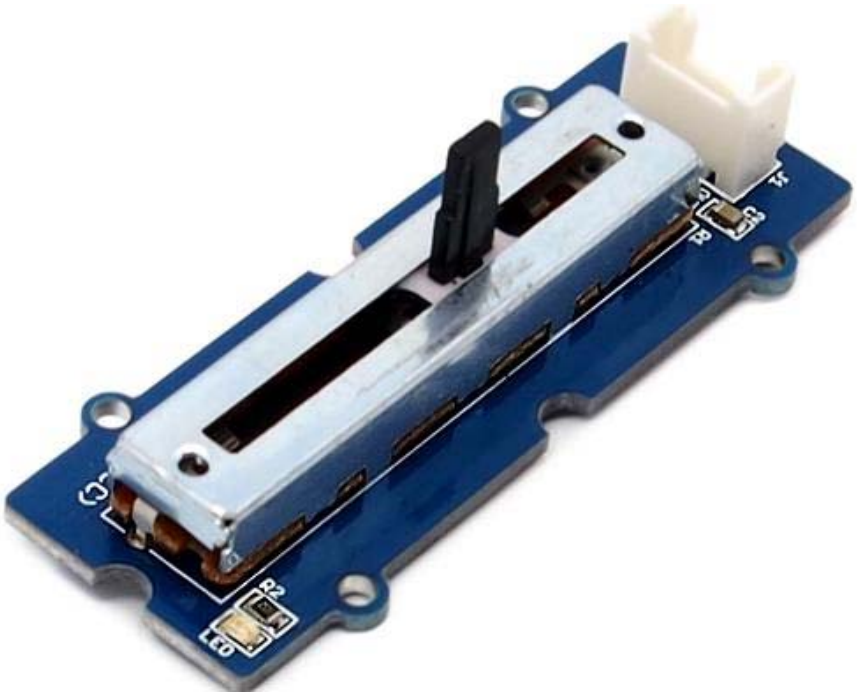
Grove - Slide Potentiometer



The Grove - Slide Potentiometer module incorporates a linear variable resistor with a maximum resistance of 10KΩ. When you move the slider from one side to the other, its output voltage will range from 0 V to the Vcc you apply. It connects to the other Grove modules through a standard 4-Pin **Grove Cable**. Three of the pins are connected to OUT (Pin 1), Vcc (Pin 3) and GND (Pin 4), while the fourth pin (Pin 2) is connected to a on-board green indicator LED. The LED is used to visually represent the resistance change on the potentiometer.

Features

- 30 mm long slide length
- Linear resistance taper
- Grove compatible

Tip
More details about Grove modules please refer to Grove System

Application Ideas

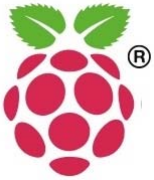
Here are some projects for your reference.

Arduino BoomBox	Arduino BeatBox
	

Specifications

Item	Min	Typical	Max
Voltage (DC)	3.3V	5.0V	30V
Current	-	-	30mA
Dimension	24mm x60mm		
Net Weight	8.6g		
Rotational life	>15,000 cycles		
Total resistance	10KΩ		
Stroke length	30mm		
Total resistance tolerance	+/- 20%		

Platforms Supported

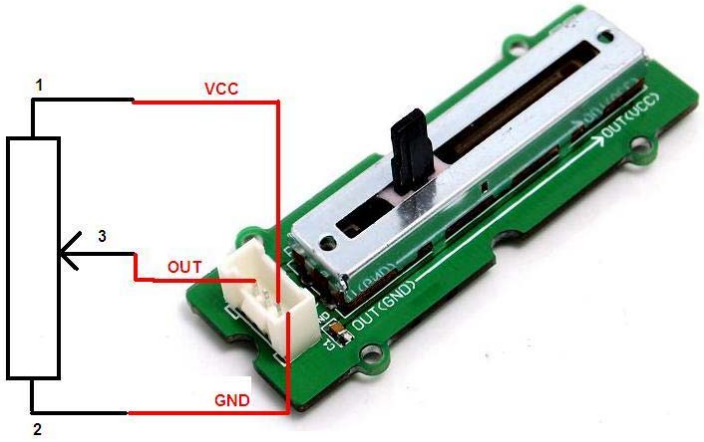
Arduino	Raspberry Pi	BeagleBone	Wio	LinkIt ONE
				

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

As an Adjustable Resistor

As shown below, the Grove - Slide Potentiometer can be used as a simple slide potentiometer in any MCU controlled or stand-alone project.



Standalone

Follow these steps to build a sample Grove circuit using this module but without using any microcontroller board:

1. Connect the slide potentiometer module to the input side of your circuit (to the left of the power module). On the output side of the circuit, you may use a range of User Interface modules (Grove - Red LED, Grove - LED String Light, Grove - Mini Fan, Grove - Buzzer, Grove - Recorder etc.)

2. Power up the circuit when complete.
3. The slide potentiometer module can now be used to trigger an output. For example:
    - When used in conjunction with a Grove - Red LED output module, observe that the brightness of the LED increases as you move the slider from GND to Vcc. At Vcc, the resistance of the potentiometer is minimum and the LED burns the brightest. The same behavior can be seen when the slide potentiometer is used with the Grove - LED String Light module - the more voltage you apply by taking the slider towards the Vcc mark, the brighter the LED lights would become.
    - Similarly, you can use the slide potentiometer to vary the speed of your Grove - Mini Fan or the frequency with which your Grove - Buzzer module sounds
    - The slide potentiometer can also be used as an ON/OFF switch for any circuit. Take the slider to the Vcc position to switch it ON and move it down to GND to switch OFF a circuit.

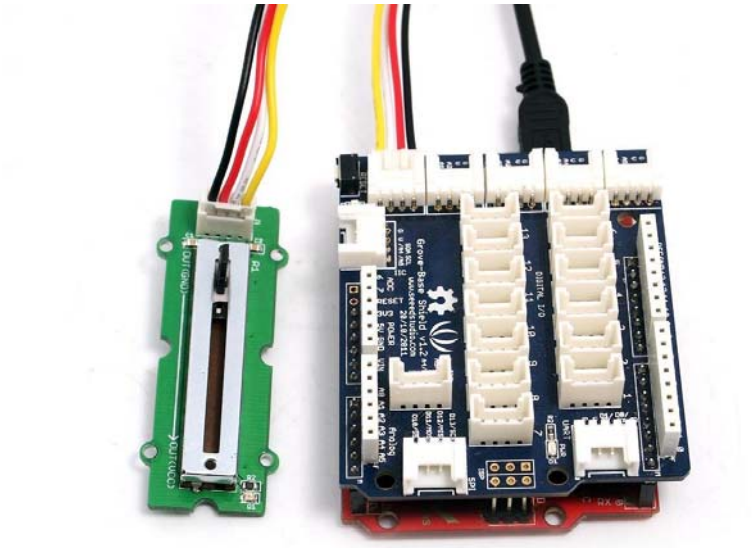
In terms of choosing a power module, you can use either the Grove - USB Power module or the Grove - DC Jack Power module for building standalone Grove circuits.

With Arduino

As a Voltage Divider

Follow these simple steps to make the slide potentiometer module function as a voltage divider:

1.When using the module in conjunction with an Arduino or a Seeeduino, use the Grove - Base Shield and connect the Grove - Slide Potentiometer module to the shield using a designated Grove Interface (e.g. Analog Port 0 as shown below).

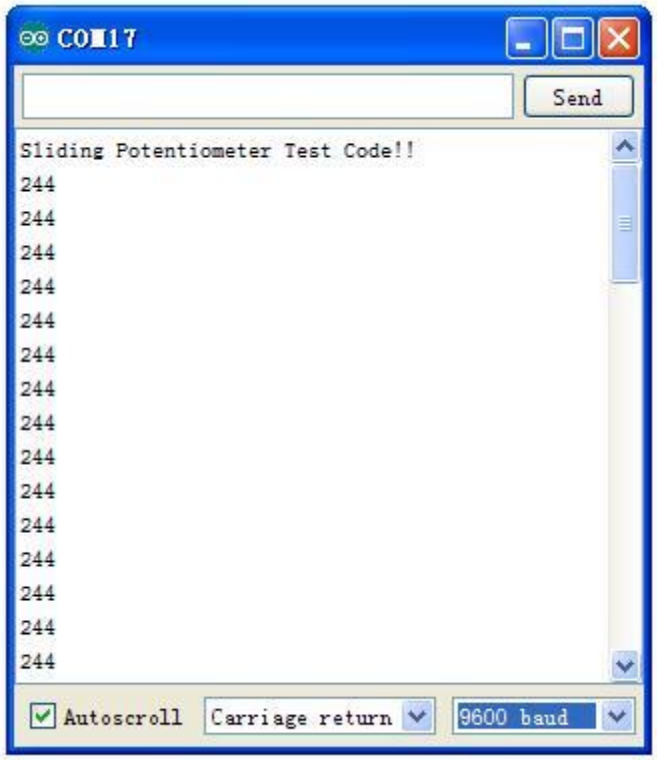


2.Connect the board to PC using USB cable.

3.Upload the following sample sketch.

```
int adcPin = A0; // select the input pin for the potentiometer
int ledPin = A1; // select the pin for the LED
int adcIn = 0;   // variable to store the value coming from the sensor
void setup()
{
    Serial.begin(9600); // init serial to 9600b/s
    pinMode(ledPin, OUTPUT); // set ledPin to OUTPUT
    Serial.println("Sliding Potentiometer Test Code!!");
}
void loop()
{
    // read the value from the sensor:
    adcIn = analogRead(adcPin);
    if(adcIn &gt;= 500) digitalWrite(ledPin,HIGH);  // if adc in &gt; 500,
led light
    else digitalWrite(ledPin, LOW);
    Serial.println(adcIn);
    delay(100);
  }
```

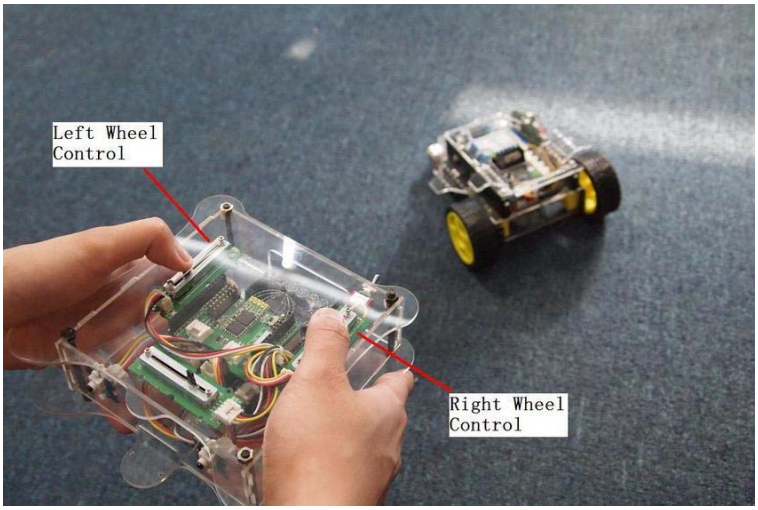
4.Open the serial monitor. You should see some data from ADC.



5.Move the lever back and forth. The serial data will change correspondingly. When the output resistance exceeds a certain preset value, the on-board indicator LED will also light up.

As an HID Device

Slide Potentiometer can be an effective Human Interface Device (HID) and can be used, for example, in the radio controller of a Radio Controlled toy car. The picture below shows two Slide Potentiometers on the control panel - one to control the speed of the left wheel, and the other to control the speed of the right wheel of the toy car respectively. Now you can change the speeds of both motors and see the behavior. You will see that if you make the right wheel spin faster than the left wheel, the car will turn rightwards, and if you make the left wheel spin faster than the right wheel, the car will turn leftwards.



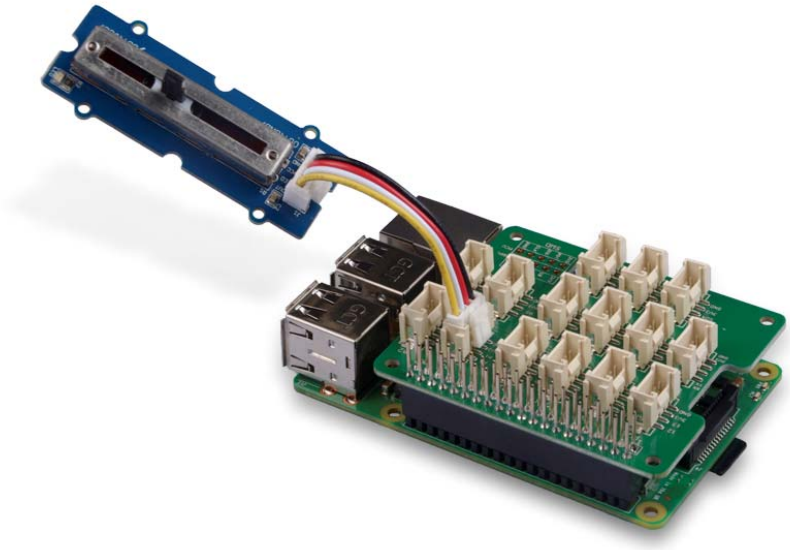
Play With Raspberry Pi (With Grove Base Hat for Raspberry Pi)

Hardware

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

Raspberry pi	Grove Base Hat for RasPi	Grove - Slide Potentiometer
		

- **Step 2**. Plug the Grove Base Hat into Raspberry.
- **Step 3**. Connect the Slide Potentiometer to A0 port 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 slide potentiometer to **any Analog 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_slide_potentiometer.py 0
```

Following is the grove_slide_potentiometer.py code.

```
import math
import sys
import time
from grove.adc import ADC


class GroveSlidePotentiometer(ADC):
    def __init__(self, channel):
        self.channel = channel
        self.adc = ADC()

```

```
    @property
    def value(self):
        return self.adc.read(self.channel)


Grove = GroveSlidePotentiometer


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

    sensor = GroveSlidePotentiometer(int(sys.argv[1]))

    while True:
        print('Slide potentiometer value: {}'.format(sensor.value))
        time.sleep(.2)


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

Success

If everything goes well, you will be able to see the following result

```
pi@raspberrypi:~/grove.py/grove $ python grove_slide_potentiometer.py 0
Slide potentiometer value: 987
Slide potentiometer value: 988
Slide potentiometer value: 986
Slide potentiometer value: 8
Slide potentiometer value: 2
Slide potentiometer value: 0
Slide potentiometer value: 1
Slide potentiometer value: 0
Slide potentiometer value: 24
Slide potentiometer value: 0
Slide potentiometer value: 0
Slide potentiometer value: 11
Slide potentiometer value: 995
Slide potentiometer value: 999
Slide potentiometer value: 999
^CTraceback (most recent call last):
  File "grove_slide_potentiometer.py", line 66, in <module>
    main()
  File "grove_slide_potentiometer.py", line 62, in main
    time.sleep(.2)
KeyboardInterrupt
```

You can quit this program by simply press `Ctrl` + `C` .

Notice
You may have noticed that for the analog port, the silkscreen pin number is something like **A0, A1**, however in the command we use parameter **0** and **1**, just the same as digital port. So please make sure you plug the module into the correct port, otherwise there may be pin conflicts.

Play With Raspberry Pi (with GrovePi_Plus)

1.You should have got a raspberry pi and a grovepi or grovepi+.

2.You should have completed configuring the development enviroment, otherwise follow here.

3.Connection

- Plug the sensor to grovepi socket A0 by using a grove cable.

4.Navigate to the demos' directory:

```
cd yourpath/GrovePi/Software/Python/
```

- To see the code

```
nano grove_slide_potentiometer.py   # "Ctrl+x" to exit #
```

```
import time
import grovepi

# Connect the Grove Slide Potentiometer to analog port A0
# OUT,LED,VCC,GND
slide = 0   # pin 1 (yellow wire)

# The device has an onboard LED accessible as pin 2 on port A0
# OUT,LED,VCC,GND
led = 1     # pin 2 (white wire)

grovepi.pinMode(slide,"INPUT")
grovepi.pinMode(led,"OUTPUT")
time.sleep(1)

while True:
    try:
        # Read sensor value from potentiometer
        sensor_value = grovepi.analogRead(slide)

        # Illuminate onboard LED
        if sensor_value > 500:
```

```
            grovepi.digitalWrite(led,1)
        else:
            grovepi.digitalWrite(led,0)

        print "sensor_value =", sensor_value

    except IOError:
        print "Error"
```

5.Run the demo.

```
sudo python grove_slide_potentiometer.py
```

Resources

- Sliding Potentiometer Eagle File
  https://raw.githubusercontent.com/SeeedDocument/Grove-Slide_Potentiometer/master/res/Sliding_Potentiometer.rar
- Sliding Potentiometer in PDF

  https://raw.githubusercontent.com/SeeedDocument/Grove-Slide_Potentiometer/master/res/Sliding_protentiometer_sch.pdf

- Sliding Potentiometer datasheet

  https://raw.githubusercontent.com/SeeedDocument/Grove-Slide_Potentiometer/master/res/Sliding_potentiometer_datasheet.pdf

Projects

Raspberry pi music server: A first step to Raspberry Pi project

Tech Support

Please submit any technical issue into our forum.