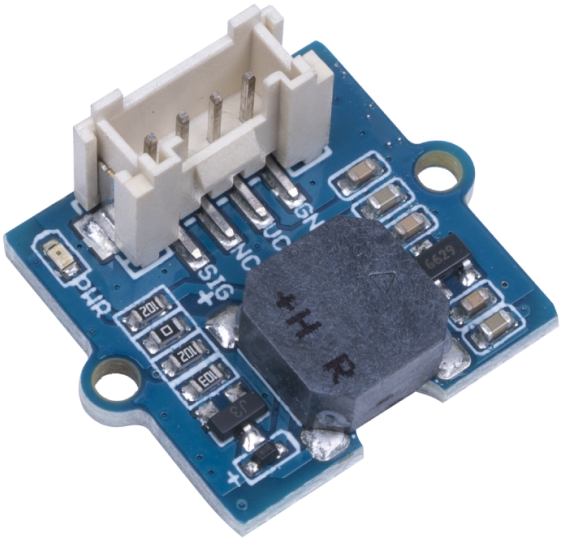
Grove - Passive Buzzer



This is a 3-5V passive buzzer. You can change the PMW frequency to award different beep sound to get a "buzzer music". Also, the buzzer can be set as an alarm for security applications. So get one and start your own project!

[https://www.seeedstudio.com/Grove-Passive-Buzzer-p-4525.html]

Features

- Passive: Tunable passive buzzer
- Interface: Grove

Specification

Item	Value
Voltage range	3V–5V
Resonant Frequency	2700 Hz
sound output	> 80dB
Working temperature	-20-70 °C
Dimensions	20mm * 20mm * 10mm
Weight	3g
Battery	Exclude

Platform Supported

Arduino	Raspberry Pi		

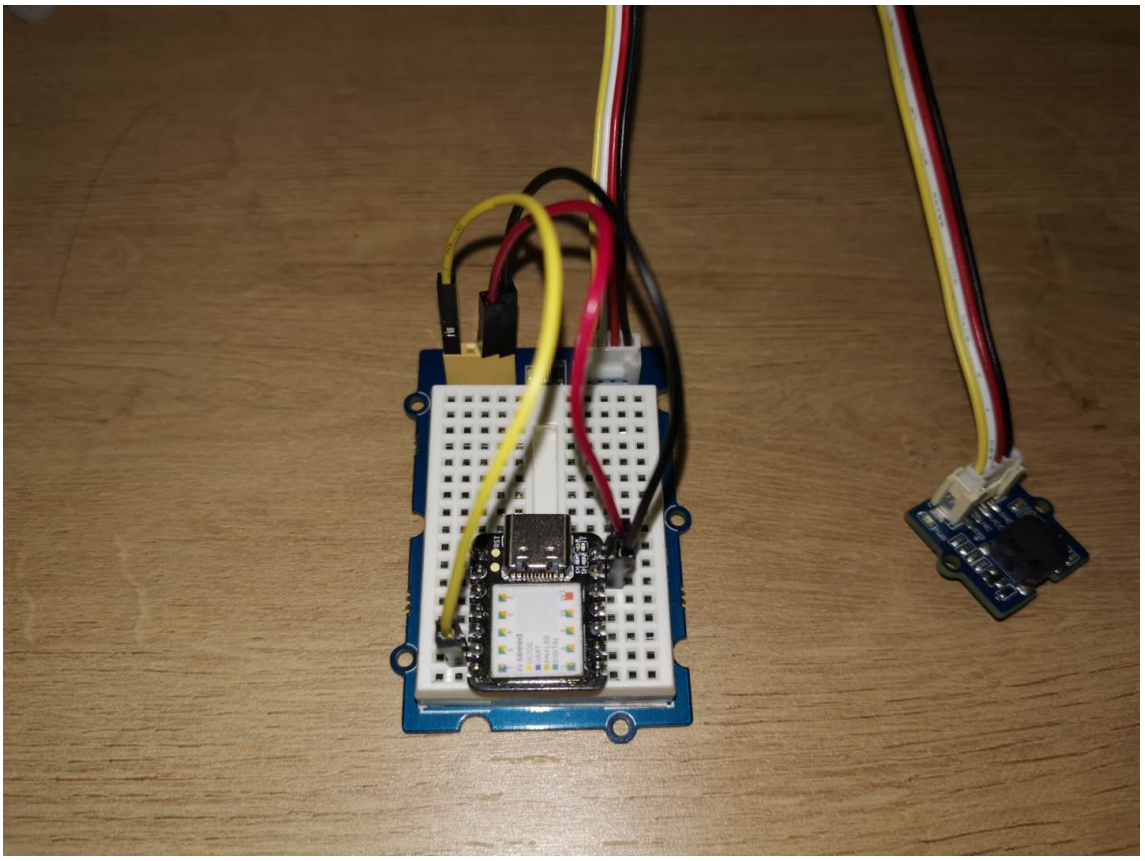
Getting Started

Play with Arduino

Materials Required

Seeeduino XIAO	Grove Breadboard
	
Get one now [https://www.seeedstudio.com/Seeeduino-XIAO-Arduino-Microcontroller-SAMD21-Cortex-M0+-p-4426.html]	Get one now [https://www.seeedstudio.com/Grov Breadboard-p-4034.html]

Hardware Connection



The Grove interface on the breadboard and on the Grove Passive Buzzer are connected by the Grove cable.

Software

- **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/].

Code example1 - simply get the beep sound

```
int buzzer = 5; // Buzzer connect with Pin 5
int frequency = 2700; //reach the Resonant Frequency
int cycle = 1000000/frequency;
void setup()
{
```

```
Serial.begin(9600); // set the baud rate
pinMode(buzzer,OUTPUT); // set buzzer as output
}

void loop()
{
  digitalWrite(buzzer,HIGH);
  delayMicroseconds(cycle/2);
  digitalWrite(buzzer,LOW);
  delayMicroseconds(cycle/2);  // run the PMW cycle
}
```

- **Step2** After uploading the code tp the board, you will hear thr buzzer beep.

Code example2 - Use buzzer to play music

```
//set the corresponding notes with frequency
#define NOTE_D0 0
#define NOTE_D1 294
#define NOTE_D2 330
#define NOTE_D3 350
#define NOTE_D4 393
#define NOTE_D5 441
#define NOTE_D6 495
#define NOTE_D7 556

#define NOTE_DL1 147
#define NOTE_DL2 165
#define NOTE_DL3 175
#define NOTE_DL4 196
#define NOTE_DL5 221
#define NOTE_DL6 248
#define NOTE_DL7 278

#define NOTE_DH1 589
#define NOTE_DH2 661
#define NOTE_DH3 700
#define NOTE_DH4 786
```

```
#define NOTE_DH5 882
#define NOTE_DH6 990
#define NOTE_DH7 112

#define WHOLE 1
#define HALF 0.5
#define QUARTER 0.25
#define EIGHTH 0.25
#define SIXTEENTH 0.625

//the note part of the whole song
int tune[] =
{
  NOTE_DH1, NOTE_D6, NOTE_D5, NOTE_D6, NOTE_D0,
  NOTE_DH1, NOTE_D6, NOTE_D5, NOTE_DH1, NOTE_D6, NOTE_
  NOTE_D6, NOTE_D6, NOTE_D5, NOTE_D6, NOTE_D0, NOTE_D6
  NOTE_DH1, NOTE_D6, NOTE_D5, NOTE_DH1, NOTE_D6, NOTE_

  NOTE_D1, NOTE_D1, NOTE_D3,
  NOTE_D1, NOTE_D1, NOTE_D3, NOTE_D0,
  NOTE_D6, NOTE_D6, NOTE_D6, NOTE_D5, NOTE_D6,
  NOTE_D5, NOTE_D1, NOTE_D3, NOTE_D0,
  NOTE_DH1, NOTE_D6, NOTE_D6, NOTE_D5, NOTE_D6,
  NOTE_D5, NOTE_D1, NOTE_D2, NOTE_D0,
  NOTE_D7, NOTE_D7, NOTE_D5, NOTE_D3,
  NOTE_D5,
  NOTE_DH1, NOTE_D0, NOTE_D6, NOTE_D6, NOTE_D5, NOTE_D
  NOTE_D0, NOTE_D5, NOTE_D1, NOTE_D3, NOTE_D0,
  NOTE_DH1, NOTE_D0, NOTE_D6, NOTE_D6, NOTE_D5, NOTE_D
  NOTE_D0, NOTE_D5, NOTE_D1, NOTE_D2, NOTE_D0,
  NOTE_D3, NOTE_D3, NOTE_D1, NOTE_DL6,
  NOTE_D1,
  NOTE_D3, NOTE_D5, NOTE_D6, NOTE_D6,
  NOTE_D3, NOTE_D5, NOTE_D6, NOTE_D6,
  NOTE_DH1, NOTE_D0, NOTE_D7, NOTE_D5,
  NOTE_D6,
};

//the duration time of each note
float duration[] =
{
```

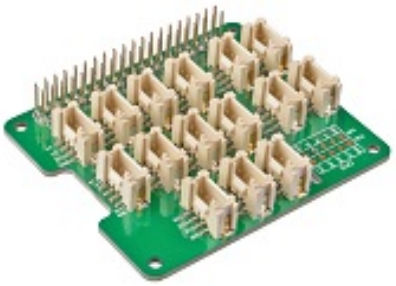
```
    1, 1, 0.5, 0.5, 1,
    0.5, 0.5, 0.5, 0.5, 1, 0.5, 0.5,
    0.5, 1, 0.5, 1, 0.5, 0.5,
    0.5, 0.5, 0.5, 0.5, 1, 1,

    1, 1, 1 + 1,
    0.5, 1, 1 + 0.5, 1,
    1, 1, 0.5, 0.5, 1,
    0.5, 1, 1 + 0.5, 1,
    0.5, 0.5, 0.5, 0.5, 1 + 1,
    0.5, 1, 1 + 0.5, 1,
    1 + 1, 0.5, 0.5, 1,
    1 + 1 + 1 + 1,
    0.5, 0.5, 0.5 + 0.25, 0.25, 0.5 + 0.25, 0.25, 0.5 +
    0.5, 1, 0.5, 1, 1,
    0.5, 0.5, 0.5 + 0.25, 0.25, 0.5 + 0.25, 0.25, 0.5 +
    0.5, 1, 0.5, 1, 1,
    1 + 1, 0.5, 0.5, 1,
    1 + 1 + 1 + 1,
    0.5, 1, 0.5, 1 + 1,
    0.5, 1, 0.5, 1 + 1,
    1 + 1, 0.5, 0.5, 1,
    1 + 1 + 1 + 1
};

int length;//define the number of notes
int buzzer = 5; //set the buzzer Pin
void setup()
{
  pinMode(buzzer, OUTPUT); // set the buzzer as output
  length = sizeof(tune) / sizeof(tune[0]); //count the
}

void loop()
{
   for (int x = 0; x < length; x++) //"sing" the note
   {
    tone(buzzer, tune[x]); //output the "x" note
    delay(400 * duration[x]); //rythem of the music,it
    noTone(buzzer);//stop the current note and go to t
   }
```

```
105       delay(5000);//after playing the whole song, delay f
106   }
```

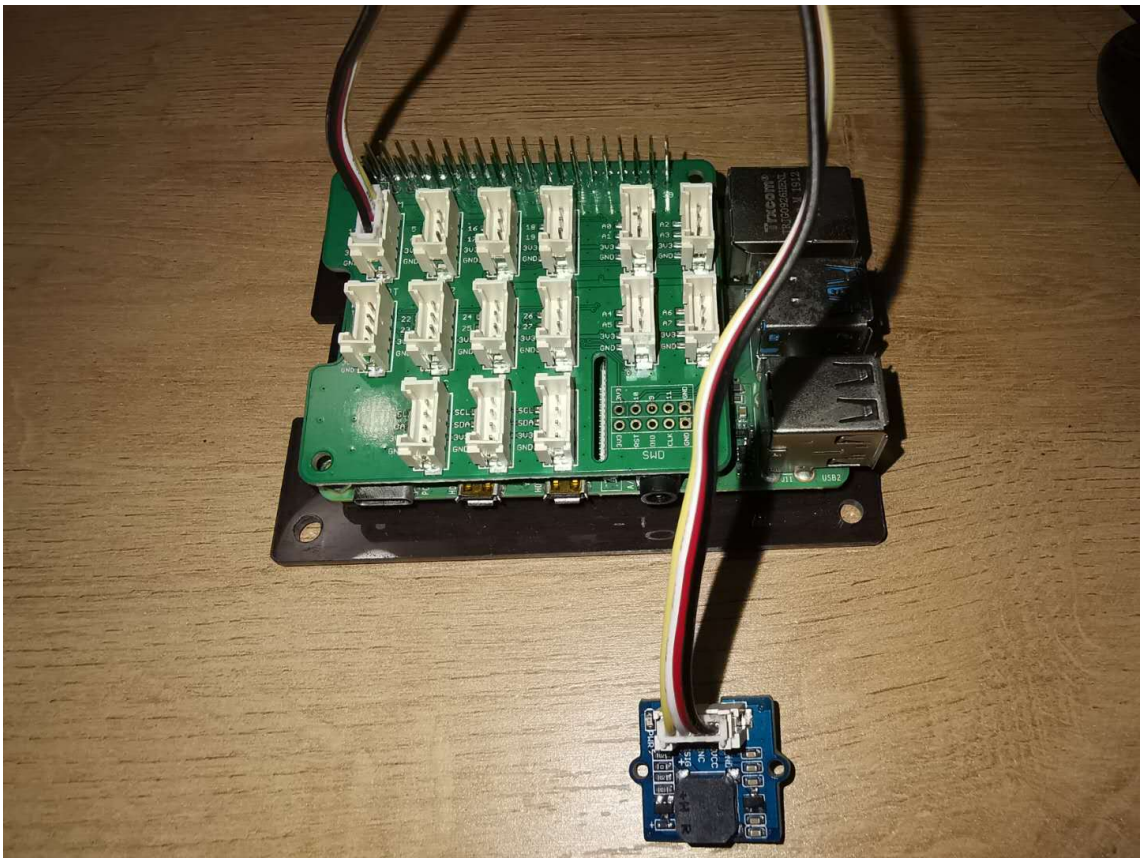
- **Step3** After uploading the code to the board, you can hear a music from the buzzer.

Play with Raspberry Pi

Materials Requied

Raspberry Pi 4B(4GB)	Grove Base Hat for Raspberry Pi
	
Get one now [https://www.seeedstudio.com/Raspberry-Pi-4-Computer-Model-B-4GB-p-4077.html]	Get one now [https://www.seeedstudio.com/Grove Base-Hat-for-Raspberry-Pi.html]

Hardware Connection



Connect the Buzzer with "PWM" on the Grove Base Hat.

Software 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.

``` Successfully installed grove.py-0.5

Lastest Grove.py from github install complete !!!!!

``` -

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**.

```
git clone https://github.com/Seeed-Studio/grove.py
cd grove.py
# Python2
sudo pip install .
# Python3
sudo pip3 install .
```

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

```
cd grove.py
nano example.py
```

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

```
#!/usr/bin/env python
import time
from mraa import getGpioLookup
from upm import pyupm_buzzer as upmBuzzer

def main():
    # Grove - Buzzer connected to PWM port
    buzzer = upmBuzzer.Buzzer(getGpioLookup('GPIO12'))

    CHORDS = [upmBuzzer.BUZZER_DO,  upmBuzzer.BUZZER_RE,
        upmBuzzer.BUZZER_FA,  upmBuzzer.BUZZER_SOL,  upmE
        upmBuzzer.BUZZER_SI]
    for i in range(0,  len(CHORDS)):
        buzzer.playSound(CHORDS[i],  500000)
        time.sleep(0.1)

    del buzzer
    print('application exiting...')

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

- **Step 4** Run the program

```
sudo chmod +x example.py
sudo ./example.py
```

If everything goes well, you can hear"do,re,mi,fa,so.la.xi".

Schematic Online Viewer

Resource

- **[PDF]** MLT_8530_DATASHEET [https://files.seeedstudio.com/products/107020109/document/MLT_8530_datasheet.pdf]
- **[PDF]** Hardware schematic [https://files.seeedstudio.com/products/107020109/document/Grove_Passive_Buzzer_SCH_190925.pdf]

Tech Support

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

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