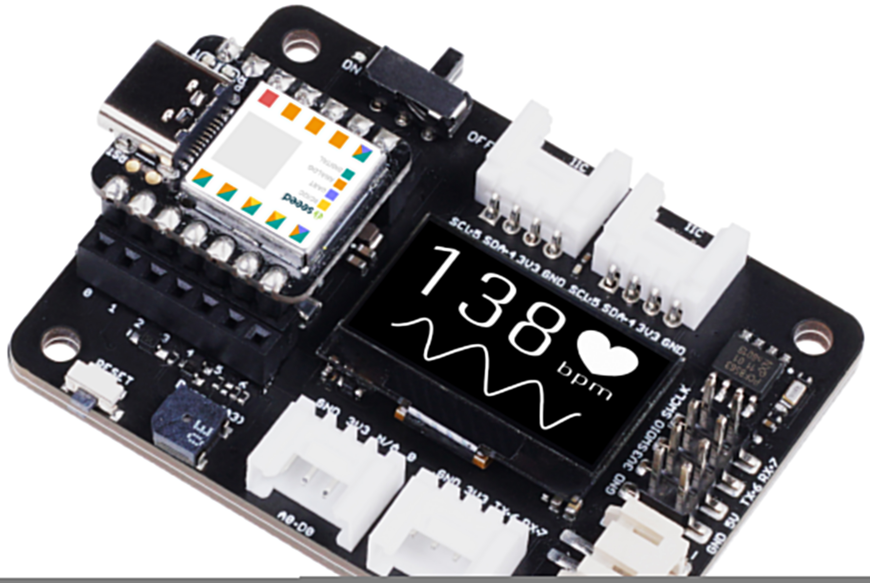
Seeeduino XIAO Expansion board



Get One Now

[https://www.seeedstudio.com/Seeeduino-XIAO-Expansion-board-p-4746.html]

A powerful functional expansion board for Seeeduino XIAO of only half Raspberry Pi 4 size. It enables build prototype and project in easy and quick way. With its rich peripherals including OLED, RTC, expandable memory, passive buzzer, RESET/User button, 5V servo connector, multiple data interfaces… you could explore infinite possibilities of Seeeduino XIAO. Circuit python is also well supported by this board.

As XIAO form factor, all XIAO boards support both Grove Shield for XIAO [https://www.seeedstudio.com/Seeeduino-XIAO-Expansion-board-p-4746.html] and XIAO Expansion board [https://www.seeedstudio.com/Seeeduino-XIAO-Expansion-board-p-4746.html]. There is a slight difference between the pins and refer to the Pinout it is easy to manage.

Seeeduino XIAO, XIAO RP2040, and XIAO BLE are compatible to the XIAO expansion board.

Features

- **Quick Prototyping:** Easy debug and Reset with RESET button and SWD pin led out to male header. 0.96”OLED, enables visual data display without PC serial monitor; Convenient plug and play Grove connectors support multiple data protocols, including IIC, Uart, Analog/Digital; No soldering needed with all pin led out.
- **Rich peripherals:** OLED display, RTC, expandable memory space, passive buzzer, user button, on board battery management chip.
- **No Soldering Needed:** All pin led out. Convenient plug and play Grove connectors support multiple data protocols, including IIC, Uart, Analog/Digital.

- **Circuit Python Supported:** Well supports circuit python. MicroSD card slot enables memory space expand, making it possible to allocate more libraries needed in prototyping and project building.
- **Mini Size:** Compact and elegant with only half Raspberry Pi 4 size, especially suitable for projects require mini size.

Specification

Item	Value
Operating voltage	5V / 3.7V Lithium Battery
Charging current	460mA (Max)
RTC timer precision	± 1.5S/DAY(25°C)
RTC battery	CR1220
Display	0.96" OLED display
Expandable memory	MicroSD card
Grove Interface	Grove IIC*2, Grove UART*1, A0/D0 Grove*1
Other External Equipment	Passive buzzer, user button, 5V servo connector

Applications

- SWD debug

- Rapid prototyping
- Data display
- Mini Size Project

Part List

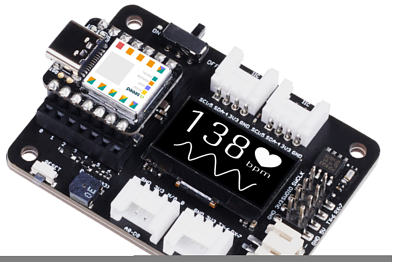
Item	Value
SeeeduinoXIAO Expansion Board	*1

Note

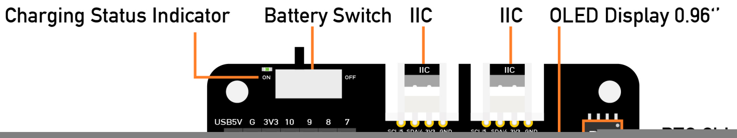
This product does not include Seeeduino XIAO and battery, please click this link to get **Seeeduino XIAO** [https://www.seeedstudio.com/Seeeduino-XIAO-Arduino-Microcontroller-SAMD21-Cortex-M0+-p-4426.html?gclid=Cj0KCQjwufn8BRCwARIsAKzP695mYBI8wwzrR8rXiJgv9QBK5DeTJGCU9bzXvzGUheFVZxqHcuw0SgYaAqDqEALw_wcB]

Getting Started

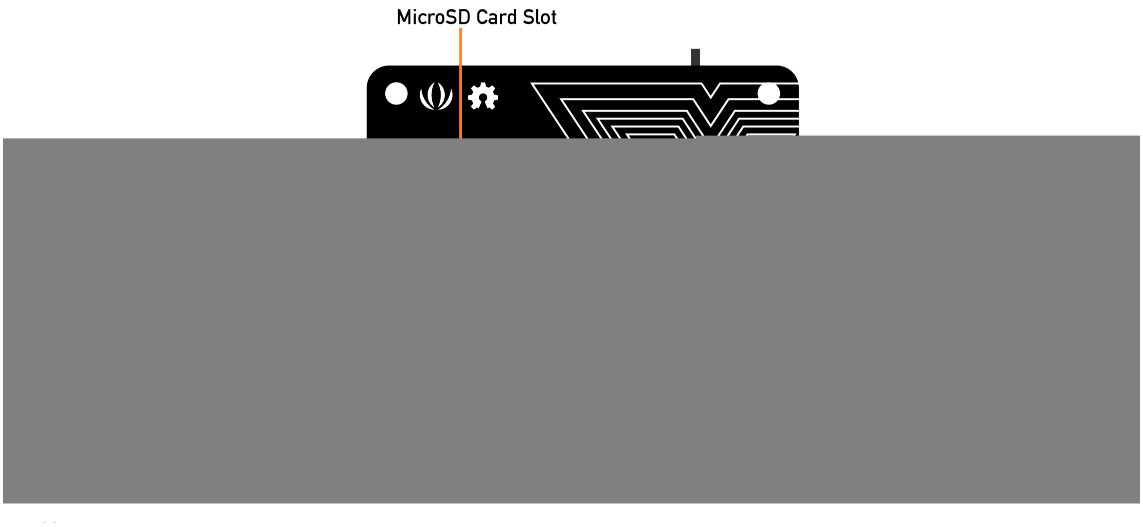
Materials Required

Seeeduino XIAO	Seeeduino XIAO Expansion board
	
Get ONE Now [https://www.seeedstudio.com/Seeeduino-XIAO-Arduino-Microcontroller-SAMD21-Cortex-M0+-p-4426.html]	**Get ONE Now** [https://www.seeedstudio.com/Seee XIAO-Expansion-board-p-4746.html]

Hardware Overview

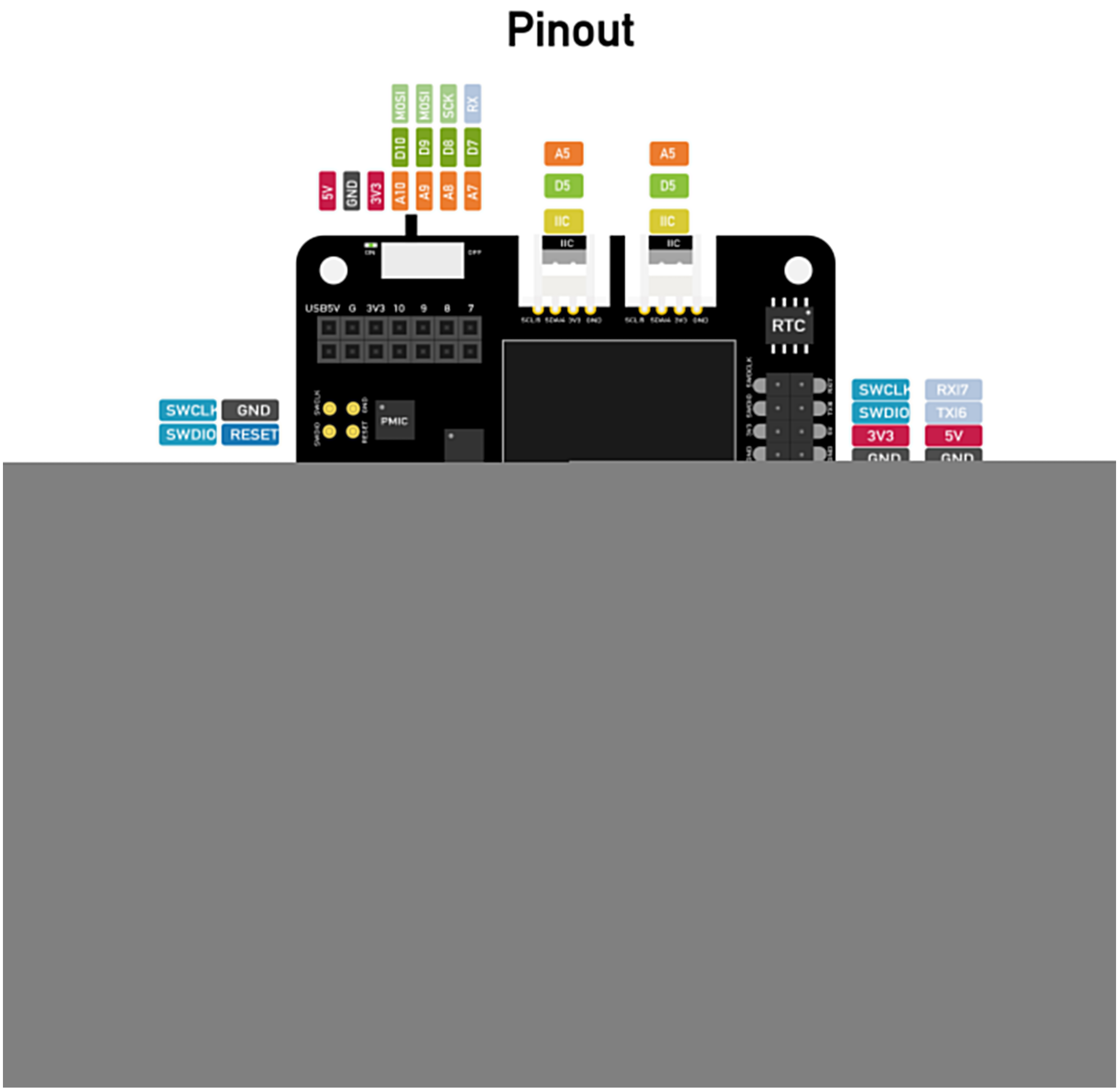


There are an external MicroSD card slot and RTC Battery Holder, the MicroSD card mostly is used for saving and run the `python.py` file, the RTC is for tracks the current time and can be used to program actions at a specific time.



Pinout Diagram

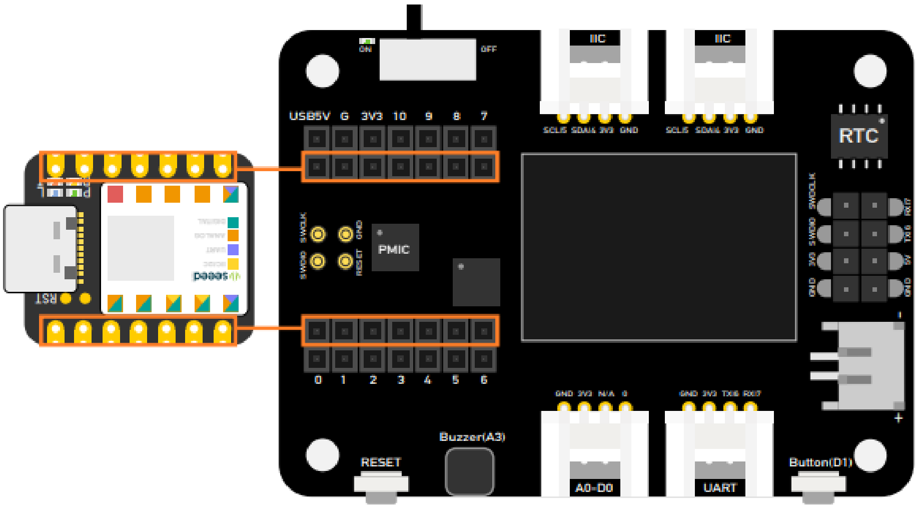
External headers-pin description for Grove-Shield for Seeeduino XIAO.



Expansion Board Usage

Connection

Put the Seeeduino XIAO on the expansion board, the XIAO green LED should light up. If you want to know more about Seeeduino XIAO, please click **Seeeduino XIAO** [https://wiki.seeedstudio.com/Seeeduino-XIAO/].

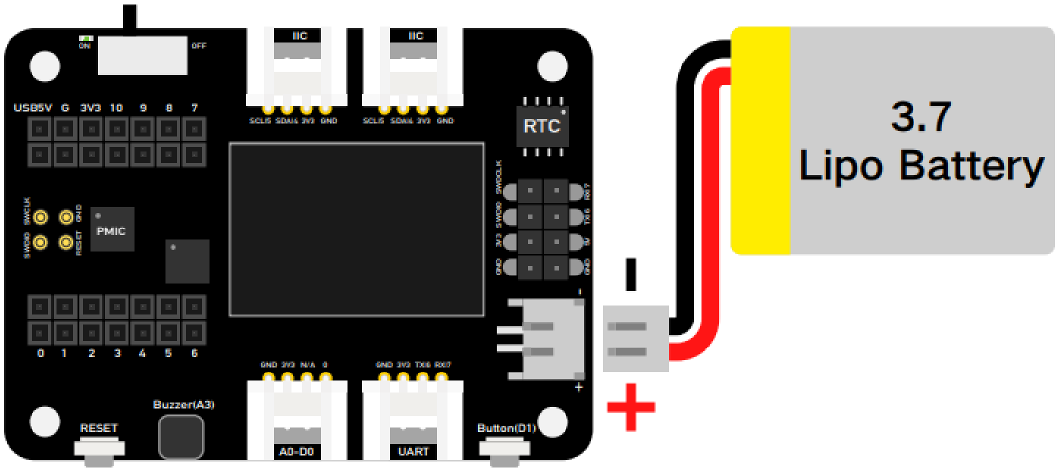


Note

Please plug the Seeeduino XIAO on the expansion board first, then plug Type-C, Remember pluing Seeeduino XIAO into the **middle of the two female header connectors**, otherwise you will damage the Seeeduino XIAO and the expansion board.

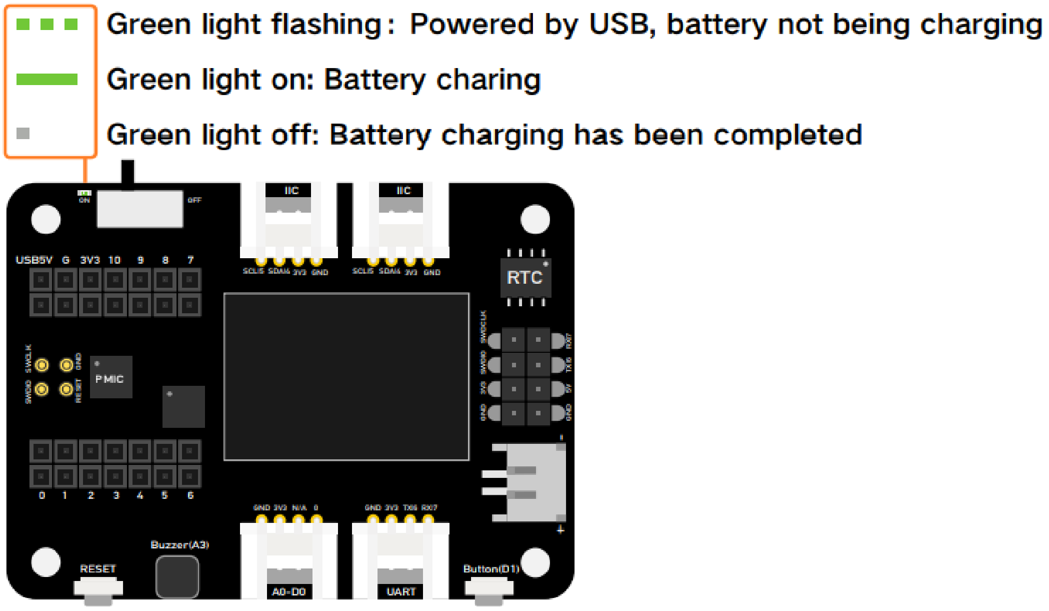
Battery usage

The Seeeduino XIAO expansion board can be powered by a battery, so if you do some demo that needs to be moved, that battery will help you to solve the power supply issue. when you plug the battery please watch out for the positive and negative anodes, follow the picture to connect the battery in case of damage the board.



In addition, the board charged the battery when you plug the battery cable and the type-C cable and switch the button to on.

As the below picture, if the LED flashing which means the battery does not charge or the board does not connect battery if the LED keeps light on which is mean the battery is charging.

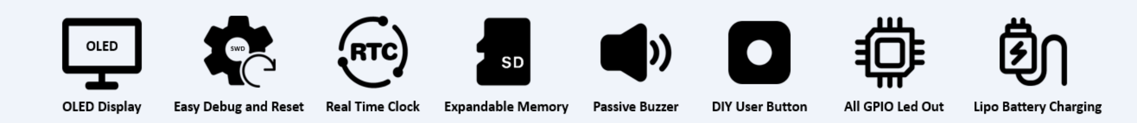

Green light flashing： Powered by USB, battery not being charging
Green light on: Battery charing
Green light off: Battery charging has been completed
ON
OFF
IIC
IIC
USB5V G 3V3 10 9 8 7
RTC
PMIC
0 1 2 3 4 5 6
Buzzer(A3)
RESET
Button(D1)
A0-D0
UART

Modules on the expansion board

Rich peripherals on board including:

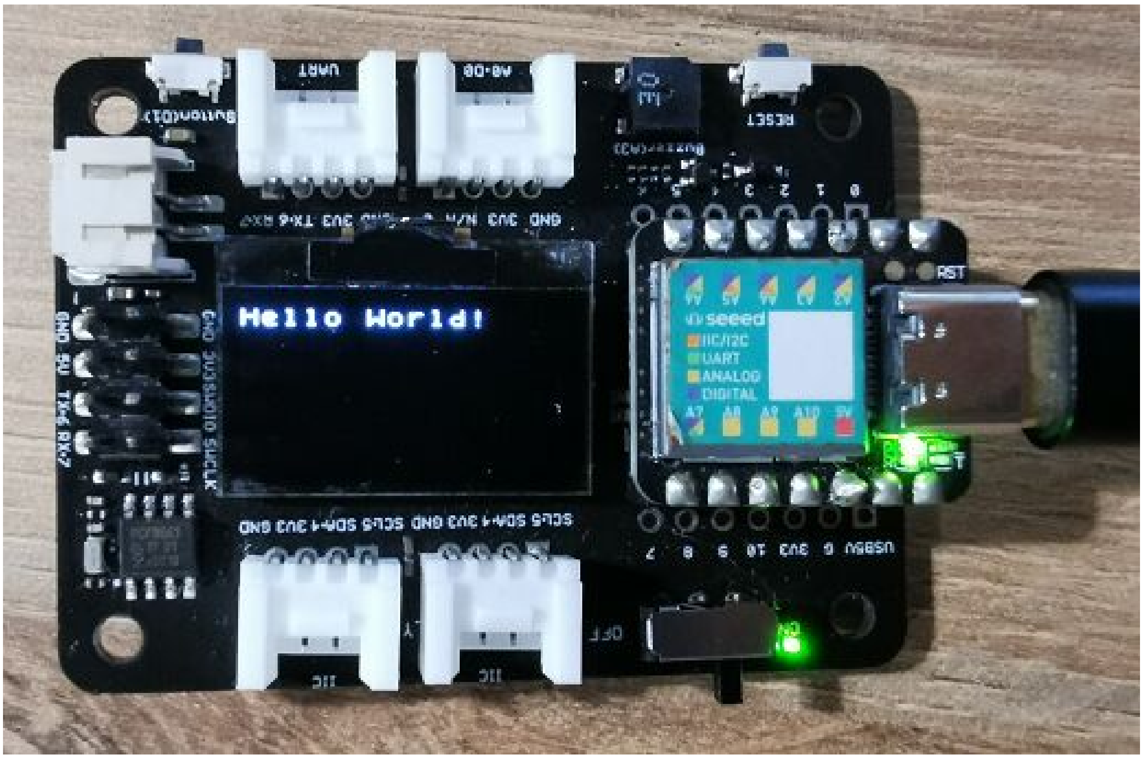
- **OLED display:** Visual data display without connecting to PC, which enables debug in a more efficient way, and builds application such as a sensor hub, data monitor system, etc.
- **RESET button:** No more jumper wire and short circuit, easy reset with just one click.
- **SWD debug:** SWD pin led out as male pin header, making debugger connection and firmware download much easier.
- **High precision RTC:** High precision real-time clock with battery backup, enable maintain accurate time when the main power is turned off.
- **Expandable memory:** With a MicroSD card slot on the back, no worry on memory limit any more when adding libraries and using circuit python.
- **User button:** Besides the RESET button, also provide with another user-defined button.
- **Passive buzzer:** Same passive buzzer on Wio Terminal, with which you could change the PMW frequency to award different beep sound to get a "buzzer music".
- **Grove connectors:** All pin led out, plug and play grove connectors support common data protocols (Grove IIC*2, Grove UART*1, A0/D0 Grove*1)
- **Lipo Battery Charging:** JST2.0mm standard lipo battery connector and battery management system, supports both USB and lipo battery power supply, and easy onboard battery recharge.

- **5V servo connector:** 5V output led out to male header for 5V servo and sensor connection.



OLED Display

This example introduces how to use the OLED display on the XIAO expansion board.



Step 1. Install the Seeeduino XIAO on the Expansion board then conect the Type-C cable.

Step 2. Install the **u8g2** [https://github.com/olikraus/U8g2_Arduino] library, this is the guide **how to install the library** [https://wiki.seeedstudio.com/How_to_install_Arduino_Library/].

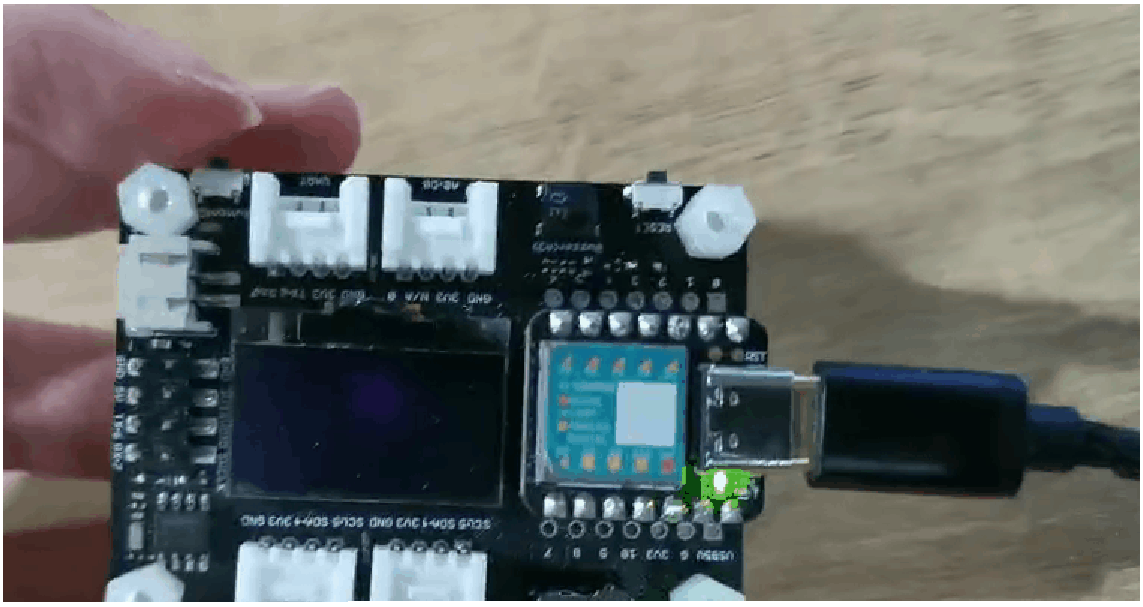
Step 3. Copy the code and stick on the Ardiono IDE then upload it.

OLED Code

```
#include <Arduino.h>
#include <U8x8lib.h>
#include <Wire.h>

U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(/* clock=*/ PIN_W

void setup(void) {
  u8x8.begin();
  u8x8.setFlipMode(1);   // set number from 1 to 3, the
}

void loop(void) {
  u8x8.setFont(u8x8_font_chroma48medium8_r);
  u8x8.setCursor(0, 0);
  u8x8.print("Hello World!");
}
```

LED control by User Button

This example introduces how to use the button on the XIAO expansion board to control the LED on the Seeeduino XIAO.



Step 1. Install the Seeeduino XIAO on the Expansion board then conect the Type-C cable.

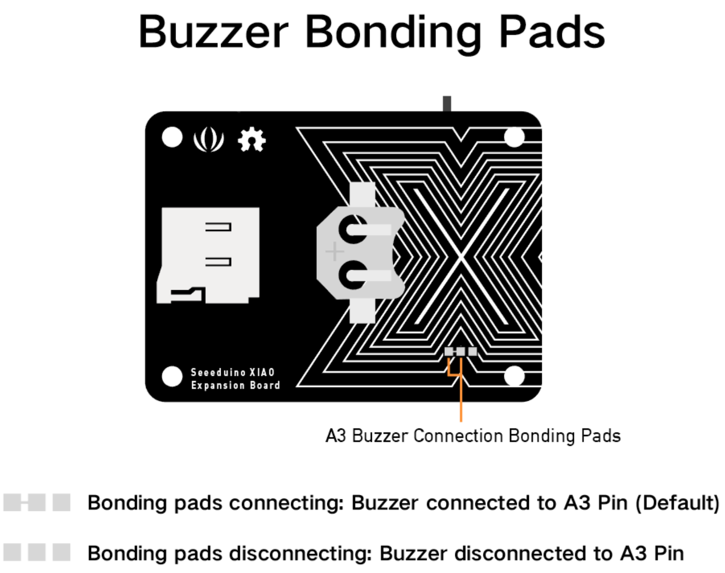
Step 2. Open Arduino IDE, Copy the code and stick on the Ardiono IDE then upload it.

Code

```
const int buttonPin = 1;     // the number of the pushbu
int buttonState = 0;         // variable for reading the

void setup() {
  // initialize the LED pin as an output:
  pinMode(LED_BUILTIN, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT_PULLUP);

}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed. If it is, the b
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(LED_BUILTIN, HIGH);
  } else {
    // turn LED off:
    digitalWrite(LED_BUILTIN, LOW);
  }

}
```

Buzzer

The Buzzer is default connected to the Pin A3, if you want to remove the buzzer function, just follow the below picture, cut off the line.


Buzzer Bonding Pads
Seeeduino XIAO Expansion Board
A3 Buzzer Connection Bonding Pads
Bonding pads connecting: Buzzer connected to A3 Pin (Default)
Bonding pads disconnecting: Buzzer disconnected to A3 Pin

Play Song with Passive Buzzer

This example uses Buzzer on the XIAO expansion board to play Happy birthday.

Step 1. Install the Seeeduino XIAO on the Expansion board then conect the Type-C cable.

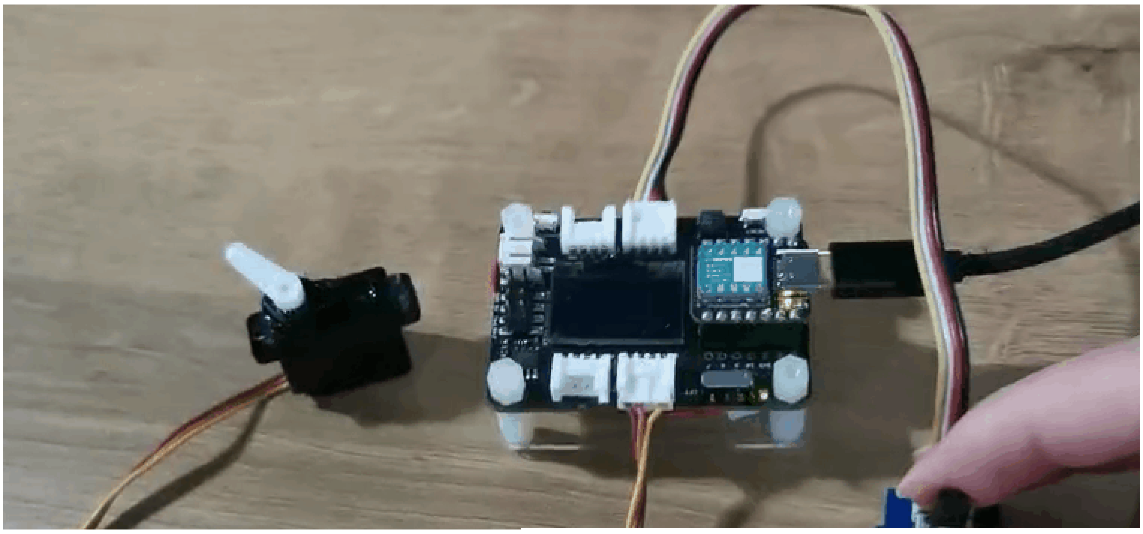
Step 2. Open Arduino IDE, Copy the code and stick on the Ardiono IDE then upload it.

Code

```
int speakerPin = A3;
int length = 28; // the number of notes
char notes[] = "GGAGcB GGAGdc GGxecBA yyecdc";
int beats[] = { 2, 2, 8, 8, 8, 16, 1, 2, 2, 8, 8, 8, 16,
int tempo = 150;
void playTone(int tone, int duration) {
  for (long i = 0; i < duration * 1000L; i += tone * 2)
    digitalWrite(speakerPin, HIGH);
    delayMicroseconds(tone);
    digitalWrite(speakerPin, LOW);
    delayMicroseconds(tone);
  }
}

void playNote(char note, int duration) {
  char names[] = {'C', 'D', 'E', 'F', 'G', 'A', 'B',
                  'c', 'd', 'e', 'f', 'g', 'a', 'b',
                  'x', 'y'
                 };
  int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 10
                  956,  834,  765,  593,  468,  346,  22
                  655 , 715
                };
  int SPEE = 5;

  // play the tone corresponding to the note name

  for (int i = 0; i < 17; i++) {
    if (names[i] == note) {
      int newduration = duration / SPEE;
      playTone(tones[i], newduration);
    }
  }
}

void setup() {
  pinMode(speakerPin, OUTPUT);
}

void loop() {
  for (int i = 0; i < length; i++) {
    if (notes[i] == ' ') {
      delay(beats[i] * tempo); // rest
    } else {
      playNote(notes[i], beats[i] * tempo);
    }
    // pause between notes
    delay(tempo);
  }
}
```

Servo Control by Rotary Angle Sensor

This example uses a rotary angle sensor to control servo via integration ports on the XIAO expansion board.



Step 1. Install the Seeeduino XIAO on the Expansion board then conect the Type-C cable.

Step 2. Connect the Servo cable to **I2C** port, rotary angle sensor to **D0**.

Step 3. Open Arduino IDE, Copy the code and stick on the Ardiono IDE then upload it.

```
#include <Servo.h>
#define ROTARY_ANGLE_SENSOR A0
#define ADC_REF 3 //reference voltage of ADC is 3v.If th
#define GROVE_VCC 3 //VCC of the grove interface is norm
#define FULL_ANGLE 300 //full value of the rotary angle

Servo myservo;  // create servo object to control a serv
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  Serial.begin(9600);
  pinMode(ROTARY_ANGLE_SENSOR, INPUT);
  myservo.attach(5);  // attaches the servo on pin 9 to
}

void loop() {

  float voltage;
  int sensor_value = analogRead(ROTARY_ANGLE_SENSOR);
  voltage = (float)sensor_value * ADC_REF / 1023;
  float degrees = (voltage * FULL_ANGLE) / GROVE_VCC;
  Serial.println("The angle between the mark and the sta
  Serial.println(degrees);
  delay(50);
  myservo.write(degrees);
}
```

RTC clock display

This example uses RTC to display the clock on the OLED.

Step 1. Install the Seeeduino XIAO on the Expansion board then conect the Type-C cable.

Step 2. Install the **u8g2** [https://github.com/olikraus/U8g2_Arduino] and **PCF8563** [https://github.com/Bill2462/PCF8563-Arduino-Library] library, this is the guide **how to install the library** [https://wiki.seeedstudio.com/How_to_install_Arduino_Library/].

Step 3. Copy the code and stick on the Ardiono IDE then upload it.

```
#include <Arduino.h>
#include <U8x8lib.h>
#include <PCF8563.h>
PCF8563 pcf;
#include <Wire.h>

U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(/* clock=*/ PIN_W

void setup() {
  Serial.begin(115200);
  u8x8.begin();
  u8x8.setFlipMode(1);
  Wire.begin();
  pcf.init();//initialize the clock
  pcf.stopClock();//stop the clock
  pcf.setYear(20);//set year
  pcf.setMonth(10);//set month
  pcf.setDay(23);//set dat
  pcf.setHour(17);//set hour
  pcf.setMinut(33);//set minut
  pcf.setSecond(0);//set second
  pcf.startClock();//start the clock
}

void loop() {
  Time nowTime = pcf.getTime();//get current time
  u8x8.setFont(u8x8_font_chroma48medium8_r);   // choose

  u8x8.setCursor(0, 0);
  u8x8.print(nowTime.day);
  u8x8.print("/");
  u8x8.print(nowTime.month);
  u8x8.print("/");
  u8x8.print("20");
  u8x8.print(nowTime.year);
  u8x8.setCursor(0, 1);
  u8x8.print(nowTime.hour);
  u8x8.print(":");
  u8x8.print(nowTime.minute);
  u8x8.print(":");
  u8x8.println(nowTime.second);
  delay(1000);
}
```

Acrylic Case for Seeeduino XIAO Expansion Board

Get One Now 🛒

[https://www.seeedstudio.com/XIAO-p-4812.html]

We made this **acrylic case** [https://www.seeedstudio.com/XIAO-p-4812.html] for protecting the Seeeduino XIAO expansion board, those are acrylic case components.

Compare with the Grove Shield for Seeeduino XIAO, the Seeeduino XIAO expansion board added a lot useful modules for the users.

This Acrylic case easy to build it up and it also can dress the case look neater.

Circuitpython on Seeeduino XIAO with expansion board

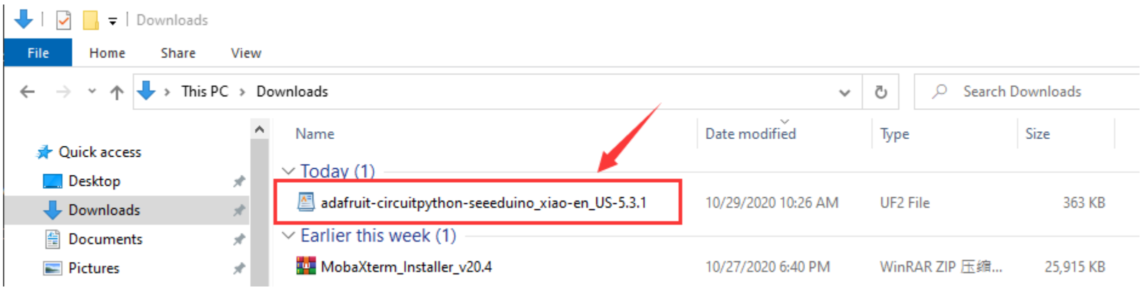
This wiki introduce how to install and run the official **CircuitPython** [https://circuitpython.org/] by Adafruit Industries on the **Seeeduino XIAO development board** [https://www.seeedstudio.com/Seeeduino-XIAO-Pre-Soldered-p-4747.html] !

CircuitPython is a programming language designed to simplify experimenting and learning to program on low-cost microcontroller boards. It makes getting started easier than ever with no upfront desktop downloads needed. Once you get your board set up, open any text editor, and get started editing code. For more info, please refer to **here** [https://learn.adafruit.com/welcome-to-circuitpython/what-is-circuitpython].

Installing CircuitPython

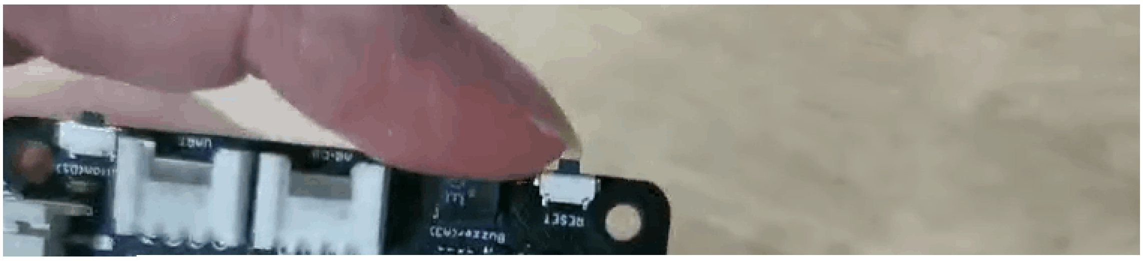
Step 1. Install the Seeeduino XIAO on the Expansion board then conect the Type-C cable.

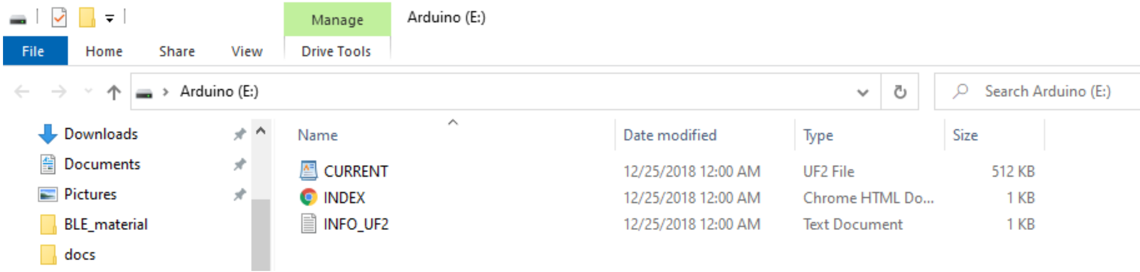
Step 2. Download the official **CircuitPython Bootloader for Seeeduino XIAO** [https://circuitpython.org/board/seeeduino_xiao/]. A `.uf2` , the flie will be store in your PC download.



Name	Date modified	Type	Size
Today (1)			
adafruit-circuitpython-seeeduino_xiao-en_US-5.3.1	10/29/2020 10:26 AM	UF2 File	363 KB
Earlier this week (1)			
MobaXterm_Installer_v20.4	10/27/2020 6:40 PM	WinRAR ZIP 压缩...	25,915 KB

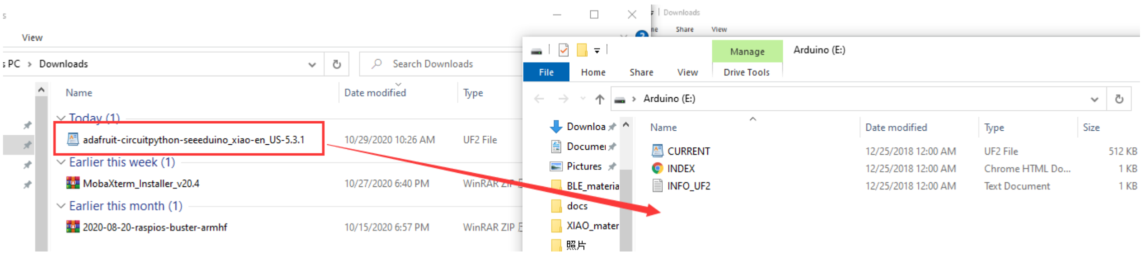
Step 3. Entering the DFU bootloader mode by press the reset button twice quickly on the XIAO expansion board, then your PC will appear Arduino drive.





Name	Date modified	Type	Size
CURRENT	12/25/2018 12:00 AM	UF2 File	512 KB
INDEX	12/25/2018 12:00 AM	Chrome HTML Do...	1 KB
INFO_UF2	12/25/2018 12:00 AM	Text Document	1 KB

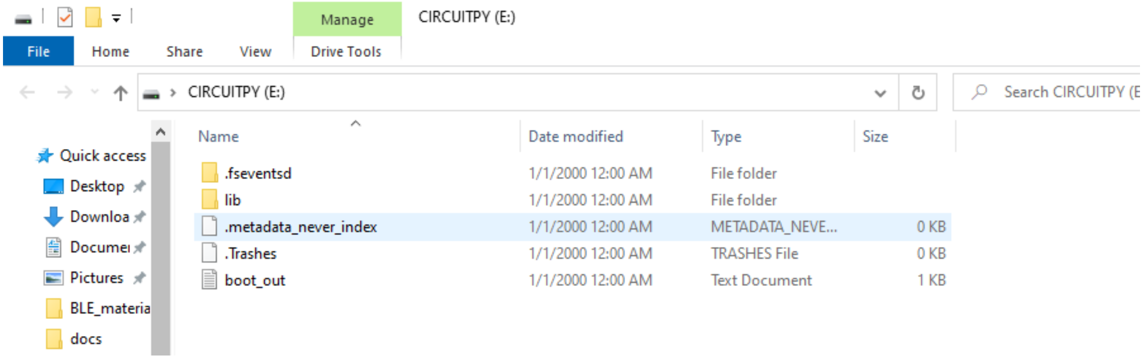
Step 4. An external drive named `Arduino` should appear in your PC. Drag the downloaded CircuitPython uf2 files to the `Arduino` drive.



Name	Date modified	Type
Today (1)		
adafruit-circuitpython-seeeduino_xiao-en_US-5.3.1	10/29/2020 10:26 AM	UF2 File
Earlier this week (1)		
MobaXterm_Installer_v20.4	10/27/2020 6:40 PM	WinRAR ZIP
Earlier this month (1)		
2020-08-20-raspios-buster-armhf	10/15/2020 6:57 PM	WinRAR ZIP

Name	Date modified	Type	Size
CURRENT	12/25/2018 12:00 AM	UF2 File	512 KB
INDEX	12/25/2018 12:00 AM	Chrome HTML Do...	1 KB
INFO_UF2	12/25/2018 12:00 AM	Text Document	1 KB

Step 5. Once loaded the CircuitPython bootloader, unplug the USB Type-C and re-connect. A new external drive called `CIRCUITPY` should be appear.



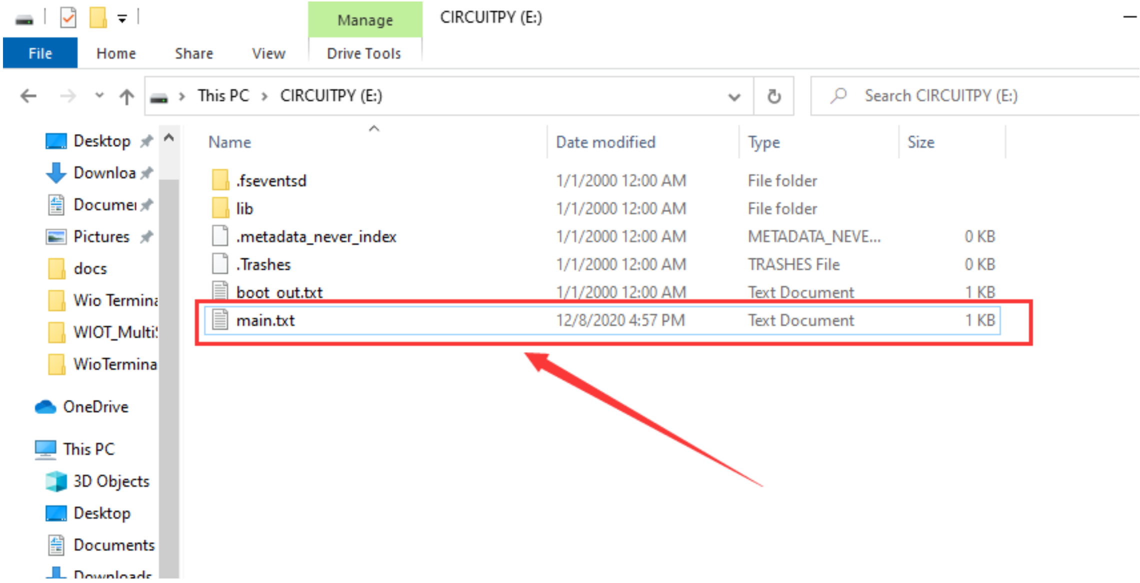
Name	Date modified	Type	Size
.fseventsd	1/1/2000 12:00 AM	File folder	
lib	1/1/2000 12:00 AM	File folder	
.metadata_never_index	1/1/2000 12:00 AM	METADATA_NEVE...	0 KB
.Trashes	1/1/2000 12:00 AM	TRASHES File	0 KB
boot_out	1/1/2000 12:00 AM	Text Document	1 KB

Step 6. Now, CircuitPython is loaded on Seeeduino XIAO! All you need to do it's to write you python program and name it `main.py` and drag it onto the `CIRCUITPY` drive.

CircuitPyhton Blink example

There is a simple example introduce how to use the CirsuitPython on the seeeduino XIAO.

Step 1 Create a txt file name `main` on the `CIRCUITPY` drive.



Name	Date modified	Type	Size
.fseventsd	1/1/2000 12:00 AM	File folder	
lib	1/1/2000 12:00 AM	File folder	
.metadata_never_index	1/1/2000 12:00 AM	METADATA_NEVE...	0 KB
.Trashes	1/1/2000 12:00 AM	TRASHES File	0 KB
boot_out.txt	1/1/2000 12:00 AM	Text Document	1 KB
main.txt	12/8/2020 4:57 PM	Text Document	1 KB

Note

The `main` name is one of these: **code.txt**, **code.py**, **main.py**, **main.txt**, there is more detail about **this behavior**

> [https://circuitpython.readthedocs.io/en/6.0.x/README.html#behavior].

Step 2 Paste the code on the `main` file then save it, you will see the orange LED blinking on the Seeeduino XIAO board.

Code

```python
import time
import board
from digitalio import DigitalInOut, Direction

led = DigitalInOut(board.D13)
led.direction = Direction.OUTPUT

while True:
    led.value = True
    time.sleep(1)
    led.value = False
    time.sleep(1)
```

MicroSD card for circuitpython

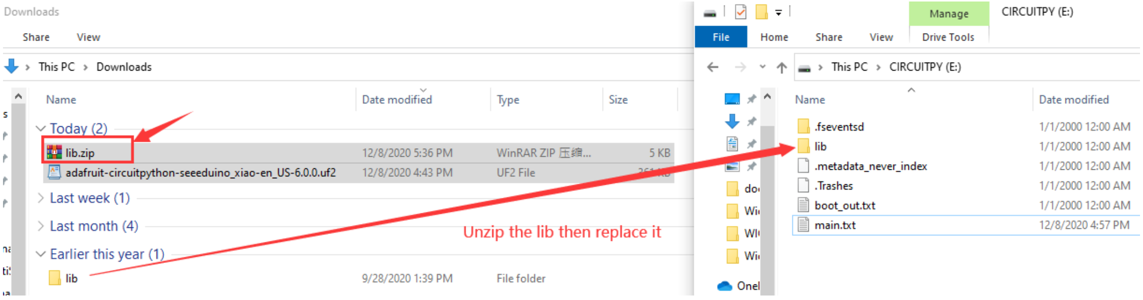
The Seedruino XIAO build-in about 40 KB flash, but it may have not enough space to store the large size python code file, fortunately, Seeeduino XIAO expansion board built-in an MicroSD card slot for extending the store space, so you can follow this instruction to learn how to run the circuitpython on the MicroSD card.

> **Note**
> The MicroSD card system format is FAT or exFAT. If you use other MicroSD card system format that will cause the MicroSD card can not be recognized.

Step 1. Prepare a micro SD Card to plug in the XIAO expansion board.

Step 2. Assume you do not download the **circuitPython file** [https://circuitpython.org/board/seeeduino_xiao/] yet, please refer to **Installing CircuitPython** [https://wiki.seeedstudio.com/Seeeduino-XIAO-Expansion-Board/#installing-circuitpython] chapter.

Step 3. Download **lib** [https://files.seeedstudio.com/wiki/Seeeduino-XIAO-Expansion-Board/lib.zip] unzips the file, then replace it with the new lib in the `CIRCUITPY`.



Name	Date modified	Type	Size
Today (2)			
lib.zip	12/8/2020 5:36 PM	WinRAR ZIP 压缩...	5 KB
adafruit-circuitpython-seeeduino_xiao-en_US-6.0.0.uf2	12/8/2020 4:43 PM	UF2 File	
Last week (1)			
Last month (4)			
Earlier this year (1)			
lib	9/28/2020 1:39 PM	File folder	

Unzip the lib then replace it

Name	Date modified
.fseventsd	1/1/2000 12:00 AM
lib	1/1/2000 12:00 AM
.metadata_never_index	1/1/2000 12:00 AM
.Trashes	1/1/2000 12:00 AM
boot_out.txt	1/1/2000 12:00 AM
main.txt	12/8/2020 4:57 PM

Step 4. Download the **main.py** [https://files.seeedstudio.com/wiki/Seeeduino-XIAO-Expansion-Board/main.py] file in the `CIRCUITPY` drive.

The `main.py` code

```python
import sd
f = open("/sd/hello.txt", "r")   ## read the file from SD
print(f.read())
```

Step 5. Download the **sd.py** [https://files.seeedstudio.com/wiki/Seeeduino-XIAO-Expansion-

Board/sd.py] file in the `CIRCUITPY` drive.

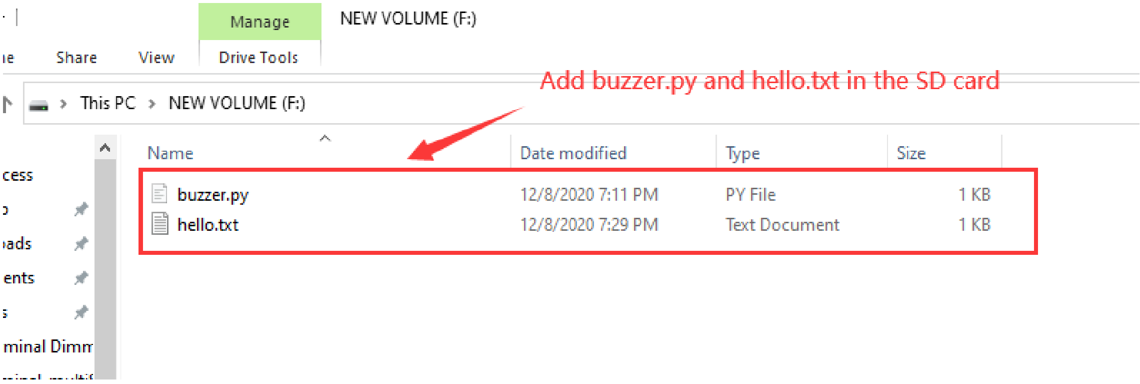
The `sd.py` code

```
import os
import adafruit_sdcard
import board
import busio
import digitalio
import storage
import sys

# Connect to the card and mount the filesystem for Seeed
spi = busio.SPI(board.SCK, board.MOSI, board.MISO)
cs = digitalio.DigitalInOut(board.D2)
sdcard = adafruit_sdcard.SDCard(spi, cs)
vfs = storage.VfsFat(sdcard)
storage.mount(vfs, "/sd")
sys.path.append("/sd")
sys.path.append("/sd/lib")  ## switch to the path to SD
```

Buzzer Example

This example for test the buzzer via the run the `buzzer.py` in the MicroSD card.

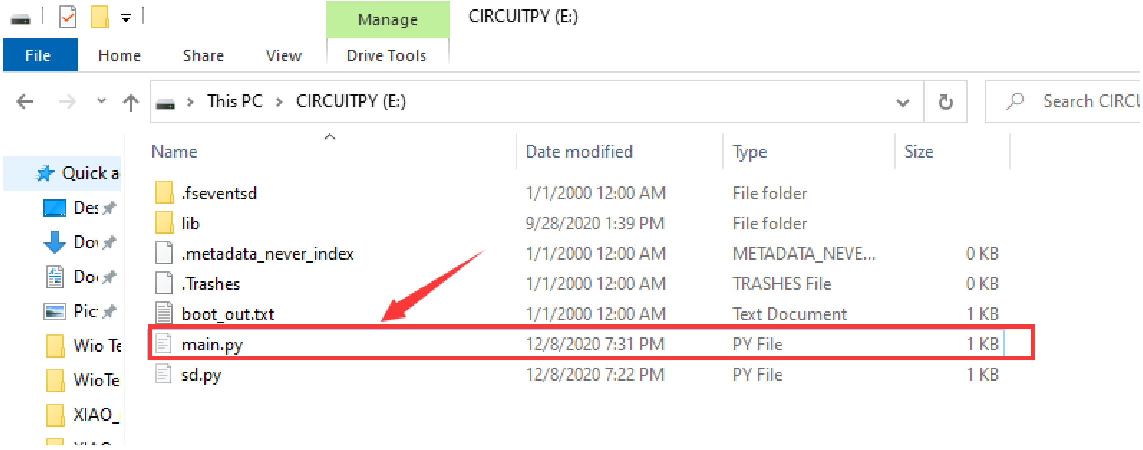
Step 1. You can just paste **buzzer.py** [https://files.seeedstudio.com/wiki/Seeeduino-XIAO-Expansion-Board/buzzer.py] in the MicroSD card.



Add buzzer.py and hello.txt in the SD card

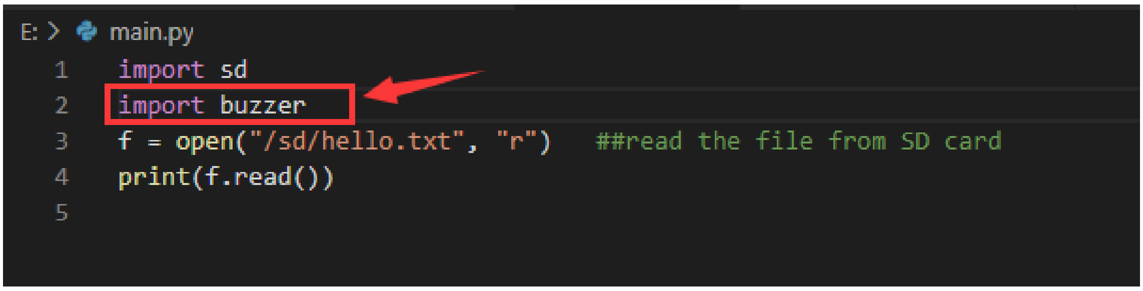
Name	Date modified	Type	Size
buzzer.py	12/8/2020 7:11 PM	PY File	1 KB
hello.txt	12/8/2020 7:29 PM	Text Document	1 KB

Step 2. Open `main.py` in the `CIRCUITPY` drive.



Name	Date modified	Type	Size
.fseventsd	1/1/2000 12:00 AM	File folder	
lib	9/28/2020 1:39 PM	File folder	
.metadata_never_index	1/1/2000 12:00 AM	METADATA_NEVE...	0 KB
.Trashes	1/1/2000 12:00 AM	TRASHES File	0 KB
boot_out.txt	1/1/2000 12:00 AM	Text Document	1 KB
main.py	12/8/2020 7:31 PM	PY File	1 KB
sd.py	12/8/2020 7:22 PM	PY File	1 KB

Step 3. Add `import buzzer` in the main.py file.



```
import sd
import buzzer
f = open("/sd/hello.txt", "r")    ##read the file from SD card
print(f.read())

```

When you finish all the step, the buzzer will work. If you to run other python files in the MicroSD card, please imitate the example.

> **Note**
> If you want back to Arduino mode, you just need to upload any programme on the Arduino IDE.

Demo

Project 1 - Remote control fan

Overview

This wiki introduce how to make a Mini fan to plase on your room keep cool.

Feature

- Automatic swing fan

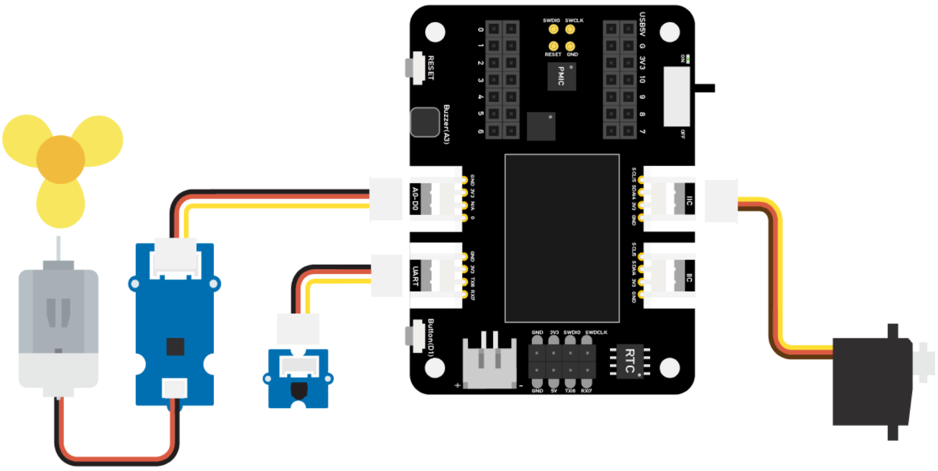
Component required

- **Seeeduino XIAO** [https://www.seeedstudio.com/Seeeduino-XIAO-Arduino-Microcontroller-SAMD21-Cortex-M0+-p-4426.html]

- **Seeeduino XIAO expansion board** [https://www.seeedstudio.com/Seeeduino-XIAO-Expansion-board-p-4746.html]
- **Grove mini fan** [https://www.seeedstudio.com/Grove-Mini-Fan-v1-1.html]
- **Grove-Servo** [https://www.seeedstudio.com/Grove-Servo.html]
- **Grove - IR (Infrared) Receiver** [https://www.seeedstudio.com/Grove-Infrared-Receiver.html]

Hardware Connection

Please follow the same color line to connect each sensor on the board. Please connect the fan grove cable to D0, servo grove cable to **I2C**, IR grove cable to **D7**.



Arduino Instructions

Step 1. Follow the connection picture connect all the sensor on the board.

Step 2. Install the **Arduino-IRremote** [https://github.com/Seeed-Studio/Seeed_Arduino_IRSendRev] library, this is the guide **how to install the library** [https://wiki.seeedstudio.com/How_to_install_Arduino_Library/].

Step 4. Copy the code stick on the Aruino IDE then upload it.

Step 5. Place the Fan in the safety position, try to press the button make sure it can work safely.

Code

```
#include <IRremote.h>
#include <Servo.h>

Servo myservo;  // create servo object to control a serv
int RECV_PIN = 7; // set pin 2 as IR control

IRrecv irrecv(RECV_PIN);

decode_results results;

int pos = 90;    // variable to store the servo position
int fanPin = 0;  // set D6 as control switch
int fanState = LOW;
int IO = 0;

void setup()
{
  Serial.begin(9600);
  Serial.println("Enabling IRin");  // remind enabling I
  irrecv.enableIRIn(); // Start the receiver
  Serial.println("Enabled IRin");
  myservo.attach(5);  // attaches the servo on pin 2 to
  pinMode(fanPin, OUTPUT);

}

```

```
void loop() {
  if (irrecv.decode(&results)) { //checking IR signal
    if (results.value == 2155829415) {    // Power off/o
      IO++;
      if (IO % 2 == 0) {
        fanState = HIGH;
        digitalWrite(fanPin, fanState);
        delay(100);
      }
      else {
        fanState = LOW;
        digitalWrite(fanPin, fanState);
        delay(100);
      }
    }

    if (results.value == 2155821255 ) {   // fan swing
      for (pos; pos <= 89; pos += 1) { // goes from 0 de
        // in steps of 1 degree
        myservo.write(pos);              // tell servo t

        delay(40);                       // waits 15ms f
        if (irrecv.decode(&results)) {
          irrecv.resume();
          if (results.value == 2155870215)
            break;
        }
      }
    }

    if (results.value == 2155870215 ) {   // fan swing
      for (pos; pos >= 1; pos -= 1) { // goes from 90 de
        myservo.write(pos);              // tell servo t
        delay(40);                       // waits 15ms f

        if (irrecv.decode(&results)) {
          irrecv.resume();
          if (results.value == 2155821255)
            break;
        }
      }
    }
    Serial.println(pos);
    Serial.println(results.value, HEX);
    Serial.println(results.value);
    irrecv.resume();                    //recive next in
  }
  delay(100);
}
```

Project 2 - Remote control car

Overview

This wiki introduce how to make a remote control car.

Feature

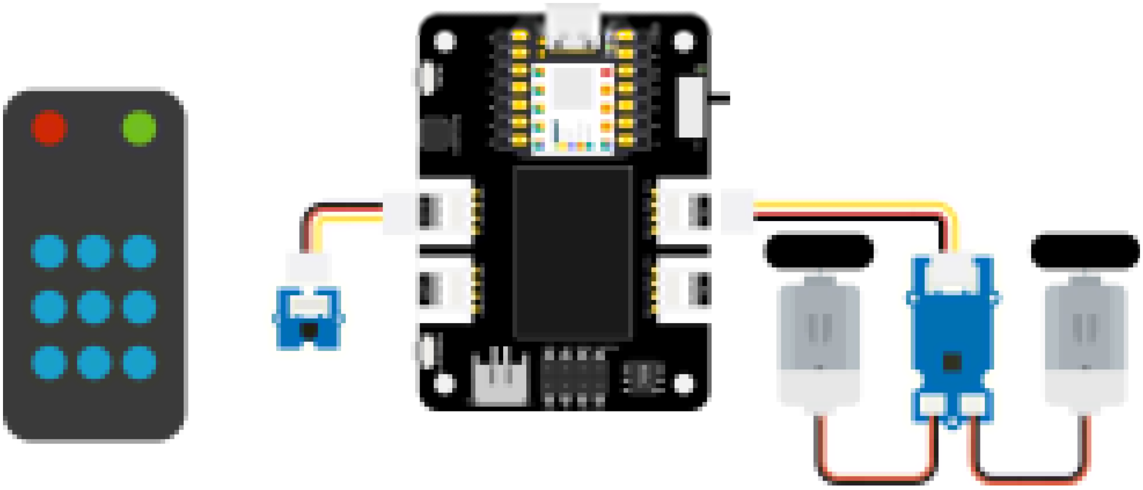
- Mini size car easy to across the narrow road

Component required

- **Seeeduino XIAO** [https://www.seeedstudio.com/Seeeduino-XIAO-Arduino-Microcontroller-SAMD21-Cortex-M0+-p-4426.html]
- **Seeeduino XIAO expansion board** [https://www.seeedstudio.com/Seeeduino-XIAO-Expansion-board-p-4746.html]
- **Grove - I2C Mini Motor Driver** [https://www.seeedstudio.com/Grove-I2C-Mini-Motor-Driver.html]
- **DC Motor** [https://www.seeedstudio.com/130-DC-Motor-p-2023.html]
- **Grove - IR (Infrared) Receiver** [https://www.seeedstudio.com/Grove-Infrared-Receiver.html]

Hardware Connection

Please follow the same color line to connect each sensor on the board. Please connect the IR sensor grove cable to D0, Mini Motor Driver grove cable to I2C.



Arduino Instructions

Step 1. Follow the connection picture connect all the sensor on the board.

Step 2. Download the **Aruidno IDE** [https://www.arduino.cc/en/Main/software]

Step 3. Install the **Arduino-IRremote** [https://github.com/Seeed-Studio/Seeed_Arduino_IRSendRev] and **Motor driver** [https://github.com/Seeed-Studio/Drv8830_Motor_Driver] library, this is the guide **how to install the library** [https://wiki.seeedstudio.com/How_to_install_Arduino_Library/].

Step 4. Copy the code stick on the Aruino IDE then upload it.

Code

```
#include <Arduino.h>
#include <U8g2lib.h>
#include <IRremote.h>
#include <SparkFunMiniMoto.h>  // Include the MiniMoto l
// Create two MiniMoto instances, with different address
MiniMoto motor0(0xC4); // A1 = 1, A0 = clear
MiniMoto motor1(0xC0); // A1 = 1, A0 = 1 (default)
```

```

#define FAULTn  1     // Pin used for fault detection.

int RECV_PIN = 0; // set pin 2 as IR control
IRrecv irrecv(RECV_PIN);
decode_results results;

void setup() {
  Serial.begin(9600);
  Serial.println("Enabling IRin");  // remind enabling I
  irrecv.enableIRIn(); // Start the receiver
  pinMode(FAULTn, INPUT);
}

void loop() {
  if (irrecv.decode(&results)) { //checking IR signal
    if (results.value == 2155862055) {
      //Forward  2155862055
      motor0.drive(-600);
      motor1.drive(600);
      delayUntil(20);
    }
    if (results.value == 2155813095) {
      //Brake   2155813095
      motor0.brake();
      motor1.brake();
      delay(100);
    }
    if (results.value == 2155823295) {
      //backward  2155823295
      motor0.drive(600);
      motor1.drive(-600);
      delayUntil(20);
    }
    if (results.value == 2155829415) {
      //Stop  2155829415
      motor0.stop();
      motor1.stop();
      delay(100);
    }
    if (results.value == 2155821255) {
      //turn right   2155821255
      motor0.drive(600);
      motor1.drive(600);
      delayUntil(20);
    }
    if (results.value == 2155837575) {
      //turn left    2155837575
      motor0.drive(-600);
      motor1.drive(-600);
      delayUntil(20);
    }
    irrecv.resume();                    //recive next in

  }
  delay(100);
}

void delayUntil(unsigned long elapsedTime) {
  unsigned long startTime = millis();
  while (startTime + elapsedTime > millis()) {
    if (digitalRead(FAULTn) == LOW) {
      byte result = motor0.getFault();
      result = motor1.getFault();
    }
  }
}
```

Project 3 - Fingerprint unlocks treasure box - SeeeduinoXIAO

Overview

This box can store your important stuff, and you are not worried about some people will take your thing, the box has the fingerprint function to protecting your thing, and if the fingerprint authorization failed, the buzzer will alarm and the LED ring will display the red colour, only your finger has registered on the board when the begin, then put your finger on the board, when fingerprint pass the authorization, the LED ring will display green colour.

Feature

- Easy to record your fingerprint
- LED ring can remind you the lock state
- The OLED screen can display the current information
- The buzzer can remind you the fingerprint whether pass authorization

Component required

- **Seeeduino XIAO** [https://www.seeedstudio.com/Seeeduino-XIAO-Arduino-Microcontroller-SAMD21-Cortex-M0+-p-4426.html]

- **Seeeduino XIAO expansion board** [https://www.seeedstudio.com/Seeeduino-XIAO-Expansion-board-p-4746.html]
- **Seeed Grove - Capacitive Fingerprint Scanner/Sensor** [https://www.hackster.io/products/buy/80263?s=BAhJIhMzNzA5MzAsUHJvamVjdAY6BkVG%0A]
- **Seeed Grove RGB LED Ring - 24** [https://www.hackster.io/products/buy/80264?s=BAhJIhMzNzA5MzAsUHJvamVjdAY6BkVG%0A]
- **Seeed Grove - Servo** [https://www.hackster.io/products/buy/80265?s=BAhJIhMzNzA5MzAsUHJvamVjdAY6BkVG%0A]

Hardware Connection

Please follow the same color line to connect each sensor on the board. Please connect the IR sensor grove cable to D0, Mini Motor Driver grove cable to I2C.

Arduino Instructions

Step 1. Follow the connection picture connect all the sensor on the board.

Step 2. Download the **Aruidno IDE** [https://www.arduino.cc/en/Main/software]

Step 3. Install the **u8g2** [https://github.com/olikraus/U8g2_Arduino], **Servo** [https://github.com/arduino-libraries/Servo], **Seeed_Arduino_KCT202** [https://github.com/Seeed-Studio/Seeed_Arduino_KCT202] and **Seeed_LED_Ring** [https://github.com/Seeed-Studio/Seeed_LED_Ring] library, this is the guide **how to install the library** [https://wiki.seeedstudio.com/How_to_install_Arduino_Library/].

Step 4. Copy the code stick on the Aruino IDE then upload it.

Demonstration

1. Record your fingerprint

The screen will display finger recording at the begin, you just need to put your finger on the finger device, after that, the program will analyze your fingerprint, then finish registered.

1. Identity authorization(pass certification)

The screen will display "Please verify", you need to put your finger on the fingerprint device, then the LED ring will turn to green colour.

1. Identity authorization(unpass certification)

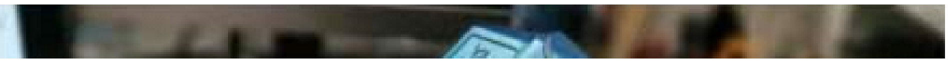
If other people put their finger on it, the LED ring will turn to red colour and the board will display "Identity deny" meanwhile the alarm will be work.

Code

```
#include <Servo.h>
#include <Arduino.h>
#include <U8x8lib.h>
#include "ATSerial.h"
#include "Protocol.h"
#include "KCT202.h"
#include "Adafruit_NeoPixel.h"

#define PIXEL_PIN    2    // Digital IO pin connected t
#define PIXEL_COUNT 24
#define debug SerialUSB
#define uart  Serial1
FingerPrint_KCT202<Uart, Serial_> kct202;
Adafruit_NeoPixel strip = Adafruit_NeoPixel(PIXEL_COUNT

Servo myservo;

Protocol_oprt oprt;
uint8_t err_code = 0;
uint8_t param[10];
uint32_t param_len;
int pos = 0;
const int buttonPin = 1;
int buttonState = 0;
int BuzzerPin = A3;

U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(/* reset=*/ U8X8_

void setup(void) {
  Serial.begin(115200);
  strip.setBrightness(255);
  strip.begin();
  strip.show(); // Initialize all pixels to 'off'
  colorWipe(strip.Color(125, 0, 125), 50);
  u8x8.begin();
  u8x8.setFlipMode(0);
  debug.begin(115200);
  pinMode(buttonPin, INPUT_PULLUP);
  pinMode(BuzzerPin, OUTPUT);
  kct202.begin(uart, debug);
  myservo.attach(0);
  myservo.write(0);
  kct202.autoRegisterFingerPrint(1, 4, LED_OFF_AFTER_GE

  u8x8.setFont(u8x8_font_chroma48medium8_r);
  u8x8.setCursor(0, 3);
  u8x8.print("finger recording");
  if (0 == kct202.getRegisterResponAndparse()) {
    debug.println("Register ok!");
    u8x8.setFont(u8x8_font_chroma48medium8_r);
    u8x8.setCursor(0, 3);
    u8x8.print("    be ready    ");
    delay(500);
    colorWipe(strip.Color(0, 125, 125), 50);
    u8x8.setCursor(0, 3);
    u8x8.print("   *** 3 ***    ");
    delay(500);
    u8x8.setCursor(0, 3);
    u8x8.print("   *** 2 ***    ");
    delay(500);
    u8x8.setCursor(0, 3);
    u8x8.print("   *** 1 ***    ");
    delay(500);
    u8x8.setCursor(0, 3);
    u8x8.print("   Registered");
    delay(800);
  }
}

void loop(void) {
  uint16_t finger_num = 0;

  kct202.autoVerifyFingerPrint(CHECK_ALL_FINGER_TEMP,
                               LED_OFF_AFTER_GET_GRAGH
  u8x8.setFont(u8x8_font_chroma48medium8_r);
  u8x8.setCursor(0, 3);
  u8x8.print(" Please verify  ");

  if (0 == kct202.getVerifyResponAndparse(finger_num))
    debug.println("Verify ok!");
    debug.print("Your finger temp id = ");
    debug.println(finger_num, HEX);
    colorWipe(strip.Color(0, 255, 30), 50);
    u8x8.setFont(u8x8_font_chroma48medium8_r);
    u8x8.setCursor(0, 3);
    u8x8.print("Identity comfirm");
    delay(800);

    analogWrite(BuzzerPin, 128);
    delay(100);
    analogWrite(BuzzerPin, 0);
    delay(100);
    analogWrite(BuzzerPin, 128);
    delay(100);
    analogWrite(BuzzerPin, 0);
    delay(100);

    for (pos = 0; pos <= 90; pos += 1) {
      myservo.write(pos);
      delay(15);
    }
    while (1) {
      //      pinMode(buttonPin, INPUT);
      buttonState = digitalRead(buttonPin);
      u8x8.setFont(u8x8_font_chroma48medium8_r);
      u8x8.setCursor(0, 3);
      u8x8.print("Please close    ");
      Serial.println(pos);
      Serial.println(buttonState);
      if (buttonState == LOW && pos == 91) {
        for (pos = 91; pos >= 0; pos -= 1) { // goes fr
          u8x8.setFont(u8x8_font_chroma48medium8_r);
          u8x8.setCursor(0, 3);
          u8x8.print("Lock closing    ");
          myservo.write(pos);              // tell serv
          delay(15);                       // waits 15m
        }
        colorWipe(strip.Color(255, 0, 0), 50);
        break;
      }
    }
  }

  else {
    colorWipe(strip.Color(255, 0, 0), 50);
    u8x8.setFont(u8x8_font_chroma48medium8_r);
    u8x8.setCursor(0, 3);
    u8x8.print(" Identity deny ");
    //    myservo.write(0);
    delay(200);

  analogWrite(BuzzerPin, 250);
  delay(2000);
    analogWrite(BuzzerPin, 0);
  delay(100);

    u8x8.setCursor(0, 3);
    u8x8.print("  Please retry  ");
    delay(1500);
  }
}

void colorWipe(uint32_t c, uint8_t wait) {
  for (uint16_t i = 0; i < strip.numPixels(); i++) {
    strip.setPixelColor(i, c);
    strip.show();
    delay(70);
  }
}
```

Project 4 - Seeedruino XIAO Expansion Board - mjolnir

Overview



This hammer is simulated Mjolnir, you need you to record your fingerprint on this device then you will become its master. The hammer needs a magnet to adsorb on the grove - electromagnet until its master to unlock via fingerprint, the hammer can take away.

Component required

- **Seeeduino XIAO** [https://www.seeedstudio.com/Seeeduino-XIAO-Arduino-Microcontroller-SAMD21-Cortex-M0+-p-4426.html]
- **Seeeduino XIAO expansion board** [https://www.seeedstudio.com/Seeeduino-XIAO-Expansion-board-p-4746.html]

- **Seeed Grove - Capacitive Fingerprint Scanner/Sensor** [https://www.hackster.io/products/buy/81052?s=BAhJIhMzNzQxMDUsUHJvamVjdAY6BkVG%0A]
- **Seeed Grove - Electromagnet** [https://www.hackster.io/products/buy/32769?s=BAhJIhMzNzQxMDUsUHJvamVjdAY6BkVG%0A]

Hardware Connection

Please follow the same color line to connect each sensor on the board. Please connect the IR sensor grove cable to D0, Mini Motor Driver grove cable to I2C.

Arduino Instructions

Step 1. Follow the connection picture connect all the sensor on the board.

Step 2. Download the **Aruidno IDE** [https://www.arduino.cc/en/Main/software]

Step 3. Install the **u8g2** [https://github.com/olikraus/U8g2_Arduino] and **Seeed_Arduino_KCT202** [https://github.com/Seeed-Studio/Seeed_Arduino_KCT202] library, this is the guide **how to**

install the library [https://wiki.seeedstudio.com/How_to_install_Arduino_Library/].

Step 4. Copy the code stick on the Aruino IDE then upload it.

Code

```
#include <U8x8lib.h>
#include "ATSerial.h"
#include "Protocol.h"
#include "KCT202.h"

#define debug SerialUSB
#define uart  Serial1
FingerPrint_KCT202<Uart, Serial_> kct202;

Protocol_oprt oprt;
uint8_t err_code = 0;
uint8_t param[10];
uint32_t param_len;

int Electromagnet = 0;

U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(/* reset=*/ U8X8_

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.

  u8x8.begin();
  u8x8.setFlipMode(0);
  debug.begin(115200);
  pinMode(Electromagnet, OUTPUT);
  digitalWrite(Electromagnet, HIGH);  // turn the Electr
  kct202.begin(uart, debug);
  kct202.autoRegisterFingerPrint(1, 4, LED_OFF_AFTER_GET_
  u8x8.setFont(u8x8_font_chroma48medium8_r);
  u8x8.setCursor(0, 3);
  u8x8.print("finger recording");
  if (0 == kct202.getRegisterResponAndparse()) {
    u8x8.setFont(u8x8_font_chroma48medium8_r);
    u8x8.setCursor(0, 3);
    u8x8.print("    be ready    ");
    delay(500);
    u8x8.setCursor(0, 3);
    u8x8.print("   *** 3 ***    ");
    delay(500);
    u8x8.setCursor(0, 3);
    u8x8.print("   *** 2 ***    ");
    delay(500);
    u8x8.setCursor(0, 3);
    u8x8.print("   *** 1 ***    ");
    delay(500);
    u8x8.setCursor(0, 3);
    u8x8.print("   Registered");
    delay(800);
  }

}

  // the loop routine runs over and over again forever:
  void loop() {

    uint16_t finger_num = 0;
    kct202.autoVerifyFingerPrint(CHECK_ALL_FINGER_TEMP,
    u8x8.setFont(u8x8_font_chroma48medium8_r);
    u8x8.setCursor(0, 3);
    u8x8.print(" Please verify  ");

    if (0 == kct202.getVerifyResponAndparse(finger_num))
      u8x8.setFont(u8x8_font_chroma48medium8_r);
      u8x8.setCursor(0, 3);
      u8x8.print("Identity comfirm");
      delay(800);
      digitalWrite(Electromagnet, LOW);  // turn the Ele
      delay(5000);
      digitalWrite(Electromagnet, HIGH);
    }

    else {
      u8x8.setFont(u8x8_font_chroma48medium8_r);
      u8x8.setCursor(0, 3);
      u8x8.print(" Identity deny ");
      //    myservo.write(0);
      delay(200);

      u8x8.setCursor(0, 3);
      u8x8.print("  Please retry  ");
      delay(1500);
      digitalWrite(Electromagnet, HIGH);  // turn the El

    }
  }
```

Project 5 - Air Quality Sensor Hub - Seeeduino XIAO Expansion Board

Overview

This is an environment detect device to collect PM2.5, PM10, temperature, humidity, CO2 and dust particle via Grove - Laser PM2.5 Sensor, Grove - CO2 & Temperature & Humidity sensor and Grove - dust Sensor respectively.

Component required

- **Seeeduino XIAO** [https://www.seeedstudio.com/Seeeduino-XIAO-Arduino-Microcontroller-SAMD21-Cortex-M0+-p-4426.html]
- **Seeeduino XIAO expansion board** [https://www.seeedstudio.com/Seeeduino-XIAO-Expansion-board-p-4746.html]
- **Seeed Grove - CO2 & Temperature & Humidity Sensor for Arduino (SCD30) - 3-in-1** [https://www.hackster.io/products/buy/80471?s=BAhJIhMzNzE2NzQsUHJvamVjdAY6BkVG%0A]
- **Seeed Grove - Laser PM2.5 Dust Sensor - Arduino Compatible - HM3301** [https://www.hackster.io/products/buy/80472?s=BAhJIhMzNzE2NzQsUHJvamVjdAY6BkVG%0A]
- **Seeed Grove - Dust Sensor（PPD42NS)** [https://www.hackster.io/products/buy/30140?s=BAhJIhMzNzE2NzQsUHJvamVjdAY6BkVG%0A]

Hardware Connection

Please follow the same color line to connect each sensor on the board. Please connect the IR sensor grove cable to D0, Mini Motor Driver grove cable to I2C.

Arduino Instructions

Step 1. Follow the connection picture connect all the sensor on the board.

Step 2. Download the **Aruidno IDE** [https://www.arduino.cc/en/Main/software]

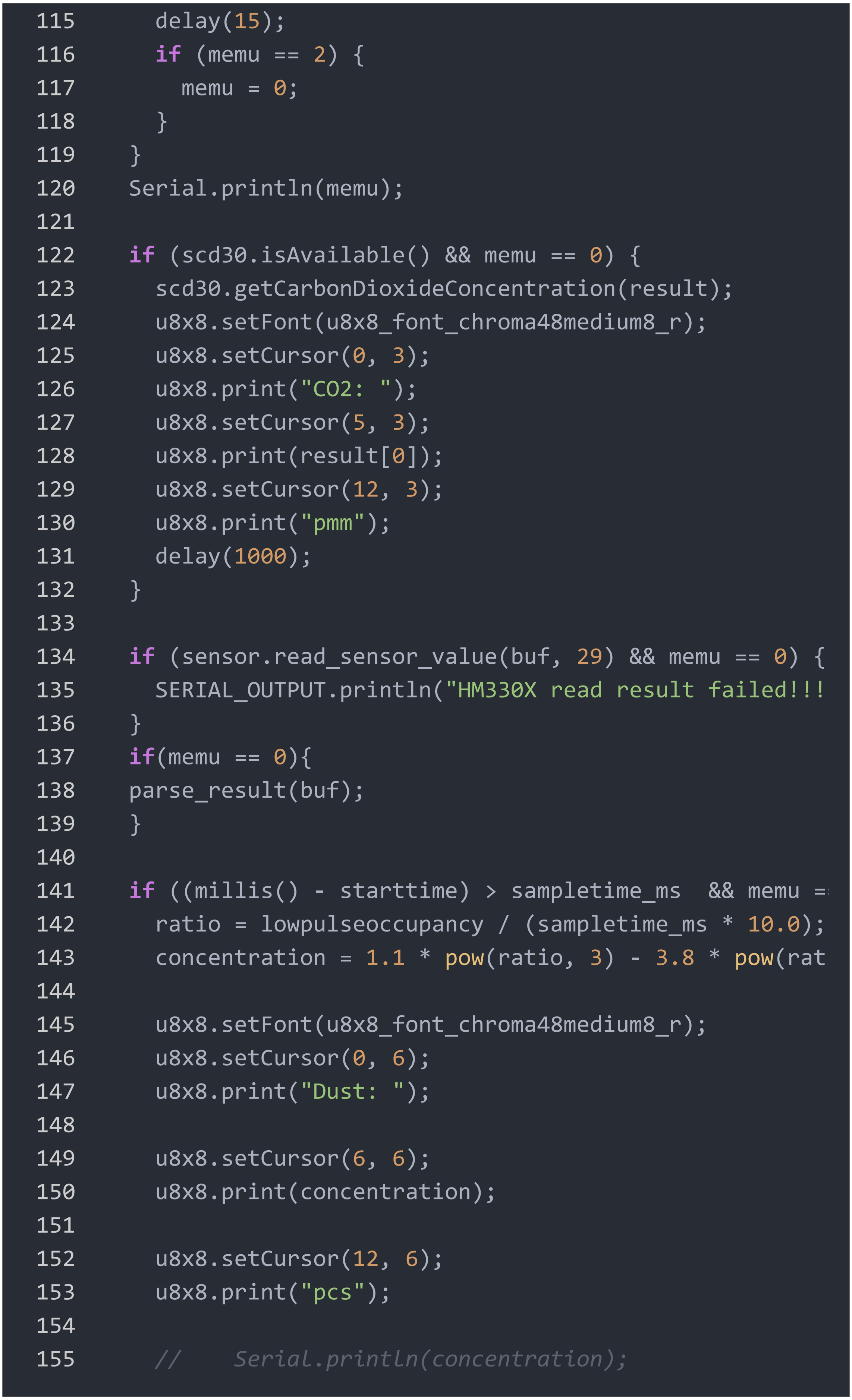
Step 3. Install the **u8g2** [https://github.com/olikraus/U8g2_Arduino], **Seeed_PM2_5_sensor_HM3301** [https://github.com/Seeed-Studio/Seeed_PM2_5_sensor_HM3301] and **Seeed_SCD30** [https://github.com/Seeed-Studio/Seeed_SCD30] library, this is the guide **how to install the library** [https://wiki.seeedstudio.com/How_to_install_Arduino_Library/].

Step 4. Copy the code stick on the Aruino IDE then upload it.

Code

```
#include <Arduino.h>
#include <U8x8lib.h>
#include <Seeed_HM330X.h>
#include "SCD30.h"

#define SERIAL_OUTPUT SerialUSB
#define SERIAL SerialUSB

int pin = 7;
unsigned long duration;
unsigned long starttime;
unsigned long sampletime_ms = 5000;//sampe 30s ;
unsigned long lowpulseoccupancy = 0;
float ratio = 0;
float concentration = 0;

const int buttonPin = 1;
int buttonState = 0;
int memu = 0;

U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(/* reset=*/ U8X8_
HM330X sensor;
uint8_t buf[30];

const char* str[] = {"sensor num: ", "PM1.0 concentrati
                     "PM2.5 concentration(CF=1,Standard
                     "PM10 concentration(CF=1,Standard
                     "PM1.0 concentration(Atmospheric e
                     "PM2.5 concentration(Atmospheric e
                     "PM10 concentration(Atmospheric en
                    };

```

```

///////////////////////////////////////////////////////
//PM2.5 concentration(Atmospheric environment,unit:ug/m
///////////////////////////////////////////////////////
HM330XErrorCode print_result(const char* str, uint16_t 
  if (NULL == str) {
    return ERROR_PARAM;
  }
  //  SERIAL_OUTPUT.print(str);
  u8x8.setFont(u8x8_font_chroma48medium8_r);
  u8x8.setCursor(0, 0);
  u8x8.print("PM2.5: ");
  u8x8.setCursor(7, 0);
  u8x8.print(value);
  u8x8.setCursor(11, 0);
  u8x8.print("ug/m");
  Serial.println(value);
  return NO_ERROR;
}

HM330XErrorCode print_result_1(const char* str, uint16_
  if (NULL == str) {
    return ERROR_PARAM;
  }
  //  SERIAL_OUTPUT.print(str);
  u8x8.setFont(u8x8_font_chroma48medium8_r);
  u8x8.setCursor(0, 0);
  u8x8.print("PM10: ");
  u8x8.setCursor(7, 0);
  u8x8.print(value);
  u8x8.setCursor(11, 0);
  u8x8.print("ug/m");
  Serial.println(value);
  return NO_ERROR;
}

/*parse buf with 29 uint8_t-data*/
HM330XErrorCode parse_result(uint8_t* data) {
  uint16_t value = 0;
  if (NULL == data) {
    return ERROR_PARAM;
  }
  value = (uint16_t) data[6 * 2] << 8 | data[6 * 2 + 1]
  print_result(str[6 - 1], value);
  return NO_ERROR;
}


HM330XErrorCode parse_result2(uint8_t* data) {
  uint16_t value = 0;
  if (NULL == data) {
    return ERROR_PARAM;
  }
  value = (uint16_t) data[7 * 2] << 8 | data[7 * 2 + 1]
  print_result_1(str[7 - 1], value);
  return NO_ERROR;
}

///////////////////////////////////////////////////////

/*30s*/
void setup() {
  Serial.begin(115200);
  Wire.begin();
  u8x8.begin();
  u8x8.setFlipMode(0);
  scd30.initialize();
  pinMode(pin, INPUT);
  pinMode(buttonPin, INPUT_PULLUP);
  starttime = millis();//get the current time;

}

void loop() {
  float result[3] = {0};
  duration = pulseIn(pin, LOW);
  lowpulseoccupancy = lowpulseoccupancy + duration;

  buttonState = digitalRead(buttonPin);

  if (buttonState == LOW) {
    memu++;
```



```
    delay(15);
    if (memu == 2) {
      memu = 0;
    }
  }
  Serial.println(memu);

  if (scd30.isAvailable() && memu == 0) {
    scd30.getCarbonDioxideConcentration(result);
    u8x8.setFont(u8x8_font_chroma48medium8_r);
    u8x8.setCursor(0, 3);
    u8x8.print("CO2: ");
    u8x8.setCursor(5, 3);
    u8x8.print(result[0]);
    u8x8.setCursor(12, 3);
    u8x8.print("pmm");
    delay(1000);
  }

  if (sensor.read_sensor_value(buf, 29) && memu == 0) {
    SERIAL_OUTPUT.println("HM330X read result failed!!!
  }
  if(memu == 0){
  parse_result(buf);
  }

  if ((millis() - starttime) > sampletime_ms  && memu =
    ratio = lowpulseoccupancy / (sampletime_ms * 10.0);
    concentration = 1.1 * pow(ratio, 3) - 3.8 * pow(rat

    u8x8.setFont(u8x8_font_chroma48medium8_r);
    u8x8.setCursor(0, 6);
    u8x8.print("Dust: ");

    u8x8.setCursor(6, 6);
    u8x8.print(concentration);

    u8x8.setCursor(12, 6);
    u8x8.print("pcs");

    //    Serial.println(concentration);
    lowpulseoccupancy = 0;
    starttime = millis();
  }



  if (scd30.isAvailable() && memu == 1) {
    scd30.getCarbonDioxideConcentration(result);
    u8x8.setFont(u8x8_font_chroma48medium8_r);
    u8x8.setCursor(0, 3);
    u8x8.print("Temp: ");
    u8x8.setCursor(6, 3);
    u8x8.print(result[1]);
    u8x8.setCursor(10, 3);
    u8x8.print(" C   ");

    u8x8.setCursor(0, 6);
    u8x8.print("Humi: ");
    u8x8.setCursor(5, 6);
    u8x8.print(result[2]);
    u8x8.setCursor(8, 6);
    u8x8.print("  %     ");

    delay(1000);
  }

  if (sensor.read_sensor_value(buf, 29) && memu == 1) {
    SERIAL_OUTPUT.println("HM330X read result failed!!!
  }
  if(memu == 1){
  parse_result2(buf);
  }
}
```

Project 6 - Seeeduino XIAO Expansion Board - Heart Rate

Overview

This simple and inexpensive project is based on the Seeeduino XIAO expansion board to report the heart rate. The device used has an I2C two-wire interface and therefore keeps the wiring down to a minimum.

Component required

- **Seeeduino XIAO** [https://www.seeedstudio.com/Seeeduino-XIAO-Arduino-Microcontroller-SAMD21-Cortex-M0+-p-4426.html]
- **Seeeduino XIAO expansion board** [https://www.seeedstudio.com/Seeeduino-XIAO-Expansion-board-p-4746.html]
- **Seeed Grove - Finger-clip Heart Rate Sensor** [https://www.hackster.io/products/buy/80359?s=BAhJIhMzNzExNzMsUHJvamVjdAY6BkVG%0A]

Hardware Connection

Please follow the same color line to connect each sensor on the board. Please connect the IR sensor grove cable to D0, Mini Motor Driver grove cable to I2C.

Arduino Instructions

Step 1. Follow the connection picture connect all the sensor on the board.

Step 2. Download the **Aruidno IDE** [https://www.arduino.cc/en/Main/software]

Step 3. Install the **u8g2** [https://github.com/olikraus/U8g2_Arduino] library, this is the guide **how to install the library** [https://wiki.seeedstudio.com/How_to_install_Arduino_Library/].

Step 4. Copy the code stick on the Aruino IDE then upload it.

Code

```
#include <Arduino.h>
#include <U8x8lib.h>

#include <Wire.h>

U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(/* reset=*/ U8X8_

void setup() {
  Serial.begin(9600);
  Serial.println("heart rate sensor:");

  u8x8.begin();
  u8x8.setFlipMode(1);
  Wire.begin();
}
void loop() {
  Wire.requestFrom(0xA0 >> 1, 1);    // request 1 bytes
  while (Wire.available()) {         // slave may send l
    unsigned char c = Wire.read();   // receive heart ra
    u8x8.setFont(u8x8_font_chroma48medium8_r);
//    u8x8.setCursor(0, 3);
//    u8x8.print("blood detecting ");
//    delay(10000);

    u8x8.setCursor(0, 3);
    u8x8.print("HeartRate: ");
    u8x8.setCursor(10, 3);
    u8x8.print(c);
    u8x8.setCursor(13, 3);
    u8x8.print("bpm");
    Serial.println(c);

  }
  delay(500);
}
```

Resources

- **[PDF]**ETA1038 [https://files.seeedstudio.com/wiki/Seeeduino-XIAO-Expansion-Board/document/ETA1038.pdf]
- **[PDF]**ETA3410 [https://files.seeedstudio.com/wiki/Seeeduino-XIAO-Expansion-Board/document/ETA3410.pdf]
- **[PDF]**ETA6003 [https://files.seeedstudio.com/wiki/Seeeduino-XIAO-Expansion-Board/document/ETA6003.pdf]
- **[PDF]**PCF8563T [https://files.seeedstudio.com/wiki/Seeeduino-XIAO-Expansion-Board/document/PCF8563T.pdf]
- **[PDF]**Seeeduino XIAO Expansion board_v1.0_SCH_200824 [https://files.seeedstudio.com/wiki/Seeeduino-XIAO-Expansion-Board/document/Seeeduino%20XIAO%20Expansion%20board_v1.0_SCH_200824.pdf]
- **[SCH]**Seeeduino XIAO Expansion board_v1.0_200824 [https://files.seeedstudio.com/wiki/Seeeduino-XIAO-Expansion-Board/document/Seeeduino%20XIAO%20Expansion%20board_v1.0_200824.sch]
- **[BRD]**Seeeduino XIAO Expansion board_v1.0_200824 [https://files.seeedstudio.com/wiki/Seeeduino-XIAO-Expansion-Board/document/Seeeduino%20XIAO%20Expansion%20board_v1.0_200824.brd]

Tech Support

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