M5STACK

Atom EchoS3R

1. Description

Atom EchoS3R is a highly integrated IoT voice interaction controller specially designed for intelligent voice control and human-computer interaction scenarios. At its core is the ESP32-S3-PICO-1-N8R8 main control chip, which supports Wi-Fi wireless communication and comes with built-in 8MB Flash and 8MB PSRAM, meeting the diverse application development needs and providing excellent performance and scalability. The audio system employs the ES8311 monaural codec, combined with a high-sensitivity MEMS microphone and NS4150B power amplifier, to achieve clear sound pickup and high-fidelity audio output, enhancing the voice recognition and interaction experience. It is suitable for voice interaction scenarios such as AI voice assistants and smart home control.

2. Specifications

Specification	Parameters
SoC	Specification
PSRAM	ESP32-S3-PICO-1-N8R8 @ Dual-core Xtensa LX7 processor, up to 240MHz main frequency
Flash	8MB
Input Power	8MB
Audio CodeC	USB: DC 5V
MEMS Microphone	ES8311: 24-bit resolution, using I2S protocol
Power Amplifier	MSM381A3729H9BPC, Signal-to-Noise Ratio (SNR): ⩾65 dB
Speaker	1318 cavity speaker: 1W@8Ω
Operating Temperature	0 ~ 40°C
Product Size	24.0 x 24.0 x 16.8mm

3. Quick Start

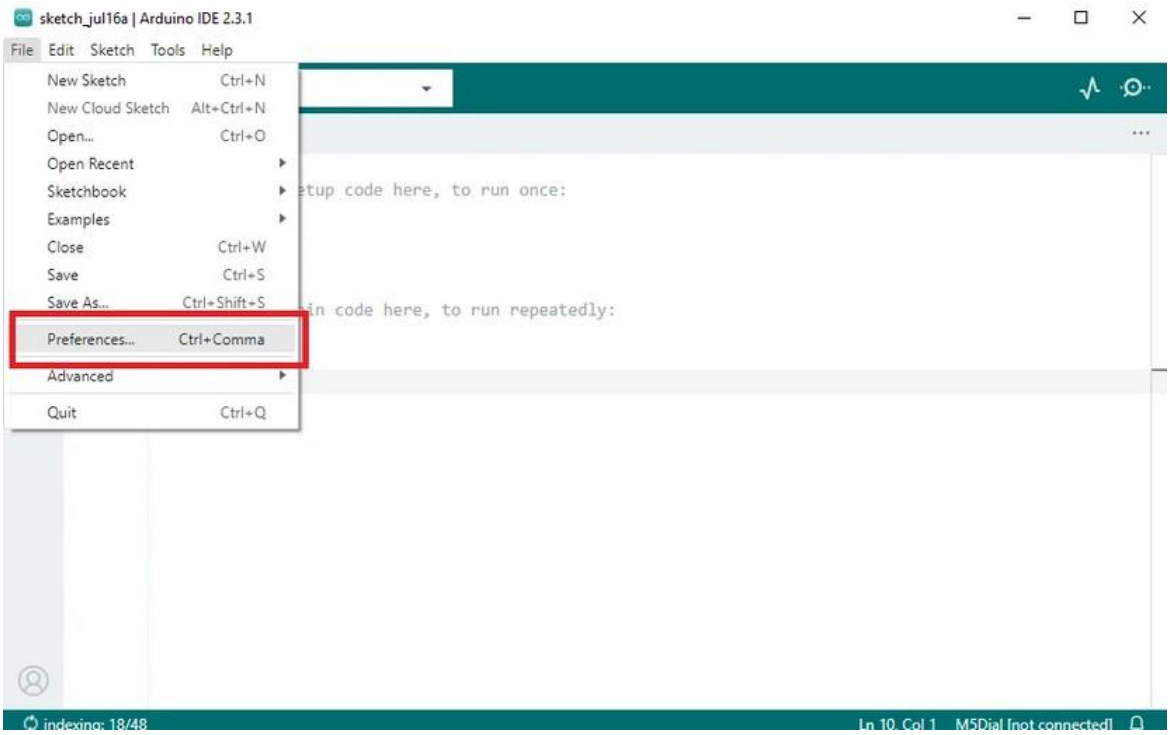
3.1 Preparation

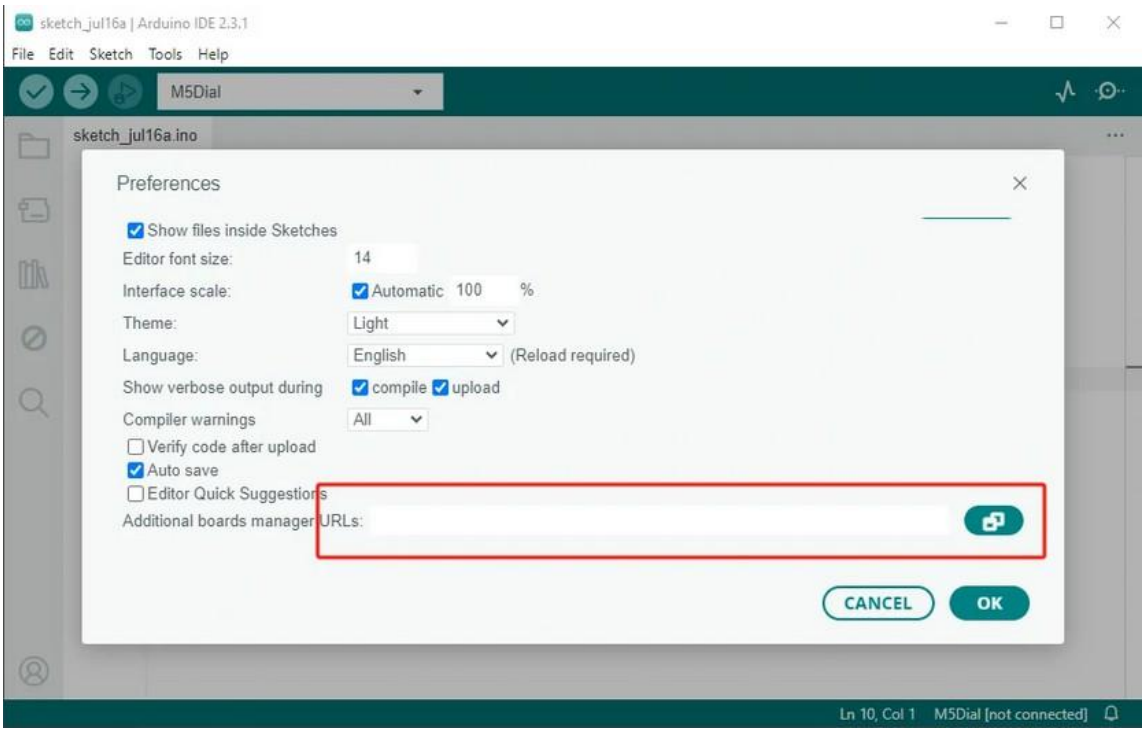
1. Visit the official Arduino website and install the Arduino IDE

https://www.arduino.cc/en/Main/Software

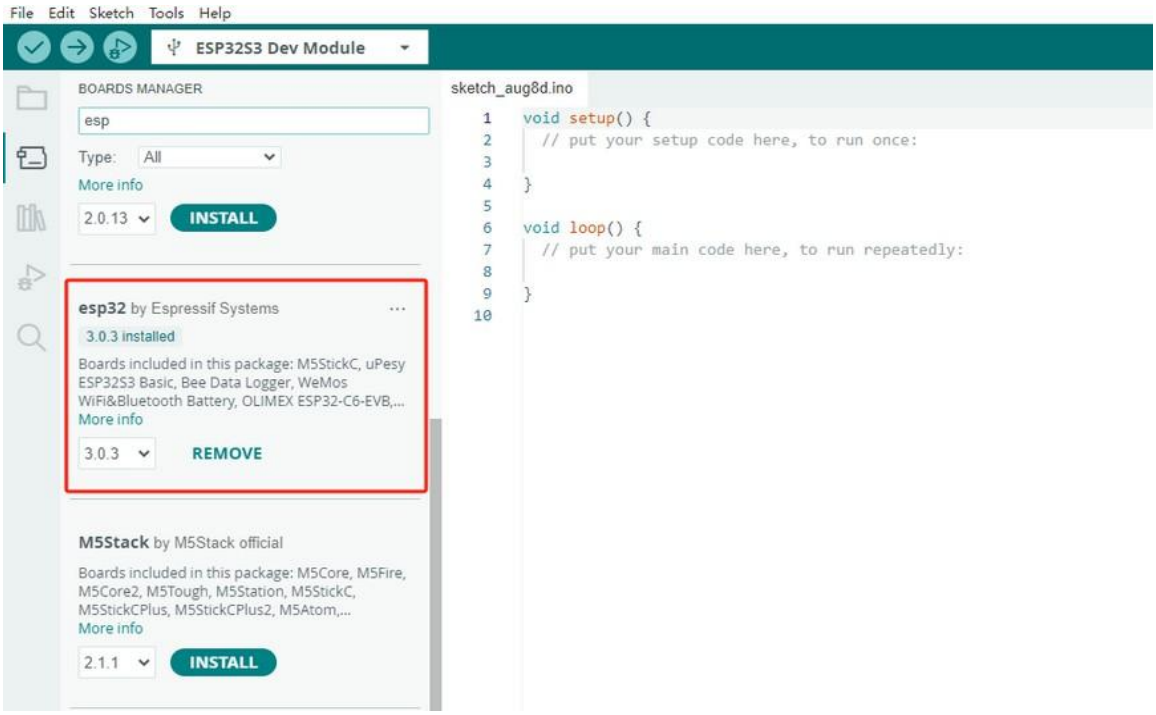
2. Add the following Board Manager URL to File → Preferences → Additional Boards Manager URLs:

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

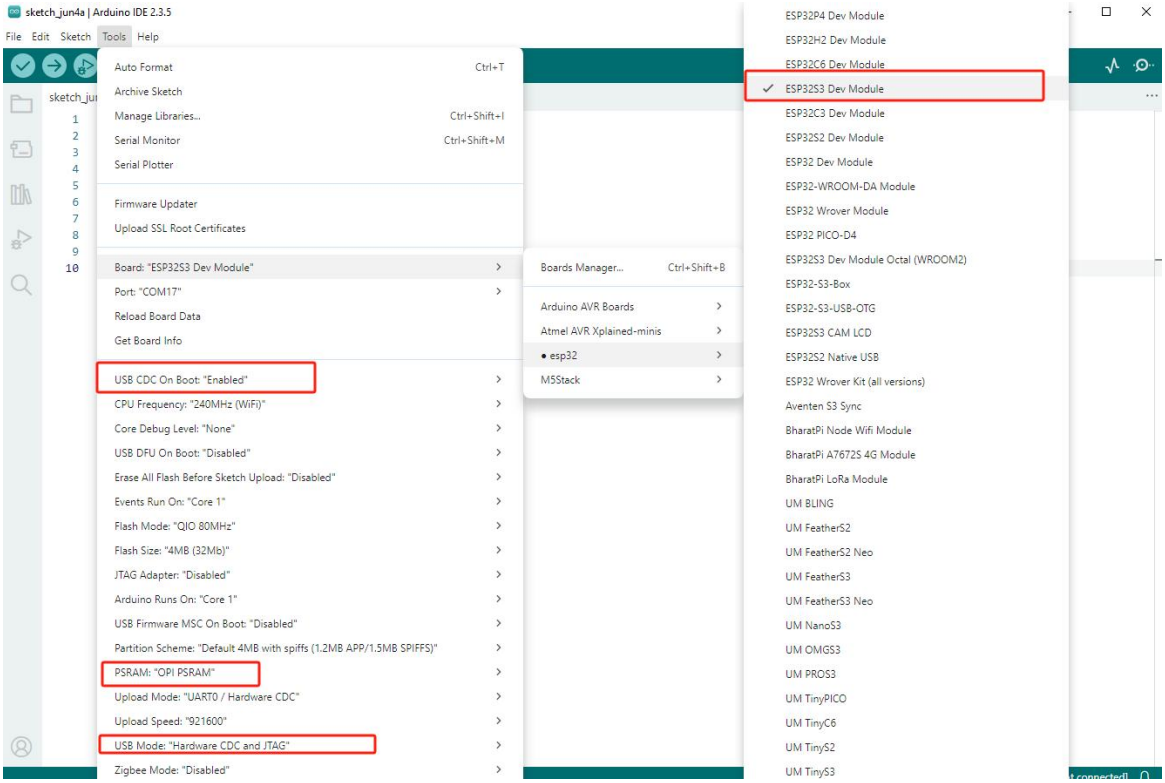




3. Open the Boards Manager, search for "ESP32", and click install.

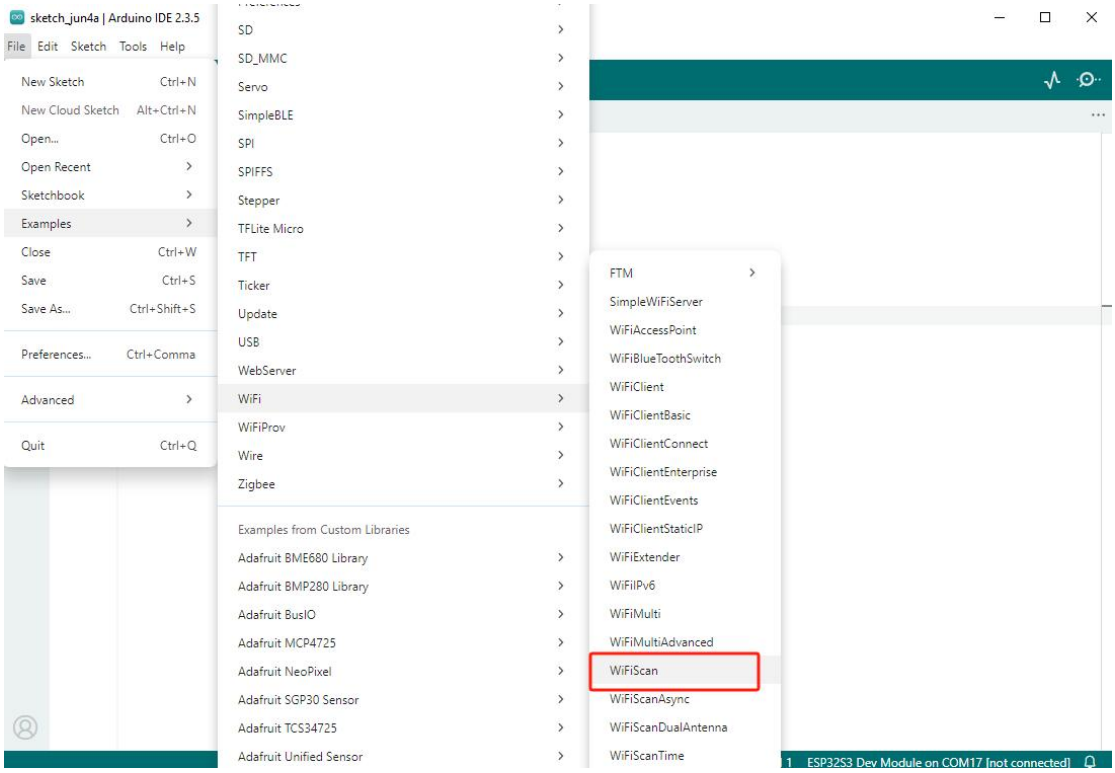


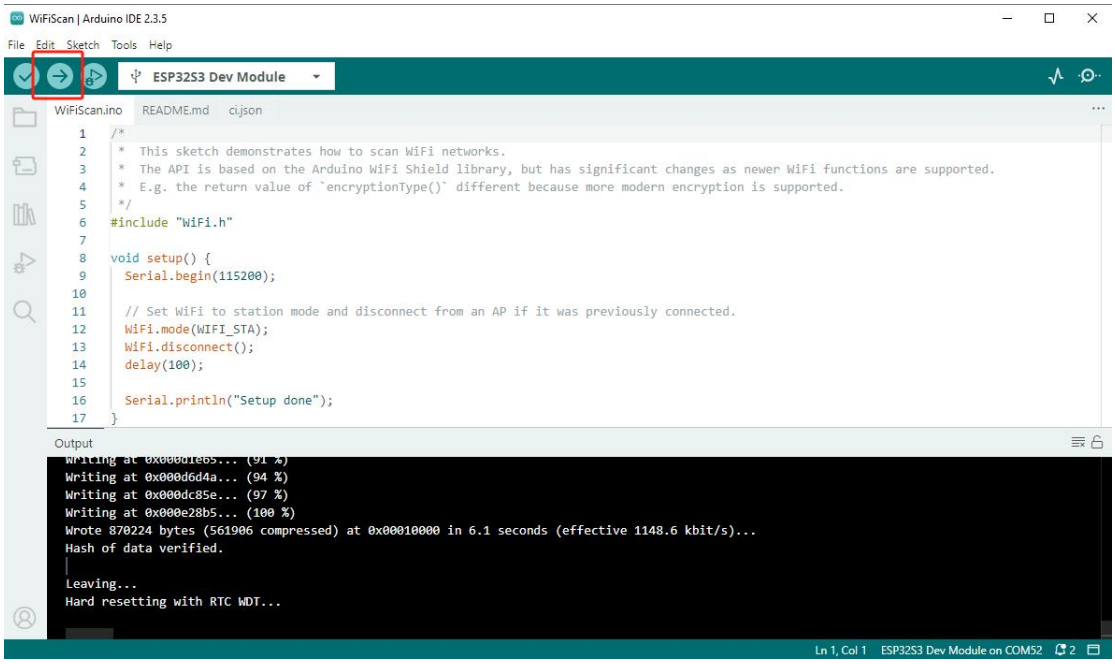
4. After installation, select the board "ESP32S3 Dev Module"
5. Configure the following options. USB CDC On Boot: "Enabled", PSRAM: "OPI PSRAM", USB Mode: "Hardware CDC and JTAG"

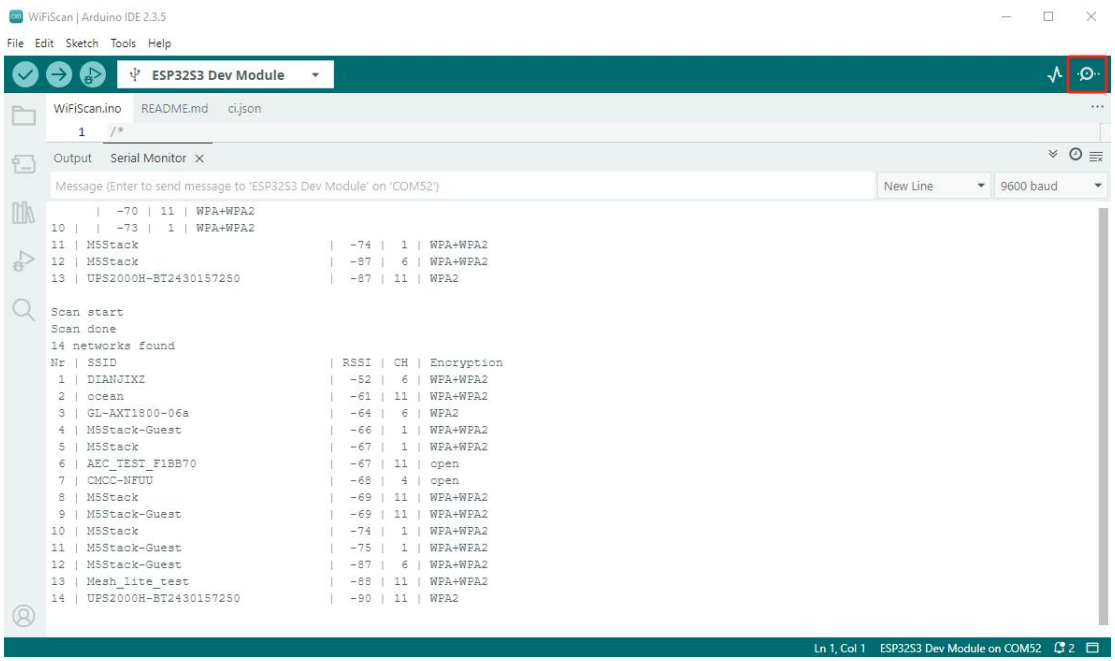


3.2 Wi-Fi Scan

Select the example program "Examples" → "WiFi" → "WiFiScan", choose the port corresponding to your device, and click the compile and upload button in the top-left corner. After uploading is complete, open the Serial Monitor to view Wi-Fi scan information.

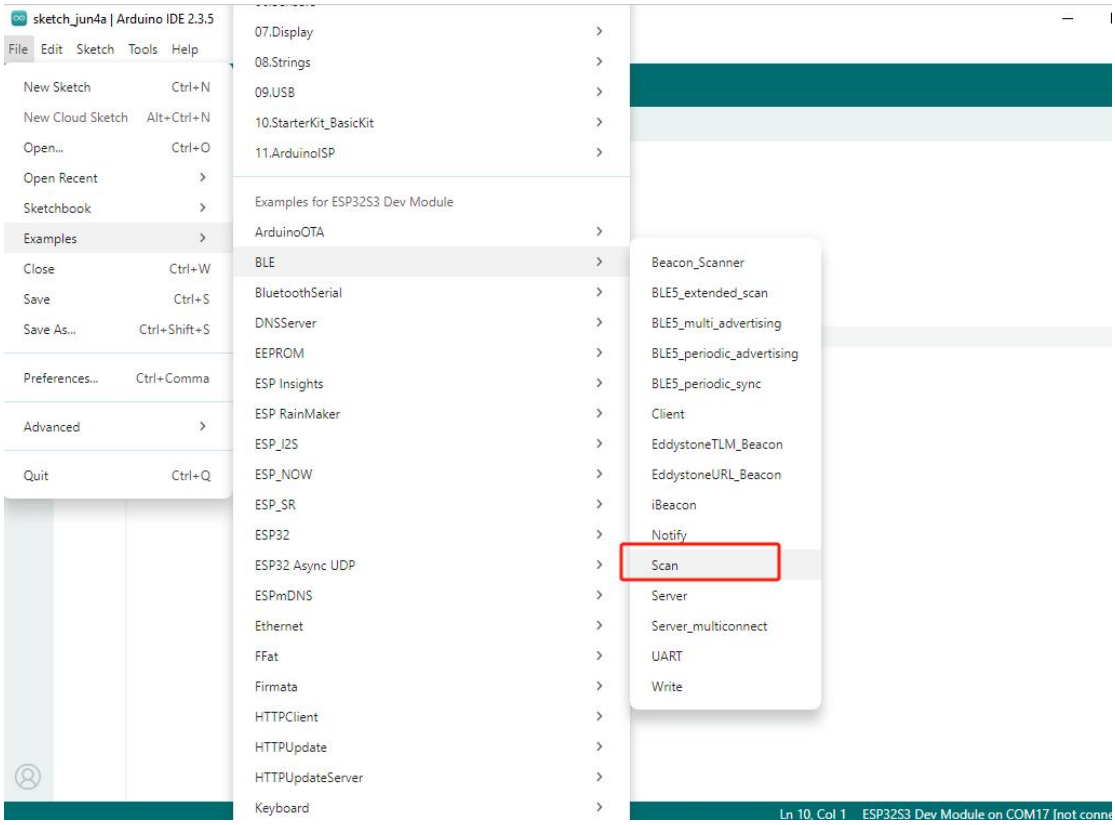


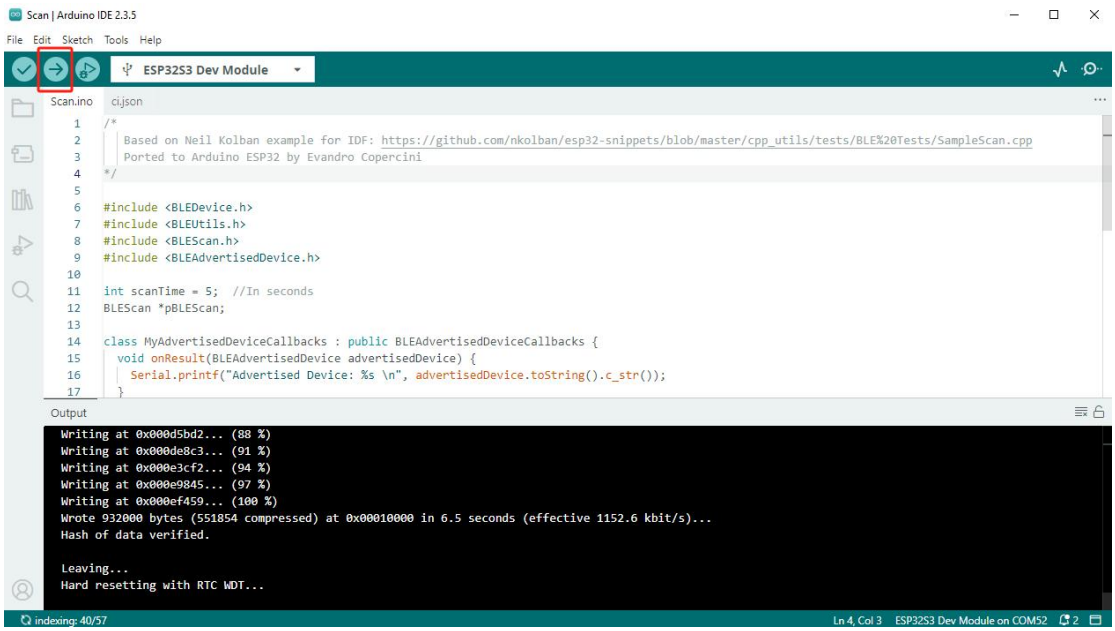


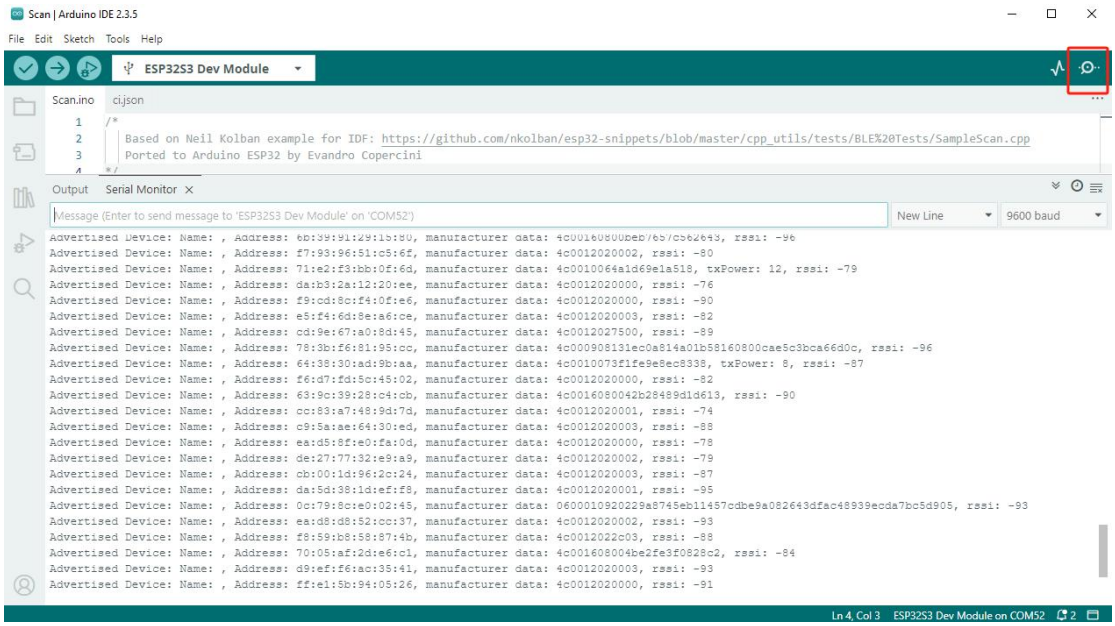


3.3 BLE Scan

Select the example program "Examples" → "BLE" → "Scan", choose the port corresponding to your device, and click the compile and upload button in the top-left corner.
After uploading is complete, open the Serial Monitor to view BLE scan information.


sketch_jun4a | Arduino IDE 2.3.5
File Edit Sketch Tools Help
New Sketch Ctrl+N
New Cloud Sketch Alt+Ctrl+N
Open... Ctrl+O
Open Recent
Sketchbook
Examples
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
Preferences... Ctrl+Comma
Advanced
Quit Ctrl+Q
07.Display
08.Strings
09.USB
10.StarterKit_BasicKit
11.ArduinoISP
Examples for ESP32S3 Dev Module
ArduinoOTA
BLE
BluetoothSerial
DNSServer
EEPROM
ESP Insights
ESP RainMaker
ESP_I2S
ESP_NOW
ESP_SR
ESP32
ESP32 Async UDP
ESPmDNS
Ethernet
FFat
Firmata
HTTPClient
HTTPUpdate
HTTPUpdateServer
Keyboard
Beacon_Scanner
BLE5_extended_scan
BLE5_multi_advertising
BLE5_periodic_advertising
BLE5_periodic_sync
Client
EddystoneTLM_Beacon
EddystoneURL_Beacon
iBeacon
Notify
Scan
Server
Server_multiconnect
UART
Write
Ln 10, Col 1 ESP32S3 Dev Module on COM17 [not conne


Scan | Arduino IDE 2.3.5
File Edit Sketch Tools Help
ESP32S3 Dev Module
Scan.ino
ci.json
/*
Based on Neil Kolban example for IDF: https://github.com/nkolban/esp32-snippets/blob/master/cpp_utils/tests/BLE%20Tests/SampleScan.cpp
Ported to Arduino ESP32 by Evandro Copercini
*/
#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());
}
Output
Writing at 0x000d5bd2... (88 %)
Writing at 0x000de8c3... (91 %)
Writing at 0x000e3cf2... (94 %)
Writing at 0x000e9845... (97 %)
Writing at 0x000ef459... (100 %)
Wrote 932000 bytes (551854 compressed) at 0x00010000 in 6.5 seconds (effective 1152.6 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting with RTC WDT...
indexing: 40/57
Ln 4, Col 3 ESP32S3 Dev Module on COM52



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.

The device has been evaluated to meet general RF exposure requirement. The device can be used in portable exposure condition without restriction.