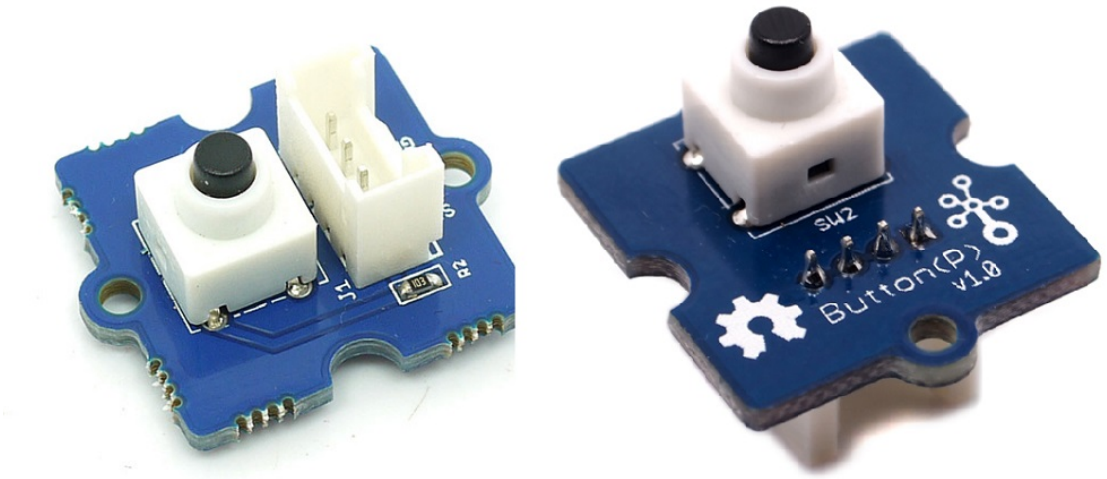
Grove - Button



Grove - Button is a momentary push button. It contains one independent "momentary on/off" button. “Momentary” means that the button rebounds on its own after it is released. The button outputs a HIGH signal when pressed, and LOW when released. The Sig marked on silk layer stands for signal while NC stands for not used at all. There are two versions of this button available as showed in the pictures. The only difference is the direction of the Grove socket.

Get One Now

[https://www.seeedstudio.com/Grove-Button-p-766.html]

Version

Product Version	Changes	Released Date
Grove-Button	Initial	Nov 25 2010

Features

- Easy to use momentary ON/OFF button
- Uses Standard 4-pin Grove Cables

Tip
More details about Grove modules please refer to Grove System [https://wiki.seeedstudio.com/Grove_System/]

Specification

Parameter	Value/Range
Operating voltage	3.3⁄5V
Electrical Life	200,000 cycles
Operation Force	100 ± 50gf
Operation Temperature	-25℃ to +70℃
Size	20mmX20mm

Platforms Supported

Arduino	Raspberry Pi

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

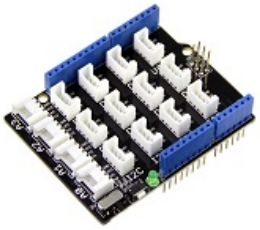
compatibility. We only provide software library or code examples for Arduino platform in most cases. It is not possible to provide software library / demo code for all possible MCU platforms. Hence, users have to write their own software library.

Getting Started

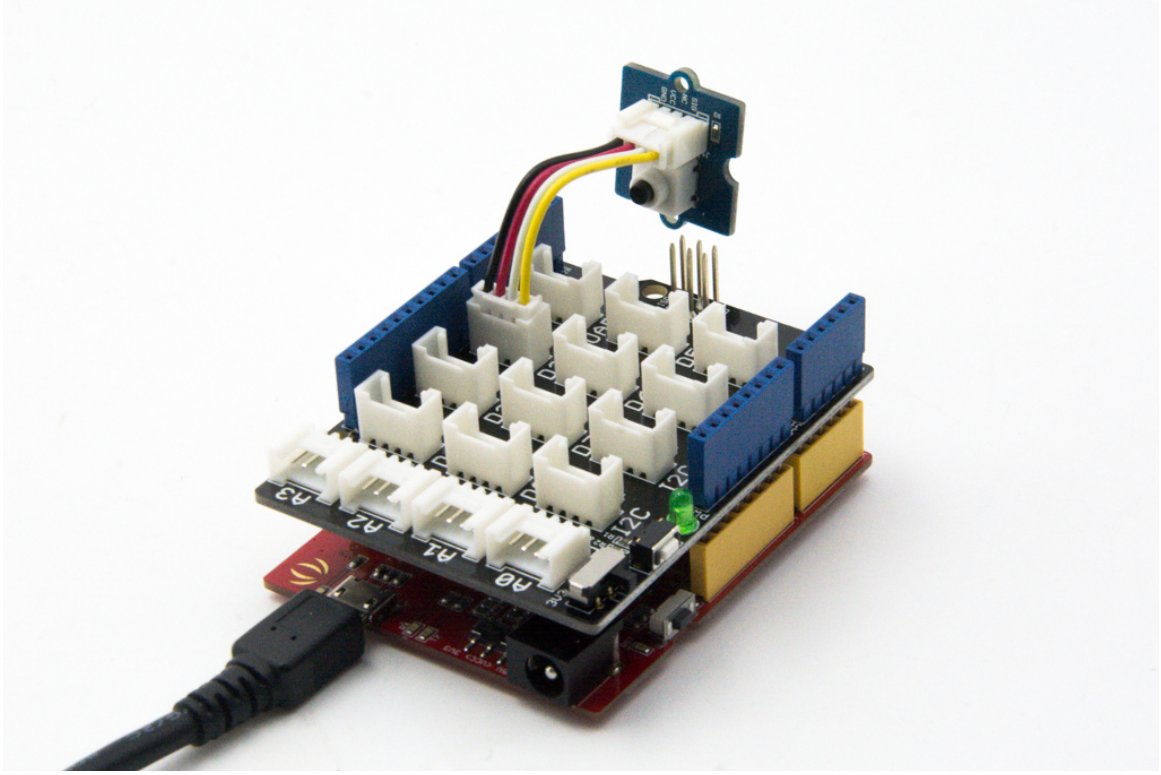
Play With Arduino

Hardware

- Step 1. Prepare the below stuffs:

Seeeduino V4.2	Base Shield	Grove - Button
		
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]	Get ONE Now [https://www.seeedstudio.com/Grove-Button-p-766.html]

- Step 2. Connect Grove-Button to port D2 of Grove-Base Shield.
- Step 3. Plug Grove - Base Shield into Seeeduino.
- Step 4. Connect Seeeduino to PC through a USB cable.



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

Seeeduino	Grove-Button
5V	Red
GND	Black
Not Conencted	White
D2	Yellow

Software

- Step 1. Copy the code into Arduino IDE and upload.

```
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin
```

```
3
4  // variables will change:
5  int buttonState = 0;         // variable for reading the pushbutton statu
6
7  void setup() {
8      // initialize the LED pin as an output:
9      pinMode(ledPin, OUTPUT);
10     // initialize the pushbutton pin as an input:
11     pinMode(buttonPin, INPUT);
12 }
13
14 void loop(){
15     // read the state of the pushbutton value:
16     buttonState = digitalRead(buttonPin);
17
18     // check if the pushbutton is pressed.
19     // if it is, the buttonState is HIGH:
20     if (buttonState == HIGH) {
21         // turn LED on:
22         digitalWrite(ledPin, HIGH);
23     }
24     else {
25         // turn LED off:
26         digitalWrite(ledPin, LOW);
27     }
28 }
```

- Step 2. We will see the on board Pin13 LED on and off.

Play with Codecraft

Hardware

Step 1. Connect a Grove - Button to port D2 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.

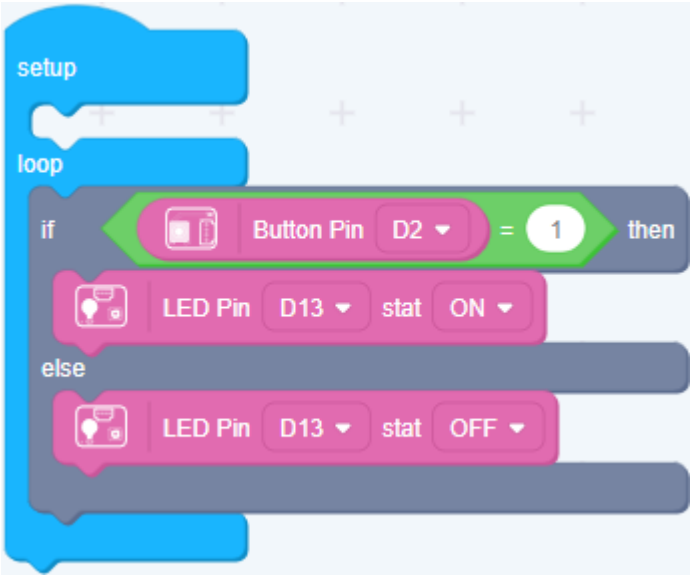
Software

Step 1. Open Codecraft [https://ide.chmakered.com/], 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 [https://wiki.seeedstudio.com/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.



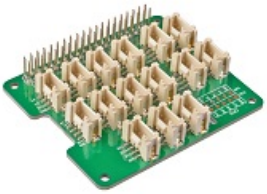
Upload the program to your Arduino/Seeeduino.

Success
When the code finishes uploaded, the LED on the Arduino/Seeeduino board will goes on when the Button pressed.

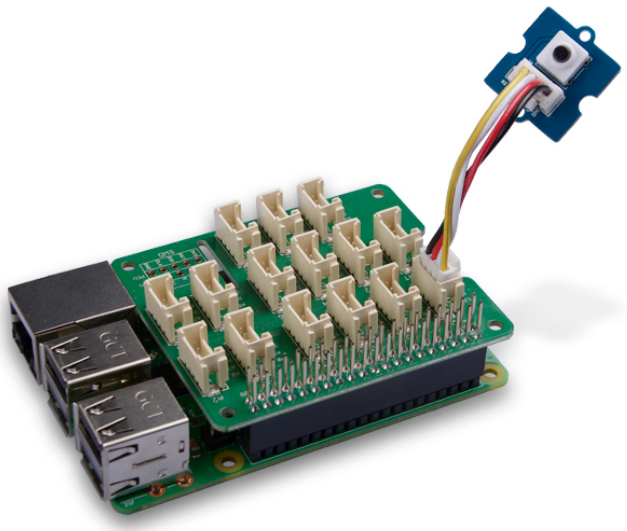
Play With Raspberry Pi (With Grove Base Hat for Raspberry Pi)

Hardware

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

Raspberry pi	Grove Base Hat for RasPi	Grove - Button
		
Get ONE Now [https://www.seeedstudio.com/Raspberry-Pi-3-Model-B-p-2625.html]	Get ONE Now [https://www.seeedstudio.com/Grove-Base-Hat-for-Raspberry-Pi-p-3186.html]	Get ONE Now [https://www.seeedstudio.com/Grove-Button-p-766.html]

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



Software

> **Attention**
> If you are using **Raspberry Pi with Raspberrypi OS >= Bullseye**, you have to use this command line **only with Python3**.

- **Step 1**. Follow Setting Software [https://wiki.seeedstudio.com/Grove_Base_Hat_for_Raspberry_Pi/#installation] 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 command to run the code.

```
cd grove.py/grove
python3 grove_button.py 12
```

If you connect the Red LED to the different port of the Base Hat, instead of excuting **python grove_led.py 12**, you should run the following command.

```
python3 grove_button.py portnumber
```

Following is the grove_button.py code.

```
import time
from grove.button import Button
from grove.factory import Factory


class GroveButton(object):
    def __init__(self, pin):
        # High = pressed
        self.__btn = Factory.getButton("GPIO-HIGH", pin)
        self.__last_time = time.time()
        self.__on_press = None
        self.__on_release = None
        self.__btn.on_event(self, GroveButton.__handle_event)
```

```
14
15      @property
16      def on_press(self):
17          return self.__on_press
18
19      @on_press.setter
20      def on_press(self, callback):
21          if not callable(callback):
22              return
23          self.__on_press = callback
24
25      @property
26      def on_release(self):
27          return self.__on_release
28
29      @on_release.setter
30      def on_release(self, callback):
31          if not callable(callback):
32              return
33          self.__on_release = callback
34
35      def __handle_event(self, evt):
36          dt, self.__last_time = evt["time"] - self.__last_time, evt["time"
37          # print("event index:{} event:{} pressed:{}".format(evt["index"],
38          if evt["code"] == Button.EV_LEVEL_CHANGED:
39              if evt["pressed"]:
40                  if callable(self.__on_press):
41                      self.__on_press(dt)
42              else:
43                  if callable(self.__on_release):
44                      self.__on_release(dt)
45
46
47  Grove = GroveButton
48
49  def main():
50      from grove.helper import SlotHelper
51      sh = SlotHelper(SlotHelper.GPIO)
52      pin = sh.argv2pin()
53
54      button = GroveButton(pin)
55
56      def on_press(t):
57          print('Button is pressed')
58      def on_release(t):
59          print("Button is released, pressed for {0} seconds".format(round(
60
61      button.on_press = on_press
62      button.on_release = on_release
```

```
63
64      while True:
65          time.sleep(1)
66
67
68  if __name__ == '__main__':
69      main()
```

Success

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

```
1   pi@raspberrypi:~/grove.py/grove $ python3 grove_button.py 12
2   Hat Name = 'Grove Base Hat RPi'
3   Button is pressed
4   Button is pressed
5   Button is pressed
6   Button is pressed
7   Button is pressed
8   Button is pressed
9   Button is released, pressed for 0.002685 seconds
10  Button is pressed
11  Button is released, pressed for 0.219019 seconds
12  Button is pressed
13  Button is released, pressed for 0.001372 seconds
14  Button is pressed
15  Button is pressed
16  Button is released, pressed for 0.043143 seconds
17  Button is pressed
18  Button is released, pressed for 1.083292 seconds
19  ^CTraceback (most recent call last):
20    File "grove_button.py", line 103, in <module>
21      main()
22    File "grove_button.py", line 99, in main
23      time.sleep(1)
24  KeyboardInterrupt
```

You can press `Ctrl`+`C` to quit this program.

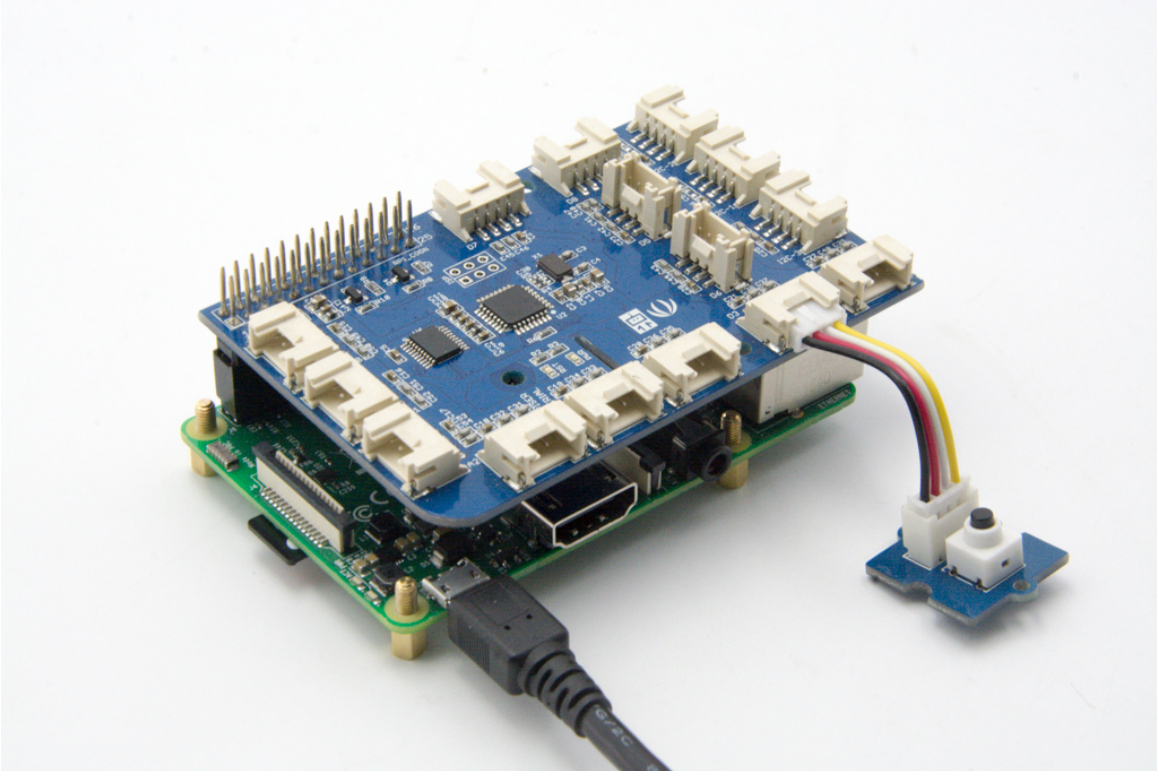
Play With Raspberry Pi(with GrovePi_Plus)

Hardware

- Step 1. Prepare the below stuffs:

Raspberry pi	GrovePi_Plus	Grove - Button
		
Get ONE Now [https://www.seeedstudio.com/Raspberry-Pi-3-Model-B-p-2625.html]	Get ONE Now [https://www.seeedstudio.com/GrovePi%2B-p-2241.html]	Get ONE Now [https://www.seeedstudio.com/Grove-Button-p-766.html]

- Step 2. Plug the GrovePi_Plus into Raspberry.
- Step 3. Connect Grove-Button to D3 port of GrovePi_Plus.
- Step 4. Connect the Raspberry to PC through USB cable.



Software

Attention
If you are using **Raspberry Pi with Raspberrypi OS >= Bullseye**, you have to use this command line **only with Python3**.

- Step 1. Follow Setting Software [https://www.dexterindustries.com/GrovePi/get-started-with-the-grovepi/setting-software/] to configure the development environment.
- Step 2. Git clone the Github repository.

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

- Step 3. Excute below commands.

```
cd ~/GrovePi/Software/Python
python3 grove_button.py
```

Here is the grove_button.py code.

```
import time
import grovepi

# Connect the Grove Button to digital port D3
# SIG,NC,VCC,GND
button = 3

grovepi.pinMode(button,"INPUT")

while True:
    try:
        print(grovepi.digitalRead(button))
        time.sleep(.5)

    except IOError:
        print ("Error")
```

- Step 4. We will see the button on and off.

```
pi@raspberrypi:~/GrovePi/Software/Python $ python3 grove_button.py
```

```
2  0
3  1
4  1
5  1
6  1
7  0
8  0
```

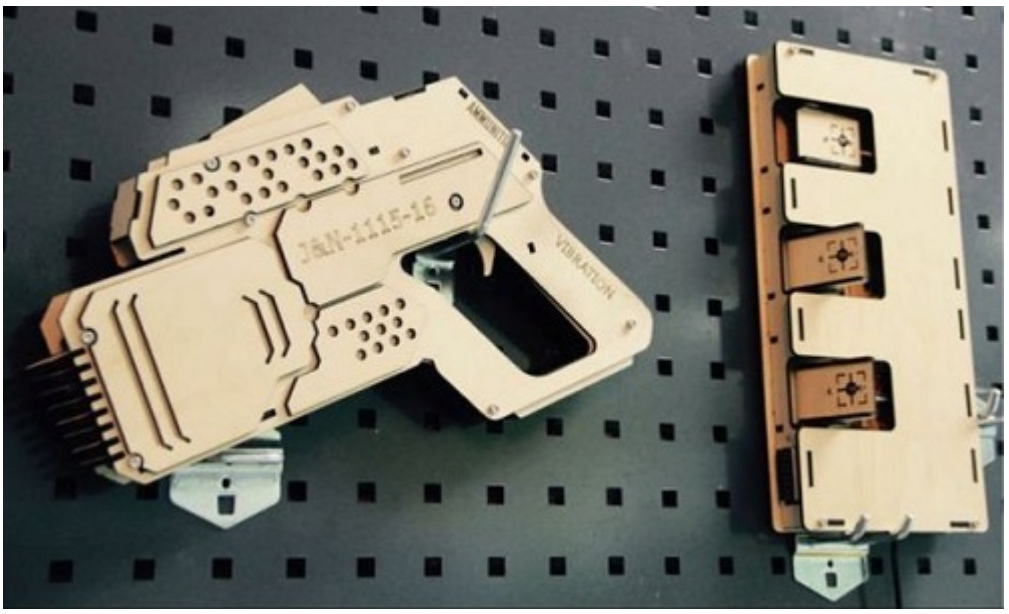
Schematic Online Viewer

Resources

- **[Eagle&PDF]** Grove-Button Eagle Files [https://files.seeedstudio.com/wiki/Grove_Button

/resources/Grove_-_Button_v1.0_Source_File.zip]

- **[More Reading]** Wooden Laser Gun [https://www.instructables.com/id/DIY-a-Wooden-Laser-Gun-As-a-Xmas-Present-for-Your-/]
- **[Codecraft]** CDC File [https://files.seeedstudio.com/wiki/Grove_Button/res/Grove_Button_CDC_File.zip]



Inspired by OVERWATCH, we have made a very cool Wooden Laser Gun toy for fun these day!

The Wooden Laser Gun and the Gun Target are all based on an Arduino board called Seeeduino Lotus. The laser emitter on the Laser Gun is controlled to fire laser pulse to "activate" the Gun Target. And there are 3 light sensors on the Gun Target to detect the laser pulse. It seems very simple right? If you are interested in our project, please make one for yourself or your child! It's worth to spend one day DIY it as a Xmas present.

Projects

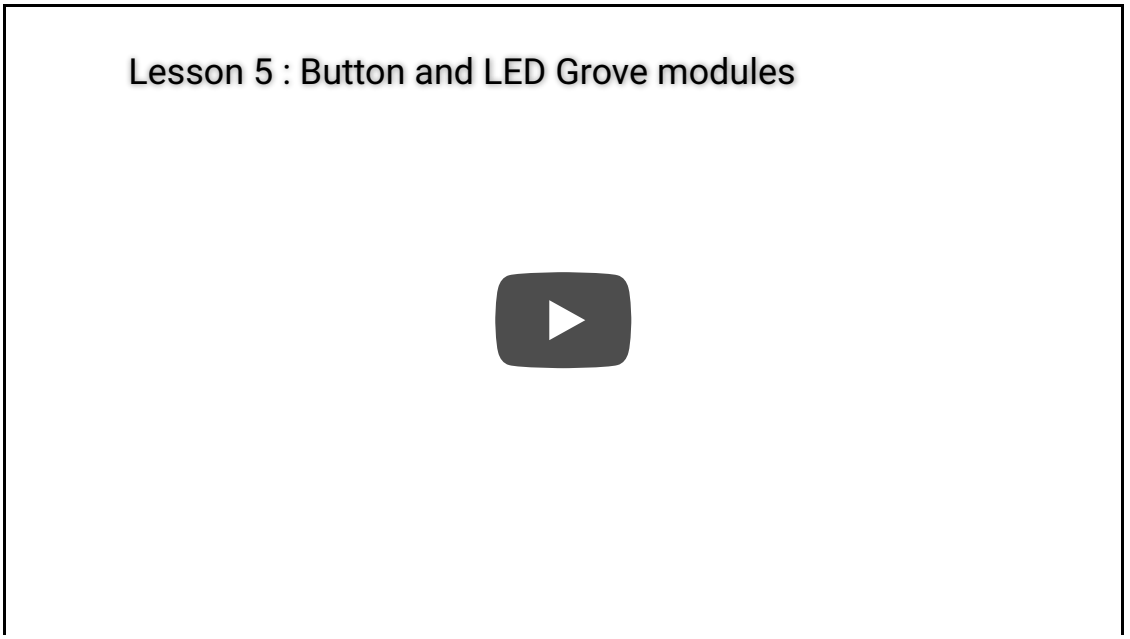
Grove - Introduction in a Button & LED String Light: Beginner-Example - I bet Beginners will smile after project - sent me an selfie!

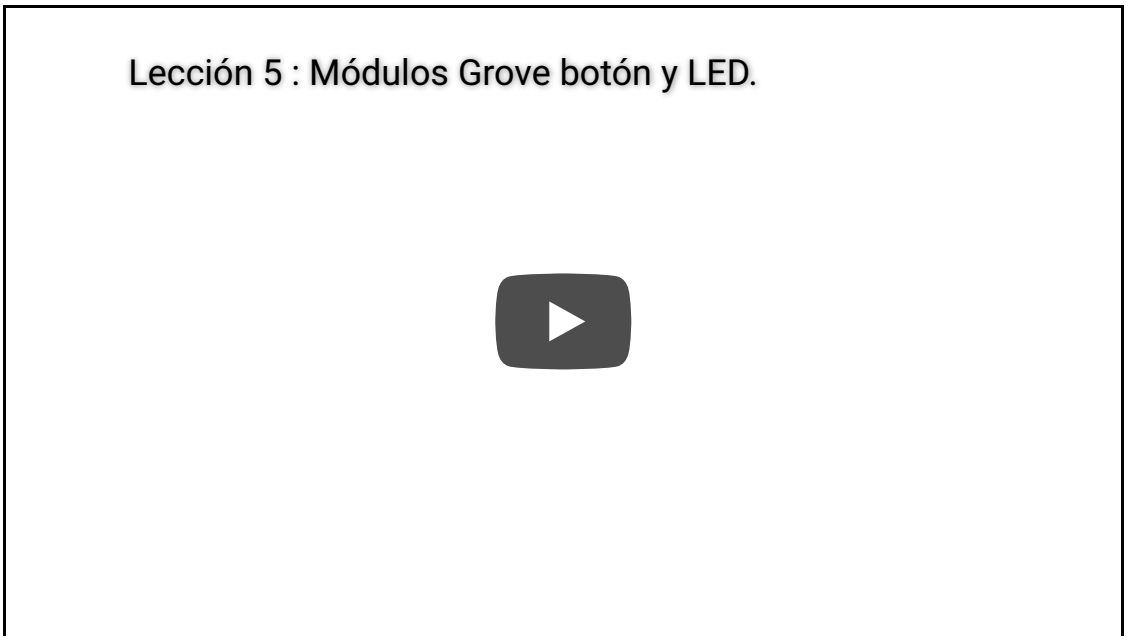
(https://www.hackster.io/ingo-lohs/grove-introduction-in-a-button-led-string-light-f7e4d6)

Using Grove Button To Control Grove LED: How to connect and use Grove Button to control Grove LED socket kit.

(https://www.hackster.io/user50338573/using-grove-button-to-control-grove-led-96d00b)

Button and LED Grove modules:





Tech Support

Please submit any technical issue into our forum [https://forum.seeedstudio.com/].