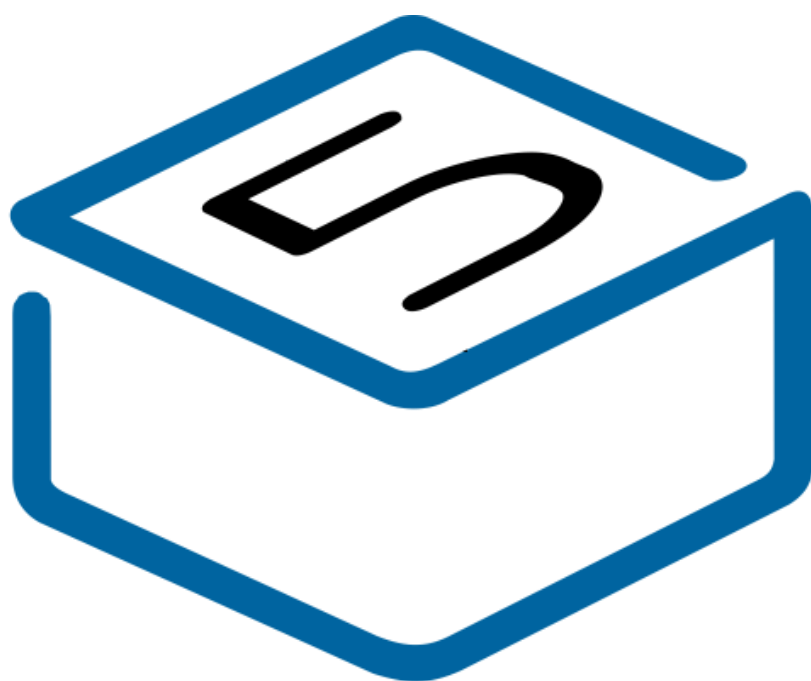


M5STACK

NanoH2



M5STACK

2025

CONTENTS

- CONTENTS 2
- 1.OUTLINE 3
 - 1.1.ESP32-H2 3
- 2.SPECIFICATIONS 4
- 3.QUICK START..... 5
 - 3.1.ARDUINO IDE 5
 - 3.2.BLUETOOTH SERIAL 5
- 4.ArduinoInstall..... 6
- 5.User manual, how to use zigbee function..... 7
 - 5.1.Zigbee OnOff Light(End Device)..... 7
 - 5.2.Zigbee OnOff Switch(Coordinator) 8
 - 5.3.Zigbee Scan Network..... 9
 - 5.4. OpenThread 9
- 6.FCC Warning 10

1.OUTLINE

M5NanoH2 is a miniature, low-power IoT development board within the M5Stack development kit series. Powered by the ESP32-H2 MCU, it boasts advanced wireless communication support Zigbee, facilitating seamless control of infrared IoT devices through its built-in infrared transmitter. The onboard ceramic antenna ensures a stable wireless communication connection. Additionally, the device features programmable RGB LEDs, adding a personalized visual touch to projects. The inclusion of Grove interfaces allows M5NanoH2 to flexibly expand with various M5 devices, supporting the connection of different device types through protocols such as UART and I2C. This provides developers with abundant hardware expansion possibilities. Suitable for applications in smart homes, industrial automation, health monitoring, and IoT devices, M5NanoH2 delivers a comprehensive development solution for innovative projects.

1.1.ESP32-H2

1)Communication Capabilities:

- Bluetooth® 5 (LE): Incorporates Bluetooth 5.0 technology for extended wireless coverage and faster data transmission.
- Zigbee and Thread (802.15.4): Supports Zigbee and Thread communication protocols, offering flexible connectivity for IoT applications.

2)Processor and Performance:

- RISC-V 32-bit single-core processor: Delivers a highly flexible and scalable processor architecture.
- Up to 160 MHz clock frequency: Ensures the device has fast and efficient data processing capabilities.
- Industry-leading low-power and RF performance: Achieves a leading position in both power consumption and RF performance.

3)Memory:

- Built-in 96 KB ROM: Used for storing firmware or program code.
- 320 KB SRAM: Utilized for runtime data storage.
- 4 KB low-power SRAM: Specifically designed for low-power operations.
- Supports external Flash: Can extend the storage space for program code through an external memory.

4)GPIO Pins and Programmable Interfaces:

- Supports SPI, UART, I2C, I2S, RMT, TWAI, and PWM: Multiple communication interfaces, enhancing flexibility for data exchange with other de

2.SPECIFICATIONS

Specification	Parameter
MCU	ESP32-H2FH4S@RISC-V 32-bit single-core processor 96MHZ, 128 KB ROM, 320 KB SRAM, 4 KB low-power SRAM, supports external flash
Programmable RGB	WS2812-2020
Input voltage	5V
Communication mode	Zigbee And Thread (802.15.4)And other mainstream communication methods
Communication Protocol	Support SPI、UART、I2C、I2S、RMT、TWAI And PWM
Antenna type	Multilayer Antenna
Other peripherals	Board infrared transmitter, programmable onboard control buttons
Operating temperature	0-40°C

3.QUICK START

3.1.ARDUINO IDE

To access the Arduino official website (<https://www.arduino.cc/en/Main/Software>) and download the installation package for your operating system, follow these steps:

- 1)Open the Arduino IDE and navigate to **File -> Preferences -> Settings**.
- 2)Copy the following M5Stack Boards Manager URL and paste it into "Additional Boards Manager URLs": https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/arduino/package_m5stack_index.json
- 3)Go to: **Tools -> Board: -> Boards Manager...**
- 4)Search for **M5Stack**, find it, and click "Install."
- 5)Choose **Tools -> Board: -> M5Stack Arduino**
- 6)Select **M5NanoH2**

3.2.BLUETOOTH SERIAL

- Open the Arduino IDE and load the example program: **File -> Examples -> BluetoothSerial -> SerialToSerialBT**.
- Connect the device to the computer.
- Choose the appropriate port for uploading. Once completed, the device will automatically enable Bluetooth.
- The device name is set to **ESP32test**. Now, use a Bluetooth serial communication tool on your PC to achieve transparent transmission of Bluetooth serial data.

4.ArduinoInstall

```
#include "BluetoothSerial.h"

//#define USE_PIN // Uncomment this to use PIN during pairing. The pin is specified on the line below
const char *pin = "1234"; // Change this to more secure PIN.

String device_name = "ESP32test";

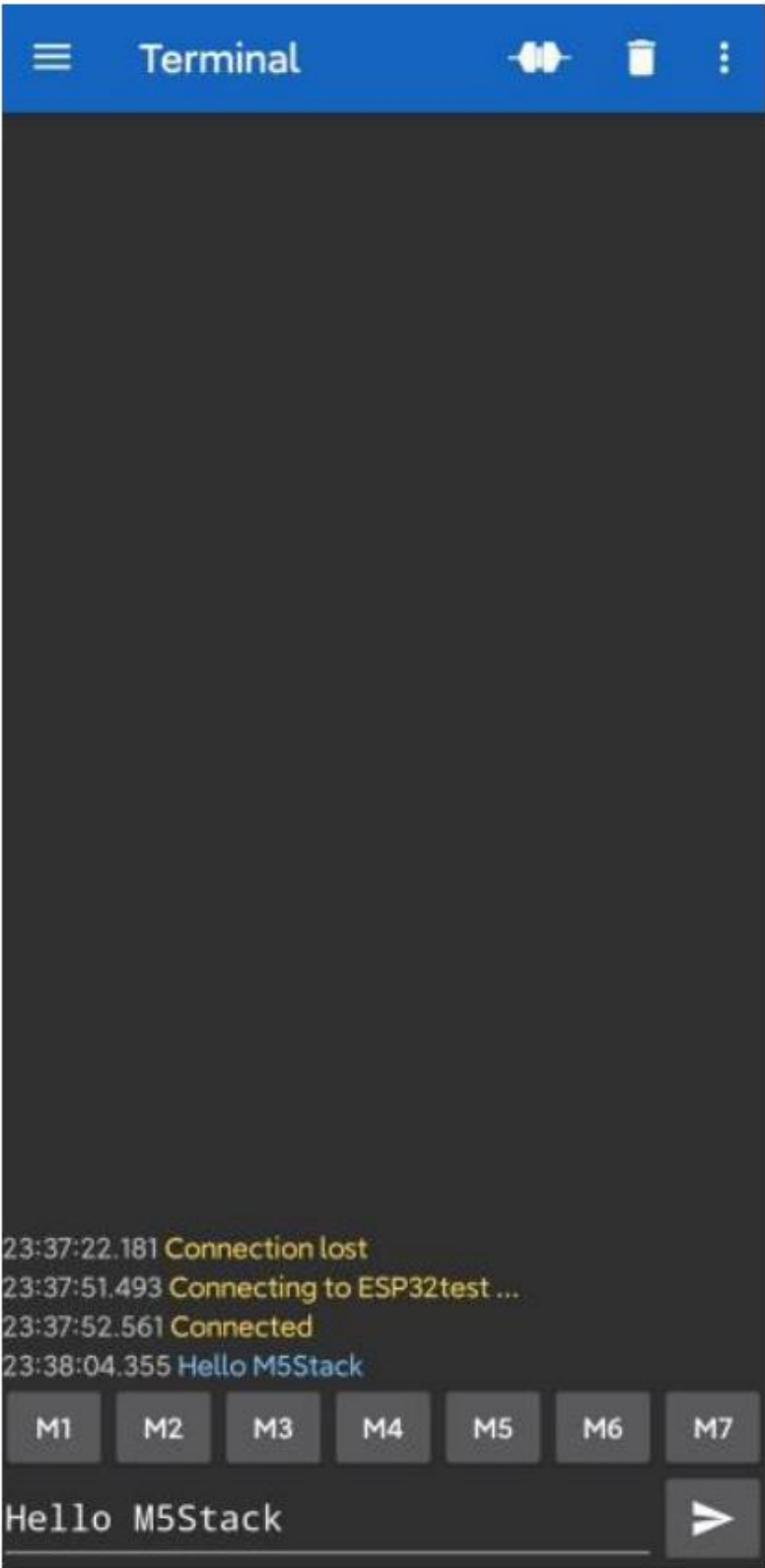
#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif

#if !defined(CONFIG_BT_SPP_ENABLED)
#error Serial Bluetooth not available or not enabled. It is only available for the ESP32 chip.
#endif

BluetoothSerial SerialBT;

void setup() {
  Serial.begin(115200);
  SerialBT.begin(device_name); //Bluetooth device name
  Serial.printf("The device with name \"%s\" is started.\nNow you can pair it with Bluetooth!\n", devi
  //Serial.printf("The device with name \"%s\" and MAC address %s is started.\nNow you can pair it wit
  #ifdef USE_PIN
    SerialBT.setPin(pin);
    Serial.println("Using PIN");
  #endif
}

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



5.User manual, how to use zigbee function

5.1.Zigbee OnOff Light(End Device)

Arduino IDE Tools:

- Tools -> Board: ESP32H2 Dev Module
- Tools -> Erase All Flash Before Sketch Upload: Enable
- Tools -> Flash Size: 2MB
- Tools -> Zigbee mode: Zigbee ED (end device)
- Tools -> Partition Scheme: custom

#	Name,	Type,	SubType,	Offset,	Size,	Flags
nvs,		data,	nvs,	0x9000,	0x5000,	
otadata,		data,	ota,	0xe000,	0x2000,	
app0,		app,	ota_0,	0x10000,	0xC0000,	
app1,		app,	ota_1,	0xd0000,	0xC0000,	
spiffs,		data,	spiffs,	0x190000,	0x5a000,	
zb_storage,		data,	fat,	0x1ea000,	0x4000,	
zb_fct,		data,	fat,	0x1ee000,	0x1000,	
coredump,		data,	coredump,	0x1f0000,	0x10000,	

zigbee_end_device | Arduino IDE 2.3.4

File Edit Sketch Tools Help

12345678910111213141516171819202122232425

Tools menu:

- Auto Format (Ctrl+T)
- Archive Sketch
- Manage Libraries... (Ctrl+Shift+I)
- Serial Monitor (Ctrl+Shift+M)
- Serial Plotter
- Firmware Updater
- Upload SSL Root Certificates
- Board: "ESP32H2 Dev Module"
- Port
- Reload Board Data
- Get Board Info
- USB CDC On Boot: "Disabled"
- Core Debug Level: "Verbose"
- Erase All Flash Before Sketch Upload: "Enabled"
- Flash Frequency: "64MHz"
- Flash Mode: "QIO"
- Flash Size: "2MB (16Mb)"
- JTAG Adapter: "Disabled"
- Partition Scheme: "Custom"
- Upload Speed: "921600"
- Zigbee Mode: "Zigbee ED (end device)"
- Programmer
- Burn Bootloader

Tools->Zigbee mode"

EMP_SENSOR_ENDPOINT_NUMBER);

***** /

value to %.2f°C\r\n", tsens_value);

sensor EP

***** /

zigbee_end_device | Arduino IDE 2.3.4

File Edit Sketch Tools Help

ESP32H2 Dev Module

zigbee_end_device.ino

```
27  /*
28
29  #ifndef ZIGBEE_MODE_ED
30  #error "Zigbee end device mode is not selected"
31  #endif
32
33  #include "Zigbee.h"
34
35  /* Zigbee light bulb configuration
36  #define ZIGBEE_LIGHT_ENDPOINT 1
37  uint8_t led = RGB_BUILTIN;
38  uint8_t button = BOOT_PIN;
39
40  ZigbeeLight zbLight = ZigbeeLight(1, 1);
41
42  /***** RGB LED
43  void setLED(bool value) {
44    if(value){
45      Serial.println("LED ON!");
46    }else{
47      Serial.println("LED OFF!");
48    }
49    digitalWrite(led, value);
50  }
51
52  /***** Arduino
53  void setup() {
54    Serial.begin(115200);
55
56    // Init LED and turn it OFF (if LED_PIN == RGB_BUILTIN, the rgbLedWrite() will be used under the hood)
57    pinMode(led, OUTPUT);
```

partitions.csv - Notepad

File Edit Format View Help

Name, Type, SubType, Offset, Size, Flags

nvs, data, nvs, 0x9000, 0x5000,

otadata, data, ota, 0xe000, 0x2000,

app0, app, ota_0, 0x10000, 0xC0000,

app1, app, ota_1, 0xd0000, 0xC0000,

spiffs, data, spiffs, 0x190000, 0x5a000,

zb_storage, data, fat, 0x1ea000, 0x4000,

zb_fct, data, fat, 0x1ee000, 0x1000,

coredump, data, coredump, 0x1f0000, 0x10000,

File Home

Documents

Downloads

M5Stack-Dev

M5Stack-Market

Music

Pictures

Videos

workspace

WSL2_Ubuntu

System (C:)

Workspace (D:)

2 items | 1 item selected 411 bytes

partitions.csv

zigbee_end_device.ino

Ln 49, Col 28 ESP32H2 Dev Module on COM19

5.2.Zigbee OnOff Switch(Coordinator)

Arduino IDE Tools

- Tools -> Board: ESP32H2 Dev Module
- Tools -> Erase All Flash Before Sketch Upload: Enable
- Tools -> Flash Size: 2MB
- Tools -> Zigbee mode: Zigbee ZCZR (coordinator/router)
- Tools -> Partition Scheme: custom
- Zigbee ZCZR 2MB with spiffs

#	Name,	Type,	SubType,	Offset,	Size,	Flags
nvs,	data,	nvs,		0x9000,	0x5000,	
otadata,	data,	ota,		0xe000,	0x2000,	
app0,	app,	ota_0,		0x10000,	0xC0000,	
app1,	app,	ota_1,		0xd0000,	0xC0000,	
spiffs,	data,	spiffs,		0x190000,	0x5a000,	
zb_storage,	data,	fat,		0x1ea000,	0x4000,	
zb_fct,	data,	fat,		0x1ee000,	0x1000,	
rcp_fw,	data,	spiffs,		0x1ef000,	0x1000,	
coredump,	data,	coredump,		0x1f0000,	0x10000,	

zigbee_coordinator | Arduino IDE 2.3.4

File Edit Sketch Tools Help

✓ → ⚙

zigbee_coordinator.ino

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

Auto FormatCtrl+T

Archive Sketch

Manage Libraries...Ctrl+Shift+I

Serial MonitorCtrl+Shift+M

Serial Plotter

Firmware Updater

Upload SSL Root Certificates

Board: "ESP32H2 Dev Module"▶

Port: "COM17"▶

Reload Board Data

Get Board Info

USB CDC On Boot: "Disabled"▶

Core Debug Level: "Verbose"▶

Erase All Flash Before Sketch Upload: "Enabled"▶

Flash Frequency: "64MHz"▶

Flash Mode: "QIO"▶

Flash Size: "2MB (16Mb)"▶

JTAG Adapter: "Disabled"▶

Partition Scheme: "Custom"▶

Upload Speed: "921600"▶

Zigbee Mode: "Zigbee ZCZR (coordinator/router)"▶

Programmer▶

Burn Bootloader

Tools->Zigbee mode"

...THERMOSTAT_ENDPOINT_NUMBER);

...n", temperature);

...temp, float tolerance) {

...%.2f°C, max %.2f°C, tolerance %.2f°C\n",

zigbee_coordinator | Arduino IDE 2.3.4

File Edit Sketch Tools Help

✓ → ⚙

ESP32H2 Dev Module

zigbee_coordinator.ino

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

void setup() {

Serial.begin(115200);

//Optional: set Zigbee device

zbSwitch.setManufacturerAndMo

//Optional to allow multiple

zbSwitch.allowMultipleBinding

//Add endpoint to Zigbee Core

Serial.println("Adding Zigbee

Zigbee.addEndpoint(&zbSwitch)

//Open network for 180 second

Zigbee.setRebootOpenNetwork(1

// When all EPs are registered, start Zigbee with ZIGBEE_COORDINATOR mode

if (!Zigbee.begin(ZIGBEE_COORDINATOR)) {

partitions.csv - Notepad

File Edit Format View Help

Name, Type, SubType, Offset, Size, Flags

nvs, data, nvs, 0x9000, 0x5000,

otadata, data, ota, 0xe000, 0x2000,

app0, app, ota_0, 0x10000, 0xC0000,

app1, app, ota_1, 0xd0000, 0xC0000,

spiffs, data, spiffs, 0x190000,0x5a000,

zb_storage, data, fat, 0x1ea000,0x4000,

zb_fct, data, fat, 0x1ee000,0x1000,

rcp_fw, data, spiffs, 0x1ef000,0x1000,

coredump, data, coredump,0x1f0000,0x10000,

File Home

Quick access

Desktop

Downloads

Documents

Pictures

PIC

Temp

zigbee_coordinator

zigbee_test

OneDrive

2 items 1 item selected 456 bytes

Ln 60, Col 1 ESP32H2 Dev Module on COM17 2

Usage steps

- 1.Ensure that the coordinator is running and creating a network, burn the OnOff Light code to the terminal node device.
- 2. After the device is started, it will automatically search and join the network, and OnOff Switch will send light switching instructions at regular intervals.

5.3.Zigbee Scan Network

Arduino IDE Tools

- Tools -> Board: ESP32H2 Dev Module
- Tools -> Erase All Flash Before Sketch Upload: Enable
- Tools -> Flash Size: 2MB
- Tools -> Zigbee mode: Zigbee ZCZR (coordinator/router)
- Tools -> Partition Scheme: custom
- Zigbee ZCZR 2MB with spiffs

#	Name,	Type,	SubType,	Offset,	Size,	Flags
nvs,		data,	nvs,	0x9000,	0x5000,	
otadata,		data,	ota,	0xe000,	0x2000,	
app0,		app,	ota_0,	0x10000,	0xC0000,	
app1,		app,	ota_1,	0xd0000,	0xC0000,	
spiffs,		data,	spiffs,	0x190000,	0x5a000,	
zb_storage,		data,	fat,	0x1ea000,	0x4000,	
zb_fct,		data,	fat,	0x1ee000,	0x1000,	
rcp_fw,		data,	spiffs,	0x1ef000,	0x1000,	
coredump,		data,	coredump,	0x1f0000,	0x10000,	

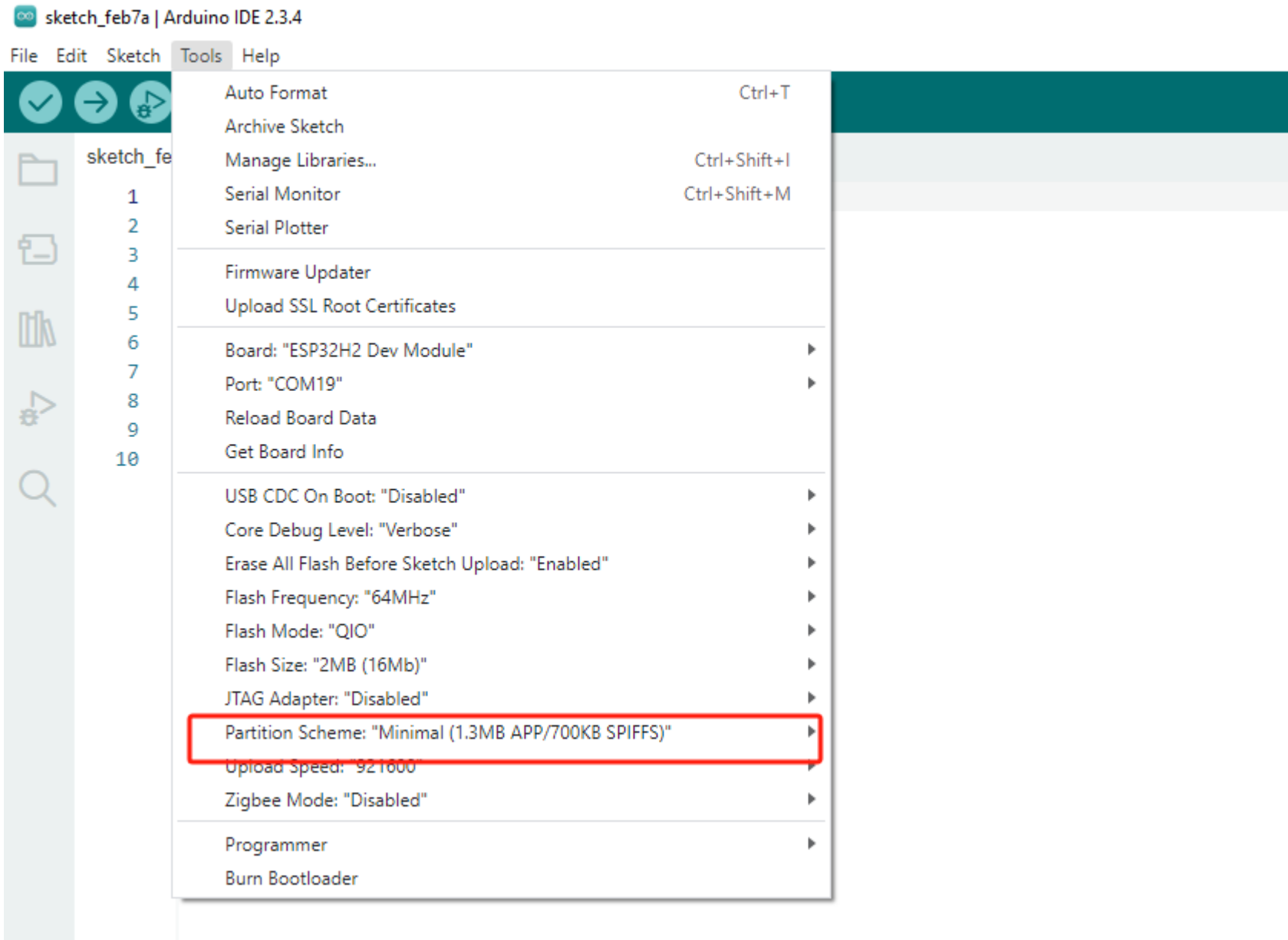
Usage steps

- 1. After the device is started, it automatically starts scanning. If there is an active Zigbee network in the surrounding area, the current display result will be scanned after each scan is completed, and the next round of scanning will be automatically started.

5.4、OpenThread

Arduino IDE Tools

- Tools -> Board: ESP32H2 Dev Module
- Tools -> Erase All Flash Before Sketch Upload: Enable
- Tools -> Flash Size: 2MB
- Tools -> Partition Scheme: Minimal SPIFFS (1.3MB APP/700K SPIFFS)



6.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.

FCC Radiation Exposure Statement: 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.

Frequency Range:

BLE 1M/2M: 2402-2480MHz
ZigBee: 2405-2480MHz

Maximum EIRP for CE:

BLE: 1.32 dBm
ZigBee: 1.4 dBm

Manufacturer : M5Stack Technology Co., Ltd

Block A10, Expo Bay South Coast, Fuhai Street, Bao'an District, Shenzhen, China