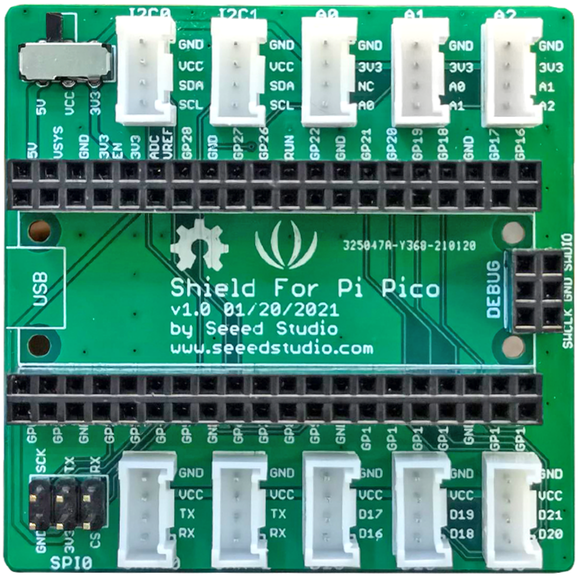
Grove Shield for Pi Pico V1.0



The Raspberry Pi Pico is a new popular low-cost, high-performance microcontroller board, so how to integrate Grove sensor to it? The grove shield perfectly solved this problem.

The Grove Shield for Pi Pico v1.0 is a plug-and-play shield for Raspberry Pi Pico which integrates with various kinds of Grove connectors, including 2*I2C, 3*Analog, 2*UART, 3*Digital ports, SWD debug interface and SPI pin, 3.3v/5v selectable power switch. It enables the build prototype and project in an easy and quick way without jumper wire and breadboard, you could explore infinite possibilities of Pico. The shield board is a stackable add-on board which acts as a bridge for Pi Pico and Seeed's Grove system.

Get One Now

[https://www.seeedstudio.com/Grove-Shield-for-Pi-Pico-v1-0-p-4846.html]

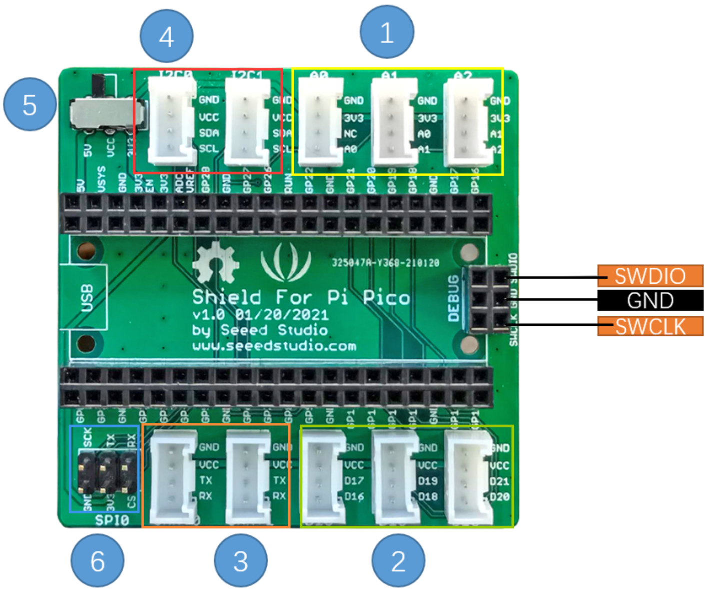
Specification

Parameter	Value/Range
Operating voltage	3.⅗V
Operation Temperature	-25℃ to +85℃
Analog Ports	3
Digital Ports	3
UART Ports	2
I2C Ports	2
Size	56mm x56mm

Compatible Boards

The Base Shield is tested and fully compatible with Pi Pico [https://www.seeedstudio.com/Raspberry-Pi-Pico-p-4832.html].

Hardware Overview



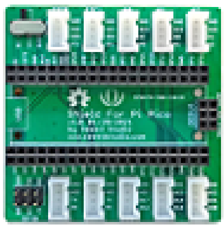
- **1-Analog Ports**: include 3 anlog ports, A0, A1, A2.
- **2-Digital Ports**: include 3 digital ports, D16, D18, D20.
- **3-UART Port**: 2 UART port.
- **4-I2C Ports**: 2 I2C ports.
- **5-Power Switch**: 5V/3.3V selectable power switch.
- **6-SPI port**: 1 spi0 port.
- **Dimension**: 56mm * 56mm

Getting Started

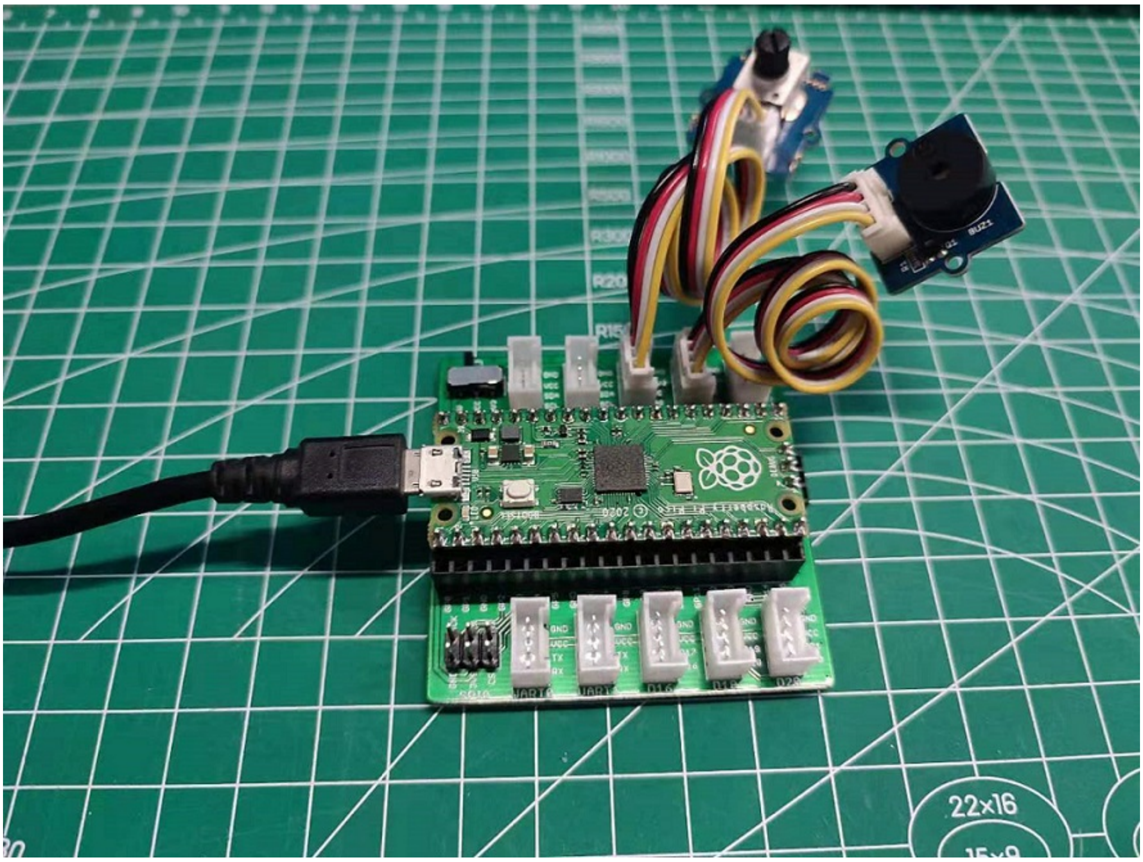
Project of Different Sounds of Buzzer

Materials Required

- Step 1. Prepare the below stuffs:

Pi Pico	Grove Shield for Pi Pico
	
Get ONE Now [https://www.seeedstudio.com/Raspberry-Pi-Pico-p-4832.html]	Get ONE Now [https://www.seeedstudio.com/Grov Shield-for-Pi-Pico-v1-0-p-4846.html]

- Step 2. Connect Grove Buzzer to port A1 of Base Shield.
- Step 3. Connect Grove Rotary Angle Sensor to port A0 of Base Shield.
- Step 4. Plug Grove Shield for Pi Pico into Pi Pico.
- Step 5. Connect Pi Pico to PC through a USB cable.



Software

- Step 1. Download Thonny(Windows [https://github.com/thonny/thonny/releases/download/v3.3.3/thonny-3.3.3.exe], Mac [https://github.com/thonny/thonny/releases/download/v3.3.3/thonny-3.3.3.pkg])Integrated Development Environment(IDE) according to your computer system.

For linux develop environment:

Binary bundle for PC (Thonny+Python):

```
bash <(wget -O - https://thonny.org/installer-for-linux)
```

With pip:

```
pip3 install thonny
```

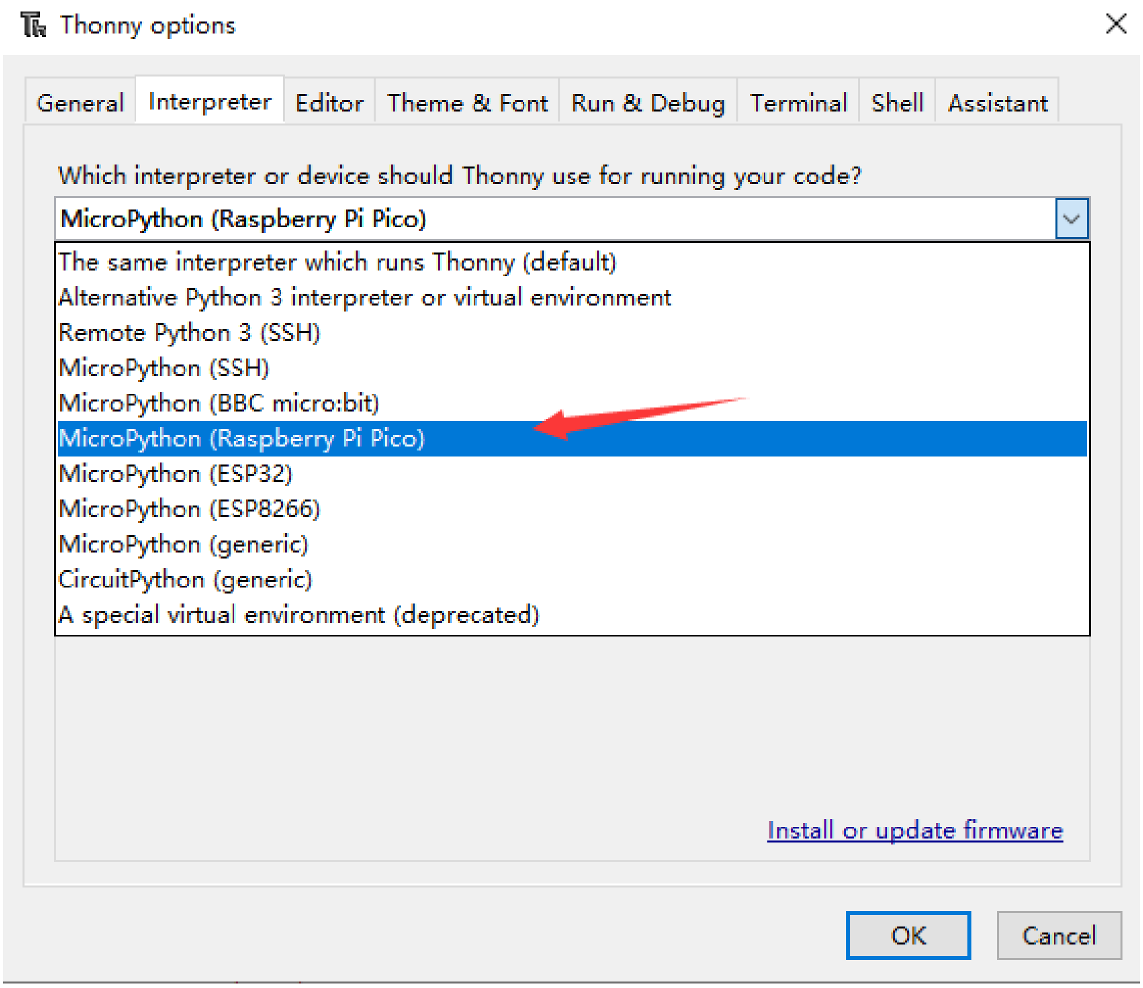
Debian, Rasbian, Ubuntu, Mint and others:

```
sudo apt install thonny
```

Fedora:

```
sudo dnf install thonny
```

- Step 2. Start Thonny and Connect your computer and the Raspberry Pi Pico together, then open up the **Run** menu and select **Run**->"**Select Interpreter**", picking "**MicoPython(Raspberry Pi Pico)**" from the drop down, and select the com port of your Pi Pico. If you can't find "Micopython (Raspberry Pico)" in Select Interpreter, please download the latest version of Thonny.

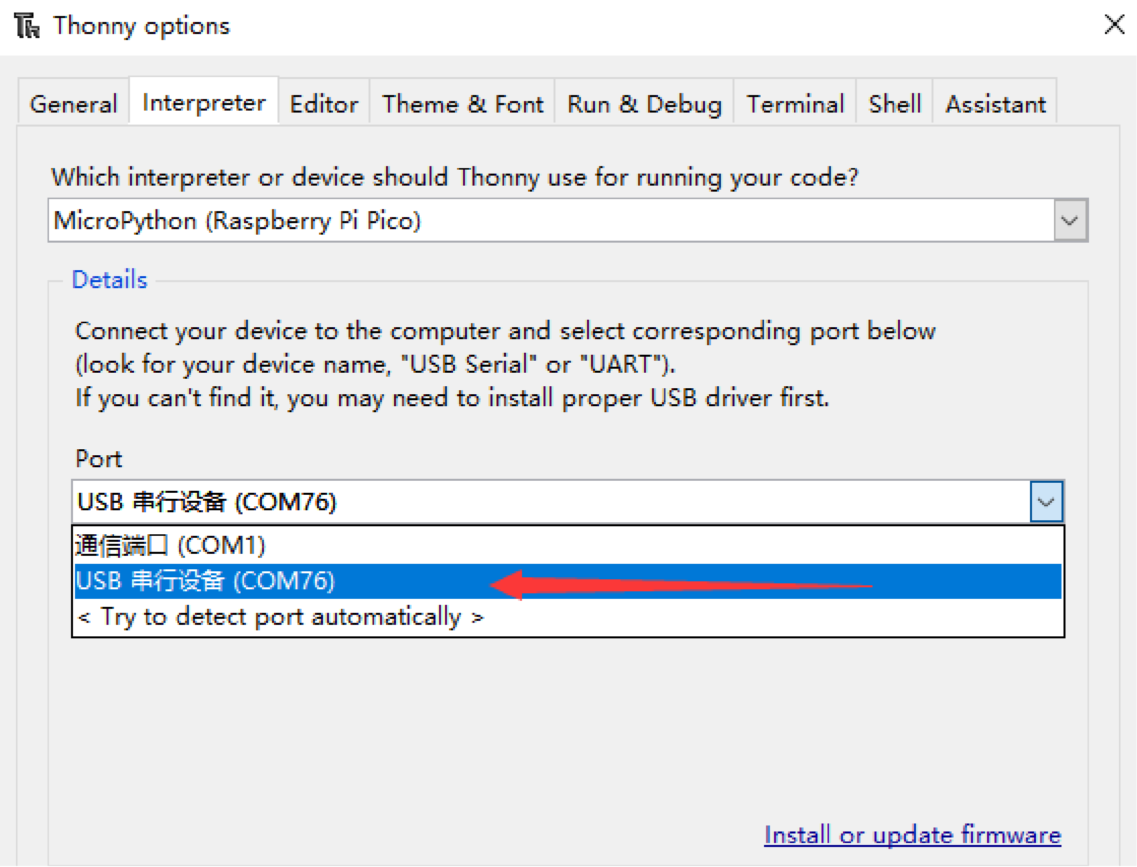


Thonny options

Which interpreter or device should Thonny use for running your code?

MicroPython (Raspberry Pi Pico)

General
Interpreter
Editor
Theme & Font
Run & Debug
Terminal
Shell
Assistant
The same interpreter which runs Thonny (default)
Alternative Python 3 interpreter or virtual environment
Remote Python 3 (SSH)
MicroPython (SSH)
MicroPython (BBC micro:bit)
MicroPython (Raspberry Pi Pico)
MicroPython (ESP32)
MicroPython (ESP8266)
MicroPython (generic)
CircuitPython (generic)
A special virtual environment (deprecated)
Install or update firmware
OK
Cancel



Thonny options

Which interpreter or device should Thonny use for running your code?

MicroPython (Raspberry Pi Pico)

Details

Connect your device to the computer and select corresponding port below
(look for your device name, "USB Serial" or "UART").
If you can't find it, you may need to install proper USB driver first.

Port

USB 串行设备 (COM76)

Hit "ok" if your Raspberry Pi Pico is plugged in and running MicroPython Thonny should automatically connect to the Repl.

- Step 3. Copy below code to the Thonny IDE, and click the green run button.

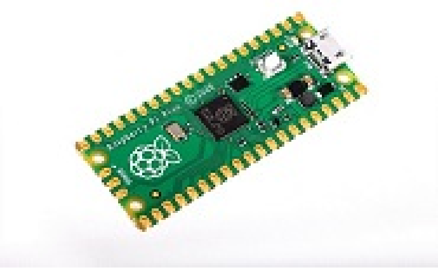
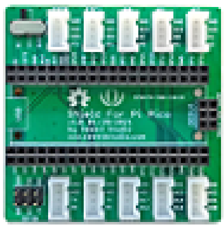
```
from machine import Pin,PWM,ADC
from time import sleep
adc = ADC(0) #ADC input (knob potentiometer) connect
pwm = PWM(Pin(27))#DAC output (buzzer) connected to
pwm.freq(10000)
while True:

    '''Analog port test'''
    val = adc.read_u16()#Read A2 port adc value (655
    #Drive the buzzer, turn off the buzzer when the
    if val > 300:
        pwm.freq(int(val/10))
        pwm.duty_u16(10000)
    else:
        pwm.duty_u16(0)
    print(val)
    sleep(0.05)
```

Rotate Grove-Rotary-Angle-Sensor, you will hear different sounds from the buzzer.

Project of Detecting Temperature and Humidity

Materials Required

- Step 1. Prepare the below stuffs:

Pi Pico	Grove Shield for Pi Pico
	
Get ONE Now [https://www.seeedstudio.com/Raspberry-Pi-Pico-p-4832.html]	Get ONE Now [https://www.seeedstudio.com/Grove-Shield-for-Pi-Pico-v1-0-p-4846.html]

Note
In this kit, we have upgraded the Grove Temperature Humidity Sensor(DHT11) to Grove Temperature Humidity Sensor(DHT20) [https://www.seeedstudio.com/Grove-Temperature-Humidity-Sensor-V2-0-DHT20-p-4967.html]. If you want to get started with Grove Temperature Humidity Sensor(DHT20), you can click here [https://wiki.seeedstudio.com/Grove-Temperature-Humidity-Sensor-DH20/].

- Step 2. Connect Grove OLED Display 0.96" to port I2C1 of Base Shield.
- Step 3. Connect Grove Temperature Humidity Sensor DHT11 to port D18 of Base Shield.
- Step 4. Plug Grove Shield for Pi Pico into Pi Pico.

- Step 5. Connect Pi Pico to PC through a USB cable.

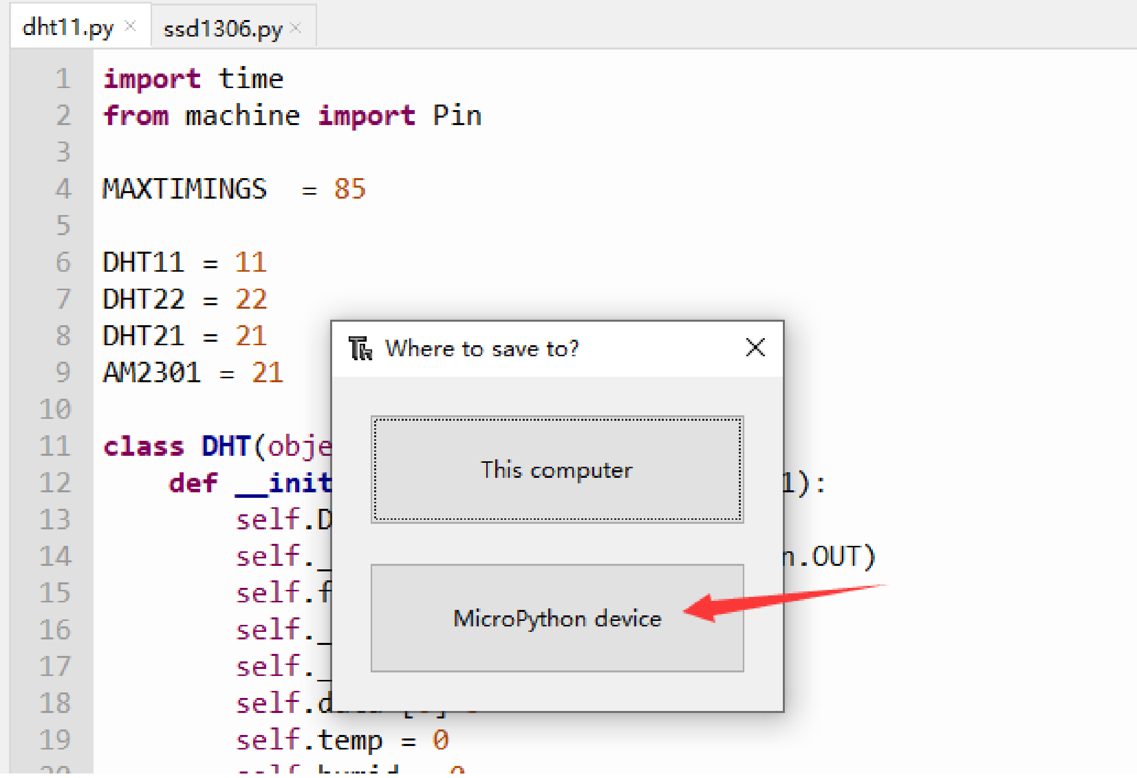
Software

Please refer to the demo1 software part.

Copy below code to the Thonny IDE at first.

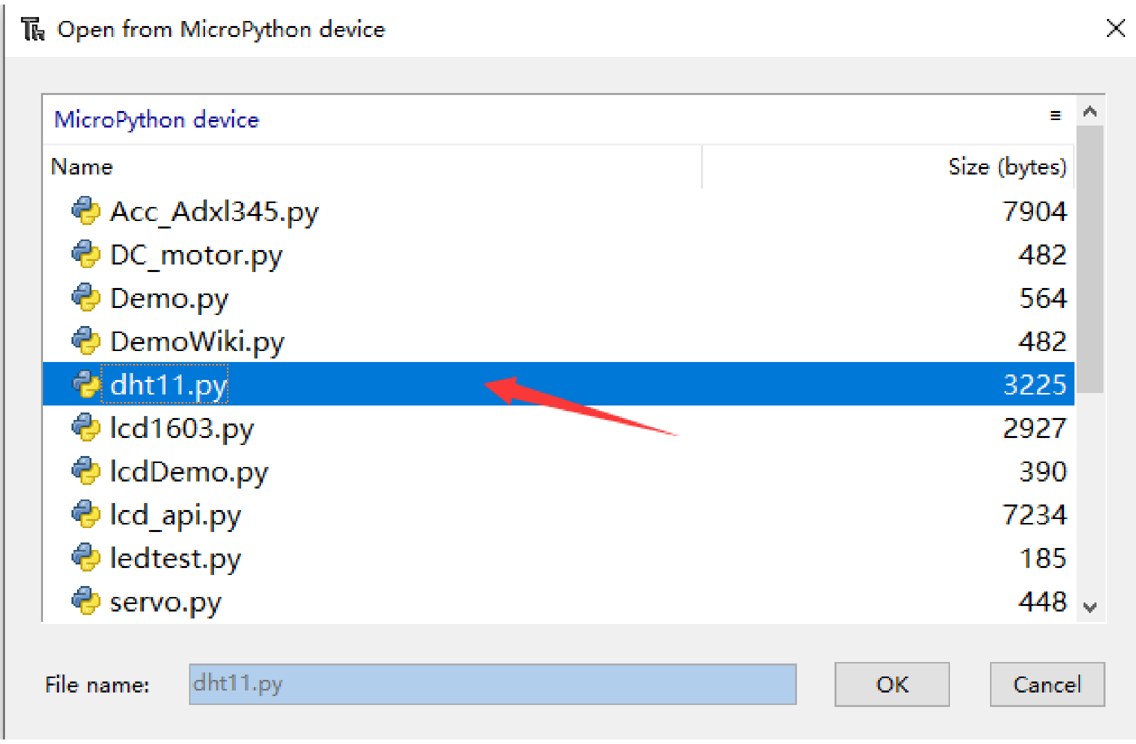
```
from ssd1306 import SSD1306_I2C
from dht11 import *
from machine import Pin, I2C
from time import sleep

i2c = I2C(1, scl=Pin(7), sda=Pin(6), freq=200000)#oled c
oled = SSD1306_I2C(128, 64, i2c)
dht2 = DHT(18) #temperature and humidity sensor connect


while True:

    temp,humid = dht2.readTempHumid()#temp:  humid:
    '''I2C port test'''
    ''' oled display test'''
    oled.fill(0)#Clear screen
    oled.text("Temp:  " + str(temp),0,0)#display tempear
    oled.text("Humid: " + str(humid),0,8)
    oled.show()
    sleep(0.5)
```

Then please download the ssd1306.py [https://github.com/micropython/micropython/blob/master/drivers/display/ssd1306.py] and dht11.py [http://47.106.166.129/Embeded/pico-micropython-grove/blob/master/Digital/dht11.py] to your local. Use Thonny open dht11.py, click file->save as->MicroPython device.



```
import time
from machine import Pin

MAXTIMINGS  = 85

DHT11 = 11
DHT22 = 22
DHT21 = 21
AM2301 = 21

class DHT(obje
    def __init                          1):
        self.D
        self._                          n.OUT)
        self.f
        self._
        self._
        self.d
        self.temp = 0
```

Input **dht11.py** in File name column, click "ok", then the file is saved at File->Open->MicroPython device, the same with the **ssd1306.py** file.

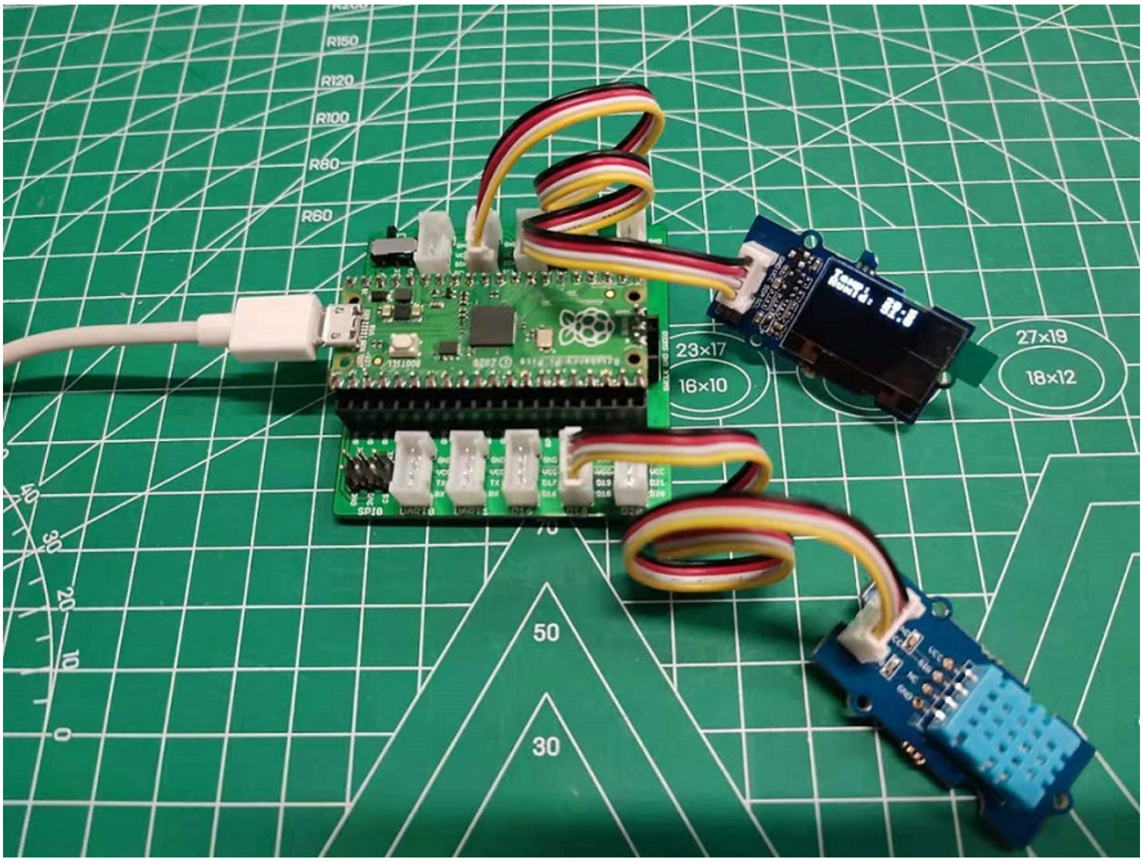


Name	Size (bytes)
Acc_Adxl345.py	7904
DC_motor.py	482
Demo.py	564
DemoWiki.py	482
dht11.py	3225
lcd1603.py	2927
lcdDemo.py	390
lcd_api.py	7234
ledtest.py	185
servo.py	448

File name: dht11.py

After you have save both of the files to your MicroPython device, now please click the green button to run the demo code.

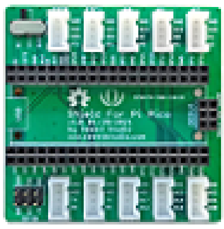
Then you can get the temperature and humidity displayed on the OLED as below.



Project of Controling Led and Relay

Materials Required

- Step 1. Prepare the below stuffs:

Pi Pico	Grove Shield for Pi Pico
	
Get ONE Now [https://www.seeedstudio.com/Raspberry-Pi-Pico-p-4832.html]	Get ONE Now [https://www.seeedstudio.com/Grove-Shield-for-Pi-Pico-v1-0-p-4846.html]

- Step 2. Connect Grove button to digital pin 18 of Base Shield.
- Step 3. Connect Grove led to port D16 of Base Shield.
- Step 4. Connect Grove Relay to port D20 of Base Shield.
- Step 5. Plug Grove Shield for Pi Pico into Pi Pico.
- Step 6. Connect Pi Pico to PC through a USB cable.

Software

Please refer to the demo1 software part.

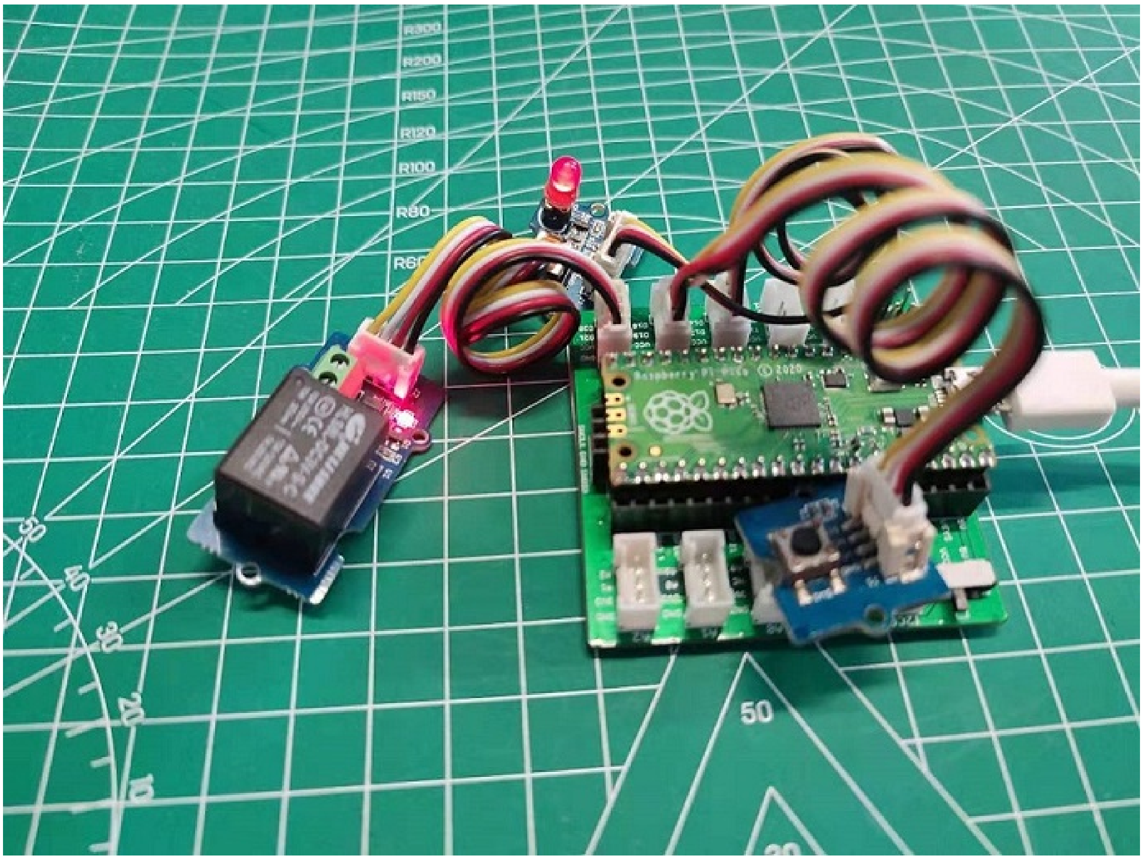
Copy below code to the Thonny IDE at first.

```
from machine import Pin

button = Pin(18, Pin.IN, Pin.PULL_UP)# button connect to
button.irq(lambda pin: InterruptsButton(),Pin.IRQ_FALLIN
led = Pin(16, Pin.OUT)#led connect to D16
```

```
relay = Pin(20, Pin.OUT)
tmp = 0
'''Key interrupt function, change the state of the light
def InterruptsButton(): #button input
    global tmp
    tmp = ~tmp
    led.value(tmp)
    relay.value(tmp)
while True:
    pass
```

Now please click the green button to run the demo code.

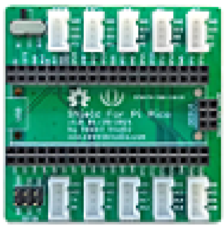
Then you can press the grove button, you can control the led and relay open and close.



Project of Flashing Colors

Materials Required

- Step 1. Prepare the below stuffs:

Pi Pico	Grove Shield for Pi Pico
	
Get ONE Now [https://www.seeedstudio.com/Raspberry-Pi-Pico-p-4832.html]	Get ONE Now [https://www.seeedstudio.com/Grove-Shield-for-Pi-Pico-v1-0-p-4846.html]

- Step 2. Connect RGB LED WS2813 mini to port 18 of Base Shield.
- Step 3. Plug Grove Shield for Pi Pico into Pi Pico.
- Step 4. Connect Pi Pico to PC through a USB cable.

Software

Please refer to the demo1 software part.

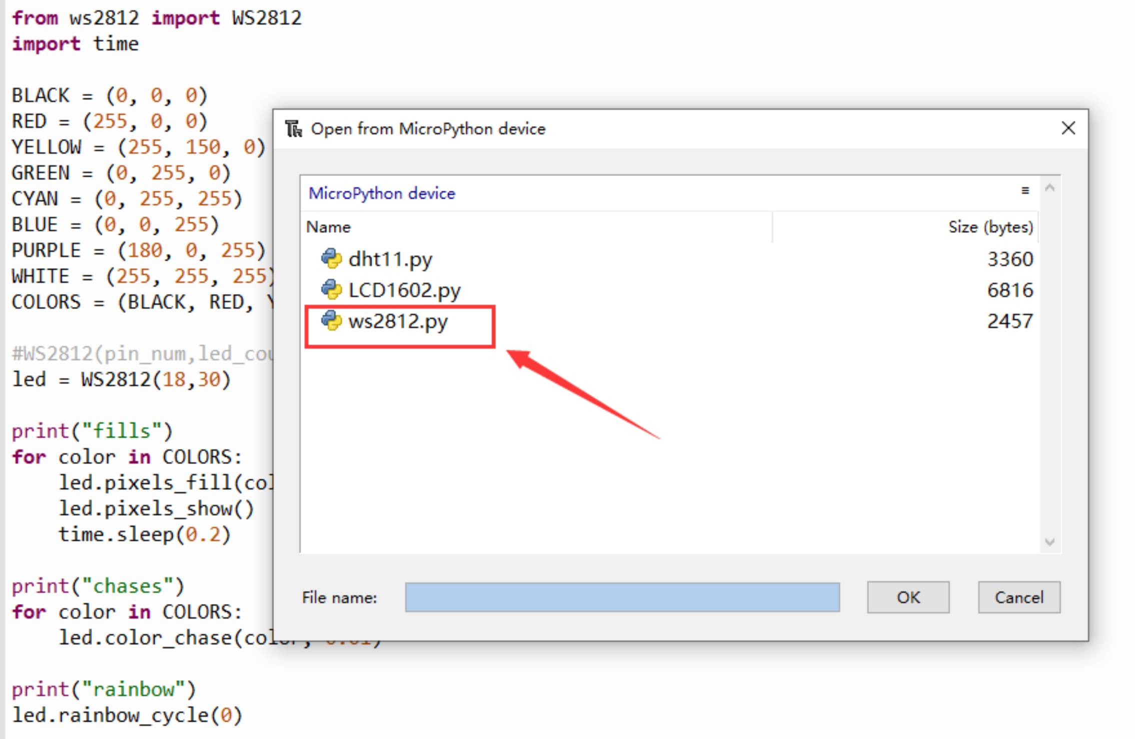
Copy below code to the Thonny IDE at first.

```
from ws2812 import WS2812
import time

```

```
BLACK = (0, 0, 0)
RED = (255, 0, 0)
YELLOW = (255, 150, 0)
GREEN = (0, 255, 0)
CYAN = (0, 255, 255)
BLUE = (0, 0, 255)
PURPLE = (180, 0, 255)
WHITE = (255, 255, 255)
COLORS = (BLACK, RED, YELLOW, GREEN, CYAN, BLUE, PURPLE,

#WS2812(pin_num,led_count)
led = WS2812(18,30)

print("fills")
for color in COLORS:
    led.pixels_fill(color)
    led.pixels_show()
    time.sleep(0.2)

print("chases")
for color in COLORS:
    led.color_chase(color, 0.01)

print("rainbow")
led.rainbow_cycle(0)
```

Then please download the ws2812.py [http://47.106.166.129/Embeded/pico-micropython-grove/blob/master/Digital/ws2812.py] to your local. Use Thonny open ws2812.py, click file->save as->MicroPython device.

Input **ws2812.py** in File name column, click "ok", then the file is saved at File->Open->MicroPython device.



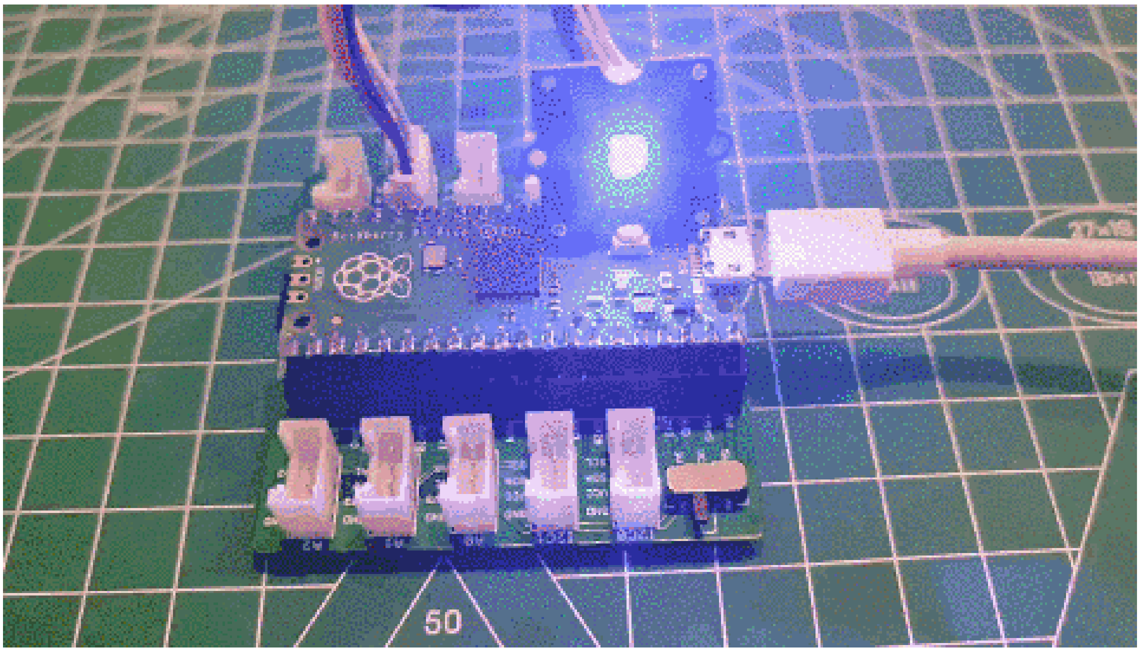
```
from ws2812 import WS2812
import time

BLACK = (0, 0, 0)
RED = (255, 0, 0)
YELLOW = (255, 150, 0)
GREEN = (0, 255, 0)
CYAN = (0, 255, 255)
BLUE = (0, 0, 255)
PURPLE = (180, 0, 255)
WHITE = (255, 255, 255
COLORS = (BLACK, RED, Y

#WS2812(pin_num,led_cou
led = WS2812(18,30)

print("fills")
for color in COLORS:
    led.pixels_fill(col
    led.pixels_show()
    time.sleep(0.2)

print("chases")
for color in COLORS:
    led.color_chase(co     0.01)

print("rainbow")
led.rainbow_cycle(0)
```

Open from MicroPython device

MicroPython device

Name	Size (bytes)
dht11.py	3360
LCD1602.py	6816
ws2812.py	2457

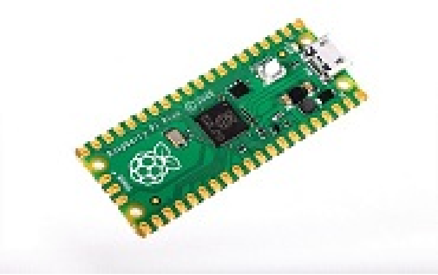
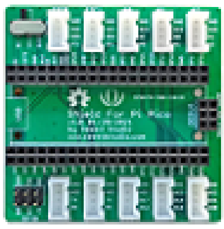
After you have save the files to your MicroPython device, now please click the green button to run the demo code. Then you can get the RGB LED WS2813 mini flash beautiful color as below.



Project of Detecting Sound and Light

Materials Required

- Step 1. Prepare the below stuffs:

Pi Pico	Grove Shield for Pi Pico
	
Get ONE Now [https://www.seeedstudio.com/Raspberry-Pi-Pico-p-4832.html]	Get ONE Now [https://www.seeedstudio.com/Grove Shield-for-Pi-Pico-v1-0-p-4846.html]

- Step 2. Connect Grove sound sensor to analog pin 0 of Base Shield.
- Step 3. Connect Grove light to port A1 of Base Shield.
- Step 4. Connect Grove 16X2 lcd to port I2C1 of Base Shield.
- Step 5. Plug Grove Shield for Pi Pico into Pi Pico.
- Step 6. Connect Pi Pico to PC through a USB cable.

Software

Please refer to the demo1 software part.

Copy below code to the Thonny IDE at first.

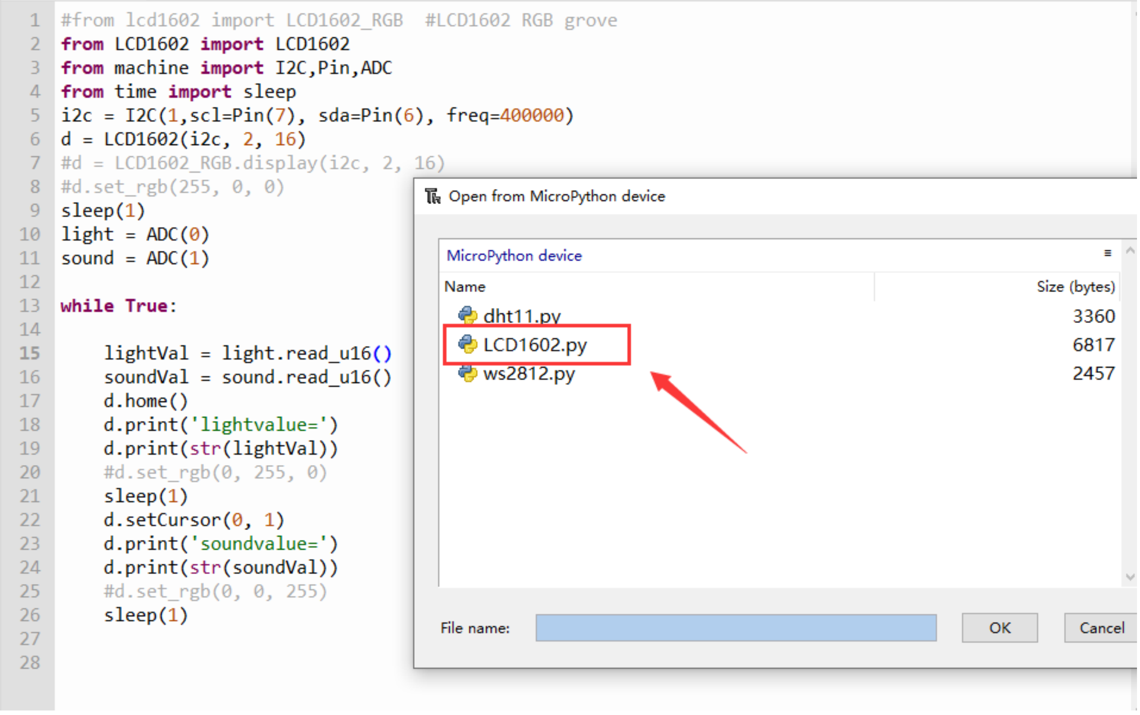
```python
#from lcd1602 import LCD1602_RGB  #LCD1602 RGB grove
from lcd1602 import LCD1602
from machine import I2C,Pin,ADC
from time import sleep
i2c = I2C(1,scl=Pin(7), sda=Pin(6), freq=400000)
d = LCD1602(i2c, 2, 16)
#d = LCD1602_RGB.display(i2c, 2, 16)
#d.set_rgb(255, 0, 0)
sleep(1)
light = ADC(0)
sound = ADC(1)

while True:

    lightVal = light.read_u16()
    soundVal = sound.read_u16()
    d.home()
    d.print('lightvalue=')
    d.print(str(lightVal))
    #d.set_rgb(0, 255, 0)
    sleep(1)
    d.setCursor(0, 1)
    d.print('soundvalue=')
    d.print(str(soundVal))
    #d.set_rgb(0, 0, 255)
    sleep(1)
```

Then please download the LCD1602.py [http://47.106.166.129/Embeded/pico-micropython-grove/blob/master/I2C/lcd1602.py] to your local. Use Thonny open LCD1602.py, click file->save as->MicroPython device.

Input **LCD1602.py** in File name column, click "ok", then the file is saved at File->Open->MicroPython device.

Note

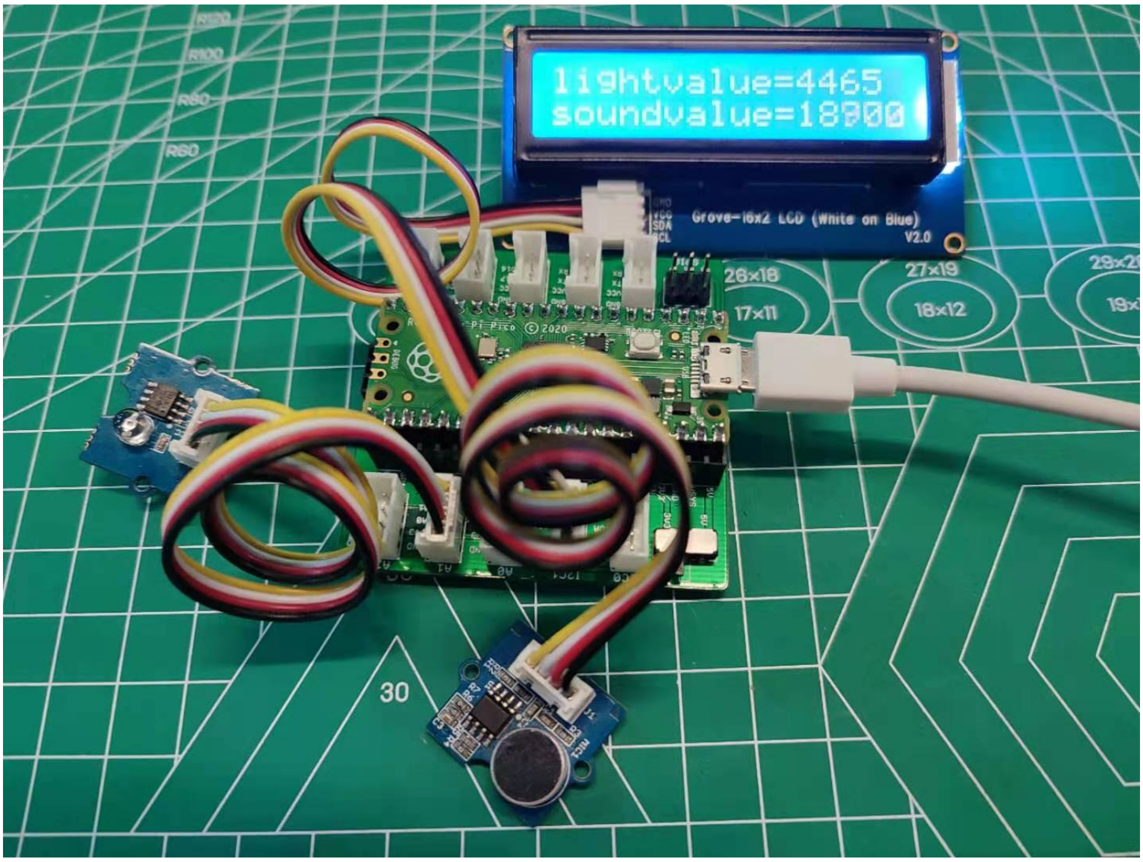
In this example, the version of LCD1602 we are using is a monochrome backlit version, If you need to control the full-color backlit version of LCD1602, check out the functions in this library file to learn how to use it.



```
#from lcd1602 import LCD1602_RGB  #LCD1602 RGB grove
from LCD1602 import LCD1602
from machine import I2C,Pin,ADC
from time import sleep
i2c = I2C(1,scl=Pin(7), sda=Pin(6), freq=400000)
d = LCD1602(i2c, 2, 16)
#d = LCD1602_RGB.display(i2c, 2, 16)
#d.set_rgb(255, 0, 0)
sleep(1)
light = ADC(0)
sound = ADC(1)

while True:

    lightVal = light.read_u16()
    soundVal = sound.read_u16()
    d.home()
    d.print('lightvalue=')
    d.print(str(lightVal))
    #d.set_rgb(0, 255, 0)
    sleep(1)
    d.setCursor(0, 1)
    d.print('soundvalue=')
    d.print(str(soundVal))
    #d.set_rgb(0, 0, 255)
    sleep(1)
```

Name	Size (bytes)
dht11.py	3360
LCD1602.py	6817
ws2812.py	2457

After you have save the files to your MicroPython device, now please click the green button to run the demo code. Then you can get the sound sensor and light sensor data as below.

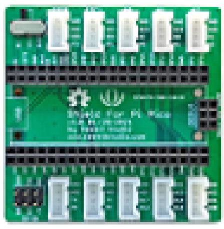


```
lightvalue=4465
soundvalue=18900
```

Project of Detecting Motion

Materials Required

- Step 1. Prepare the below stuffs:

Pi Pico	Grove Shield for Pi Pico
	
Get ONE Now [https://www.seeedstudio.com/Raspberry-Pi-Pico-p-4832.html]	Get ONE Now [https://www.seeedstudio.com/Grove-Shield-for-Pi-Pico-v1-0-p-4846.html]

- Step 2. Connect Grove servo to analog pin 1 of Base Shield.
- Step 3. Connect Grove Mini fan to port D16 of Base Shield.
- Step 4. Connect Grove Mini pir motion sensor to port D18 of Base Shield.
- Step 5. Plug Grove Shield for Pi Pico into Pi Pico.
- Step 6. Connect Pi Pico to PC through a USB cable.

Software

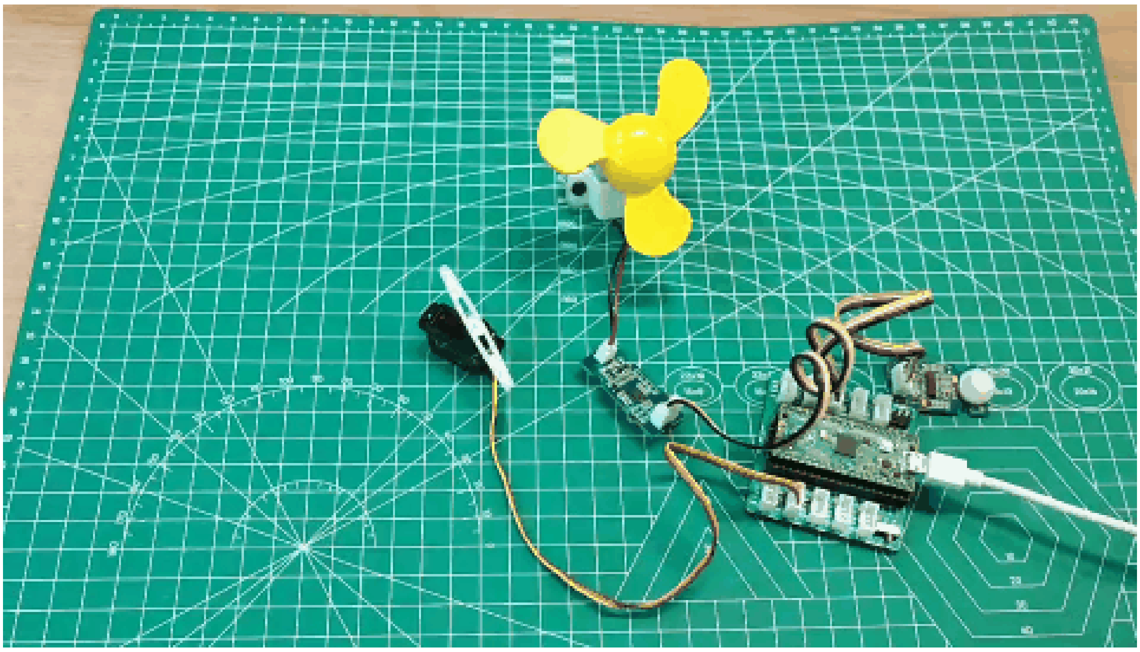
Please refer to the demo1 software part.

Copy below code to the Thonny IDE at first.

```
from machine import Pin,ADC,PWM
from time import sleep
import utime
```

```

miniFun = Pin(16, Pin.OUT)
miniPir = Pin(18, Pin.IN)

pwm_Servo=PWM(Pin(27))
pwm_Servo.freq(500)
Servo_Val =0

while True:

    if  miniPir.value() == 1 :
        miniFun.value(1)

        while  Servo_Val<65535:
            Servo_Val=Servo_Val+50
            utime.sleep_ms(1)
            pwm_Servo.duty_u16(Servo_Val)
        while Servo_Val>0:
            Servo_Val=Servo_Val-50
            utime.sleep_ms(1)
            pwm_Servo.duty_u16(Servo_Val)

    else :
        miniFun.value(0)

        pwm_Servo.duty_u16(0)
```

Now please click the green button to run the demo code. Then you can get the grove mini fan and grove servo run When you hands swiping past the pir sensor as below.



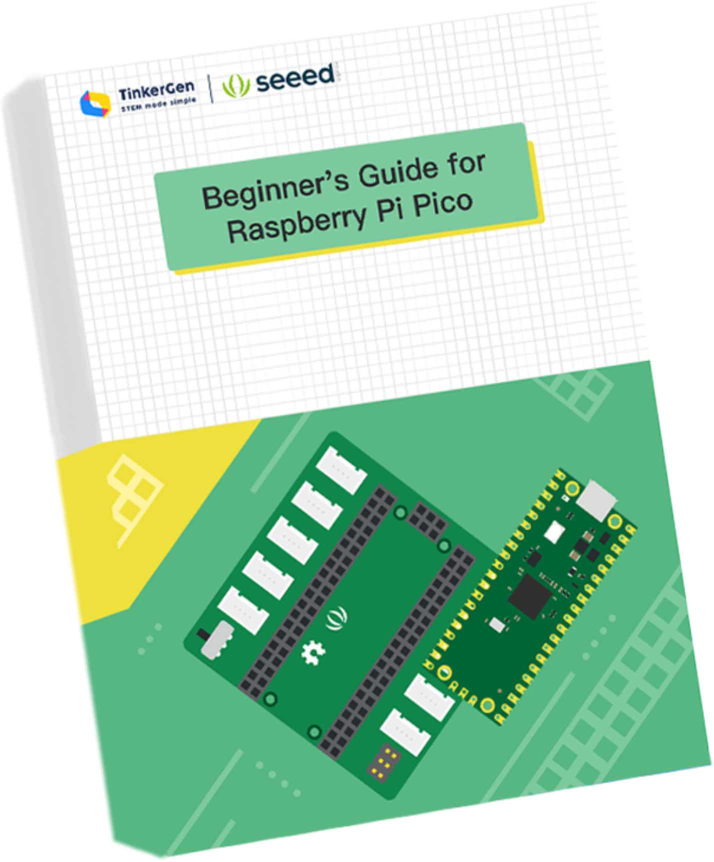
Schematic Online Viewer

Resources

- **[PDF]** Pico python SDK [https://files.seeedstudio.com/wiki/Grove_Shield_for_Pi_Pico_V1.0/pico_python_sdk.pdf]
- **[PDF]** SCH [https://files.seeedstudio.com/wiki/Grove_Shield_for_Pi_Pico_V1.0/Grove_shield_for_PI_PICOv1.0SCH.pdf]

- **[Eagle]** PCB&SCH [https:///files.seeedstudio.com/wiki/Grove_Shield_for_Pi_Pico_V1.0/res/Grove_Shield_for_Pi_Pico.zip]

Course Resources



- **[ZIP]** Beginners Guide of Raspberry Pi Pico Based on MicroPython [https://files.seeedstudio.com/Seeed_EDU/Course_documents/Beginner's-Guide-for-Raspberry-Pi-Pico.zip]
- **[ZIP]** Codes [https://files.seeedstudio.com/wiki/Grove_Shield_for_Pi_Pico_V1.0/Codes.rar]

- **[ZIP]** Libraries [https:///files.seeedstudio.com/wiki/Grove_Shield_for_Pi_Pico_V1.0/Libraries.rar]

Tech Support

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

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