M5 STACK Flow Connect Software User Guide

M5 STACK Flow Connect Software

Contents

OUTLINE

The Flow Connect is a highly integrated industrial controller designed for complex automation and communication environments. It features the ESP32-S3R8 microcontroller at its core, equipped with a dual-core Xtensa LX7 processor running at up to 240MHz, and includes 8MB PSRAM and 16MB FLASH memory, capable of handling high-performance computing and multitasking demands. For storage, it utilizes 128Mbit (16MB) 3.3V NOR flash, ensuring long- term, stable operation for firmware, data, and configuration file storage.
The controller supports multiple communication protocols, including dual CAN bus, RS232, RS485, and TTL interfaces, making it suitable for a wide range of industrial and IoT applications. To enhance user interaction, the Flow Connect integrates Neopixel RGB LED lighting control, enabling dynamic color and lighting effects for intuitive visual feedback.
In addition, the Flow Connect power management system employs multiple DC-DC converters that support various voltage outputs from 12V to 3.3V. It also features built in electronic fuses (eFuse) to protect each voltage channel from overcurrent, ensuring safe operation even in harsh environments.
The Flow Connect is designed with the demanding requirements of industrial control, intelligent transportation, and IoT gateway applications in mind, offering reliable multiprotocol communication, robust data storage, dynamic RGB display, and comprehensive power protection.

Flow Connect

1. Communication Capabilities:
   - Main Controller: ESP32-S3R8
   - Wireless Communication: Wi-Fi (2.4 GHz), Bluetooth Low Energy (BLE) 5.0
   - Dual CAN Bus: Supports dual CAN bus interfaces for reliable data communication in industrial environments.
   - Serial Communication: RS232, RS485, and TTL interfaces for versatile wired communication options.
2. Processor and Performance:
   - Processor Model: Xtensa LX7 Dual-core (ESP32-S3R8)
   - Storage Capacity: 16MB Flash, 8MB PSRAM
   - Processor Operating Frequency: Xtensa® dual-core 32-bit LX7 microprocessor, up to 240 MHz
3. Display and Input:
   - RGB LED: Integrated Neopixel RGB LED for dynamic visual feedback.
4. Memory:
   - NOR Flash: 128Mbit (16MB), 3.3V for firmware and data storage.
5. Power Management:
   - Power Supply: DC-DC converters supporting 12V to 3.3V outputs.
   - Protection: Built-in electronic fuses (eFuse) for overcurrent protection across all voltage channels.
6. GPIO Pins and Programmable Interfaces:

1. Grove Interface: Supports connection and expansion of I2C sensors and other modules.

7. Others:
   - Onboard Interface: Type-C interface for programming, power supply, and serial communication.
   - Physical Dimensions: 60*60*15 mm

SPECIFICATIONS

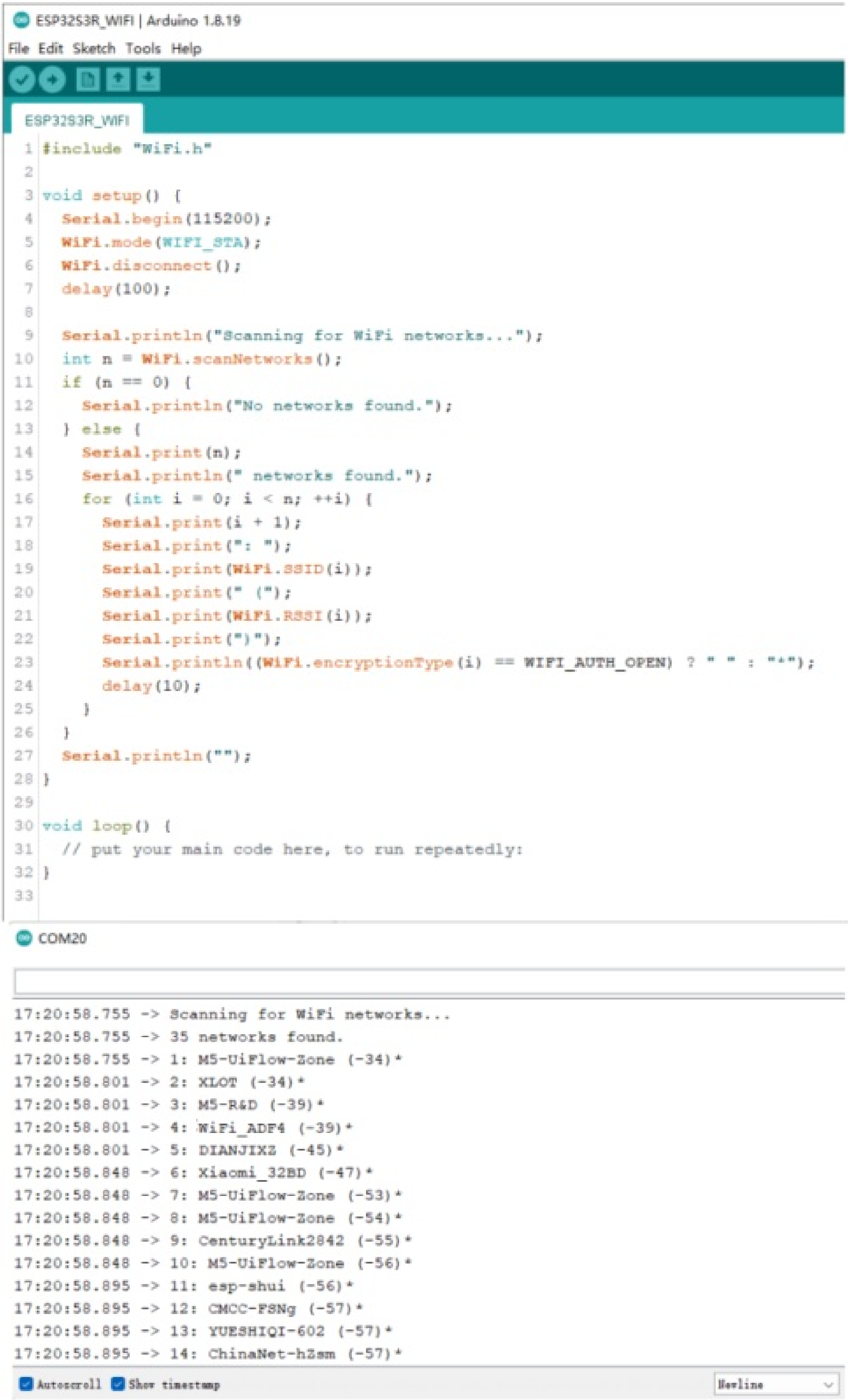
Parameter and Specification	**Value**
MCU	ESP32-S3R8@ Xtensa dual – core 32- bit LX7, 240MHz
Communication Capability	Wi-Fi, BLE, Dual CAN Bus, RS232, RS485, TTL
Supply Voltage	12V to 3. 3V DC (via DC-DC converters)
Flash Storage Capacity	16MB Flash
PSRAM Storage Capacity	8MB PSRAM
NOR Flash	GD25Q128/ W25Q128, 128 Mbi t (16MB), 3. 3V
RGB LED	6 x Neopixel RGB LEDs for dynamic lighting
Expansion Interface	Grove interface for connecting and expanding I2C sensors
Operating Temperature	0° C – 40° C
Wi- Fi Working Frequency	802. llb/ g/ n: 2412 MHz – 2482 MHz
BLE Working Frequency	2402 MHz – 2480 MHz
Manufacturer	M5Stack Technology Co. , Ltd

QUICK START

Before you do this step, look at the text in the final appendix: Installing Arduino

Print WiFi information

1. Open Arduino IDE (Refer to https://docs.m5stack.com/en/arduino/arduino_ide for the installation guide for the development board and software)
2. Select the ESP32S3 DEV Module board and the corresponding port, then upload the code
3. Open the serial monitor to display the scanned WiFi and signal strength information



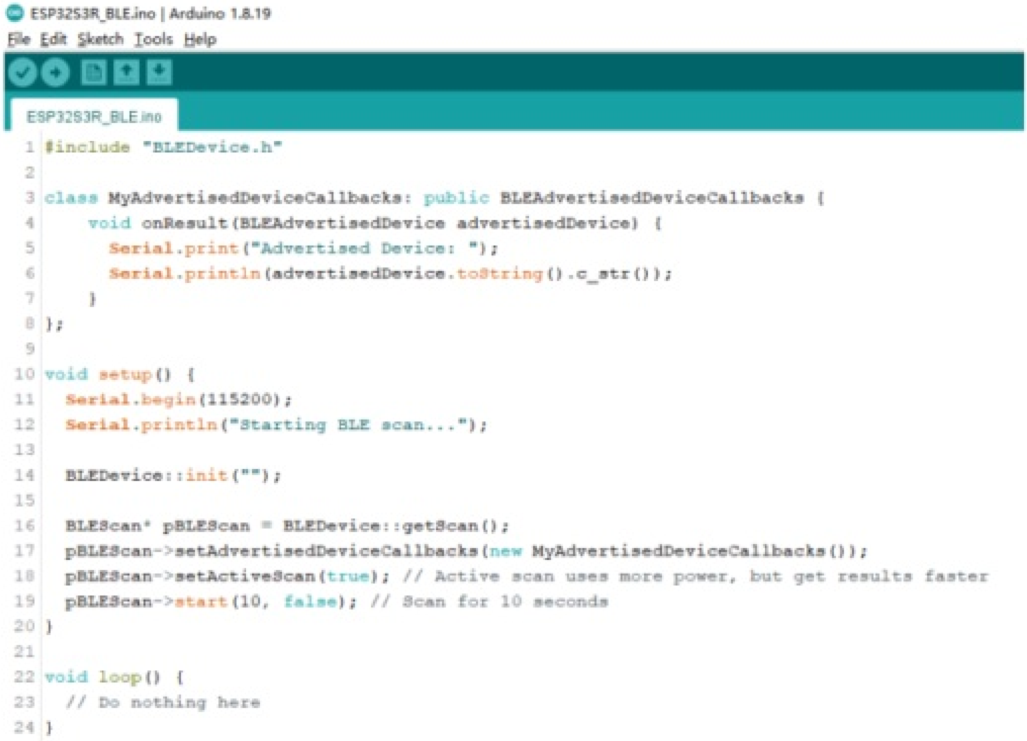
```
#include "WiFi.h"

void setup() {
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);

  Serial.println("Scanning for WiFi networks...");
  int n = WiFi.scanNetworks();
  if (n == 0) {
    Serial.println("No networks found.");
  } else {
    Serial.print(n);
    Serial.println(" networks found.");
    for (int i = 0; i < n; ++i) {
      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("");
}

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

```
17:20:58.755 -> Scanning for WiFi networks...
17:20:58.755 -> 35 networks found.
17:20:58.755 -> 1: M5-UiFlow-Zone (-34)*
17:20:58.801 -> 2: XLOT (-34)*
17:20:58.801 -> 3: M5-R&D (-39)*
17:20:58.801 -> 4: WiFi_ADF4 (-39)*
17:20:58.801 -> 5: DIANJIXZ (-45)*
17:20:58.848 -> 6: Xiaomi_32BD (-47)*
17:20:58.848 -> 7: M5-UiFlow-Zone (-53)*
17:20:58.848 -> 8: M5-UiFlow-Zone (-54)*
17:20:58.848 -> 9: CenturyLink2842 (-55)*
17:20:58.848 -> 10: M5-UiFlow-Zone (-56)*
17:20:58.895 -> 11: esp-shui (-56)*
17:20:58.895 -> 12: CMCC-FSNg (-57)*
17:20:58.895 -> 13: YUESHIQI-602 (-57)*
17:20:58.895 -> 14: ChinaNet-hZsm (-57)*
```

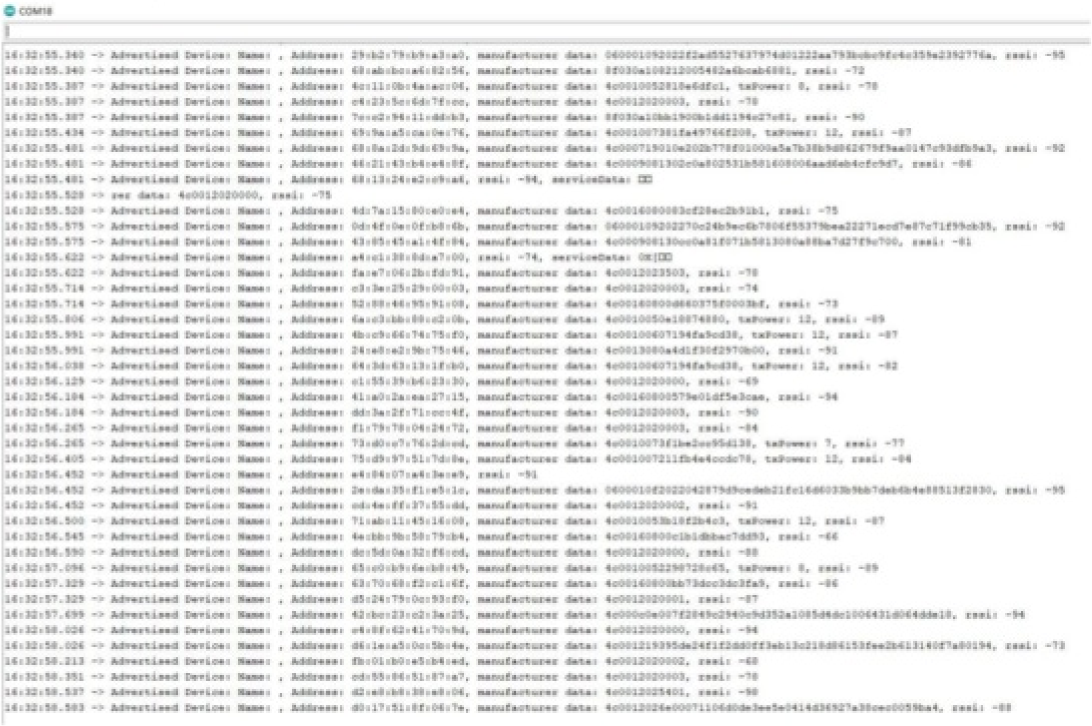
☑ Autoscroll ☑ Show timestamp Newline

Print BLE information

1. Open Arduino IDE (Refer to https://docs.m5stack.com/en/arduino/arduino_ide for the installation guide for the development board and software)
2. Select the ESP32S3 DEV Module board and the corresponding port, then upload the code
3. Open the serial monitor to display the scanned BLE and signal strength information


ESP32S3R_BLE.ino | Arduino 1.8.19
File Edit Sketch Tools Help
ESP32S3R_BLE.ino

```
#include "BLEDevice.h"

class MyAdvertisedDeviceCallbacks: public BLEAdvertisedDeviceCallbacks {
    void onResult(BLEAdvertisedDevice advertisedDevice) {
      Serial.print("Advertised Device: ");
      Serial.println(advertisedDevice.toString().c_str());
    }
};

void setup() {
  Serial.begin(115200);
  Serial.println("Starting BLE scan...");

  BLEDevice::init("");

  BLEScan* pBLEScan = BLEDevice::getScan();
  pBLEScan->setAdvertisedDeviceCallbacks(new MyAdvertisedDeviceCallbacks());
  pBLEScan->setActiveScan(true); // Active scan uses more power, but get results faster
  pBLEScan->start(10, false); // Scan for 10 seconds
}

void loop() {
  // Do nothing here
}
```



```
14:32:55.340 -> Advertised Device: Name: , Address: 29:b2:79:b9:a3:a0, manufacturer data: 0600010920222f2ad5527637974d01222aa793bcbc9fc4c359e2392776a, rssi: -95
14:32:55.340 -> Advertised Device: Name: , Address: 68:ab:bc:a6:82:56, manufacturer data: 8f030a1082120005482a6bcab6881, rssi: -72
14:32:55.387 -> Advertised Device: Name: , Address: 4c:11:0b:4a:ac:06, manufacturer data: 4c0010052810e6dfc1, txPower: 8, rssi: -78
14:32:55.387 -> Advertised Device: Name: , Address: c4:23:5c:6d:7f:cc, manufacturer data: 4c0012020003, rssi: -78
14:32:55.387 -> Advertised Device: Name: , Address: 7e:c2:94:11:dd:b3, manufacturer data: 8f030a10bb1900b1dd1194e27e81, rssi: -90
14:32:55.434 -> Advertised Device: Name: , Address: 69:9a:a5:ca:0e:76, manufacturer data: 4c00100738 1fa49766f208, txPower: 12, rssi: -87
14:32:55.481 -> Advertised Device: Name: , Address: 68:0a:2d:9d:69:9a, manufacturer data: 4c000719010e202b778f01000a5a7b38b9d062679f9aa0147c93dfb9a3, rssi: -92
14:32:55.481 -> Advertised Device: Name: , Address: 46:21:43:b4:e4:8f, manufacturer data: 4c0009081302c0a802531b501608006aad6eb4cfc9d7, rssi: -86
14:32:55.481 -> Advertised Device: Name: , Address: 68:13:24:e2:c9:a4, rssi: -94, serviceData: □□
14:32:55.528 -> ser data: 4c0012020000, rssi: -75
14:32:55.528 -> Advertised Device: Name: , Address: 4d:7a:15:80:e0:e4, manufacturer data: 4c0016080083cf28ec2b91b1, rssi: -75
14:32:55.575 -> Advertised Device: Name: , Address: 0d:4f:0e:0f:b8:6b, manufacturer data: 06000109202270c24b5ec6b7806f55379bea22271ecd7e87c71f99cb35, rssi: -92
14:32:55.575 -> Advertised Device: Name: , Address: 43:85:45:a1:4f:84, manufacturer data: 4c000908130cc0a81f071b5013080a88ba7d27f9c700, rssi: -81
14:32:55.622 -> Advertised Device: Name: , Address: a4:c1:38:0d:a7:00, rssi: -74, serviceData: 0X|□□
14:32:55.622 -> Advertised Device: Name: , Address: fa:e7:06:2b:fd:91, manufacturer data: 4c0012023503, rssi: -78
14:32:55.714 -> Advertised Device: Name: , Address: c3:3e:25:29:00:03, manufacturer data: 4c0012020003, rssi: -74
14:32:55.714 -> Advertised Device: Name: , Address: 52:88:46:95:91:08, manufacturer data: 4c001608000d660375f0003bf, rssi: -73
14:32:55.806 -> Advertised Device: Name: , Address: 6a:c3:bb:80:c2:0b, manufacturer data: 4c0010050e10874880, txPower: 12, rssi: -89
14:32:55.991 -> Advertised Device: Name: , Address: 4b:c9:66:74:75:f0, manufacturer data: 4c00100607194fa9cd38, txPower: 12, rssi: -87
14:32:55.991 -> Advertised Device: Name: , Address: 24:e8:e2:9b:75:46, manufacturer data: 4c0013080a4d1f30f2970b00, rssi: -91
14:32:56.038 -> Advertised Device: Name: , Address: 64:3d:63:13:1f:b0, manufacturer data: 4c00100607194fa9cd38, txPower: 12, rssi: -82
14:32:56.129 -> Advertised Device: Name: , Address: c1:55:39:b6:23:30, manufacturer data: 4c0012020000, rssi: -69
14:32:56.184 -> Advertised Device: Name: , Address: 41:a0:2a:ea:27:15, manufacturer data: 4c00160800579e01df5e3cae, rssi: -94
14:32:56.184 -> Advertised Device: Name: , Address: dd:3e:2f:71:cc:4f, manufacturer data: 4c0012020003, rssi: -90
14:32:56.265 -> Advertised Device: Name: , Address: f1:79:78:06:24:72, manufacturer data: 4c0012020003, rssi: -84
14:32:56.265 -> Advertised Device: Name: , Address: 73:d0:e7:76:2d:ed, manufacturer data: 4c0010073f1be2ce95d138, txPower: 7, rssi: -77
14:32:56.405 -> Advertised Device: Name: , Address: 75:d9:97:51:7d:8e, manufacturer data: 4c001007211fb4e4ccdc78, txPower: 12, rssi: -84
14:32:56.452 -> Advertised Device: Name: , Address: e4:84:07:a4:3e:e9, rssi: -91
14:32:56.452 -> Advertised Device: Name: , Address: 2e:da:35:f1:e5:1c, manufacturer data: 0600010f2022042879d9cedeb21fc16d6033b9bb7deb4b4e88513f2830, rssi: -95
14:32:56.452 -> Advertised Device: Name: , Address: cd:4e:ff:37:55:dd, manufacturer data: 4c0012020002, rssi: -91
14:32:56.500 -> Advertised Device: Name: , Address: 71:ab:11:45:16:08, manufacturer data: 4c0010053b18f2b4c3, txPower: 12, rssi: -87
14:32:56.545 -> Advertised Device: Name: , Address: 4e:bb:9b:58:79:b4, manufacturer data: 4c00160800c1b1dbbac7dd93, rssi: -66
14:32:56.590 -> Advertised Device: Name: , Address: dc:5d:0a:32:f6:cd, manufacturer data: 4c0012020000, rssi: -88
14:32:57.096 -> Advertised Device: Name: , Address: 65:c0:b9:6e:b8:49, manufacturer data: 4c0010052290728c65, txPower: 8, rssi: -89
14:32:57.329 -> Advertised Device: Name: , Address: 63:70:68:f2:c1:6f, manufacturer data: 4c00160800bb73dcc3dc3fa9, rssi: -86
14:32:57.329 -> Advertised Device: Name: , Address: d5:24:79:0c:93:f0, manufacturer data: 4c0012020001, rssi: -87
14:32:57.699 -> Advertised Device: Name: , Address: 42:bc:23:c2:3a:25, manufacturer data: 4c000c0e007f2049c2940c9d352a1085d6dc1006431d064dde18, rssi: -94
14:32:58.026 -> Advertised Device: Name: , Address: e4:8f:62:41:70:9d, manufacturer data: 4c0012020000, rssi: -94
14:32:58.026 -> Advertised Device: Name: , Address: d6:1e:a5:0c:5b:4e, manufacturer data: 4c001219395de24f1f2dd0ff3eb13c218d86153fee2b613140f7a80194, rssi: -73
14:32:58.213 -> Advertised Device: Name: , Address: fb:01:b0:e5:b4:ed, manufacturer data: 4c0012020002, rssi: -68
14:32:58.351 -> Advertised Device: Name: , Address: cd:55:86:51:87:a7, manufacturer data: 4c0012020003, rssi: -78
14:32:58.537 -> Advertised Device: Name: , Address: d2:e8:b8:38:e8:06, manufacturer data: 4c0012025401, rssi: -98
14:32:58.583 -> Advertised Device: Name: , Address: d0:17:51:8f:06:7e, manufacturer data: 4c0012026e00071106d0de3ee5e0414d36927a38cec0059ba4, rssi: -88
```

FCC Warning

FCC Caution:

Any Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.
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.

IMPORTANT NOTE:

Note: 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:

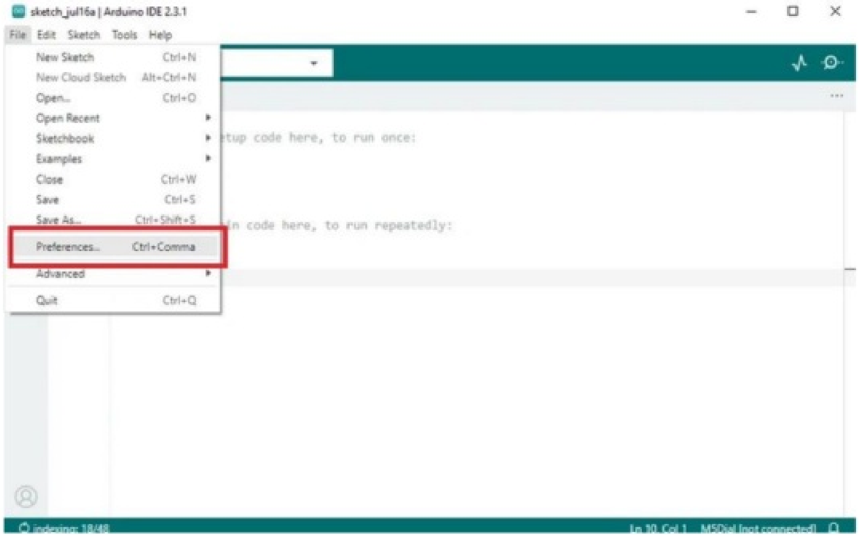
- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 20cm between the radiator & your body.

Arduino Install

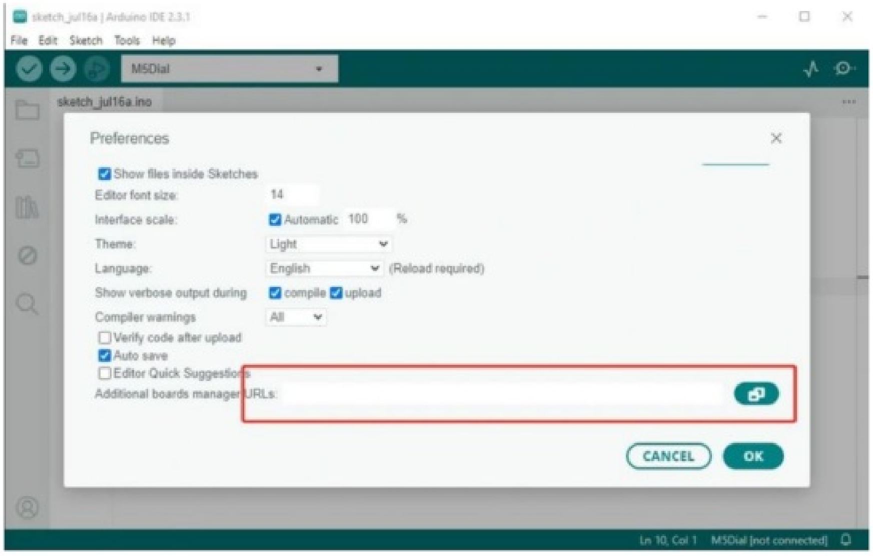
- Installing Arduino IDE(**https://www.arduino.cc/en/Main/Software**) Click to visit the Arduino official website , and select the installation package for your operating system to download.
- Installing Arduino Board Management

1. The Board Manager URL is used to index the development board information for a specific platform. In the Arduino IDE menu, select File -> Preferences



```
tup code here, to run once:

in code here, to run repeatedly:
```

2. Copy the ESP board management URL below into the Additional Board Manager URLs: field, and save.
https://espressif.github.io/arduino-esp32/package_esp32_dev_index.json



sketch_jul16a | Arduino IDE 2.3.1

File Edit Sketch Tools Help

M5Dial

sketch_jul16a.ino

Preferences

☑ Show files inside Sketches

Editor font size: 14

Interface scale: ☑ Automatic 100 %

Theme: Light

Language: English (Reload required)

Show verbose output during ☑ compile ☑ upload

Compiler warnings All

☐ Verify code after upload

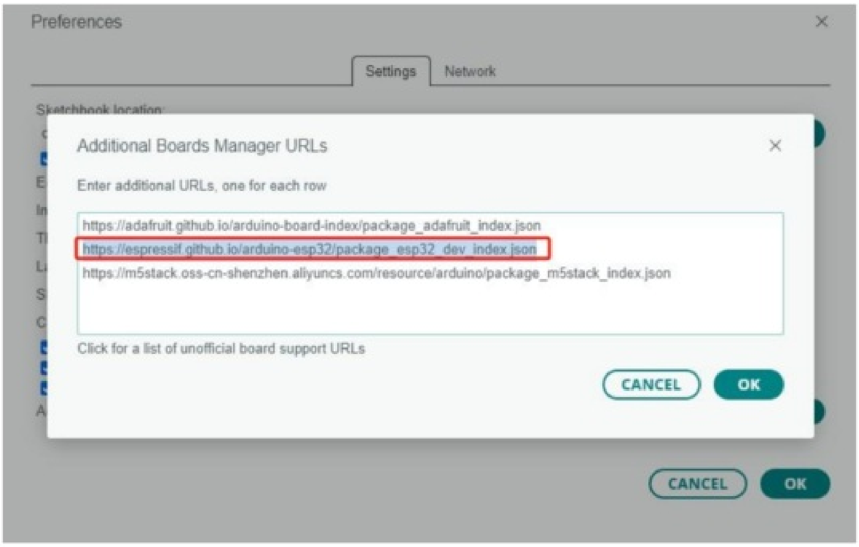
☑ Auto save

☐ Editor Quick Suggestions

Additional boards manager URLs:

CANCEL OK

Ln 10, Col 1 M5Dial [not connected]



Preferences

Settings Network

Sketchbook location:

Additional Boards Manager URLs

Enter additional URLs, one for each row

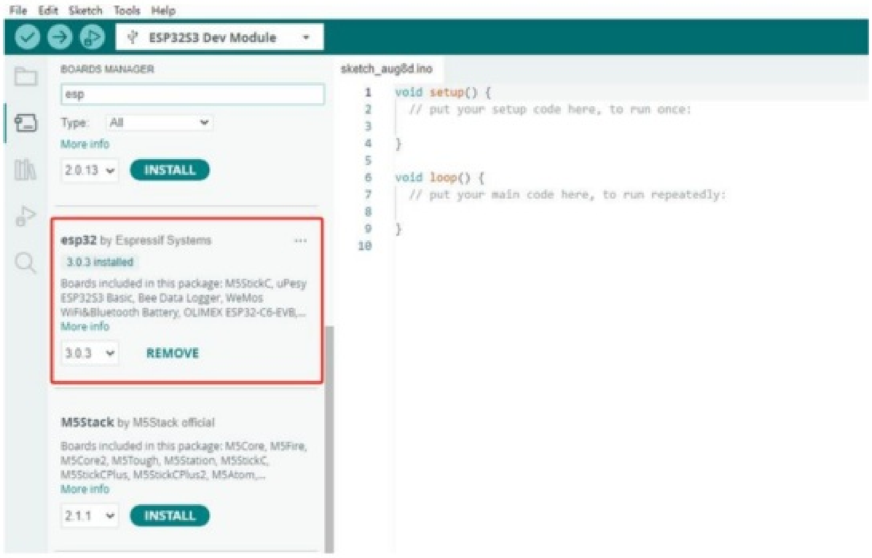
```
https://adafruit.github.io/arduino-board-index/package_adafruit_index.json
https://espressif.github.io/arduino-esp32/package_esp32_dev_index.json
https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/arduino/package_m5stack_index.json
```

Click for a list of unofficial board support URLs

CANCEL OK

CANCEL OK

3. In the sidebar, select Board Manager, search for M5Stack, and click Install.



File Edit Sketch Tools Help

ESP32S3 Dev Module

BOARDS MANAGER

esp

Type: All

More info

2.0.13 INSTALL

esp32 by Espressif Systems

3.0.3 installed

Boards included in this package: M5StickC, uPesy ESP32S3 Basic, Bee Data Logger, WeMos WiFi&Bluetooth Battery, OLIMEX ESP32-C6-EVB,...

More info

3.0.3 REMOVE

M5Stack by M5Stack official

Boards included in this package: M5Core, M5Fire, M5Core2, M5Tough, M5Station, M5StickC, M5StickCPlus, M5StickCPlus2, M5Atom,...

More info

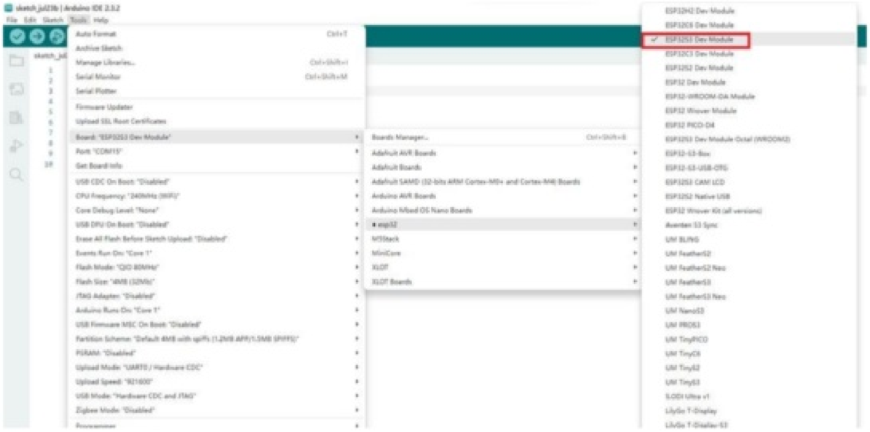
2.1.1 INSTALL

sketch_aug8d.ino

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

}

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

}
```

4. In the sidebar, select Board Manager, search for M5Stack, and click Install.

   Depending on the product used, select the corresponding development board under Tools -> Board -> M5Stack -> {ESP32S3 DEV Module board}.



5. Connect the device to your computer with a data cable to upload the program

Documents / Resources

	[M5 STACK Flow Connect Software](#) [pdf] User Guide M5FCV1, 2AN3WM5FCV1, Flow Connect Software, Connect Software, Software

References

- espressif.github.io/arduino-esp32/package_esp32_dev_index.json
- Software | Arduino
- User Manual

Manuals+, Privacy Policy

This website is an independent publication and is neither affiliated with nor endorsed by any of the trademark owners. The "Bluetooth®" word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. The "Wi-Fi®" word mark and logos are registered trademarks owned by the Wi-Fi Alliance. Any use of these marks on this website does not imply any affiliation with or endorsement.