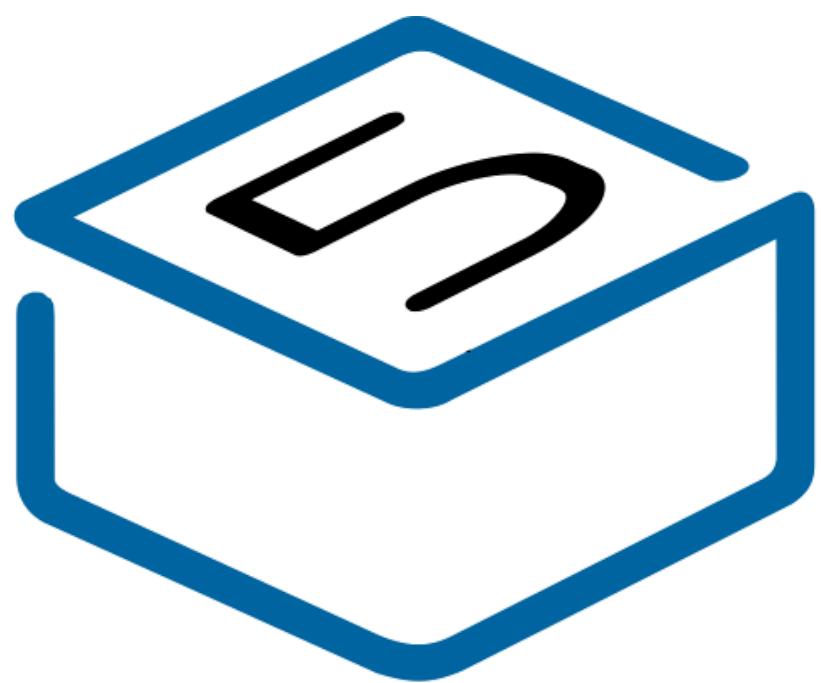


M5Stamp Fly



M5STACK

2024

CONTENTS

1. OUTLINE	3
1.1. M5Stamp Fly.....	3
2. SPECIFICATIONS.....	4
2.1. Module Size.....	7
3. QUICK START.....	6
3.1. SCAN Wi-Fi.....	7
3.2 SCAN BLE Device.....	8
4. FCC Warning.....	9
Arduino Install.....	10

1. OUTLINE

M5Stamp Fly is a programmable open-source quadcopter kit, featuring the StampS3 as the main controller. It integrates a BMI270 6-axis gyroscope and a BMM150 3-axis magnetometer for attitude and direction detection. The BMP280 barometric pressure sensor and two VL53L3 distance sensors enable precise altitude hold and obstacle avoidance. The PMW3901MB-TXQT optical flow sensor provides displacement detection. The kit includes a buzzer, a reset button, and WS2812 RGB LEDs for interaction and status indication. It is equipped with a 300mAh high-voltage battery and four high-speed coreless motors. The PCB features an INA3221AIRGVR for real-time current/voltage monitoring and has two Grove connectors for additional sensors and peripherals. Preloaded with debugging firmware, the Stamp Fly can be controlled using an Atom Joystick via the ESP-NOW protocol. Users can choose between automatic and manual modes, allowing for easy implementation of functions like precise hovering and flips. The firmware source code is open-source, making the product suitable for education, research, and various drone development projects.

1.1. STAMPS3A

1. Communication Capabilities:

- Main Controller: M5StampS3, based on the ESP32-S3 chip, supports Wi-Fi and Bluetooth communication.
- Wireless Communication: Supports ESP-NOW protocol and can be controlled via the Atom Joystick.

2. Processor and Performance:

- Processor Model: Xtensa LX7 Dual-core (ESP32-S3).
- Storage Capacity: 8MB Flash.
- Processor Operating Frequency: Up to 240 MHz.

3. Display and Input:

- RGB LED: Built-in WS2812 RGB LED for interaction and status indication.

4. GPIO Pins and Programmable Interfaces:

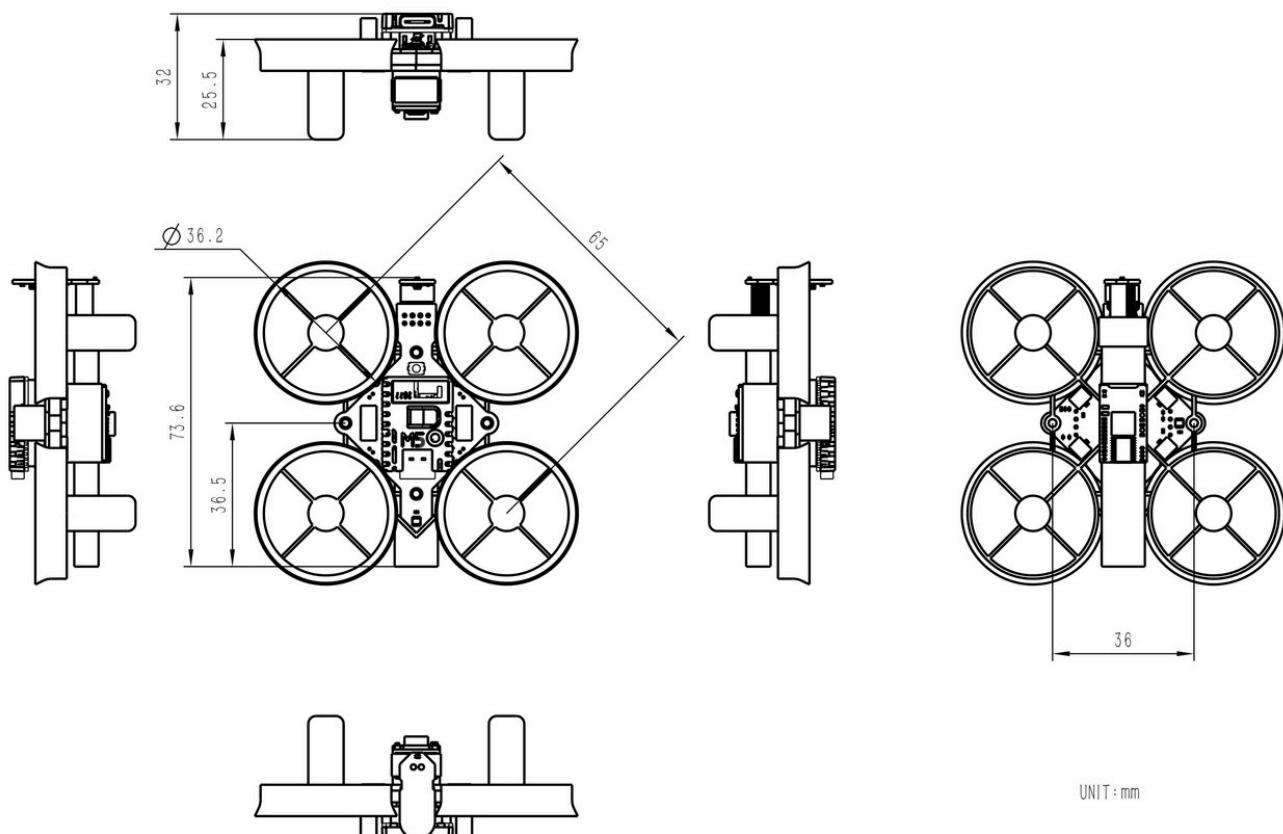
- Grove Interface: Equipped with two Grove connectors for extending sensors and peripherals.

5. Others:

- Sensors: Integrates a BMI270 six-axis gyroscope, BMM150 three-axis magnetometer, BMP280 barometric pressure sensor, two VL53L3 distance sensors, and a PMW3901MB-TXQT optical flow sensor. These enable attitude detection, direction sensing, precise altitude holding, and obstacle avoidance.
- Battery: Equipped with a 300mAh high-voltage lithium battery providing approximately 4 minutes of flight time.
- Power Management: Onboard INA3221AIRGVR chip for real-time current/voltage monitoring.
- Physical Dimensions: 81.5 x 81.5 x 31 mm.
- Weight: 36.8 grams.
- The product comes preloaded with debugging firmware. Users can switch between automatic and manual modes, allowing for precise hovering and maneuvers like flips. The firmware source code is open-source, making it ideal for education, research, and various UAV development projects.

2. SPECIFICATIONS

2.1 Module Size



3. QUICK START

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

3.1. 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



The screenshot shows the Arduino IDE interface. The top bar displays the title 'ESP32S3R_WIFI | Arduino 1.8.19' and the menu options 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. The main code editor window contains the following C++ code:

```
1 #include "WiFi.h"
2
3 void setup() {
4     Serial.begin(115200);
5     WiFi.mode(WIFI_STA);
6     WiFi.disconnect();
7     delay(100);
8
9     Serial.println("Scanning for WiFi networks...");
10    int n = WiFi.scanNetworks();
11    if (n == 0) {
12        Serial.println("No networks found.");
13    } else {
14        Serial.print(n);
15        Serial.println(" networks found.");
16        for (int i = 0; i < n; ++i) {
17            Serial.print(i + 1);
18            Serial.print(": ");
19            Serial.print(WiFi.SSID(i));
20            Serial.print(" (");
21            Serial.print(WiFi.RSSI(i));
22            Serial.print(")");
23            Serial.println((WiFi.encryptionType(i) == WIFI_AUTH_OPEN) ? " " : "*");
24            delay(10);
25        }
26    }
27    Serial.println("");
28 }
29
30 void loop() {
31     // put your main code here, to run repeatedly:
32 }
```

The bottom half of the screenshot shows the 'Serial Monitor' window. It displays the output of the code, which is a list of WiFi networks found by the device. The output text is as follows:

```
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)*
```

At the bottom of the serial monitor window, there are checkboxes for 'Autoscroll' and 'Show timestamp', and a dropdown menu for 'Newline'.

3. QUICK START

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

3.1. Print BLE information

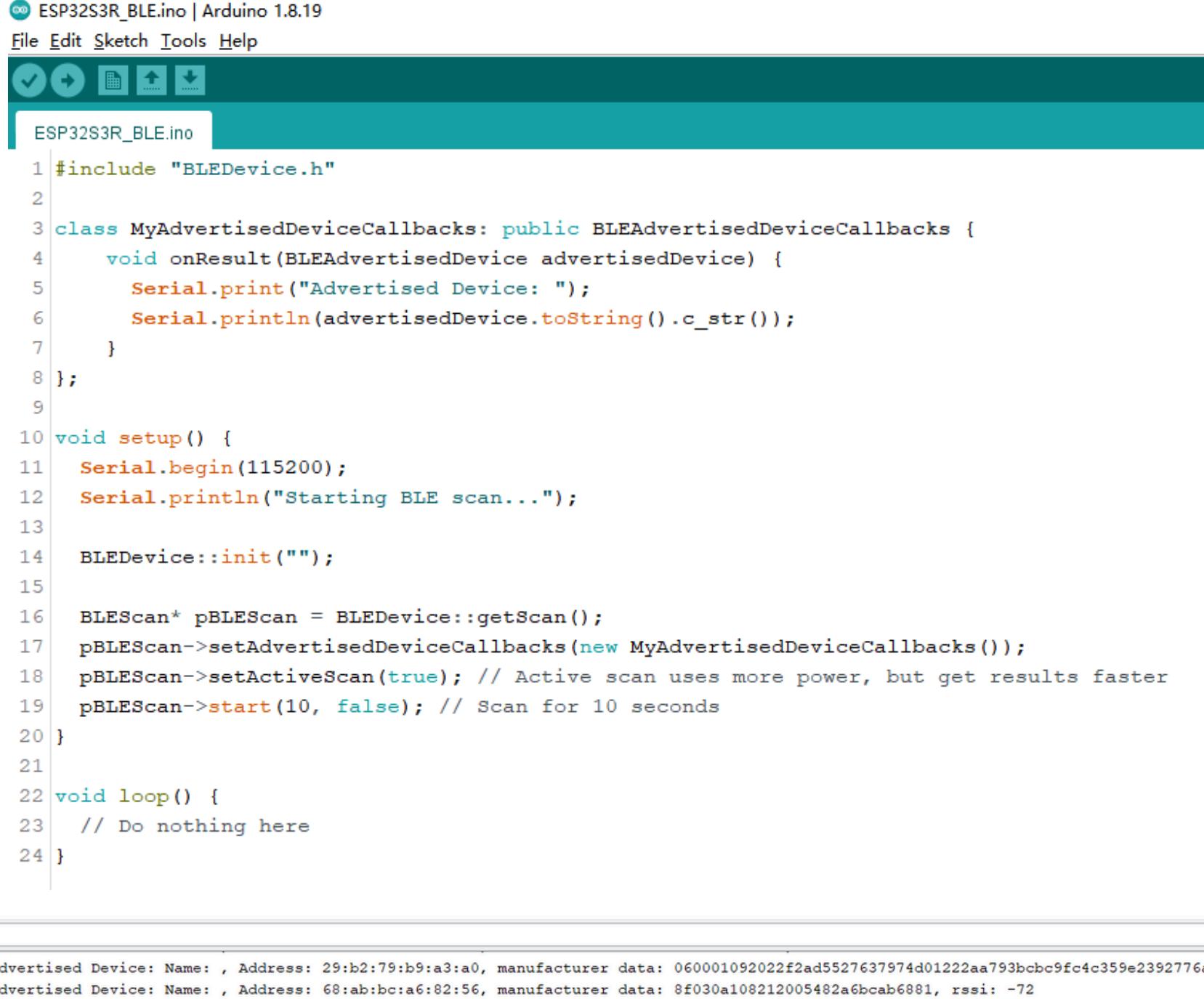
1. Open Arduino IDE (Refer to

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



The screenshot shows the Arduino IDE interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. The main window shows the code for `ESP32S3R_BLE.ino`. The code initializes a BLE device, starts an active scan for 10 seconds, and prints the results to the serial monitor. The serial monitor window below shows a list of nearby BLE devices with their names, addresses, manufacturer data, and signal strengths (RSSI).

```
#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
}
```

COM18

```
16:32:55.340 -> Advertised Device: Name: , Address: 29:b2:79:b9:a3:a0, manufacturer data: 060001092022f2ad5527637974d01222aa793bc9fc4c359e2392776a, rssi: -95
16:32:55.340 -> Advertised Device: Name: , Address: 68:ab:bc:a6:82:56, manufacturer data: 8f030a108212005482a6bcab6881, rssi: -72
16:32:55.387 -> Advertised Device: Name: , Address: 4c:11:0b:4a:ac:06, manufacturer data: 4c0010052818e6dfc1, txPower: 8, rssi: -78
16:32:55.387 -> Advertised Device: Name: , Address: c4:23:5c:6d:7f:cc, manufacturer data: 4c0012020003, rssi: -78
16:32:55.387 -> Advertised Device: Name: , Address: 7c:c2:94:11:dd:b3, manufacturer data: 8f030a10bb1900b1dd1194c27c81, rssi: -90
16:32:55.434 -> Advertised Device: Name: , Address: 69:9a:a5:ca:0e:76, manufacturer data: 4c001007381fa49766f208, txPower: 12, rssi: -87
16:32:55.481 -> Advertised Device: Name: , Address: 68:8a:2d:9d:69:9a, manufacturer data: 4c000719010e202b778f01000a5a7b38b9d862679f9aa0147c93dfb9a3, rssi: -92
16:32:55.481 -> Advertised Device: Name: , Address: 46:21:43:b4:e4:8f, manufacturer data: 4c0009081302c0a802531b581608006aad6eb4cf9d7, rssi: -86
16:32:55.481 -> Advertised Device: Name: , Address: 68:13:24:e2:c9:a6, rssi: -94, serviceData: 00
16:32:55.528 -> Advertised Device: Name: , Address: 4c0012020000, rssi: -75
16:32:55.528 -> Advertised Device: Name: , Address: 4d:7a:15:80:e0:e4, manufacturer data: 4c0016080083cf28ec2b91b1, rssi: -75
16:32:55.575 -> Advertised Device: Name: , Address: 0d:4f:0e:0f:b8:6b, manufacturer data: 06000109202270c24b9ec6b7806f55379bea22271ecd7e87c71f99cb35, rssi: -92
16:32:55.575 -> Advertised Device: Name: , Address: 43:85:45:a1:4f:84, manufacturer data: 4c000908130cc0a81f071b5813080a88ba7d27f9c700, rssi: -81
16:32:55.622 -> Advertised Device: Name: , Address: a4:c1:38:8d:a7:00, rssi: -74, serviceData: 0X[00]
16:32:55.622 -> Advertised Device: Name: , Address: fa:e7:06:2b:fd:91, manufacturer data: 4c0012023503, rssi: -78
16:32:55.714 -> Advertised Device: Name: , Address: c3:3e:25:29:00:03, manufacturer data: 4c0012020003, rssi: -74
16:32:55.714 -> Advertised Device: Name: , Address: 52:88:46:95:91:08, manufacturer data: 4c00160800d660375f0003bf, rssi: -73
16:32:55.806 -> Advertised Device: Name: , Address: 6a:c3:bb:88:c2:0b, manufacturer data: 4c0010050e18874880, txPower: 12, rssi: -89
16:32:55.991 -> Advertised Device: Name: , Address: 4b:c9:66:74:75:f0, manufacturer data: 4c00100607194fa9cd38, txPower: 12, rssi: -87
16:32:55.991 -> Advertised Device: Name: , Address: 24:e8:e2:9b:75:46, manufacturer data: 4c0013080a4d1f30f2970b00, rssi: -91
16:32:56.038 -> Advertised Device: Name: , Address: 64:3d:63:13:1f:b0, manufacturer data: 4c00100607194fa9cd38, txPower: 12, rssi: -82
16:32:56.129 -> Advertised Device: Name: , Address: c1:55:39:b6:23:30, manufacturer data: 4c0012020000, rssi: -69
16:32:56.184 -> Advertised Device: Name: , Address: 41:a0:2a:ea:27:15, manufacturer data: 4c00160800579e01df5e3cae, rssi: -94
16:32:56.184 -> Advertised Device: Name: , Address: dd:3a:2f:71:cc:4f, manufacturer data: 4c0012020003, rssi: -90
16:32:56.265 -> Advertised Device: Name: , Address: f1:79:78:04:24:72, manufacturer data: 4c0012020003, rssi: -84
16:32:56.265 -> Advertised Device: Name: , Address: 73:d0:c7:76:2d:cd, manufacturer data: 4c0010073f1be2cc95d138, txPower: 7, rssi: -77
16:32:56.405 -> Advertised Device: Name: , Address: 75:d9:97:51:7d:8e, manufacturer data: 4c001007211fb4e4ccdc78, txPower: 12, rssi: -84
16:32:56.452 -> Advertised Device: Name: , Address: e4:84:07:a4:3e:e9, rssi: -91
16:32:56.452 -> Advertised Device: Name: , Address: 2e:da:35:f1:e5:1c, manufacturer data: 0600010f2022042879d9cedeb21fc16d6033b9bb7deb6b4e88513f2830, rssi: -95
16:32:56.452 -> Advertised Device: Name: , Address: cd:4e:ff:37:55:dd, manufacturer data: 4c0012020002, rssi: -91
16:32:56.500 -> Advertised Device: Name: , Address: 71:ab:11:45:16:08, manufacturer data: 4c0010053b18f2b4c3, txPower: 12, rssi: -87
16:32:56.545 -> Advertised Device: Name: , Address: 4e:bb:9b:58:79:b4, manufacturer data: 4c00160800c1b1dbbac7dd93, rssi: -66
16:32:56.590 -> Advertised Device: Name: , Address: dc:5d:0a:32:f6:cd, manufacturer data: 4c0012020000, rssi: -88
16:32:57.096 -> Advertised Device: Name: , Address: 65:c0:b9:6e:b8:49, manufacturer data: 4c0010052298728c65, txPower: 8, rssi: -89
16:32:57.329 -> Advertised Device: Name: , Address: 63:70:68:f2:c1:6f, manufacturer data: 4c00160800bb73dcc3dc3fa9, rssi: -86
16:32:57.329 -> Advertised Device: Name: , Address: d5:24:79:0c:93:f0, manufacturer data: 4c0012020001, rssi: -87
16:32:57.699 -> Advertised Device: Name: , Address: 42:bc:23:c2:3a:25, manufacturer data: 4c000c0e007f2849c2940c9d352a1085d4dc1006431d064dde18, rssi: -94
16:32:58.026 -> Advertised Device: Name: , Address: c4:8f:62:41:70:9d, manufacturer data: 4c0012020000, rssi: -94
16:32:58.026 -> Advertised Device: Name: , Address: d6:1e:a5:0c:5b:4e, manufacturer data: 4c001219395de24f1f2dd0ff3eb13c218d86153fee2b613140f7a80194, rssi: -73
16:32:58.213 -> Advertised Device: Name: , Address: fb:01:b0:e5:b4:ed, manufacturer data: 4c0012020002, rssi: -68
16:32:58.351 -> Advertised Device: Name: , Address: cd:55:86:51:87:a7, manufacturer data: 4c0012020003, rssi: -78
16:32:58.537 -> Advertised Device: Name: , Address: d2:e8:b8:38:e8:06, manufacturer data: 4c0012025401, rssi: -98
16:32:58.583 -> Advertised Device: Name: , Address: d0:17:51:8f:06:7e, manufacturer data: 4c0012026e00071106d0de3ee5e0414d36927a38cec0059ba4, rssi: -88
```

4. 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.

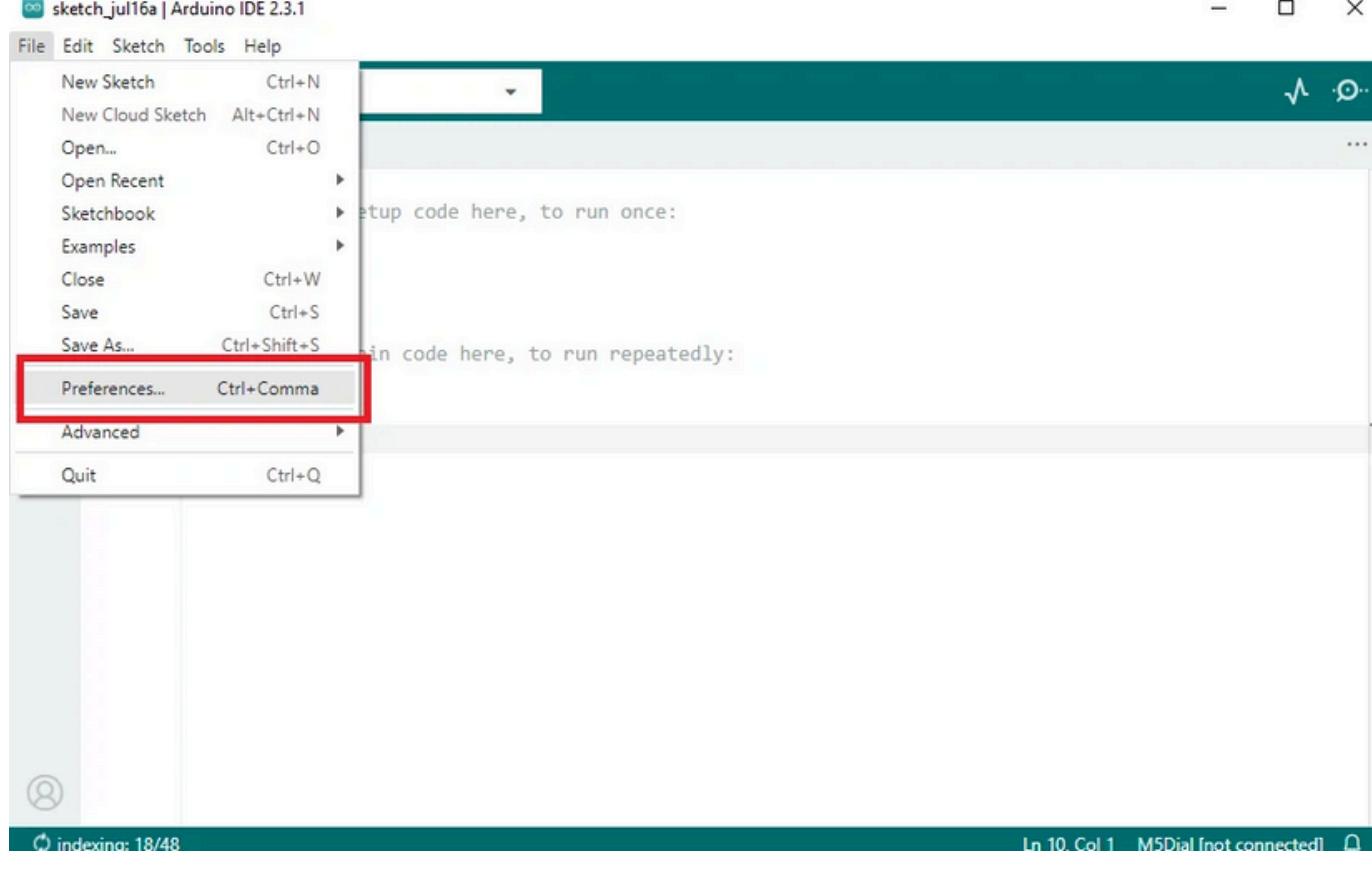
Arduino Install

—. 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.

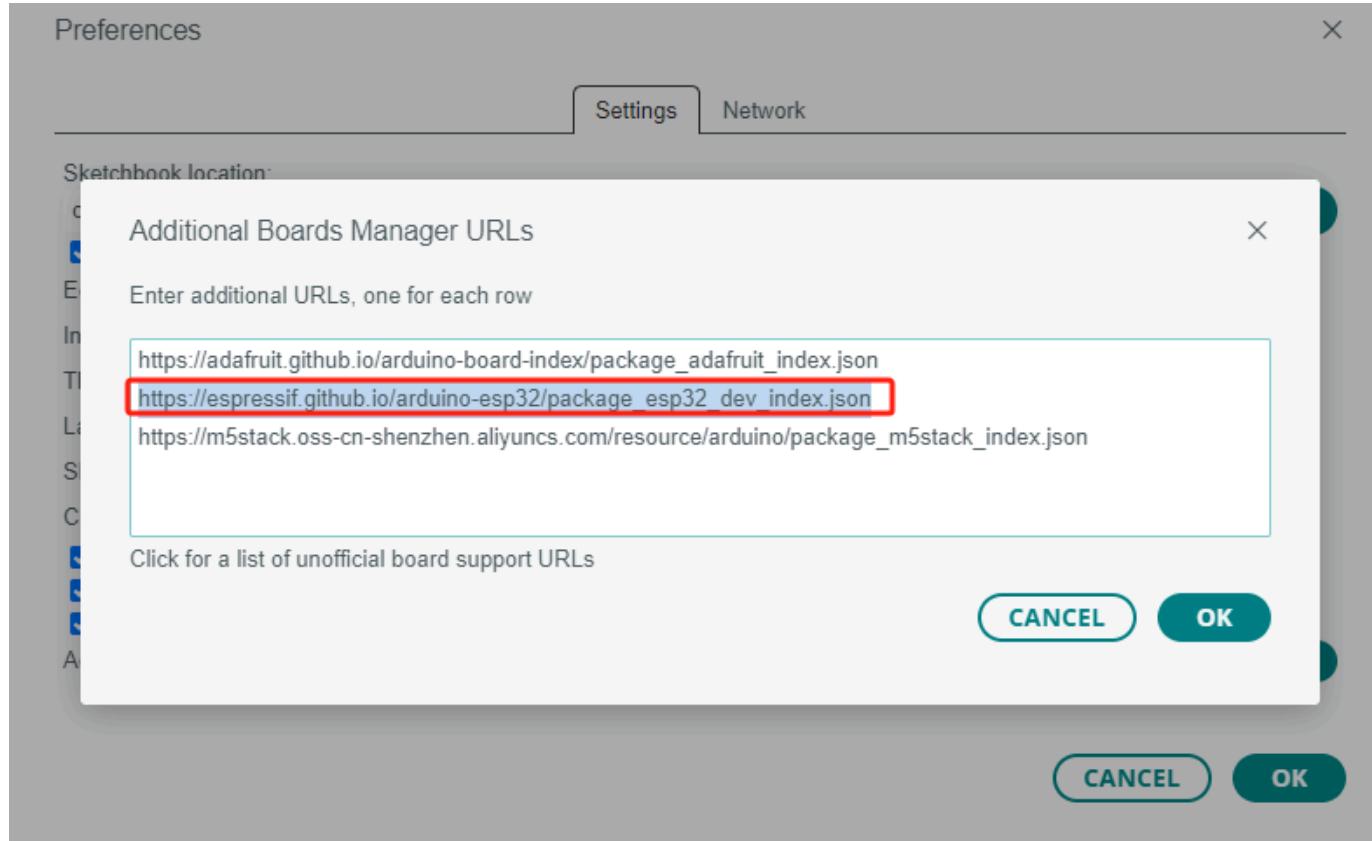
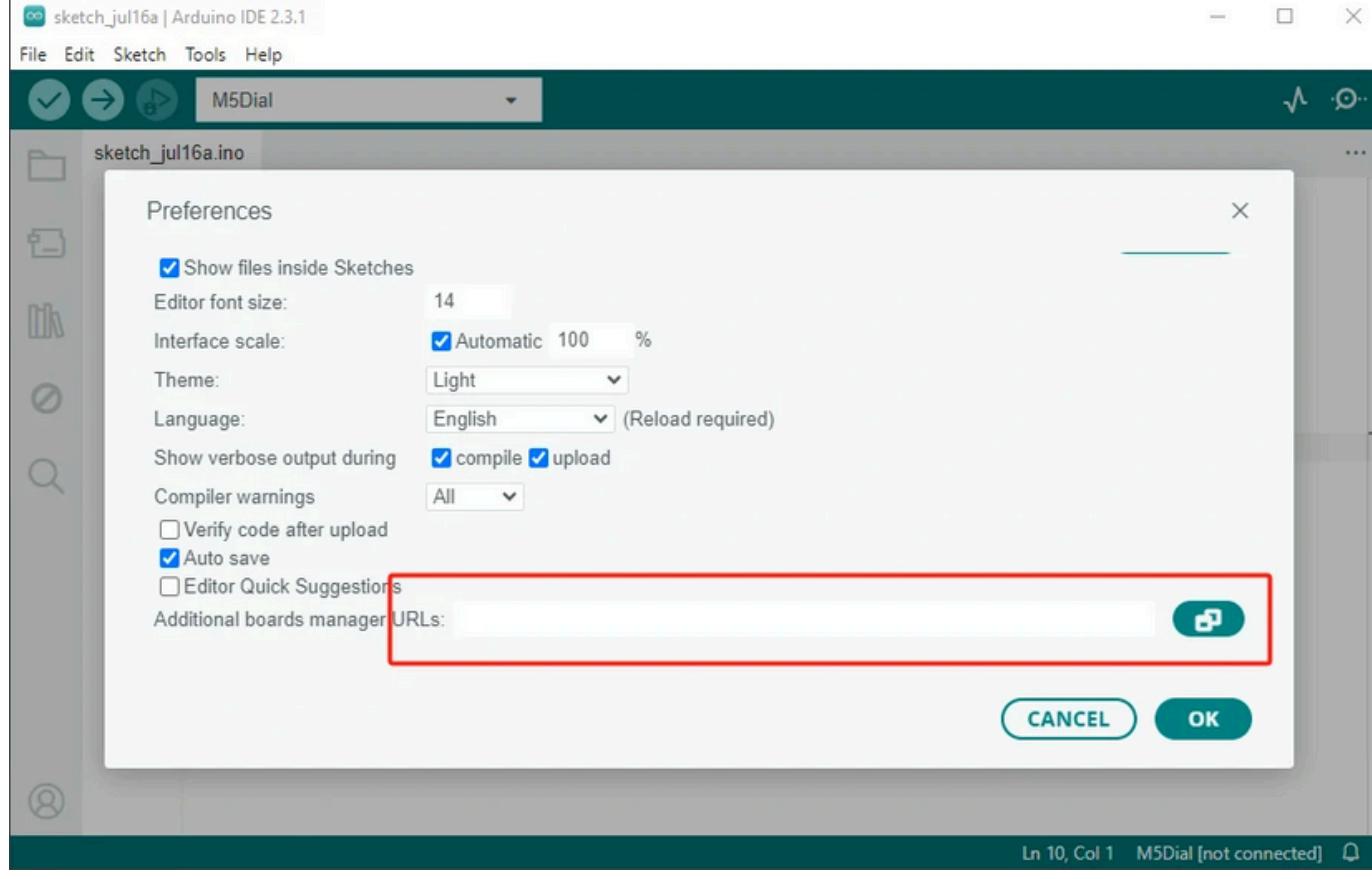
二. Installing Arduino Board Management

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

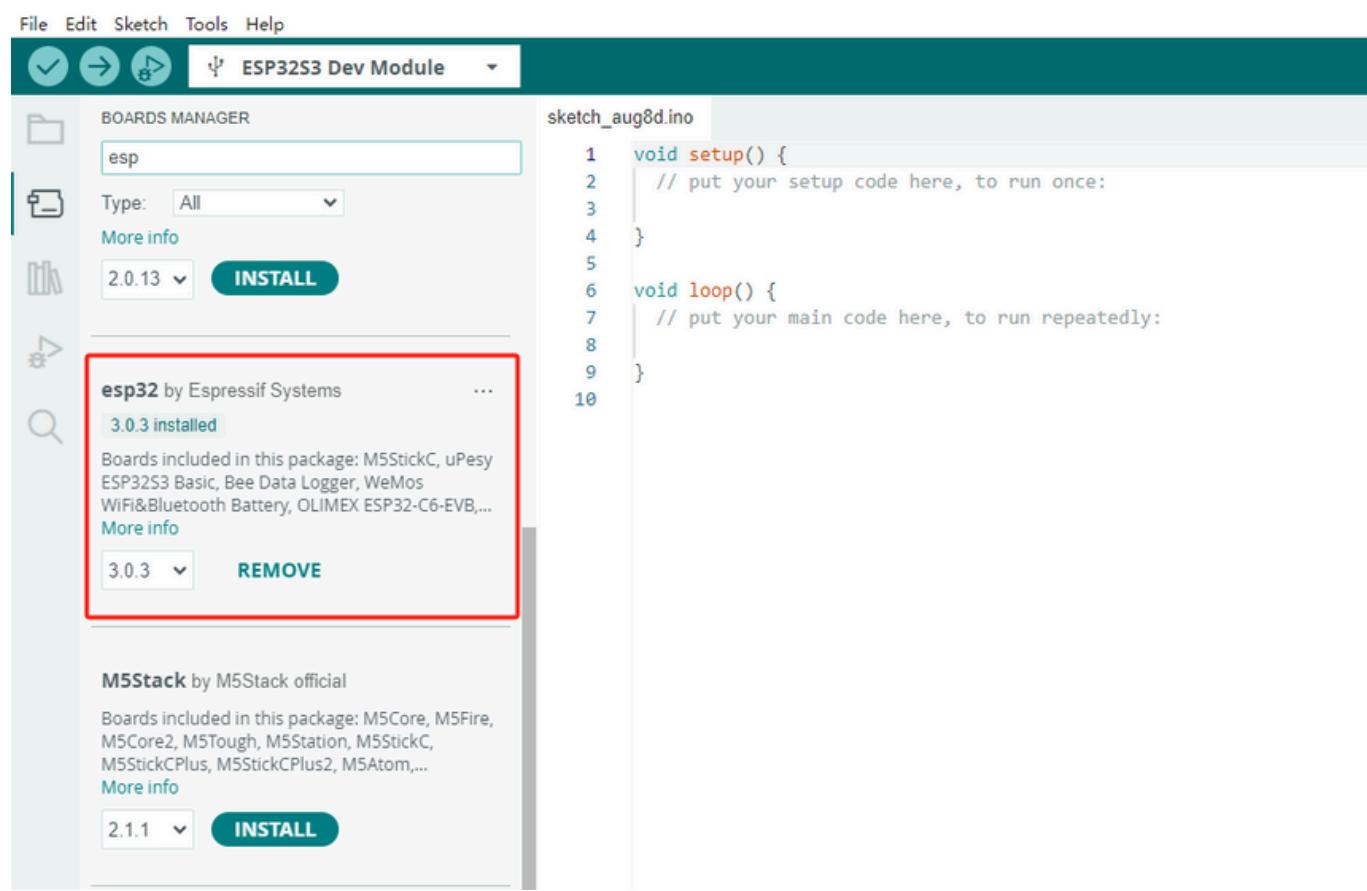


2. Copy the ESP board management URL below into the Additional Board Manager URLs: field, and save.

https://espressif.github.io/arduino-esp32/package_esp32_dev_index.json

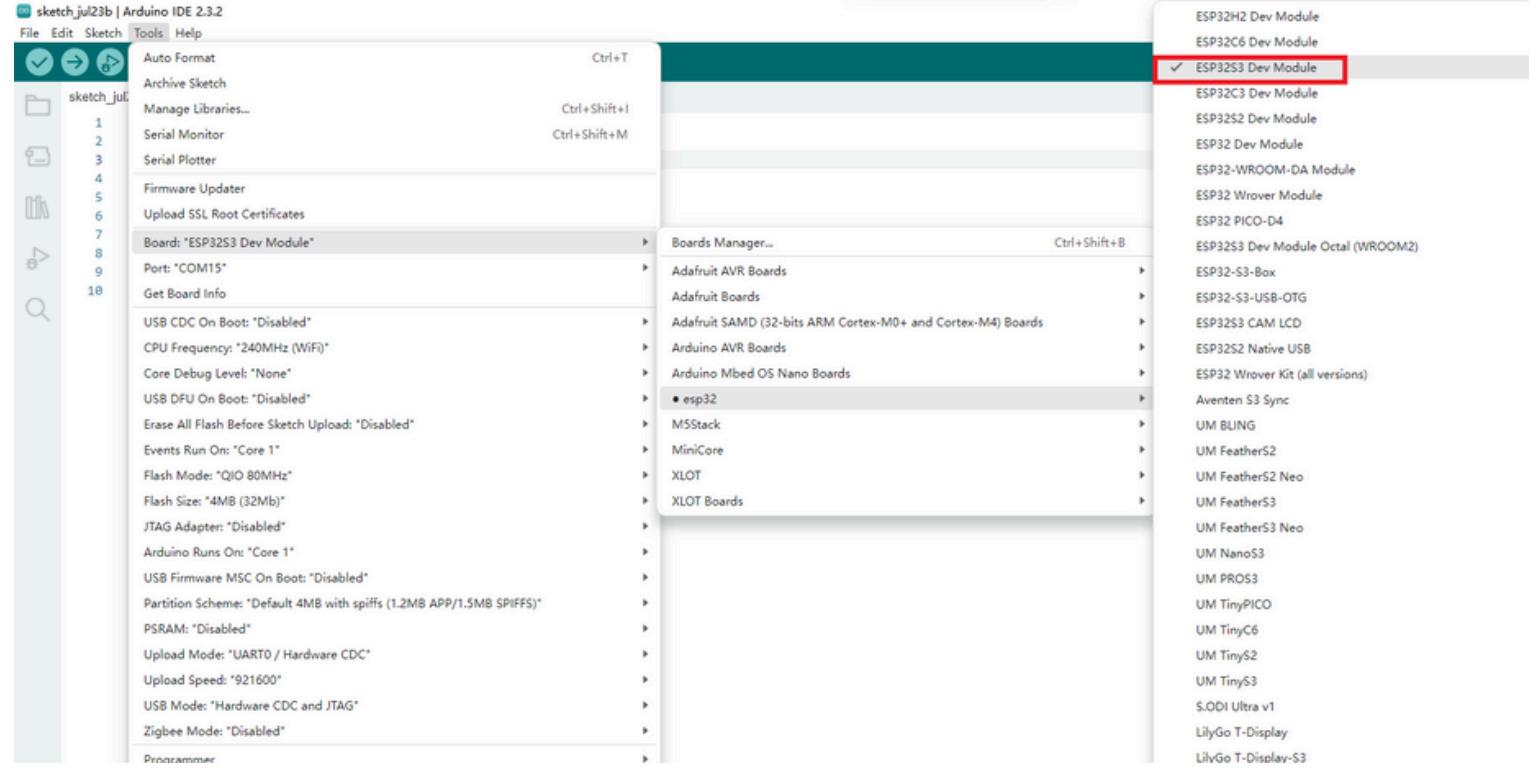


3. In the sidebar, select Board Manager, search for ESP, and click Install.



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