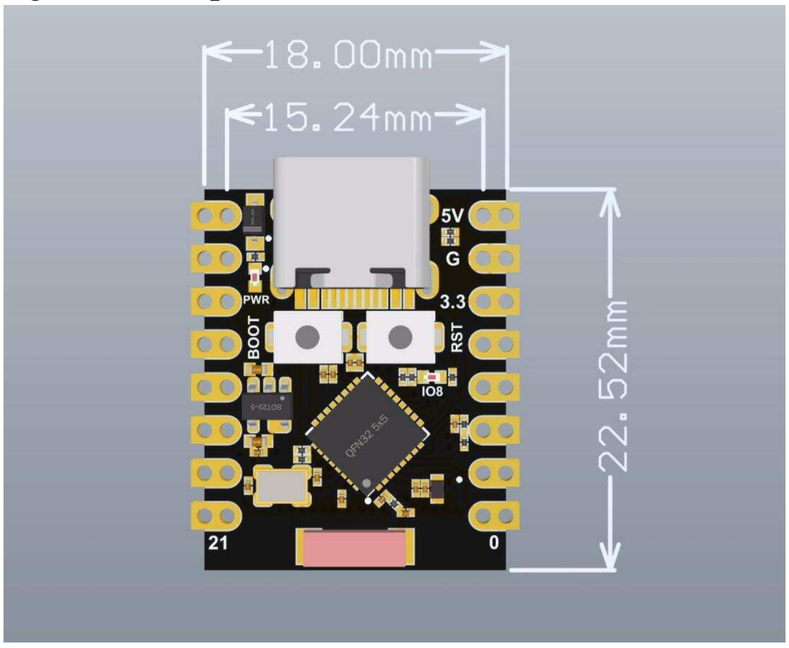
Product Introduction:

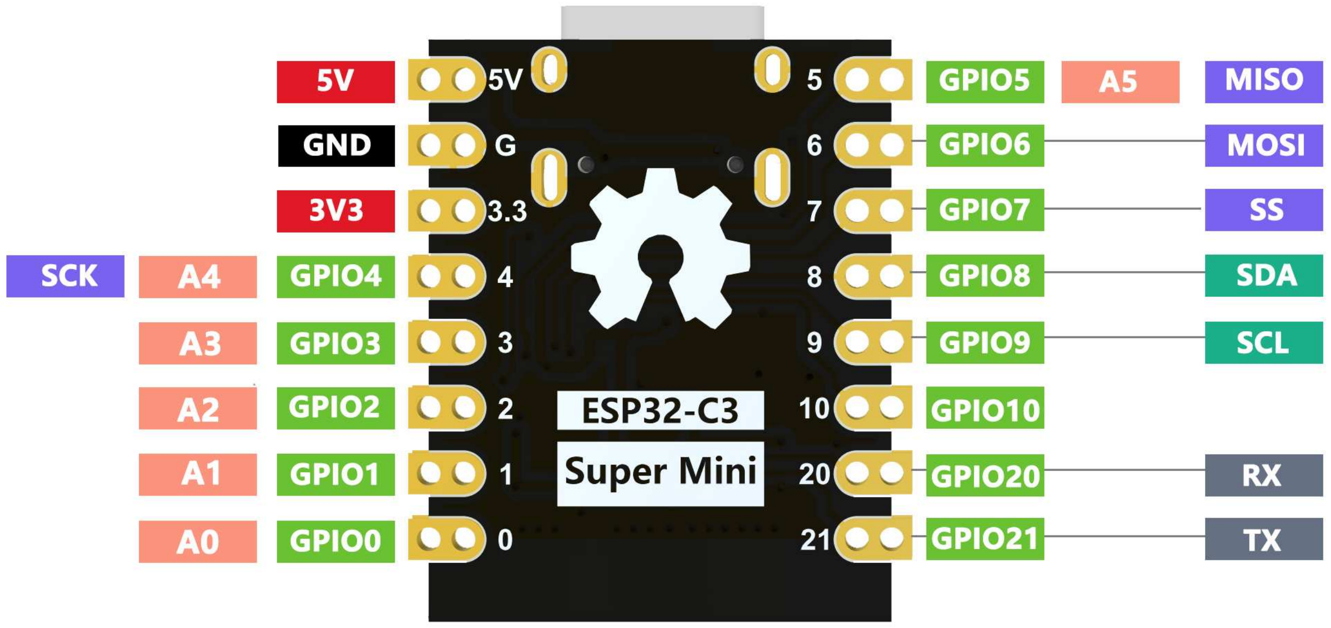
The ESP32 C3 SuperMini is an loT mini development board based on the Espressif ESP32-C3 WiFi/ Bluetoot hdual-mode chip. The ESP32-C3 is a 32-bit RISC-V CPU that contains the FPU (floating point unit) for32-bit single-precision operations with powerful computing power. It has excellent F performance and supportsIEEE 802.11b /g/n WiFi and Bluetooth 5 (LE) protocols. The board comes with an external antenna to enhance signal strength for wireless applications. It also has a small and delicate form factor combined with a single-sided surface mount design. It is equipped with a wealth of interfaces, with 11 digital I/ OS that can be used as PWM pins and 4 analog I/OS that can be usedas ADC pins. It supports four serial interfaces: UART, I2C and SPI. The oard also has a small reset button and a boot loader mode button.

Combined with the above features, the ESP32C3SuperMini is positioned as a high-performance,low-power,cost-effective iot mini development board for low-power iot applications and wireless wearable applications.

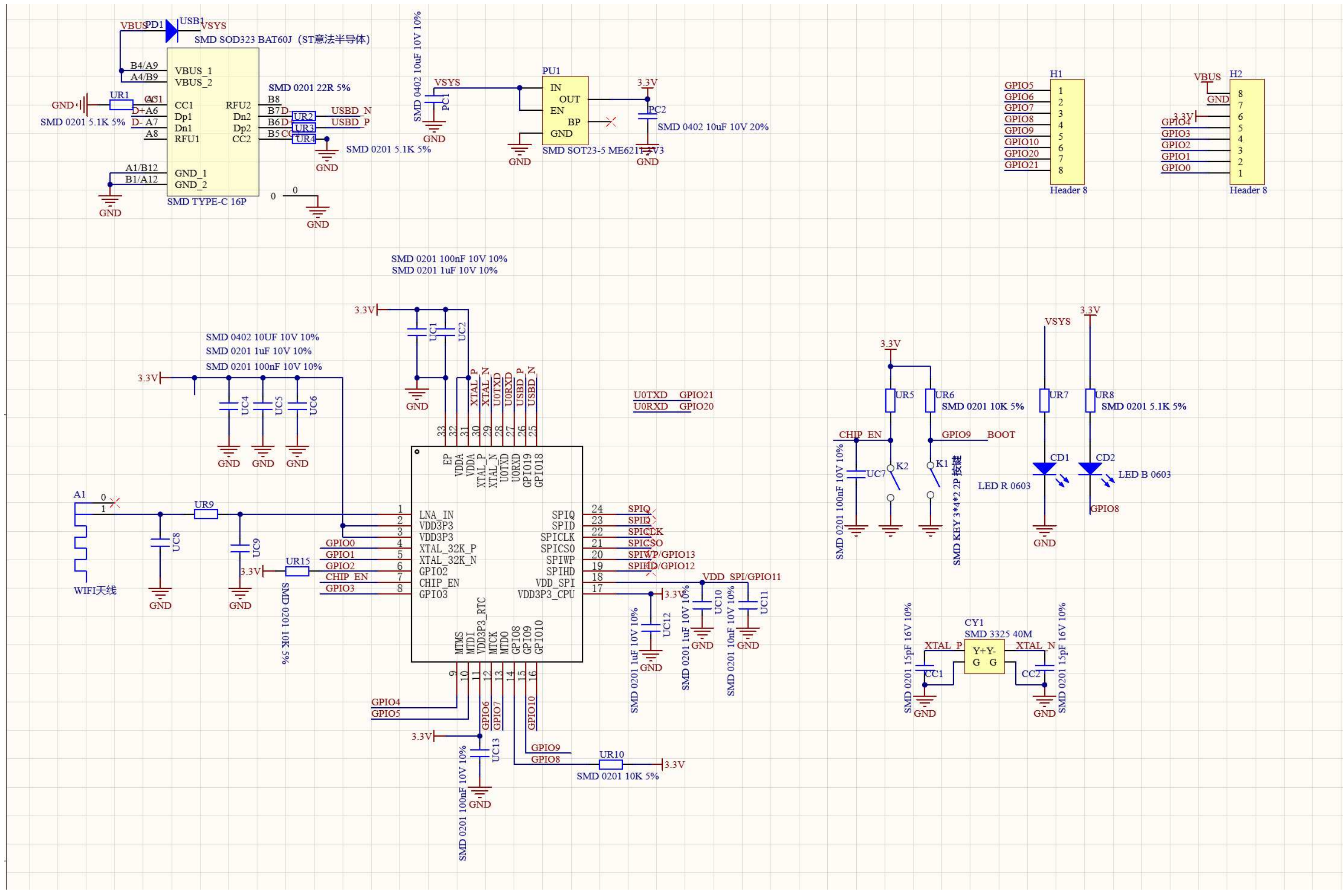
Product parameter:

1. Powerful CPU: ESP32-C3, 32-bit RISC-V single-core processor, running up to 160 MHz
.WiFi: 802.11b/g/n protocol, 2.4GhHz, support Station mode, SoftAP mode, SoftAP+Station mode, hybrid mode
3. Bluetooth: Bluetooth 5.0
4.Ultra-low power consumption: deep sleep power consumption of about 43uA
5.Rich board resources: 400KB SRAM, 384KB ROM built-in 4Mflash.
6.Chip model: ESP32C3FN4
7.Ultra-small size: As small as the thumb (22.52x18mm) classic shape, suitable for wearables and small projects
8.Reliable security features: Encryption hardware accelerators that support AES-128/256, hashing, RSA, HMAC,digital signatures, and secure startup
9.Rich interface: 1x12C, 1xSPI, 2xUART, 11xGPIO(PWM), 4xADC
0.Single-sided components, surf ace mount design
11 Onboard LED blue light: GPIO8 pin





🟩 Pin No. 🟥 Power 🟩 ADC 🟪 SPI ⬛ GND ⬜ UART 🟦 Digital

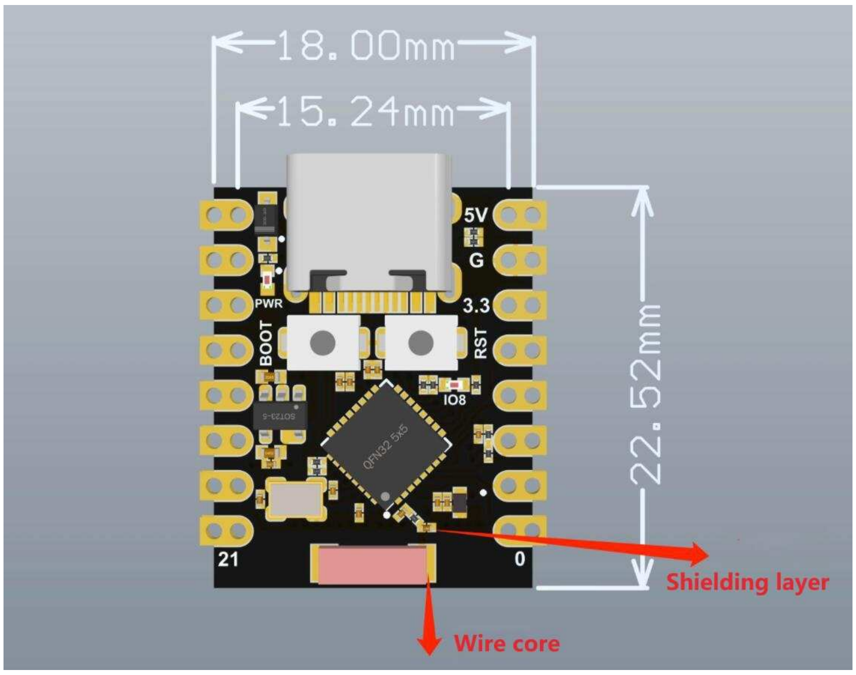


External power supply:

If external power supply is required, just connect the + level of the external power supply to the position of 5V.GND connects to the negative terminal. (Support 3.3 - 6V power supply). Remember that when connecting the external power supply, you cannot access USB. USB and external power supply can only choose one.

When welding, please be careful not to short-circuit the positive and negative electrodes, otherwise it will burn the battery and equipment.

WIFI antenna



Hardware setup

You need to prepare the following:

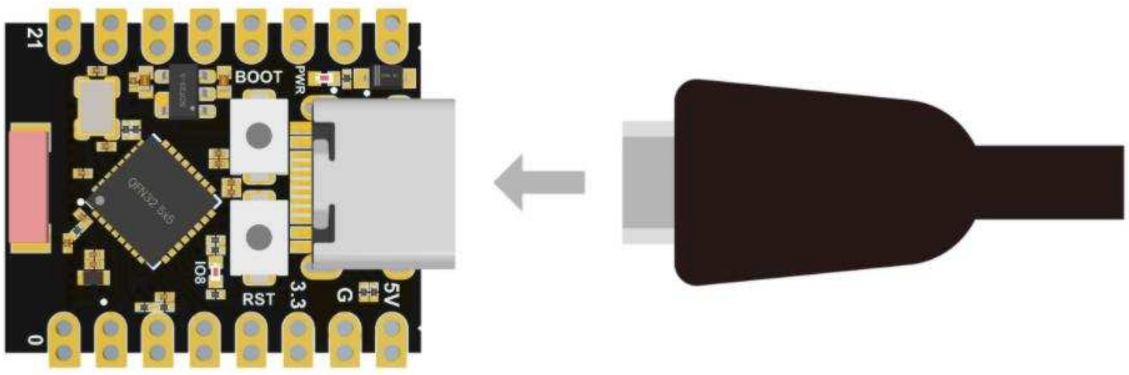
1x ESP32 C3 SuperMini

1 x Computer

1x USB Type-C data cable

Some USB cables can only supply power, not transmit data.

Make sure your USB cable can transfer data.



Software setup

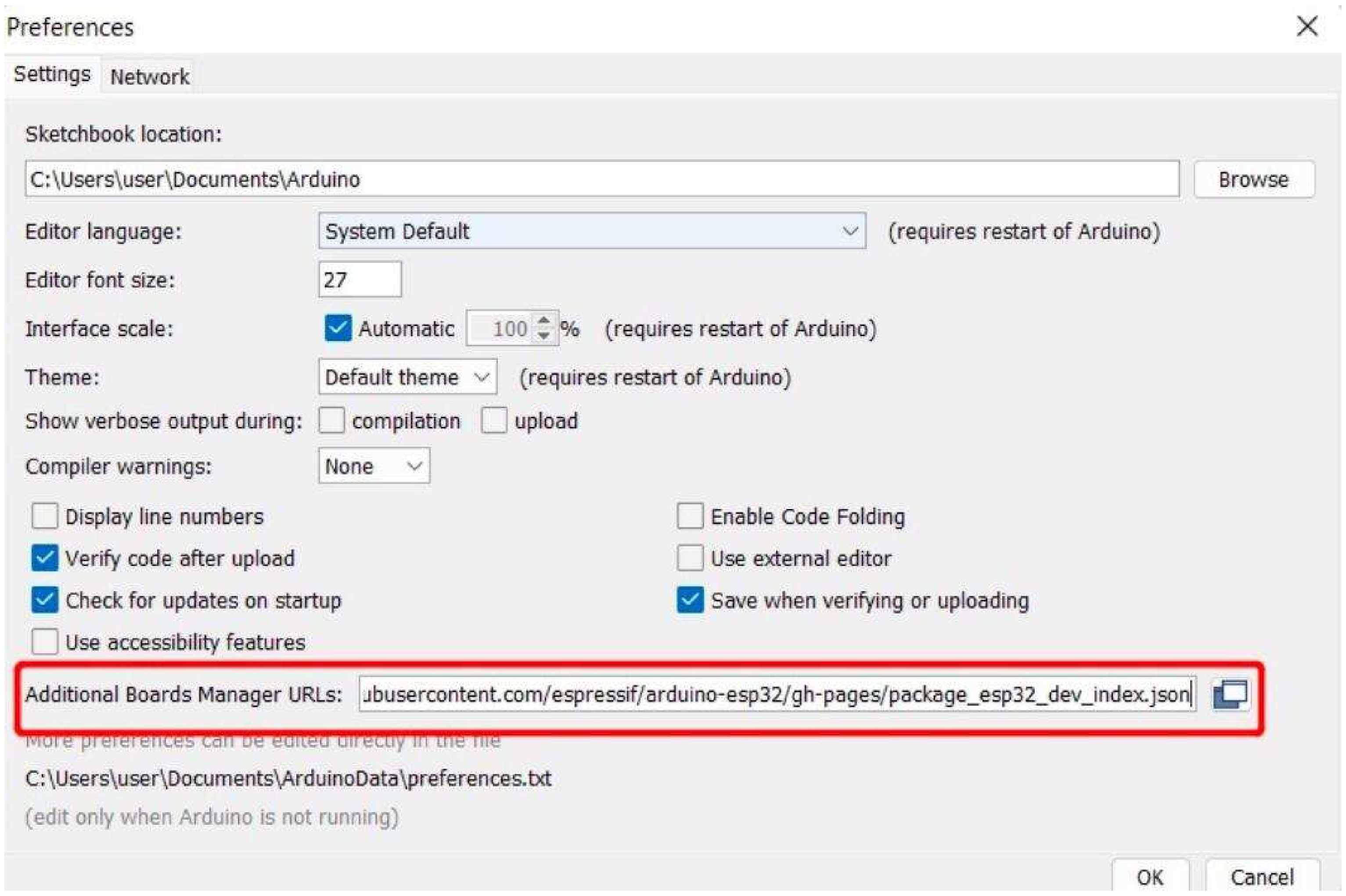
Step 1. Download and install the latest version of IDE based on your operating system.

Download ˙-duino IDE

Step 2. Start the IDE application

Step 3. Add the ESP32 board package to the IDE

Navigate to File ->Preferences, then fill in the "Additional Boards Manager URL" using the following URL:
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json



Preferences

Settings | Network

Sketchbook location: C:\Users\user\Documents\Arduino [Browse]

Editor language: System Default (requires restart of Arduino)

Editor font size: 27

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: ubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json

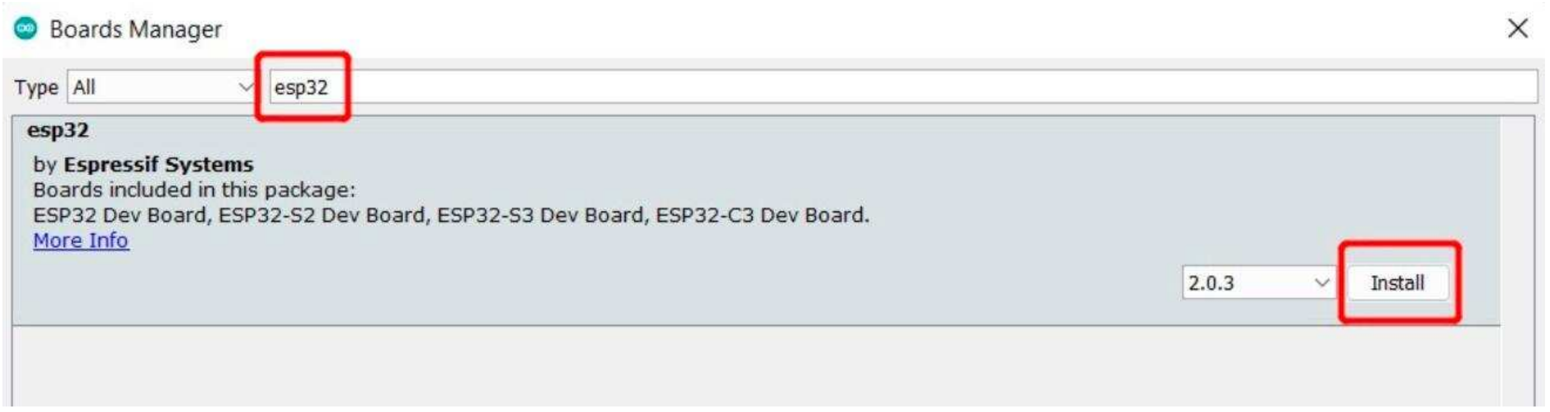
More preferences can be edited directly in the file

C:\Users\user\Documents\ArduinoData\preferences.txt

(edit only when Arduino is not running)

[OK] [Cancel]

Navigate to Tools > Board > Boards Manager... Enter the keyword "esp32" in the search box, select the latest version of esp32 and install it.



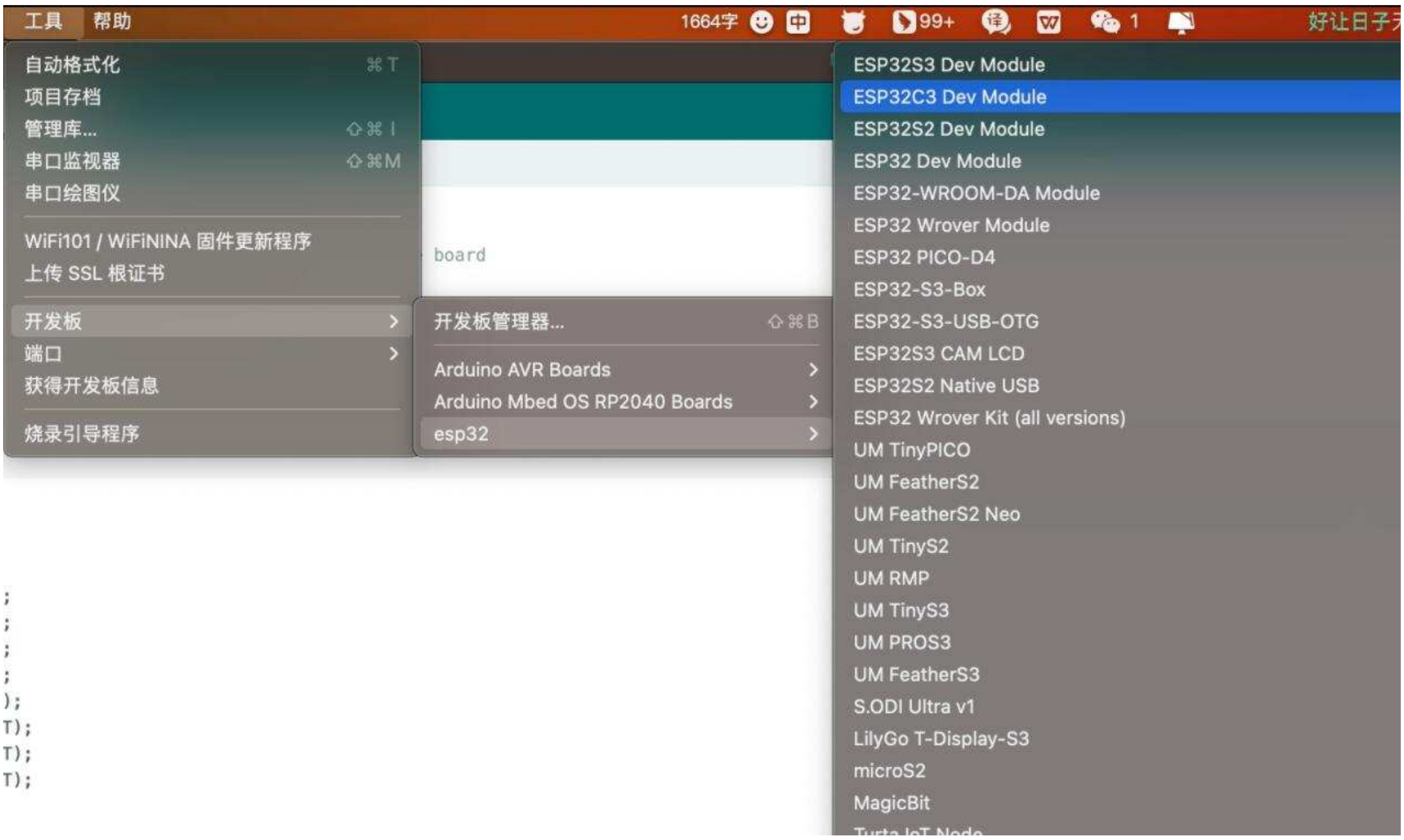
Boards Manager

Type All | esp32

esp32
by **Espressif Systems**
Boards included in this package:
ESP32 Dev Board, ESP32-S2 Dev Board, ESP32-S3 Dev Board, ESP32-C3 Dev Board.
More Info

2.0.3 [Install]

Navigate to Tools > Development board > ESP32 and select ESP32C3 Dev Module. The list of boards is a bit long and you need to scroll to the bottom to get to it.



Navigate to "Tools" > Port, then select the serial port name of the ESP32 C3 SuperMini you are connecting to. This could be COM3 or later (COM1 and COM2 are usually reserved for hardwareserial ports)

Flashing LED

Copy the Following code in the the IDE

```
// define led according to pin diagramint
led = 8;
void setup() {
// initialize digital pin led as an output
pinMode(led, OUTPUT);
}
void loop() {
digitalWrite(led, HIGH);    // turn the LED on
delay(1000);                // wait for a second
digitalWrite(led, LOW);     // turn the LED off
delay(1000);                // wait for a second
}
```

After uploading, you will see the LED flashing on the board with a 1-second delay between each flashing.

FAQ

Com port cannot be recognized on IDE

Enter the download mode:Method 1: Press and hold BOOT to power on.Method

2: Press and hold down the BOOT button of the ESP32C3,

press the RESET button,release the RESET button, and then release the BOOT button. Then the ESP32C3 will

enter download mode. (Each connection needs to re-enter the download mode, sometimes press once, the port

instability will be disconnected, you can judge by the port identification sound)

The program will not run after upload

fter the upload succeeds, you need to press the Reset button to execute the upload.

ESP32 C3 SuperMini serial port cannot print

Set the USB CDC On Boot on the toolbar to Enabled

WiFi function

Connect the ESP32C3SuperMini to your computer using a USB Type-C data cable

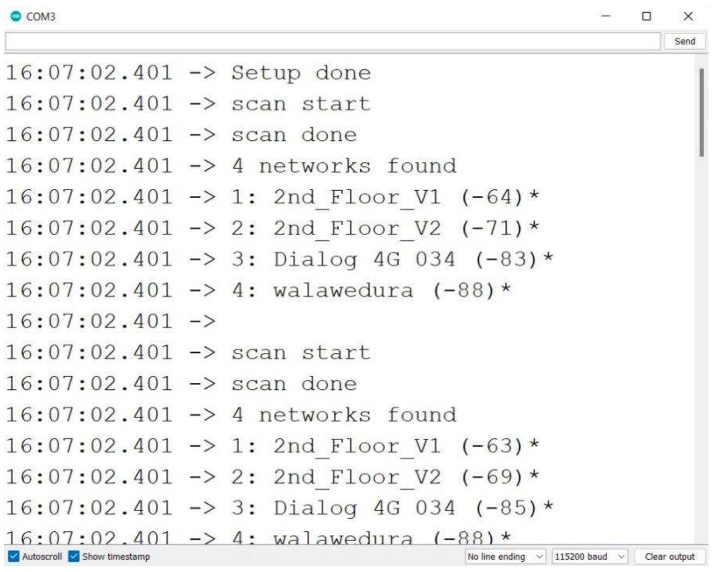
Scan WiFi networks (Station Mode)

We will use the ESP32C3SueprMini to scan the available WiFi networks around it.Here, the boardwill be configured in s tation (STA) mode

1. Copy and paste the following code into the IDE

```
#include "WiFi.h"
void setup()
{
Serial.begin(115200);
// Set WiFi to station mode and disconnect from an AP if it was previously connected
WiFi.mode(WIFI_STA);
WiFi.disconnect();
delay(100);
Serial.println("Setup done");
}
void loop()
{
Serial.println("scan start");
// WiFi.scanNetworks will return the number of networks found
int n = WiFi.scanNetworks();
Serial.println("scan done");
if (n ==0) {
Serial.println("no networks found");
} else {
Serial.print(n);
Serial.println(" networks found");
for (int i = 0; i < n; ++i) {
// Print SSID and RSSI for each network found
Serial.print(i + 1);
Serial.print(": ");
Serial.print(WiFi.SSID(i));
Serial.print(" (");
Serial.print(WiFi.RSSI(i));Serial.print(")");
Serial.println((WiFi.encryptionType(i) == WIFI_AUTH_OPEN)?" ":"*");
delay(10);}
}
Serial.println("");
// Wait a bit before scanning again
delay(5000);}
```

2. Upload the code and turn on the serial monitor to start scanning the WiFi network



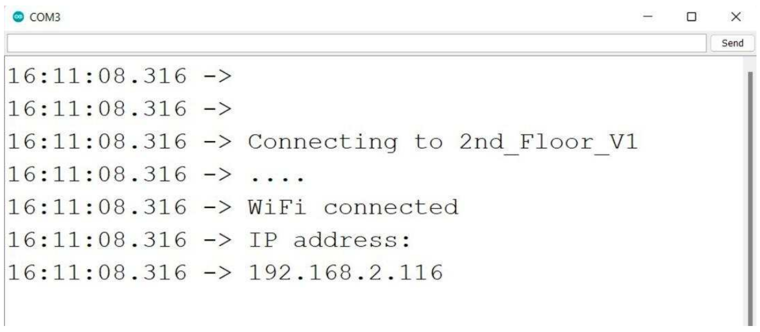
```
16:07:02.401 -> Setup done
16:07:02.401 -> scan start
16:07:02.401 -> scan done
16:07:02.401 -> 4 networks found
16:07:02.401 -> 1: 2nd_Floor_V1 (-64)*
16:07:02.401 -> 2: 2nd_Floor_V2 (-71)*
16:07:02.401 -> 3: Dialog 4G 034 (-83)*
16:07:02.401 -> 4: walawedura (-88)*
16:07:02.401 ->
16:07:02.401 -> scan start
16:07:02.401 -> scan done
16:07:02.401 -> 4 networks found
16:07:02.401 -> 1: 2nd_Floor_V1 (-63)*
16:07:02.401 -> 2: 2nd_Floor_V2 (-69)*
16:07:02.401 -> 3: Dialog 4G 034 (-85)*
16:07:02.401 -> 4: walawedura (-88)*
```

Connect to WiFi network

1. Copy and paste the following code into the IDE

```
#include <WiFi.h>
const char* ssid = "your-ssid"; //your WiFi Name
const char* password = "your-password"; //your WiFi
passwordvoid setup()
{
Serial.begin(115200);
delay(10);
// We start by connecting to a WiFi network
Serial.println();
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}
void loop()
{
}
```

2. Upload the code and turn on the serial monitor to check whether the development board isconnected to the WiF| network



```
16:11:08.316 ->
16:11:08.316 ->
16:11:08.316 -> Connecting to 2nd_Floor_V1
16:11:08.316 -> ....
16:11:08.316 -> WiFi connected
16:11:08.316 -> IP address:
16:11:08.316 -> 192.168.2.116
```

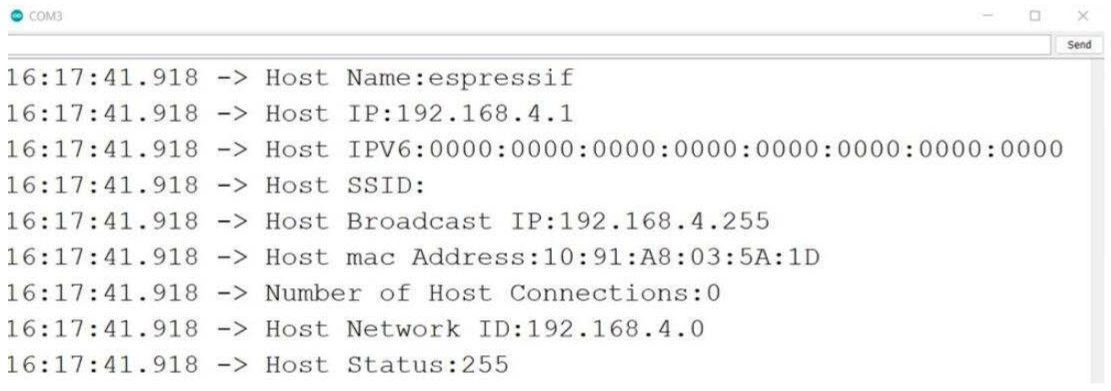
WiFi hotspot

In this example, we will use the ESP32C3SuperMini as a WiFi access point that other devices canconnect to. This is similar to the WiFi hotspot function on your phone.

1. Copy and paste the following code into the IDE

```
#include "WiFi.h"
void setup()
{
Serial.begin(115200);
WiFi.softAP("ESP_AP", "123456789");
}
void loop()
{
Serial.print("Host Name:");
Serial.println(WiFi.softAPgetHostname());
Serial.print("Host IP:");
Serial.println(WiFi.softAPIP());
Serial.print("Host IPV6:");
Serial.println(WiFi.softAPIPv6());
Serial.print("Host SSID:");
Serial.println(WiFi.SSID());
Serial.print("Host Broadcast IP:");
Serial.println(WiFi.softAPBroadcastIP());
Serial.print("Host mac Address:");
Serial.println(WiFi.softAPmacAddress());
Serial.print("Number of HostConnections:");
Serial.println(WiFi.softAPgetStationNum());
Serial.print("Host Network ID:");
Serial.println(WiFi.softAPNetworkID());
Serial.print("Host Status:");
Serial.println(WiFi.status());delay(1000);
}
```

2. Upload the code and turn on the serial monitor to check for more details aboutthe WiFi access point



```
16:17:41.918 -> Host Name:espressif
16:17:41.918 -> Host IP:192.168.4.1
16:17:41.918 -> Host IPV6:0000:0000:0000:0000:0000:0000:0000:0000
16:17:41.918 -> Host SSID:
16:17:41.918 -> Host Broadcast IP:192.168.4.255
16:17:41.918 -> Host mac Address:10:91:A8:03:5A:1D
16:17:41.918 -> Number of Host Connections:0
16:17:41.918 -> Host Network ID:192.168.4.0
16:17:41.918 -> Host Status:255
```

Bluetooth function

Connect the ESP32C3SuperMini to your computer via a USB Type-C cable

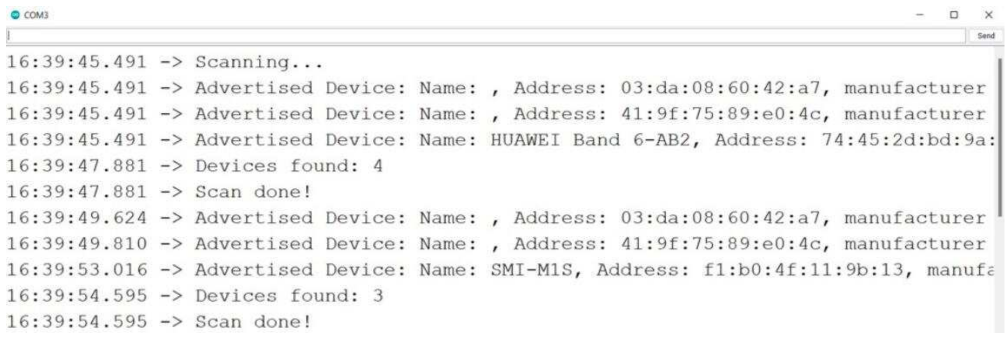
Scan Bluetooth

We will use the ESP32C3SueprMini to scan for available Bluetooth devices around it

1. Copy and paste the following code into the IDE

```
#include <BLEDevice.h>
#include <BLEUtils.h>
#include <BLEScan.h>
#include <BLEAdvertisedDevice.h>
int scanTime = 5; //In seconds
BLEScan* pBLEScan;
class MyAdvertisedDeviceCallbacks: public BLEAdvertisedDeviceCallbacks {void onResult(BLEAdvertisedDevice
advertisedDevice) {Serial.printf("Advertised Device: %s \n", advertisedDevice.toString().c_str());
}
};
void setup() {
Serial.begin(115200);
Serial.println("Scanning...");
BLEDevice::init("");
pBLEScan = BLEDevice::getScan(); //create new scan
pBLEScan->setAdvertisedDeviceCallbacks(new MyAdvertisedDeviceCallbacks());
pBLEScan->setActiveScan(true); //active scan uses more power, but get results faster
pBLEScan->setInterval(100);
pBLEScan->setWindow(99); // less or equal setInterval value
}
void loop() {
// put your main code here, to run repeatedly:
BLEScanResults foundDevices = pBLEScan->start(scanTime, false);
Serial.print("Devices found: ");
Serial.println(foundDevices.getCount());
Serial.println("Scan done!");
pBLEScan->clearResults(); // delete results fromBLEScan buffer to release memory
delay(2000);
}
```

2. Upload the code and turn on the serial monitor to start scanning Bluetooth devices



```
16:39:45.491 -> Scanning...
16:39:45.491 -> Advertised Device: Name: , Address: 03:da:08:60:42:a7, manufacturer
16:39:45.491 -> Advertised Device: Name: , Address: 41:9f:75:89:e0:4c, manufacturer
16:39:45.491 -> Advertised Device: Name: HUAWEI Band 6-AB2, Address: 74:45:2d:bd:9a:
16:39:47.881 -> Devices found: 4
16:39:47.881 -> Scan done!
16:39:49.624 -> Advertised Device: Name: , Address: 03:da:08:60:42:a7, manufacturer
16:39:49.810 -> Advertised Device: Name: , Address: 41:9f:75:89:e0:4c, manufacturer
16:39:53.016 -> Advertised Device: Name: SMI-M1S, Address: f1:b0:4f:11:9b:13, manufa
16:39:54.595 -> Devices found: 3
16:39:54.595 -> Scan done!
```

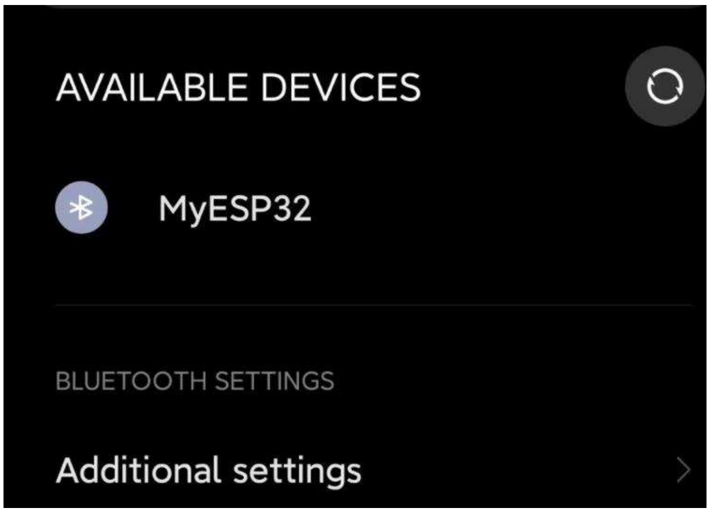
As a Bluetooth server

In this example, we will use the ESP32C3SuperMini as the Bluetooth server. Here we will use a smartp hone to search the ESP32C3SuperMini board and send a string todisplay on the serial monitor

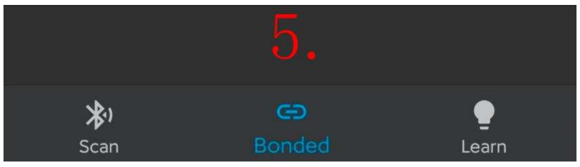
1. Copy and paste the following code into the IDE

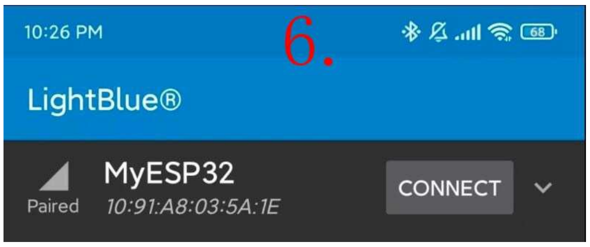
```
#include <BLEDevice.h>
#include <BLEUtils.h>
#include <BLEServer.h>
// See the following for generating UUIDs:
/ https://www.uuidgenerator.net/
#define SERVICE_UUID "4fafc201-1fb5-459e-8fcc-c5c9c331914b"
#define CHARACTERISTIC_UUID "beb5483e-36e1-4688-b7f5-ea07361b26a8"
class MyCallbacks: public BLECharacteristicCallbacks {
void onWrite(BLECharacteristic *pCharacteristic) {
std::string value = pCharacteristic->getValue();
if (value.length() > 0) {
Serial.println("*********");
Serial.print("New value: ");
for (int i = 0; i < value.length(); i++)
Serial.print(value[i]);
Serial.println();
Serial.println("*********");
}
}
};
void setup() {
Serial.begin(115200);
BLEDevice::init("MyESP32");
BLEServer *pServer = BLEDevice::createServer();
BLEService *pService = pServer->createService(SERVICE_UUID);
BLECharacteristic *pCharacteristic = pService->createCharacteristic(
CHARACTERISTIC_UUID,
BLECharacteristic::PROPERTY_READ |
BLECharacteristic::PROPERTY_WRITE
);
pCharacteristic->setCallbacks(new MyCallbacks());
pCharacteristic->setValue("Hello World");
pService->start();
BLEAdvertising *pAdvertising = pServer->getAdvertising();
pAdvertising->start();
}
void loop() {
// put your main code here, to run repeatedly:
delay(2000);
}
```

2. Upload the code and open the serial monitor
3. Download and install the LightBlue app on your smartphone
4. Turn on the Bluetooth of the phone, place the phone near the ESP32C3SuperMini, scan the device and connect the MyESP32 device

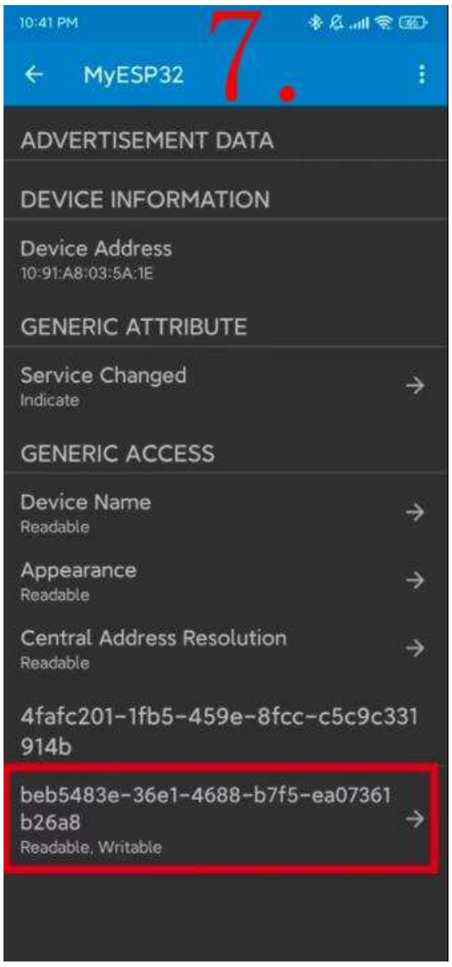


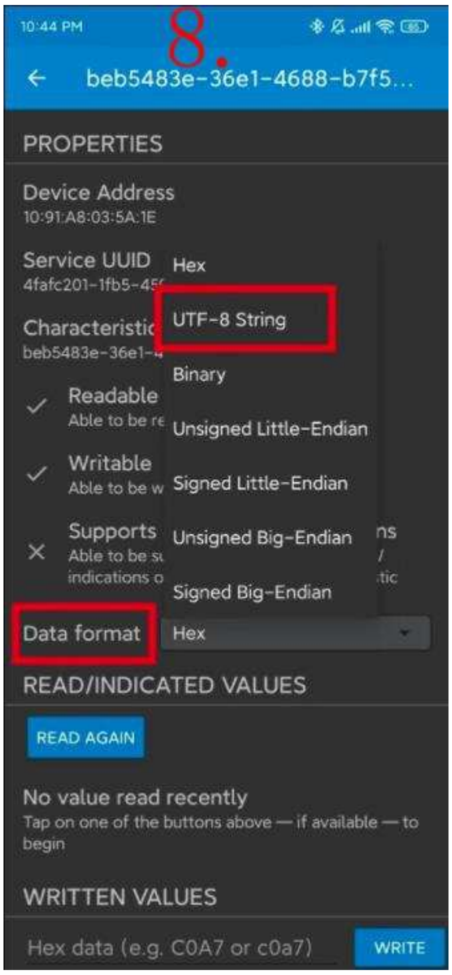
5. Open the LightBlue application and click the Bonded TAB
6. Click CONNECT next to MyESP32



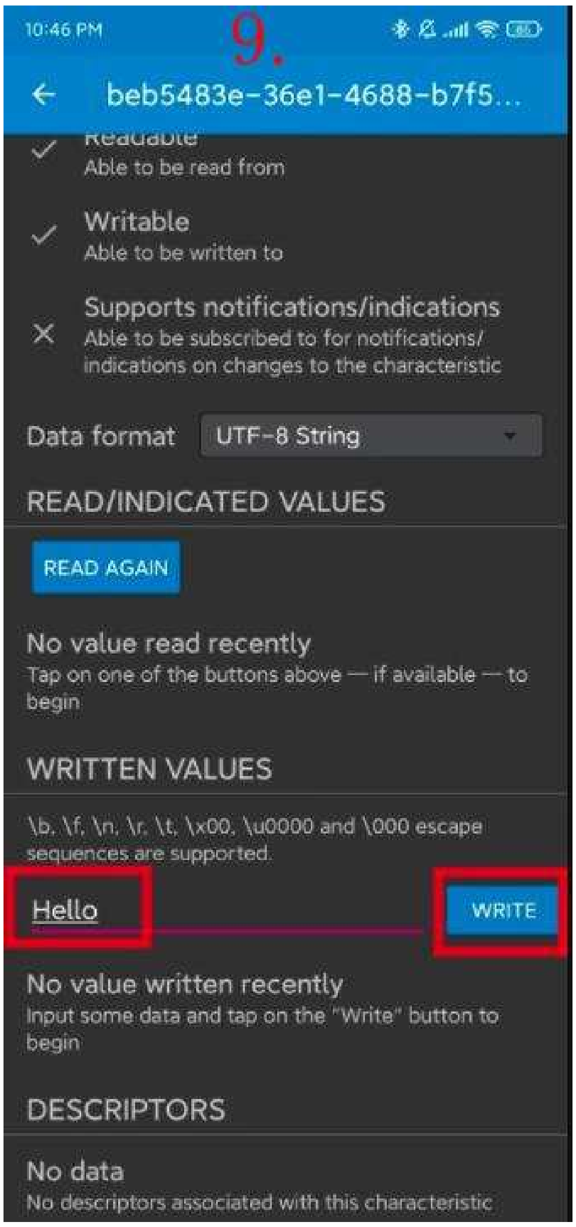


7. Click at the bottom where Readable, Writable are displayed
8. Under the Data Format drop-down menu, select UTF-8 string





9. Type "Hello" under "WRITTEN VALUES" and click "WRITE"


10:46 PM

9.

← beb5483e-36e1-4688-b7f5...

✓ Readable
Able to be read from

✓ Writable
Able to be written to

✕ Supports notifications/indications
Able to be subscribed to for notifications/indications on changes to the characteristic

Data format: UTF-8 String

READ/INDICATED VALUES

READ AGAIN

No value read recently
Tap on one of the buttons above — if available — to begin

WRITTEN VALUES

\b, \f, \n, \r, \t, \x00, \u0000 and \000 escape sequences are supported.

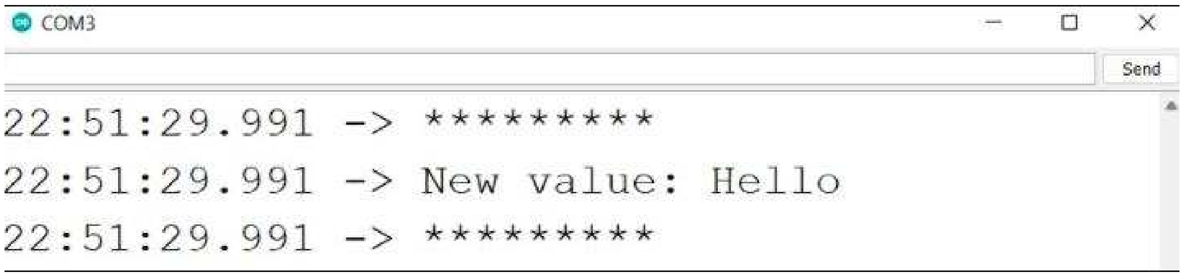
Hello WRITE

No value written recently
Input some data and tap on the "Write" button to begin

DESCRIPTORS

No data
No descriptors associated with this characteristic

:) You will see the text string"Hello" output on the serial monitor of the IDE


COM3 — Send

```
22:51:29.991 -> *********
22:51:29.991 -> New value: Hello
22:51:29.991 -> *********
```

ChatGPT

We can use the ESP32 Supermini to do some applications in ChatGPT.
For example, we can configure our own ChatGPT Q&A page using the SuperMini.
In this page, you can enter your question, the ESP32C3SuperMini will record your question, and using the API call method provided by OpenAl, using HTTP Client to send a request command, get the answer of ChatGPT and print it in the serial port.

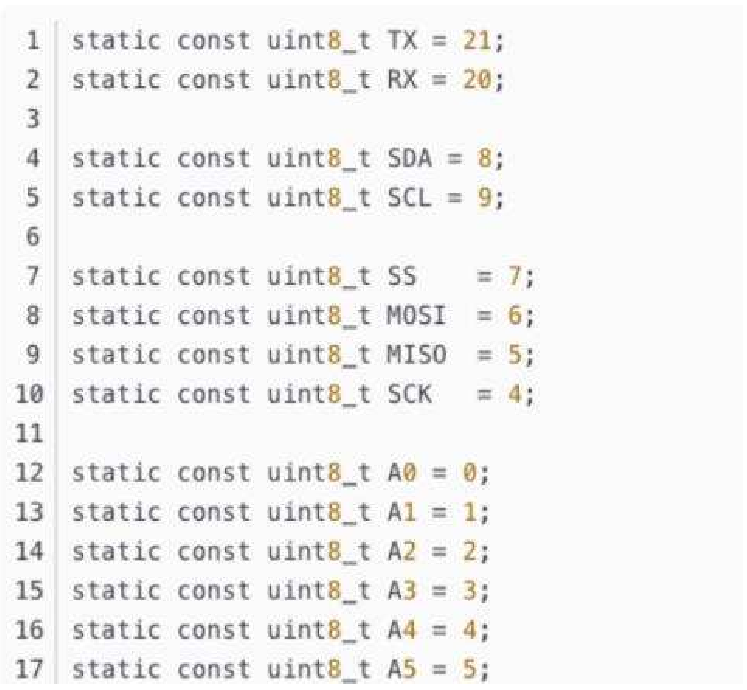
The steps are as follows:

Connect the ESP32C3SuperMini to the network
Building embedded web pages
Submit questions via the built-in web page
Get answers from ChatGPT
If you are interested, you can search for related materials.

Pin use

The ESP32C3SuperMini has various interfaces. There are 11 digital I/ OS that can be used as PWM pins and 4 analog inputs that can be used as ADC pins. It supports four serial communication interface s such as UART, I2C, SPI and 12S. This article will help you understand these interfaces and impleme nt them in your next project!

About pin A0A5, GPIOOGPIO10 (010), and the beginning of D, here to explain, the default motherboard only GPIO beginning is 010, 20, 21, A0~A5 pin is a mapping problem, in order to facilitate the user to tell the function of this pin is analog pin or digital pin. When the Arduino selects the development board type and selects the ESP32C3 Dev Module, you can reference its pin map. The pin map is shown below:



```
static const uint8_t TX = 21;
static const uint8_t RX = 20;

static const uint8_t SDA = 8;
static const uint8_t SCL = 9;

static const uint8_t SS    = 7;
static const uint8_t MOSI  = 6;
static const uint8_t MISO  = 5;
static const uint8_t SCK   = 4;

static const uint8_t A0 = 0;
static const uint8_t A1 = 1;
static const uint8_t A2 = 2;
static const uint8_t A3 = 3;
static const uint8_t A4 = 4;
static const uint8_t A5 = 5;
```

Digital pin

Upload the code to the board, and the on-board LED will light up every second.

```
// define led according to pin diagram
int led = 8;void setup() {
// initialize digital pin led as an output
pinMode(led, OUTPUT);
}
void loop() {digitalWrite(led, HIGH); // turn the LED on
delay(1000); // wait for a second
digitalWrite(led, LOW); // turn the LED off
delay(1000); // wait for a second
}
```

Digital PWM

Upload the following code to see the on-board LED gradually dim.

```
int ledPin = 8; // LED connected to digital pin 10
void setup() {
// declaring LED pin as output
pinMode(ledPin, OUTPUT);
}
void loop() {
// fade in from min to max in increments of 5 points:
for (int fadeValue = 0 ; fadeValue <= 255; fadeValue += 5) {
// sets the value (range from 0 to 255):
analogWrite(ledPin, fadeValue);
// wait for 30 milliseconds to see the dimming effect
delay(30);
}
// fade out from max to min in increments of 5 points:
for (int fadeValue = 255 ; fadeValue >= 0; fadeValue -= 5) {
// sets the value (range from 0 to 255):
analogWrite(ledPin, fadeValue);
// wait for 30 milliseconds to see the dimming effect
delay(30);
}
}
```

Analog pin

Connect the potentiometer to pin A5 and upload the following code to control the flashing interval of the LED by turning the potentiometer knob.

```
const int sensorPin = A5;
const int ledPin = 8;
void setup() {
pinMode(sensorPin, INPUT); // declare the sensorPin as an INPUT
pinMode(ledPin, OUTPUT); // declare the ledPin as an OUTPUT
}
void loop() {
// read the value from the sensor:
int sensorValue = analogRead(sensorPin);
// turn the ledPin on
digitalWrite(ledPin, HIGH);
// stop the program for <sensorValue> milliseconds:
delay(sensorValue);
// turn the ledPin off:
digitalWrite(ledPin, LOW);
// stop the program for for <sensorValue> milliseconds:
delay(sensorValue);
}
```

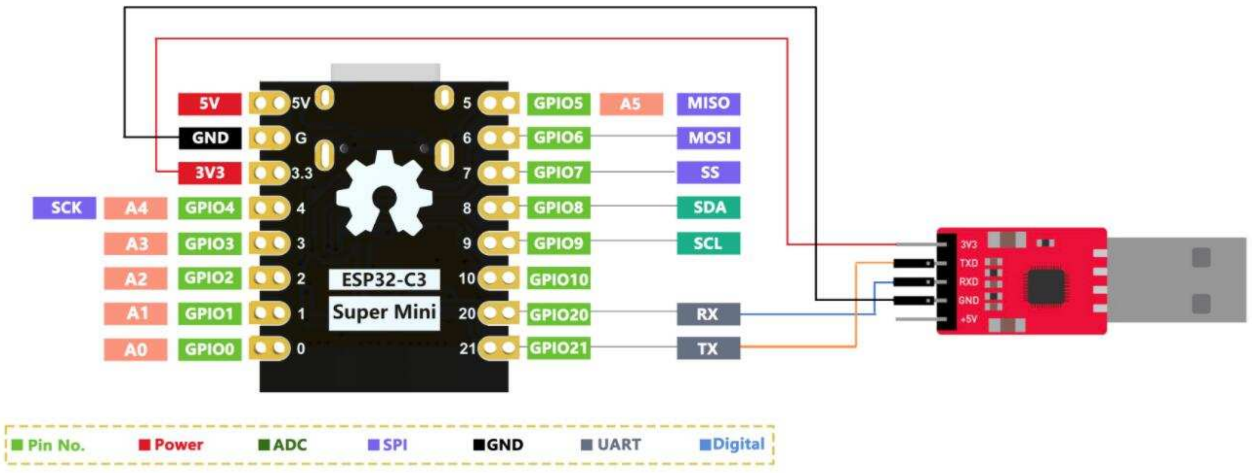
Serial port

Hardware serial port, there are two hardware serial ports on the board:

USB serial port

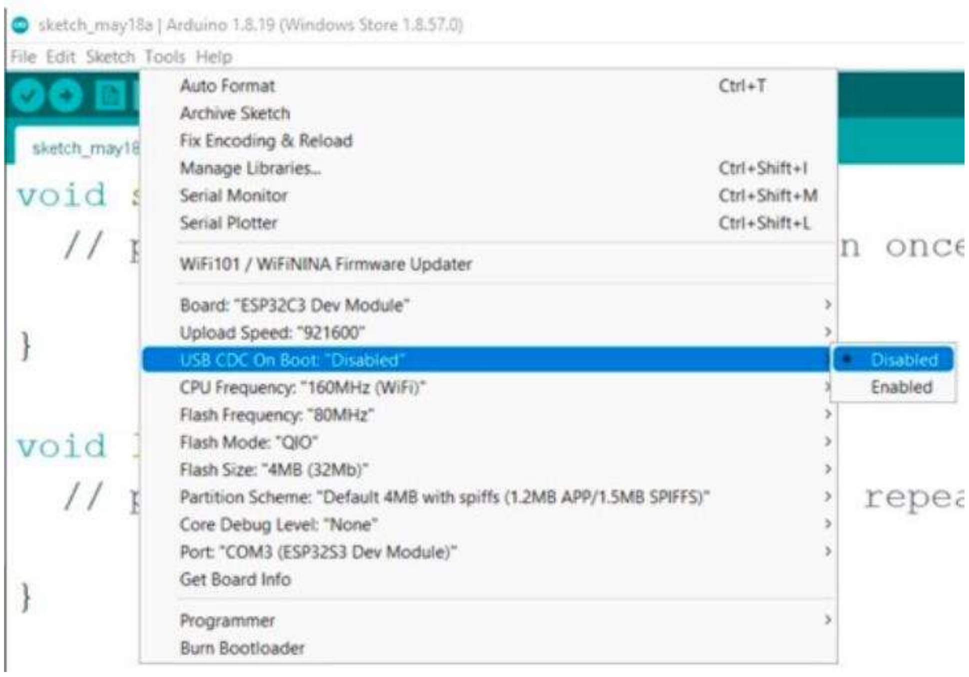
ART serial port

By default, USB serial is enabled, which means you can connect the development board to a PC via USB Type-C and turn on the serial monitor on the Arduino IDE to see the data sent via serial.



However, if you want to use ART as a serial port, you will need to connect pin 20 as a TX pin and pin 21 as an RX pin using a USB serial adapter.

Also, you need to set USB CDC On Boot to disabled from the Arduino IDE.


sketch_may18a | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help

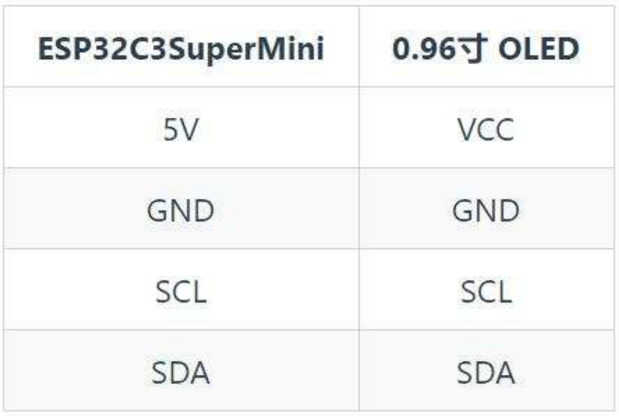
```
Auto Format                                   Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries...                           Ctrl+Shift+I
Serial Monitor                                Ctrl+Shift+M
Serial Plotter                                Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "ESP32C3 Dev Module"
Upload Speed: "921600"
USB CDC On Boot: "Disabled"                   • Disabled
                                                Enabled
CPU Frequency: "160MHz (WiFi)"
Flash Frequency: "80MHz"
Flash Mode: "QIO"
Flash Size: "4MB (32Mb)"
Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"
Core Debug Level: "None"
Port: "COM3 (ESP32S3 Dev Module)"
Get Board Info
Programmer
Burn Bootloader
```

Software serial port

If you want to use more serial ports, you need to use the SoftwareSerial library tocreate soft serial ports

I2C

Connection of ESP32 C3 Supermini and 0.96 OLED



ESP32C3SuperMini	0.96寸 OLED
5V	VCC
GND	GND
SCL	SCL
SDA	SDA

1. Open Arduino IDE and navigate to Sketch ->Include Library-Manage Libraries...
2. Search for u8g2 and install it
3. Upload the code to display the text string on the OLED display