIO):
    def __init__(self, pin):
        super(GroveRelay, self).__init__(pin, GPIO.OUT)

    def on(self):
        self.write(1)

    def off(self):
        self.write(0)


Grove = GroveRelay


def main():
    import sys
    import time

    if len(sys.argv) < 2:
        print('Usage: {} pin'.format(sys.argv[0]))
        sys.exit(1)

    relay = GroveRelay(int(sys.argv[1]))

    while True:
        try:
            relay.on()
            time.sleep(1)
            relay.off()
            time.sleep(1)
        except KeyboardInterrupt:
            relay.off()
```

```
            print("exit")
            exit(1)

if __name__ == '__main__':
    main()
```

!!!success If everything goes well, you will be able to see the LED indicator blinking.

You can quit this program by simply press ++ctrl+c++.

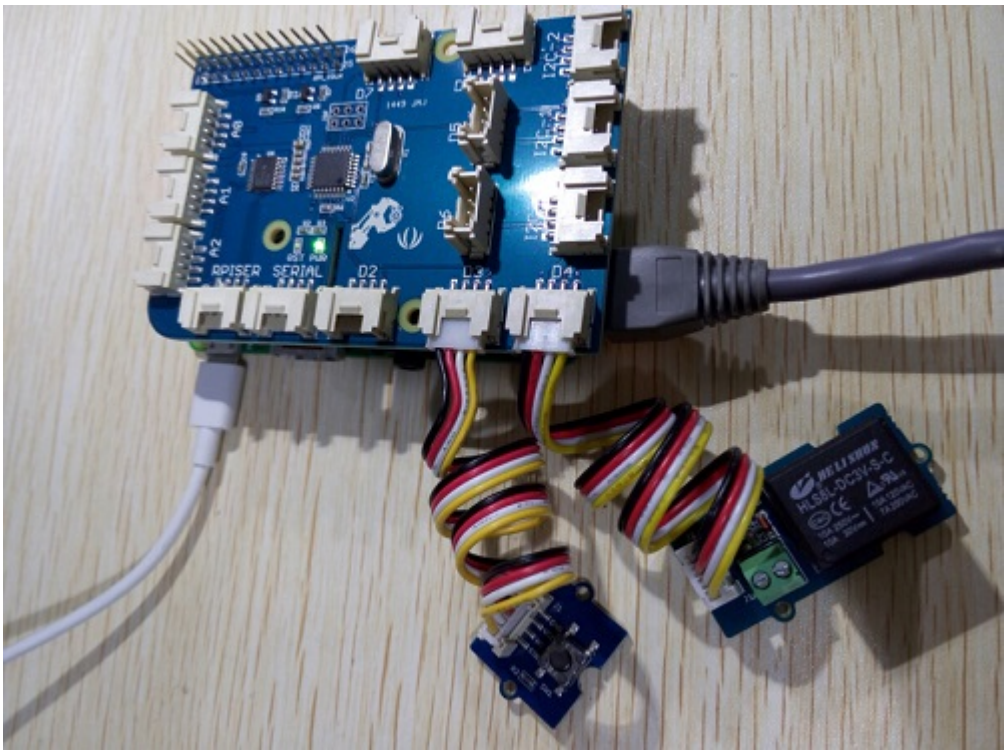
Play With Raspberry Pi (with GrovePi_Plus)

Hardware

Materials required

Raspberry pi	GrovePi_Plus	Grove-Button	Grove-Relay
			

- **Step 1.** Plug the GrovePi_Plus into Raspberry.
- **Step 2.** Connect the Grove-Relay to **D4** port of GrovePi_Plus.
- **Step 3.** Connect the Grove-Button to **D3** port of GrovePi_Plus.
- **Step 4.** Connect the Raspberry to PC via USB cable.



Software

If this is the first time you use GrovePi, please do this part step by step. If you are an old friend with GrovePi, you can skip **Step1** and **Step2**.

- **Step 1.** Setting Up The Software. In the command line, type the following commands:

```
sudo curl -kL dexterindustries.com/update_grovepi | bash
```

```
sudo reboot
```

```
cd /home/pi/Desktop
```

```
git clone https://github.com/DexterInd/GrovePi.git
```

For more detail about this part, please refer to Setting Software.

- **Step 2.** Follow Updating the Firmware to update the latest firmware of GrovePi.

!!!Note We firmly suggest you to update the firmware, or for some sensors you may get errors.

- **Step 3.** Run the following command to get the result.

```
cd /home/pi/Desktop/GrovePi/Software/Python/
sudo python grove_switch_relay.py
```

If you want to check the code, you can use the following command:

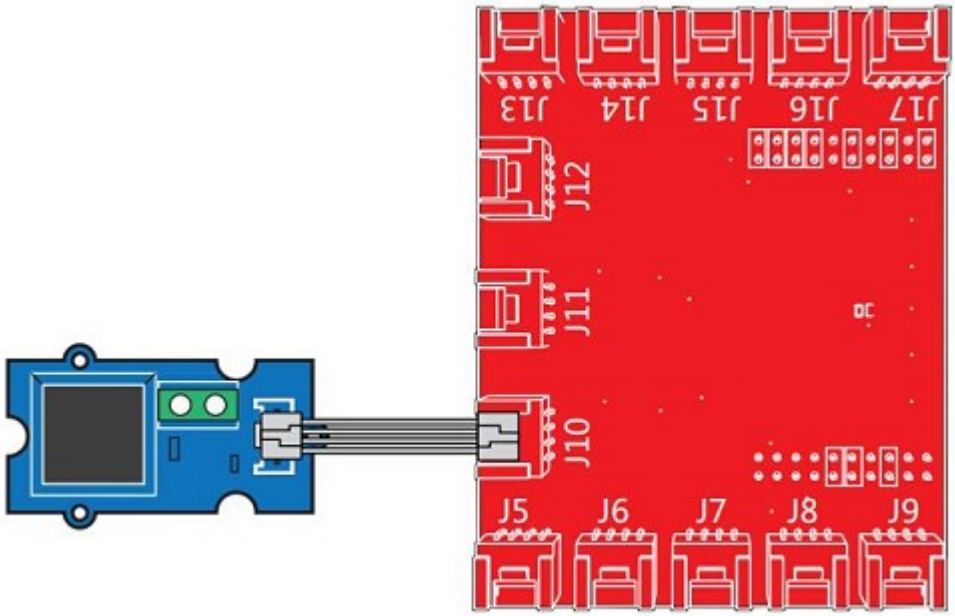
```
sudo nano grove_switch_relay.py
```

The code :

```
# Raspberry Pi + Grove Switch + Grove Relay

import time
import grovepi
# Connect the Grove Switch to digital port D3
# SIG,NC,VCC,GND

switch = 3
# Connect the Grove Relay to digital port D4
# SIG,NC,VCC,GND

relay = 4
grovepi.pinMode(switch,"INPUT")
grovepi.pinMode(relay,"OUTPUT")
while True:
    try:
        if grovepi.digitalRead(switch):
            grovepi.digitalWrite(relay,1)
        else:
            grovepi.digitalWrite(relay,0)
            time.sleep(.05)
    except KeyboardInterrupt:
        grovepi.digitalWrite(relay,0)
        break
    except IOError:
        print "Error"
```

Play With TI LaunchPad

Controlling other electronics (Relay)



This example shows how to use the Grove-relay module to control larger load, i.e. a desk lamp light. A 3V voltage signal can cause the relay to switch on, allowing current to flow through the connected appliance.

```
/*
Relay
The basic Energia example.
This example code is in the public domain.
*/

#define RELAY_PIN 39

// the setup routine runs once when you press reset:
void setup() {
        pinMode(RELAY_PIN, OUTPUT); // initialize the digital pin as an output.
}

// the loop routine runs over and over again forever:
void loop() {
        digitalWrite(RELAY_PIN, HIGH); // turn the relay on (HIGH is the voltage
level)
        delay(1000);   // wait for a second
        digitalWrite(RELAY_PIN, LOW);   // turn the relay o by making the voltage
LOW
        delay(1000);   // wait for a second
}
```

Resources

- **[Eagle]** Grove - Relay Schematic and PCB in Eagle format
- **[PDF]** Grove - Relay PCB in PDF format
- **[PDF]** Grove - Relay Schematic in PDF format
- **[Datasheet]** HLS8-T73 Series Relay Datasheet

- **[Codecraft]** CDC File

Tech Support

Please submit any technical issue into our forum or drop mail to techsupport@seeed.cc.