Manuals+ — User Manuals Simplified.

M5STACK ESP32 Devolopment Board Kit Instructions

Home » M5STACK » **M5STACK ESP32 Devolopment Board Kit Instructions**

M5STACK

ESP32 Development Board Kit Instructions

Contents

OUTLINE

Atomy is a very small and flexible IoT speech recognition development board, using Espressif's `ESP32` main control chip, equipped with two low-power `Xtensa® 32-bit LX6` microprocessors, main frequency Up to `240MHz`. It has the characteristics of compact size, strong performance and low power consumption. Integrated USB-A interface, plug and play, easy to upload, download and debug the program. Integrated `Wi-Fi` and `Bluetooth` modules, with built-in digital microphone SPM1423 (I2S), can achieve clear audio recording, suitable for various IoT human-computer interaction, voice input recognition scenarios (STT)

1.1.ESP32 PICO

The ESP32-PICO-D4 is a System-in-Package (SiP) module that is based on ESP32, providing complete Wi-Fi and Bluetooth functionalities. The module has a size as small as (7.000±0.100) mm × (7.000±0.100) mm × (0.940±0.100) mm, thus requiring minimal PCB area. The module integrates a 4-MB SPI flash. At the core of this module is the ESP32 chip*, which is a single 2.4 GHz Wi-Fi and Bluetooth combo chip designed with TSMC's 40 nm ultra-low power technology. ESP32-PICO-D4 integrates all peripheral components seamlessly, including a crystal oscillator, flash, filter capacitors, and RF matching links in one single package. Given that no other peripheral components are involved, module welding and testing are not required either. As such, ESP32-PICO-D4 reduces the complexity of the supply chain and improves control efficiency. With its ultra-small size, robust performance, and low-energy consumption, ESP32PICO-D4 is well suited for any space-limited or battery-operated applications, such as wearable electronics, medical equipment, sensors and other IoT products.

SPECIFICATIONS

Resources	I Parameter
ESP32-PICO-D4	240MHz dual-core, 600 DMIPS, 520KB SRAM, 2.4GHz Wi-Fi, dual-mode Blue tooth
Flash	j 4MB
Input voltage	5V @ 500mA
button	Programmable buttons x 1
Programmable RGB LED	SK6812 x 1
Antenna	2.4GHz 3D Antenna
Operating temperature	32°F to 104°F (0°C to 40°C)

QUICKSTART

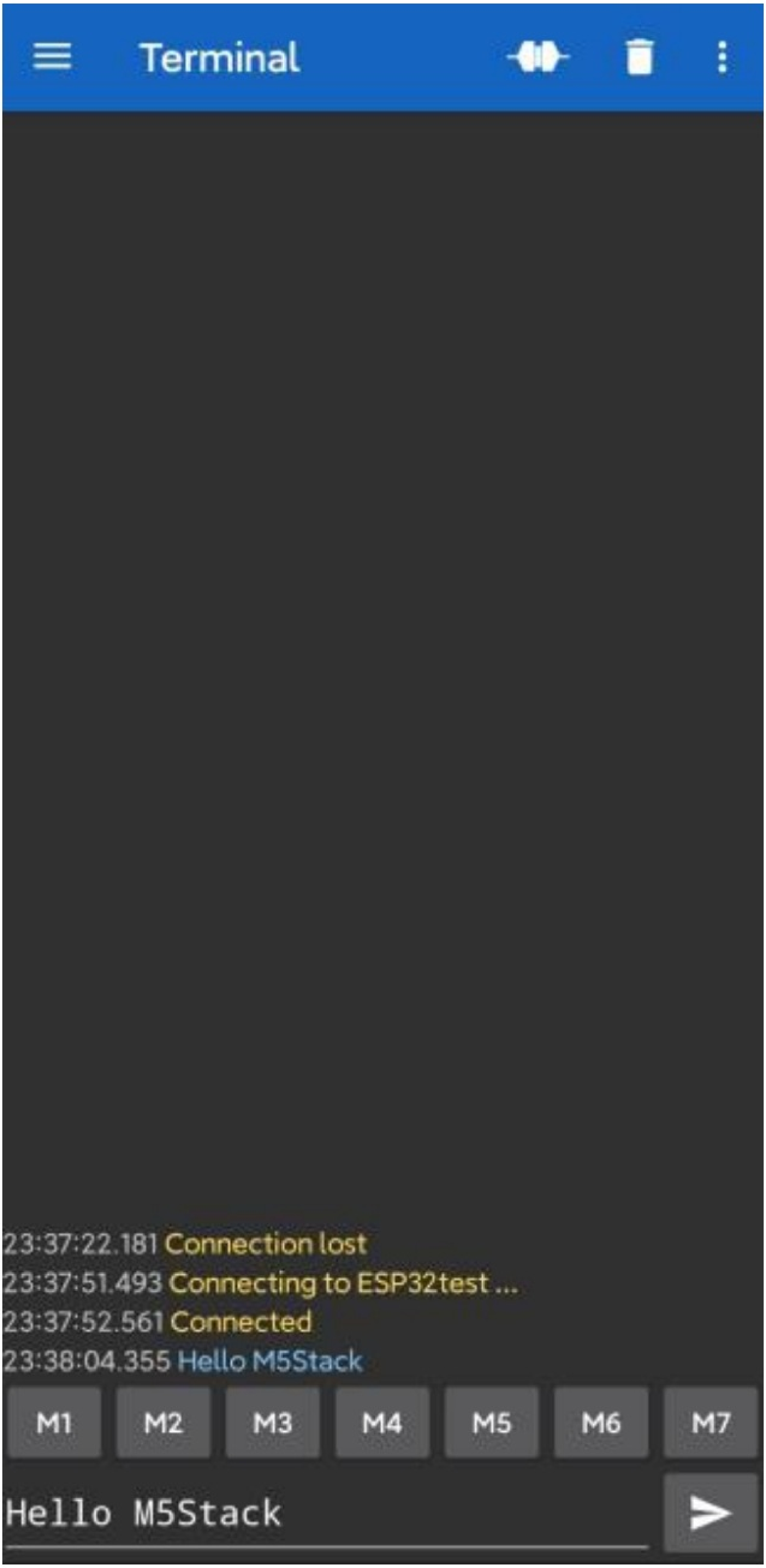
3.1.ARDUINO IDE

Visit Arduino’s official website(https://www.arduino.cc/en/Main/Software), select the installation package for your own operating system to download.

1. Open up Arduino IDE, navigate to `File`->` Peferences`->`Settings`
2. Copy the following M5Stack Boards Manager URL to `Additional Boards Manager URLs:`
   https://raw.githubusercontent.com/espressif/arduino-esp32/ghpages/package_esp32_dev_index.json
3. Navigate to `Tools`->` Board:`->` Boards Manager…`
4. Search `ESP32` in the pop-up window, find it and click `Install`
5. select `Tools`->` Board:`->`ESP32-Arduino-ESP32 DEV Module
6. Please install the FTDI driver before use: https://docs.m5stack.com/en/download

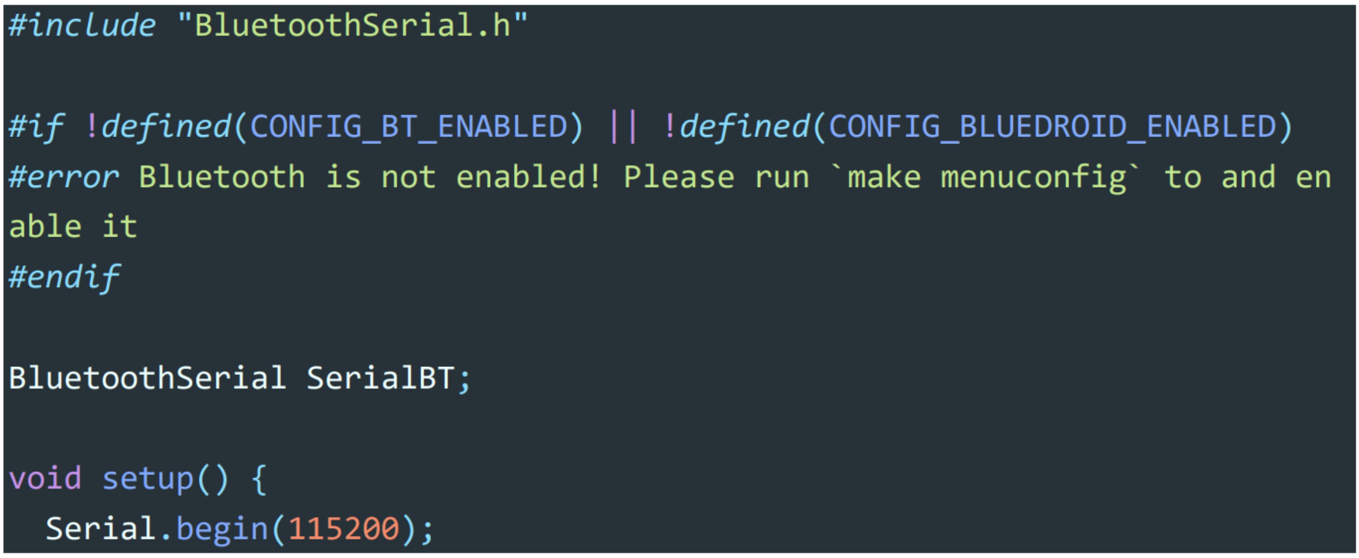
3.2.BLUETOOTH SERIAL

Open the Arduino IDE and open the example program ` File`->` Examples`->`BluetoothSerial`->`SerialToSerialBT`. Connect the device to the computer and select the corresponding port to burn. After completion, the device will automatically run Bluetooth, and the device name is `ESP32test`. At this time, use the Bluetooth serial port sending tool on the PC to realize the transparent transmission of Bluetooth serial data.

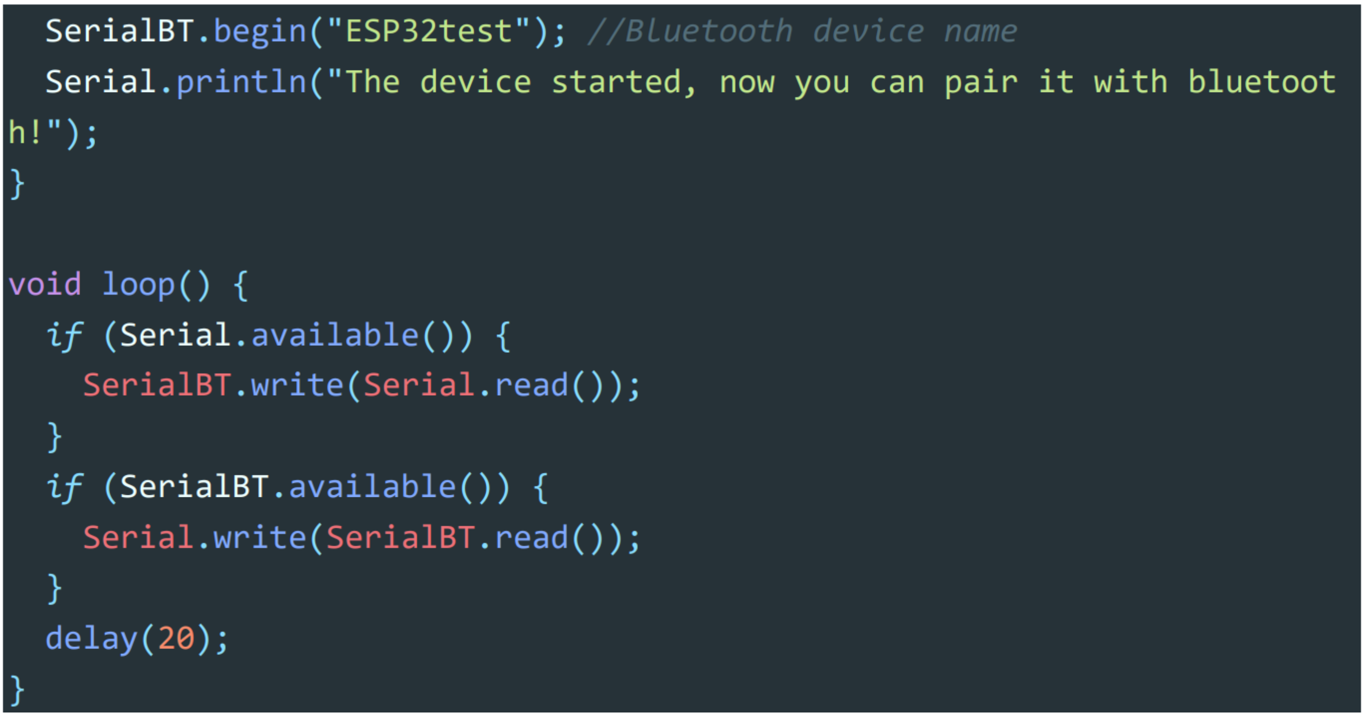


```
23:37:22.181 Connection lost
23:37:51.493 Connecting to ESP32test ...
23:37:52.561 Connected
23:38:04.355 Hello M5Stack
```

Hello M5Stack



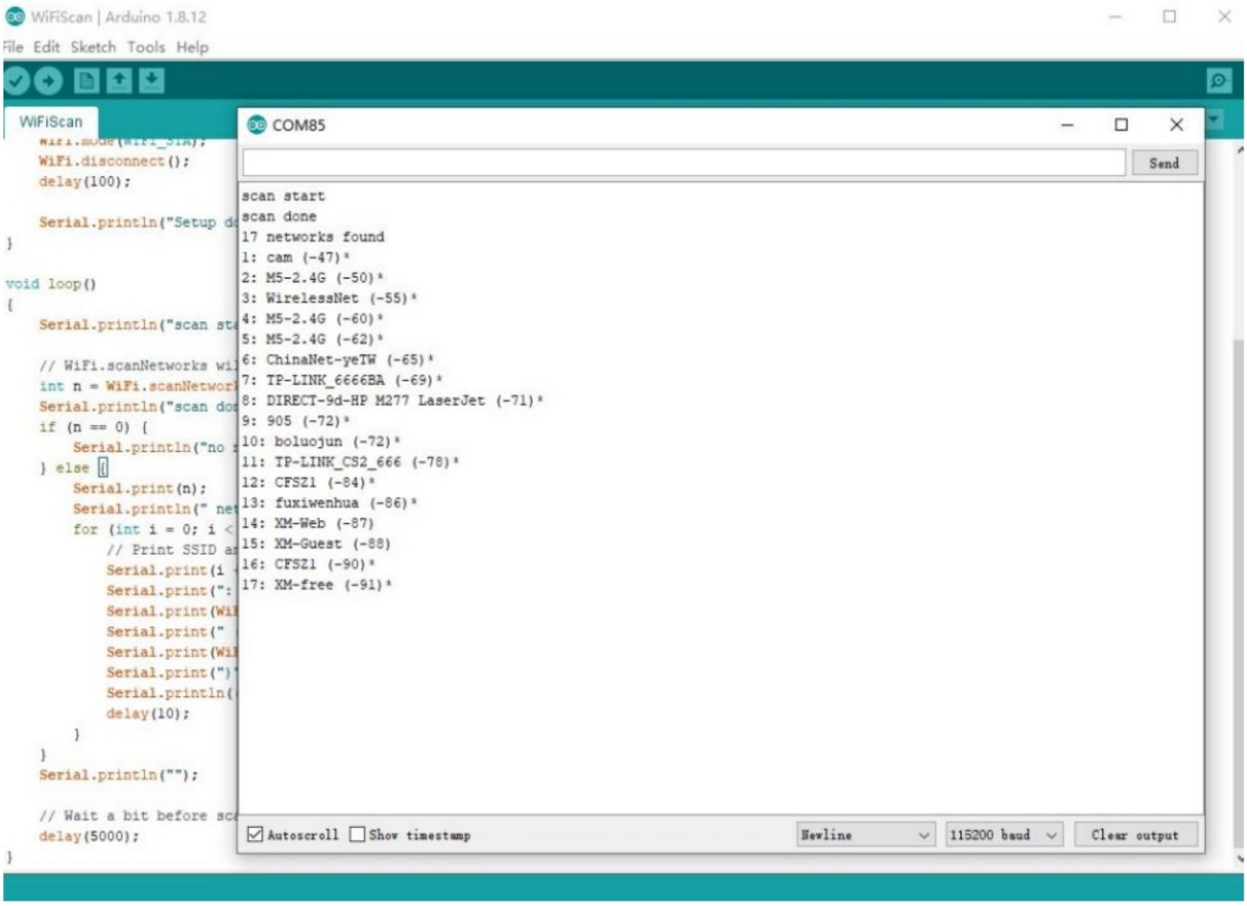
```
#include "BluetoothSerial.h"

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and en
able it
#endif

BluetoothSerial SerialBT;

void setup() {
  Serial.begin(115200);
```



```
  SerialBT.begin("ESP32test"); //Bluetooth device name
  Serial.println("The device started, now you can pair it with bluetoot
h!");
}

void loop() {
  if (Serial.available()) {
    SerialBT.write(Serial.read());
  }
  if (SerialBT.available()) {
    Serial.write(SerialBT.read());
  }
  delay(20);
}
```

3.3.WIFI SCANNING

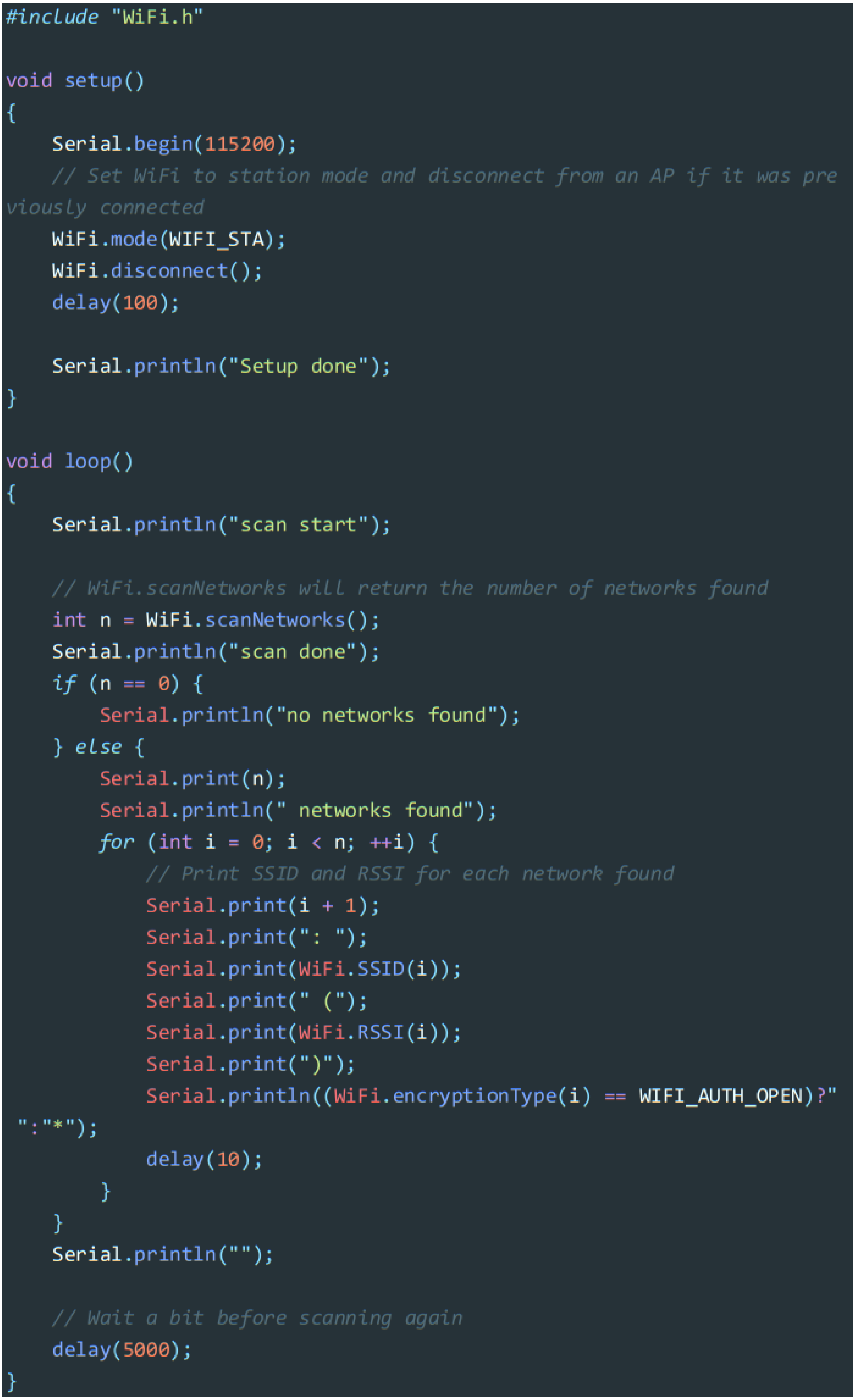
Open the Arduino IDE and open the example program `File`->` Examples`->`WiFi`->` WiFiScan`. Connect the device to the computer and select the corresponding port to burn. After completion, the device will automatically run the WiFi scan, and the current WiFi scan result can be obtained through the serial port monitor that comes with the Arduino.



```
    WiFi.mode(WIFI_STA);
    WiFi.disconnect();
    delay(100);

    Serial.println("Setup d
}

void loop()
{
    Serial.println("scan st

    // WiFi.scanNetworks wi
    int n = WiFi.scanNetwor
    Serial.println("scan do
    if (n == 0) {
        Serial.println("no
    } else {
        Serial.print(n);
        Serial.println(" ne
        for (int i = 0; i <
            // Print SSID a
            Serial.print(i
            Serial.print(":
            Serial.print(Wi
            Serial.print("
            Serial.print(Wi
            Serial.print(")
            Serial.println(
            delay(10);
        }
    }
    Serial.println("");

    // Wait a bit before sc
    delay(5000);
}
```

```
scan start
scan done
17 networks found
1: cam (-47)*
2: M5-2.4G (-50)*
3: WirelessNet (-55)*
4: M5-2.4G (-60)*
5: M5-2.4G (-62)*
6: ChinaNet-yeTW (-65)*
7: TP-LINK_6666BA (-69)*
8: DIRECT-9d-HP M277 LaserJet (-71)*
9: 905 (-72)*
10: boluojun (-72)*
11: TP-LINK_CS2_666 (-78)*
12: CFSZ1 (-84)*
13: fuxiwenhua (-86)*
14: XM-Web (-87)
15: XM-Guest (-88)
16: CFSZ1 (-90)*
17: XM-free (-91)*
```

☑ Autoscroll ☐ Show timestamp Newline 115200 baud Clear output



```
#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);
}
```

Federal Communications Commission (FCC) Statement
You are cautioned that changes or modifications not expressly approved by the party responsible for compliance could void the user’s authority to operate the equipment.
This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: 1) this device may not cause harmful interference, and 2) this device must accept any interference received, including interference that may cause undesired operation of the device.
FCC RF Radiation Exposure Statement:
The product complies with the FCC portable RF exposure limit set forth for an uncontrolled environment and is safe for intended operation as described in this manual. Further RF exposure reduction can be achieved if the product can be kept as far as possible from the user body.

Documents / Resources

	M5STACK ESP32 Devolopment Board Kit [pdf] Instructions M5ATOMU, 2AN3WM5ATOMU, ESP32 Devolopment Board Kit, ESP32, Devolopment Board Kit