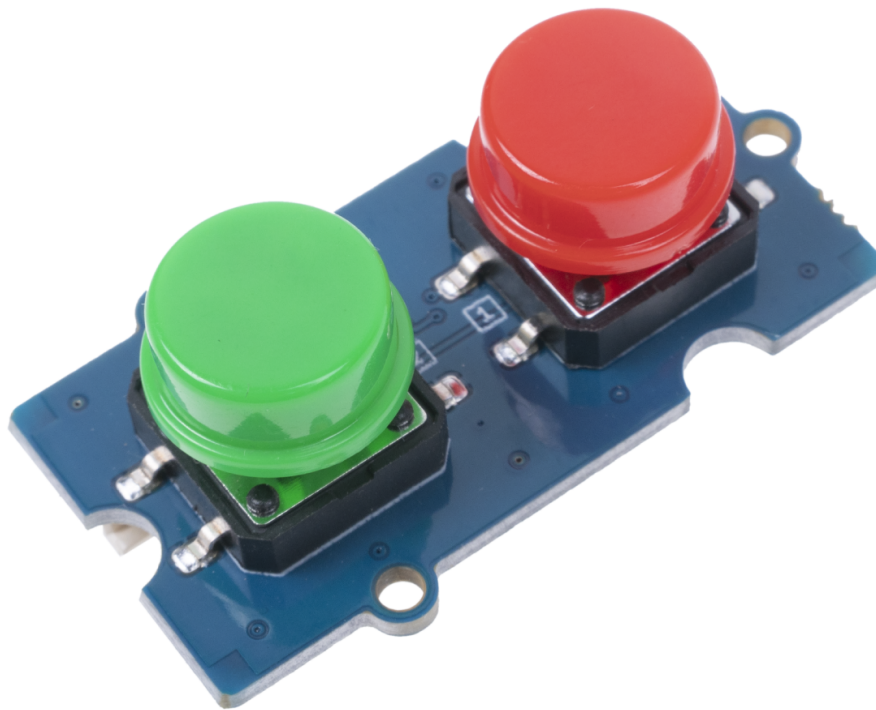


Grove - Dual Button



Grove - Dual Button includes 2 buttons, enables you control two signal channel with one grove module. With 4 different color keycaps provided, you can freely adjust the color combination as needed.

The button is simply driven by the GPIO Digital Pins. When the button is pressed, the Pins can reach a LOW signal from the button;

on the other hand, when it is in the loosen state, Pins will always gain a HIGH signal until it is pressed.

Get One Now 

[<https://www.seeedstudio.com/Grove-Dual-Button-p-4529.html>]

Features

- Compact modular design and less wire connection with two buttons in one grove module
- Different color keycaps provided, freely adjust the color combination as needed.

Specification

Item	Value
Voltage range	3V-5V
Interface	Grove
Dimensions	20mm * 40mm
Battery	Exclude

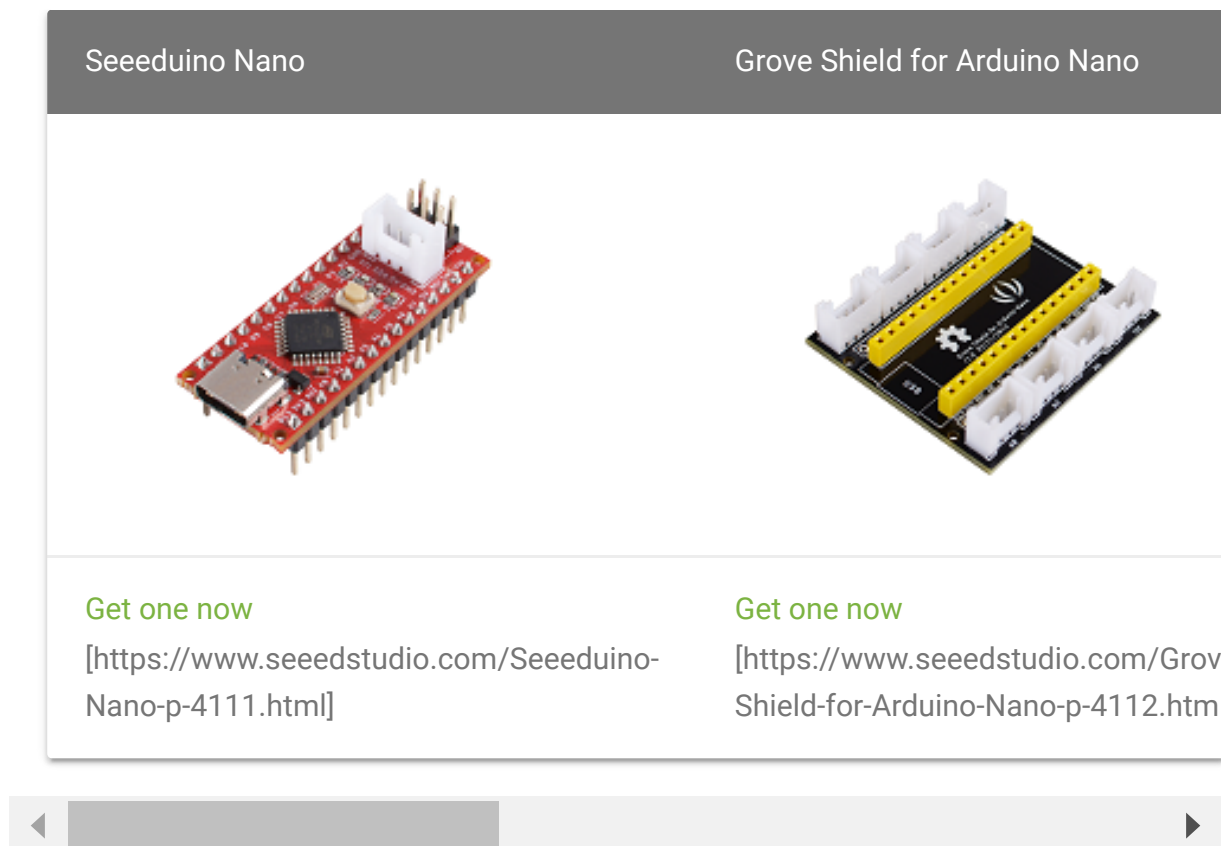
Platform Supported



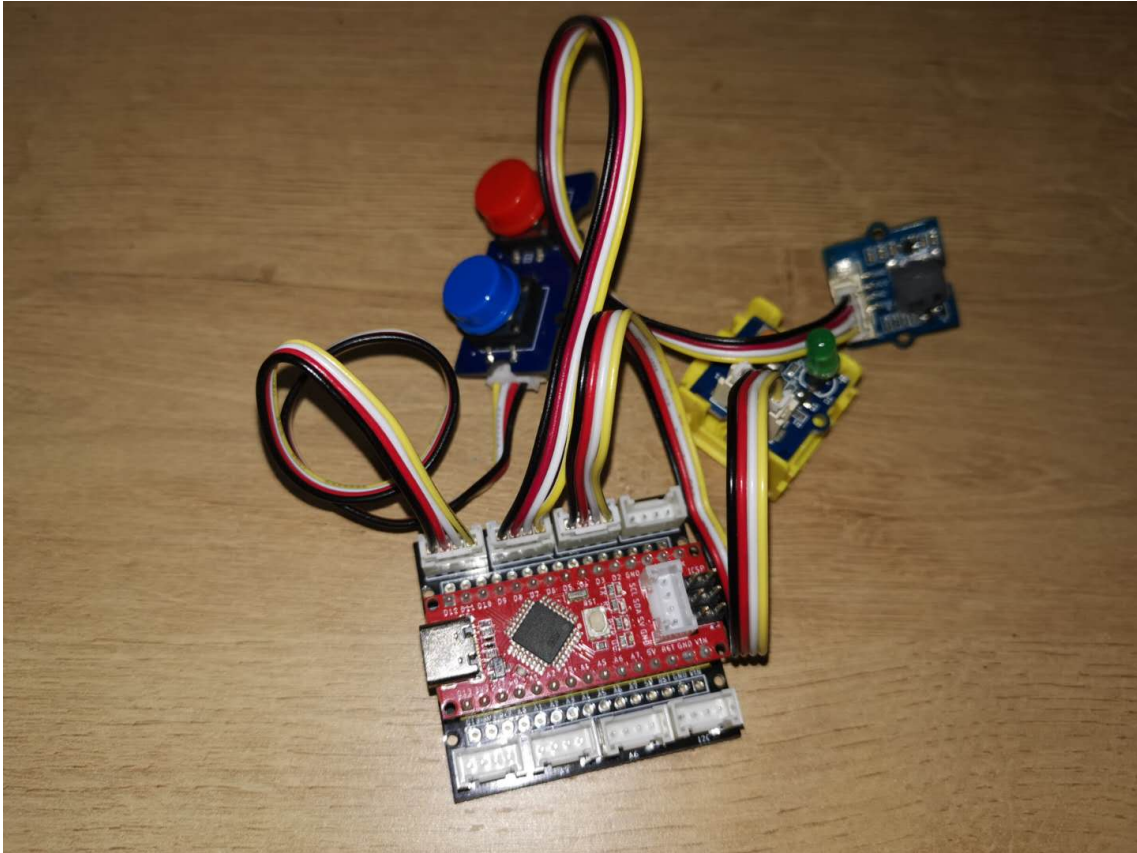
Getting Started

Getting Started with Arduino

Materials Required



Hardware Connection



The Grove Dual Button connects with "D2" interface on the Grove shield, Grove Passive Buzzer connects with "D4" interface and Grove LED Pack connects with "D6" interface.

Softwawre

- **Step1** Copy the code below to the Arduino IDE and upload. If you do not know how to update the code, please check [How to upload code](https://wiki.seeedstudio.com/Upload_Code/) [https://wiki.seeedstudio.com/Upload_Code/].

```
1 //set the corresponding notes with frequency
2 #define NOTE_D0 0
3 #define NOTE_D1 294
4 #define NOTE_D2 330
5 #define NOTE_D3 350
6 #define NOTE_D4 393
7 #define NOTE_D5 441
8 #define NOTE_D6 495
```



```
9  #define NOTE_D7 556
10
11  #define NOTE_DL1 147
12  #define NOTE_DL2 165
13  #define NOTE_DL3 175
14  #define NOTE_DL4 196
15  #define NOTE_DL5 221
16  #define NOTE_DL6 248
17  #define NOTE_DL7 278
18
19  #define NOTE_DH1 589
20  #define NOTE_DH2 661
21  #define NOTE_DH3 700
22  #define NOTE_DH4 786
23  #define NOTE_DH5 882
24  #define NOTE_DH6 990
25  #define NOTE_DH7 112
26
27  #define WHOLE 1
28  #define HALF 0.5
29  #define QUARTER 0.25
30  #define EIGHTH 0.25
31  #define SIXTEENTH 0.625
32
33  //the note part of the whole song
34  int tune[] =
35  {
36      NOTE_DH1, NOTE_D6, NOTE_D5, NOTE_D6, NOTE_D0,
37      NOTE_DH1, NOTE_D6, NOTE_D5, NOTE_DH1, NOTE_D6, N
38      NOTE_D6, NOTE_D6, NOTE_D5, NOTE_D6, NOTE_D0, NOT
39      NOTE_DH1, NOTE_D6, NOTE_D5, NOTE_DH1, NOTE_D6, N
40
41      NOTE_D1, NOTE_D1, NOTE_D3,
42      NOTE_D1, NOTE_D1, NOTE_D3, NOTE_D0,
43      NOTE_D6, NOTE_D6, NOTE_D6, NOTE_D5, NOTE_D6,
44      NOTE_D5, NOTE_D1, NOTE_D3, NOTE_D0,
45      NOTE_DH1, NOTE_D6, NOTE_D6, NOTE_D5, NOTE_D6,
46      NOTE_D5, NOTE_D1, NOTE_D2, NOTE_D0,
47      NOTE_D7, NOTE_D7, NOTE_D5, NOTE_D3,
48      NOTE_D5,
49      NOTE_DH1, NOTE_D0, NOTE_D6, NOTE_D6, NOTE_D5, NO
```

```

50     NOTE_D0, NOTE_D5, NOTE_D1, NOTE_D3, NOTE_D0,
51     NOTE_DH1, NOTE_D0, NOTE_D6, NOTE_D6, NOTE_D5, NO
52     NOTE_D0, NOTE_D5, NOTE_D1, NOTE_D2, NOTE_D0,
53     NOTE_D3, NOTE_D3, NOTE_D1, NOTE_DL6,
54     NOTE_D1,
55     NOTE_D3, NOTE_D5, NOTE_D6, NOTE_D6,
56     NOTE_D3, NOTE_D5, NOTE_D6, NOTE_D6,
57     NOTE_DH1, NOTE_D0, NOTE_D7, NOTE_D5,
58     NOTE_D6,
59 };
60
61 //the duration time of each note
62 float duration[] =
63 {
64     1, 1, 0.5, 0.5, 1,
65     0.5, 0.5, 0.5, 0.5, 1, 0.5, 0.5,
66     0.5, 1, 0.5, 1, 0.5, 0.5,
67     0.5, 0.5, 0.5, 0.5, 1, 1,
68
69     1, 1, 1 + 1,
70     0.5, 1, 1 + 0.5, 1,
71     1, 1, 0.5, 0.5, 1,
72     0.5, 1, 1 + 0.5, 1,
73     0.5, 0.5, 0.5, 0.5, 1 + 1,
74     0.5, 1, 1 + 0.5, 1,
75     1 + 1, 0.5, 0.5, 1,
76     1 + 1 + 1 + 1,
77     0.5, 0.5, 0.5 + 0.25, 0.25, 0.5 + 0.25, 0.25, 0.
78     0.5, 1, 0.5, 1, 1,
79     0.5, 0.5, 0.5 + 0.25, 0.25, 0.5 + 0.25, 0.25, 0.
80     0.5, 1, 0.5, 1, 1,
81     1 + 1, 0.5, 0.5, 1,
82     1 + 1 + 1 + 1,
83     0.5, 1, 0.5, 1 + 1,
84     0.5, 1, 0.5, 1 + 1,
85     1 + 1, 0.5, 0.5, 1,
86     1 + 1 + 1 + 1
87 };
88
89 int length;//define the number of notes
90 int tonePin = 4; //set the buzzer Pin

```

```
91  int button1 = 2; //set the button1 pin
92  int button2 = 3; //set the button2 pin
93  int LED = 6;  //set the LED pin
94  bool state1 = 1; //set button1 state
95  bool state2 = 1; //set button2 state
96  void setup()
97  {
98      pinMode(tonePin, OUTPUT); // set the buzzer as o
99      pinMode(button1,INPUT);
100     pinMode(button2,INPUT);
101     pinMode(LED,OUTPUT);
102     length = sizeof(tune) / sizeof(tune[0]); //count
103 }
104
105 void loop()
106 {
107     state1 = digitalRead(button1);
108     state2 = digitalRead(button2);
109     if (state1 == 0)
110     {
111         digitalWrite(LED,HIGH);
112     }
113     else
114     {
115         digitalWrite(LED,LOW);
116     }
117     if (state2 == 0)
118     {
119         for (int x = 0; x < length; x++) //"sing" th
120         {
121             tone(tonePin, tune[x]); //output the "x" note
122             delay(400 * duration[x]); //rythem of the musi
123             noTone(tonePin); //stop the current note and go
124         }
125     }
126     else
127     {
128         digitalWrite(tonePin,LOW);
129     }
130
131 }
```

- **Step2** Press each button to control the LED and Buzzer.

Getting Started with Raspberry Pi

Materials Required

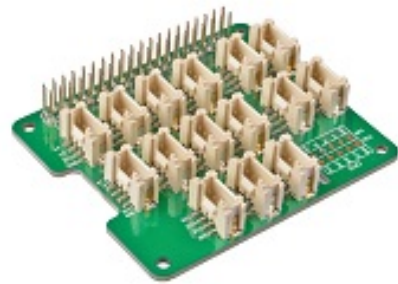
Raspberry Pi 4B(4GB)



[Get one now](#)

[<https://www.seeedstudio.com/Raspberry-Pi-4-Computer-Model-B-4GB-p-4077.html>]

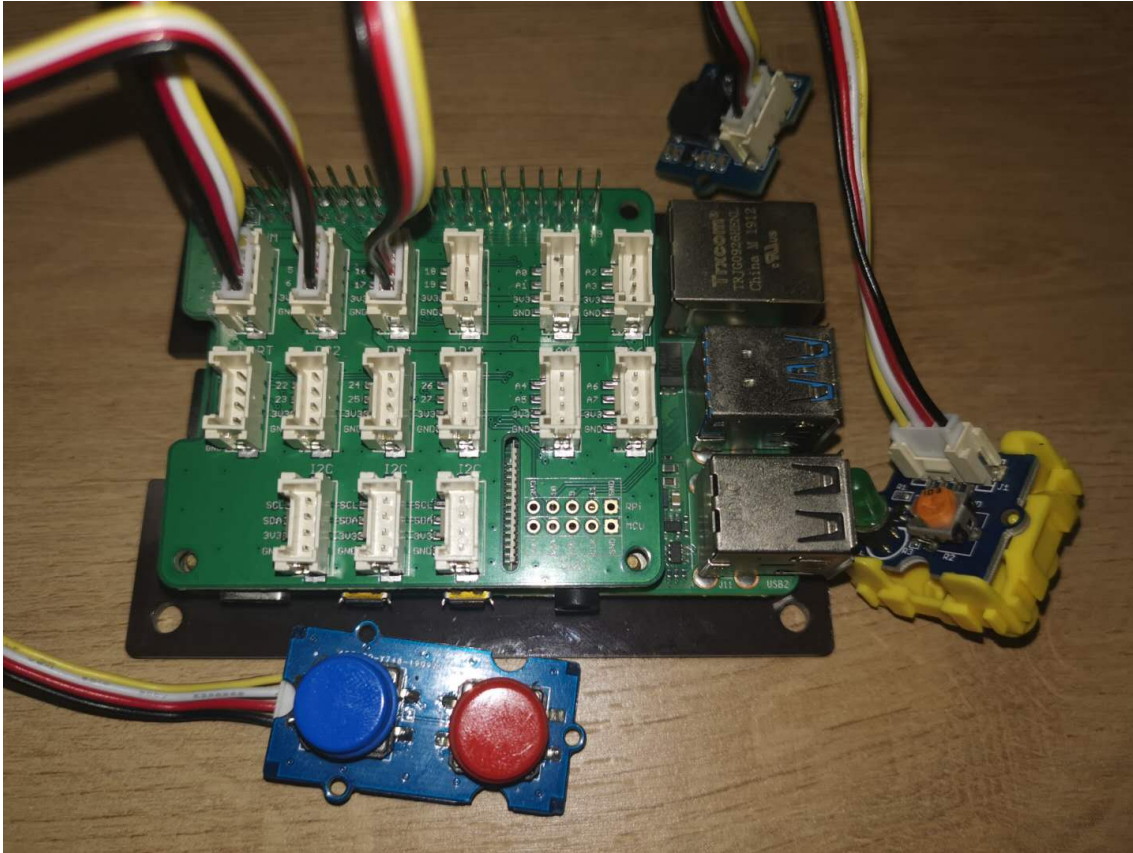
Grove Base Hat for Raspberry Pi



[Get one now](#)

[<https://www.seeedstudio.com/Grove-Base-Hat-for-Raspberry-Pi.html>]

Hardware Connection



Connect the passive with PWM pin "12", button "D5", LED "D16".

Code

- **Step 1** Install Grove.py on your Raspberry.

One-click installation, quick start, what ever you call, with the single command below, we can install/update all dependencies and latest grove.py.



Attention

If you are using **Raspberry Pi with Raspberrypi OS >= Bullseye**, you **cannot use this command line**.

```
curl -sL https://github.com/Seeed-Studio/grove.py/raw/master
```



**Success**

if everything goes well, you will see the following notice.

```
1 Successfully installed grove.py-0.5
2 #####
3 Lastest Grove.py from github install complete !!!!!
4 #####
```



Besides the one-click installation, you can also install all the dependencies and latest grove.py step by step.

**Attention**

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

```
1 git clone https://github.com/Seeed-Studio/grove.py
2 cd grove.py
3 # Python2
4 sudo pip install .
5 # Python3
6 sudo pip3 install .
```



- **Step 2** Create a python file for the code.

```
1 cd grove.py
2 nano button.py
```



- **Step 3** Copy the following code to the python file

```
1 #!/usr/bin/env python
2 import time
```



```
3 from grove.factory import Factory
4 from grove.grove_led import GroveLed
5
6 led = GroveLed(16)
7 button1 = Factory.getButton("GPIO-HIGH", 5)
8 button2 = Factory.getButton("GPIO-HIGH", 6)
9 buzzer = Factory.getGpioWrapper("Buzzer", 12)
10
11 while True:
12     if button1.is_pressed():
13         led.on()
14     else:
15         led.off()
16     if button2.is_pressed():
17         buzzer.on()
18     else:
19         buzzer.off()
```

- **Step 4** Run the program

```
1 sudo chmod +x button.py
2 sudo ./button.py
```



If everything goes well, you can control the led and buzzer by dual button.

Schematic Online Viewer



Resource

- **[PDF]** [Hardware schematic](https://files.seeedstudio.com/products/111020103/document/Grove-Dual-Button_v1_0_SCH_190916.pdf)
[https://files.seeedstudio.com/products/111020103/document/Grove-Dual-Button_v1_0_SCH_190916.pdf]

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]