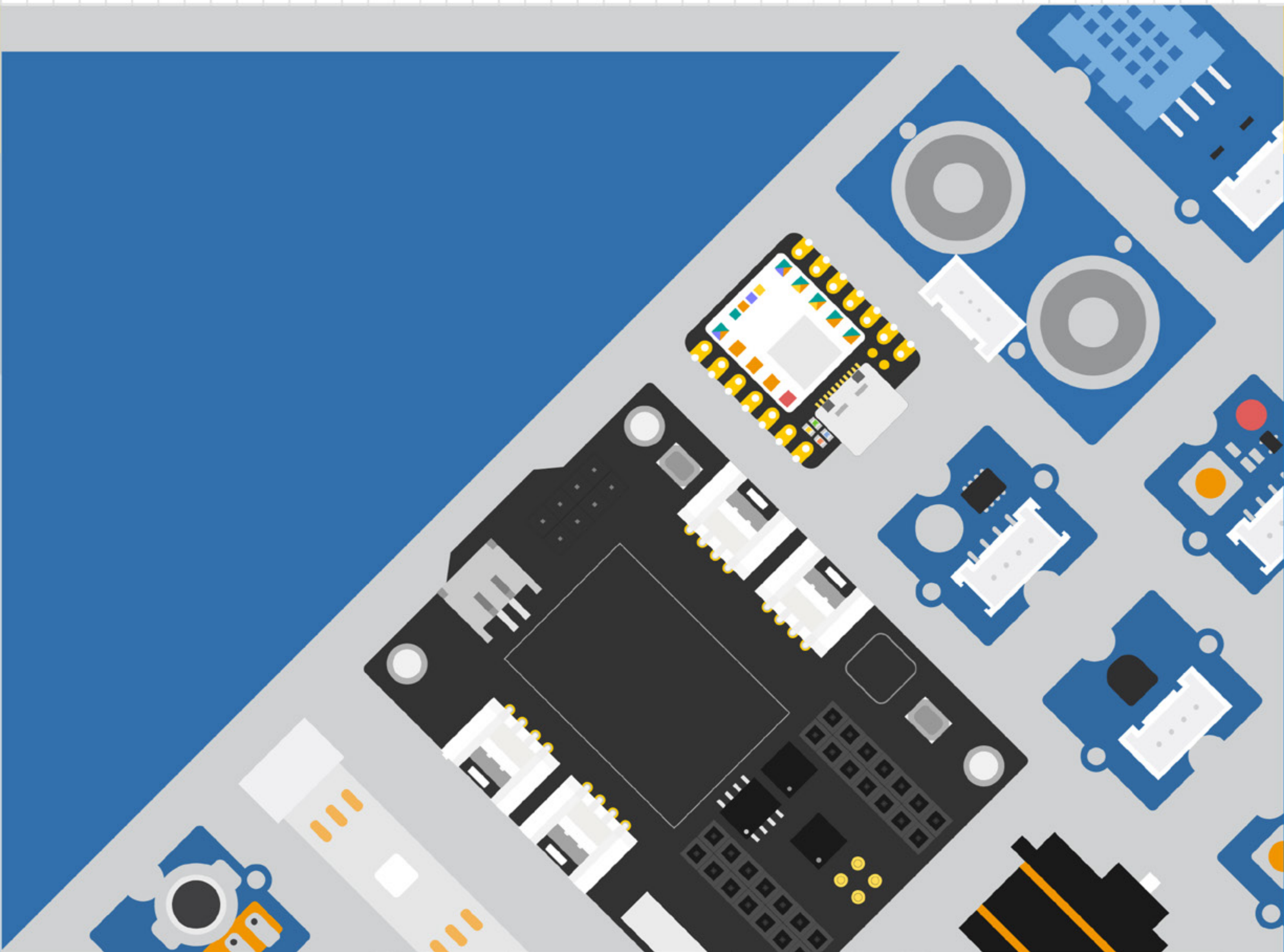
TinkerGen STEM made simple | seeed studio

Seeeduino XIAO in Action

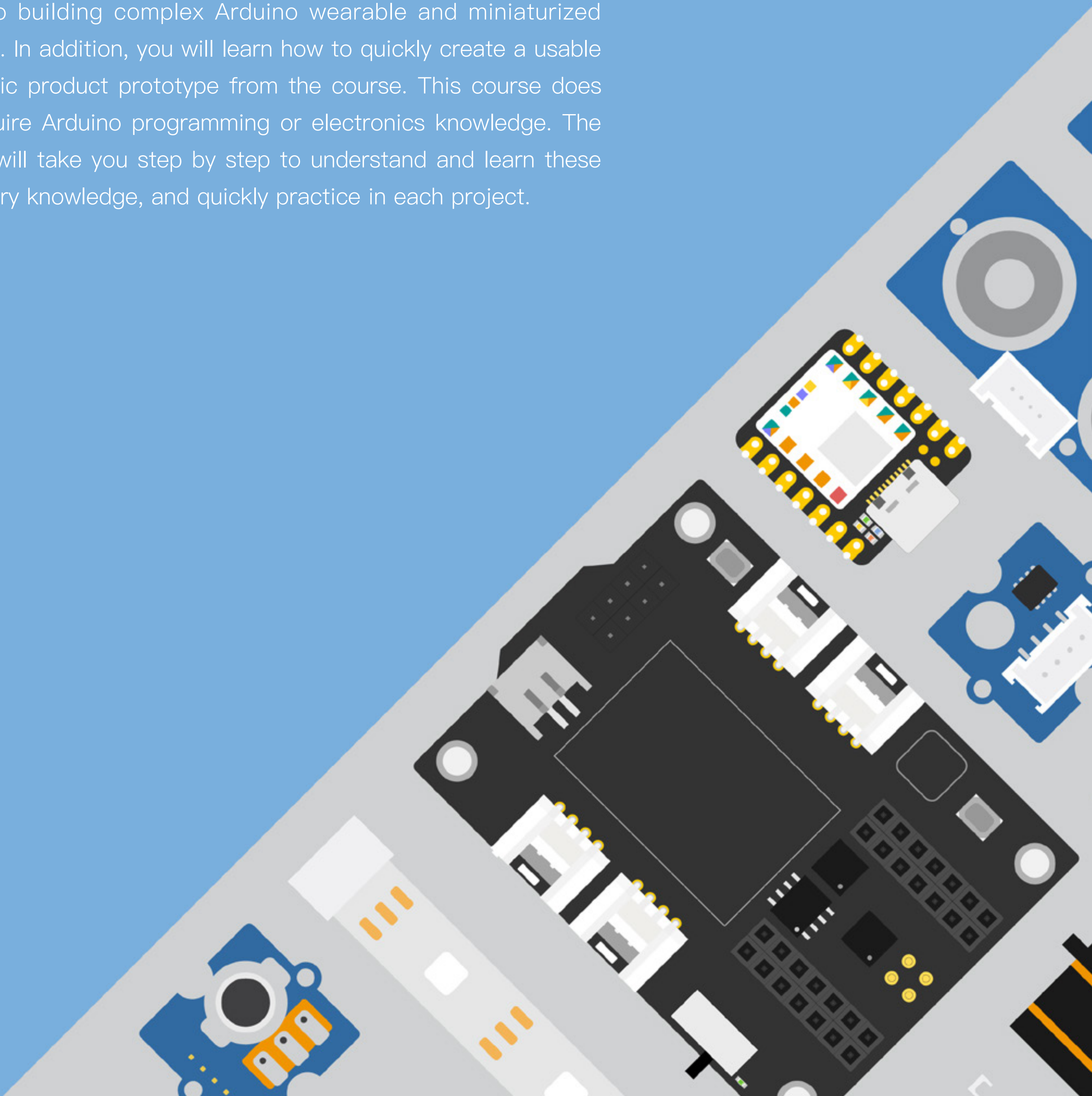
Minitype & Wearable Projects Step by Step



Catalog

Preface

Early Arduino development boards, such as the 8-bit, 16 MHz Arduino Uno, were very large and limited in terms of capacity and performance. Now the Seeeduino XIAO, the size of a thumb cover, offers more powerful performance and a much smaller size, providing more possibilities for Arduino creation. Through hands-on practice and a project-based approach, this course will take you step by step to learn how to use Seeeduino XIAO to learn Arduino from scratch, from simple lighting of LED lights to building complex Arduino wearable and miniaturized projects. In addition, you will learn how to quickly create a usable electronic product prototype from the course. This course does not require Arduino programming or electronics knowledge. The course will take you step by step to understand and learn these necessary knowledge, and quickly practice in each project.

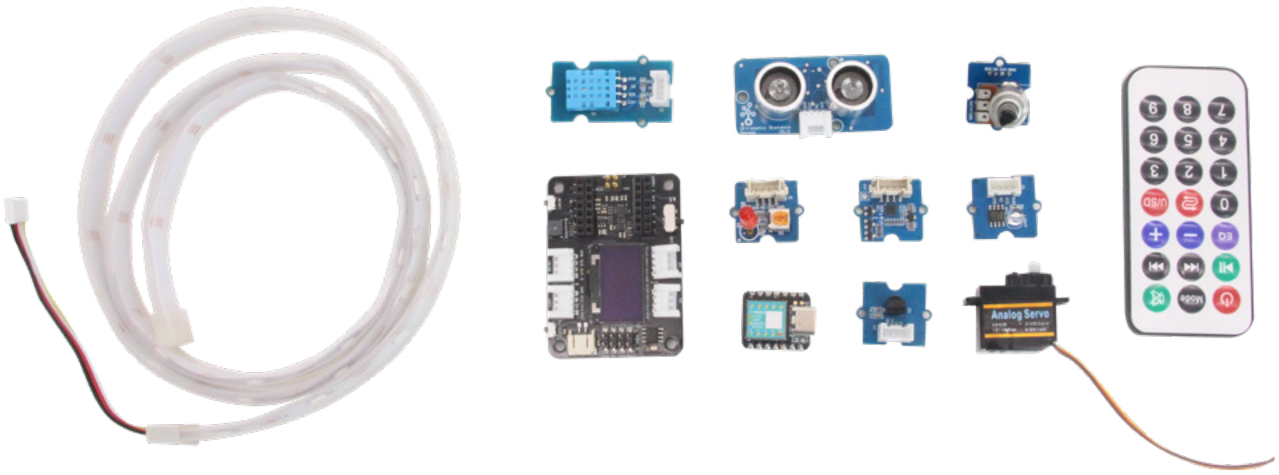


Seeeduino XIAO Starter Kit for Arduino

It's based on Seeeduino Xiao, along with a shield and several sensors/ actuators. With this kit, you can get started with Arduino programming easily. And then you can make your wearable project with everything you just learned.

Part List:

- Seeeduino XIAO ×1
- Seeeduino XIAO expansion board ×1
- Grove – LED Pack ×1
- Grove – Light Sensor ×1
- Grove – Servo ×1
- Grove – Temperature & Humidity Sensor ×1
- Grove – Rotary Angle Sensor ×1
- Grove – RGB LED Strip ×1
- Grove – IR (Infrared) Receiver ×1
- Grove – Ultrasonic Distance Sensor ×1
- Grove – 3–Axis Digital Accelerometer ×1
- Cables ×6



Arduino IDE

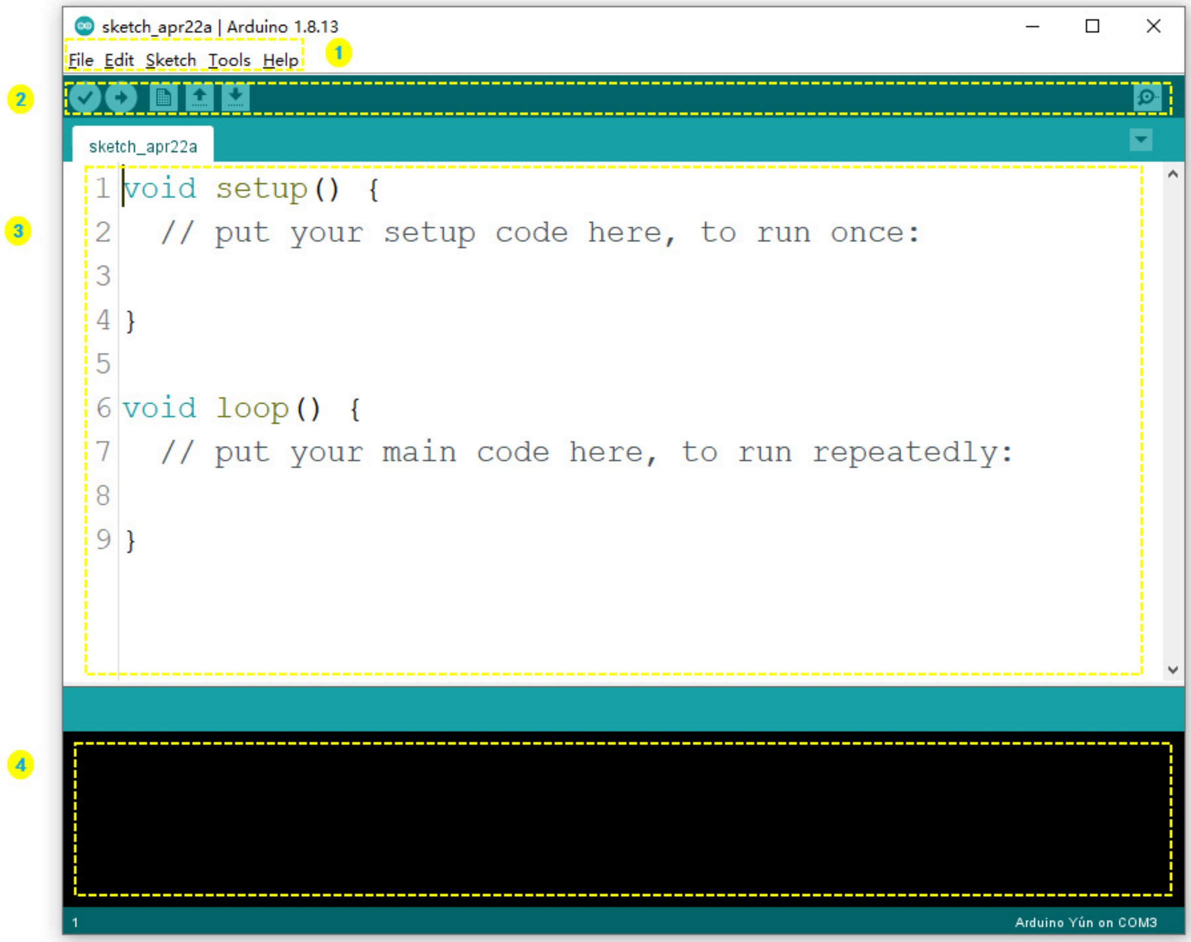
We need to program the hardware through the Arduino IDE text editor. To put it simply, Arduino IDE (Integrated Development Environment) is a programming software specially designed for Arduino, through which we can write and upload different programs for Arduino hardware. When we open the Arduino IDE software, a new file named sketch will be created, of course we can rename it. As shown in the figure below, the interface of Arduino IDE is very simple and can be divided into four parts: menu bar, tool bar, editing area, and debugging window.

1 The menu bar contains files, edits, projects, tools and help, such as creating, saving, sample programs, selecting serial ports, etc.;

2 Toolbar, contains several commonly used function buttons, compile, upload, create new program, open program, save program and serial monitor;

3 The editing area is the area where the program code is written. Just like we usually type on the word page, we write the program code in this area;

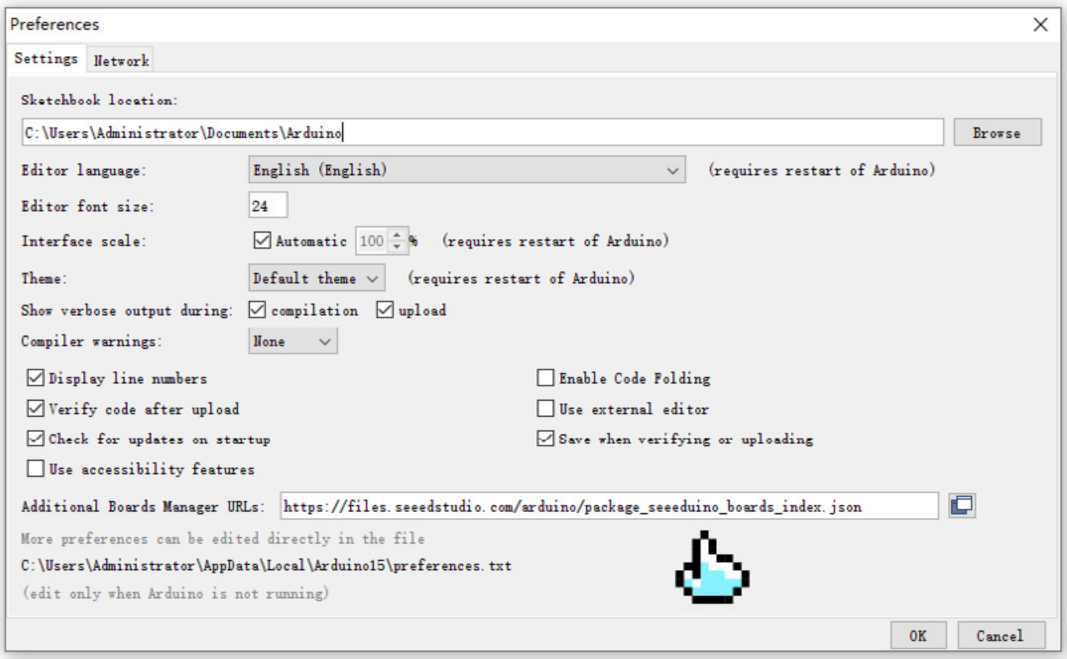
4 Debugging window, displaying program compilation and uploading information, if the program reports an error, it will also be displayed in this area, and we can modify it according to the content of the error.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Add Seeeduino to Your Arduino IDE

Click on **File >→ Preference**, and fill Additional Boards Manager URLs with the url below: https://files.seeedstudio.com/arduino/package_seeeduino_boards_index.json


Preferences

Settings | Network

Sketchbook location:
C:\Users\Administrator\Documents\Arduino
Browse

Editor language: English (English) (requires restart of Arduino)
Editor font size: 24
Interface scale: ☑ Automatic 100 % (requires restart of Arduino)
Theme: Default theme (requires restart of Arduino)
Show verbose output during: ☑ compilation ☑ upload
Compiler warnings: None

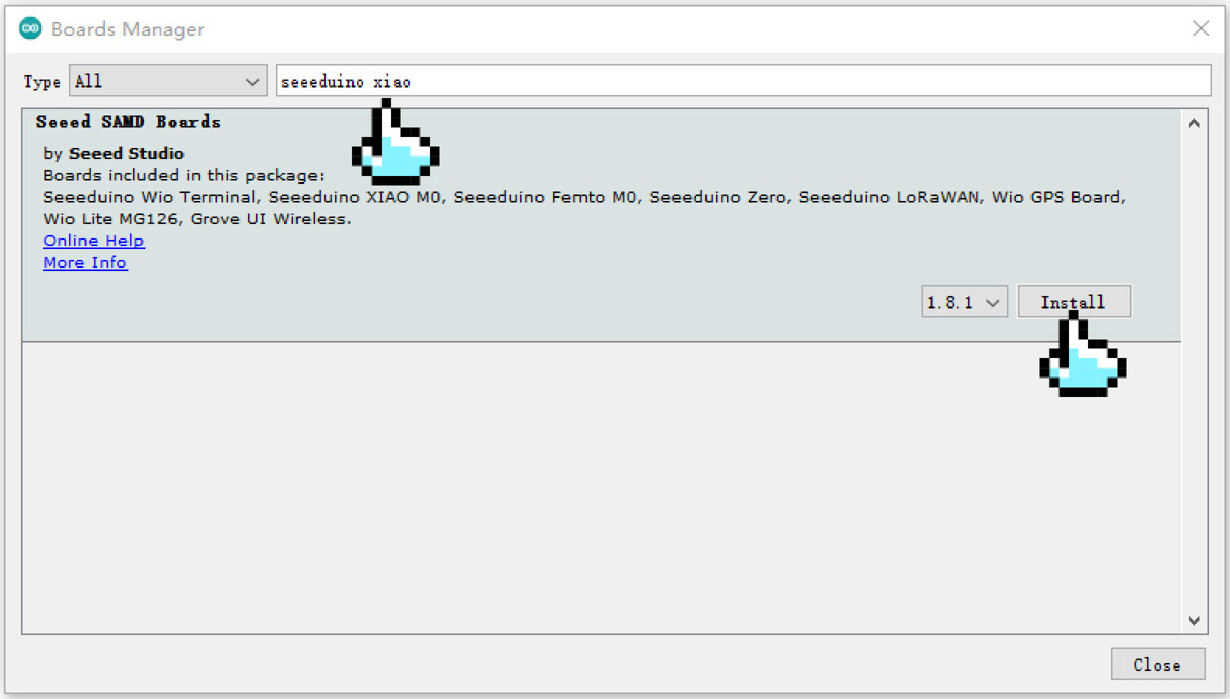
☑ Display line numbers
☑ Verify code after upload
☑ Check for updates on startup
☐ Use accessibility features
☐ Enable Code Folding
☐ Use external editor
☑ Save when verifying or uploading

Additional Boards Manager URLs: https://files.seedstudio.com/arduino/package_seeeduino_boards_index.json

More preferences can be edited directly in the file
C:\Users\Administrator\AppData\Local\Arduino15\preferences.txt
(edit only when Arduino is not running)

OK Cancel

Click **Tools → Board → Boards Manager**..., print keyword "**Seeeduino XIAO**" in the searching blank. Here comes the "**Seeed SAMD Boards**". Install it.



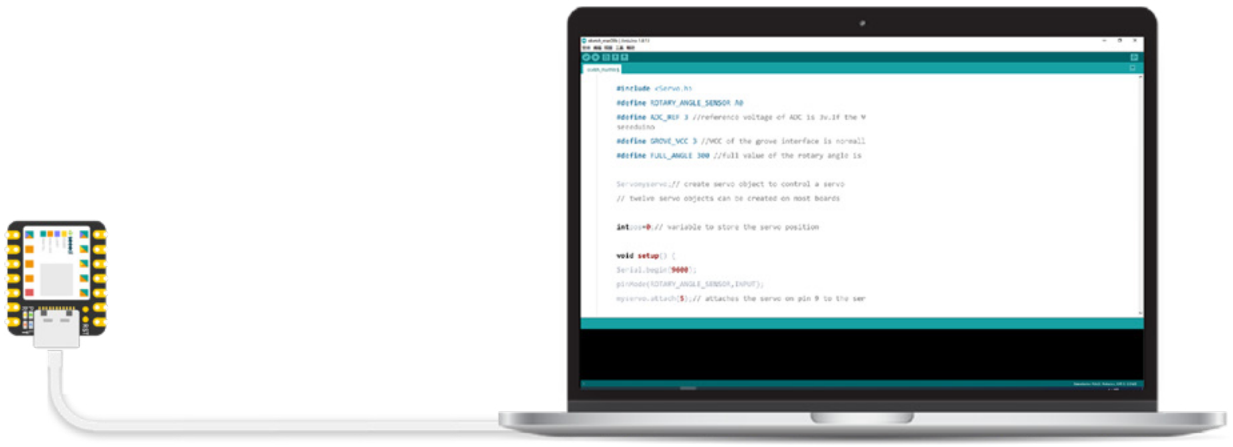
Boards Manager

Type All seeeduino xiao

Seeed SAMD Boards
by **Seeed Studio**
Boards included in this package:
Seeeduino Wio Terminal, Seeeduino XIAO M0, Seeeduino Femto M0, Seeeduino Zero, Seeeduino LoRaWAN, Wio GPS Board, Wio Lite MG126, Grove UI Wireless.
Online Help
More Info

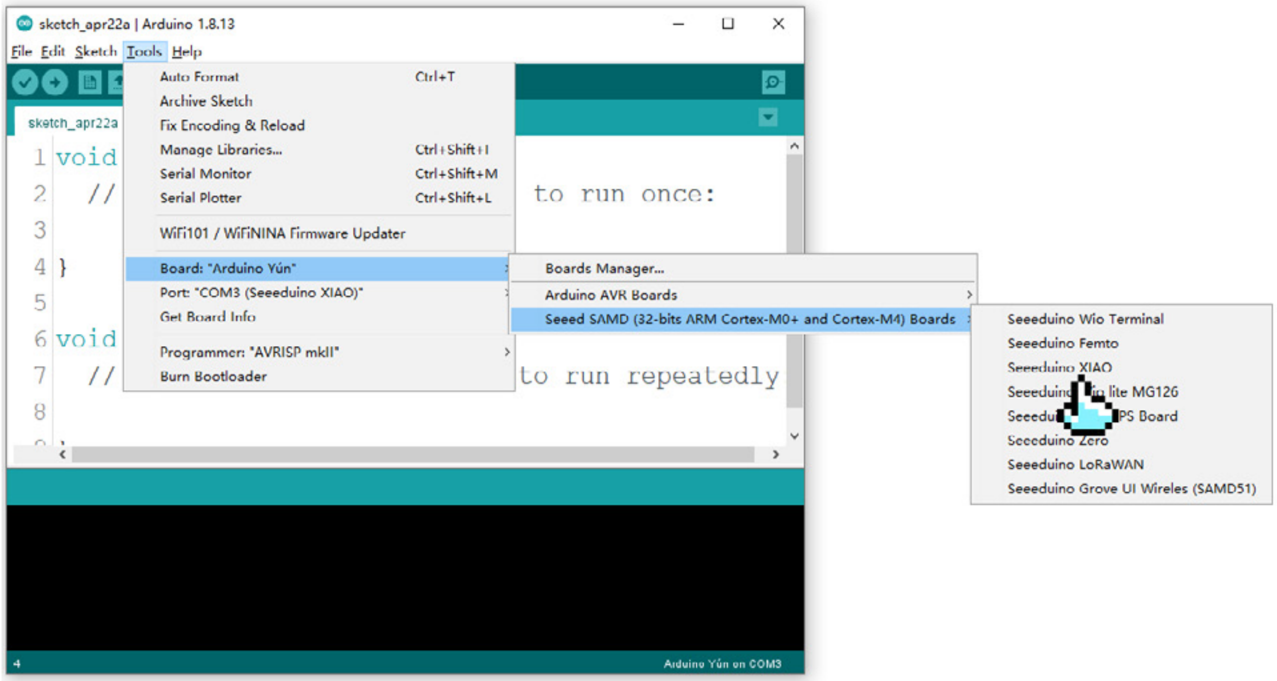
Connect Seeeduino XIAO to Arduino IDE

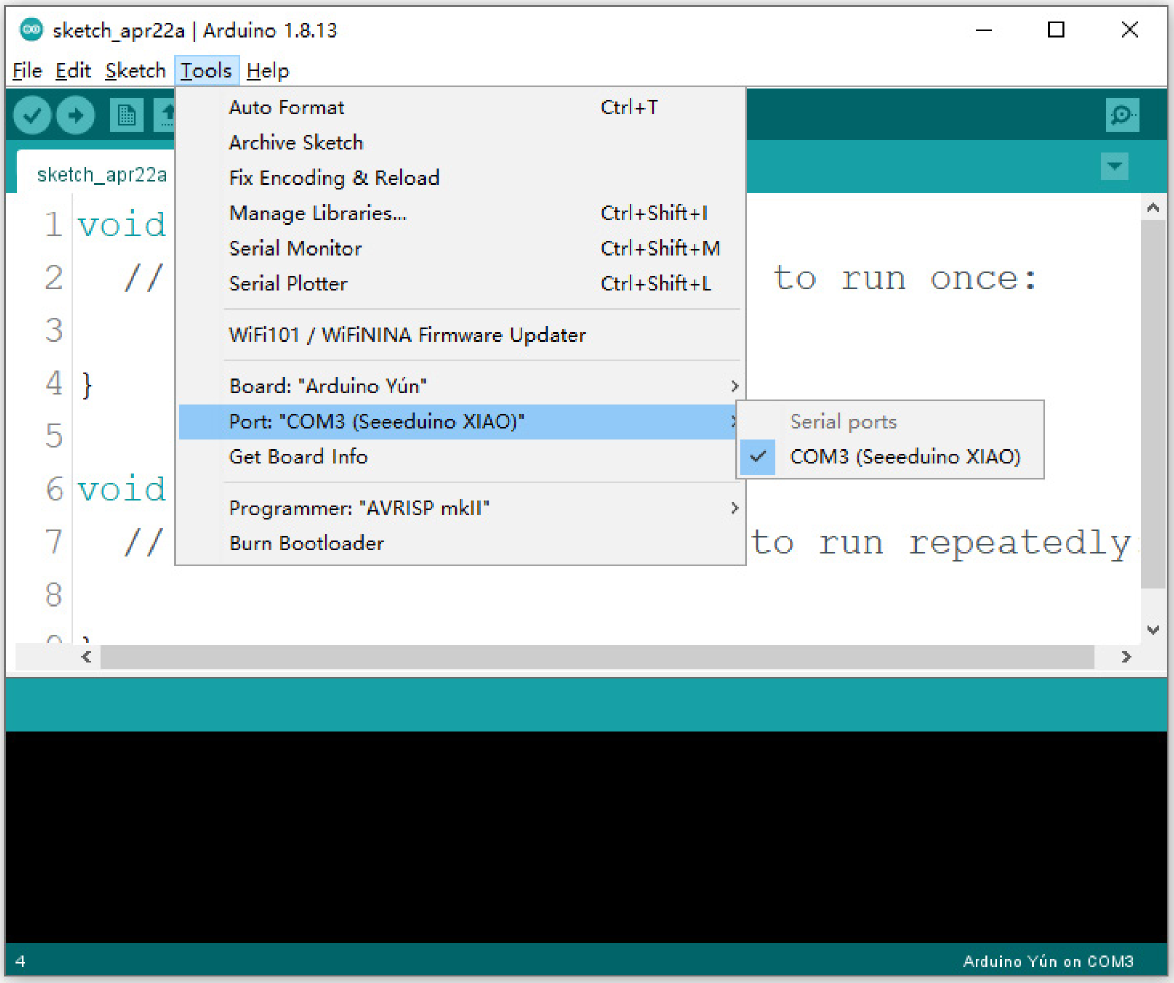
Connect XIAO to the computer with a data cable, as shown in the figure below:



After installing the board, click **Tools → Board**, find "Seeeduino XIAO " and select it. Now you have already set up the board of Seeeduino XIAO for Arduino IDE.

Select the serial device of the Arduino board from the Tools | Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.

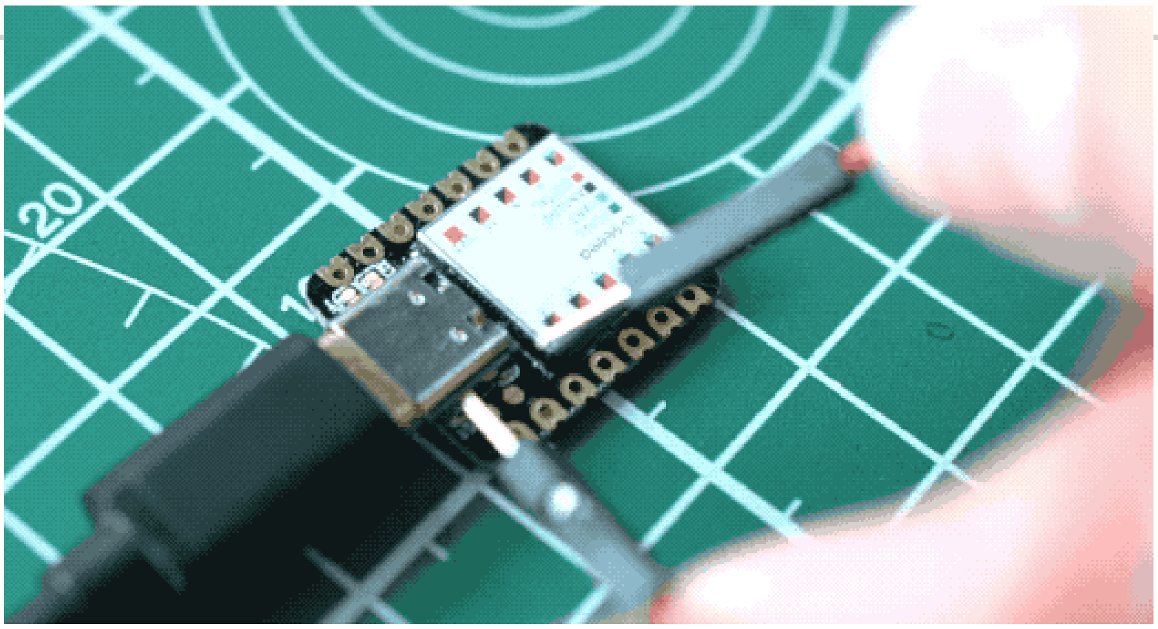

sketch_apr22a | Arduino 1.8.13
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Yún"
Port: "COM3 (Seeeduino XIAO)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Boards Manager...
Arduino AVR Boards
Seeed SAMD (32-bits ARM Cortex-M0+ and Cortex-M4) Boards
Seeeduino Wio Terminal
Seeeduino Femto
Seeeduino XIAO
Seeeduino Cortex lite MG126
Seeeduino GPS Board
Seeeduino Zero
Seeeduino LoRaWAN
Seeeduino Grove UI Wireles (SAMD51)
to run once:
to run repeatedly
Arduino Yún on COM3


sketch_apr22a | Arduino 1.8.13
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Yún"
Port: "COM3 (Seeeduino XIAO)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM3 (Seeeduino XIAO)
to run once:
to run repeatedly
Arduino Yún on COM3

·Note·

Sometimes the Seeeduino XIAO port may disappear when user programming process fails. we can solve this problem by the following operation to reset the Seeeduino XIAO:

- Connect the Seeeduino XIAO to your computer.
- Use tweezers or short lines to short the RST pins only once
- The orange LED lights flicker on and light up.



How to Install an Arduino Library

Here we will show you how to install an Arduino library. You should notice that almost all of our library was stored at Github. We will provide Arduino library when a product need a library. For some simple product, there is no need to write a library such as Grove – Button.

Download the Library

There are two ways to download a Arduino library.

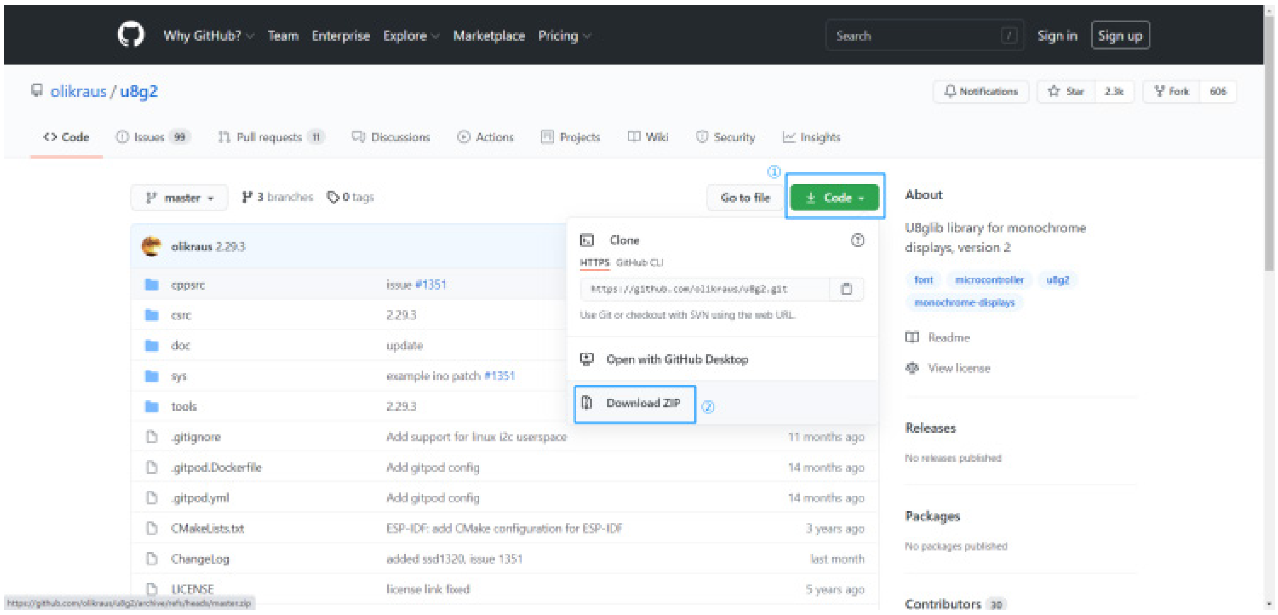
1.Download in the product page

Normally, if a product needs a library, you will find a download library button bar like this below:

Click on the button to start downloading. Seconds you will get a package.

2.Download from github

If you need to download from a Github page, then you can click on the **Clone or download → Download ZIP** button to get the library package.

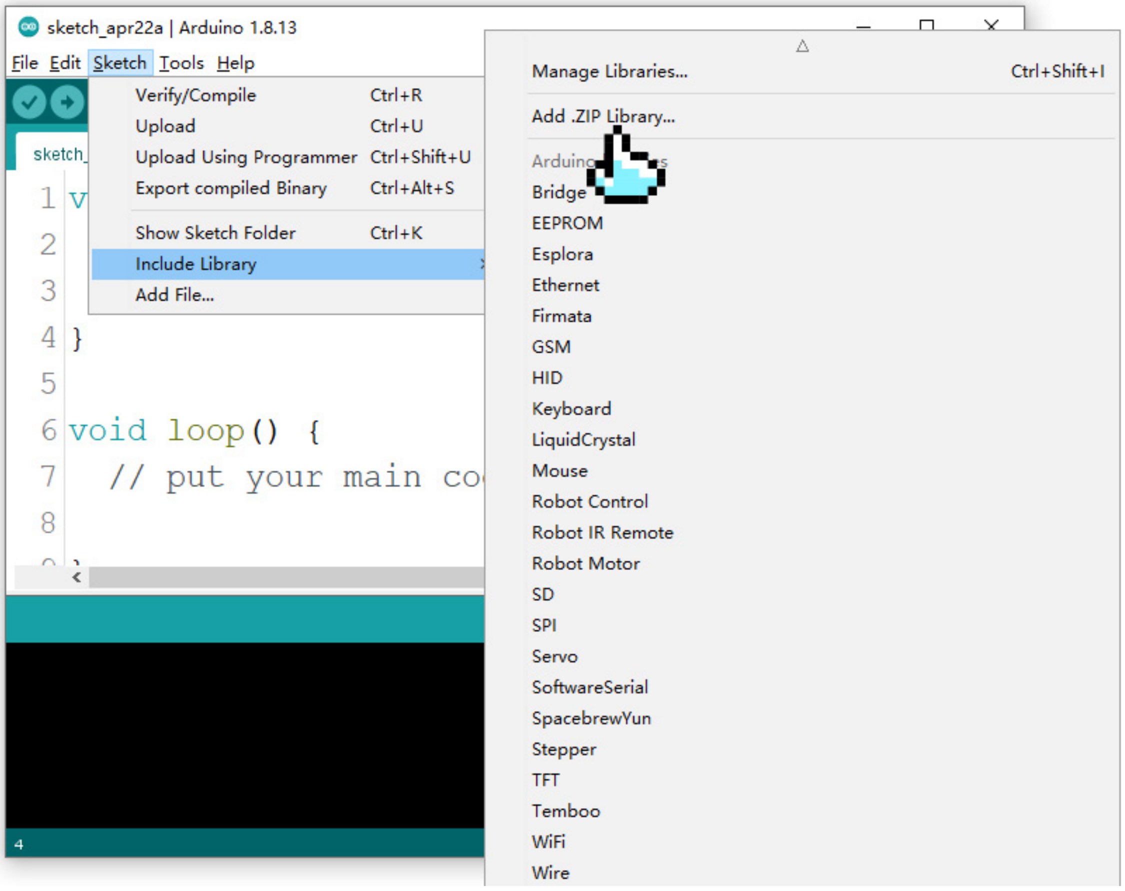


Add the Library

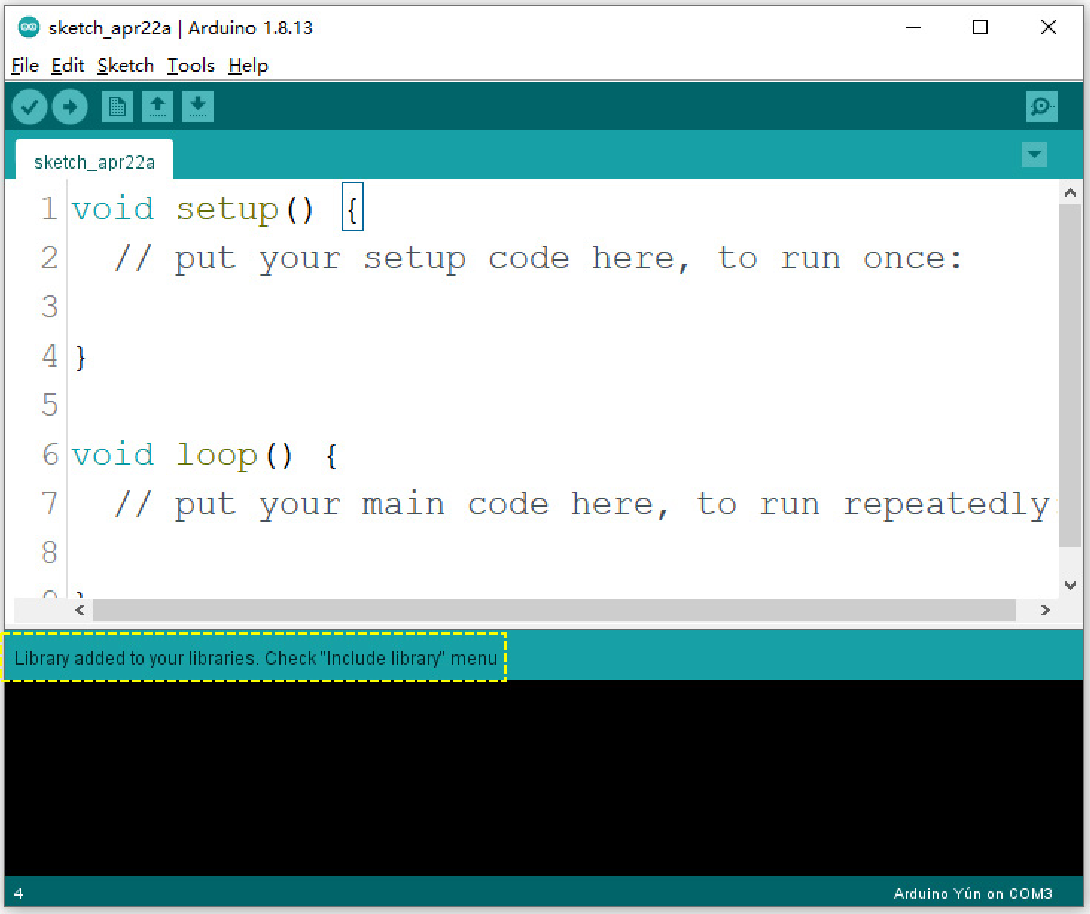
Also there are two ways to add a library to Arduino IDE.

1.Add ZIP Library

Since you have downloaded the zip Library, open your Arduino IDE, click on **Sketch → Include Library → Add .ZIP Library**.



Choose the zip file you just downloaded, and if the library install correct, you will see **Library added to your libraries** in the notice window. Which means the library is installed successfully.

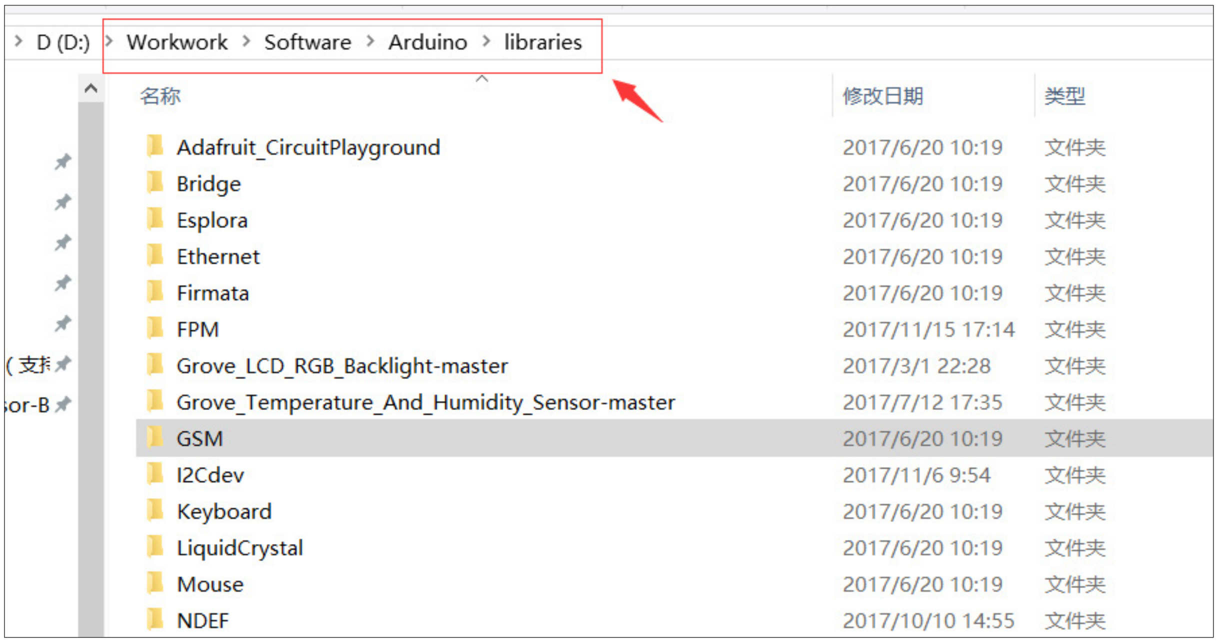


```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly
```

```
Library added to your libraries. Check "Include library" menu
```

2.Add Library folder manually

Sometimes you can not add a ZIP library correctly, because the root folder of the Zip Library lacks of .cpp or .h file, the Arduino can not recognize. Then you can Unzip the zip file and copy the Library folder into the following path **...\Arduino\libraries. ....** is the path you setup Arduino. In my case is **D:\Workwork\Software\Arduino\libraries**.

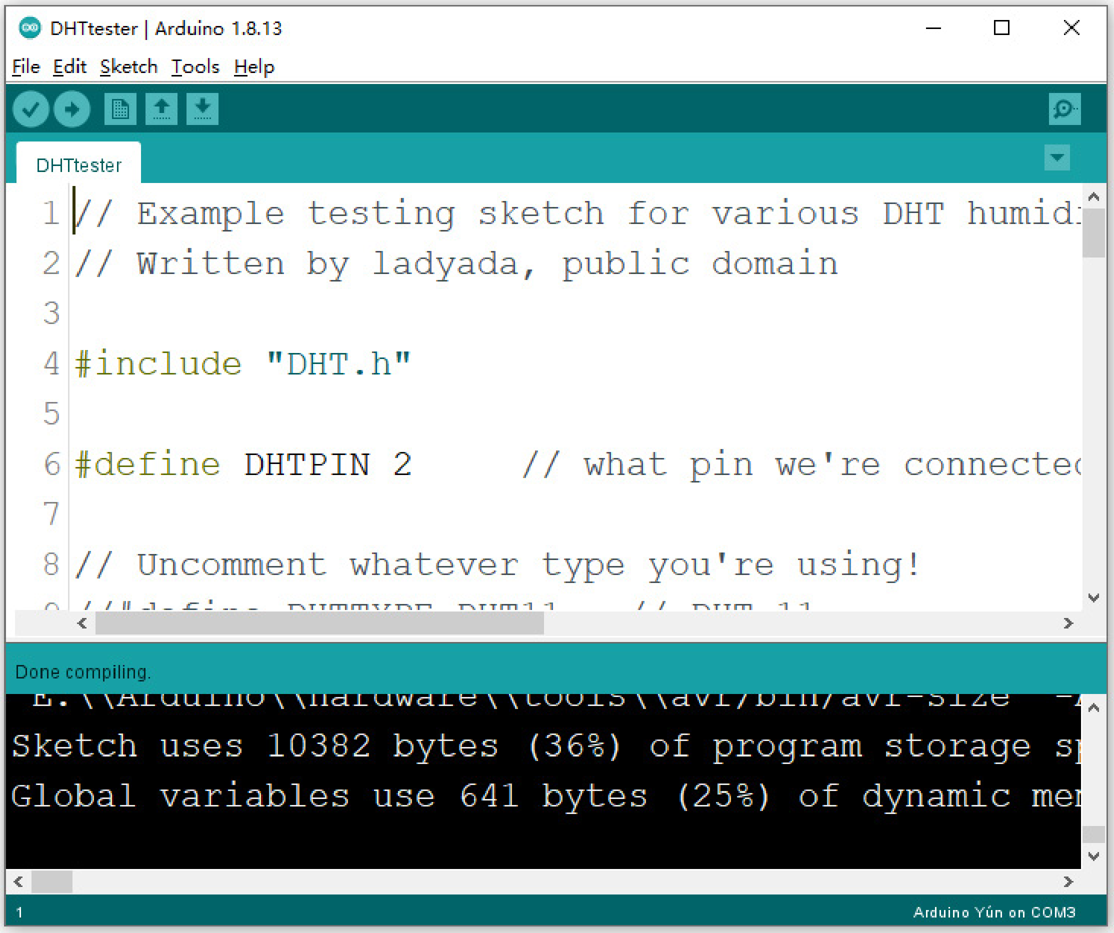


名称	修改日期	类型
Adafruit_CircuitPlayground	2017/6/20 10:19	文件夹
Bridge	2017/6/20 10:19	文件夹
Esplora	2017/6/20 10:19	文件夹
Ethernet	2017/6/20 10:19	文件夹
Firmata	2017/6/20 10:19	文件夹
FPM	2017/11/15 17:14	文件夹
Grove_LCD_RGB_Backlight-master	2017/3/1 22:28	文件夹
Grove_Temperature_And_Humidity_Sensor-master	2017/7/12 17:35	文件夹
GSM	2017/6/20 10:19	文件夹
I2Cdev	2017/11/6 9:54	文件夹
Keyboard	2017/6/20 10:19	文件夹
LiquidCrystal	2017/6/20 10:19	文件夹
Mouse	2017/6/20 10:19	文件夹
NDEF	2017/10/10 14:55	文件夹

3.Check

Then let's check if the library install correctly.

When you add a library successfully, there will be a demo in the Example. In this case, click on **File → Example → OLED_Display_96x96-master → OLED_Hello_World** to open an example, click on the Verify button, if there's no error, congratulation, the library is installed perfectly.



```
// Example testing sketch for various DHT humid
// Written by ladyada, public domain

#include "DHT.h"

#define DHTPIN 2     // what pin we're connecte

// Uncomment whatever type you're using!
```

```
Sketch uses 10382 bytes (36%) of program storage s
Global variables use 641 bytes (25%) of dynamic me
```

Unit 1
Introduction to Hardware and Programming

In this chapter, we will enter the gate of electronic hardware and programming, and initially explore how to control electronic hardware through code. We will learn from the example program blink to learn how to light a lamp, how to control the light on and off through the button, how to control the sound of the passive buzzer, and so on. We will master the common programming languages in each task, such as digital input and output, analog input and output, tone function, map function, and so on, as well learn the basic library usage. The programs learned in this chapter are relatively simple. In the process of learning, we try our best to ensure that the program code of each task is written by ourselves, form a good habit, and avoid the failure of uploading the program due to some symbol errors or unfamiliar rules.

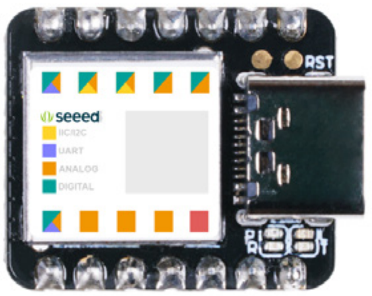
Lesson 1 The first Arduino Program: Blink in Seeeduino XIAO

Arduino is an open source electronic prototype platform popular all over the world, including various types of Arduino development board and Arduino IDE software platform. Because it is flexible, convenient and easy to use, it has become the first choice for many software and hardware beginners. They can quickly complete the project development and realize their own creativity through it. With the development of Arduino, many different types of controllers and many peripheral modules have been produced, such as sensors, actuators, expansion boards and so on.These modules can be used with Arduino to achieve a variety of interesting and useful projects. Seeeduino Xiao, which we are going to learn today, is a development board derived from Arduino. It belongs to the seeeduino series and is the smallest member among them.

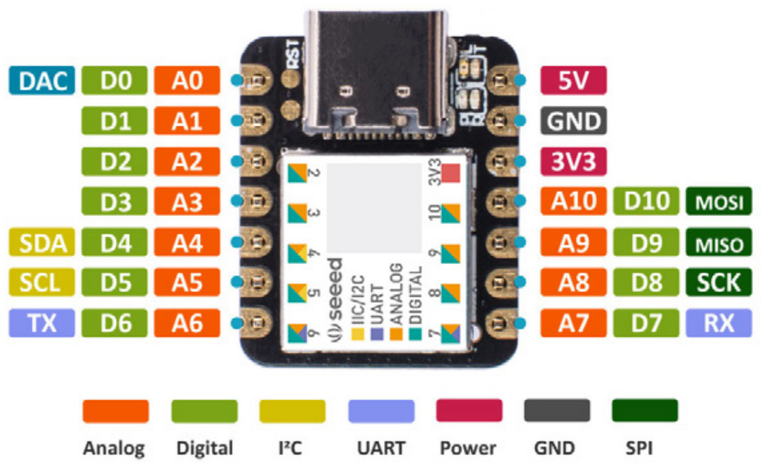
Background Knowledge

Introduction of Seeeduino XIAO

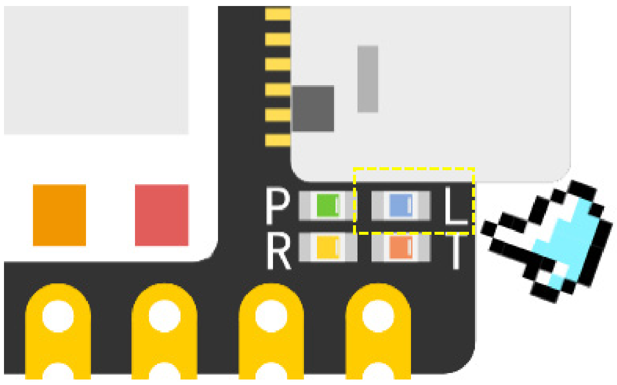
The early Arduino development board, such as the 8-bit, 16 MHz Arduino uno, has a large volume and is limited in capacity and performance. However Seeeduino Xiao is only the size of a thumb cover and provides a 32-bit Arm-Cortex-M0 + processor core, which runs at 48 MHz and has 256 KB flash memory and 64 KB SRAM. In short, it has more powerful performance and smaller size, which provides more possibilities for project creation.



Seeeduino XIAO has 14 GPIO PINs, which can be used for 11 digital interfaces, 11 mock interfaces, 10 PWM interfaces (d1-d10), 1 DAC output pin D0, 1 SWD pad interface, 1 I2C interface, 1 SPI interface, 1 UART interface.We will introduce the functions of these pins one by one in the following study. Moreover, Seeeduino XIAO has a Type-C interface which can supply power and download code. There are two reset pads, you can short connect them to reset the board.



On seeeduino Xiao board, there is a programmable LED light, also called light emitting diode, which can be used for signal indication or lighting. Blink, the first program we will learn today, will use the LED light.



The Structure of Arduino Program

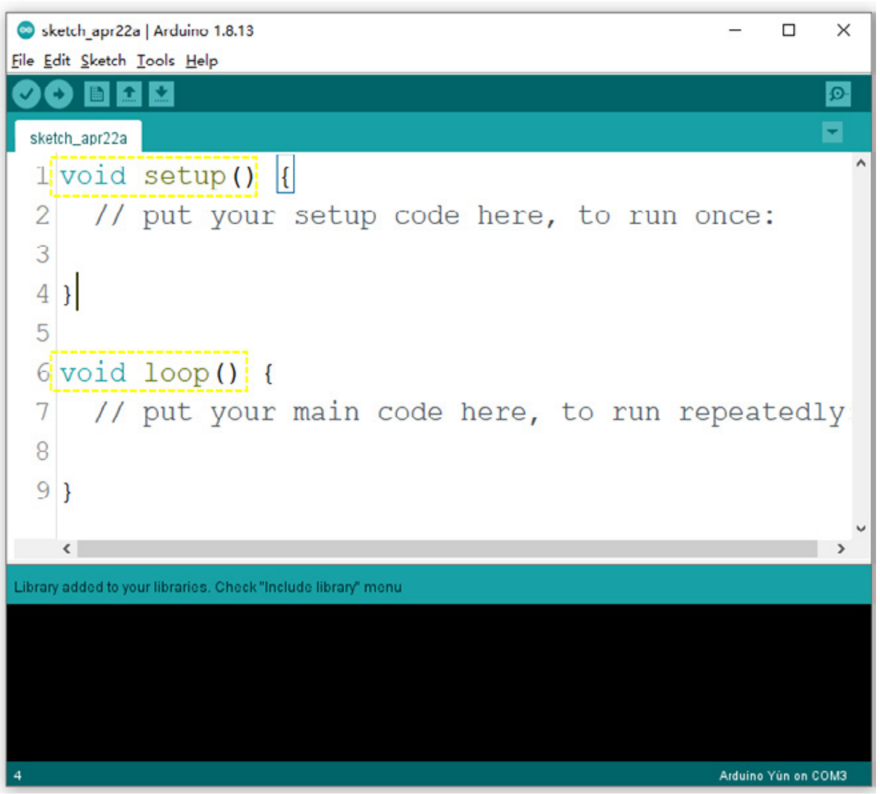
Every Arduino program contains these two functions:

·setup()

The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup() function will only run once, after each powerup or reset of the Arduino board.

·loop()

After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly

}
```

Library added to your libraries. Check "Include library" menu

·Note·

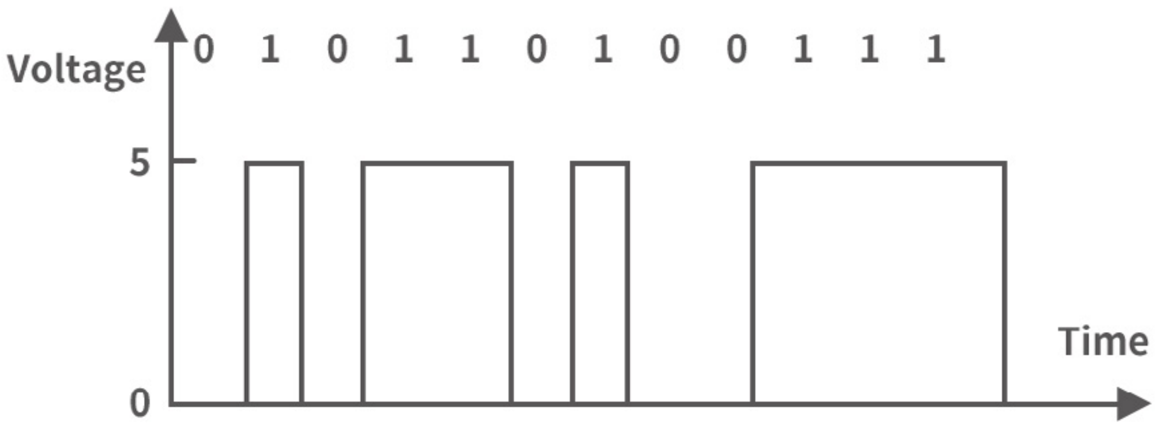
Comments after the “/ * /” ”/ / ”flag help you understand and manage the code. Comments will not affect the normal operation of the program.

When we write a program, we need to package a set of code with “{}”.

After each line of code is written, use “;” as the terminator to tell Arduino editor that this line of code instruction is over.

Digital Signal and I / O Setting

In short, digital signal is the signal in the form of 0, 1 binary.In Arduino, digital signal is represented by high and low value. High value is digital signal 1 and low value is digital signal 0.Seeeduino Xiao has 11 digital pins. We can set these pins to input or output digital signals.



Next, let’s learn how to set the pins.

- **pinMode()**

ppinMode()Configures the specified pin to behave either as an input or an output.

Syntax:

pinMode(pin,mode);

Parameters:

pin:Specified pin number.

mode:Configurable mode. The pins on the Arduino can be configured as either INPUT or OUTPUT, and INPUT_PULLUP.

- **digitalWrite()**

Write a HIGH or a LOW value to a digital pin.

Syntax:

digitalWrite(pin, value);

Parameters:

pin:The Arduino pin number.

value:The output value is HIGH or LOW.In general, high represents high value and low represents low value. Seeeduino Xiao (like Arduino of other models) outputs a low value of 0V, and the output high value is the working voltage of the current board. If it is 5V, the output high value is also 5V.

Example :

digitalWrite(13,HIGH);Sets the digital pin 13 on.

- **digitalRead(pin)**

Reads the value from a specified digital pin, either HIGH or LOW.

Syntax:

digitalRead(pin);

Parameters:

pin:The Arduino pin number you want to read.Digital input generally reads 0 and 1.HIGH or LOW.

Example:

digitalRead(13);

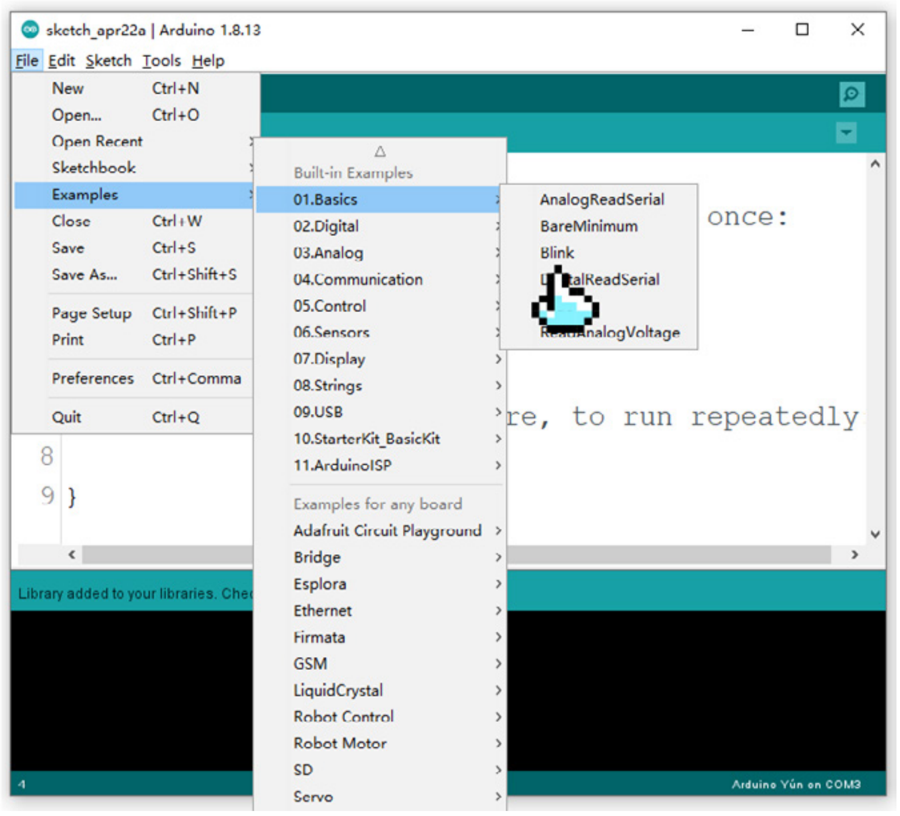
Indicates to read the digital signal of pin 13.

Release Tasks

Task: Upload Sample Program Blink

Just as hello world is the first lesson of all programming languages, blink is equivalent to Hello world in Arduino programming, which is a key to our Arduino learning journey.Arduino provides a lot of sample code to help us get started, and blink is one of them.

We can select “**File** → **Example** → **01. Basics** → **blink**” in Arduino window to open the sample program blink.



After opening the sample program, we can see the following code, which realizes the effect of LED flashing.There are orange codes and green codes are displayed, which proves that your input is correct, pay attention to the difference between case.

```
/*
  Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

Analysis

pinMode(LED_BUILTIN, OUTPUT);

The code firstly initializes LED_BUILTIN as the output pin. Generally, the onboard LED of Arduino series boards is the 13th pin by default.The constant "LED_BUILTIN" is to connect the onboard LED to pin 13.

digitalWrite(LED_BUILTIN, HIGH);

In the loop function, we set the LED_BUILTIN pin to "on" state, and output 5V or 3.3 voltage to this pin, which can be expressed as HIGH.For general I/O pins: Working voltage of MCU is 3.3V. Voltage input connected to general I/O pins may cause chip damage if it's higher than 3.3V.

digitalWrite(LED_BUILTIN, LOW);

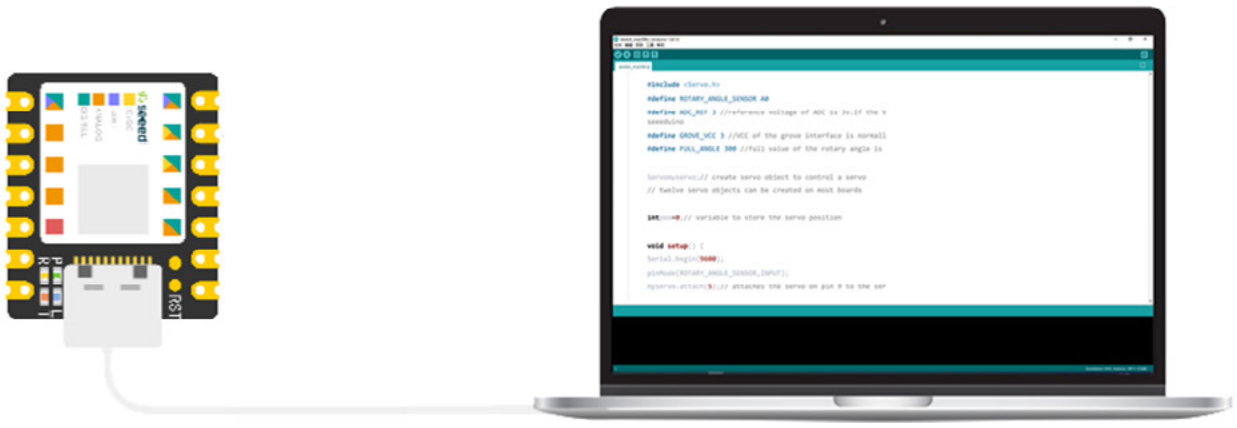
If it is on, this statement sets the LED_BUILTIN pin to "off", and outputs 0V voltage to this pin, which can be expressed by LOW.

delay(1000);

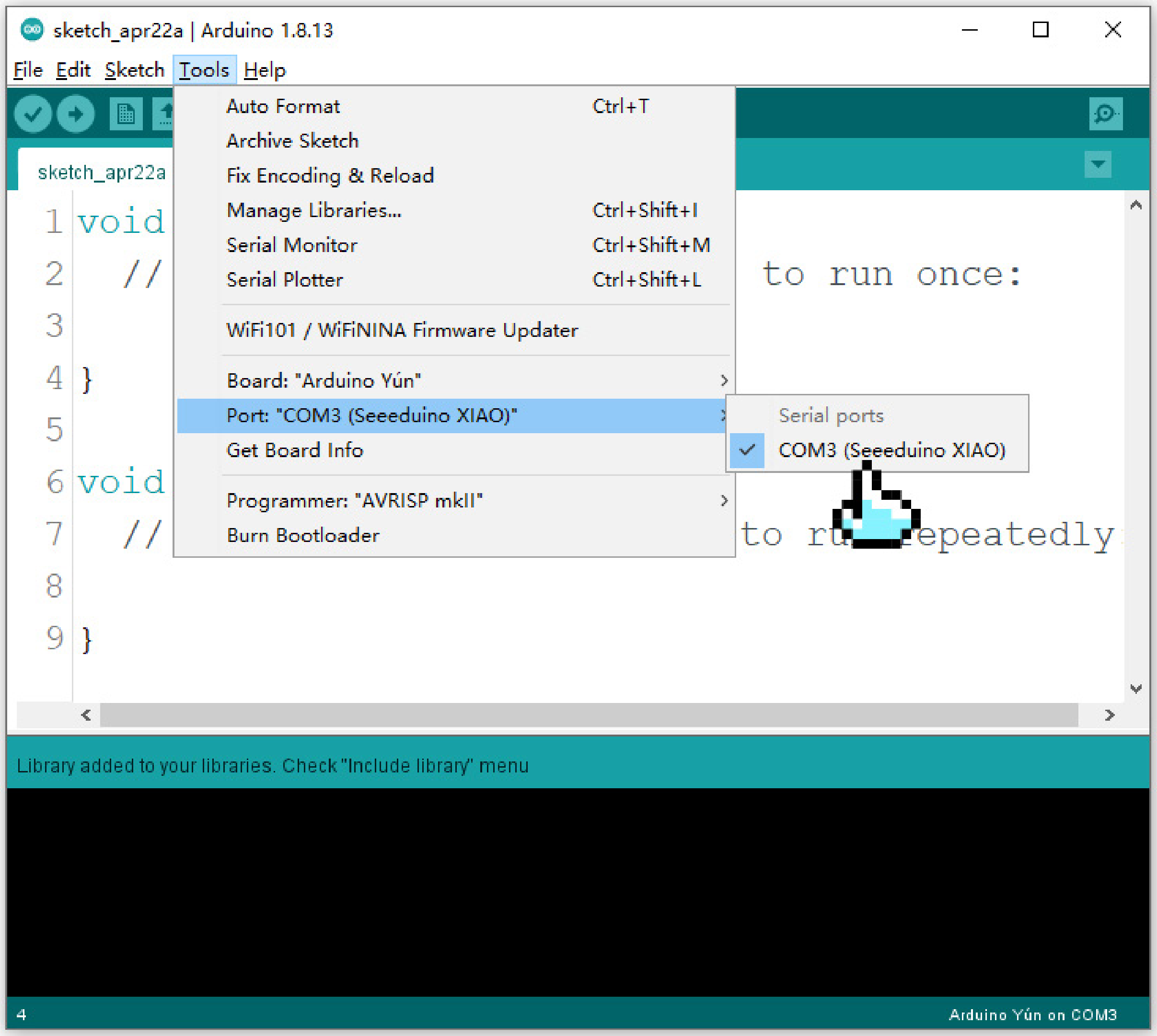
This is a delay statement, which means that the LED lamp can be turned on or off for 1 second. Because the parameter in the function is in milliseconds, 1000 milliseconds is 1 second. After the "on" and "off" statements, you need to add a delay, and the waiting time is the same, in order to ensure that the LED lights flash evenly.

Upload program

Next, we'll learn how to upload the program and connect XIAO to the computer with the data cable in the suite, as shown in the figure.



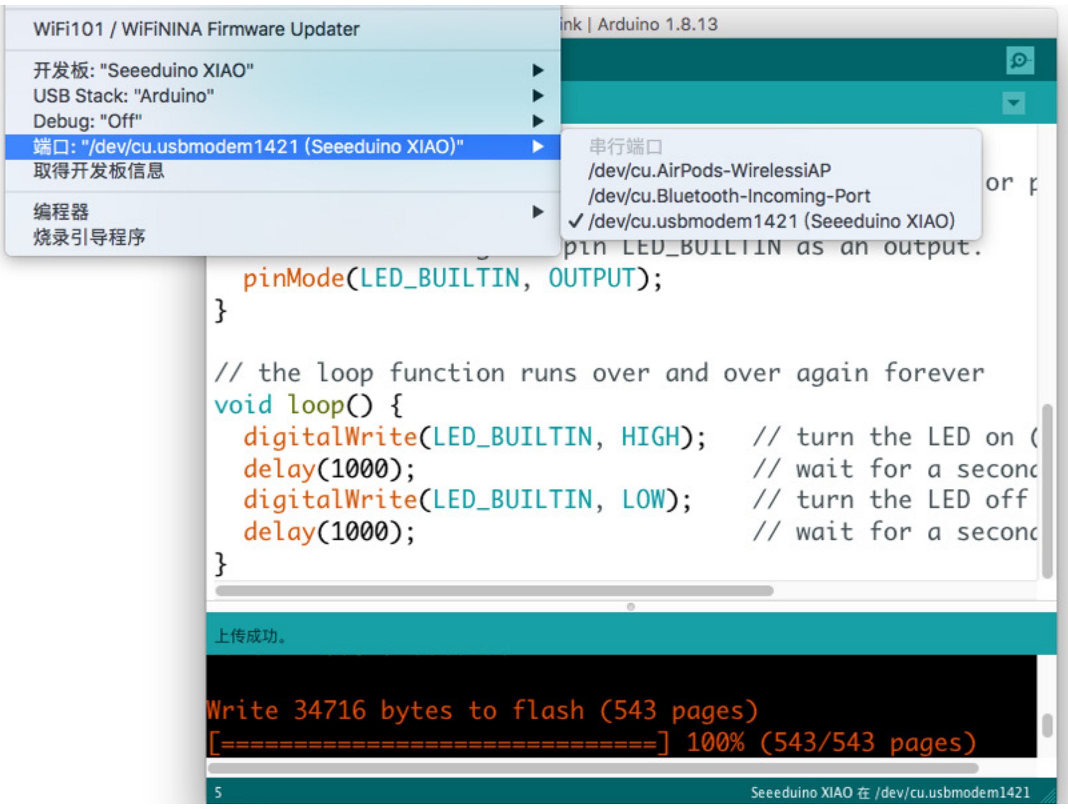
Select the serial port of the development board from the “Tools” column, which is usually COM3 or higher. If there are several port selections, unplug the data cable and reopen the ”Tools” bar, and the missing port is XIAO’s port.Reconnect the circuit board and select the serial port. After that, you can see the currently selected controller model and corresponding serial port in the lower right corner of IDE interface.


sketch_apr22a | Arduino 1.8.13
File Edit Sketch Tools Help
sketch_apr22a
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Yún"
Port: "COM3 (Seeeduino XIAO)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM3 (Seeeduino XIAO)
4
Arduino Yún on COM3

```
void
  //                    to run once:

}

void
  //                    to ru epeatedly

}
```

Library added to your libraries. Check "Include library" menu

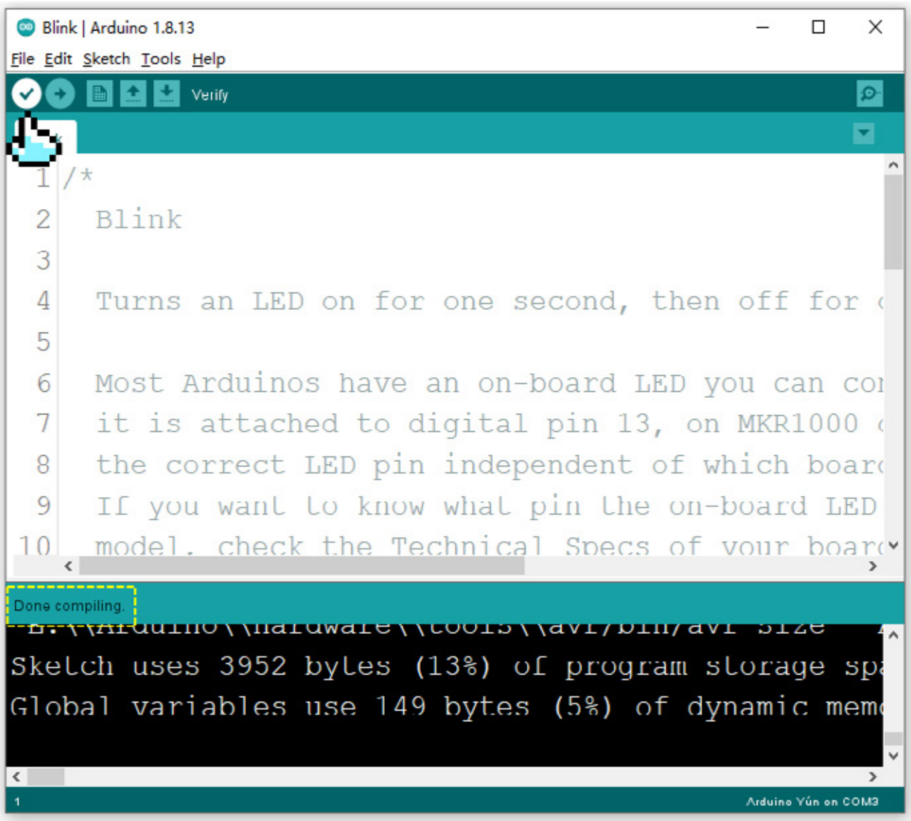
In Mac or Linux system, the serial port name is generally /dev/tty.usbmodem+ number or /dev/cu.usbmodem+ number, as shown in the figure.


WiFi101 / WiFiNINA Firmware Updater
开发板: "Seeeduino XIAO"
USB Stack: "Arduino"
Debug: "Off"
端口: "/dev/cu.usbmodem1421 (Seeeduino XIAO)"
取得开发板信息
编程器
烧录引导程序
串行端口
/dev/cu.AirPods-WirelessiAP
/dev/cu.Bluetooth-Incoming-Port
✓ /dev/cu.usbmodem1421 (Seeeduino XIAO)
ink | Arduino 1.8.13
上传成功。
Seeeduino XIAO 在 /dev/cu.usbmodem1421

```
                                      or p
                        pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (
  delay(1000);                       // wait for a seconc
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off
  delay(1000);                       // wait for a seconc
}
```

```
Write 34716 bytes to flash (543 pages)
[==============================] 100% (543/543 pages)
```

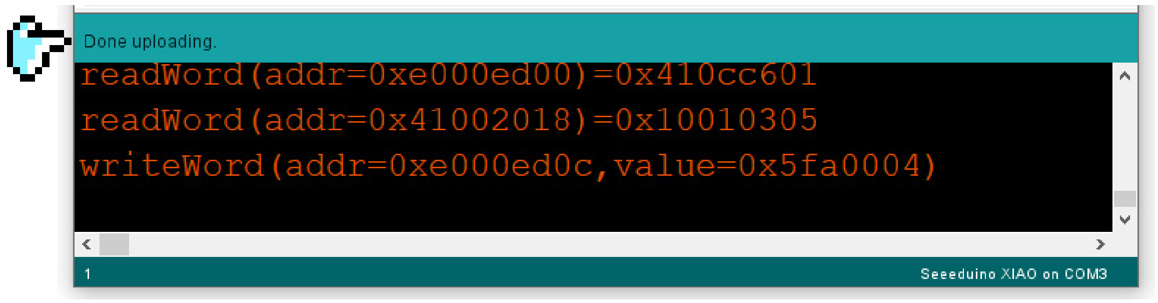
Next, we can upload the program. Before uploading, we can click the “Verify” button to verify whether the program is correct or not. If “Done compiling” is displayed, the program is correct.



```
/*
  Blink

  Turns an LED on for one second, then off for c

  Most Arduinos have an on-board LED you can cor
  it is attached to digital pin 13, on MKR1000 c
  the correct LED pin independent of which boarc
  If you want to know what pin the on-board LED
  model, check the Technical Specs of your boarc
```

```
E:\\Arduino\\hardware\\tools\\avr/bin/avr-size
Sketch uses 3952 bytes (13%) of program storage sp
Global variables use 149 bytes (5%) of dynamic memc
```

Click the “Upload” button, and the debugging window will display “Project being compiled → Upload”. When “Done uploading” is displayed, the program will begin to run on XIAO.

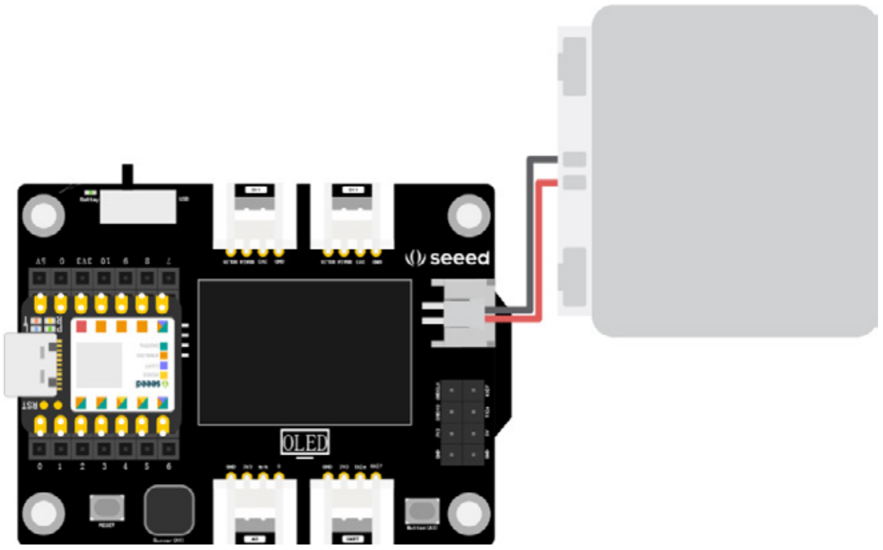


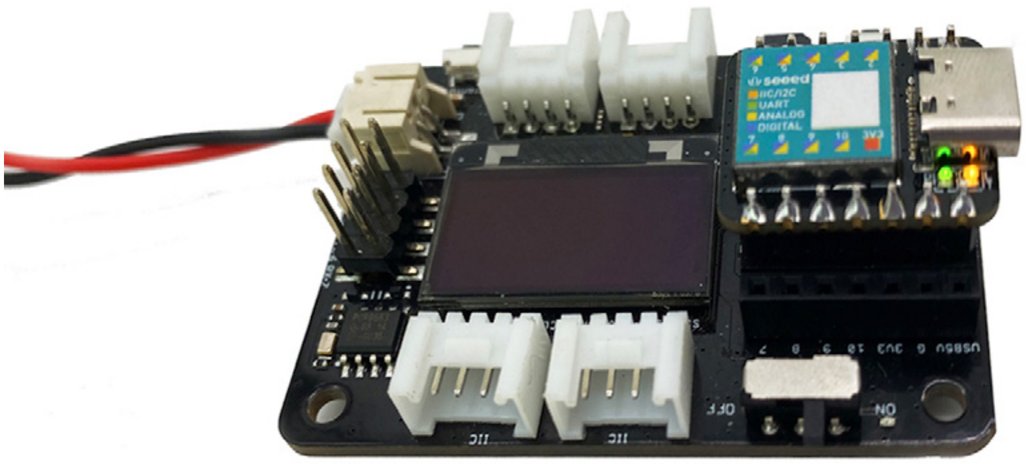
Done uploading.

```
readWord(addr=0xe000ed00)=0x410cc601
readWord(addr=0x41002018)=0x10010305
writeWord(addr=0xe000ed0c,value=0x5fa0004)
```

1 Seeeduino XIAO on COM3

In addition to using the data cable to connect to the computer for power supply, it can also be connected to an external lithium battery, which is convenient when creating mobile projects.





·Note·

When you start to write code, you will often forget the case, punctuation rules and other errors, so try to write your own code instead of using copy and paste. After the sample program is uploaded successfully, try to create a new sketch with your own self-written code.

⭐ Intensive Learning

Blink Program Rewrite: In the example program, the LED lights flash on and off at a regular interval of one second.Try to adjust the waiting time to let the LED lights have a different flashing effect.

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(500);
```

Please see details on entire program:

L1-Blinks.ino

Lesson 2 Button Switch LED Light

Last lesson, we learned how to control the flashing of LED lights by code, which can be realized with just the Seeeduino XIAO and onboard LED lights, but the whole process can not form interaction with the external environment, such as controlling the turning on and off of LED lights by light and sound.In this lesson, we will add a simple button switch to form an automatic control system of sensor controller and actuator.

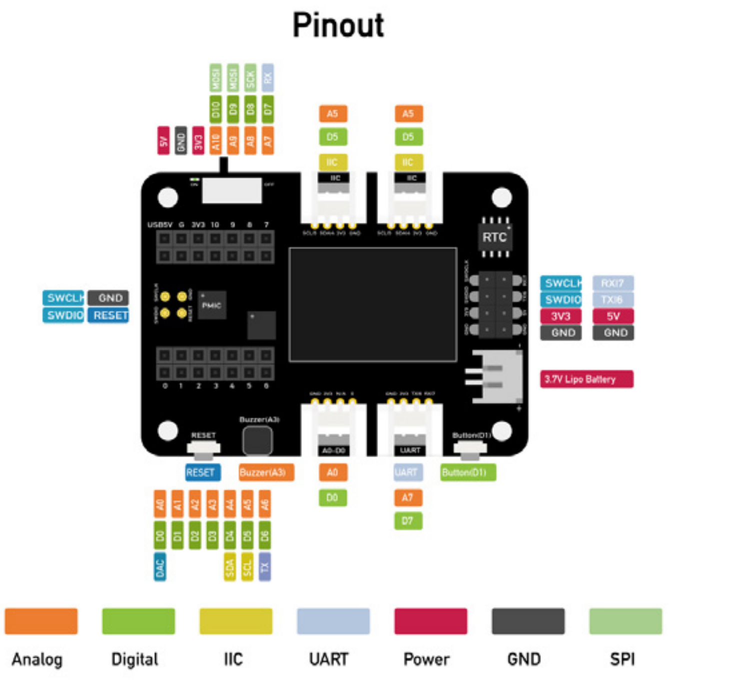
Before starting the task, we should learn some basic knowledge, such as what variables are and the commonly used program structures. Only by understanding these can we better understand and run the program.

Background Knowledge

In the last lesson, we used only the LED lights onboard Seeeduino XIAO, and did not connect other modules. To interface with an external sensor, this thumb sized board will need to use jumper wires or a breadboard, which may be a lot of effort for the novice. Is there a simpler way?

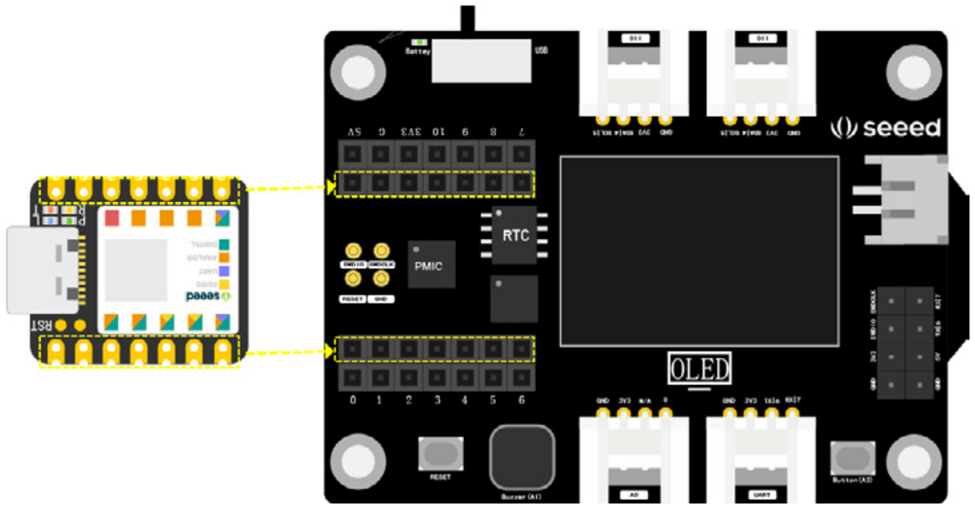
Seeeduino XIAO Extension Board

In order to make it easier and faster through Seeeduino XIAO to build the project, we are equipped with a powerful expansion board with rich peripherals on board, and can quickly connect more electronic modules to realize various functions. The expansion board leads out all the pins of XIAO, and the pin diagram is as follows:



GPIO	0	1	2	3	4	5	6	7	8	9	10	SWCLK	SWDIO	RESET
Interface	A0、D0	D1	SPI	A3	IIC		UART		SPI			SWD		RESET
Components	Grove*1	User Button	mini SD	Buzzer	Grove*2		Grove*1、Headers		mini SD			Headers		Button

When we want to use XIAO's expansion board, we need to connect XIAO's main board to the corresponding position of the expansion board, as shown in the following figure.Connect the pin header on the XIAO motherboard to the position circled by the yellow box on the expansion board. Be careful to align,then press them down to avoid damaging the pins.

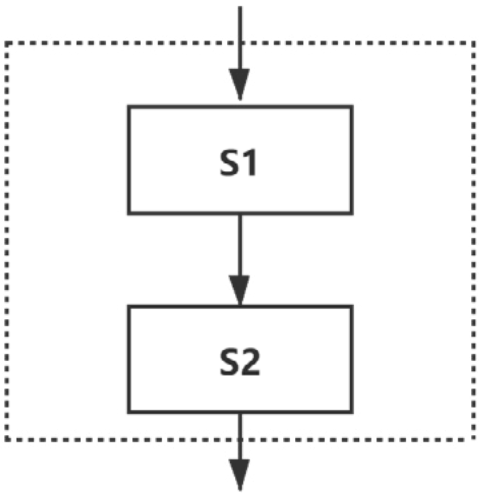


Three Basic Structures of Program

The three basic structures of the program are the sequential structure, the selective structure and the loop structure.

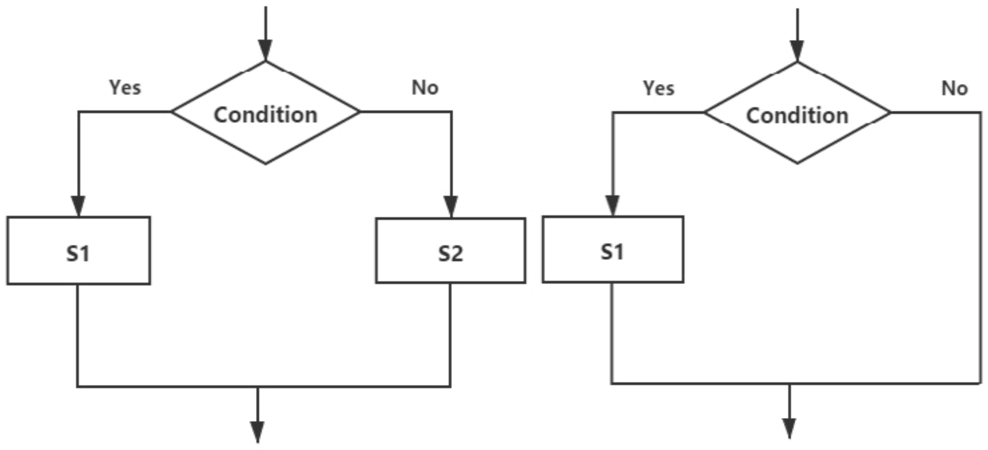
Sequential structure

As the name implies, sequential structure's program is the most basic and simple program structure, which is executed in the order of the statements.As shown in the figure, the program will execute the operations in the S1 box first and then the operations in the S2 box, sequentially.



Selective structure

In the program, it is sometimes necessary to make a judgment according to the situation so as to decide the next step.For example, the program makes a judgment on the light value in the current environment. If the light value is high, the light does not need to be turned on. If the light value instead low, the light needs to be turned on.A selective structure is used at this time, as shown in the following figure.The selective structure determines whether the condition is true, if yes, S1 is executed, otherwise, S2 is executed; Or S1 is perform, otherwise that selection structure is disengaged.



• if

If statement is the most common selective structure, when a given expression is true,the nested statement will run after the expression. An if statement has three structural forms as shown in the figure below.

Simple branch structure： When satisfied...then execute

```
if (expression) {
  statement;
}
```

Double–branched structure:When satisfied...execute...Otherwise execute

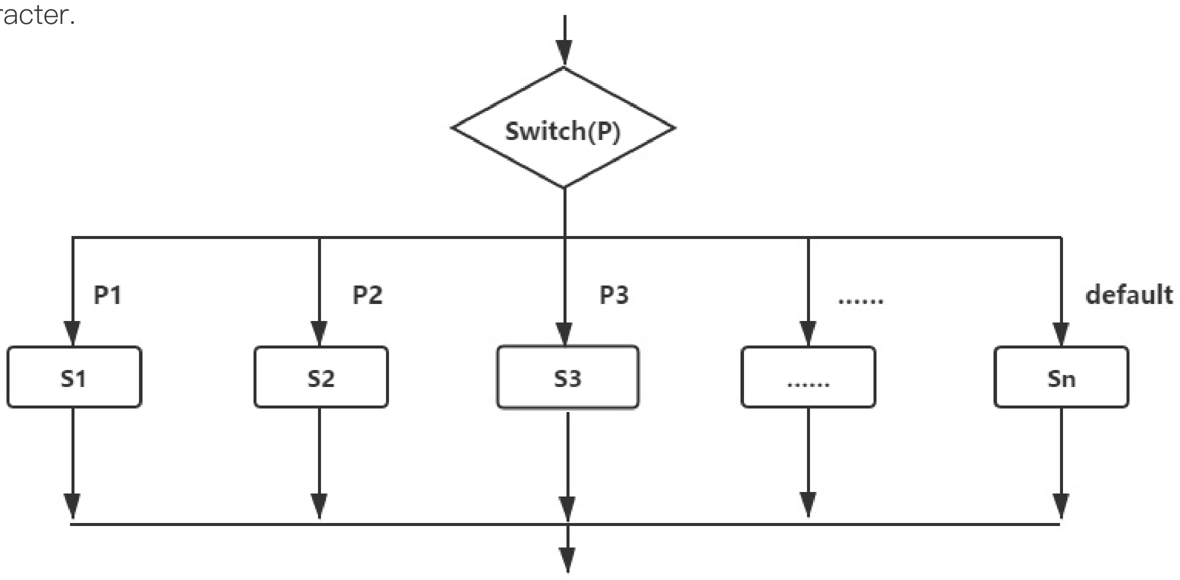
```
if (expression) {
  statement 1;
}
else {
  statement 2;
}
```

Multi–branch structure:if statements nested to judge different situations

```
if (expression 1) {
  statement 1;
}
else if (expression 2) {
  statement 2;
}
else if (expression 3) {
  statement 3;
}
```

• switch……case

When dealing with multiple-choice branches, it would be very tedious to write a program with the "if ... else" structure, and it would be much more convenient to use the switch statement at this time.switch.The switch structure compares the expression in parentheses with the constant after case, and executes the statement corresponding to the expression if they match,finally exiting the structure through a break statement. Or it runs the statement after default if none match.It should be noted that the expression in parentheses after "switch" must be an integer or a character.



```
switch (expression) {
  constant expression 1:
    statement 1;
    break;
  constant expression 2:
    statement 2;
    break;
    ……
  default:
    statement n;
    break;
}
```

• break

The break statement can only be used in a switch multi-branch selection structure and a loop structure. It terminates the current program structure so that the program can jump to subsequent statements.

Loop structure

A loop structure is used when we need to repeatedly perform a part of the code, according to a given set of conditions that determine whether to continue an operation or exit the loop. There are three common loop statements:

• while

a while loop is an "while" type of loop that executes statements in the body of the loop when certain conditions are met.

```
while (expression) {
  statement;
}
```

• do……while

A "until" type loop in which statements in the body of the loop are executed once before the expression is checked, and the loop continues if the expression is true.

```
do {
  statement;
} while (expression);
```

• for

contains three expressions, expression 1 is initialization, expression 2 is judgment, and expression 3 is increment.

```
for (expression 1;expression 2;expression 3) {
  statement;
}
```

In addition to the loop statements above, there is also a control statement in the loop structure, break and continue, to end the loop or jump out of the loop in advance. In this lesson,we only need to understand the structure of these programs.In later courses, we will go through the projects step by step to master them.

Release Tasks

Task: Button to Control the LED Lights On and Off

Analysis

For the LED lights to switch on when the button is pressed, and then switch off once the button is released.The program is written in three steps:

• Define pins, create variables

• Initialize and set pin status

• Read the button status to judge the condition. If the button is pressed, the light will be on, otherwise the light will be off.

·Note·

Variables

Variable values in a program are called variables, for example, to define an integer variable I, which is called int i; We can assign values to variables while defining them, for example, int i =0; In addition, depending on the data type, the statements used to define variables are different, such as defining floating–point numbers, float x = 1.9, and so on. Details availablehttps://www.arduino.cc/reference/en/#variables

Write Program

Step 1: Define the pin and create the variable. The button on XIAO expansion board is D1,so we define it as pin 1 and set the variable for the button state.

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

Step 2: Set the pinmode of the LED as output, the pin of the button as INPUT_PULLUP. When the button is not pressed, it will return“1” or “HIGH”.When the button is pressed, it will return“0” or “LOW”.

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

Step 3: Repeatedly read the button status, and turn on the light when the button is pressed, and vice versa. Because XIAO onboard LED uses negative logic, so when the button is pressed, a 0 is returned and the LED lights switch on. When the button is released, 1 is returned instead and the LED lights are turned off.

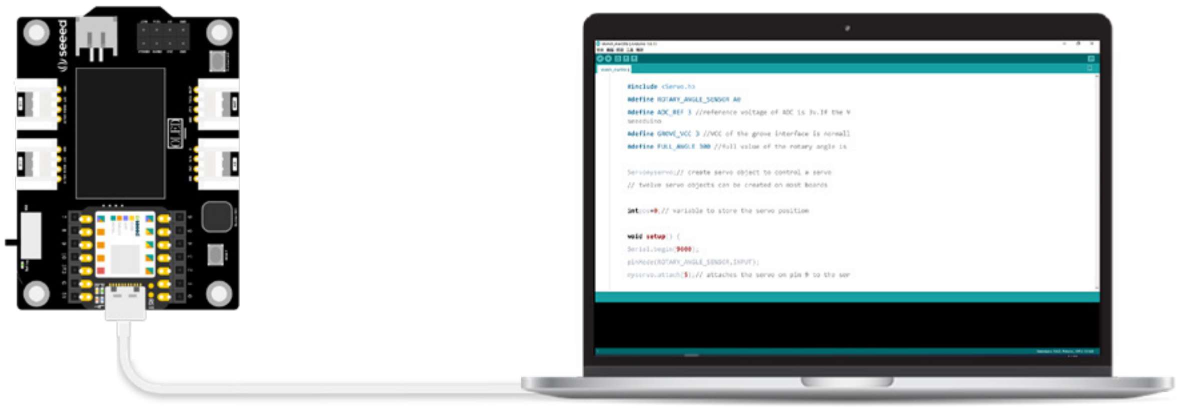
```
void loop() {
// read the state of the pushbutton value
 buttonState = digitalRead(buttonPin);
  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on
    digitalWrite(LED_BUILTIN, HIGH);
  }
  else {
    // turn LED off
    digitalWrite(LED_BUILTIN, LOW);
  }
}
```

The complete procedure is as follows:

L2-Button.ino

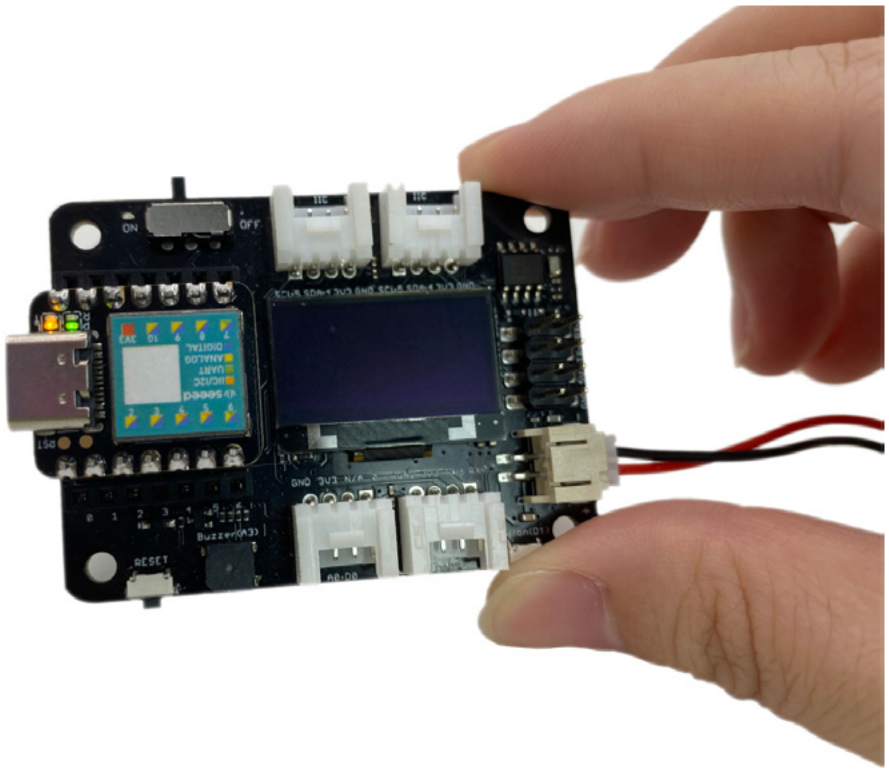
Upload Program

We will upload the written program to the hardware,The XIAO is firstly connected to the computer with the data cable in the kit.



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

Servomyservo;// create servo object to control a servo
// 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 the ser
```

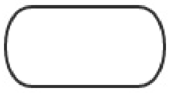
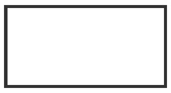
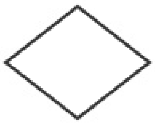
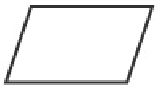
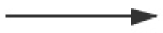
Next, click "Verification button" to verify the program. If verification is correct, click "Upload" button to upload the program to the hardware. When "Done uploading" is displayed in the debugging area, we can press the button to test if the LED light will light up.



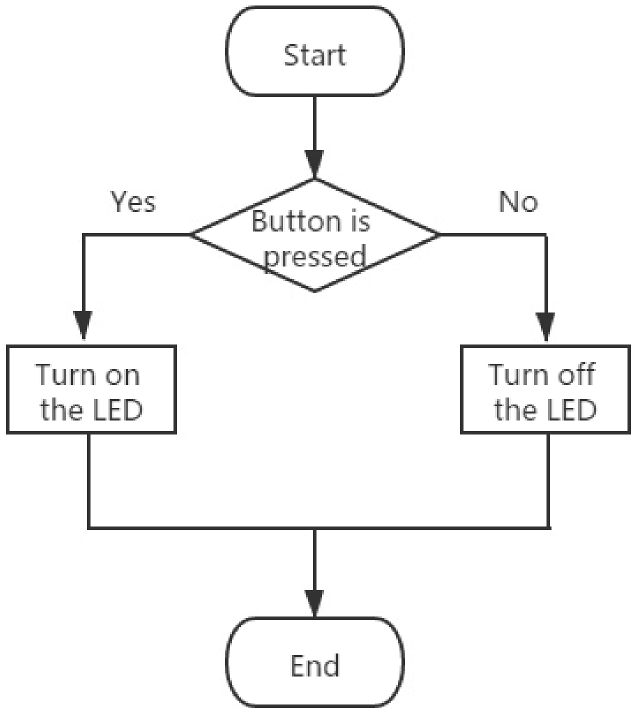
Intensive Learning

Flow chart

Before writing the program, we can draw the flow chart of the program to help us sort out our thoughts.The commonly used flow chart symbols are as follows:

	Start/End, an oval represents a start or end point
	Process, a rectangle represents a process
	Decision,a diamond indicates a decision
	Input/Output, a parallelogram input or output
	Arrows,a line is a connector that shows relationships between the representative shapes

In this lesson,we achieved control of the LED lamp with our program,which is represented by the flow chart as follows.You can try to draw a picture.



Lesson 3 Morse Code Transmitter

Everyone knows that "SOS" is an international common emergency signal and a kind of Morse code. Today, we will transform Seeeduino XIAO into a Morse code transmitter and send the message automatically through an on-board buzzer. We will also learn how to use a button to control the buzzer to sound and complete the task of a manual transmitter.

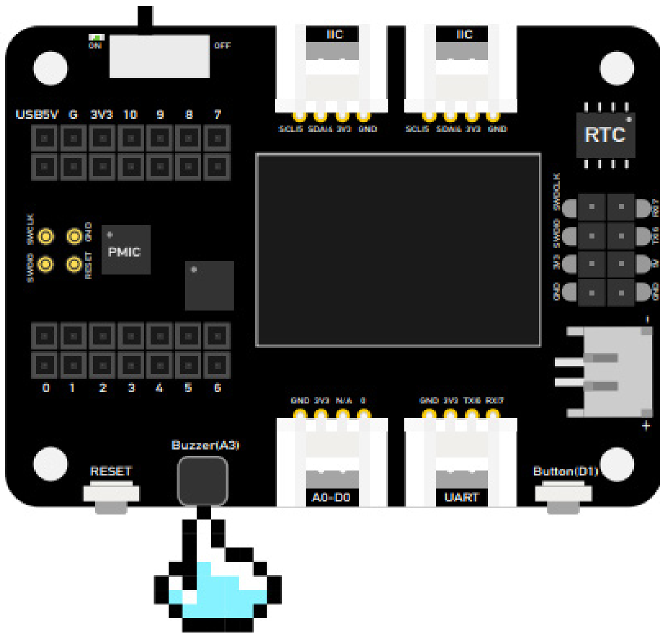
Background Knowledge

Buzzer

A buzzer is a kind of electronic sounding component, which is widely used in various electronic products, such as computers, printers, copiers, alarms, electronic toys, telephones, timers, etc.There are two types of buzzers: active buzzers and passive buzzers.The difference between the two types of buzzers is the presence of an internal oscillator.

- **Active buzzer:** There is an internal oscillator that will make a sound as long as the power is turned on.
- **Passive buzzer:** There is no internal oscillator, and waveform pulse signals are required to drive it to produce sounds of various tones and effects.

Independent buzzer module as shown in the figure right:



In the Seeduino XIAO expansion board, there is an onboard passive buzzer connected to the A3 pin, to which we can output PWM waveform pulse signals to control the buzzer sound.

tone()and noTone()function

• tone()

The tone() function can generate a PWM signal at a fixed frequency to drive the passive buzzer, and can define the frequency and duration of the sound..

Syntax:

tone(pin, frequency);

tone(pin, frequency, duration);

Parameters:

pin: Sounding pin (pin for buzzer connection, A3 in seeeduino XIAO expansion board).

frequency: The frequency(unit:HZ) of the sounding tone, the permissible type being an unsigned integer.

duration: The duration of the sound (in milliseconds, an optional parameter) and the allowed types are unsigned long integers.

• noTone()

This function is used to stop a buzzer that is sounding under the control of the tone () function. If no sound is being generated, the function is invalid.

Syntax:

noTone(pin);

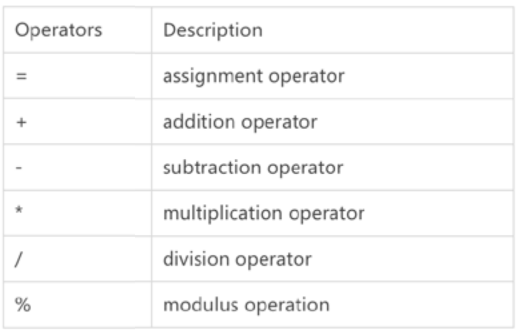
Parameters:

pin:the Arduino pin on which to stop generating the tone.

Common Operators

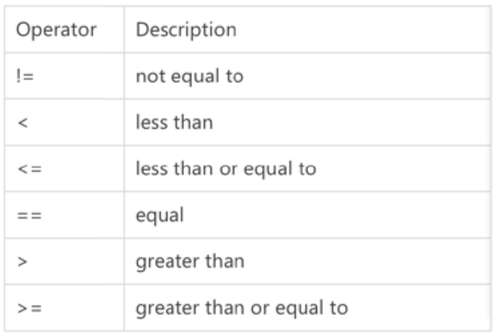
In the previous study, we have already used some operators. We will now study some common types of operators and the methods used.

• Arithmetic operators



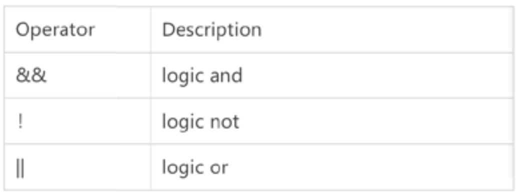
Operators	Description
=	assignment operator
+	addition operator
-	subtraction operator
*	multiplication operator
/	division operator
%	modulus operation

•Comparison operator



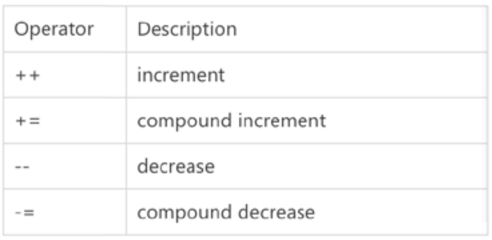
Operator	Description
!=	not equal to
<	less than
<=	less than or equal to
==	equal
>	greater than
>=	greater than or equal to

• Boolean operator



Operator	Description
&&	logic and
!	logic not
||	logic or

• Composite operator



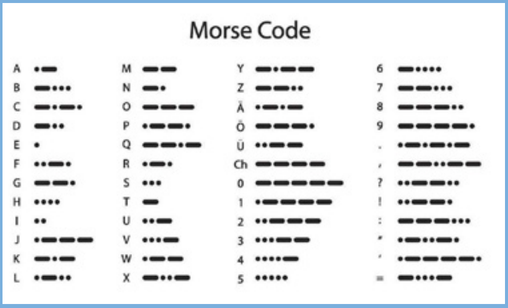
Operator	Description
++	increment
+=	compound increment
--	decrease
-=	compound decrease

Details available: https://www.arduino.cc/reference/en/

·Note·

Morse Code

Most people have heard of Morse Code and it often appears in detective novels or movies. What is it and how to use it?In fact, Morse code is a kind of code used for communication, which is characterized by time on and time off. It shows different English letters, numbers and punctuation marks through different arrangemen of two basic signals,Short mark is one time unit long called "dit"; longer mark is three time units long called "dah". Through these two signals and pauses in the middle, the individual characters and punctuation marks are sent out independently by telegraph, which is the original morse code.


Morse Code

A	•—	M	——	Y	—•——	6	—••••
B	—•••	N	—•	Z	——••	7	——•••
C	—•—•	O	———	Ä	•—•—	8	———••
D	—••	P	•——•	Ö	———•	9	————•
E	•	Q	——•—	Ü	••——	.	•—•—•—
F	••—•	R	•—•	Ch	————	,	——••——
G	——•	S	•••	0	—————	?	••——••
H	••••	T	—	1	•————	!	••——•
I	••	U	••—	2	••———	:	———•••
J	•———	V	•••—	3	•••——	"	•—••—•
K	—•—	W	•——	4	••••—	'	•————•
L	•—••	X	—••—	5	•••••	=	—•••—

Release Tasks

Task 1: Automatically Send "SOS"

Analysis

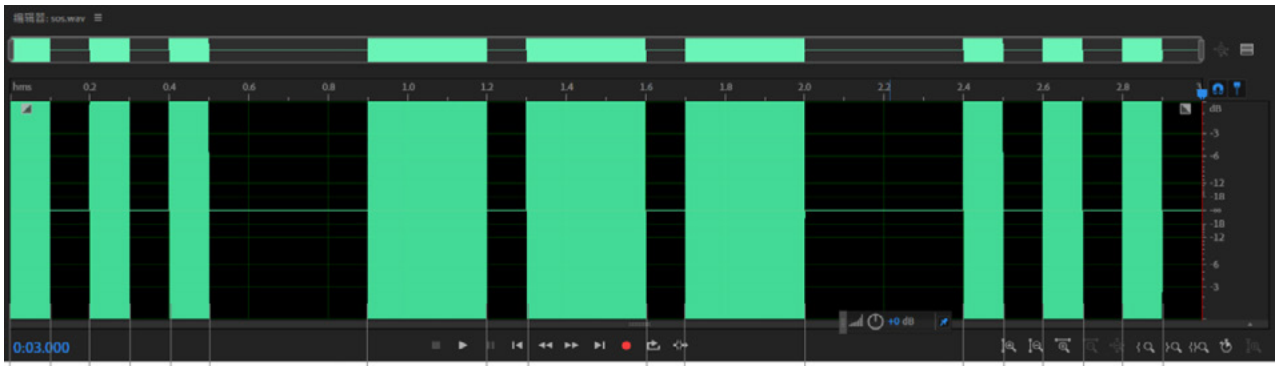
Automatic transmission means that when the control panel is started, the on-board buzzer will automatically send out the Morse code of "SOS". The program is written in three steps:

• Define buzzer pin

• Initialize, set buzzer pin state

• Loop buzzer to play "SOS" Morse code

At first, let's study how the "SOS" morse code is embodied through the program.By importing Morse code audio files into audio editing software, we can see the sound waveform and the duration of each syllable, which can be divided into two types: long sound and short sound.In order to understand programming conveniently, we mark the switch of buzzer in binary way. 1 means buzzer on, 0 means buzzer off, and gray number dictates how long the current state needs to last.When one section ends, it needs to be set to 0.8 seconds between the two codes because it needs to be played in a loop.



1 0.1		1 0.1		1 0.1		1 0.3		1 0.3		1 0.3		1 0.1		1 0.1		1 0.1	
	0 0.1		0 0.1		0 0.4		0 0.1		0 0.1		0 0.4		0 0.1		0 0.1		0 0.8

We use tone(), noTone(), and delay() to make the buzzer transmit “SOS”,as shown in the code below,which corresponds to a short tone.

```
tone(pinBuzzer, 200);
delay(100);
noTone(pinBuzzer);
delay(100);
```

Writing Program

Step 1: Define Pins and Create Variables int

```
int pinBuzzer = 3;//define the buzzer as pin 3, and the onboard buzzer on the XIAO expansion board is A3
```

Step 2: Set the pin state

```
void setup() {
pinMode(pinBuzzer, OUTPUT);// initialize the buzzer pin as an output
```

Step 3: Play “SOS” Morse Code in a loop

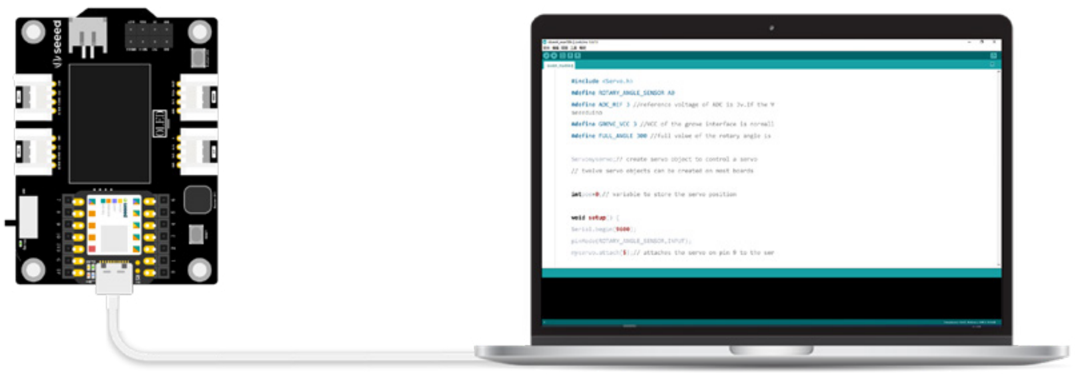
```
void loop() {
//transmit a short tone three times
  tone(pinBuzzer, 200);
  delay(100);
  noTone(pinBuzzer);
  delay(100);
  tone(pinBuzzer, 200);
  delay(100);
  noTone(pinBuzzer);
  delay(100);
  tone(pinBuzzer, 200);
  delay(100);
  noTone(pinBuzzer);
  delay(400);

//transmit a long tone three times
  tone(pinBuzzer, 200);
  delay(300);
  noTone(pinBuzzer);
  delay(300);
  tone(pinBuzzer, 200);
  delay(300);
  noTone(pinBuzzer);
  delay(300);
  tone(pinBuzzer, 200);
  delay(300);
  noTone(pinBuzzer);
  delay(400);

//transmit a short tone three times
  tone(pinBuzzer, 200);
  delay(100);
  noTone(pinBuzzer);
  delay(100);
  tone(pinBuzzer, 200);
  delay(100);
  noTone(pinBuzzer);
  delay(100);
  tone(pinBuzzer, 200);
  delay(100);
  noTone(pinBuzzer);
  delay(800);
}
```

Please see details on entire program:

L3-SOS.ino

Upload Program

We will upload the written program to the hardware, and at first connect XIAO to the computer with the data cable in the kit.



Next, click the “verity” to verify the program.if the verification is correct, click the “Upload” button to upload the program to the hardware.When the debugging area shows “Done uploading”, just listen to the Morse code and see if it’s what you want.

Task 2: Control Buzzer by Button

Button controls buzzer sound, or you can transmit Morse code manually.The logic of the code is very simple, an if function can be used to determine whether the button is pressed. If pressed, the buzzer will sound.

Analysis:

Three steps for programming:

• Define buzzer, button pin

• Initialize, set buzzer, button pin state

• Check whether the button has been pressed, if so, it will sound.

Writing Program:

Step 1: Define buzzer and button pin

```
const int buttonPin = 1; // the number of the pushbutton pin
int pinBuzzer = 3;//the number of the buzzer pin
```

Step 2: Set pin state of button and buzzer

```
void setup() {
  // initialize the buzzer pin as an output
  pinMode(pinBuzzer, OUTPUT);
  // initialize the pushbutton pin as an input
  pinMode(buttonPin, INPUT_PULLUP);
}
```

Step 3: Check the button status. If the button is pressed, the buzzer will sound.The tone () function will control the sound of the passive buzzer.

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

  // check if the pushbutton is pressed. If it is, the buttonState is LOW
  if (buttonState == LOW) {
  // play a note on buzzer pin for 200 ms
  tone(pinBuzzer, 200, 200);
  }
}
```

Please see details on entire program:

L3-ButtonSOS.i
no

Upload Program

We will upload the written program to the hardware, and at first connect XIAO to the computer with the data cable in the kit.

Next, click the “verity” to verify the program.if the verification is correct, click the “Upload” button to upload the program to the hardware.When the debugging area shows “Done uploading”,just test whether buzzer will sound.

Intensive Learning

The passive buzzer can emit different tones to form a simple melody. Try to play the happy birthday song with the buzzer!

L3-HappyBirthd
ay.ino

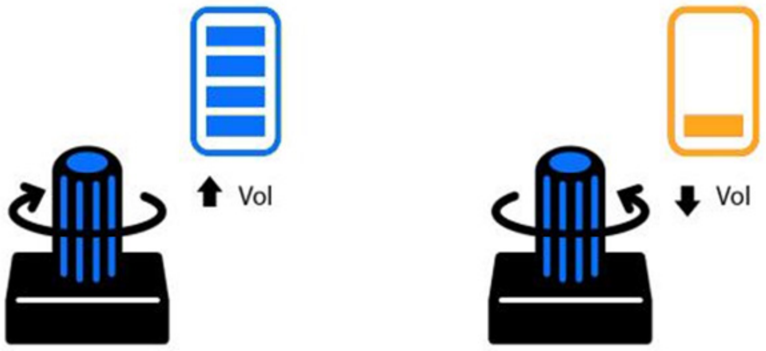
Lesson 4 Communicate with your Application using the Serial Port Monitor

We can visually observe the external hardware working when we write a few lines of code to control the development board to light up the LED lights, drive the sensor to control the actuator or control the buzzer with a button .If we're lucky, our program will work as expected, but what if it doesn't? There are no errors in the compilation of the program, so where did we go wrong? If only these mistakes could tell us.In this lesson, we'll learn how to talk to your computer and view the status and information of programs and hardware through the serial monitor.

Background Knowledge

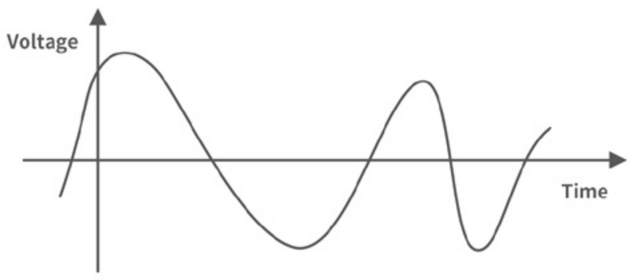
Knob Potentiometer

Knob potentiometer is an unusual term, but it is widely used in home appliances or industrial equipment. For example, a knob on a stereo might be used to adjust the sound volume.



The knob potentiometer produces an analog output value between 0 and VCC on its connected pin. By rotating the knob, the output voltage can be changed. the angle range of the knob is 300 degrees,and the output value is 0–1023.We can control the LED lamp to produce a brightness change by the knob potentiometer,and also control the steering engine to rotate at different angles, etc.

Analog I/O



In the Arduino Series development boards, analog input pins are the pin numbers preceded by an “A”. We can read the analog values on these pins to achieve the desired effect.

Analog signal

Analog signals can be seen everywhere in life, such as changes in sound, light, temperature, etc.The frequency and amplitude of analog signals change continuously with time.

How can we read pin analog values through the development board?The analog input pin has an analog-to-digital converter, which converts the analog signal from the external input into a digital signal that the can be recognized by the board,thereby enabling the ability to read in analog values.For example, a 0-5V voltage signal can be converted to an integer value of 0-1023.

• analogRead()

Reads the value from the specified analog pin.

Syntax:

analogRead(pin);

Parameters:

pin：the name of the analog input pin to read from.

• analogWrite()

The analog input corresponds to the analog output, which is achieved by the analogWrite().It should be noted that when we use this function,we only output different voltages in a special way to achieve an approximate analog effect, which is called pulse width modulation, Therefore we write the PWM square wave to the specified pin instead of the real analog value.

Syntax:

analogWrite(pin, value);

Parameters:

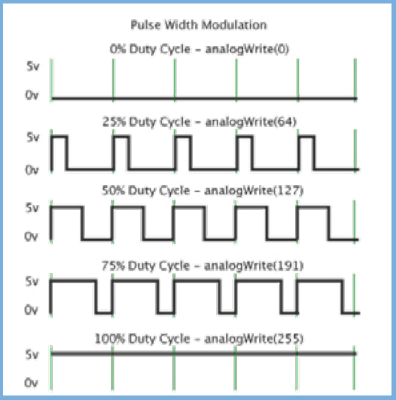
pin：the Arduino pin to write to. Allowed data types: int.

value：the duty cycle: between 0 (always off) and 255 (always on). Allowed data types: int.

·Note·

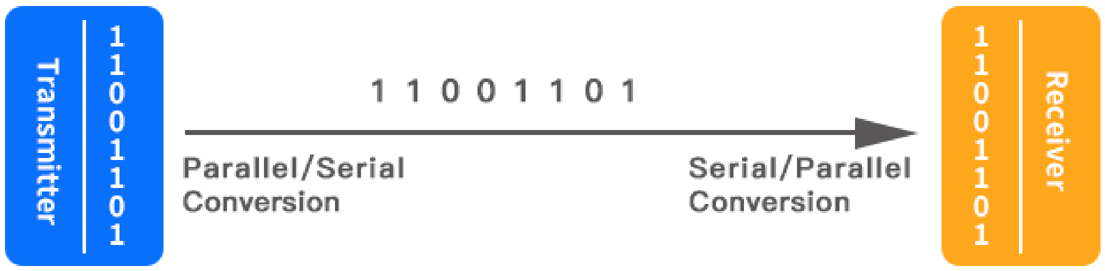
PMW Pulse Width Modulation

Pulse width modulation (PWM) is a method to obtain analog results from a digital output. The purpose of controlling the charging current is achieved by adjusting the period of PWM and the duty cycle of PWM.As shown in the figure, the voltage is switched back and forth between 0V (low value) and 5V (high value), and a single switching is a cycle.If the time of high voltage is 25%, the time of low voltage is 75%, the duty cycle is 25%, and the output voltage is 5V*25%=1.25V. Taking an LED lamp as an example, if the LED is connected to this pin at this time, it will generate the brightness presented by a voltage of 1.25V, which is also equal to the brightness presented by analogWrite(64).In this way, the brightness of the LED can be changed. The PWM output signals at 0% duty cycle, 25% duty cycle, 50% duty cycle, 75% duty cycle, and 100% duty cycle are shown.


Pulse Width Modulation
0% Duty Cycle - analogWrite(0)
25% Duty Cycle - analogWrite(64)
50% Duty Cycle - analogWrite(127)
75% Duty Cycle - analogWrite(191)
100% Duty Cycle - analogWrite(255)
5v
0v

Serial Communication

If we want XIAO to communicate with other devices, the most common way to do so is serial communication.The development boards in the Arduino series are capable of serial communication. The language recognized by the computer is binary, such as 1010.Therefore, serial communication between electronic devices is achieved by sending and receiving binary data through serial port. The key component to achieve this function is USART.In Arduino IDE, we can observe the data sent and received through the serial port monitor, which can be achieved by the relevant serial port communication functions.



- **Serial.begin()**

Sets the data rate in bits per second (baud) for serial data transmission.

Syntax:

Serial.begin(speed);

Parameters:

Serial: serial port object.

speed: in bits per second (baud).General values for 9600, 115200, etc

- **Serial.println()**

Prints data to the serial port as human-readable ASCII text followed by a carriage return character (ASCII 13, or '\r') and a newline character (ASCII 10, or '\n'). This command takes the same forms as Serial.print().

Syntax:

Serial.println(val);

Parameters:

Serial: serial port object.

speed: the value to print. Allowed data types: any data type.

Example: in serial monitor output:

We initialize the serial port in the setup() function, and send an output to the serial port:"hello world!!!" in the loop().

```
void setup() {
Serial.begin(9600);//start serial port at 9600 bps
}
void loop() {
Serial.println("hello world!!!"); //print"hello world!!!"
}
```

Returning to the question at the beginning of this lesson: When the code we wrote was verified successfully but did not work as expected (for example, when hardware is unresponsive), how can we tell what went wrong?Next, we will learn how to use a serial port monitor "talks" to the hardware.

Release Tasks

Task 1: Checking Whether the Button is Pressed Through the Serial Monitor

Analysis

Do you remember that the button can control the LED light switch? Some of the codes can be reused.We only need to get the code of setting the button switch,add the initialization serial port, and send the data to the serial port.The program is written in three steps:

- Define button pins, define variables
- Initialize the serial port, set the serial baud rate, set the button switch pin state
- Read button status and send to serial port

Write Program

Step 1: Define button pins, define variables.

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

Step 2: Initialize the serial port, set the serial baud rate, set the button switch pin state.

```
void setup() {
pinMode(buttonPin, INPUT_PULLUP);  // initialize the pushbutton pin as an input
Serial.println(9600);// start serial port at 9600 bps:
}
```

Step 3: Read button status and send to serial port.

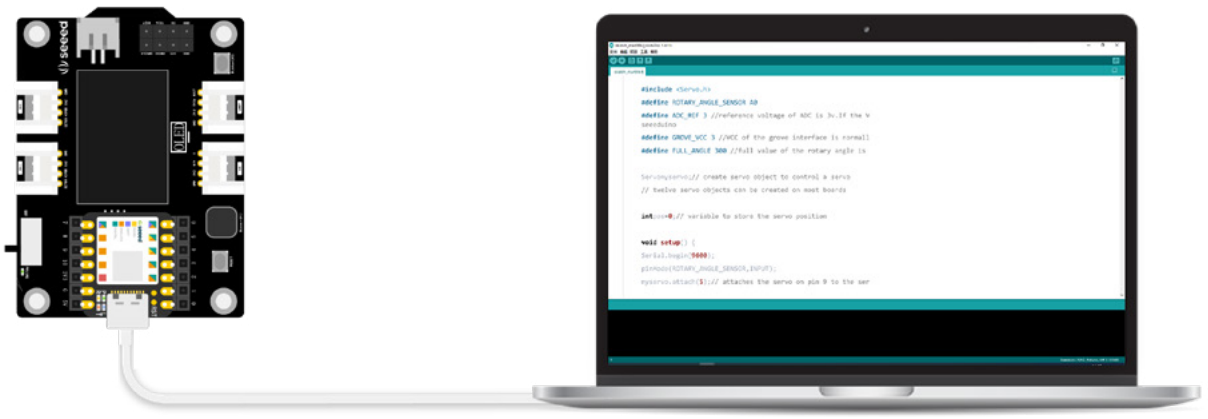
```
void loop() {
buttonState = digitalRead(buttonPin);  // read the state of the pushbutton value
Serial.println(buttonState);//Send button status data to serial port
delay(500);
}
```

Please see details on entire program:

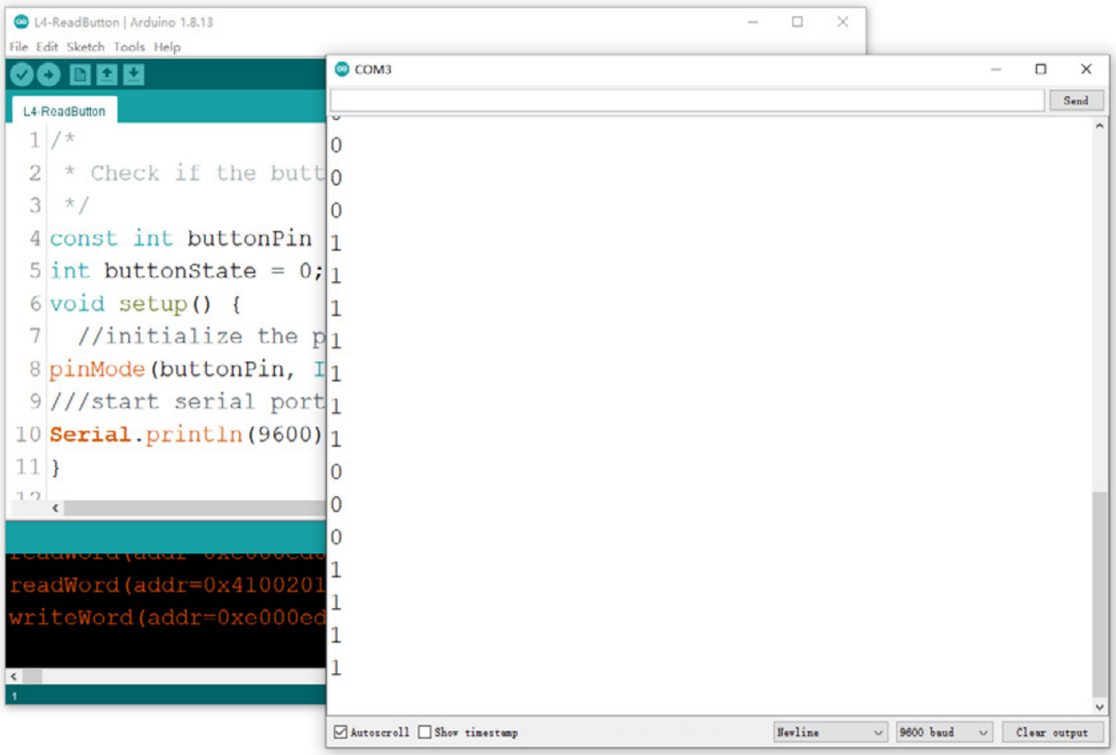
L4-ReadButton.i
no

Upload Program

We will upload the written program to the hardware, and at first connect XIAO to the computer with the data cable in the kit.



We will upload the written program to the hardware. First connect XIAO to the computer with the data cable in the kit.Next, click the “Verify ” ,if the verification is correct, click the “Upload” button to upload the program to the hardware.When the debugging area shows “Done uploading”, the serial monitor, we can observe the change of the value printed by the serial monitor when the button is pressed and released. What did you find?



```
/*
 * Check if the butt
 */
const int buttonPin
int buttonState = 0;
void setup() {
  //initialize the p
pinMode(buttonPin, I
///start serial port
Serial.println(9600)
}
```

```
readWord(addr=0x4100201
writeWord(addr=0xe000ed
```

```
0
0
0
1
1
1
1
1
1
1
0
0
0
1
1
1
1
```

When we press the button of XIAO expansion board, the serial monitor displays 0. When the button is released, the serial monitor displays 1.

Task 2: Viewing Knob Potentiometer through Serial Port Monitor

Analysis

In Task 1, the button switch is a digital input, which sends out a digital signal of 0 and 1, while the knob potentiometer returns an analog signal. We need to read the rotation angle of the knob potentiometer on the A0 pin and send it to the serial port. The program is also divided into three steps.

- Define knob potentiometer pins, define variables
- Initialize the serial port and set the knob potentiometer pin status
- Read and calculate the knob potentiometer rotation angle value and send to the serial port

Write Program

Step1: Define knob potentiometer pins, define variables.The voltage value of ADC and the reference voltage of grove module need to be defined, because we need to calculate the voltage variation after the knob switch is connected into the circuit through these voltage values.

```
#define ROTARY_ANGLE_SENSOR A0
#define ADC_REF 3 //reference voltage of ADC is 3v.
#define GROVE_VCC 3 //VCC of the grove interface is normally 3v
#define FULL_ANGLE 300 //full value of the rotary angle is 300 degrees
```

Step 2: Initialize the serial port, set the serial baud rate, set the button switch pin state.

```
void setup()
{
  Serial.begin(9600);//start serial port at 9600 bps:
  pinMode(ROTARY_ANGLE_SENSOR, INPUT);//initialize the knob pin as an input
}
```

Step 3: Read and calculate the knob potentiometer rotation angle value and send to the serial port.First, we set the data type of the voltage variable, set the analog value variable of the knob potentiometer pin, and then calculate the real-time voltage.After this, the rotation angle of the knob potentiometer is calculated.

```
void loop()
{
    float voltage;//the variable voltage is floating point type
     int sensorValue = analogRead(ROTARY_ANGLE_SENSOR);//read the analog value at the knob potentiometer pin
    voltage = (float)sensorValue*ADC_REF/1023;//calculate the real-time voltage
     float degrees = (voltage*FULL_ANGLE)/GROVE_VCC;//calculate the rotation angle of the knob potentiometer
     Serial.println("The angle between the mark and the starting position:");//serial port printing characters
    Serial.println(degrees);//serial printing Knob Potentiometer Rotation angle value
    delay(100);
}
```

·Note·

#define

#define is a preprocessing command that allow the programmer to give a name to a constant value before the program is compiled.Such as #define ledPin 5,at the time of compilation, ledPin will be assigned a value of 5.

Syntax:#define constantName value.

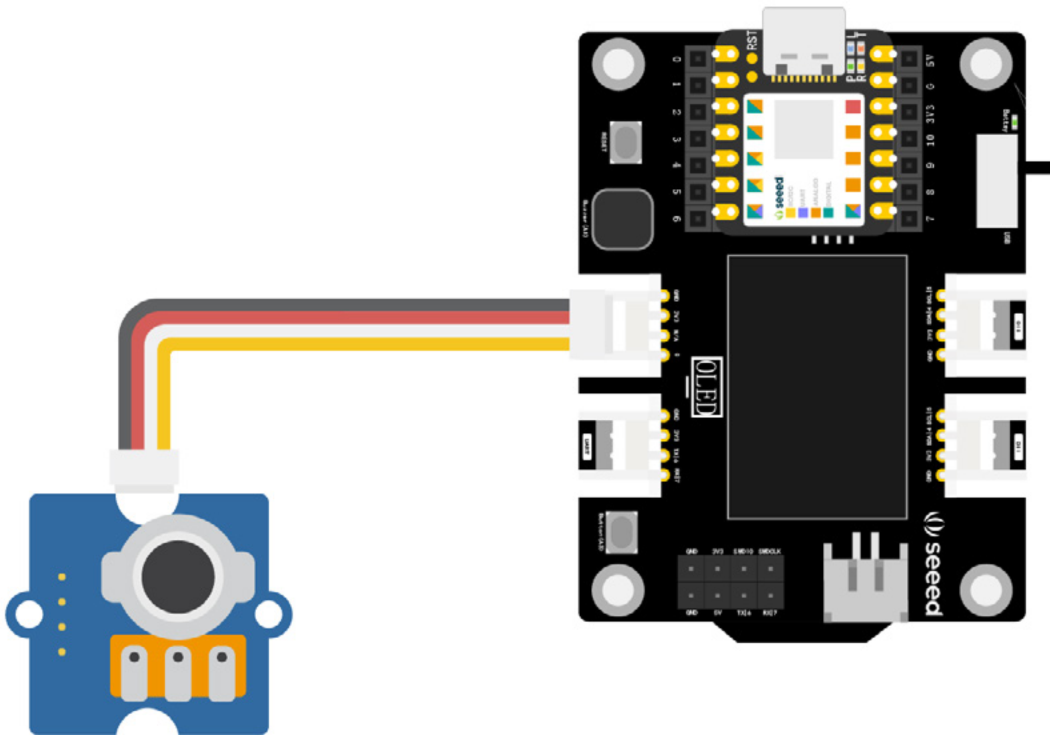
The constant name constant value "#" sign is required and does not need to be used at the end of a sentence ";" symbol

Please see details on entire program:

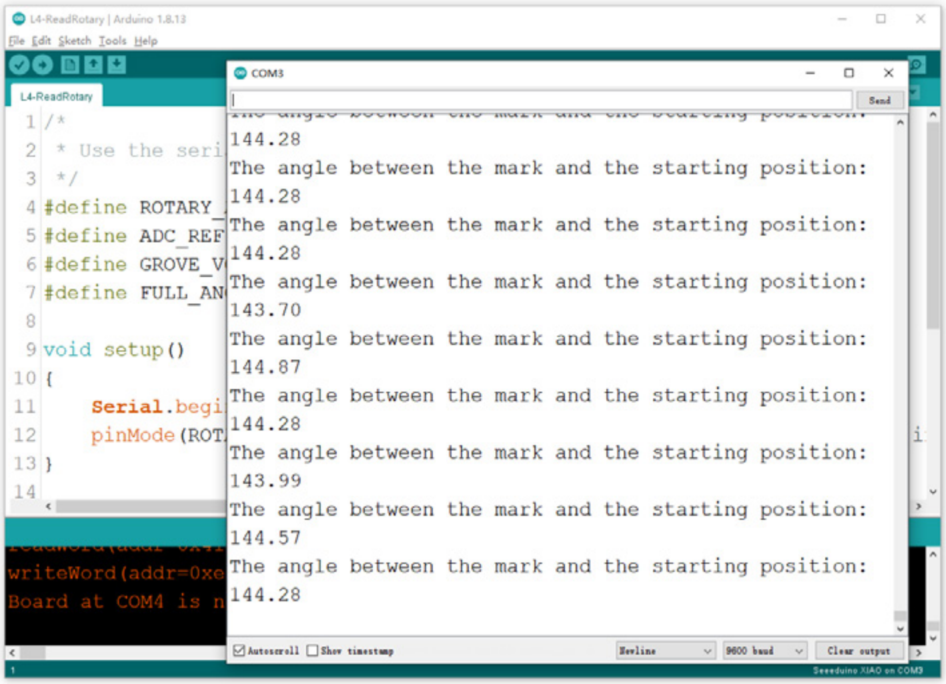
L4-ReadRotary.ino

Upload Program

After the program is written, we need to connect the XIAO main control board to the computer with data wires. In addition, we need to connect external sensors, and connect the knob potentiometer to A0 interface with four–color Grove wire, as shown in the following figure:



Next, click the "verify" to verify the program.If the verification is correct, click the "Upload" button to upload the program to the hardware.When the debugging area shows "Done uploading". Open the serial monitor and turn the knob potentiometer to observe the data change with the angle of the knob.



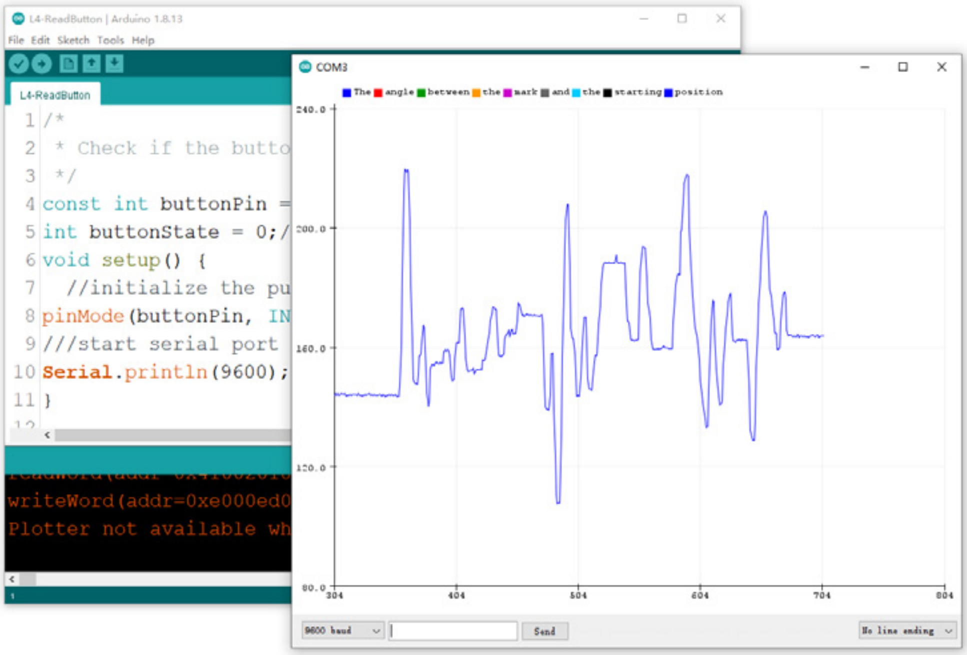
```
/*
 * Use the seri
 */
#define ROTARY_
#define ADC_REF
#define GROVE_V
#define FULL_AN

void setup()
{
    Serial.begi
    pinMode(ROT
}
```

```
writeWord(addr=0xe
Board at COM4 is n
```

```
144.28
The angle between the mark and the starting position:
144.28
The angle between the mark and the starting position:
144.28
The angle between the mark and the starting position:
143.70
The angle between the mark and the starting position:
144.87
The angle between the mark and the starting position:
144.28
The angle between the mark and the starting position:
143.99
The angle between the mark and the starting position:
144.57
The angle between the mark and the starting position:
144.28
```

⭐ Intensive Learning

When we use the serial port monitor to observe the angle value of the knob potentiometer, we find that the value changes continuously. To make data visualisation more intuitive, we can also use the serial port plotter to chart the data printed to the serial port of Arduino in real time. On the basis of task 2, close the serial monitor, and open the “tools–> serial plotter” as shown in the figure below:



L4-ReadButton | Arduino 1.8.13
File Edit Sketch Tools Help
L4-ReadButton
COM3
The angle between the mark and the starting position
240.0
200.0
160.0
120.0
80.0
304
404
504
604
704
804
9600 baud
Send
No line ending

```
/*
 * Check if the butto
 */
const int buttonPin =
int buttonState = 0;/
void setup() {
  //initialize the pu
pinMode(buttonPin, IN
///start serial port
Serial.println(9600);
}
```

```
writeWord(addr=0xe000ed0
Plotter not available wh
```

In the serial port plotter, the data acquired by the serial port will be plotted into an XY axis graph, where the X axis represents time, and the Y axis represents the data acquired by the serial port. Through the chart, we can see the change in the data more intuitively. Please try it.

Lesson 5 The Magic of Knob Potentiometer

In the last lesson, we learned the serial monitor and observed the difference between digital input and analog input through it.In this lesson, we will further explore analog application in combination with knob potentiometer!

Background Knowledge

Servo and Servo.h

Servo

The servo motor is a DC motor with a gear and a feedback system.The signal output to the circuit can make the servo rotate to a specific angle.It is suitable for electronic equipment or robots that need precise control.

servo.h

When we want to control a servo through XIAO or other Arduino series development board,we can use the servo library, which is one of Arudino standard libraries.

Relevant functions of servo.h files:

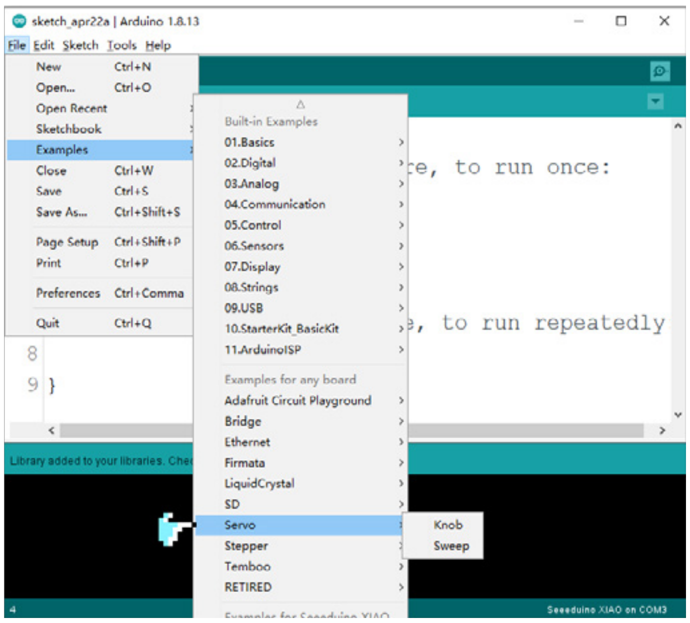
• Declaring Use of Library : #include <Servo.h>

• Create servo object to control a servo: Servo myservo;

• Attach the servo on pin number to the servo object: myservo.attach();

• Instruct servo to go to position in variable: myservo.write();

Instead of installing it manually, we can open the sample program "File → Sample → Servo" and see the sample programs "Knob" and "Sweep" to get familiar with the use of the servo.



map()

Re-maps a number from one range to another. That is, a value of fromLow would get mapped to toLow, a value of fromHigh to toHigh, values in-between to values in-between, etc.

Syntax:

map(value, fromLow, fromHigh, toLow, toHigh)

Parameters:

value:the value to map.

fromLow:the lower bound of the value's current range.

fromHigh:the upper bound of the value's current range

toLow:the lower bound of the value's target range.

toHigh:the upper bound of the value's target range.

Example:Map an analog value to 8 bits (0 to 255)

```
void setup() {}
void loop() {
  int val = analogRead(0);
  val = map(val, 0, 1023, 0, 255);
  analogWrite(9, val);
}
```

Release Tasks

Task 1: Use the Knob Potentiometer to Control the LED Lamp

Analysis

When controlling the LED lamp with the knob potentiometer, we need to implement this through the map() function, because the analog value that is read from the knob potentiometer is 0–1023, which is not the angle value of knob rotation. First, we need to calculate the angle value of knob potentiometer rotation, and then map this value to the brightness range of 0–255 of the LED lamp through the map() function. The programming steps are as follows:

• Define knob potentiometer, LED lamp pin

• Initialize the serial port and set the pin status of knob potentiometer and LED lamp

• Read and calculate the rotary angle value of the knob potentiometer and send it to the serial port.

• The angle value of the knob potentiometer is mapped to the LED lamp brightness value and stored to a brightness variable, and the LED lamp outputs the variables

Write Program

Step 1: Define the knob potentiometer, LED lamp pin, ADC, and VCC reference voltage to calculate the angle value of the knob potentiometer.

```
#define ROTARY_ANGLE_SENSOR A0
#define LEDPIN 13 //the number of the Led pin
#define ADC_REF 3 //reference voltage of ADC is 3v.
#define GROVE_VCC 3 //VCC of the grove interface is normally 3v
#define FULL_ANGLE 300 //full value of the rotary angle is 300 degrees
```

Step 2: Initialize the serial port and set the knob potentiometer and LED lamp pin status.

```
void setup()
{
    Serial.begin(9600);//start serial port at 9600 bps:
    pinMode(ROTARY_ANGLE_SENSOR, INPUT);//initialize the knob pin as an input
    pinMode(LEDPIN,OUTPUT); //initialize the LED pin as an output
}
```

Step 3: Read and calculate the knob potentiometer rotation angle value and send to the serial port.

```
void loop()
{
    float voltage;//the variable voltage is floating point type
    int sensor_value = analogRead(ROTARY_ANGLE_SENSOR);//read the analog value at the knob potentiometer pin
    voltage = (float)sensor_value*ADC_REF/1023;//calculating real-time voltage
    float degrees = (voltage*FULL_ANGLE)/GROVE_VCC;//calculate the rotation angle of the knob
    Serial.println("The angle between the mark and the starting position:");//serial port printing characters
    Serial.println(degrees);//serial port printing rotation angle value of knob potentiometer
    delay(100);
```

Step 4: Map the angle value of the knob potentiometer to the brightness value of the LED lamp and store it to the brightness variable.

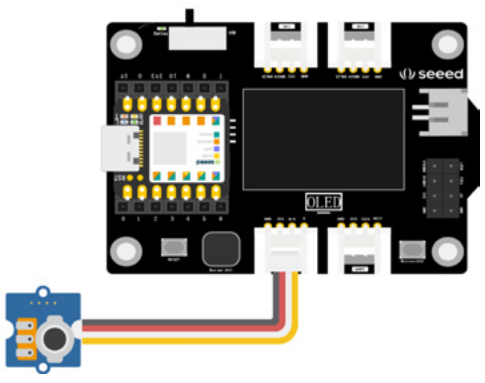
```
    //Next step 3
    int brightness;//define brightness variables
    brightness = map(degrees, 0, FULL_ANGLE, 0, 255);//The angle value of the knob potentiometer is mapped to the brightness value of the LED lamp and stored in the brightness variables
    analogWrite(LEDPIN,brightness);//LED output variable value
    delay(500);
}
```

Please see details on entire program:

L5-RotaryLed.ino

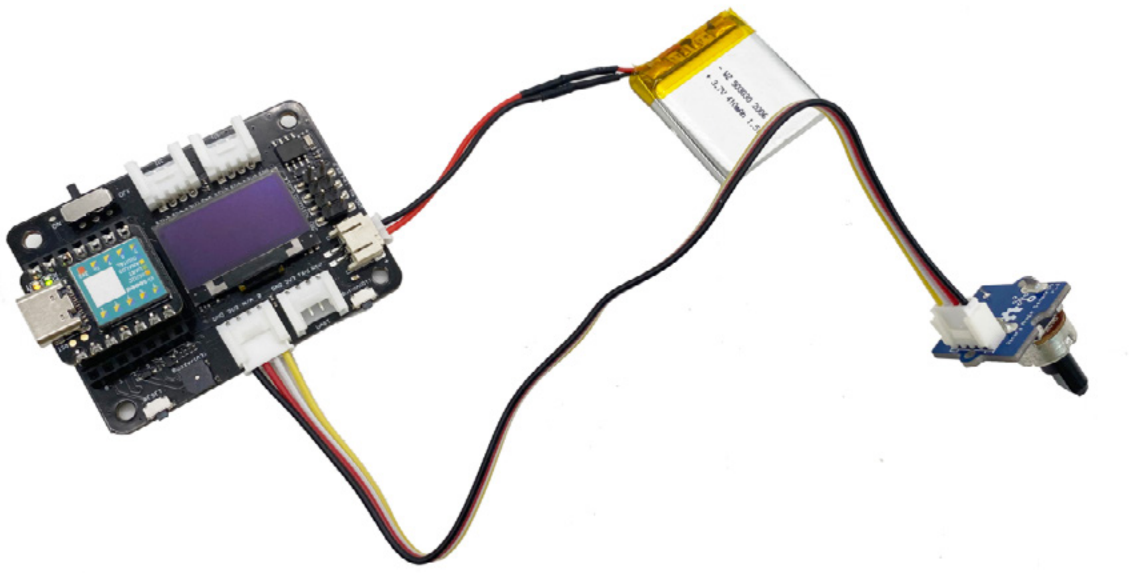
Upload Program

The program is written， we need to connect external sensors, and connect the knob potentiometer to A0 interface with four-color Grove wire, as shown in the following figure:



We need to connect the XIAO main control board to the computer with data wires, Next, click the "Verity" to verify the program.if the verification is correct, click the "Upload" button to upload the program to the hardware.When the debugging area shows "Done uploading",turn on the serial monitor to observe the angle value of the knob and the brightness change of the LED while turning the knob potentiometer.

Note: the LED lights are XIAO onboard LED lights.



Task 2: Control Servo with Knob Potentiometer

Analysis

When the servo is controlled by the knob potentiometer, we can call the serve library file to make partial adjustments to task 1. The program is divided into the following three steps:

• Declare and call servo library.Define the servo angle variable, knob potentiometer pin and voltage

• Initialize serial port, set knob potentiometer and servo pin status

• Read and calculate the rotation angle value of the knob potentiometer and send it to the serial port to drive the servo to rotate according to the change of the angle value

Write Program

Step1: Declare and call the servo.h, Define the servo angle variable, define the knob potentiometer pin and voltage.

```
#include <Servo.h>
#define ROTARY_ANGLE_SENSOR A0 //the number of the knob pin
#define ADC_REF 3 //reference voltage of ADC is 3v.
#define GROVE_VCC 3 //VCC of the grove interface is normally 3v
#define FULL_ANGLE 300 //full value of the rotary angle is 300 degrees
Servo myservo;  // create servo object to control a servo
int pos = 0; // variable to store the servo position
```

Step 2: Initialize the serial port and set the knob potentiometer and LED lamp pin status.

```
void setup() {
  Serial.begin(9600);
  pinMode(ROTARY_ANGLE_SENSOR, INPUT);//initialize the knob pin as an input
  myservo.attach(5);  // attaches the servo on pin 5 to the servo object
```

Step 3: Reads and calculates the rotation angle value of the knob potentiometer and sends it to the serial port to drive the servo to rotate according to the change of the angle value.

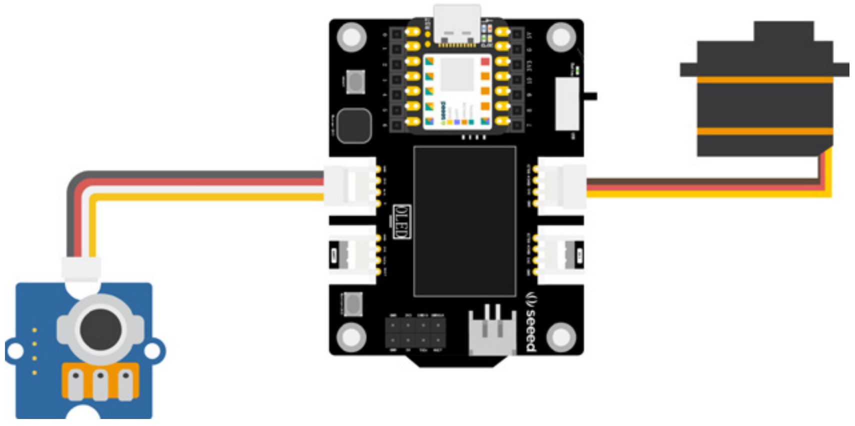
```
void loop() {
  float voltage;//the variable voltage is floating point type
  int sensor_value = analogRead(ROTARY_ANGLE_SENSOR);//read the analog value at the knob potentiometer pin
  voltage = (float)sensor_value * ADC_REF / 1023;//calculate the real-time voltage
  float degrees = (voltage * FULL_ANGLE) / GROVE_VCC;//calculate the rotation angle of the knob potentiometer
  Serial.println("The angle between the mark and the starting position:");//serial port printing characters
  Serial.println(degrees);//Serial port printing rotation angle value of knob potentiometer
  delay(50);
  myservo.write(degrees); //Write the rotation angle of knob potentiometer to the servo
}
```

Please see details on entire program:

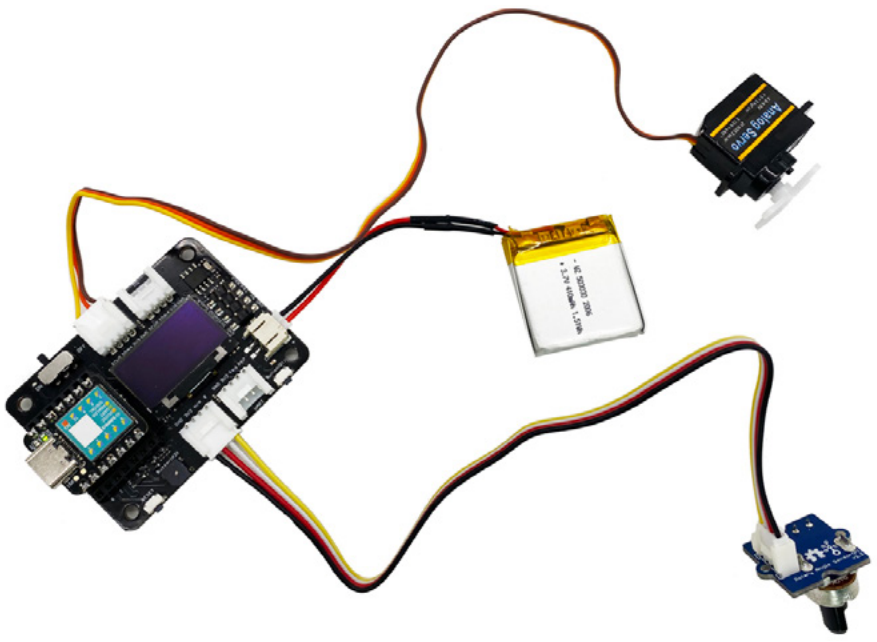
L5-RotaryServo.ino

Upload Program

After the program is written, we need to connect the XIAO main control board to the computer with data wires. Also connect the knob potentiometer and servo to XIAO expansion board as shown in the following figure:



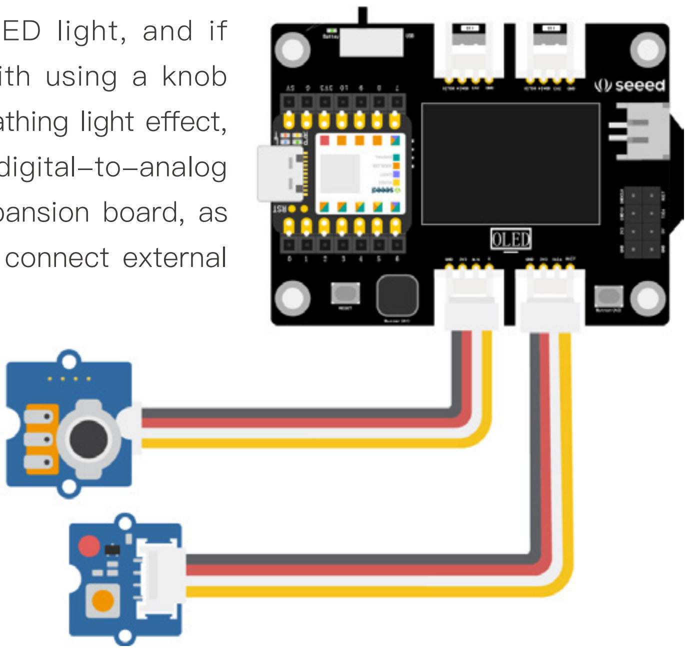
Next, click the "verify" to verify the program.if the verification is correct, click the "Upload" button to upload the program to the hardware.When the debugging area shows "Upload Successful",What do you find on the serial monitor when you turn on the serial port monitor, rotate the knob potentiometer, observe the change of the angle value and the rotation of the servo?



Note: The rotation angle range of the servo is 0–180, so when the angle value is greater than 180, the servo will stop rotating as seen in the serial monitor.

Intensive Learning

We've been using XIAO's onboard LED light, and if you want to add an external LED light with using a knob potentiometer to control it to produce a breathing light effect, how will this be achieved? There are two digital-to-analog GROVE interfaces LED out of the XIAO expansion board, as well as an A7/D7 interface, where we can connect external LED lights. As shown in the figure:



After connection, we will simply change the program of task one, changing #define LEDPIN 13 to #define LEDPIN 7.

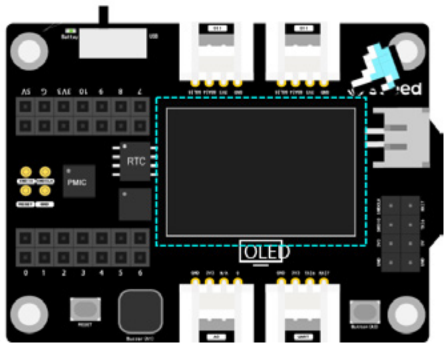
Lesson 6 OLED Shows Hello World!

In our life, we can see display screens everywhere, such as televisions, computers, mobile phones, vehicle-mounted display, LCD billboards, etc. Without them, our life would lose a lot of fun.Of course, the display is not only for leisure and entertainment, but also an indispensable tool for daily life.Common display screens include LCD display screens, LED display screens, 3D display screens, OLED display screens, etc. They also have advantages and disadvantages, and can be applied to different fields and scenes according to their respective characteristics.An OLED display is integrated on the XIAO expansion board. In this lesson, we will learn how to use this OLED to display text, patterns and pictures.

Background Knowledge

OLED Display

OLED is also called organic laser display and organic light-emitting semiconductor. Self-luminous, low power consumption, fast refresh rate, high resolution, light weight among other advantages make it widely used in different fields.On the XIAO expansion board, a 0.96-inch 128×64 pixel OLED display is integrated and available directly without a wired connection.In this project, we can display the values returned by time, temperature and humidity sensors through the OLED display screen, and can also directly display letters, numbers, graphics and even patterns to achieve the effect of visual interaction.



OLED U8g2

U8g2is a monochrome graphics library for embedded devices that supports multiple types of OLED displays so that we can write programs and achieve the desired effects.U8g2 library also includes U8x8 library, which has different functions:

U8g2

Include all graphics process (line/box/circle drawing);

Supports multiple fonts, with (almost) no restrictions on font height;

Some memory in the microcontroller is required to render the display.

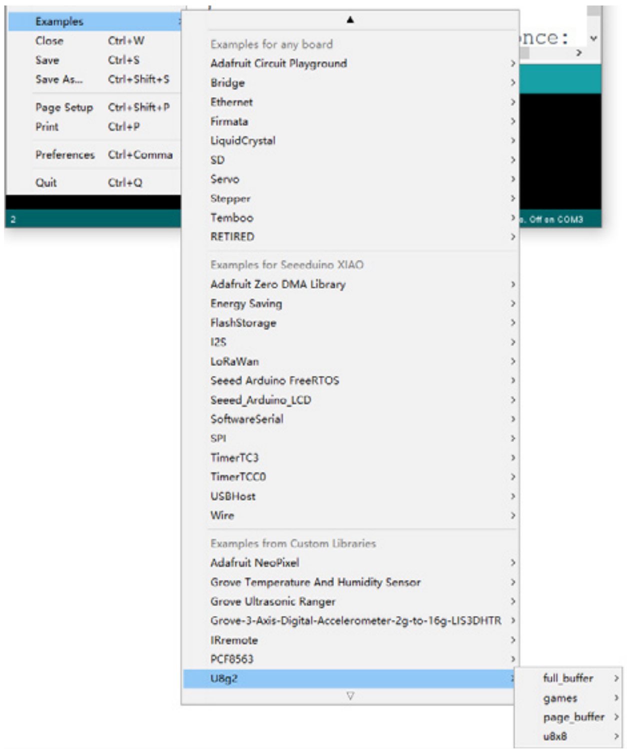
U8x8

Only text (character) output is supported;

Only fixed-size fonts (8x8 pixels) are allowed for each character;

Directly write to the display, no buffer required in microcontroller.

In a nutshell, when we want an OLED display to display a variety of visual content, such as a variety of fonts, graphics, patterns, etc., it can be achieved through U8g2 library. When we want to display characters directly with no requirement for fonts, for example where only the values and time of sensors can be displayed, it is more efficient to use U8x8 library.We can find many sample programs in “File → Examples → U8g2”. Through the sample programs, we can become familiar with the operation of this library.



Next, we will display characters and draw circles through the libraries.

Release Tasks

Task 1: OLED Displays Hello World!

Analysis

OLED displays “Hello World!” , write characters directly, with U8x8 library. The steps are as follows:

• Declare library files, set constructors, define display types, controllers, RAM buffer sizes, and communication protocols.

• Initialize display

• “Hello World!”Set the display font, set the print start position, and output “Hello World!”

Programming

Step 1: Declare library files, set constructors, define display types, controllers, RAM buffer sizes, and communication protocols.

```
#include <Arduino.h>
#include <U8x8lib.h>
U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(/* reset=*/ U8X8_PIN_NONE);
//set constructors, define display types, controllers, RAM buffer sizes, and
communication protocols
```

Step 2: Initialize the display.To use the functions in the library to set up the OLED display screen after declaring the library file.

```
void setup(void) {
  u8x8.begin();//initialize u8x8 library
   u8x8.setFlipMode(1);//enable (1) or disable (0) 180 degree rotation of the display content
  }
```

Step3: Sets the display font (multiple fonts are available in the u8x8 library.We can make our choice with reference to https://github.com/olikraus/u8g2/wiki/fntlist8x8) Set the print start position and output "Hello World!".

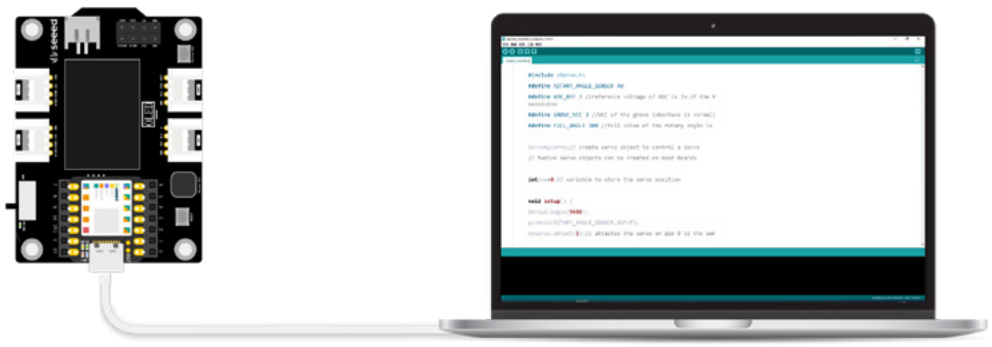
```
void loop(void) {
  u8x8.setFont(u8x8_font_chroma48medium8_r);//set the font for display.
  u8x8.setCursor(0, 0);//define the cursor for the print function.
  u8x8.print("Hello World!");//print Hello World !
}
```

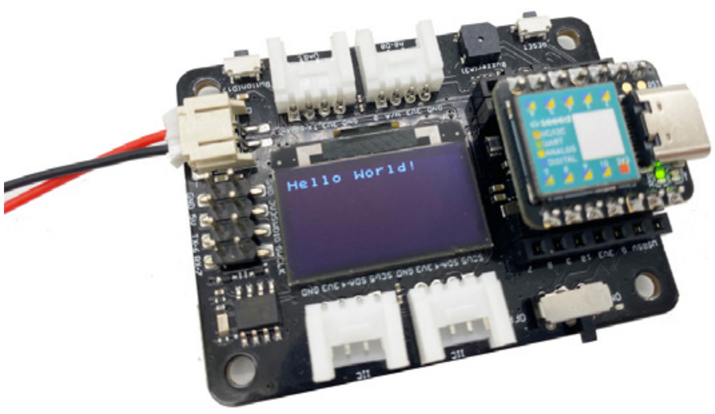
Please see details on entire program:

L6-HelloWorld.ino

Upload Program

After the program is written, we need to connect the XIAO main control board to the computer with the data wire. As shown in the following figure:





Click "Upload" to upload the program to the main control board. After the upload is completed, check whether "Hello World" is displayed on the OLED display .

Task 2: Draw a Circle On the OLED Display

Analysis

To draw a circle on OLED display screen, the U8g2 library should be used, and programming needs four steps:

• Declare use of the U8g2 library, choose SPI or I2C protocol, and set a constructor to connect OLED display screen

• The draw() function uses the u8g2.drawCircle function to draw a circle on the OLED

• Initialize U8g2 library

• In the loop() function, call the correlation function to draw an image on the OLED

Programming

Step 1:Declare use of the U8g2 library, choose SPI or I2C protocol, and set a constructor to connect OLED display screen.

```
#include<Arduino.h>
#include<U8g2lib.h>

//SPI or I2C
#ifdef U8X8_HAVE_HW_SPI
#include<SPI.h>
#endif
#ifdef U8X8_HAVE_HW_I2C
#include<Wire.h>
#endif

U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, /* reset=*/ U8X8_PIN_NONE);
//set constructors, define display types, controllers, RAM buffer sizes, and communication protocols
```

Step 2: The draw() function uses the u8g2.drawCircle function to draw a circle on the OLED, The meaning of each parameter is as follows:

• x0,y0:Position of center of circle

• rad:Defines the radius of the circle, the diameter of the circle is 2*rad+1

• opt:Select part or all of the circle

```
void draw(void) {
u8g2.drawCircle(20, 25, 10, U8G2_DRAW_ALL);//draw a full circle of diameter 21 with the coordinates (20, 25) as the center
}
```

Step 3: Initialize U8g2 library.

```
void setup(void) {
 u8g2.begin();//initialize the u8g2 library
}
```

Step 4: In the loop() function, call related functions to draw images on OLED,In the loop () function, call related functions to draw images on OLED. The firstPage and nextPage functions are used together to render pictures cyclically. As follows:

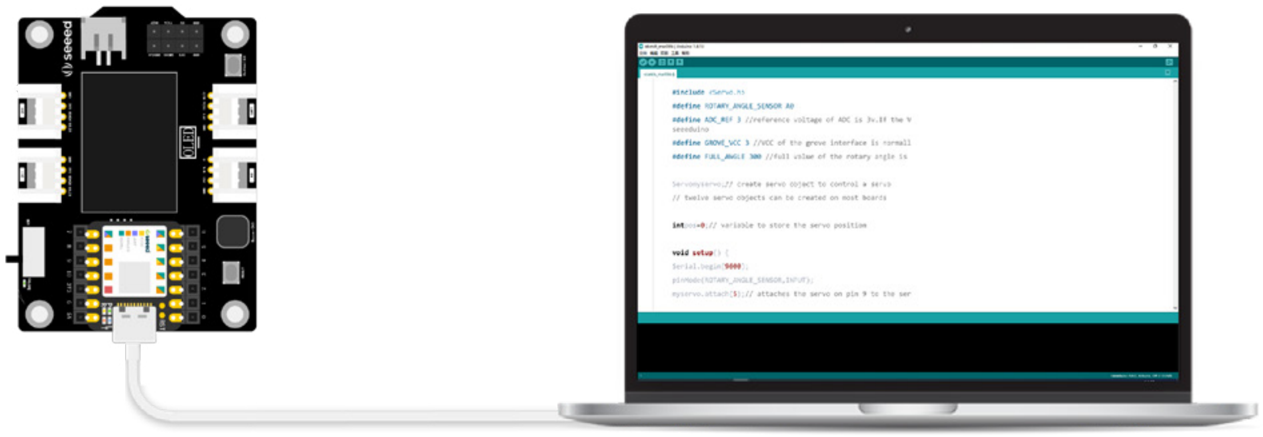
```
void loop(void) {
//picture cyclic display
 u8g2.firstPage();
 do {
     draw();//use the draw function
    } while( u8g2.nextPage() );

 delay(1000);
}
```

Please see details on entire program:

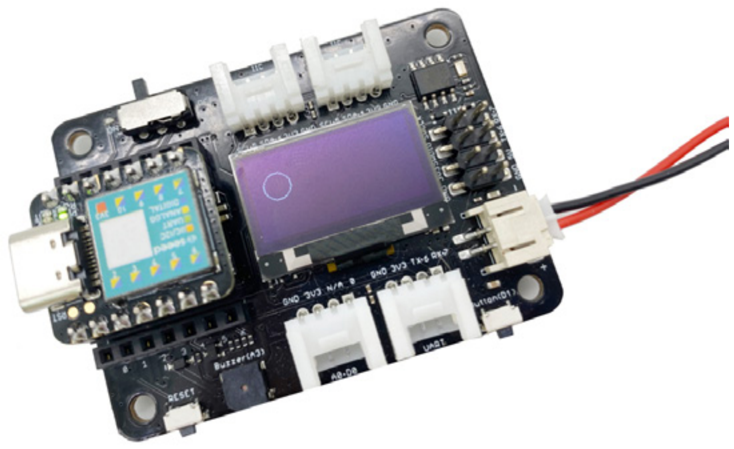
L6-DrawCircle.i
no

Upload Program

After the program is written, we need to connect the XIAO main control board to the computer with data wires.As shown in the following figure:



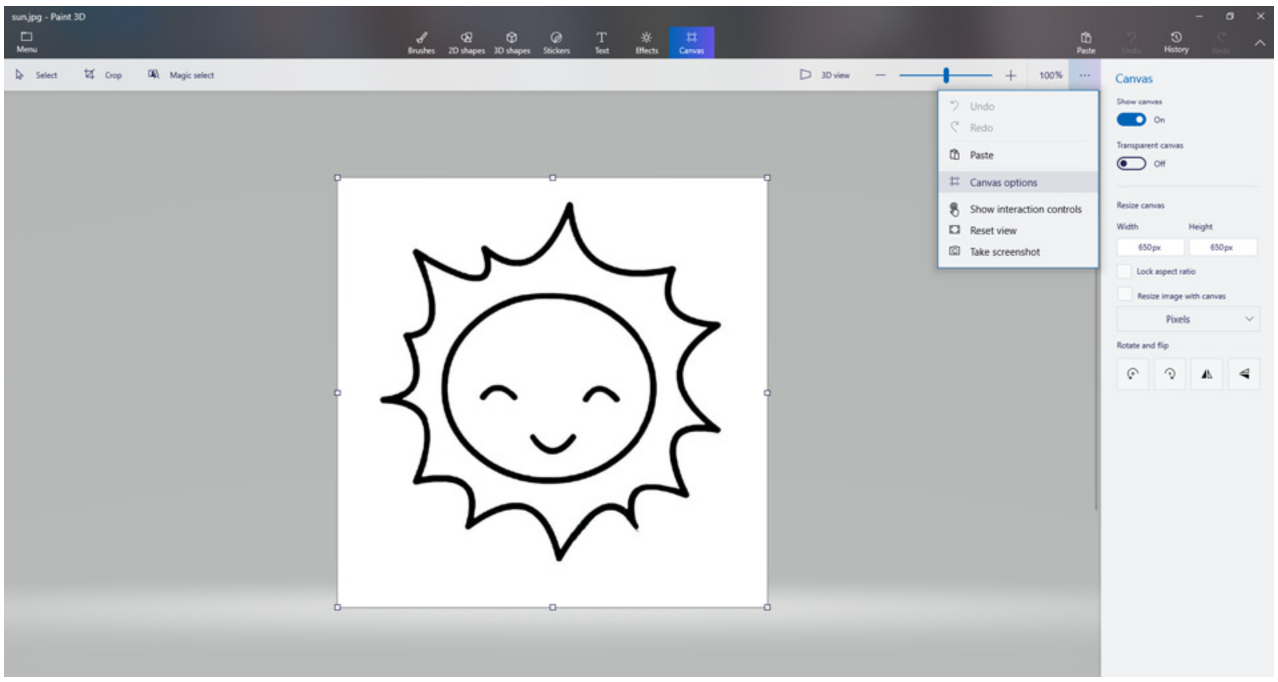
Next, click the “verification button” to verify the program.if the verification is correct, click the “Upload” button to upload the program to the hardware.When the debugging area shows “Upload Successful”,check whether there is a circular pattern on the OLED display.



Intensive Learning

OLED displays can also display external images, but the images need to be converted into two-dimensional arrays so that the machine can know where the pattern is located at each pixel on the display.Take the pattern of the sun as an example, let's try it together.

Step 1: Set the picture to 64×64 pixels, bmp grid, this is the maximum size of the image , because the OLED display we are using is 128×64 pixels. We can adjust the pixels through the drawing tools provided by the system, and then save them as bmp format.



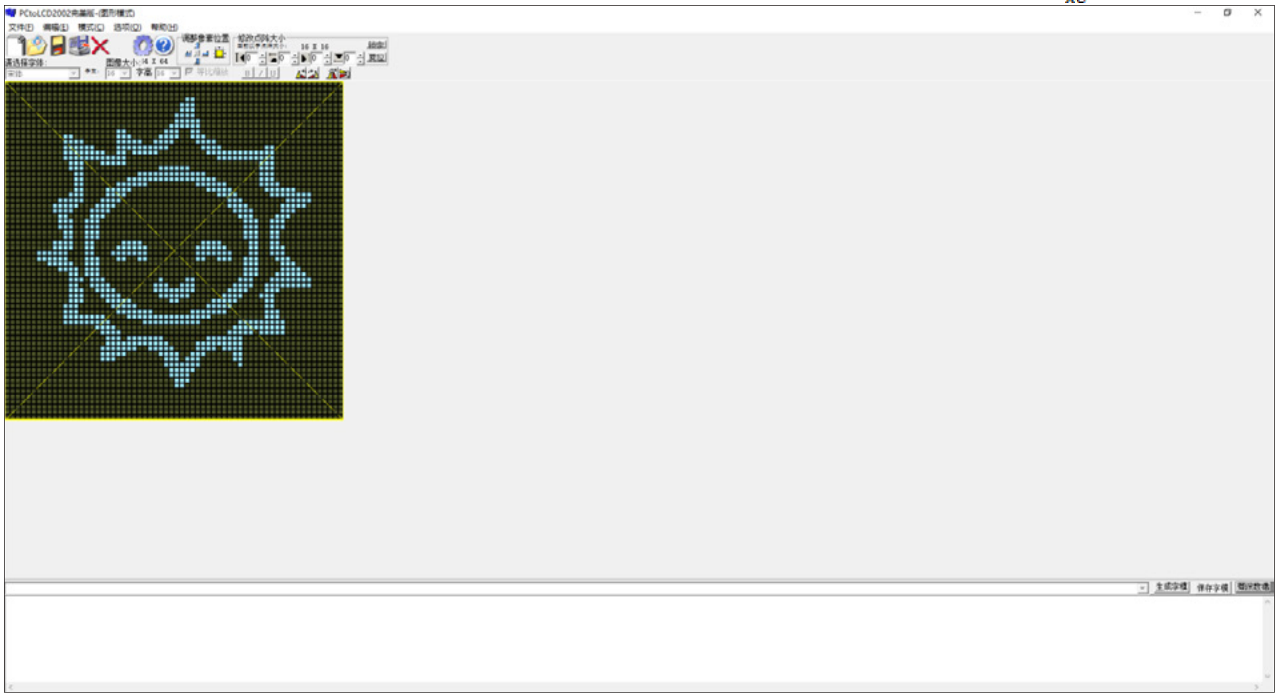
Show canvas: On
Transparent canvas: Off
Resize canvas
Width: 650px Height: 650px
Lock aspect ratio
Resize image with canvas
Pixels

If you are using an Apple computer or notebook, you can only complete the image format conversion through the image format converter or online image converter,such as this converter website: https://app.xunjiepdf.com/img2bmp/, After opening the website, we upload only the pictures we need and click Convert.

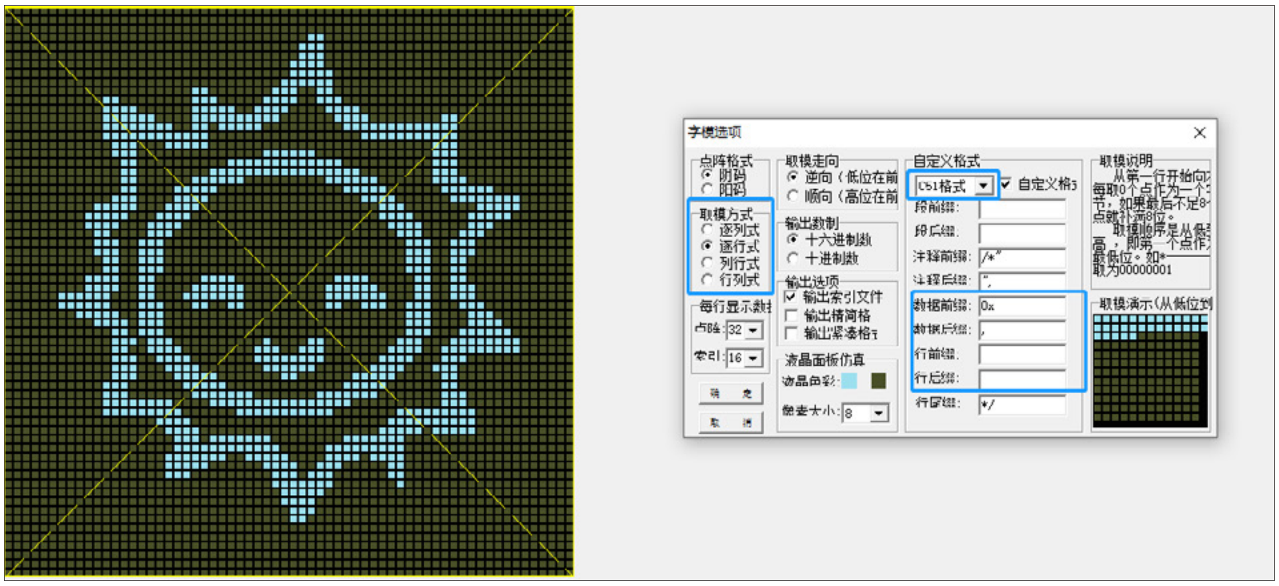
Step 2: Use the word conversion tool to convert the picture to a two-dimensional array.

Find the picture and open it by "File → Open".

PCtoLCD2002.exe



Next, set the font options, and select them as shown in the figure. Select line-by-line mode for modeling, customize the format, and select C51. When you are finished, click OK.



字模选项

点阵格式: [✓ 阴码] 阳码

取模方式: 逐列式 [✓ 逐行式] 列行式 行列式

每行显示数据 点阵: 32 索引: 16

取模走向: [✓ 逆向（低位在前] 顺向（高位在前

输出数制: [✓ 十六进制数] 十进制数

输出选项: [✓ 输出索引文件] 输出精简格 输出紧凑格式

液晶面板仿真 液晶色彩: 🟦 🟩 像素大小: 8

自定义格式: C51格式 [✓ 自定义格式]

段前缀:

段后缀:

注释前缀: /*"

注释后缀: ",

数据前缀: 0x

数据后缀: ,

行前缀:

行后缀:

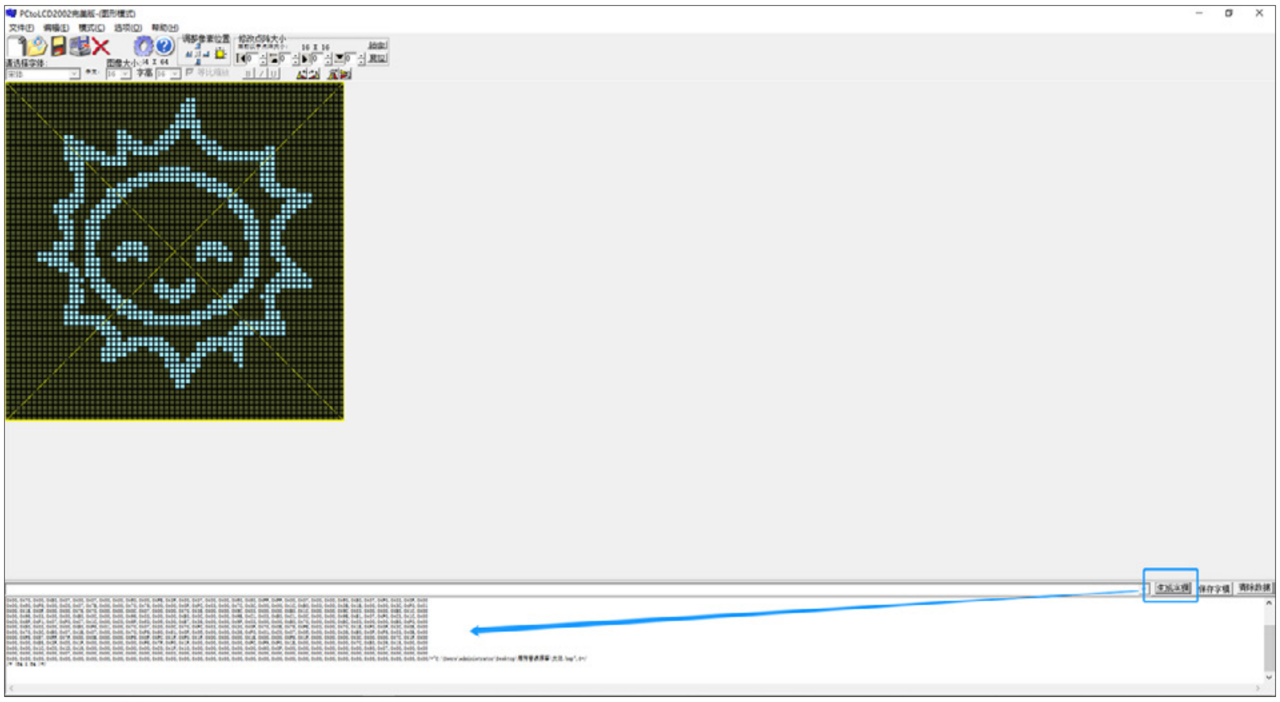
行尾缀: */

取模说明: 从第一行开始向 每取0个点作为一个字节，如果最后不足8个点就补满8位。 取模顺序是从低到高，即第一个点作为最低位。如*——— 取为00000001

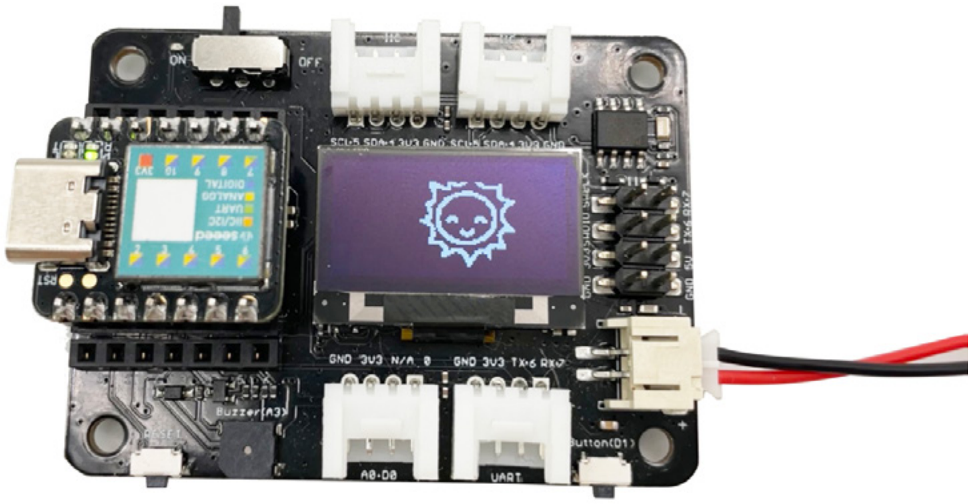
取模演示(从低位到

确定 取消

After clicking ok, click "generate font" at the bottom of the main interface, and we will copy all the arrays, which will be used in the program.

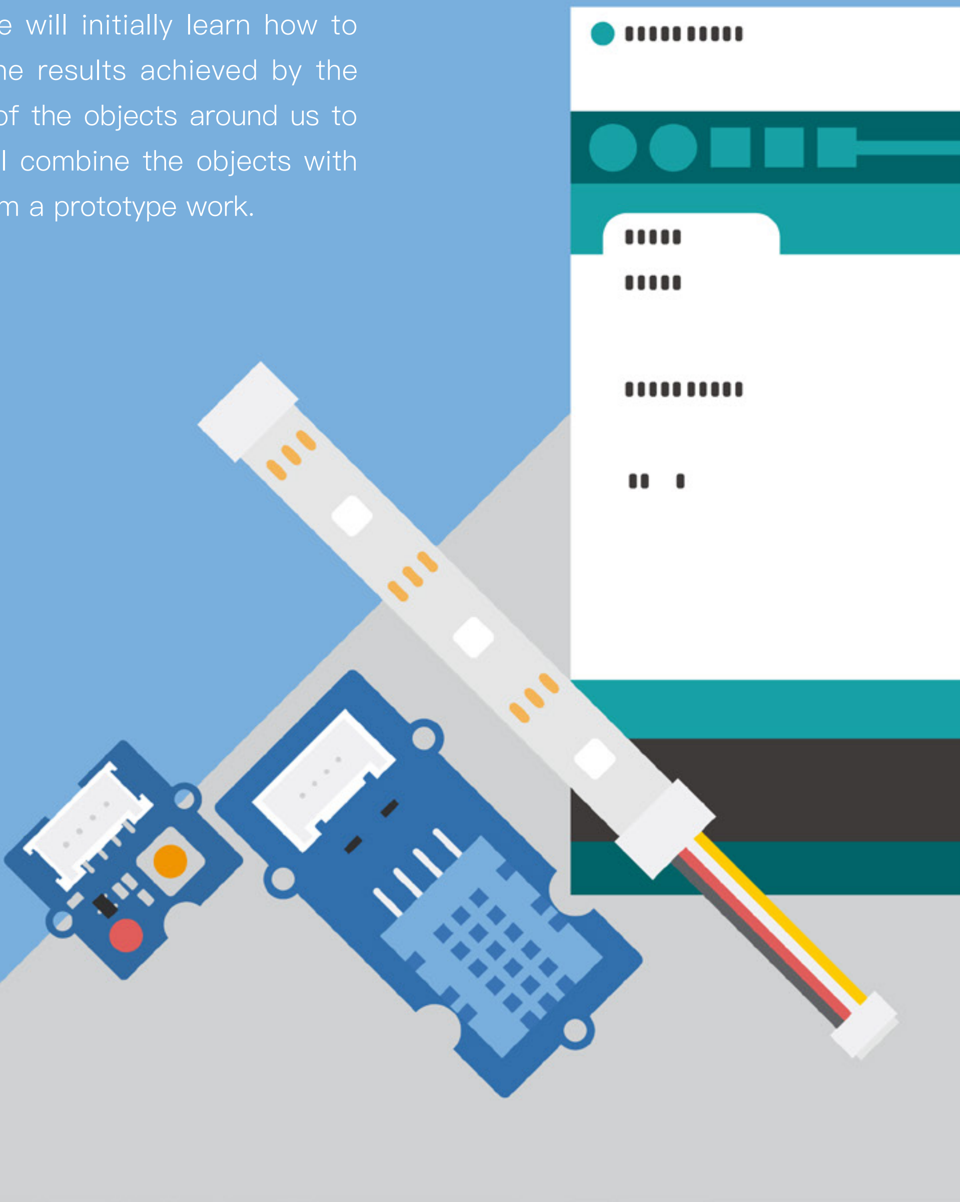


Master the picture into a two-dimensional array, we can let the OLED display a variety of pictures, but too many colors and complex pictures are not recommended, try to use stick figure. Next, open the program "L6-Sun.ino", and upload the program to XIAO's main control board, and the sun's pattern will be displayed on the OLED display.



Unit 2
Project Practice

In this unit, we will take several classic projects as examples to carry out project practice and learn how to create a prototype that can be quickly verified from an idea.In this unit, we will not analyze the code line by line, but only explain the key steps. We will learn more about the practical application of the code. Arduino has a very rich library and sample programs, as well as a large number of community resources. When we do projects, we should be good at finding resources, refer to sample programs, adjust the code according to our own needs, and achieve the desired results faster.In addition, we will initially learn how to design the appearance based on the results achieved by the program. We will start with the use of the objects around us to carry out the transformation. We will combine the objects with the electronic hardware to quickly form a prototype work.

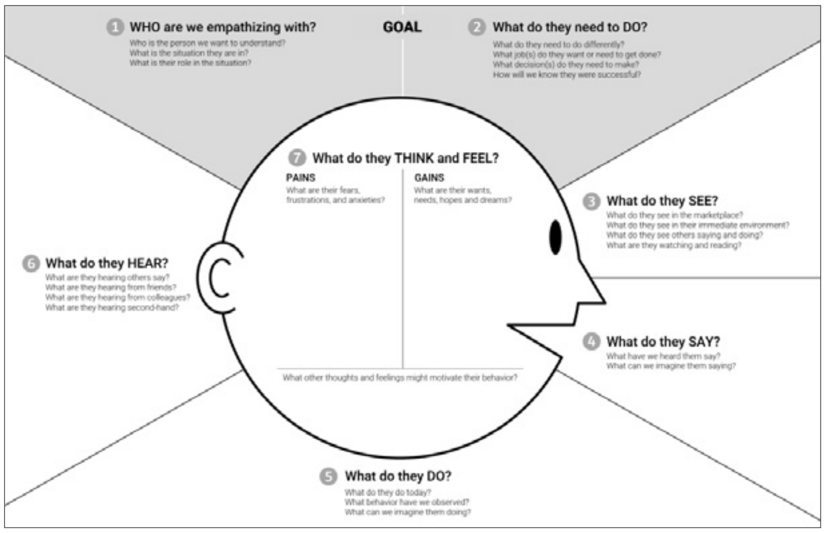


Lesson 7 Introduction to Product Prototype Design

In the first unit, we walked through the gates of electronic hardware and programming, and learned how to control electronic hardware through codes to achieve a desired effect, such as controlling LED lights, controlling buzzers, OLED display text and so on in various ways. This lesson will help us to realize an idea into a prototype, and then into a product. When you master the knowledge, you will have entered the world of product prototype design.

Discover Requirements

When we were young, although we wanted to give gifts in Mother's Day.We will make a greeting card with warm greetings or fold some lovely ornaments with colored paper according to our mother's preference.In the future, I plan to design an automatic sweeping robot and an automatic dish washing robot to relieve my mother's tiredness caused by housework.This is an unintentional product creator who inadvertently thinks about "the needs of users".Of course, "user demand" is not simple, so we should think more comprehensively and design products that really meet them.Here's a good way to do it–the empathy map. We can find the demand by the way of empathy map. In a nutshell, the empathy map intuitively depicts the users' thoughts, feelings, what they see, say, do and hear through the way of drawing, which helps us to think with the users in different scenes, open our minds, explore the users' deep motivation, and help us to discover the real needs.


1 WHO are we empathizing with?
Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?
GOAL
2 What do they need to DO?
What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?
7 What do they THINK and FEEL?
PAINS
What are their fears, frustrations, and anxieties?
GAINS
What are their wants, needs, hopes and dreams?
What other thoughts and feelings might motivate their behavior?
3 What do they SEE?
What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?
6 What do they HEAR?
What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?
4 What do they SAY?
What have we heard them say?
What can we imagine them saying?
5 What do they DO?
What do they do today?
What behavior have we observed?
What can we imagine them doing?

It is suggested that many people participate in using the empathy map to brainstorm. It is mainly divided into the following steps:

- Identify and refine the subject, age, education, health, income, occupation, etc.;
- Filling a plurality of modules in the diagram through observation or inquiry, and pasting the convenience notes in corresponding positions, wherein each convenience note only writes one thing or one content;
- Summarize and dig out similar or identical sticky notes, focus on controversial sticky notes,

analyze words, situations, actions and feelings, and sort out the key contents of each part;

• Check, replay, and refine. You can invite others to check your empathy map, make suggestions, and finally summarize the object's problems and identify the ones you want to solve most.

Empathy Game

Divide the participants into different groups. Each group determines an object, which can be "pet", "old man", "programmer", "police", "cleaner" and so on. Then focus on the problem through the empathy map and find out the method of demand. Fill in the form below.

What did she/he see/hear	Feelings	What did she/he say	Agonies	What did she/he want
Requirement definition				

Example:

Clear target: white-collar workers who lives alone in cities with pets at home, busy at work and often on business trips.

What did she/he see/hear	Feelings	What did she/he say	Agonies	What did she/he want
Because he worked overtime at night or forgot to feed the cat on time, his cat was starving all the time. When he got home, he found that the cat had eaten all the food and was sitting at the gate.	He felt very guilty when he saw that the cat was starving. Long term starvation and irregular diet will affect the cat's health	He wants the cat to eat on time and develop good eating habits, which is good for the cat's health	The cat's illness will make him feel guilty, and he needs to take a leave of absence to see a doctor. However, pet medical treatment is expensive, and the cat's constant barking will also affect the neighbors.	If only there was a machine that could remind cats regularly, it could feed them automatically even if they forgot to feed the cats or were not at home.
Requirement definition	Need a pet feeder that can remind feeding regularly and can feed automatically			

Forming a Product Plan

Product Definition

According to the final user's needs, it is clear what problems can be solved, what functions it has and how to realize it. For example, pet feeders, how to remind them? How to time? How to feed automatically?Reminder function is usually realized by sound and lighting.Because people don't stare at the machine all the time, it may be better to choose audible reminders.How to realize the function of automatic feeding by timing through the program?We can use the rotatable module and some structural design.In the end, our pet feeder is initially defined as an intelligent pet feeder that can sound to remind the owner to feed at a fixed time every day. If the owner is not at home, it can be automatically fed at a fixed time.

Functional Implementation

After the product definition is completed, relevant hardware needs to be prepared, which the two cores of the product can be realized that including timed reminder feeding and automatic feeding.XIAO+ expansion board is a good choice. There is a passive buzzer on the expansion board, which can meet the demand for audible reminder. We can control it to emit music through the program and solve various demands for time through the RTC clock module.This is also a common and low-cost solution.Regarding the function of automatic feeding, food can flow out automatically through the rotation and angle setting of the servo module and the design of some structures.Then set a suitable time for the servo to return to its original position and the feeding will be finished,If we want to control the pet's appetite, we can add a servo to XIAO and the expansion board to solve this demand.The next stage is software programming, the angle of rotation of the servo, the notes emitted by the buzzer, and how to time, all need to be debugged, which will not be described in more detail here.The complete prototype product is based on realization of function and design of appearance.

Product Appearance

In the design of product appearance, three aspects need to be conceived:

• Material and tech knowledge

It can be corrugated paper made by hand, wood structure carved by laser cutting machine, or 3D printing. Of course, combination of various materials is also possible.

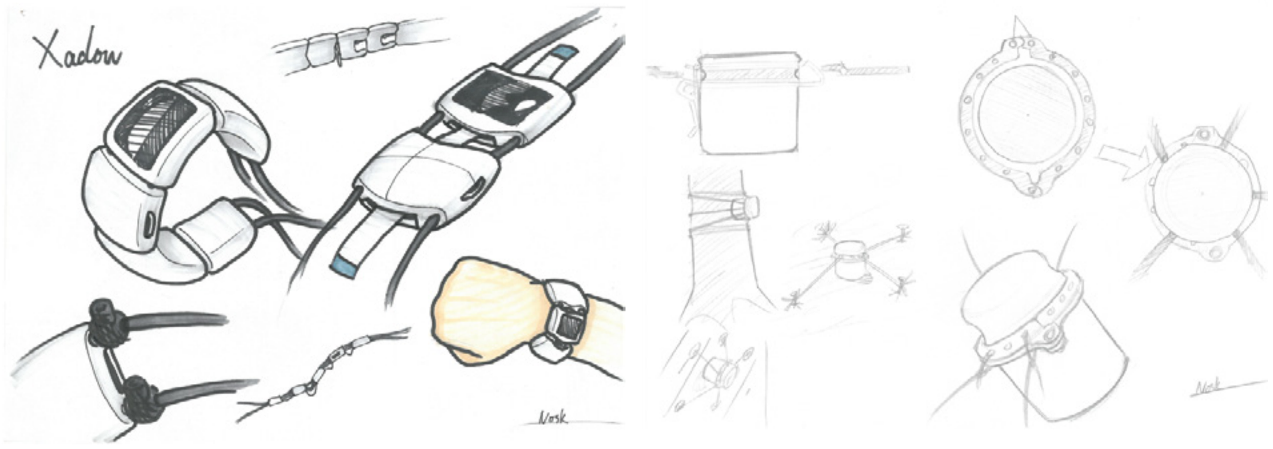
• Color matching

The reasonable application of color can not only beautify the work, but also convey the theme of the work through vision.

• Product form

Product form should consider the user's physical condition, living habits, psychological characteristics, etc.

In design, hand-drawing is a good choice. As shown in the figure, even if you are not good at drawing, you can draw the outline of the work and mark the key parts.



Prototyping, Testing and Optimizing

After the product scheme is determined, we will enter the prototyping stage,which the process may not be a linear process, we need to re–scrutiny in the process or may modify the scheme on account of inexperienced prototyping.The intelligent pet feeder needs to continuously adjust the rotation angle and duration of the servo and the structure for controlling the grain falling by combining with the program.After the prototype is completed, it will enter the stage of testing and optimization, check whether the product meets the expectations and confirm where it can be optimized and improved.Every product needs to be tested by the market, optimized with feedback and innovated.Of course, we only introduced how to make a prototype work personally. If you want to put it on the market, there are still many aspects to consider, and it is not something that an individual can accomplish, but requires the cooperation and efforts of a team.

At the end of this lesson, please try to complete a simple product plan based on the results of the empathy map mini–game.

Product name	
Product functions	
Hardware equipment	
Materials	
Appearance draft	
Presentation	

Lesson 8 Intelligent Temperature and Humidity Meter

Temperature and Humidity Meter can be seen everywhere in life. It can measure the temperature and humidity in the environment in real time.Just like the commonly used thermometer, when you feel unwell and confirm signs of fever, you will definitely use it.The invention of temperature and humidity meter brings great convenience to our life,In this lesson, we will make an intelligent temperature and humidity meter by using temperature and humidity sensors. Do you know what a temperature and humidity sensor is and what its function is?

Background Knowledge

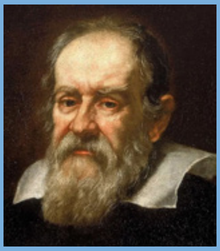
Temperature and Humidity Sensor

The temperature and humidity sensor is used to detect the temperature and humidity in the environment.There are many types of temperature and humidity sensors.We choose DHT20 which has a clear advantage in supply voltage, measurement range of temperature and humidity, precision and quality of output signal. It is a product with low power consumption, high precision and high stability, equipped with a fully calibrated digital I2C interface.

·Note·

As early as 1592, Galileo invented the thermometer, which was a thermometer made of a glass cylinder with transparent liquid and heavy objects of different densities.The principle of temperature measurement is simply to expand with heat and contract with cold. When the temperature changes, the density of the liquid will also change, causing suspended solids with different densities to move up and down until the liquid and itself have the same density. After optimization and iteration by different people, the glass thermometer is more and more accurate and stable, and is applied to various fields. The traditional glass thermometer has a history of more than 300 years and is still in use today.

With the development and application of technology, there are more and more selections for types of thermometers,such as mercury glass thermometers, dial–type thermometers, thermistor thermometers, infrared thermometers, electronic thermometers, etc. Different thermometers have different use scenarios. Some thermometers will add the function of detecting humidity and become humidistats to monitor the temperature and humidity of the environment.



Reading Temperature and Humidity Values in a Serial Monitor

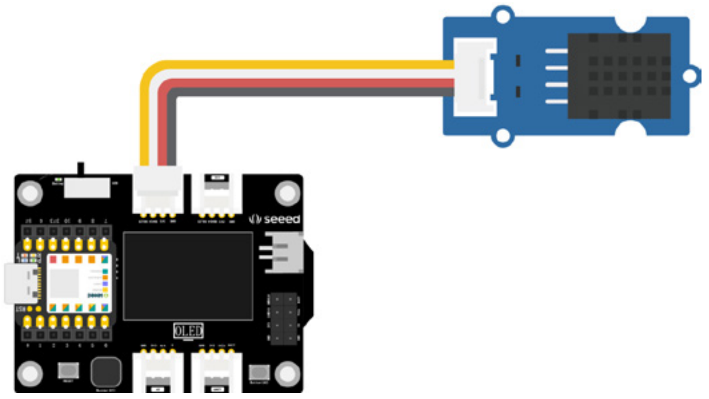
When using the sensor, we can use the DHT library to open the“DHTtester”example through the following path: **File → Example → Temperature and humidity sensor → DHT tester.** after opening the example program, we can see the following program.The program can read the temperature and relative humidity information in the environment and display real–time data on the serial monitor. Some code for the sample program needs to be modified:

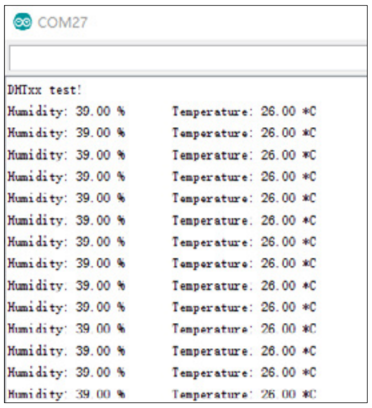
#define DHTTYPE DHT 20 , we need to modify the parameters according to the pin number of the actual connection of the temperature and humidity sensor.

DHT dht(DHTTYPE) , The DHT20 is different from other DHT* sensor ,it uses I2C interface rather than one wire, So it doesn’t require a pin.

```
#include ”DHT.h”
#define DHTTYPE DHT20   // DHT 20
DHT dht(DHTTYPE);

#if defined(ARDUINO_ARCH_AVR)
    #define debug  Serial

#elif defined(ARDUINO_ARCH_SAMD) ||  defined(ARDUINO_ARCH_SAM)
    #define debug  SerialUSB
#else
    #define debug  Serial
#endif

void setup() {

    debug.begin(115200);
    debug.println(”DHTxx test!”);
    Wire.begin();
    dht.begin();
}

void loop() {
        float temp_hum_val[2] = {0};
        if (!dht.readTempAndHumidity(temp_hum_val)) {
            debug.print(”Humidity: ”);
            debug.print(temp_hum_val[0]);
            debug.print(” %\t”);
            debug.print(”Temperature: ”);
            debug.print(temp_hum_val[1]);
            debug.println(” *C”);
        } else {
            debug.println(”Failed to get temprature and humidity value.”);
        }

        delay(1500);
    }
```

After modifying the code, connect the temperature and humidity sensor to I2C interface, upload the sample program to XIAO, and open the serial monitor, you can see the values of temperature and humidity. You can put the temperature and humidity sensors in different environments to observe whether the temperature and humidity values will change.





```
DHTxx test!
Humidity: 39.00 %       Temperature: 26.00 *C
Humidity: 39.00 %       Temperature: 26.00 *C
Humidity: 39.00 %       Temperature: 26.00 *C
Humidity: 39.00 %       Temperature: 26.00 *C
Humidity: 39.00 %       Temperature: 26.00 *C
Humidity: 39.00 %       Temperature: 26.00 *C
Humidity: 39.00 %       Temperature: 26.00 *C
Humidity: 39.00 %       Temperature: 26.00 *C
Humidity: 39.00 %       Temperature: 26.00 *C
Humidity: 39.00 %       Temperature: 26.00 *C
Humidity: 39.00 %       Temperature: 26.00 *C
Humidity: 39.00 %       Temperature: 26.00 *C
Humidity: 39.00 %       Temperature: 26.00 *C
Humidity: 39.00 %       Temperature: 26.00 *C
```

Project Making

Project Description

We will make a portable small-scale temperature and humidity detector to detect the temperature and humidity value through the temperature and humidity sensor and display the value on the OLED display screen of XIAO expansion board. We can further add the buzzer alarm function when the detected temperature and humidity exceed a certain value range, an alarm will be sounded to remind.The numerical range can be adjusted according to different application scenarios, such as home life scenarios, and the comfortable temperature and humidity numerical range can be set based on human body sensation; For example, in places where plants are planted, the numerical range of temperature and humidity is set on the basis of suitable plant growth, and once it exceeds the numerical range, an alarm can be generated to remind people to make adjustments.

Write Program

With reference to the above example program, one of the effects we want to achieve is to display the temperature and humidity values on the OLED display screen, but with a different display medium. The code for reading the temperature and humidity sensor detection values can be reused.Combined with lesson 6, we learned how to display characters with an OLED, so we only need to add an “if……else” conditional judgment statement to judge the temperature and humidity values.The programming ideas are as follows:

• Declare the DHT library, U8x8 library, etc. to be called, and connect the buzzer pin as a sounding device

• Initialize the library, define the buzzer pin state

• Define the temperature and humidity variables as readings, and display them on the OLED screen, add logical judgment, and realize buzzer alarm

Program implementation is divided into two tasks:

Task 1: Detect Temperature and Humidity and Display on OLED Display

Step 1: Header file, declaring the library file to be called.

```
#include “DHT.h”
#include <Arduino.h>
#include <U8x8lib.h>// here we use the U8x8lib.h library.
#define DHTTYPE DHT20
DHT dht(DHTTYPE);

U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(/* reset=*/ U8X8_PIN_NONE);//Set the constructor to connect the OLED display
```

Step 2: Initialize the library file of DHT and u8x8.

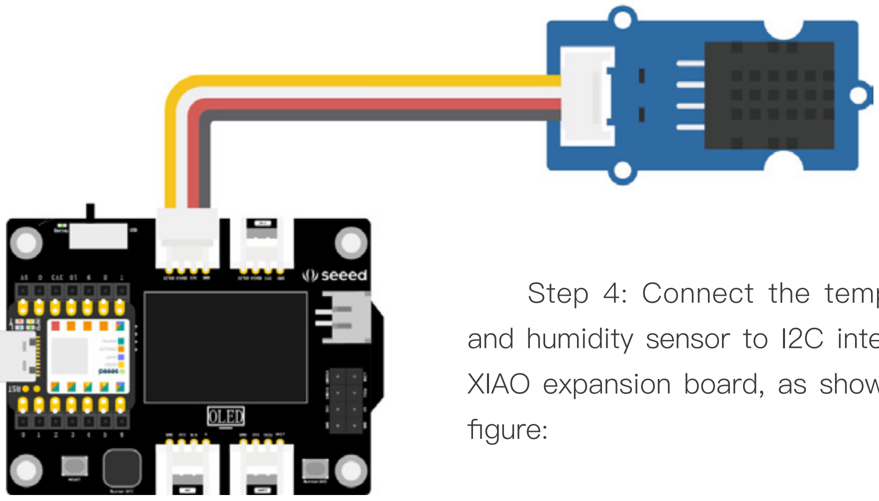
```
void setup() {
  Wire.begin();
  dht.begin();//DHT begins
  u8x8.begin();//u8x8 begins
   u8x8.setPowerSave(0);  //power-saving mode is turned off, 1 is turned on, and nothing will be seen on the screen after power-saving mode is turned on.
  u8x8.setFlipMode(1);//Flip the display 180 degrees, 0 is disabled, 1 is enabled
}
```

Step 3: Define the temperature and humidity variables as readings, and display them on the OLED screen, pay attention to the coordinate position for the display of temperature and humidity.

```
 void loop() {
    float temp, humi;//defines variables to store readings.
    temp = dht.readTemperature();//read the temperature value and store it in temp
    humi = dht.readHumidity();//read the humidity value and store it in the humi

    u8x8.setFont(u8x8_font_chroma48medium8_r);//set display font
    u8x8.setCursor(0, 33);//sets the position of the draw cursor (0, 33)
    u8x8.print("Temp:");//show temp at (0, 33)
    u8x8.print(temp);//display real-time temperature values
    u8x8.print("C");//displays the unit "c" of temperature
    u8x8.setCursor(0,50);
    u8x8.print("Humidity:");
    u8x8.print(humi);
    u8x8.print("%");
    u8x8.refreshDisplay();
    delay(200);
  }
```

Please see details on entire program:

L8-Temperature_Humidity.ino



Step 4: Connect the temperature and humidity sensor to I2C interface of XIAO expansion board, as shown in the figure:

Connect XIAO to the computer with a data cable, click the "Upload" button to upload the program to the hardware. When "Done uploading!" is displayed in the debugging area, observe whether the temperature and humidity values and numerical changes are displayed on the OLED screen, and hold the blue resistance part of the sensor with the palm of your hand.

Task 2: Add Alarm Function

Step 1: Add code,the function of the alarm can be realized by connecting a buzzer to the circuit.The XIAO expansion board has an onboard buzzer, which we can use directly.In the program, the pin mode of buzzer pin needs to be set, and the conditional statement is added. When the temperature is greater than a certain value or the humidity is less than a certain value, the buzzer will sound, which needs to form a logic expression by "& & "logic.

·Note·

Logical Operator

&&:stands for “and ”, if (expression 1 && expression 2), and the statement in if () is executed only when all expressions in parentheses are true

| |: stands for or, if (expression 1 || expression 2), satisfies one of the expressions, evaluates to true, and executes the statement in if {}.

Usage:

When the temperature is greater than 30 or the humidity is less than 40, the buzzer shall sound an alarm.

```
if (temp > 30 || humi < 40) {
tone(buzzerPin, 200, 200);
}
```

The mainly for setting the buzzer and judging the temperature and humidity, which controls the buzzer to sound:

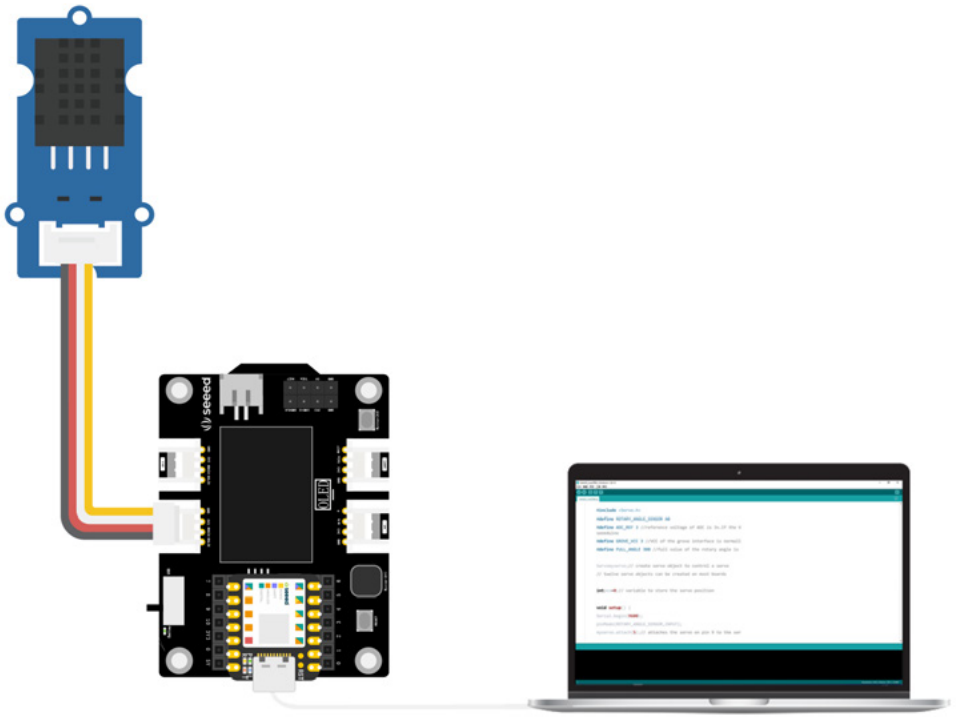
```
//Port programs do not run
int buzzerPin = A3;//the number of buzzer pin

void setup() {
  pinMode(buzzerPin , OUTPUT);//initialize the buzzer pin as an output
}

void loop() {
  float temp, humi;//defines variables to store readings
  temp = dht.readTemperature();
  humi = dht.readHumidity();
if (temp > 30 || humi < 40) { //when the temperature is greater than 30 or the humidity is less than 40, the buzzer shall sound an alarm if one of the conditions is met
   tone(buzzerPin, 200, 200);
}
```

Add the above code to the corresponding position of task one program to realize all functions. Please see details on entire program:

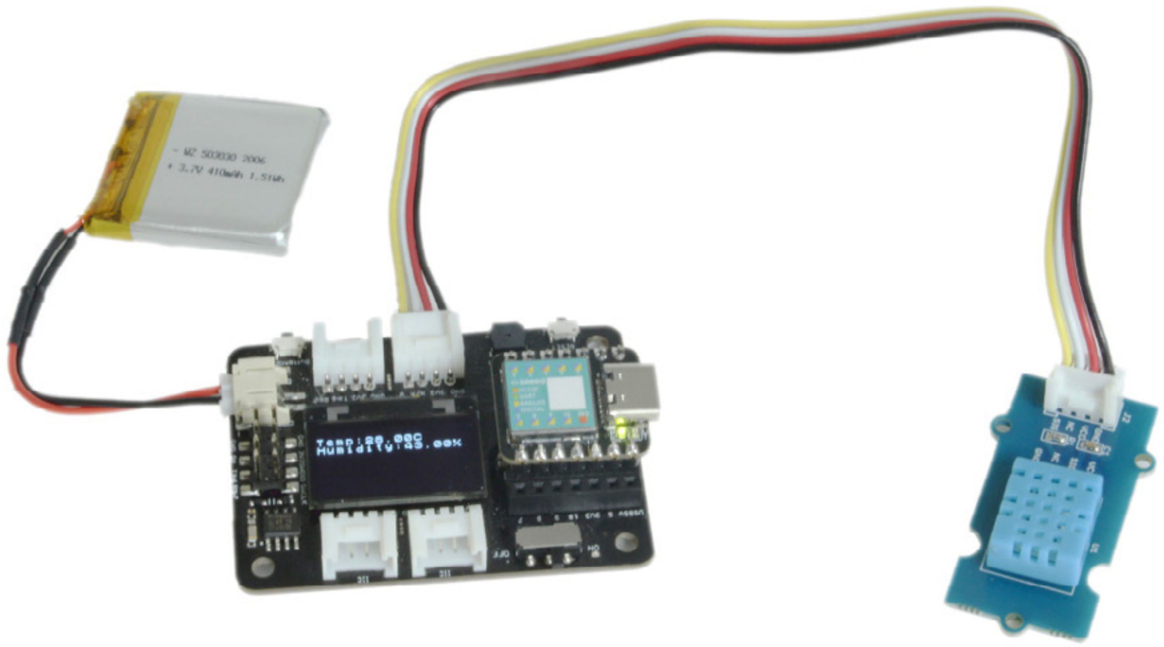
L8-Temperature_Humidity2.ino

Step 2: Upload program

After the program is written,we will XIAO main control board with data cable connected to the computer, as shown in the figure below:



After connection, click “Verify” to verify the program. If verification is correct, click “Upload” button to upload the program to hardware. When “Done uploading” is displayed in debugging area.In order to verify whether the alarm function runs smoothly, we can hold the temperature and humidity sensor tightly with our palms, and observe the numerical change of the OLED display. Listen for the buzzer when the temperature goes over 30 degrees!



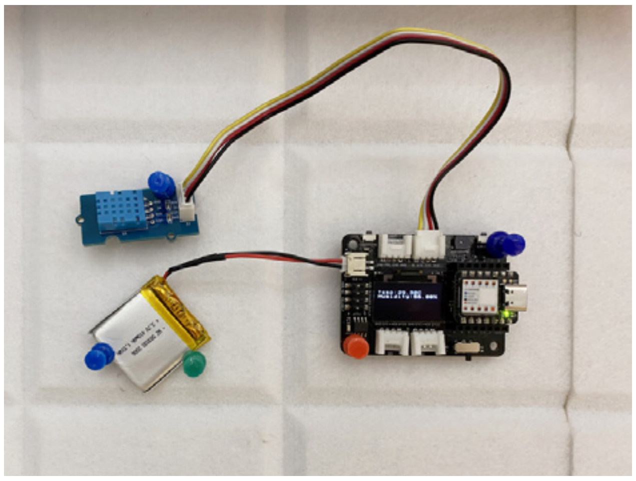
Note: We can see the characters on the OLED display screen constantly beating in the video, because the OLED display screen and the camera have different refresh rates. This has no effect on naked eye observation.

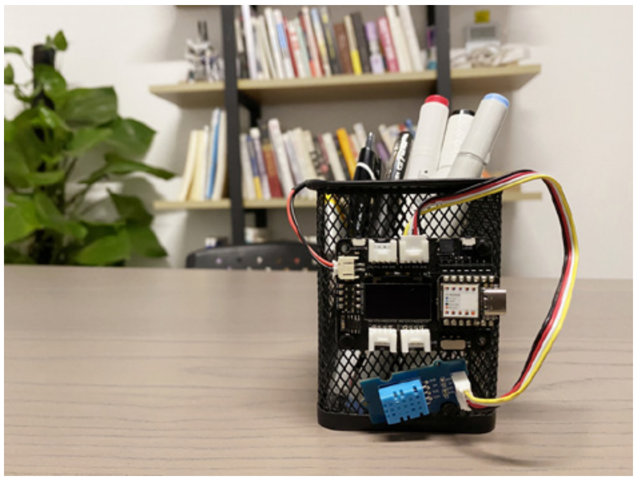
⭐ Appearance Design

From this class, we will try to explore the appearance and make a complete prototype product.First, try to draw the design drawings and make a simple transformation with the existing materials. Then with reference to the intelligent temperature and humidity meter, please design the appearance of the prototype according to the product features and functions.

Product name	Intelligent temperature and humidity meter
Product characteristics	Small, portable, highly sensitive
Product characteristics	Temperature and humidity values are displayed in real time and an alarm sounds when the temperature and humidity values are outside the comfort range
Product function	(For example: make it into a pendant to carry on the backpack, stick on the paper towel storage box in the bedroom, etc.)

References:





Lesson 9 Surprise Gift Box

On your friend's birthday, do you want to give her/him a special birthday present without spending money? It can be realized through our existing modules.In this lesson, we are going to make a surprise gift box for our friends. What kind of surprise will appear when we open the box? What kind of modules do you need to complete such a surprise gift box? Let's start with questions.

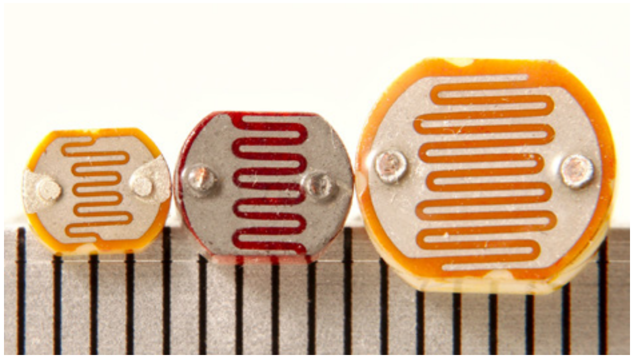
Background Knowledge

Light Sensor

A light sensor detects the intensity of light in the surrounding environment and converts the detected light energy into electrical energy.Light sensors are classified into photoresistance types, photodiode types and phototransistor types.Photoresistance and photodiode types are common in light sensors and will be briefly introduced in the lesson.

Photoresistance Type

A photoresistor is used in a photoresistor light sensor. As shown in the figure below, a photoresistor is a material whose electrical resistance changes with the intensity of light that it is exposed to. High-intensity light reduces its resistance value, while low-intensity light increases the resistance value.The resistance value of the photoresistor then affects the voltage across other elements in the circuit, such as an LED. In this way, the light intensity of the LED can be controlled.



Photodiode Type

Photodiodes are also called photoelectric sensors and photodetectors. When a beam of light hits the diode, the electrons in the tube will quickly disperse to form electron holes, which will lead to greater electrical conductivity, allowing current to pass through.The stronger the light, the more holes are created and the stronger the current. Because the current generated by the photodiode is proportional to the intensity of the light, it is very useful for light detection to change lighting response quickly. The light sensor we are going to use in this lesson is photodiode type.

We can construct the light control switch through the light sensor, for example, control the lights on and off, turn off the lights during the daytime, and turn on the lights at night. The main purpose of light control equipment is to save energy, automatically improve efficiency, light control lamp is the best example, such as light control desk lamp, light control street lamp, road tunnel lighting, etc., which brings convenience to our life at the same time also contribute to environmental protection and energy saving.

RGB LED Strip

Light sensor and RGB LED strips are used together because the light belt is integrated with multiple colored beads. Compared to a single LED, it can achieve more lighting effects and is very suitable for surprise.RGB LED lamps are available in several styles and models, we need WS2813, 30 lamp bead model.The program controls the RGB LED strip to achieve rich lighting effects.Let's learn about its library– NeoPixel together.

Open the "simple" example by following the path: **File → Example → Adafruit NeoPixel → simple**. After opening the sample program, we can see the program as shown below. The program implements that the light strip illuminates 30 beads in turn (green light). This is a simple light strip example and we need to modify some parameters:

#define PIN 0, we need to modify the pin of the light band connection according to the actual situation, and connect to the A0 interface of XIAO expansion board, so it is PIN 0.

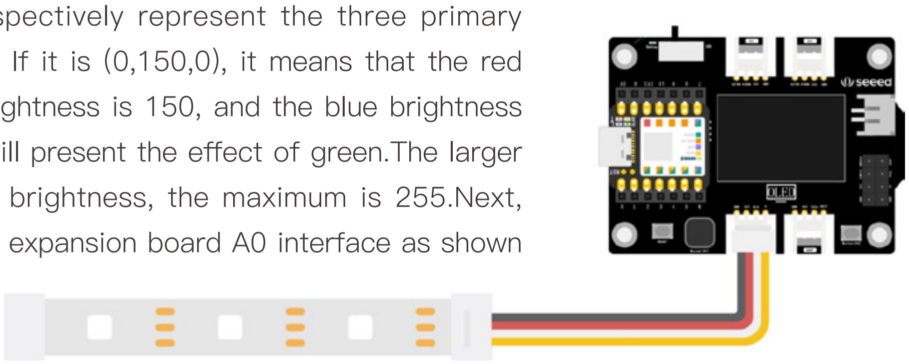
#define NUMPIXELS 30, defines the number of LED lights on the lamp strip, because the number of integrated beads varies with the type of lamp. We use a 30-bead strip, so it is NUMPIXELS 30.

After modifying the parameters, in order to see the code more clearly, we can delete the comment, which occupies a very large space.

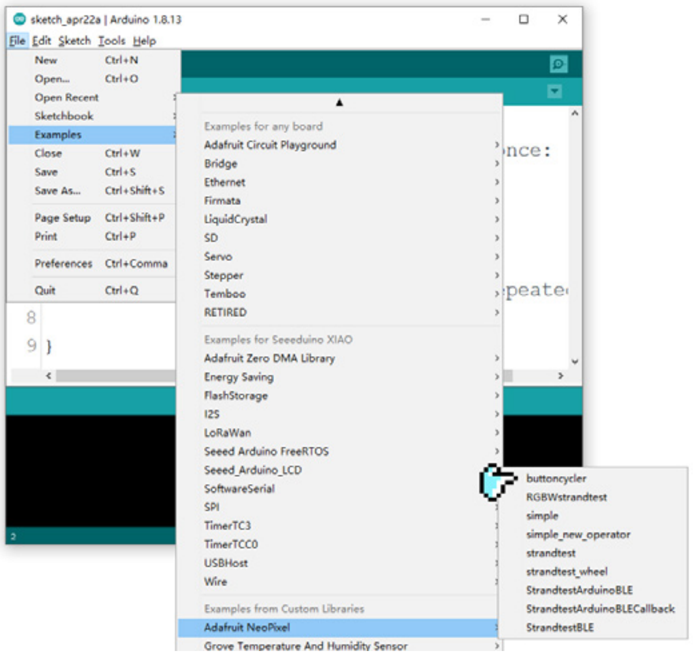
```
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
 #include <avr/power.h>
#endif

#define PIN 0 //light strip connect to Pin A0
#define NUMPIXELS 30 //number of Led on the light strip
Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);//create a new light strip object to define the data pattern
#define DELAYVAL 500 //time (in milliseconds) to pause between pixels

void setup() {
#if defined(__AVR_ATtiny85__) && (F_CPU == 16000000)
  clock_prescale_set(clock_div_1);
```

```
#endif
  pixels.begin();//intialize NeoPixel strip object

void loop() {
  pixels.clear(); //Set all pixel colors to 'off'
//The first NeoPixel in a strand is #0, second is 1, all the way up to the count of pixels minus one.
  for(int i=0; i<NUMPIXELS; i++) {
   pixels.setPixelColor(i, pixels.Color(0, 150, 0));//here we're using a moderately bright green color
  pixels.show(); //send the updated pixel colors to the hardware.
    delay(DELAYVAL); //pause before next pass through loop
  }
}
```

In the above code, pixels.Color(0,150,0) is a function of setting the LED lamp color, the numbers in parentheses respectively represent the three primary colors (red, green, and blue). If it is (0,150,0), it means that the red brightness is 0, the green brightness is 150, and the blue brightness is 0. The whole lamp strips will present the effect of green.The larger the number, the greater the brightness, the maximum is 255.Next, connect the strip to the XIAO expansion board A0 interface as shown in the following figure:



Connect the XIAO main board to the computer with a data cable and upload the program to the motherboard. After successful upload, observe the effect of the lamp strips. The light strips can be toned, blinking, breathing and other light effects, we can refer to the library of sample programs. For example, buttoncycler, in the example program, buttons are used to switch the lamp strips to different lamp effects, and we can find codes for various lamp effects in the programs, such as flashing, rainbow lamp, and chasing.



Project Making

Project Description:

Surprise gift box program steps: the light sensor can control the RGB LED strip on and off, just like the light control lamp, but the opposite effect will appear, when the value detected by the light sensor is less than a fixed value, which will turn off RGB LED in a dim environment , when the value detected by the light sensor is greater than a fixed value, which will light up RGB LED strip with rainbow color in a bright environment.

Write Program

The programming ideas are as follows:

• Declare the files that will be used, create a new light strip object, and define the number of sensor pins and light strip LED lights

• Initialize the light strip and set the light sensor pin mode

• Read the light value, if the light value is greater than 100, the led strip shows rainbow and breathing light effect, otherwise the led strip go out

The program is divided into two tasks:

Task 1: Achieve Rainbow and Breathing Light Effect in Light Strip

Step1： Declare the files that will be used, create a new light strip object, and define the number of sensor pins and light strip LED lights.

```
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
#include <avr/power.h>
#endif

#define PIXEL_PIN 0 //RGB LED strip connect to Pin 0
#define PIXEL_COUNT 30 //number of NeoPixels
Adafruit_NeoPixel strip(PIXEL_COUNT, PIXEL_PIN, NEO_GRB + NEO_KHZ800);
//declare our NeoPixel strip object:
```

Step 2: Initialize the light strip.

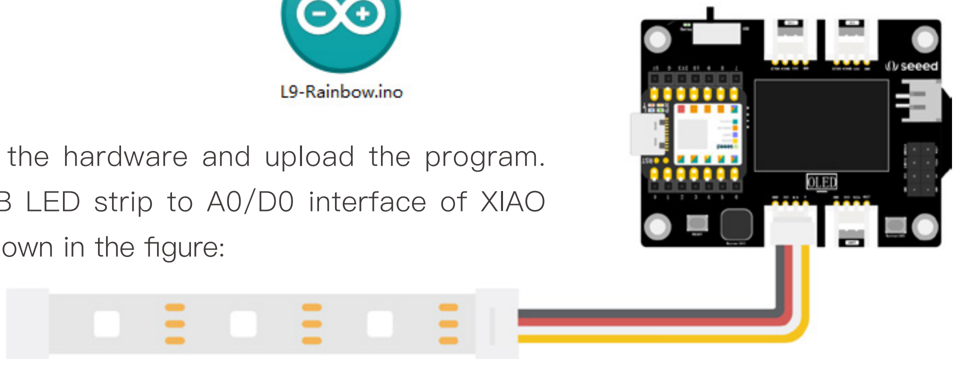
```
void setup() {
  strip.begin(); //initialize NeoPixel strip object
}
```

Step 3: The light strip now shows the rainbow and breathing light effects.This part uses the for() loop to present the effect of breathing. For example for(i = 0; i<5; I++{} indicates that the initial value I is 0. When i is less than 5, the statement in the loop body {} is run. Each loop is made up of 1, and the loop is made up of 5 times.

```
void loop() {
  strip.clear();//set all pixels in RAM to 0 (off)
   rainbow(10);//The light strip show the rainbow light effect, the number in brackets represents the speed of the rainbow light flow, when the number is smaller, the flow speed is faster
}
//The following is the rainbow light effect code,
//rendering breathing light effect, the code can be found in the example program buttoncycler
void rainbow(int wait) {
  for(long firstPixelHue = 0; firstPixelHue < 3*65536; firstPixelHue += 256) {
    for(int i=0; i<strip.numPixels(); i++) {
      int pixelHue = firstPixelHue + (i * 65536L / strip.numPixels());
      strip.setPixelColor(i, strip.gamma32(strip.ColorHSV(pixelHue)));
    }
    strip.show(); //update strip with new contents
    delay(wait);  //pause for a moment
  }
}
```

Please see details on entire program: L9-Rainbow.ino

Step 4: Connect the hardware and upload the program. First connect the RGB LED strip to A0/D0 interface of XIAO expansion board, as shown in the figure:



Connect XIAO to the computer with the data cable, and click the “Upload” button to upload the program to the hardware. when “Done uploading” is displayed in the debugging area, observe the lighting effect of the light strip.

Task 2: Add Optical Switch Function

Step 1： add code. Read the light value detected by the light sensor, and judge the light value with the statement “If ... else ...” When it is greater than 100 (the value can be adjusted according to the actual environment), the RGB LED strip shows the rainbow breathing light effect.

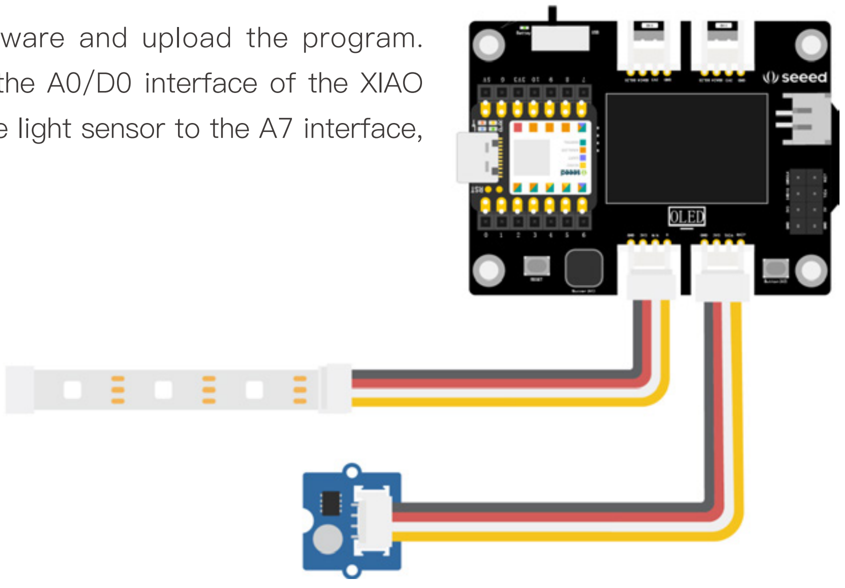
Added program part:

```
//partial program, do not run
 #define LIGHT_PIN 7//the number of light pin
 int readValue = 0;//define the variable readValue and store the light value
void setup() {
   pinMode(LIGHT_PIN , INPUT); //set the pin of the light sensor as the input state
}
void loop() {
     readValue = analogRead(A7);//read the ray analog value at pin A7 and stores it in the readValue variable.
     if(readValue > 100){ //judging by the condition, if the light value is greater than 100, the rgb led strip exhibits rainbow lamp effect; otherwise, the lamp strip is extinguished
     rainbow(10);
      }else {
      strip.clear();
      strip.show();
}
}
```

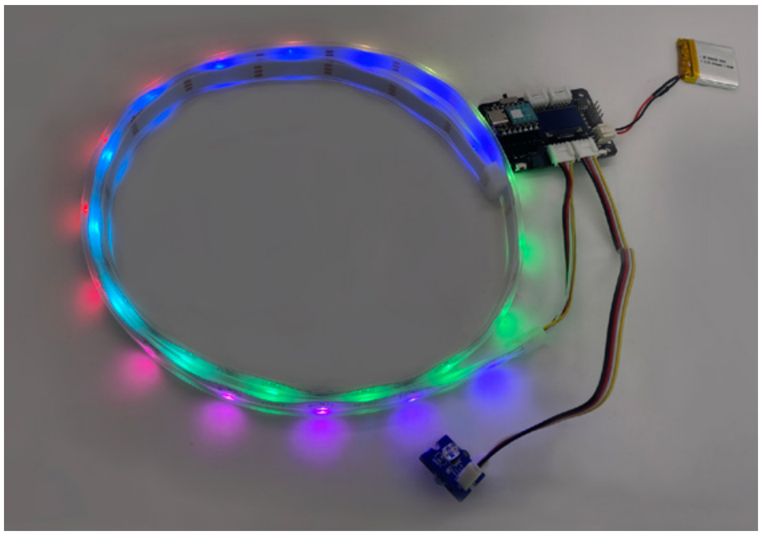
We place the added statement in the corresponding position of the task-one program,Please see details on entire program.

L9-StripLight.ino

Step 2: Connect the hardware and upload the program. Connect the RGB LED strip to the A0/D0 interface of the XIAO expansion board and connect the light sensor to the A7 interface, as shown in the figure:



Next, XIAO is connected to the computer by the data cable, and then click the "Upload" button to upload the program to the hardware. when "Done uploading" is displayed in the debugging area, we can cover the light sensor with our hands, and then release the light sensor to observe the changes of the light strip.Because the strip takes time to show the light effect, it will not go out immediately when you cover the light sensor.

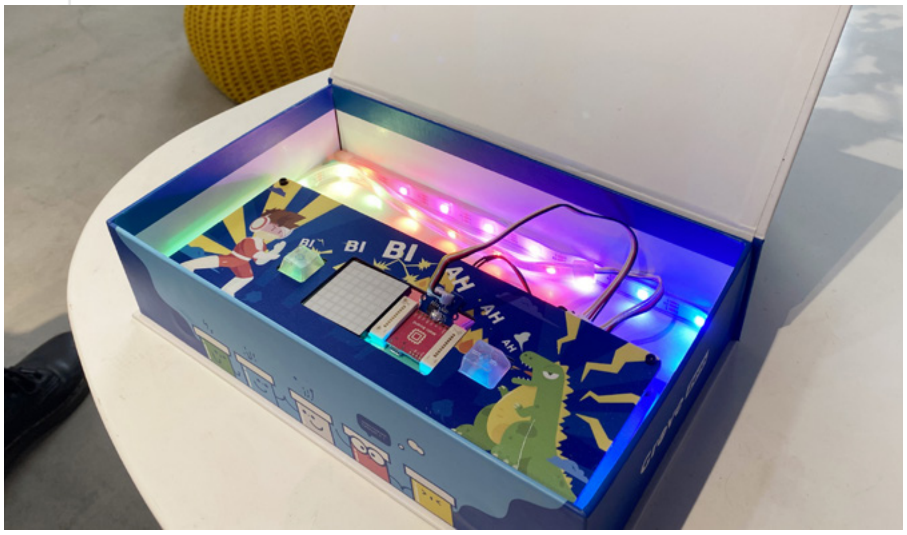


Appearance Design

It might be helpful to visualise how your surprise gift box will behave. For example, you might want your RGB LED lights to go out when your light sensor is in a dark environment, and for rainbow lights to go on once it is in a bright environment. This mirrors your electronics being placed in a dark, closed box; the lights should turn on only when it is open and exposed to surrounding light! Of course, your creativity might lead you to develop other designs!

Product name	Surprise Gift Box
Product characteristics	Cool lighting, light control, surprise, birthday
Product characteristics	Controlling the On/Off of RGB LED strip with Light Sensor
Product function	

References:



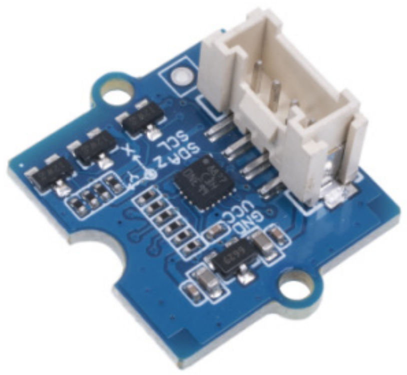
Lesson 10 Rhythmic Dance

Do you know how human beings perceive motion?Although we have acquired the basic ability of perception from human genes, we still need to adjust to the living environment and develop in a sound way through continuous training.When we were learning to walk, from crawling to standing, and then to walking in balance, all of these are senses in the human body that are constantly developing and growing.We found that mobile phones, computers and other electronic devices can also sense motion. For example, the screen of the mobile phone will rotate with the change of direction, and shaking the mobile phone can trigger the corresponding function,thanks to the three-axis accelerometer.In this lesson, we will learn to read motion data through the 3-axis accelerometer, and control the RGB LED strip to change the lighting effect with our motion.

Background Knowledge

3-Axis Digital Accelerometer

Grove -3- Axis Digital Accelerometer(LIS3DHTR) is a kind of sensor which can measure the acceleration of objects.In the process of motion, the acceleration value is obtained by measuring the inertial force of X, Y and Z axis mass and using Newton's second law. This can be achieved through a variety of sensing elements in accelerometers. Common accelerometers include capacitive, inductive, strain, piezoresistive, piezoelectric, etc.By measuring the acceleration caused by gravity, the inclination angle of the device relative to the horizontal direction can be calculated.Triaxial accelerometers are widely used in mobile phones, health bracelets and other areas of daily life, but also in virtual games, car safety, GPS satellite navigation, robots and other fields.

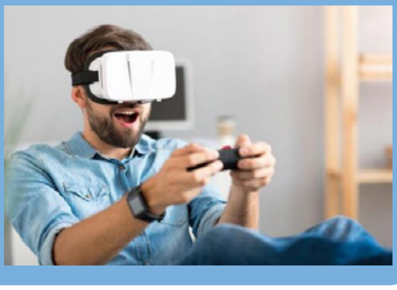


·Note·

Application of 3-axis accelerometer

Virtual game

In recent years, games have developed to step out of the screen into our world.We can synchronize human movement through AR / VR and gaming controllers. These game devices can sense the acceleration of human movement and



synchronize the same action of game virtual characters, so as to achieve a rich and immersive gaming experience game.

Automotive field

The application of 3-axis accelerometers in automobiles lies largely in safety systems and unmanned driving.The application of 3-axis accelerometer in automobile field is mainly in safety system and unmanned driving. Take the car body safety as an example, when the car body is impacted, the 3-axis accelerometer will detect the sudden change of acceleration and implement safety protection measures, such as timely ejection of the airbag to ensure the safety of passengers.

GPS satellite navigation

GPS navigation has brought great convenience to our life and travel. It can be used to plan routes, perform voice prompts, location, etc. It is good news for people who often travel to new cities and have a poor sense of direction.What is the role of the 3-axis accelerometer?When the signal of the satellite goes to an area of poor reception area or loses the signal in the normal environment, the 3-axis acceleration sensor based on MEMS technology cooperates with the gyroscope or electronic compass to create an azimuth reckoning system, which complements the GPS system where GPS positioning is not functioning well.

Read the Values of the x, y and z Axes of the 3-Axis Accelerometer

The key to making a project with3-axis accelerometer is to learn to read the values of3-axis accelerometer in X, Y and Z axes.Also referring to the library file- LIS3DHTR,the “LIS3DHTR_IIC” example can be opened through the following path: **File → Example → Grove-3-axis-digital-accelerator-2g-to-16g-LIS3DHTR → LIS3DHTR _ IIC**. The example program reads the values of the3-axis accelerometer in the X, Y and Z axes and output through the serial monitor. The example program provides us with different setting choices by way of “//”annotation, but some additional configuration is required, as follows:

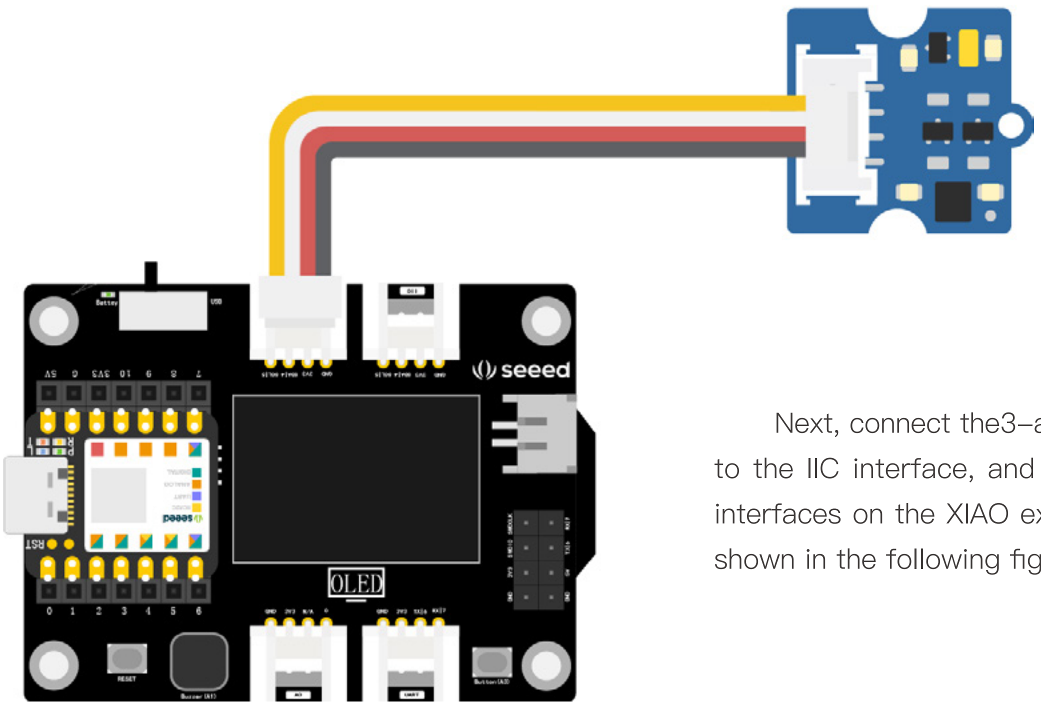
LIS.begin(WIRE);IIC initialization default value, there are 0x18 and 0x19 choice, we want to choose LIS.begin(WIRE,0x19);

LIS.setOutputDataRate(LIS3DHTR_DATARATE_50HZ); There are many choices for the output rate of the accelerometer, 50HZ is enough;

The complete procedure is as follows:

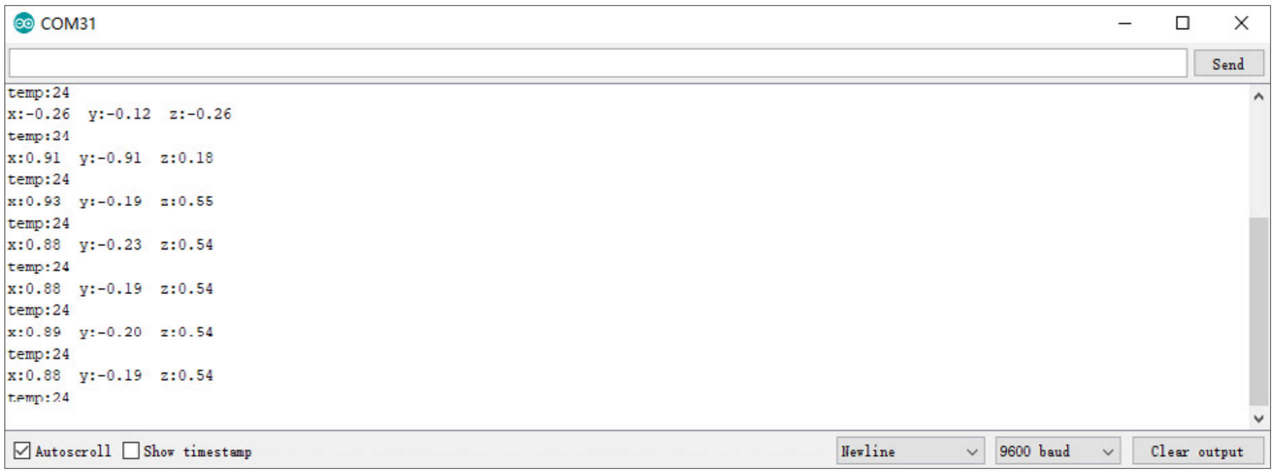
```
// This example use I2C.
#include “LIS3DHTR.h”
#include <Wire.h>
LIS3DHTR<TwoWire> LIS; //IIC
```

```
#define WIRE Wire

void setup()
{
  Serial.begin(115200);
  while (!Serial)
  {
  };
  LIS.begin(WIRE,0x19); //IIC init
  //LIS.begin(0x19);
  LIS.openTemp();  //If ADC3 is used, the temperature detection needs to be turned off.
  //  LIS.closeTemp();//default
  delay(100);
    LIS.setFullScaleRange(LIS3DHTR_RANGE_2G);
  //  LIS.setFullScaleRange(LIS3DHTR_RANGE_4G);
  //  LIS.setFullScaleRange(LIS3DHTR_RANGE_8G);
  //  LIS.setFullScaleRange(LIS3DHTR_RANGE_16G);
  //  LIS.setOutputDataRate(LIS3DHTR_DATARATE_1HZ);
  //  LIS.setOutputDataRate(LIS3DHTR_DATARATE_10HZ);
  //  LIS.setOutputDataRate(LIS3DHTR_DATARATE_25HZ);
  LIS.setOutputDataRate(LIS3DHTR_DATARATE_50HZ);
  //  LIS.setOutputDataRate(LIS3DHTR_DATARATE_100HZ);
  //  LIS.setOutputDataRate(LIS3DHTR_DATARATE_200HZ);
  //  LIS.setOutputDataRate(LIS3DHTR_DATARATE_1_6KHZ);
  //  LIS.setOutputDataRate(LIS3DHTR_DATARATE_5KHZ);
}
void loop()
{
  if (!LIS)
  {
    Serial.println("LIS3DHTR didn't connect.");
    while (1)
      ;
    return;
  }
  //3 axis
    Serial.print("x:"); Serial.print(LIS.getAccelerationX()); Serial.print("  ");
    Serial.print("y:"); Serial.print(LIS.getAccelerationY()); Serial.print("  ");
    Serial.print("z:"); Serial.println(LIS.getAccelerationZ());
  //ADC
//    Serial.print("adc1:"); Serial.println(LIS.readbitADC1());
//    Serial.print("adc2:"); Serial.println(LIS.readbitADC2());
//    Serial.print("adc3:"); Serial.println(LIS.readbitADC3());

//temperature
Serial.print("temp:");
Serial.println(LIS.getTemperature());
delay(500);
}
```



Next, connect the3-axis accelerometer to the IIC interface, and there are two IIC interfaces on the XIAO expansion board as shown in the following figure.

Connect XIAO to the computer with the data line and upload the program. After the program is uploaded successfully, open the serial monitor, and move the3-axis accelerometer in the directions of X, Y, and Z axes to observe the changes of the readings.



```
COM31
temp:24
x:-0.26  y:-0.12  z:-0.26
temp:24
x:0.91  y:-0.91  z:0.18
temp:24
x:0.93  y:-0.19  z:0.55
temp:24
x:0.88  y:-0.23  z:0.54
temp:24
x:0.88  y:-0.19  z:0.54
temp:24
x:0.89  y:-0.20  z:0.54
temp:24
x:0.88  y:-0.19  z:0.54
temp:24
☑ Autoscroll  ☐ Show timestamp    Newline    9600 baud    Clear output
```

Project Making

Project Description

We can add RGB LED lights to the project to achieve dazzling light effect transformation, use a 3–axis accelerometer to detect movement, and trigger different light effects based on the acceleration values in the X, Y and Z axes.

Write Program

The following steps are needed to control RGB LED strip through the 3–axis accelerometer:

• Declare the library to be used, and define the number of LED and RGB LED strip pins

• Initialise the 3–axis accelerometer and the RGB LED strip

• Set the LED strip to flash red, green and blue according to different intervals of X, Y, Z axis values from the 3–axis accelerometer

Task: Control RGB LED Strip Transform Lamp Efficiency via3–Axis Accelerometer

Step 1： Declare the library to be used, and define the number of LED and RGB LED strip pins.

```
#include “LIS3DHTR.h”
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
 #include <avr/power.h>
#endif

#ifdef SOFTWAREWIRE
    #include <SoftwareWire.h>
    SoftwareWire myWire(3, 2);
    LIS3DHTR<SoftwareWire> LIS; //Software I2C
    #define WIRE myWire
#else
    #include <Wire.h>
    LIS3DHTR<TwoWire> LIS; //Hardware I2C
    #define WIRE Wire
#endif

#define PIXEL_PIN 0//light strip connect to Pin 0
#define PIXEL_COUNT 30 //number of NeoPixels
Adafruit_NeoPixel strip(PIXEL_COUNT, PIXEL_PIN, NEO_GRB + NEO_KHZ800);//
declare our NeoPixel strip object
```

Step 2: Initialise the 3-axis accelerometer and the RGB LED strip.

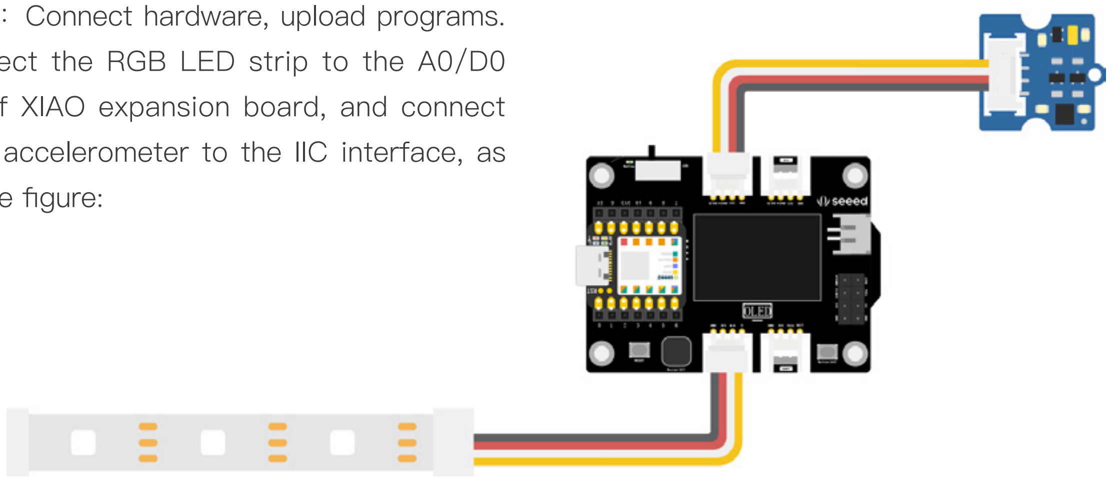
```
void setup() {
    Serial.begin(9600);
    while (!Serial) {};
    LIS.begin(WIRE, 0x19);//IIC initialization
    delay(100);
     LIS.setOutputDataRate(LIS3DHTR_DATARATE_50HZ);//set the output rate of the accelerometer to 50Hz
    strip.begin(); //initialize NeoPixel strip object
    strip.show(); //send the updated strip colors to the hardware.
}
```

Step 3: Set the LED strip to flash red, green and blue according to different intervals of X, Y, Z axis values from the 3-axis accelerometer, and the setting of the value needs us to check from a serial port monitor; when the3-axis accelerometer is moved in the x, y and z axes, the change of the value is observed to determine; because accelerometer values can be negative, we need to take their absolute value , so that we can conveniently set the condition .To find the absolute value, use the abs()function .For example, abs (LIS. getacceleration ()), which is the absolute value of the acceleration experienced in the x-axis by a 3-axis accelerometer.

```
void loop() {
    if (!LIS) {  //check whether the 3-axis accelerometer. is connected correctly
        Serial.println("LIS3DHTR didn't connect.");
        while (1);
        return;
    }

  if ((abs(LIS.getAccelerationX()) > 0.2)) {
   theaterChase(strip.Color(127, 0, 0), 50);//red
   }
  if ((abs(LIS.getAccelerationY()) > 0.2)) {
   theaterChase(strip.Color(0, 127, 0), 50); //green
   }
  if ((abs(LIS.getAccelerationZ()) > 1.0)) {
   theaterChase(strip.Color(0, 0, 127), 50); //blue
   }
   else
   {
     strip.clear();
     strip.show();
  }

    // //3 axis
    Serial.print("x:"); Serial.print(LIS.getAccelerationX()); Serial.print("  ");
    Serial.print("y:"); Serial.print(LIS.getAccelerationY()); Serial.print("  ");
    Serial.print("z:"); Serial.println(LIS.getAccelerationZ());

    delay(500);
}
//Theater-marquee-style chasing lights. Pass in a color (32-bit value,
// a la strip.Color(r,g,b) as mentioned above), and a delay time (in ms)
// between frames.
void theaterChase(uint32_t color, int wait) {
  for(int a=0; a<10; a++) {   //repeat 10 times...
    for(int b=0; b<3; b++) { //'b' counts from 0 to 2...
      strip.clear();   // set all pixels in RAM to 0 (off)
      for(int c=b; c<strip.numPixels(); c += 3) {
        strip.setPixelColor(c, color);//set pixel 'c' to value 'color'
      }
      strip.show(); // update strip with new contents
      delay(wait); //pause for a moment
    }
  }
}
```

Please see details on entire program: L10-Movement RGBLED.ino

Step 4: Connect hardware, upload programs. First, connect the RGB LED strip to the A0/D0 interface of XIAO expansion board, and connect the triaxial accelerometer to the IIC interface, as shown in the figure:



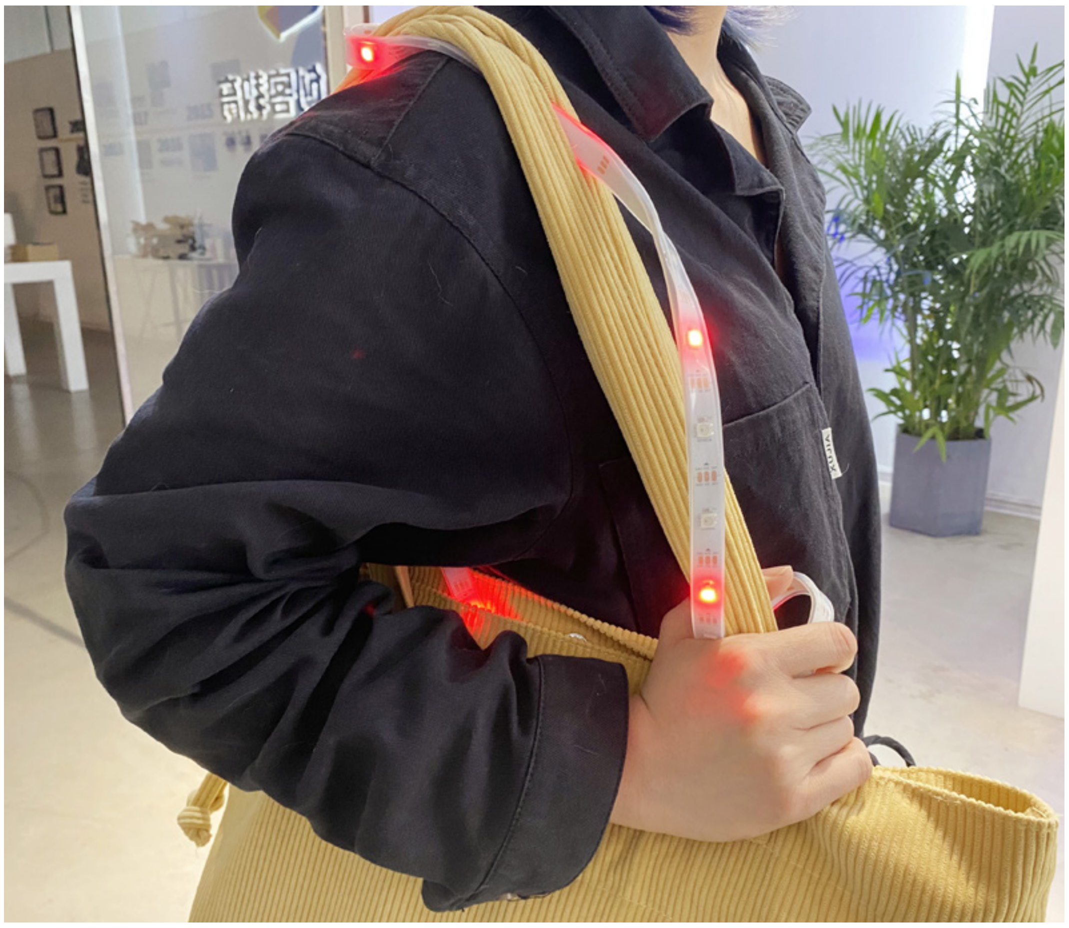
Connect XIAO to the computer with the data cable, and click the "upload" button to upload the program to the hardware. When "upload succeeded" is displayed in the debugging area, turn on the serial port monitor, and try to shake the triaxial accelerometer to the left, right and up and down to feel the change of the lamp efficiency.

⭐ Appearance Design

Imagine how cool it would be if the lights were flashing along with the dance steps when you waved your arms to show off your enthusiasm. This is the inspiration for rhythmic dance, which can be combined with clothes or accessories to make a wearable style.

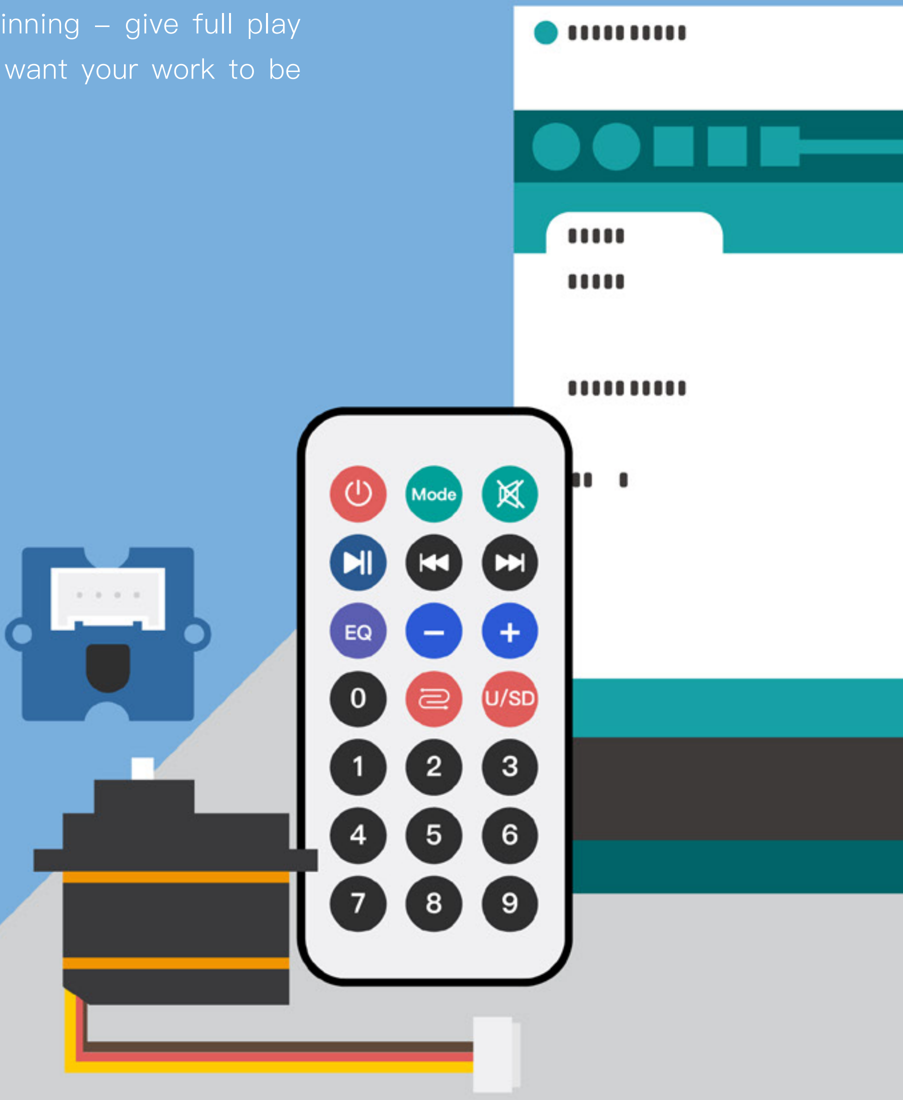
Product name	Rhythmic dance
Product characteristics	Wearable, cool lighting, attitude detection
Product characteristics	The RGB LED strip exhibits different lamp effects depending on the value change detected by the triaxial accelerometer
Product function	(For example, the waterproof layer on the outside of RGB LED strip can be removed and sewn with clothes or belts, etc.)

References:



Unit 3
Advanced Projects

In this unit, we will practice more complex and complete project.The effect of the program implementation and design structure are more directed to the finished product. For example, small smart homes, wearable electronic devices, and interactive electronic musical instruments, let us have a clearer understanding of the prototype design process.We will provide three laser cutting design drawings as reference cases. Of course, we can also use other materials, such as corrugated paper, cardboard and other more living materials to make them. The cases we provide are just the beginning – give full play to your imagination to decide how you want your work to be presented!



Lesson 11 Smart Remote Keyless Entry

In urban life, everyone attaches great importance to privacy and safety. Especially in residential areas where we live, every family not only has security doors, but they are also guarded by surveillance and security guards 24 hours a day.In recent years, the doors of residential areas have become intelligent,and can be opened by electronic keys or passwords.However, there are often residents who forget to bring their keys, and couriers or deliverymen who want to enter the community may need security guards to keep opening and closing doors.It is more convenient for them to work with smart remote keyless entry.The core of smart remote keyless entry is an infrared transmitter and infrared receiver, which can open and close doors by transmitting infrared signals from a remote control terminal and receiving infrared signals from door. In this lesson, we will try to design it.

Background Knowledge

Infrared Receiver and Transmitter

The infrared receiver is used for receiving infrared signals remotely. The infrared receiver is provided with an infrared detector which is used for obtaining infrared light emitted by an infrared emitter.The infrared receiver in the Grove family receives signals within an effective range of 10 meters, meaning it cannot receive signals from further than this distance. Generally, the infrared receiver and the infrared transmitter work together.

Infrared emitter is a kind of remote control equipment with remote control function. It emits light from inside to outside in a certain range through an infrared emitting tube, thus achieving the function of control signal. In daily life, the remote controller that controls TV, air conditioner and car door is the infrared transmitter. The common infrared transmitters are modular and also have remote controllers, which correspond to different application scenarios and usage methods.The smart remote keyless entry that we are making will be realized by infrared remote control.

In a nutshell, the principle of infrared transmission and reception is that the input signal from the infrared transmitting end is amplified and then sent to the infrared transmitting tube for final transmission, and the infrared receiving end receives the infrared signal and is processed and reduced to an electric signal by the amplifier, thus realizing infrared control.If you want to learn more about ”how does Infrared Communication work”, please refer to the following articles:

https://www.seeedstudio.com/blog/2021/02/20/make-a-universal-arduino-ir-remote-with-seeeduino-xiao/

Read the Remote Control Key Code

Want to control other equipment through the infrared remote controller, such as pressing the left key, servo turns to left, pressing the right key, servo turns to right, etc. First of all, we need to know what kind of code will be sent out by each key of the remote controller, so that we can set it through the program. How to read the codes of different keys on the remote control?We can use the IRremote library. Open the “irrecvemo” example via the following path: **File → Example → IRremote → IRrecvDemo**,the example program can read the remote control key code, but some parameters need to be modified:

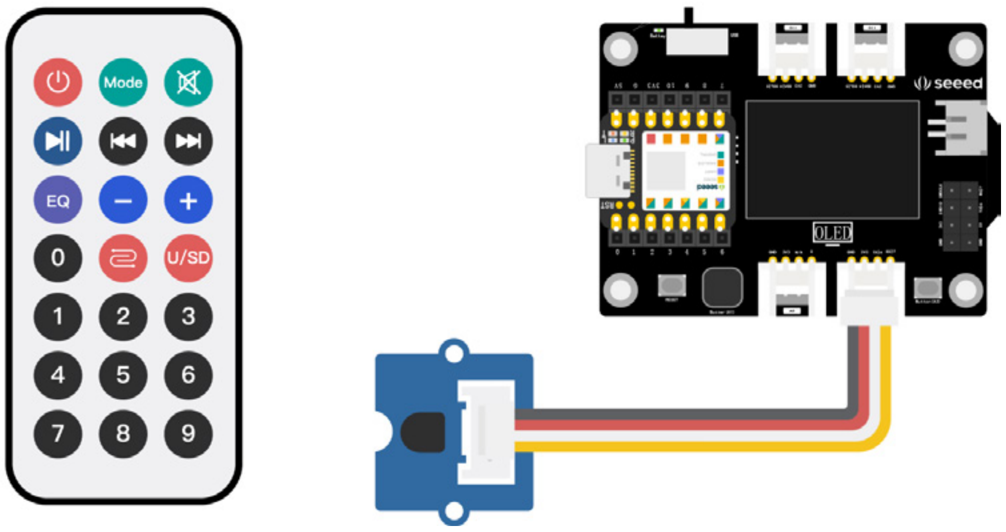
Int RECV_PIN = 7, we connected the IR receiver to pin 7 by changing the number depending on the hardware connection pin.

Next, select the useful code. We only need to define the header file and read the key code of the remote control. After cleaning up, the program is as follows:

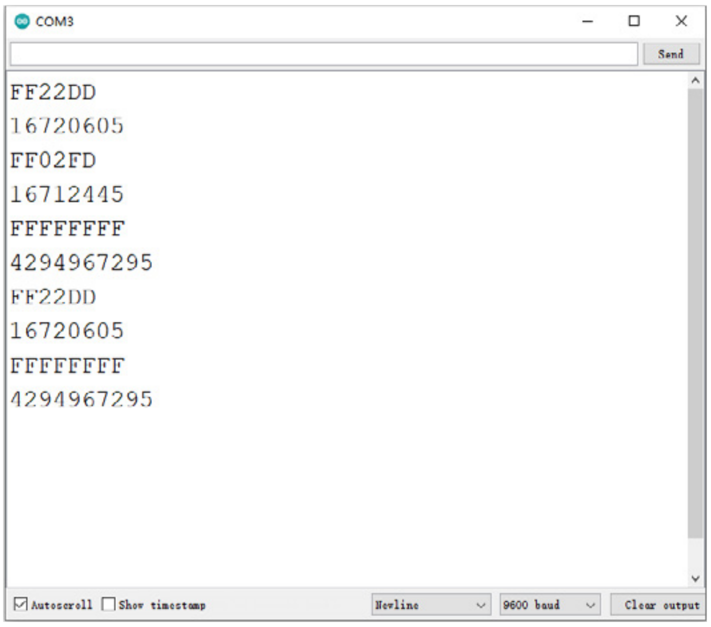
```
#include <IRremote.h>

int RECV_PIN = 7;//set pin 7 as IR control
IRrecv irrecv(RECV_PIN);//define IRrecv objects to receive infrared signals
decode_results results;//the decoding result is put in the results
void setup() {
    Serial.begin(9600);
    irrecv.enableIRIn(); //start the receiver
    Serial.println(RECV_PIN);
}

void loop() {
    if (irrecv.decode(&results)) { //checking IR signal
```

```
            Serial.println(results.value, HEX);//output infrared decoding result (hexadecimal)
            Serial.println(results.value);//output infrared decoding result (octal)
            irrecv.resume(); //recive next intrustion
        }
        delay(100);
    }
```

The infrared receiving module is connected to interface 7, as shown in the following figure:



After uploading the code, open the serial monitor, and acquiring the black element of the infrared receiver at close range with the remote controller, and press any key to observe the characters output by the serial monitor.The first line is hexadecimal code, the second line is octal code, two lines in a group, represents a key. When you press the key for too long, an "FFFFFFFF" will appear, and the line code and the number code below are invalid.



```
FF22DD
16720605
FF02FD
16712445
FFFFFFFF
4294967295
FF22DD
16720605
FFFFFFFF
4294967295
```

Project Making

Project Description

How do we reproduce the intelligent remote control door? In addition to the remote control and infrared receiver, the next step is to control the door switch.Let's recall how the remote keyless entry works in our life.When the remote control is pressed, the door opens, and then closes after reaching a certain angle.We can use the servo to control the door rotation. When the door closes, the servo's angle of rotation changes from 90 to 0. Conversely, when the door opens, that angle changes from 0 to 90. Through a transmission of two signals, we can instruct our door to both open and close!

Write Program

To control rotation of the servo with infrared remote controller, the following steps are required:

• Declare IRremote library and Servo library,define variables

• Initialize library file and servo

• Read the infrared decoding result, and control the servo to rotate according to the leftward and rightward instructions

Task: Infrared Remote Controller Controls Servo Rotation

Step 1: Declare IRremote library and Servo library,define variables.

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

Servo myservo;//create servo object to control a servo
int RECV_PIN = 7;//set pin 7 as IR control
IRrecv irrecv(RECV_PIN);
decode_results results;

int pos = 90;//variable to store the servo position
```

Step 2: Initialize library file and servo.

```
void setup()
{
  Serial.begin(9600);
  Serial.println(”Enabling IRin”);  //remind enabling IR
  irrecv.enableIRIn(); //start the receiver
  myservo.attach(5); //attaches the servo on pin 5 to the servo object
}
```

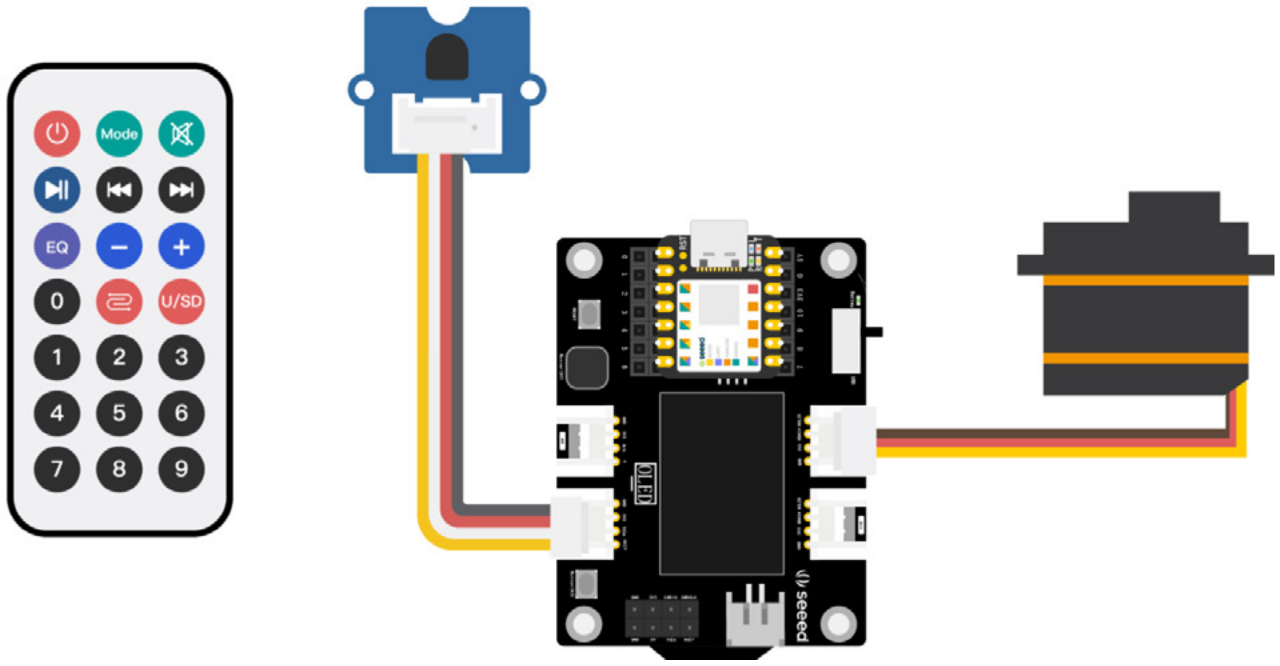
Step 3: Reading the infrared decoding result, and controlling the servo to rotate according to the leftward and rightward instructions. Refer to the notes section if you have questions about the procedure.

```
void loop() {
  if (irrecv.decode(&results)) {  // checking IR signal
   if (results.value == 16761405) {  //if the received signal is 16761405 (right key)
      for (pos; pos <= 89; pos += 1) { //goes from 0 degrees to 90 degrees
        myservo.write(pos);              //tell servo to go to position in variable 'pos'
        delay(40);
        //here are that instruction to interrupt the above to exit the loop
        if (irrecv.decode(&results)) {
          irrecv.resume();
          if (results.value == 16712445)
            break;
        }
      }
    }

    if (results.value == 16712445) {    // If the received signal is 16761405 (left key)
      for (pos; pos >= 1; pos -= 1) { // goes from 90 degrees to 0 degrees
        myservo.write(pos);              // tell servo to go to position in variable 'pos'
        delay(40);
       // here are that instruction to interrupt the above to exit the loop
        if (irrecv.decode(&results)) {
          irrecv.resume();
          if (results.value == 16761405)
            break;
        }
      }
    }
    //displays hexadecimal and octal encodings in the serial port
    Serial.println(pos);
    Serial.println(results.value, HEX);
    Serial.println(results.value);
    irrecv.resume();

  }
  delay(100);
}
```

Please see details on entire program:

L11-IR_Servo.ino

Step 4: Connect the hardware and upload the program.First, connect the infrared receiving module to No.7 interface of XIAO expansion board, and connect the steering engine to IIC interface, as shown in the figure:



Connect XIAO to the computer with the data cable, and click the “Upload” button to upload the program to the hardware. when “Done uploading” is displayed in the debugging area,open the serial monitor, aim at the infrared receiver with the remote controller, press the “Left” key and “Right” key, observe the rotation of the servo, and view the coded information output by the serial monitor.

Appearance Design

In this module, we will make the project ore complete by forming a foundational prototype. We can do this by considering how the program functions will be realised, along with the physical presentation of the product.To recap, we will control the rotation of servo through the remote controller to simulate the opening and closing of the door. When designing the appearance, we should focus on the following issues:

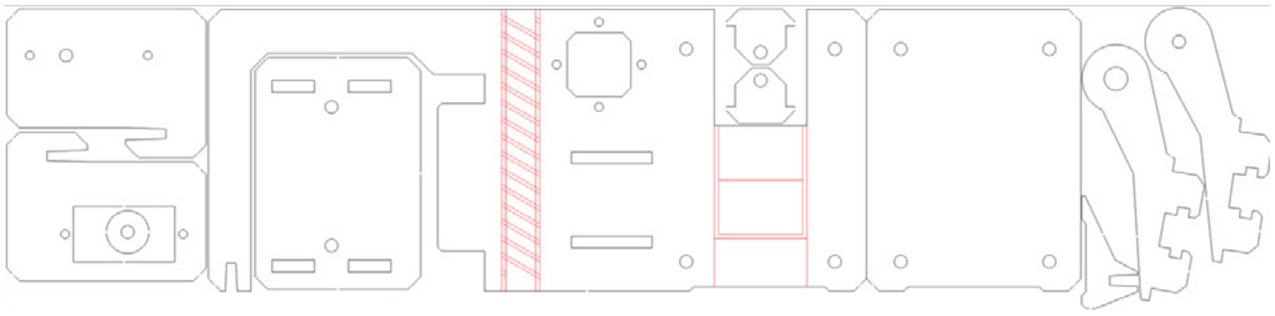
- How to combine the servo with the door panel, so that the rotation of door panel is achieved through the servo rotation.

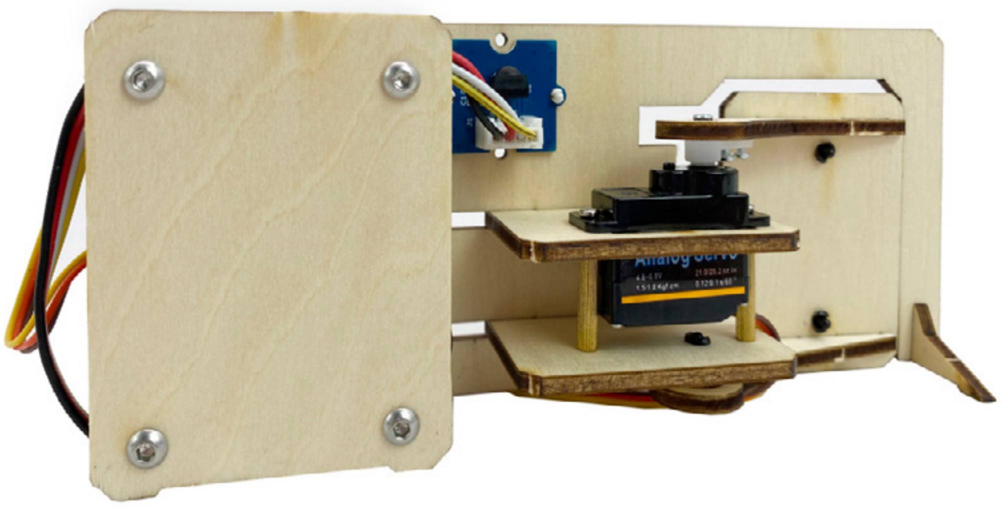
• The infrared receiver should be exposed in a conspicuous position without any shade.

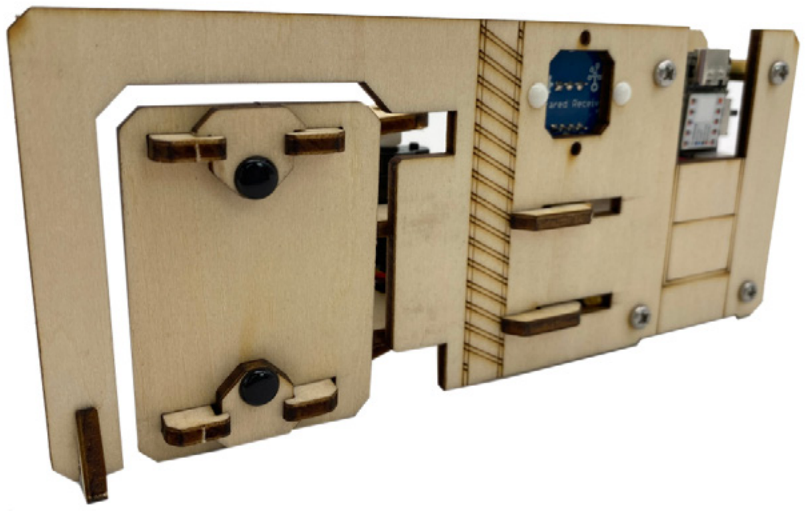
• Concealing the main control, expansion board and cable cover to keep the appearance clean and tidy

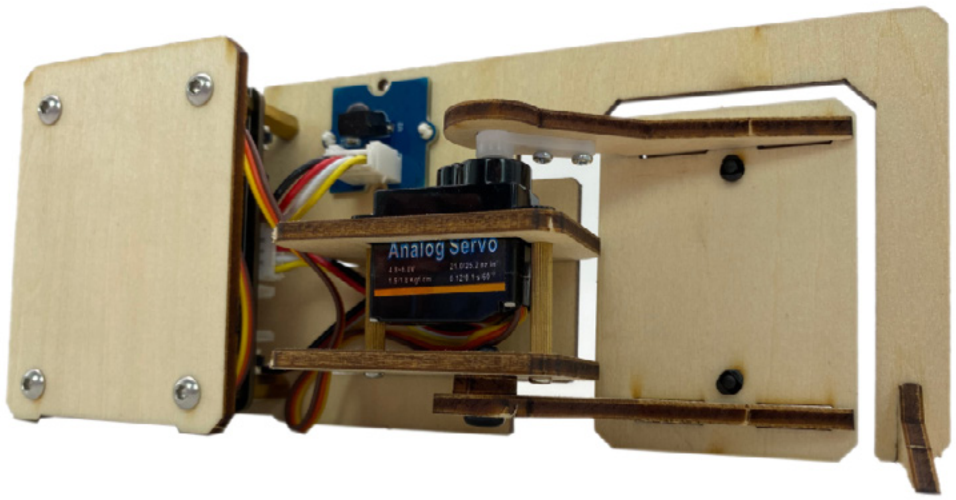
• How does the product stand up steadily

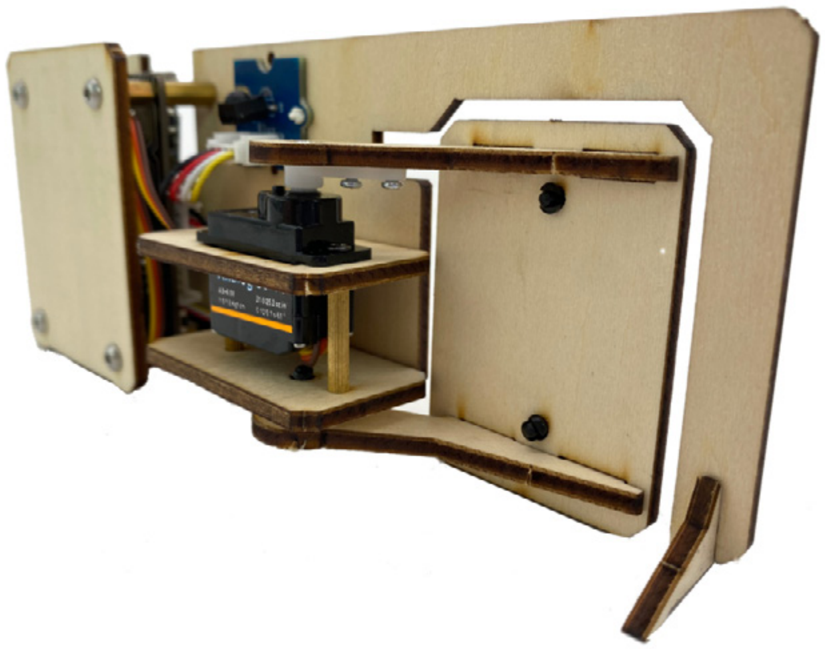
The external case design is as follows: it is laser cut from basswood board, and the cutting documents are provided for reference. If you can use drawing software, you can process the design yourself.If there is no laser cutting machine, it can also be made of corrugated paper, cardboard, non-woven cloth and other hand-made materials, which will test your practical ability.








Analog Servo



Lesson 12 Smart Watch

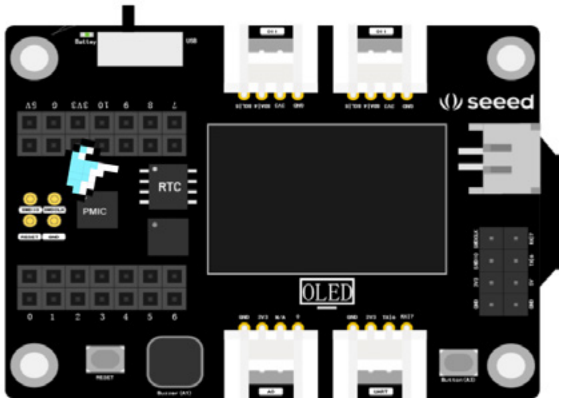
Watches are very common things in life, all kinds of electronic devices have the ability to keep time. Although the mobile phone can replace the watch as a clock, the watch is still a favorite accessory among many people. It is not only a time keeping tool, but also carries functions in fashion and aesthetics.Although watches are small and exquisite, they are manufactured through a very complicated process. But it can be easily made through XIAO and his expansion board. Can you guess how?

Background Knowledge

RTC Clock

RTC clock stands for real–time clock. RTC is an integrated circuit that can keep time, also called RTC clock .RTCs are widely used in many electronic devices. In fact, the XIAO expansion board is equipped with one!

Using the expansion board, we can display the date and time on the OLED display when we supply power externally. The RTC keeps track of time even when power is removed from the main device. Without an RTC, the system will stop keeping track of time when power is removed, giving us an inaccurate time when it is switched back on.Through the RTC clock, we can make time–dependent reminders or tasks, such as watering flowers or pet feeding at specific times of day.

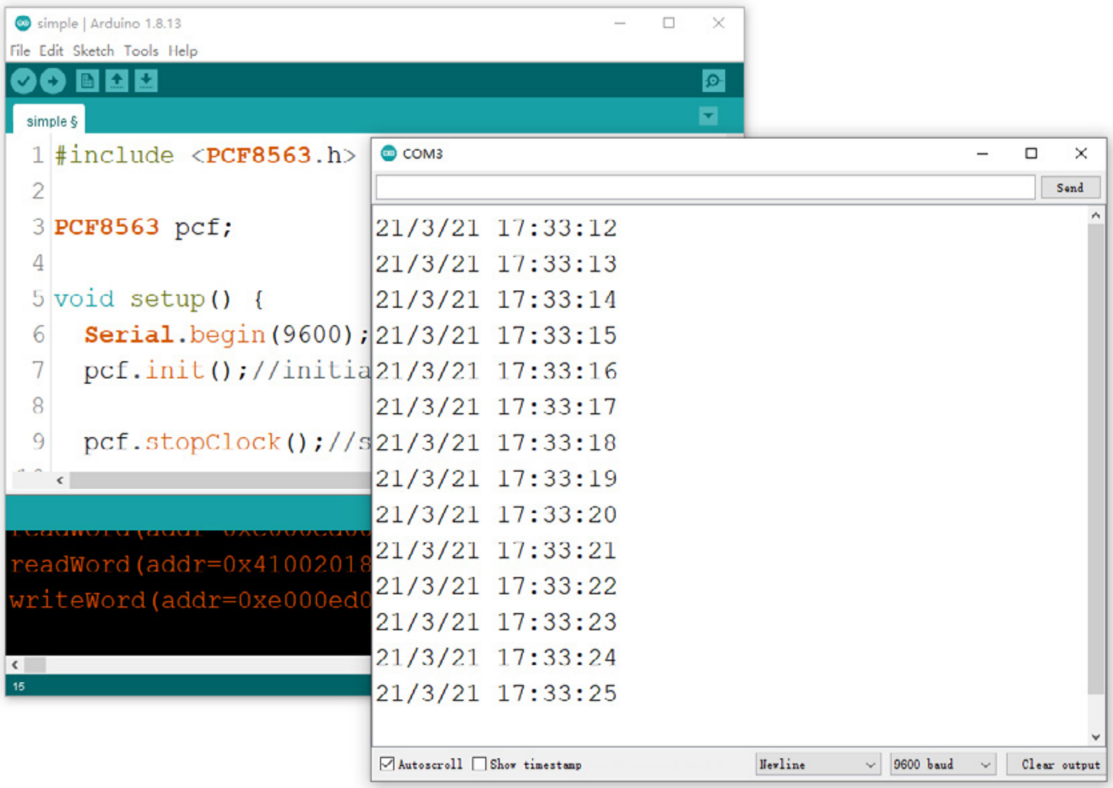


Display RTC Clock in Serial Port

To make an RTC clock, you need to use the relevant library.You can open the “simple” example through the following path: **File → Example → PCF8563 → simple**. this example program can display the RTC clock through a serial monitor. after you open the example program, you only need to modify the current date, month, day, and start time:

```
#include <PCF8563.h>
PCF8563 pcf;// Define variable pcf

void setup() {
  Serial.begin(9600);
  pcf.init();//initialize clock
  pcf.stopClock();//stop the clock

  //Set the current date and time.
  //After the setting is completed, the timing will start from this moment.

  pcf.setYear(21);//set year
  pcf.setMonth(3);//set month
  pcf.setDay(8);//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

  //serial port prints the current date and time
  Serial.print(nowTime.day);
  Serial.print("/");
  Serial.print(nowTime.month);
  Serial.print("/");
  Serial.print(nowTime.year);
  Serial.print("21"); //manually enter the year of the setting
  Serial.print(nowTime.hour);
  Serial.print(":");
  Serial.print(nowTime.minute);
  Serial.print(":");
  Serial.println(nowTime.second);
  delay(1000);
}
```

There's no need to connect other electronic modules, click on the upload program. Once the code uploaded, open the serial monitor,to see the time.



```
#include <PCF8563.h>

PCF8563 pcf;

void setup() {
  Serial.begin(9600);
  pcf.init();//initia
  
  pcf.stopClock();//s
```

```
readWord(addr=0x41002018
writeWord(addr=0xe000ed0
```

```
21/3/21 17:33:12
21/3/21 17:33:13
21/3/21 17:33:14
21/3/21 17:33:15
21/3/21 17:33:16
21/3/21 17:33:17
21/3/21 17:33:18
21/3/21 17:33:19
21/3/21 17:33:20
21/3/21 17:33:21
21/3/21 17:33:22
21/3/21 17:33:23
21/3/21 17:33:24
21/3/21 17:33:25
```

·Note·

The development of watches

In 1504, a man named Peter Haehnlein invented a portable clock, but it was not very accurate. In 1656, Christiaan Huygens invented the pendulum, and then Bryce Pascal used a rope to fasten the pocket watch to his wrist, thus the watch was born.The oldest way to measure time is to observe the position of the sun in the sky, and the sundial is based on this invention.During the day, when the sun shines on the sundial, the vertical rod on the disk will cast a shadow on the dial, which can produce relatively accurate time reading, what about reading the time at night?

To work around this issue,the mechanical clock was invented in the 14th century. It is small in size and more consistent in timing. It can be timed by a series of complicated wheels, such as gears and levers, in coordination with a pendulum. In the 18th century, such small mechanical clocks gradually entered households.The development of watches has been gradually refined into different application scenarios, with different styles, such as wall-mounted watches, tabletop watches, wearable watches, pocket watches, etc. The categories also include mechanical, Shi Ying and electronic categories, especially watches, which have developed into more forms and styles, reflecting the identity taste of the wearer.

Project Making

Project Description

In this lesson, we are going to make a smart watch, which can not only display the date and time in real time, but also display surrounding temperature and humidity.

Write Program

The program is divided into the following steps:

• Declarethe library, defines the variable

• Initialize the library file, set the current time

• Read the temperature and humidity variables, acquire the current time, and display the temperature, humidity, and date time on the OLED screen

Task: Display the Current Time and Temperature and Humidity Values on the OLED Display

Step 1: Declares the library file that needs to be called, defines the variable

```
#include <Arduino.h>
#include <U8x8lib.h>//use u8x8 library
#include <PCF8563.h>// RTC library
PCF8563 pcf;//define variable pcf
#include <Wire.h>
#include “DHT.h” // DHT library
#define DHTTYPE DHT20//DHT20
DHT dht(DHTTYPE);
U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(/* reset=*/ U8X8_PIN_NONE);
```

Step 2: Initialize the library file, set the current time.

```
void setup() {
  Serial.begin(9600);
  u8x8.begin();//initialize the u8x8 library
  u8x8.setFlipMode(1);
  Wire.begin();
  pcf.init();//initialize clock
  pcf.stopClock();//stop clock
  //set the current time and date:
  pcf.setYear(21);
  pcf.setMonth(3);
  pcf.setDay(5);
  pcf.setHour(18);
  pcf.setMinut(53);
  pcf.setSecond(0);
  pcf.startClock();//start the clock
}

```

Step 3: Read the temperature and humidity variables, acquire the current time, and display the temperature, humidity, and date time on the OLED screen.

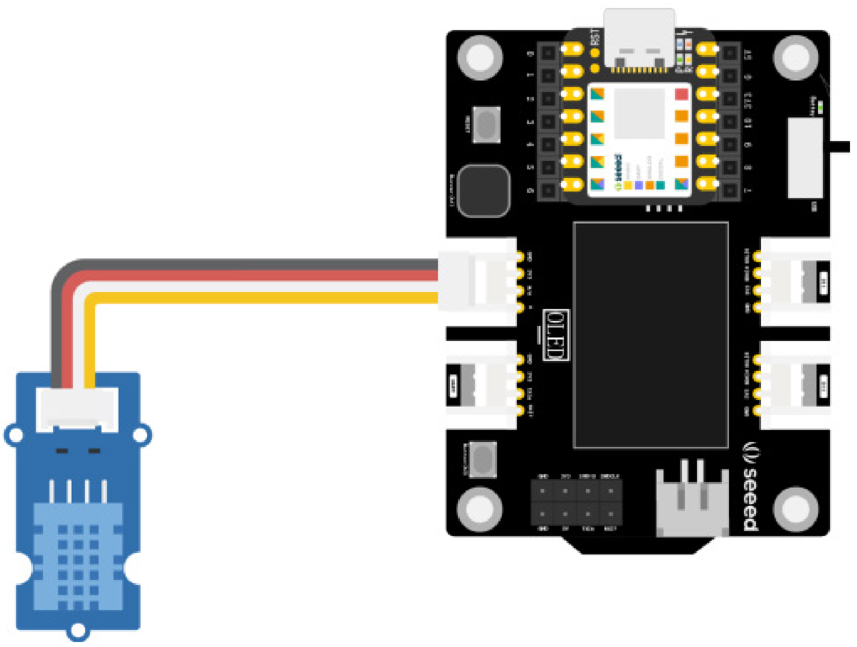
```
void loop() {
  float temp, humi;//defines variables to store readings
  temp = dht.readTemperature();//read the temperature value and store it in temp
  humi = dht.readHumidity();//read the humidity value and store it in the humi
  Time nowTime = pcf.getTime();//get current time
  u8x8.setFont(u8x8_font_chroma48medium8_r); //set the font for display.

  //The current date, time, temperature and humidity are display in different coordinates on that OLED screen
  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);
  u8x8.setCursor(0, 2);
  u8x8.print(“Temp:”);
  u8x8.print(temp);
  u8x8.print(“C”);
  u8x8.setCursor(0,3);
  u8x8.print(“Humidity:”);
  u8x8.print(humi);
```

```
    u8x8.print("%");
    u8x8.refreshDisplay();
    delay(200);
  }
```

Please see details on entire program:

L12-SmartWatc
h.ino

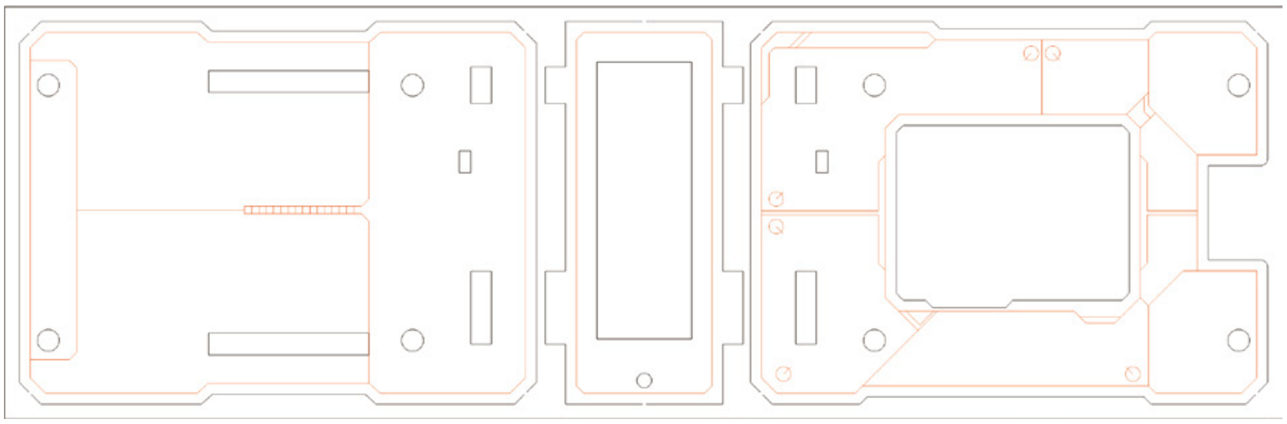
Firstly, connect the temperature and humidity sensor to A0 interface of XIAO expansion board, and connect XIAO to computer with data cable, as shown in the figure:

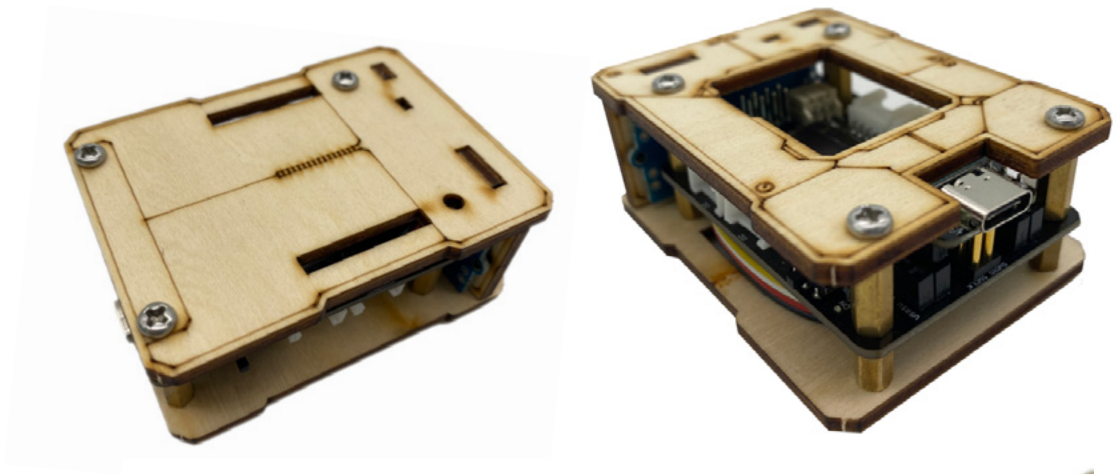


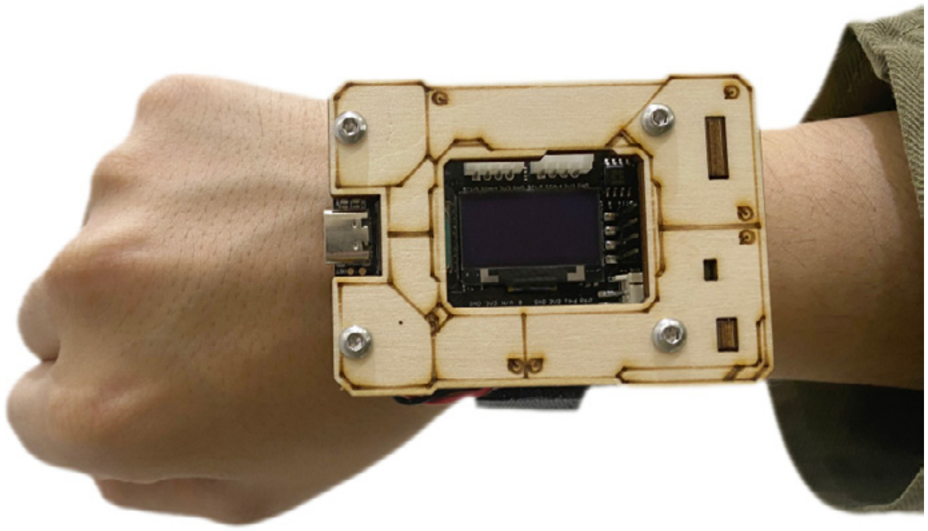
Click the "Upload" button to upload the program to the hardware, and "Done uploading" will be displayed in the debugging area.Observe whether the OLED display correctly outputs the current time and start timing, as well as the real-time temperature and humidity.

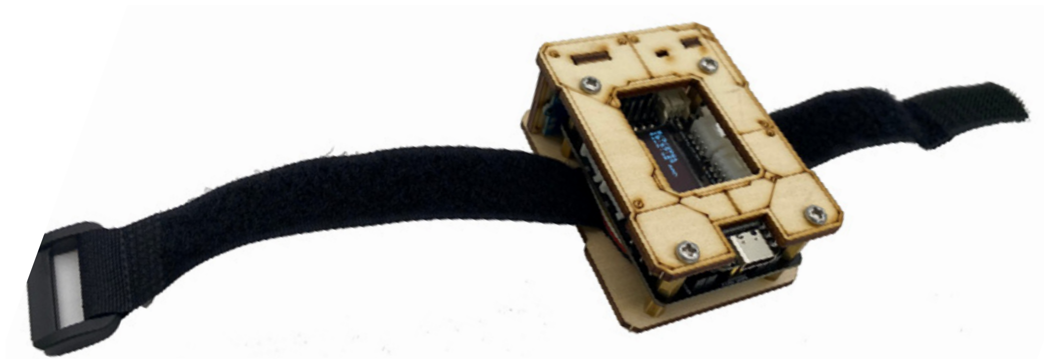
⭐ Appearance Design

XIAO is especially suitable for making wearable devices because of its small size. The RTC chip, buzzer and OLED display are integrated on the expansion board, so many variations can be explored even without external modules.In this lesson, we made a smart watch using the on-board OLED display, RTC chip and external temperature and humidity sensors.When the appearance, we need to consider the wearability, the arrangement and storage of modules and connecting wires, and the exposure of OLED display. As shown in the figure below, we have provided a wearable watch style with a laser cut file for easy installation.









Lesson 13 Air Piano

When we play an instrument, we usually make the notes sound by plucking strings or pressing keys.We can make playing music more interesting through some electronic module, such as simulating piano playing through key switches, or adding lighting effects to interact with music. However, if a key switch is used as a key, many modules need to be connected into the circuit. Do you have a simpler and more unique idea?How about a combination of ultrasonic distance measuring sensor and passive buzzer to realize this idea?Using the ultrasonic sensor to detect different notes triggered by different distances is like playing the piano in the air. Isn't this very interesting?

Background Knowledge

Ultrasonic Distance Sensor

Ultrasonic ranger is a non-contact distance measuring module. Because of its strong directionality, the emitted ultrasound can travel a long distance in the medium, and its calculation is simple and easy to use. It is often used for distance measurement.When the ultrasonic ranger is in opeartion, the transmitter emits a sound wave in one direction that is reflected when it hits the obstacle. When the ultrasonic receiver receives the reflected wave, it records the time taken for the wave to return. The actual distance from the emitted point to the obstacle is calculated from the time difference between the emitted and received, just like the echolocation of bats.Ultrasound is also used more and more widely, such as reverse radar system, intelligent blinding system, robot obstacle avoidance system, medical ultrasound imaging and so on.

Reading the Value of the Ultrasonic Ranger

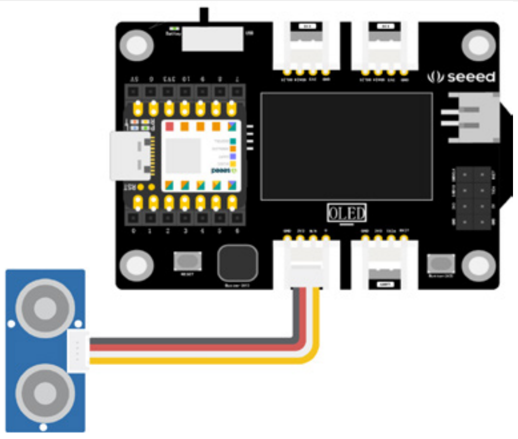
Open the UltrasonicDisplayOnTerm example: **File → Example → Ultrasonic Ranger →UltrasonicDisplayOnTerm**, which reads the distance values measured by the ultrasonic distance sensor and outputs them to the serial port monitor in inches and centimeters respectively.The changing parameters are as follows:

Ultrasonic ultrasonic(0);Defines the ultrasonic variable, which connects to the A0 pin.

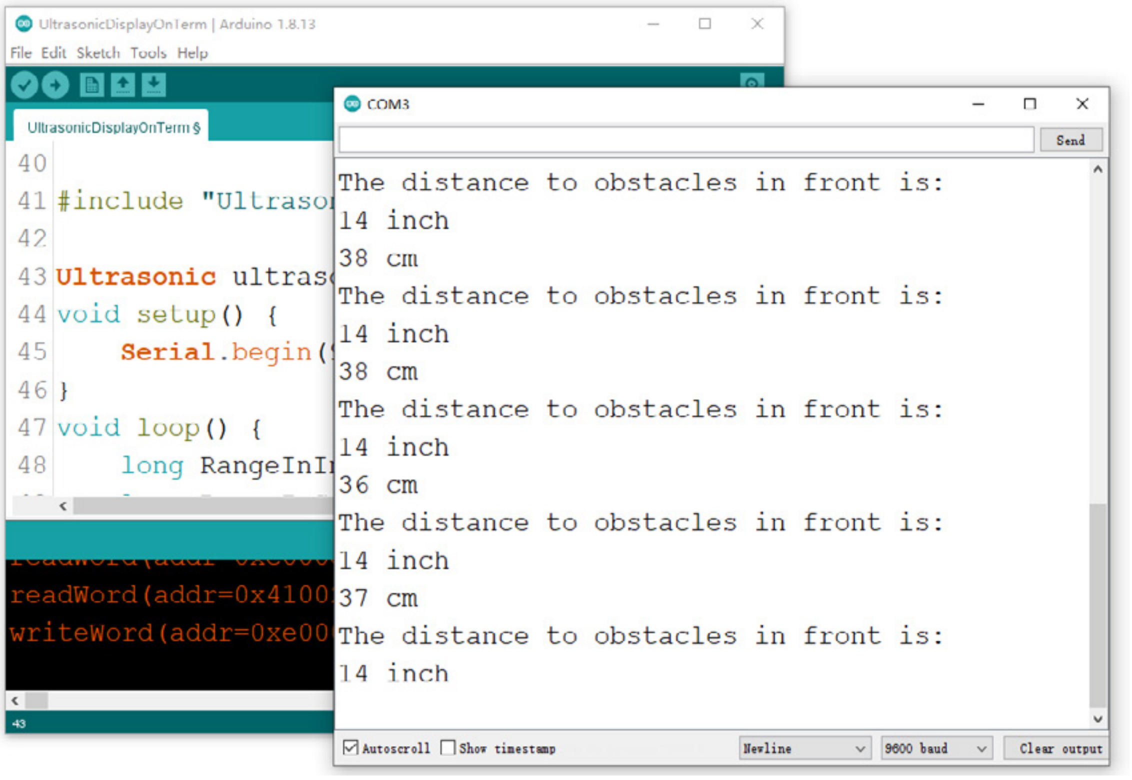
Other without modification, the program as shown in the figure below:

```
#include “Ultrasonic.h”
Ultrasonic ultrasonic(0);//define variables, connect pins
void setup() {
    Serial.begin(9600);
}
void loop() {
    long RangeInInches;//long variables
    long RangeInCentimeters;//long variables

    Serial.println(“The distance to obstacles in front is: ”);
     RangeInInches = ultrasonic.MeasureInInches();//Read the distance value (inch)
measured by the ultrasonic ranging sensor and store it in the variable RangeInInches
    Serial.print(RangeInInches);//serial port prints value
    Serial.println(“ inch”);
    delay(250);

     RangeInCentimeters = ultrasonic.MeasureInCentimeters(); //read the distance
value (centimeters) measured by the ultrasonic ranging sensor and store it in the variable
RangeInInches
    Serial.print(RangeInCentimeters);//serial port prints value
    Serial.println(“ cm”);
    delay(250);
}
```

Ultrasonic distance sensor is connected to A0 interface, as shown in the figure below:



After uploading the code, open the serial monitor. Place your hand or any object in front of the ultrasonic distance sensor and observe how the displayed value changes!


UltrasonicDisplayOnTerm | Arduino 1.8.13
File Edit Sketch Tools Help
UltrasonicDisplayOnTerm §
COM3
Send
Autoscroll
Show timestamp
Newline
9600 baud
Clear output

```
#include "Ultraso
Ultrasonic ultras
void setup() {
    Serial.begin(
}
void loop() {
    long RangeInI
```

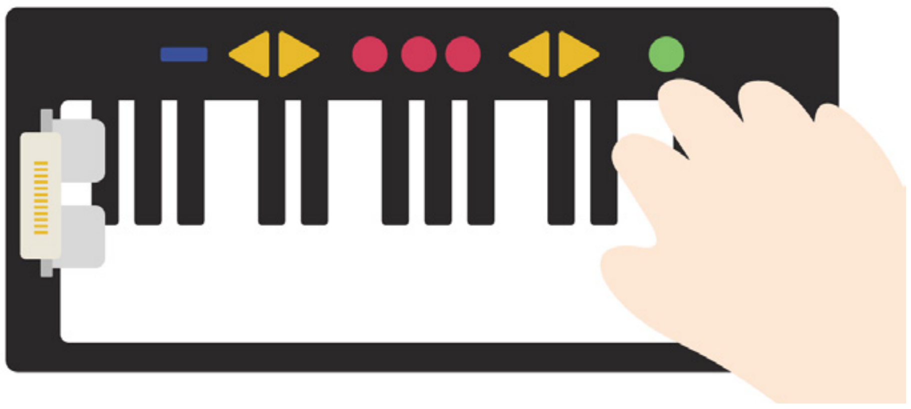
```
readWord(addr=0x4100
writeWord(addr=0xe00
```

```
The distance to obstacles in front is:
14 inch
38 cm
The distance to obstacles in front is:
14 inch
38 cm
The distance to obstacles in front is:
14 inch
36 cm
The distance to obstacles in front is:
14 inch
37 cm
The distance to obstacles in front is:
14 inch
```

Project Making

Project Description

The working principle of the air piano is to measure the distance from the module to the palm through the ultrasonic distance sensor, and control the buzzer to emit different notes according to the different distances.We have learned how to measure the distance and read the value by the ultrasonic distance sensor through the example program, and then we only need to define different notes for the corresponding distance.As shown in the following figure: Each note is played in an interval of 2cm, starting at a distance of 4cm from the ultrasonic distance sensor. Hance,“Do Re Mi Fa Sol La Xi Do” respectively corresponds to “4cm 6cm 8cm……” and so on.



Write Program

The program implementation of the air piano needs the following steps:

• Declare library file, define different note and buzzer pin

• Initialize and set buzzer pin status

• Read the distance (cm) measured by the ultrasonic ranging sensor, and judge the conditions, set different distances to emit different notes

·Note·

ToneMelody

When we control the buzzer to play notes or songs through programs, we need to set the frequency value of each note. If there are many notes in a song, it is too troublesome to adjust one by one, which will test our musical theory knowledge and intonation.Is there a simpler way? Sure! When defining notes, we can use the function is written in Arduino website, https://www.arduino.cc/en/Tutorial/BuiltInExamples/toneMelodywhich defines the corresponding frequencies of different notes, so that we can use tone () function to set the notes emitted by buzzer.As shown below:

```
/*
 * pitches.h
*/

#define NOTE_B0  31
#define NOTE_C1  33
#define NOTE_CS1 35
#define NOTE_D1  37
#define NOTE_DS1 39
#define NOTE_E1  41
#define NOTE_F1  44
#define NOTE_FS1 46
#define NOTE_G1  49
#define NOTE_GS1 52
#define NOTE_A1  55
#define NOTE_AS1 58
#define NOTE_B1  62
#define NOTE_C2  65
#define NOTE_CS2 69
#define NOTE_D2  73
#define NOTE_DS2 78
#define NOTE_E2  82
#define NOTE_F2  87
#define NOTE_FS2 93
#define NOTE_G2  98
#define NOTE_GS2 104
#define NOTE_A2  110
#define NOTE_AS2 117
#define NOTE_B2  123
#define NOTE_C3  131
#define NOTE_CS3 139
#define NOTE_D3  147
#define NOTE_DS3 156
#define NOTE_E3  165
#define NOTE_F3  175
#define NOTE_FS3 185
#define NOTE_G3  196
#define NOTE_GS3 208
#define NOTE_A3  220
#define NOTE_AS3 233
#define NOTE_B3  247
#define NOTE_C4  262
#define NOTE_CS4 277
#define NOTE_D4  294
#define NOTE_DS4 311
#define NOTE_E4  330
#define NOTE_F4  349
#define NOTE_FS4 370
```

```
#define NOTE_G4  392
#define NOTE_GS4 415
#define NOTE_A4  440
#define NOTE_AS4 466
#define NOTE_B4  494
#define NOTE_C5  523
#define NOTE_CS5 554
#define NOTE_D5  587
#define NOTE_DS5 622
#define NOTE_E5  659
#define NOTE_F5  698
#define NOTE_FS5 740
#define NOTE_G5  784
#define NOTE_GS5 831
#define NOTE_A5  880
#define NOTE_AS5 932
#define NOTE_B5  988
#define NOTE_C6  1047
#define NOTE_CS6 1109
#define NOTE_D6  1175
#define NOTE_DS6 1245
#define NOTE_E6  1319
#define NOTE_F6  1397
#define NOTE_FS6 1480
#define NOTE_G6  1568
#define NOTE_GS6 1661
#define NOTE_A6  1760
#define NOTE_AS6 1865
#define NOTE_B6  1976
#define NOTE_C7  2093
#define NOTE_CS7 2217
#define NOTE_D7  2349
#define NOTE_DS7 2489
#define NOTE_E7  2637
#define NOTE_F7  2794
#define NOTE_FS7 2960
#define NOTE_G7  3136
#define NOTE_GS7 3322
#define NOTE_A7  3520
#define NOTE_AS7 3729
#define NOTE_B7  3951
#define NOTE_C8  4186
#define NOTE_CS8 4435
#define NOTE_D8  4699
#define NOTE_DS8 4978
```

Task: Ultrasonic Air Piano

Step 1: Declare the library file and define different notes and buzzer pins.The main notes we use are “Do Re Mi Fa Sol La Xi Do”, which corresponds to “C5 D5 E5 F5 G5 A5 B5 C6”. You can just define the notes you need to avoid the program being too lengthy.

```
#include “Ultrasonic.h”
Ultrasonic ultrasonic(0);//define the ultrasonic object and connect the ultrasound to the A0 interface
int buzzerPin = 3;//buzzer connected to A3

#define NOTE_C5  523
#define NOTE_CS5 554
#define NOTE_D5  587
#define NOTE_DS5 622
#define NOTE_E5  659
#define NOTE_F5  698
```

```
#define NOTE_FS5 740
#define NOTE_G5  784
#define NOTE_GS5 831
#define NOTE_A5  880
#define NOTE_AS5 932
#define NOTE_B5  988
#define NOTE_C6  1047
```

Step 2: Initialize and set buzzer pin status.

```
void setup()
{
  Serial.begin(9600);
  pinMode(buzzerPin,OUTPUT);
}
```

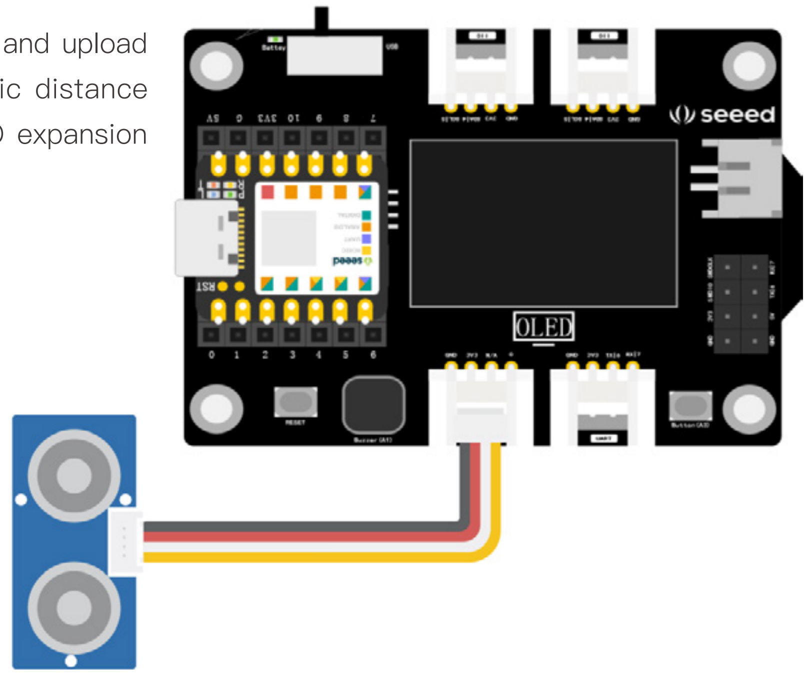
Step 3: Read the distance (cm) measured by the ultrasonic distance sensor and set different notes to be emitted from different distances.Each distance value is a long integer value, therefore the ultrasonic return value is defined by the long() function.For example (long)RangeInCentimeters= = 4, the distance value returned by the ultrasound is 4.When the buzzer emits different notes, the tone () function is used,for example, tone(3,NOTE_C5,100), means the buzzer at pin 3 emits NOTE_C5(D0) for 100 milliseconds.

```
void loop()
{
//Read the distance value detected by the ultrasonic distance sensor in centimeters and print on the serial monitor
  long RangeInCentimeters;
  RangeInCentimeters = ultrasonic.MeasureInCentimeters();
  Serial.print(RangeInCentimeters);
  Serial.println(" cm");
  delay(250);
  //Condition judgment is performed by the if statement. when the distances are 4, 6, 8, 10, 12, 14, 16, and 18, C5, D5, E5, F5, G5, A5, B5, and C6, respectively
  if (((long)RangeInCentimeters== 4)) {  //Do
    tone(3,NOTE_C5,100);
    }
  if (((long) RangeInCentimeters== 6)) { //Re
    tone(3,NOTE_D5,100);
    }
    if (((long) RangeInCentimeters== 8)) { //Mi
      tone(3,NOTE_E5,100);
        }
    if (((long) RangeInCentimeters== 10)) {  //Fa
      tone(3,NOTE_F5,100);
       }
        if (((long) RangeInCentimeters== 12)) {  //Sol
      tone(3,NOTE_G5,100);
       }
    if (((long) RangeInCentimeters== 14)) {  //La
      tone(3,NOTE_A5,100);
       }
    if (((long) RangeInCentimeters== 16)) { //Xi
      tone(3,NOTE_B5,100);
       }
    if (((long) RangeInCentimeters== 18)) {  //Do
      tone(3,NOTE_C6,100);
       }
}
```

Please see details on entire program:

L12-UltrasonicPiano.ino

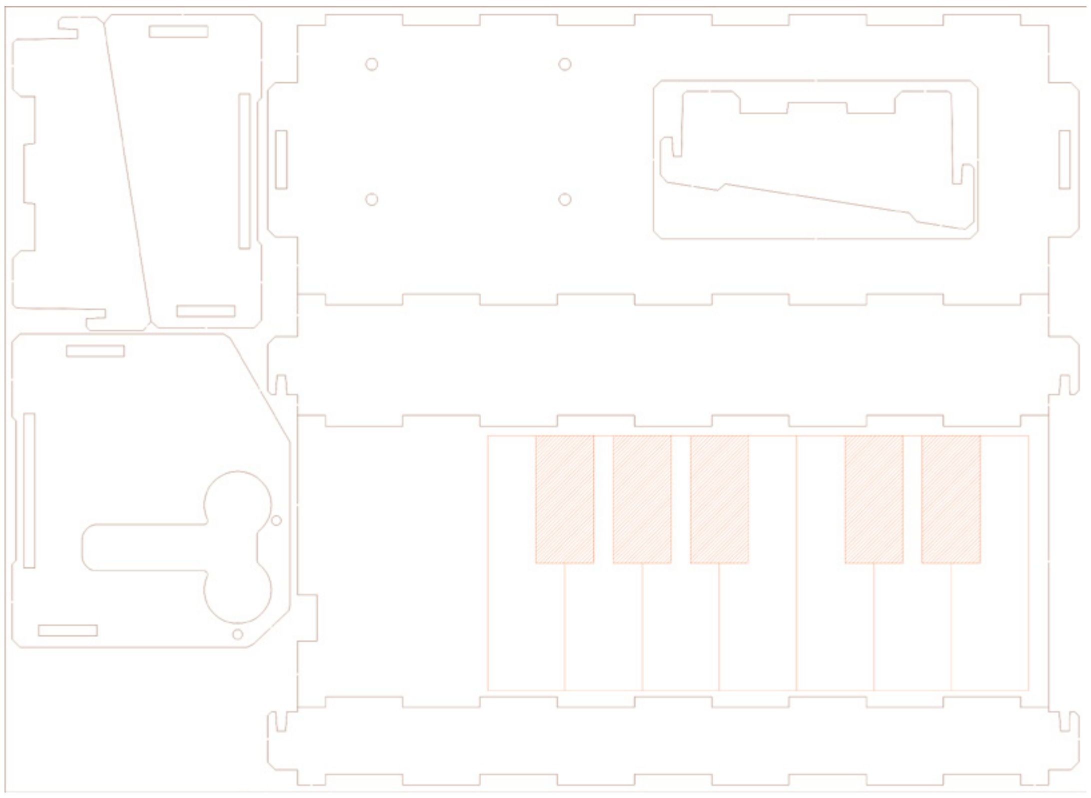
Step 4: Connect the hardware and upload the program.Connect the ultrasonic distance sensor to the A0 interface of XIAO expansion board, as shown in the figure:

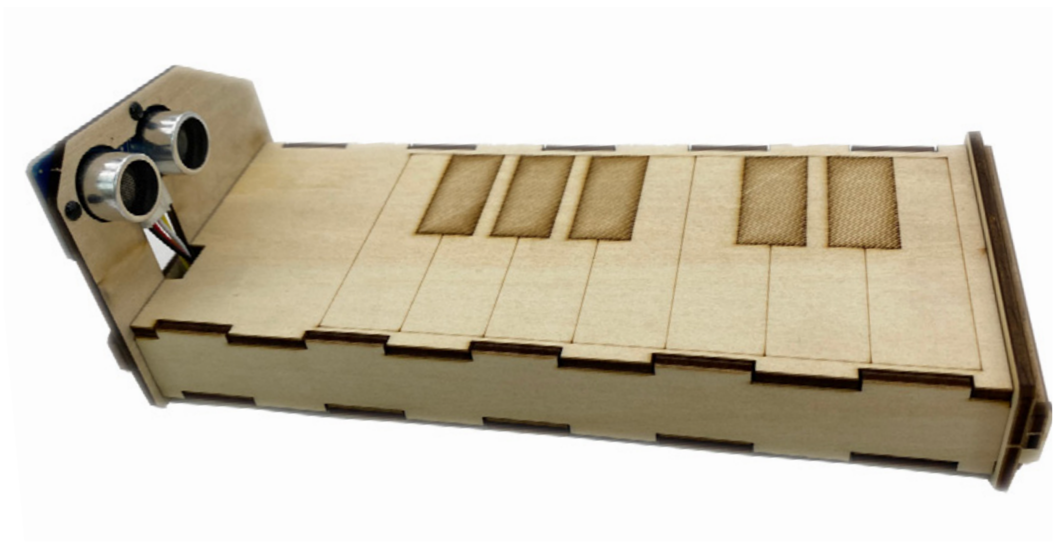


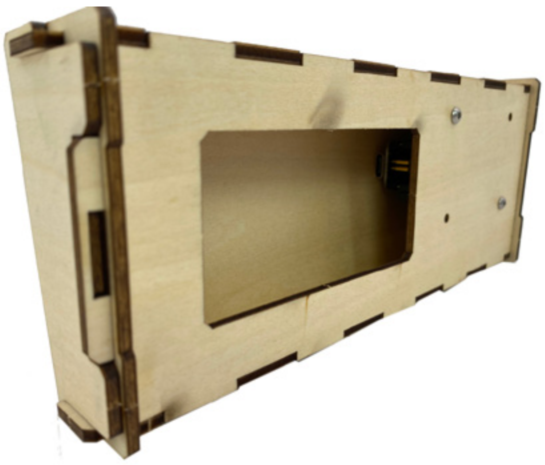
Connect XIAO to the computer with the data cable, and click the "Upload" button to upload the program to the hardware. when "Done uploading" is displayed in the debugging area, open the serial monitor, and start playing the piano with your palm.

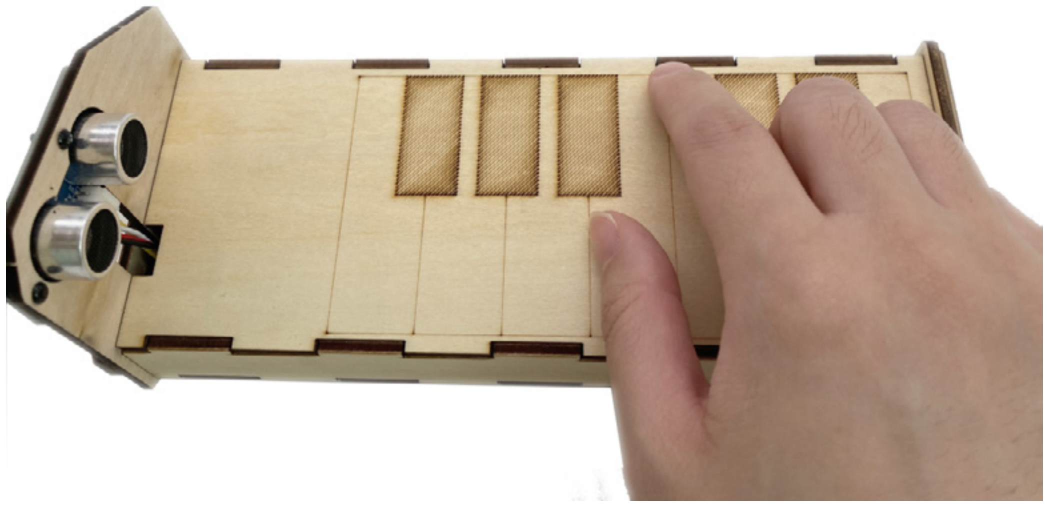
Appearance Design

The inspiration of the air piano is the piano, and every 2cm a note is also designed according to the style of the keys.When making the case, we can cut a piano surface with basswood board and fix the ultrasonic distance measuring sensor at the left end of the piano. The laser cutting document is provided as a reference, and simple assembly can be completed as shown in the figure









Unit 4
Self-developed Project

After the previous study, in this unit, we will practice what we have learned, starting from the transformation of the work, and then working together to complete a prototype work from 0, which will be presented in a press conference.I believe that in this process, you can experience more fun and learn things other than knowledge, such as teamwork, speech and eloquence, etc. This is an era of knowledge sharing. Only by learning to show, communicate and share can we gain more.

Lesson 14 Challenge Transformation Project

In this lesson, we will review the previous projects and make improvements in function and appearance. You may also try to start by adapting old projects to consolidate your knowledge. You can extend the basic functions of an existing project, redesign it, write code, or modify and beautify its appearance and structure. Go ahead boldly with your work!

Project Review

Review the produced project from the project name, function and appearance:

Project name	Transducer	Actuator	Function
Intelligent temperature and humidity meter	Temperature and humidity sensor	OLED,Buzzer	The temperature and humidity are displayed on the OLED screen, and when the temperature or humidity values are out of the set range, an alarm is raised
Surprise gift box	Light sensor	RGB LED strip	When the light sensor detects a strong light value, the RGB LED strip lights are rainbow breathing light
Intelligent remote control door	Infrared remote controller,infrared receiver	Servo	Through the remote controller and infrared receiver, the servo rotation is controlled and the function of the remote control door is simulated
Rhythmic dance	3-axis digital accelerometer	RGB LED strip	Different actions are detected and the lamp efficiency of the RGB LED strip lights are changed by by the 3-axis digital accelerometer

Air piano	Ultrasonic ranging sensor	Buzzer	The distance from the module to the palm is measured by the ultrasonic distance measuring sensor, and the buzzer is controlled to emit different notes according to different distances
Smart watch	RTC module, temperature and humidity sensor	OLED	Smart watch capable of displaying time, temperature and humidity

Project Renovation

Which projects can be revised? Select the function or appearance of any item for renovation. And fill in the following form according to your own ideas.Before filling in, please think carefully about the problem:I want to renovate _________ , make a _____________(function,users,situations)________(name), because__________.

For example, I want to transform the surprise gift box and make an acousto-optic music box that can present rainbow lights and play the happy birthday song . It is even more exciting to have music and light effects!

Renovation scheme: _________	
Project name after renovation	
Ideas	
Electronic module used	
Function after renovation	
Program code	
Appearance structure	

After determining the renovation plan, which it can be carried out. Please be sure to complete the renovation plan in class and complete the whole plan after class. If someone has finished, please show your work 5 minutes before class.

Lesson 15 Self-study Project Planning

In this lesson, we will plan our self-developed project based on what we have learned in Lesson 7, Introduction to Product Prototype Design. Each group will complete one self-developed project in groups.

Project Requirements

1. Open themes, with panellists discussing and identifying themes;

2 .Pay attention to the practicability and feasibility of the scheme;

3. When making the appearance structure, it is not limited to a certain material, but can be freely chosen;

4.In addition to the production of products, it is necessary to consider how to share and display the products;

5.At least complete the project plan for this class. If not, complete it after class. In the next class, the project will be shared and displayed. Please reserve 10 minutes for debugging and take the time as appropriate.

Roles and Responsibilities of the Team

Each team needs to select a team leader, name the team, perform a reasonable division of roles, and complete their respective tasks;

Roles and responsibilities	Project production team	Programmer	Member：	Responsibility：
		Appearance building	Member：	Responsibility：
		Scheme design and record	Member：	Responsibility：
	Project publishing group	Presentation of member(show ppt and speech draft)	Member：	Responsibility：

Responsibilities: Project production team: responsible for completing product production, including hardware programming and appearance structure production, and keeping the production source files for the press conference.Project release team: responsible for the preparation of appearance production materials in the next class, and according to the product to complete PPT for the presentation.

Project Planning

Determine the product content and complete the planning scheme through.

Product name	
Background of design	
Users	
Concept	
The problem you are trying to solve	1、 2、 3、
Functional description	1、 2、 3、
Product composition	For example, a smart watch consists of a dial made of acrylic and a bracelet.
Project	E.g. smart light
Select electronic module	
Program code	
Draft product appearance	
Material for making product appearance	E.g. wooden board, cardboard

After the completion of project planning, the production phase can be started. Please ensure good team communication and cooperation.In the next lesson, the works will be shared in the form of a product conference.Each group should prepare the following materials in advance:

1.Show the form and content, each group of 6 minutes

2.Project planning scheme

3.Show PPT with content self-defined

4.Show the form and content, each group of 6 minutes

Lesson 16 Sharing and Presentation of Self-study Projects

The whole semester's learning results will be tested in this lesson, as well share and display them in the form of a product conference. Teachers or students can host the whole conference and give brief introduction of the conference process.

Press Conference Preparation

1.Preparation of product launch team and field personnel

Host:1

Timekeeper:1

Stage supervisor(Collection and playing of PPT content):1

Counting personnel:2

Taking photos in the spot:1

2.Material personnel

Material:computer

Classroom arrangement:tables/chairs arrangement

Materials collection: Collection and playing of PPT content

Making vote: 6 p/team

Prize preparation: 6 prizes

Press Conference

1. In the opening session, announce the rules of press conference and award, and give each group 10 minutes for debugging and preparation;

2 .In the process of product display, please launch and display the products in turn under the organization of the host;

3. In the selection process, all products will be selected after release. There are six awards in total, and each group has six opportunities to vote freely. The votes only represent the opinions of the group rather than individual opinions, so please vote after synthesizing the opinions of all members of the group.

4. In the awarding process, the voting results will be announced to select the best creative award, the most common touch award, the most investment award, the most technological award, the best team award and the future star award.

Award reference:

Best Creativity Award: the product concept may not be fully realized, but the idea is bold and creative, which is likely to be realized in the future;

The most common touch award:high completion, the knowledge learned is used comprehensively, and the product design is more practical;

Most investment Award: creative marketing and business awareness, focusing on product market strategy.

Most science and Technology Award: the product function design is full of science and technology power, and the release site and content are quite shocking

Best team award: good team cooperation, clear division of labor, but mutual help, smooth cooperation

Future star award: team and project with high potential in ideas and execution, approaching perfection in details!

Afterword

Resources, downloads, links and instructions related to this lesson.

About Grove–temperature and humidity sensor DHT11

The temperature and humidity sensor is used to detect the temperature and humidity in the environment.There are many types of temperature and humidity sensors.We use the DHT11 temperature and humidity sensor with calibrated digital signal output.The capacitive sensor element on the sensor can measure the relative humidity. The temperature is measured through the thermistor.It is very suitable to use it to make an intelligent temperature and humidity meter. Temperature and humidity sensors are widely used in agriculture, environmental protection and home life.

Reading temperature and humidity values in a serial monitor

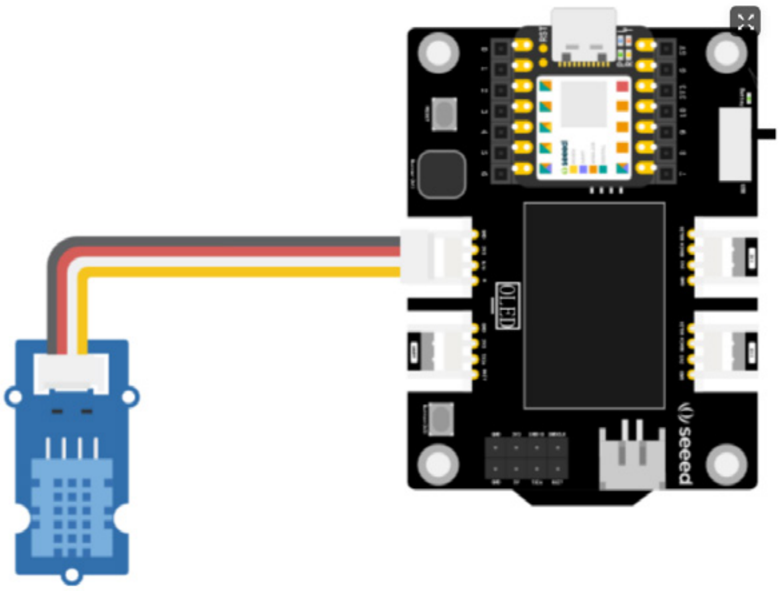
When using the sensor, we can use the DHT library to open the "DHTtester" example through the following path: "File → Example → Temperature and humidity sensor → DHT tester. "after opening the example program, we can see the following program.The program can read the temperature and relative humidity information in the environment and display real–time data on the serial monitor. Some code for the sample program needs to be modified:

#define DHTPIN 0, we need to modify the parameters according to the pin number of the actual connection of the temperature and humidity sensor.

#define DHTTYPE DHT11, because the temperature and humidity sensors have different models, we have to choose the correct model.

```
#include "DHT.h"
#define DHTPIN 0 //temperature and humidity sensor connected to A0 pin
#define DHTTYPE DHT11   //DHT 11
DHT dht(DHTPIN, DHTTYPE);
#if defined(ARDUINO_ARCH_AVR)
    #define debug  Serial
#elif defined(ARDUINO_ARCH_SAMD) ||  defined(ARDUINO_ARCH_SAM)
    #define debug  SerialUSB
#else
    #define debug  Serial
#endif
//Initializing serial port display
void setup() {
    debug.begin(115200);
    debug.println("DHTxx test!");
    Wire.begin();
    dht.begin();
}
void loop() {
//Read the temperature and humidity values and print on the serial monitor
    float temp_hum_val[2] = {0};
    if (!dht.readTempAndHumidity(temp_hum_val)) {
        debug.print("Humidity: ");
        debug.print(temp_hum_val[0]);
        debug.print(" %\t");
        debug.print("Temperature: ");
        debug.print(temp_hum_val[1]);
        debug.println(" *C");
    } else {
        debug.println("Failed to get temprature and humidity value.");
    }
    delay(1500);
}
```

After modifying the code, connect the temperature and humidity sensor to A0 interface, upload the sample program to XIAO, and open the serial monitor, you can see the values of temperature and humidity. You can put the temperature and humidity sensors in different environments to observe whether the temperature and humidity values will change.



In the course, we used the DHT20 in Lesson 8 and Lesson 12. The corresponding DHT11 version code is as follows:

Seeeduino XIAO for Seeed Wiki

https://wiki.seeedstudio.com/Seeeduino-XIAO/
https://wiki.seeedstudio.com/Seeeduino-XIAO-Expansion-Board/

Purchase link

Seeeduino XIAO & Seeeduino XIAO Expansion board
https://www.seeedstudio.com/Seeeduino-XIAO-Pre-Soldered-p-4747.html
https://www.seeedstudio.com/Seeeduino-XIAO-Expansion-board-p-4746.html
Seeeduino XIAO Starter Kit for Arduino
Coming soon...

Call for multilingual translation volunteers!

If you wish to translate course content into a language version other than English or Chinese, you can also contact us for support.

CONTACTS

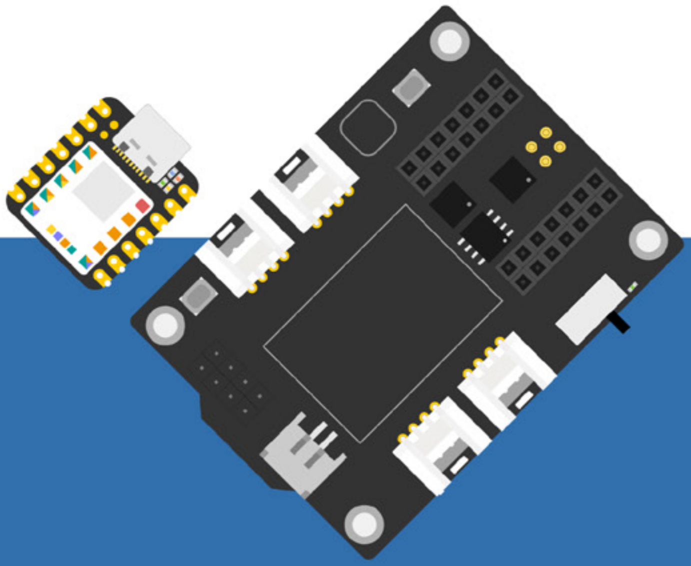
Tel: +86-0755-86716703
Address: 1002, G3 Building, TCL International E City, 1001 Zhongshan Park Road, Nanshan District, Shenzhen
General: contact@chaihuo.org
www.tinkergen.com
www.seeedstudio.com

Course Developers

This course was co-authored by the following SeeedStudio employees:
Authors:Yimeng Shi
Designer:Yihui Meng
Proofreaders:Lei Feng,Jonathan Tan

TinkerGen

STEM made simple



TinkerGen STEM made simple | seeed studio

To know more about TinkerGen

Please contact us:

contact@chaihuo.org   86-0755-86716703   www.tinkergen.com