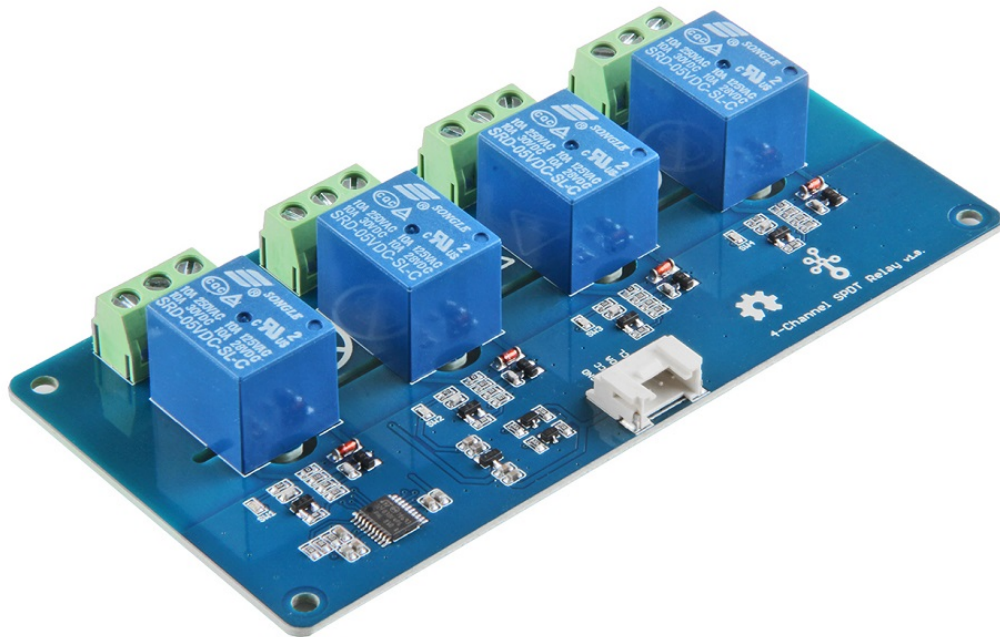


Grove - 4-Channel SPDT Relay



The Single Pole Double Throw SPDT relay is quite useful in certain applications because it has one common terminal and 2 contacts which are great for selecting between two options. The Grove - 4-Channel SPDT Relay has four single pole - double throw (SPDT) switches. It only requires low-voltage and low current signals to control those switches. Specifically, you can use 5V DC to control

max.250V AC or 110V DC. The I2C address is changeable so that you can use multiple relay modules in the same project. The Grove - 4-Channel SPDT Relay has four single pole - double throw (SPDT) switches. It only requires low-voltage and low current signals to control those switches. Specifically, you can use 5V DC to control max.250V AC or 110V DC.

We use an on-board STM32F030F4P6 to control the channels separately. The command from Arduino or other boards is transmit via the I2C interface, the on-board STM32F030F4P6 will parse the command, so that you can control the switch you want.

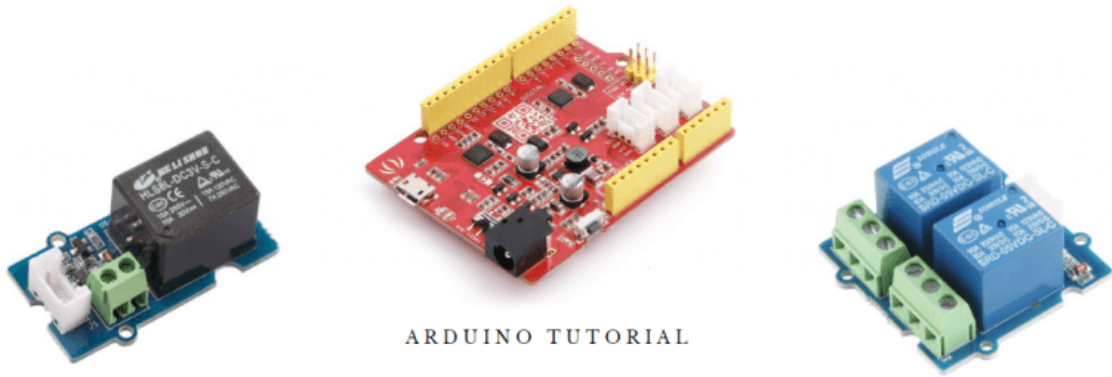
Get One Now 

[<https://www.seeedstudio.com/Grove-4-Channel-SPDT-Relay-p-3119.html>]

Pre-reading

An introduction of **What is a Grove Relay Module** and **How does a Relay work** is strongly recommended reading ahead if you are not familiar with them. Please visit our [blog](#)

[<https://www.seeedstudio.com/blog/2020/01/03/arduino-tutorial-control-high-voltage-devices-with-relay-modules/>] below for detailed information:



ARDUINO TUTORIAL

Control High Voltage Devices with Relay Modules



[<https://www.seeedstudio.com/blog/2020/01/03/arduino-tutorial-control-high-voltage-devices-with-relay-modules/>]

Features

- High temperature resistant plastic shell
- High voltage load
- Low power consumption
- Long lasting
- Optional I2c address
 - 0x00 ~ 0x7F

Specification

Item	Value
Working voltage	5V
Nominal Coil Current	89.3mA
TUV Certification Load	10A 250VAC/ 10A 30VDC
UL Certification Load	10A 125VAC 28VDC
Max. Allowable Voltage	250VAC/110VDC
Power Consumption	abt. 0.45W
Contact Resistance	100mΩ Max.
Insulation Resistance	100MΩ Min. (500VDC)
Max. ON/OFF Switching	30 operation/min
Ambient Temperature	-40°C to +85°C
Operating Humidity	45% to 85% r.h.
Contact Material	AgCdO
Input Interface	I ² C
Default I ² C Address	0x11 or 0x12
Available I ² C Address	0x00 ~ 0x7F
Output interface	3 Pins DIP Female Screw Terminal-Green

**Tip**

For the load parameter, we provide two sets of certification data. Actually, the max. load is 10A 250VAC/10A 30VDC.

Applications

- Domestic appliance
- office machine
- Remote control TV receiver
- monitor display
- audio equipment high rushing current use application

Getting Started

Platforms Supported

Arduino

Raspberry
Pi**Caution**

The platforms mentioned above as supported is/are an indication of the module's software or theoretical 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.

Play With Arduino

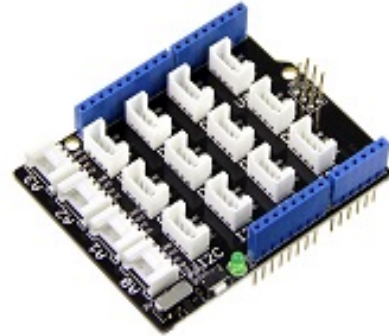
Hardware

Materials required

Seeeduino V4.2



Base Shield



[Get One Now](#)

[<https://www.seeedstudio.com/Seeeduino-V4.2-p-2517.html>]

[Get One Now](#)

[<https://www.seeedstudio.com/Base-Shield-V2-p-1378.html>]

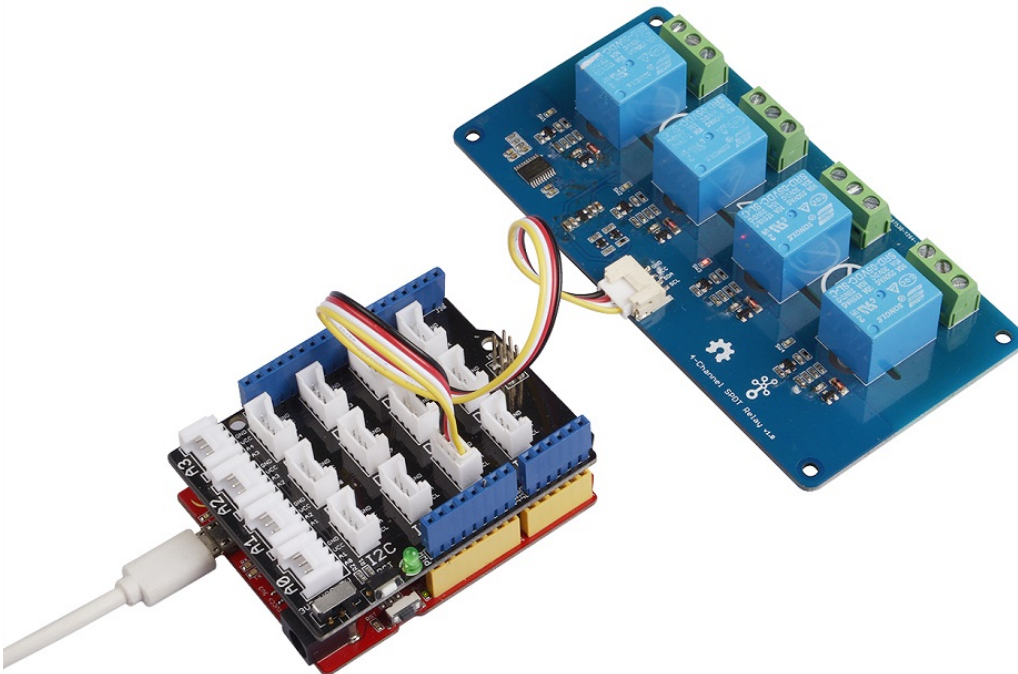


Note

- 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](https://www.seeedstudio.com/Micro-USB-Cable-48cm-p-1475.html) [<https://www.seeedstudio.com/Micro-USB-Cable-48cm-p-1475.html>] to buy.
- Each Grove module comes with a Grove cable when you buy. In case you lose the Grove cable, you can click [here](#)

[<https://www.seeedstudio.com/Grove-Universal-4-Pin-Buckled-20cm-Cable-%285-PCs-pack%29-p-936.html>] to buy.

Hardware Overview

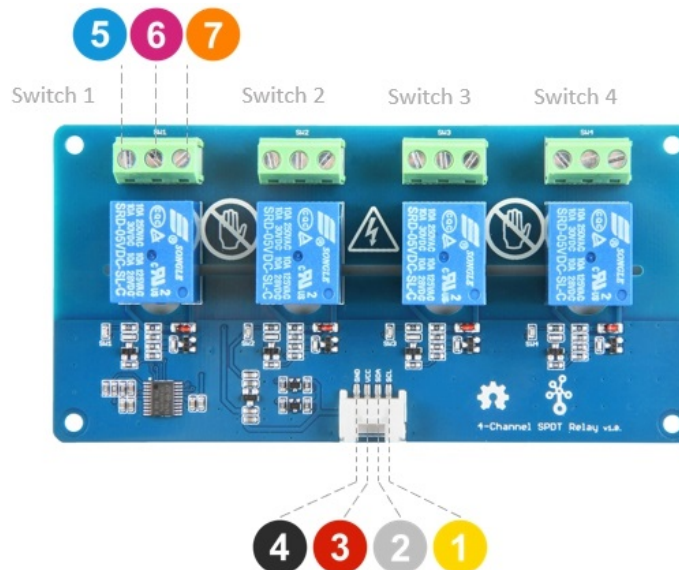


Note

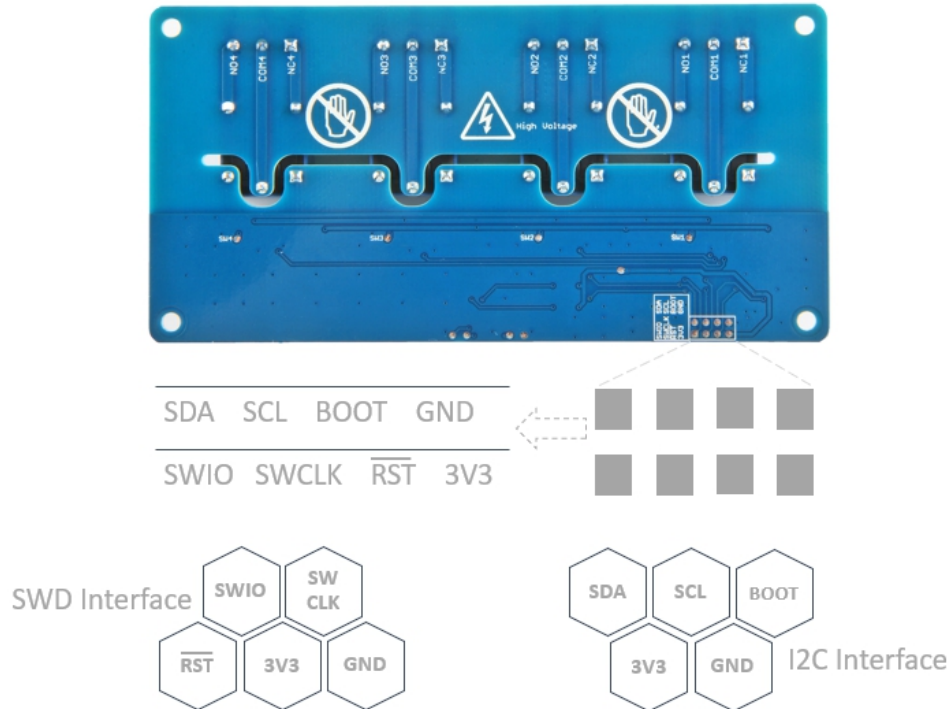
If we don't have Grove Base Shield, We also can directly connect this module to Seeeduino as below.

Seeeduino	Grove - 4-Channel SPDT Relay
5V	Red
GND	Black
SDA	White
SCL	Yellow

Pin Map



- ④ GND: connect this module to the system GND
- ③ VCC: you can use 5V for this module
- ② SDA: a bidirectional input/output pin for data transmit.
- ① SCL: a clock input pin, provide time base
- ⑤ NC1: one throw, connected to COM1 by default
- ⑥ COM1: controlled by SIG1, connected to NC1 or NO1
- ⑦ NO1: the other throw of switch1



Note

- The switch 1-4 have the same pin function, so for the other switches, you can refer to **NC1/COM1/NO1**.
- On the back of the PCB, there are two interfaces: SWD and I²C. The SWD interface is used by default when programming firmware, if you want to use the I²C (actually work as the boot UART), you should set the **BOOT** High.

- **Step 1.** Connect the Grove - 4-Channel SPDT Relay to the I²C port of the Base Shield.
- **Step 2.** Plug Grove - Base Shield into Seeeduino.
- **Step 3.** Connect Seeeduino to PC via a USB cable.

Software



Attention


```
1  #include <multi_channel_relay.h>
2
3  Multi_Channel_Relay relay;
4
5  void setup()
6  {
7      Serial.begin(9600);
8      while(!Serial);
9
10     /* Scan I2C device detect device address */
11     uint8_t old_address = relay.scanI2CDevice();
12     if((0x00 == old_address) || (0xff == old_address)) {
13         while(1);
14     }
15
16     Serial.println("Start write address");
17     relay.changeI2CAddress(old_address, 0x11); /* Set I2C
18     Serial.println("End write address");
19
20     /* Read firmware version */
21     Serial.print("firmware version: ");
22     Serial.print("0x");
23     Serial.print(relay.getFirmwareVersion(), HEX);
24     Serial.println();
25 }
26
27 void loop()
28 {
29
30     /**
31     *  channle: 8 7 6 5 4 3 2 1
32     *  state: 0b00000000 -> 0x00 (all off)
33     *  state: 0b11111111 -> 0xff (all on)
34     */
35
36     /* Begin Controlling Relay */
37     Serial.println("Channel 1 on");
38     relay.turn_on_channel(1);
39     delay(500);
40     Serial.println("Channel 2 on");
41     relay.turn_off_channel(1);
```

```
42 relay.turn_on_channel(2);
43 delay(500);
44 Serial.println("Channel 3 on");
45 relay.turn_off_channel(2);
46 relay.turn_on_channel(3);
47 delay(500);
48 Serial.println("Channel 4 on");
49 relay.turn_off_channel(3);
50 relay.turn_on_channel(4);
51 delay(500);
52 relay.turn_off_channel(4);
53
54 relay.channelCtrl(CHANNLE1_BIT |
55                  CHANNLE2_BIT |
56                  CHANNLE3_BIT |
57                  CHANNLE4_BIT);
58 Serial.print("Turn all channels on, State: ");
59 Serial.println(relay.getChannelState(), BIN);
60
61 delay(2000);
62
63 relay.channelCtrl(CHANNLE1_BIT |
64                  CHANNLE3_BIT);
65 Serial.print("Turn 1 3 channels on, State: ");
66 Serial.println(relay.getChannelState(), BIN);
67
68 delay(2000);
69
70 relay.channelCtrl(CHANNLE2_BIT |
71                  CHANNLE4_BIT);
72 Serial.print("Turn 2 4 channels on, State: ");
73 Serial.println(relay.getChannelState(), BIN);
74
75 delay(2000);
76
77
78 relay.channelCtrl(0);
79 Serial.print("Turn off all channels, State: ");
80 Serial.println(relay.getChannelState(), BIN);
81
82 delay(2000);
```

83 }

**Attention**

The library file may be updated. This code may not be applicable to the updated library file, so we recommend that you use the first methods.

- **Step 4.** Upload the demo. If you do not know how to upload the code, please check [How to upload code](https://wiki.seeedstudio.com/Upload_Code/) [https://wiki.seeedstudio.com/Upload_Code/].
- **Step 5.** Open the **Serial Monitor** of Arduino IDE by click **Tool->Serial Monitor**. Or tap the `Ctrl + Shift + M` key at the same time.

**Success**

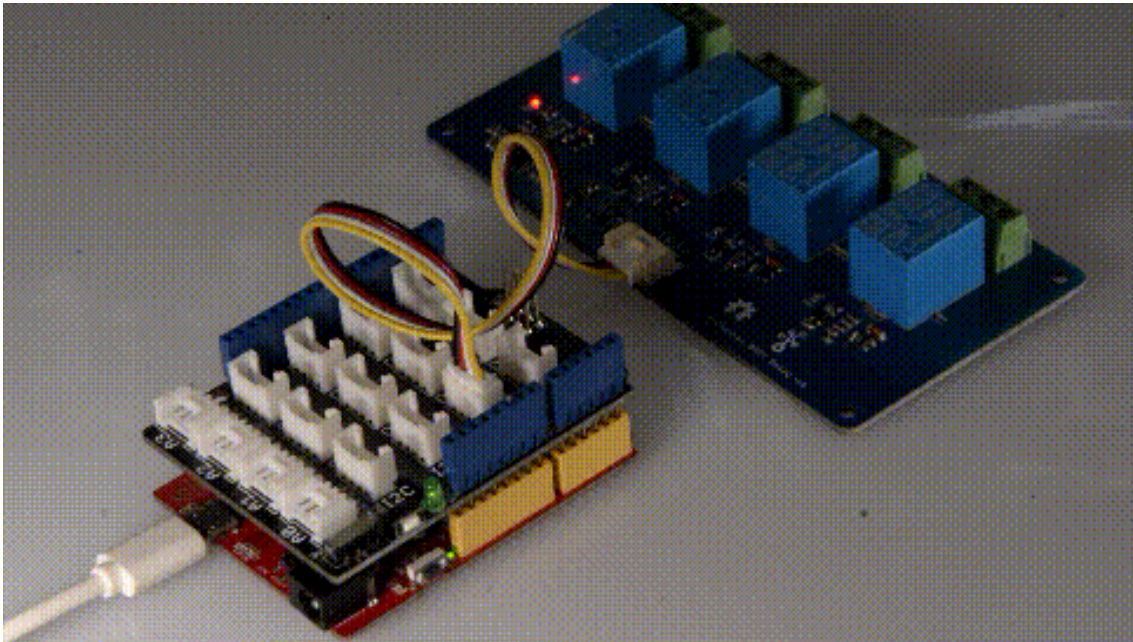
If every thing goes well, you will get the result. Meanwhile, you will see the on-board LEDs alternately lit and extinguished.

```
1 Scanning...
2 I2C device found at address 0x12 !
3 Found 1 I2C devices
4 Start write address
5 End write address
6 firmware version: 0x1
7 Channel 1 on
8 Channel 2 on
9 Channel 3 on
10 Channel 4 on
11 Turn all channels on, State: 1111
12 Turn 1 3 channels on, State: 101
13 Turn 2 4 channels on, State: 1010
14 Turn off all channels, State: 0
15 Channel 1 on
16 Channel 2 on
```



**Success**

The Grove - 4-Channel SPDT Relay will be working as below if everything goes well.

**Note**

We do not add load in this demo, if you want to check how to add load, please check the [Grove - 2-Channel SPDT Relay](https://wiki.seeedstudio.com/Grove-2-Channel_SPDT_Relay/) [https://wiki.seeedstudio.com/Grove-2-Channel_SPDT_Relay/].

Function description

Function	Description
changeI2CAddress(uint8_t old_addr, uint8_t new_addr)	change the device address, the old_addr is the address which you want to use. The new address is the address you are entering the correct old address.
scanI2CDevice()	get the old_addr (current address)
getChannelState()	get the state of every channel, for instance "turned on"
getFirmwareVersion()	get the firmware version burn into the on board
channelCtrl(uint8_t state)	to change all channels you picked immediately CHANNLE1_BIT or 0x01 CHANNLE2_BIT or 0x02 CHANNLE3_BIT or 0x04 CHANNLE4_BIT or 0x08 e.g. channelCtrl(CHANNLE2_BIT CHANNLE3_BIT) channelCtrl(01 02 08) , will turn on the channels 2 and 3. channelCtrl(0) , will turn off all the channels.
turn_on_channel(uint8_t channel)	to turn on the single channel. e.g. turn_on_channel(3) , will turn on the channel 3.
turn_off_channel(uint8_t channel)	to turn off the single channel. e.g. turn_off_channel(3) , will turn off the channel 3.



In case you want to change the address, you need to set the address before use. For example, we want to change it into 0x2f. We can use the following code.

```
1  #include <multi_channel_relay.h>
2
3  Multi_Channel_Relay relay;
4
5  void setup()
6  {
7      Serial.begin(9600);
8      while(!Serial);
9
10     /* Scan I2C device detect device address */
11     uint8_t old_address = relay. ;
12     if((0x00 == old_address) || (0xff == old_address)) {
13         while(1);
14     }
15
16     Serial.println("Start write address");
17     relay.changeI2CAddress(old_address,0x2f); /* Set I2C
18     Serial.println("End write address");
19
20     /* Read firmware version */
21     Serial.print("firmware version: ");
22     Serial.print("0x");
23     Serial.print(relay.getFirmwareVersion(), HEX);
24     Serial.println();
25 }
```

FAQ

Q1: How to burn the firmware?

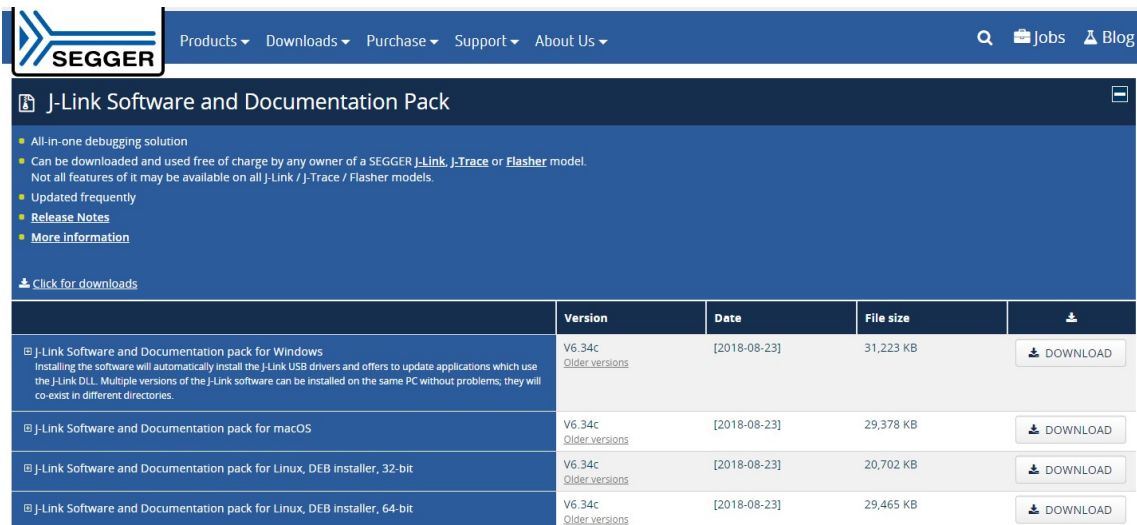
A1: We recommend you use the J-Link burner and the WSD interface to burn the firmware.

You can download the firmware here:

Factory firmware [https://files.seeedstudio.com/wiki/Grove-4-Channel_SPDT_Relay/res/Grove-4-Channel-SPDT-Relay-Firmware.bin]

We recommed you use the J-flash for the software:

J-flash [<https://www.segger.com/downloads/jlink#J-LinkSoftwareAndDocumentationPack>]



The screenshot shows the SEGGER website's 'J-Link Software and Documentation Pack' page. The page features a navigation bar with 'Products', 'Downloads', 'Purchase', 'Support', and 'About Us'. Below the navigation bar, there is a search icon, 'Jobs', and 'Blog' links. The main content area includes a list of features and a table of download links.

	Version	Date	File size	
<ul style="list-style-type: none"> All-in-one debugging solution Can be downloaded and used free of charge by any owner of a SEGGER J-Link, J-Trace or Flasher model. Not all features of it may be available on all J-Link / J-Trace / Flasher models. Updated frequently Release Notes More information <p>Click for downloads</p>	V6.34c Older versions	[2018-08-23]	31,223 KB	DOWNLOAD
J-Link Software and Documentation pack for Windows Installing the software will automatically install the J-Link USB drivers and offers to update applications which use the J-Link DLL. Multiple versions of the J-Link software can be installed on the same PC without problems, they will co-exist in different directories.	V6.34c Older versions	[2018-08-23]	29,378 KB	DOWNLOAD
J-Link Software and Documentation pack for macOS	V6.34c Older versions	[2018-08-23]	20,702 KB	DOWNLOAD
J-Link Software and Documentation pack for Linux, DEB installer, 32-bit	V6.34c Older versions	[2018-08-23]	29,465 KB	DOWNLOAD
J-Link Software and Documentation pack for Linux, DEB installer, 64-bit	V6.34c Older versions	[2018-08-23]		DOWNLOAD

Schematic Online Viewer



!!!Product Change Note:

Because ST32 series chips are out of stock globally, prices have increased several times and there is no clear delivery date. We have no choice but to switch to the MM32 chip. The specific replacement models are as follows: STM32F030F4P6TR is replaced by MM32F031F6P6. After the chip is replaced, the product functions, features, usage methods and codes remain unchanged. It should be noted that the firmware version has changed, and the factory firmware has been adjusted according to different chips. If you

need to re-burn the firmware, please download the firmware corresponding to the chip.

Resources

- **[Zip]** [Grove-4-Channel SPDT Relay eagle files](https://files.seeedstudio.com/wiki/Grove-4-Channel_SPDT_Relay/res/Grove-4-Channel_SPDT_Relay.zip)
[https://files.seeedstudio.com/wiki/Grove-4-Channel_SPDT_Relay/res/Grove-4-Channel_SPDT_Relay.zip]
- **[Bin]** [STM32F030F4P6TR-Firmware](https://files.seeedstudio.com/wiki/Grove-4-Channel_SPDT_Relay/res/Grove-4-Channel-SPDT-Relay-Firmware.bin)
[https://files.seeedstudio.com/wiki/Grove-4-Channel_SPDT_Relay/res/Grove-4-Channel-SPDT-Relay-Firmware.bin]
- **[Bin]** [MM32F031F6P6-Firmware](https://files.seeedstudio.com/wiki/Grove-4-Channel_SPDT_Relay/res/firmware-spdt-4-channels.ino.bin)
[https://files.seeedstudio.com/wiki/Grove-4-Channel_SPDT_Relay/res/firmware-spdt-4-channels.ino.bin]
- **[PDF]** [Datasheet of SRD 05VDC-SL-C Relay](https://files.seeedstudio.com/wiki/Grove-2-Channel_SPDT_Relay/res/SRD_05VDC-SL-C.pdf)
[https://files.seeedstudio.com/wiki/Grove-2-Channel_SPDT_Relay/res/SRD_05VDC-SL-C.pdf]
- **[PDF]** [Datasheet of S9013](https://files.seeedstudio.com/wiki/Grove-2-Channel_SPDT_Relay/res/Transistors_NPN_25V-500mA.pdf)
[https://files.seeedstudio.com/wiki/Grove-2-Channel_SPDT_Relay/res/Transistors_NPN_25V-500mA.pdf]
- **[PDF]** [Datasheet of STM32](https://files.seeedstudio.com/wiki/Grove-4-Channel_SPDT_Relay/res/STM32F030F4P6.pdf)
[https://files.seeedstudio.com/wiki/Grove-4-Channel_SPDT_Relay/res/STM32F030F4P6.pdf]
- **[PDF]** [MM32F031F6P6_Datasheet.pdf](https://files.seeedstudio.com/wiki/Grove-4-Channel_SPDT_Relay/res/MM32F031F6P6_Datasheet.pdf)
[https://files.seeedstudio.com/wiki/Grove-4-Channel_SPDT_Relay/res/MM32F031F6P6_Datasheet.pdf]

Project

This is the introduction Video of this product, simple demos, you can have a try.

Grove - 4-Channel SPDT Relay - #newproductsTu...



Home Automation and Monitoring: An RSL10 Sense DB based system that can monitor and control the temperature, humidity, and light intensity by Smartphone app and Alexa.



Grove - 4-Channel SPDT Relay: Hey Seekers!!! This is the second week of our #newproductsTuesday segment. There will be a product showcase of our new Grove - 4-Channel SPDT Relay in this video and a cool interesting demo as well.



Tech Support

Please do not hesitate to submit the issue into our [forum](https://forum.seeedstudio.com/)
[<https://forum.seeedstudio.com/>].



[https://www.seeedstudio.com/act-4.html?utm_source=wiki&utm_medium=wikibanner&utm_campaign=newproducts]

