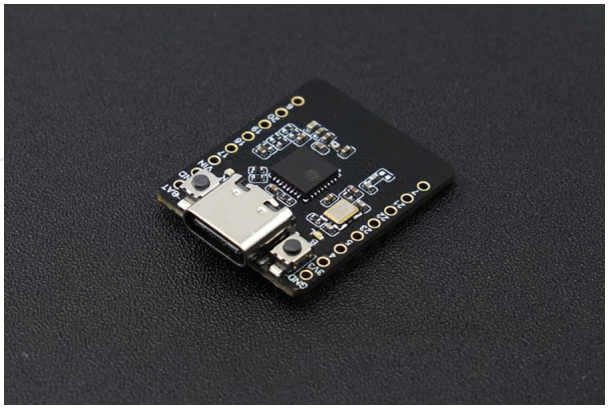
SKU:DFR1117 (https://www.dfrobot.com/product-2778.html)

(https://www.dfrobot.com/product-2778.html)

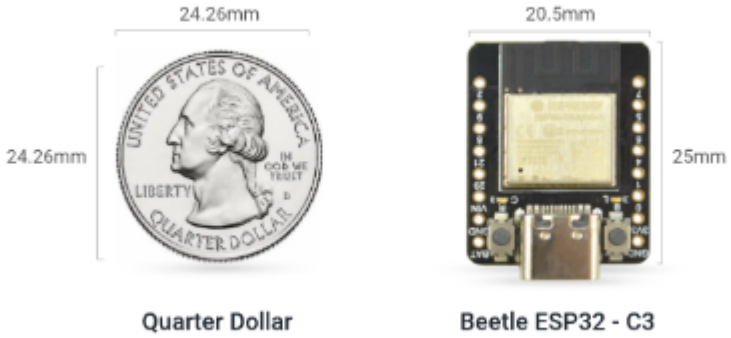


Introduction

The Beetle ESP32-C6, a minuscule Arduino IoT development board designed around the ESP32-C6 chip, exhibits a low-power consumption and is as small as a coin, measuring only 25*20.5mm. The ESP32-C6, equipped with a high-performance 160MHz RISC-V 32-bit processor, supports communication protocols such as Wi-Fi 6, Bluetooth 5, Zigbee 3.0, and Thread 1.3, thus allowing it to connect to various IoT networks. The board also integrates a lithium battery charging management system, eliminating the need for additional modules, thus making the project compact.

Highly Integrated, Ultra-Small Volume

The Beetle ESP32-C6, despite its coin-size volume, offers up to 13 IO ports, erasing any concerns about a lack of IOs while creating projects. It integrates a lithium battery charging management feature, ensuring safe charging of the lithium battery. This development board also supports battery voltage monitoring, enabling actions to be taken when the power level is insufficient to ensure the device's continuous operation.

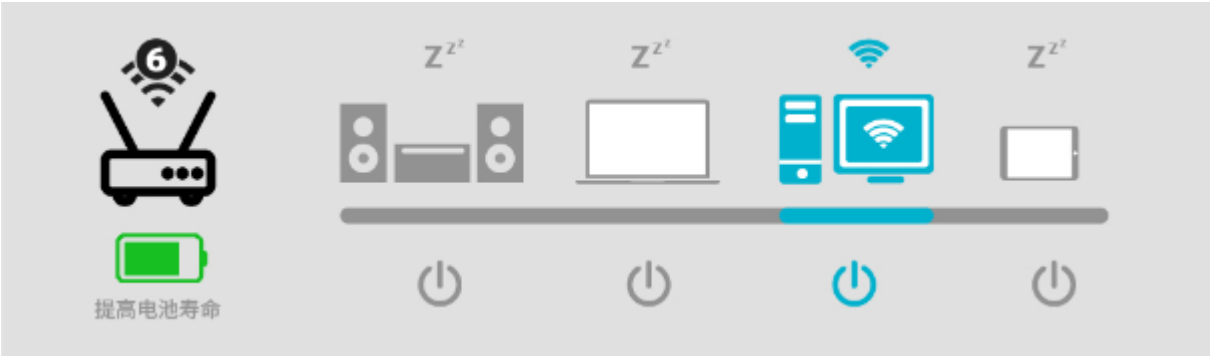


Support for Various Transmission Protocols, Expanded Wireless Connectivity

The Beetle ESP32-C6 supports Wi-Fi and Thread communication protocols, thus it can be used to create Matter Wi-Fi terminal devices and Matter Thread terminal devices, achieving seamless communication and cooperation between multi-system, multi-platform smart home devices. Furthermore, the Beetle ESP32-C6 also supports BLE and Zigbee communication protocols. When combined with other MCUs, it can serve as a Thread border router, Matter gateway, and Zigbee bridge.

Supports Wi-Fi 6, Enabling Ultra-Low Power IoT Devices

The ESP32-C6 is Espressif's first chip to support the Wi-Fi 6 protocol (802.11ax), which provides improved network capacity, enabling devices to operate with high efficiency and low latency. Moreover, the Target Wake Time (TWT) technology of Wi-Fi 6 effectively reduces device power consumption, prolongs battery life, and ensures long-term device operation.



Features

- Boasting an ultra-compact size, the dimensions are a mere 25*20.5mm.
- Equipped with the ESP32-C6 chip, it supports Wi-Fi, BLE, Zigbee, and Thread communication protocols.
- Supports the Wi-Fi 6 protocol, ensuring lower latency and reduced power consumption.
- Exhibits ultra-low power consumption, with a deep-sleep mode of 14uA.
- Incorporates a lithium battery charging function.
- Supports battery voltage detection, providing insight into device power levels.

Application Scenarios

- Intelligent lighting projects.
- Smart environmental monitoring projects.
- Intelligent switch projects.

Specification

Basic Parameters

- Operating Voltage: 3.3V
- Type-C Input Voltage: 5V DC
- Max Charging Current: 0.5A
- Sleep current: 14uA (in deep sleep mode, powered by battery)
- Operating Temperature: -10~60°C

- Dimension: 20.5x25mm

Hardware Information

- Processor: RISC-V single-core processor
- Main Frequency: 160 MHz
- SRAM: 512KB
- ROM: 320KB
- Flash: 4MB
- RTC SRAM: 16KB
- USB: USB 2.0 CDC

WIFI

- WIFI Protocol: IEEE 802.11b/g/n
  - IEEE 802.11ax (20 MHz-only non-AP mode)
- Bandwidth: Support 20 MHz and 40 MHz at 2.4 GHz band
- WIFI Mode: Station, SoftAP, SoftAP+Station combined mode
- WIFI Frequency: 2.4GHz
- Frame Aggregation: TX/RX A-MPDU, TX/RX A-MSDU

Bluetooth

- Bluetooth Protocol: Bluetooth 5, Bluetooth mesh
- Bluetooth Frequency: 125 Kbps, 500 Kbps, 1 Mbps, 2 Mbps

IEEE 802.15.4

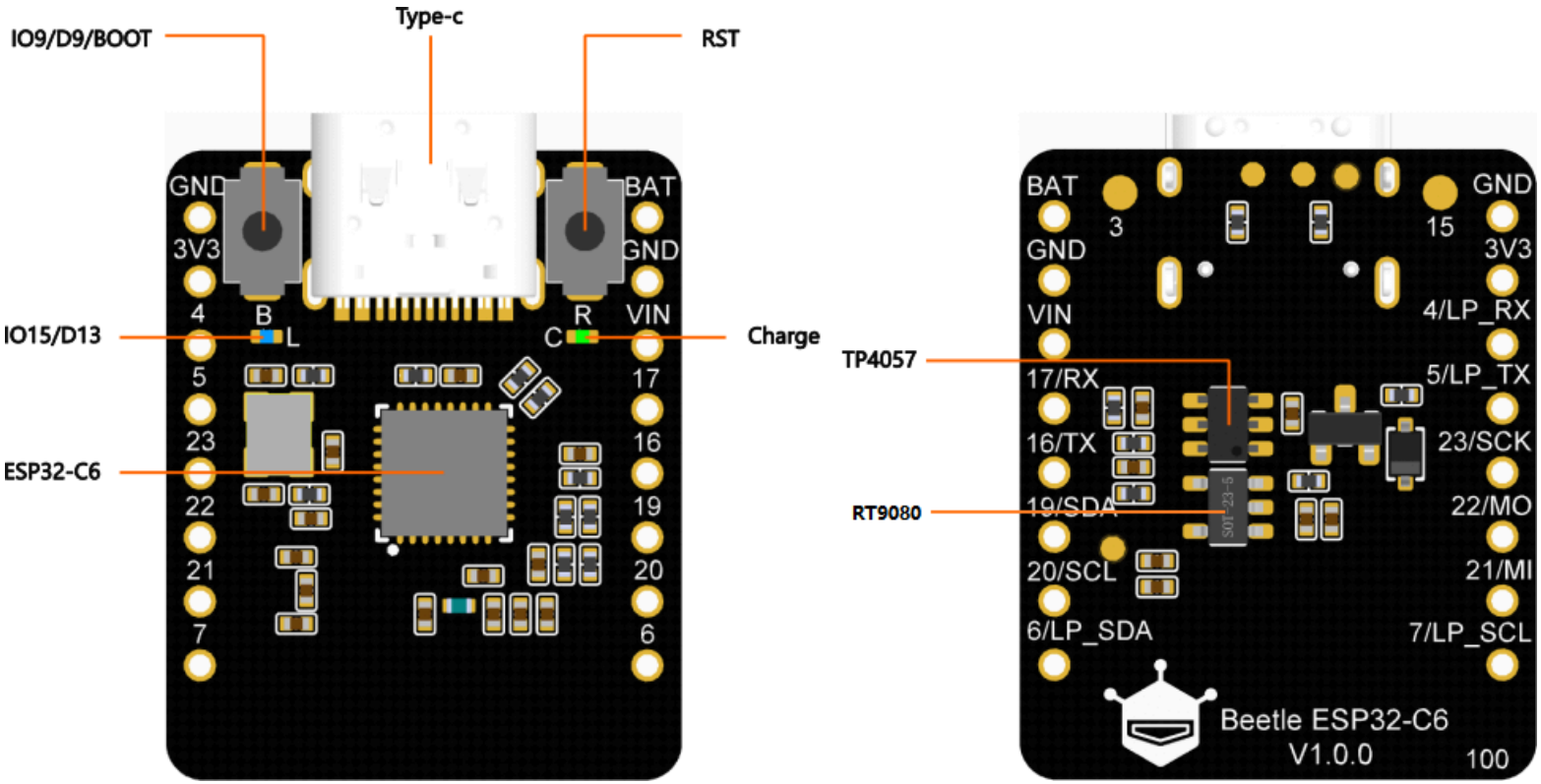
- Compatible with IEEE 802.15.4-2015 protocol
- Frequency band: 2.4GHz

- Data rate: 250Kbps
- Supports Thread 1.3 and Zigbee 3.0

Ports

- Digital I/O x13
- LED PWM 6 Channel
- SPI x1
- UART x3 (LP UART x1)
- I2C x2 (LP I2C x1)
- I2S x1
- IR Transceiver: transmit channel x5, receive channel x5
- 1 × 12-bit SAR ADC, 7 Channel
- DMA Controller: transmit channel x3, receive channel x3

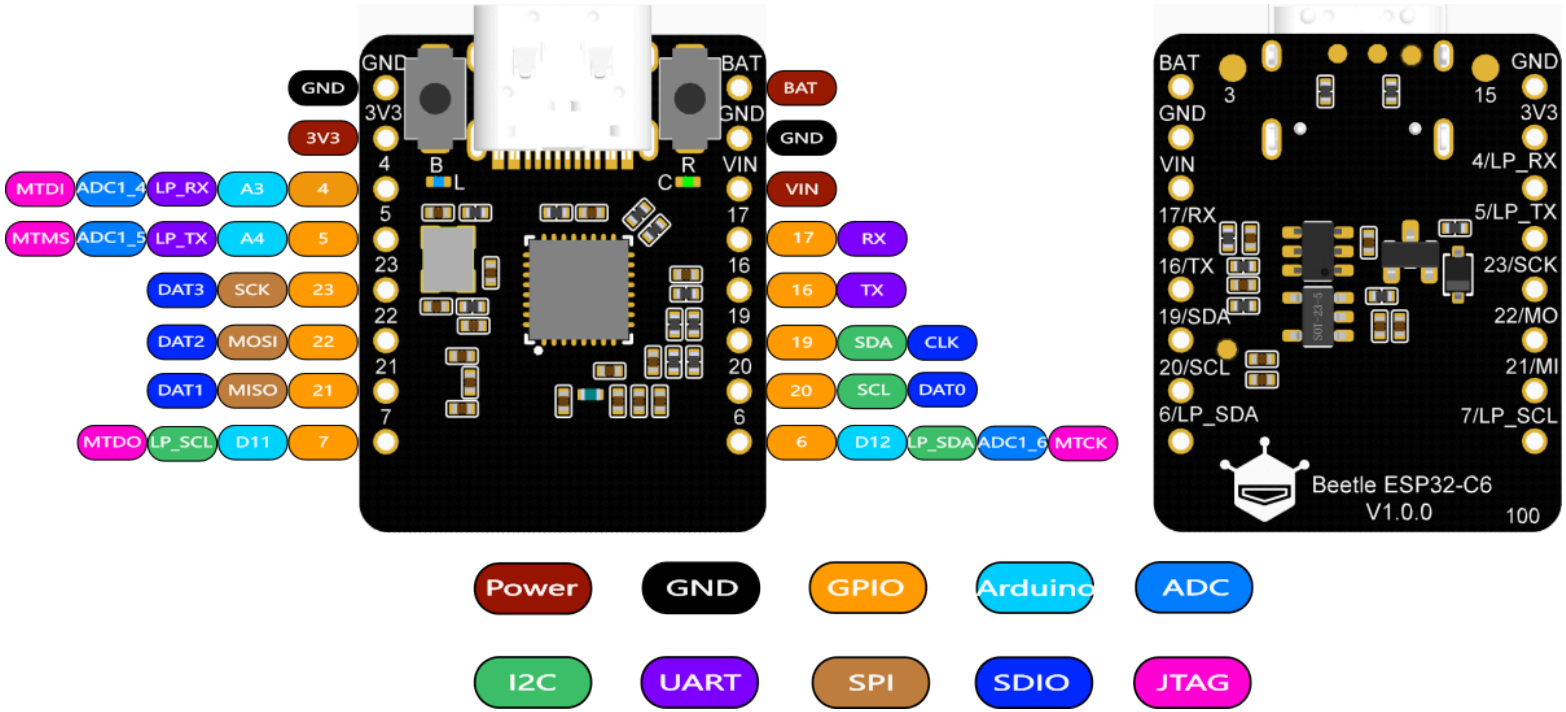
Board Overview



- **Type-C:**Type-C USB port
- **IO15/D13:**onboard LED pin
- Charge:Charging indicator
  - Off: n ot plugged in power supply or fully charged
  - On: charging
  - Blinking: battery not connected
- **RST:** Reset button

- **IO9/D9/BOOT:** GPIO9 / Boot button
- **ESP32-C6 (https://dfimg.dfrobot.com/60c1e008bddfc41c3293de80/wiki/bd5febb16dd4b86eb34cb387815080ae.pdf)**: ESP32-C6FH4 chip
- **TP4057**: TP4057 lithium battery charge management chip
- **RT9080 (https://dfimg.dfrobot.com/5d57611a3416442fa39bffca/wiki/d310c343a276135955547d238c122064.pdf)**: 3.3V Low power LDO

Pin Diagram



Pin Definition

- Power: Power pin
  - VIN: 5V DC
  - 3V3: 3.3V stable output
- GND: common ground pin
- GPIO: ESP32 default GPIO number
- Arduino: Beetle ESP32-C6 GPIO mapping in Arduino
- ADC: ESP32 default analog-to-digital conversion pin
- I2C: I2C interface
  - Beetle ESP32-C6 I2C mapping in Arduino
  - LP_SDA/SCL: Low power I2C pin
- UART: UART interface
  - LP_TX/RX: Low power UART pin
- SPI: Beetle ESP32-C6 SPI mapping in Arduino
- SDIO: ESP32 default SDIO pin
- JTAG: debug interface

Tutorial - First Time Use

Arduino IDE Configuration

Please pay attention to the followings when using FireBeetle 2 ESP32-C6 for the first time.

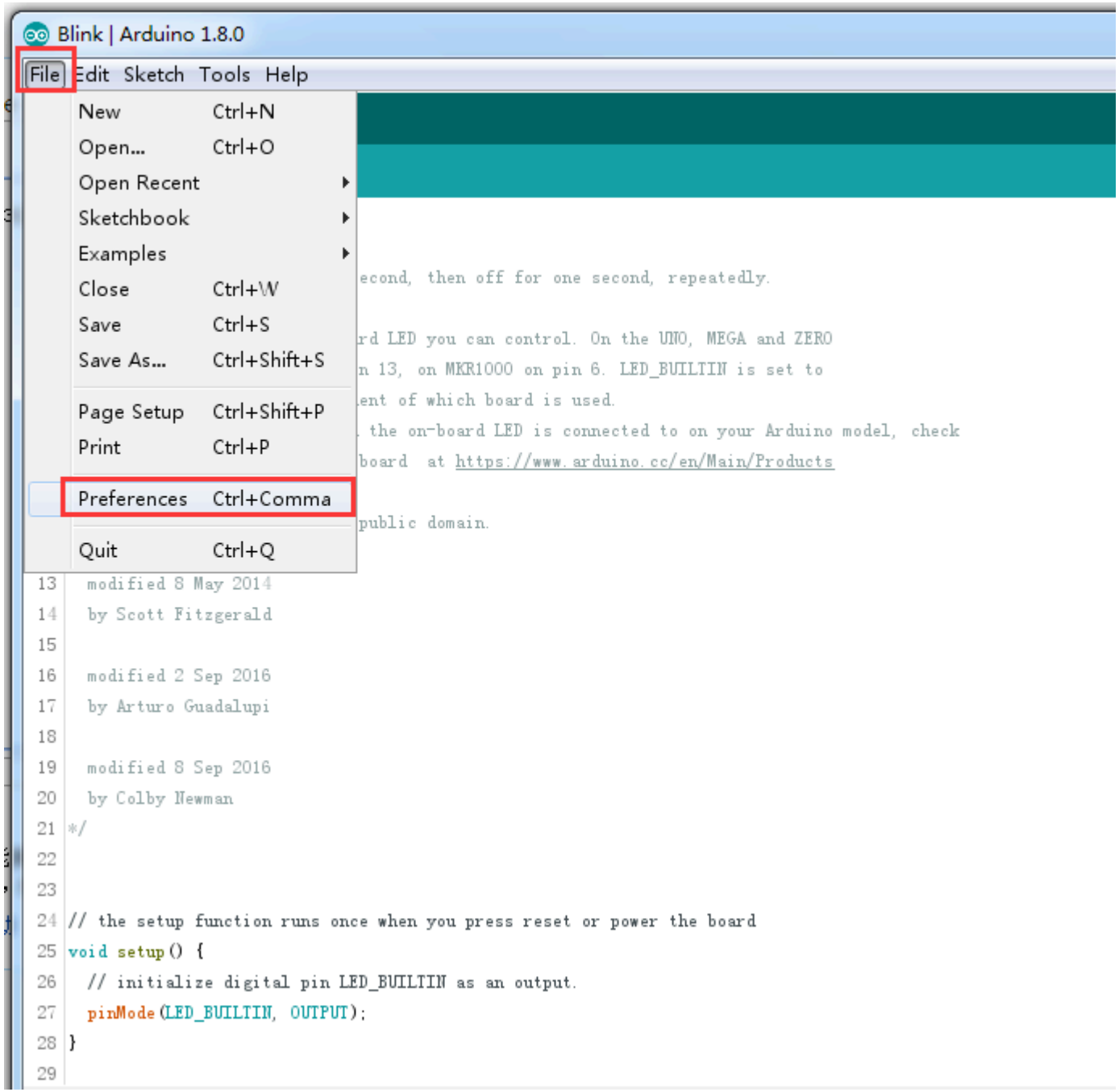
1. Add the json link in the IDE

2. Download the core of the MCU
3. Select the development board and serial port
4. Open the sample code and burn it into the board
5. Get to know the serial monitor

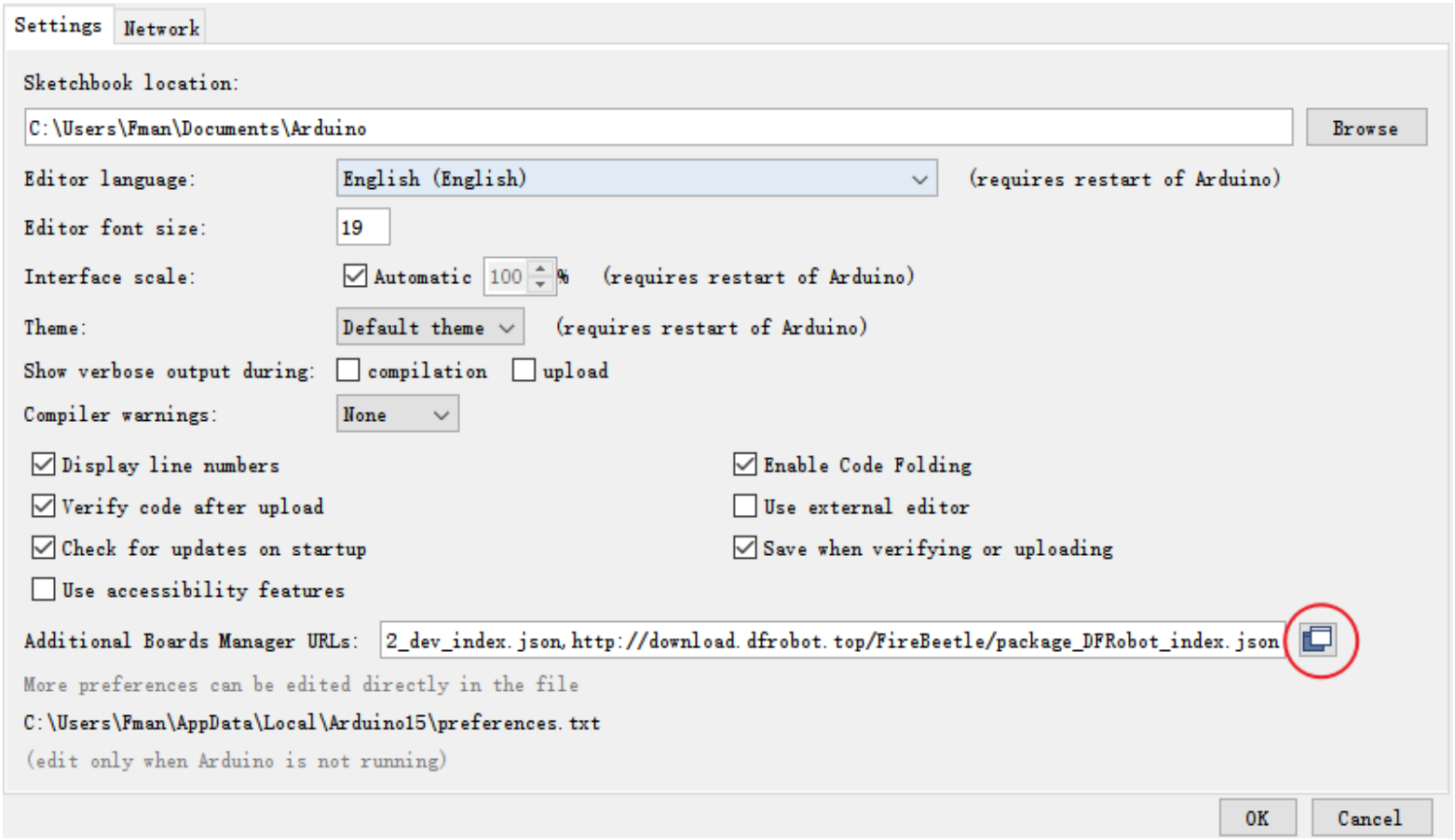
Arduino IDE compiler environment config

- Configure URL to the Arduino IDE

1. Open Arduino IDE and click File->Preferences, as shown below.



2. In the newly opened interface, click the button in the red circle as shown below



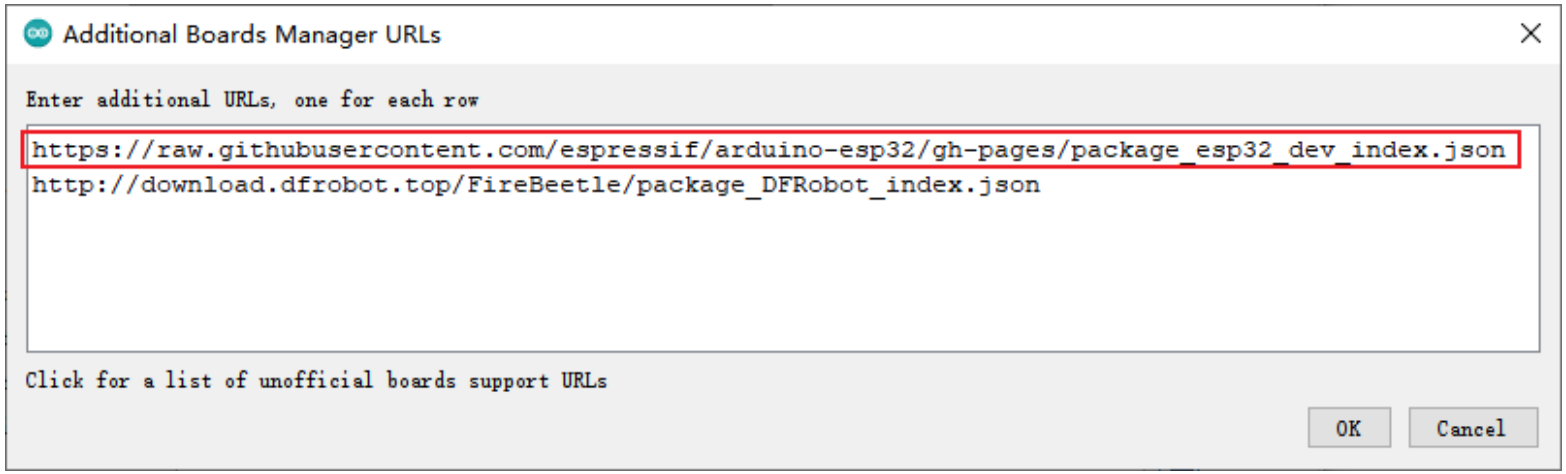
3. Copy the following link into the new pop-up dialog box:

Stable version：https://espressif.github.io/arduino-esp32/package_esp32_index.json (https://espressif.github.io/arduino-esp32/package_esp32_index.json)

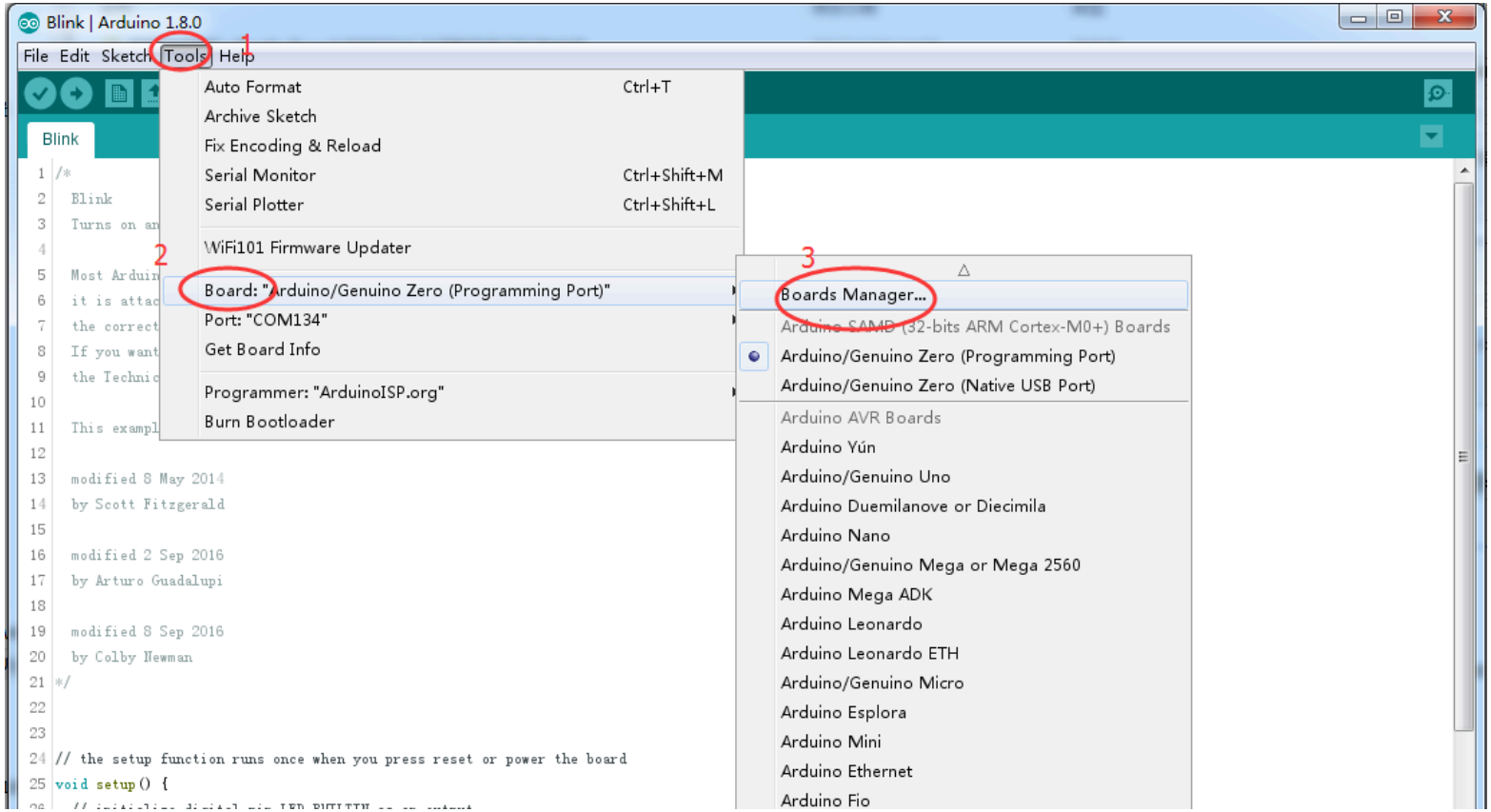
Development release：https://espressif.github.io/arduino-esp32/package_esp32_dev_index.json (https://espressif.github.io/arduino-esp32/package_esp32_dev_index.json)

Note:

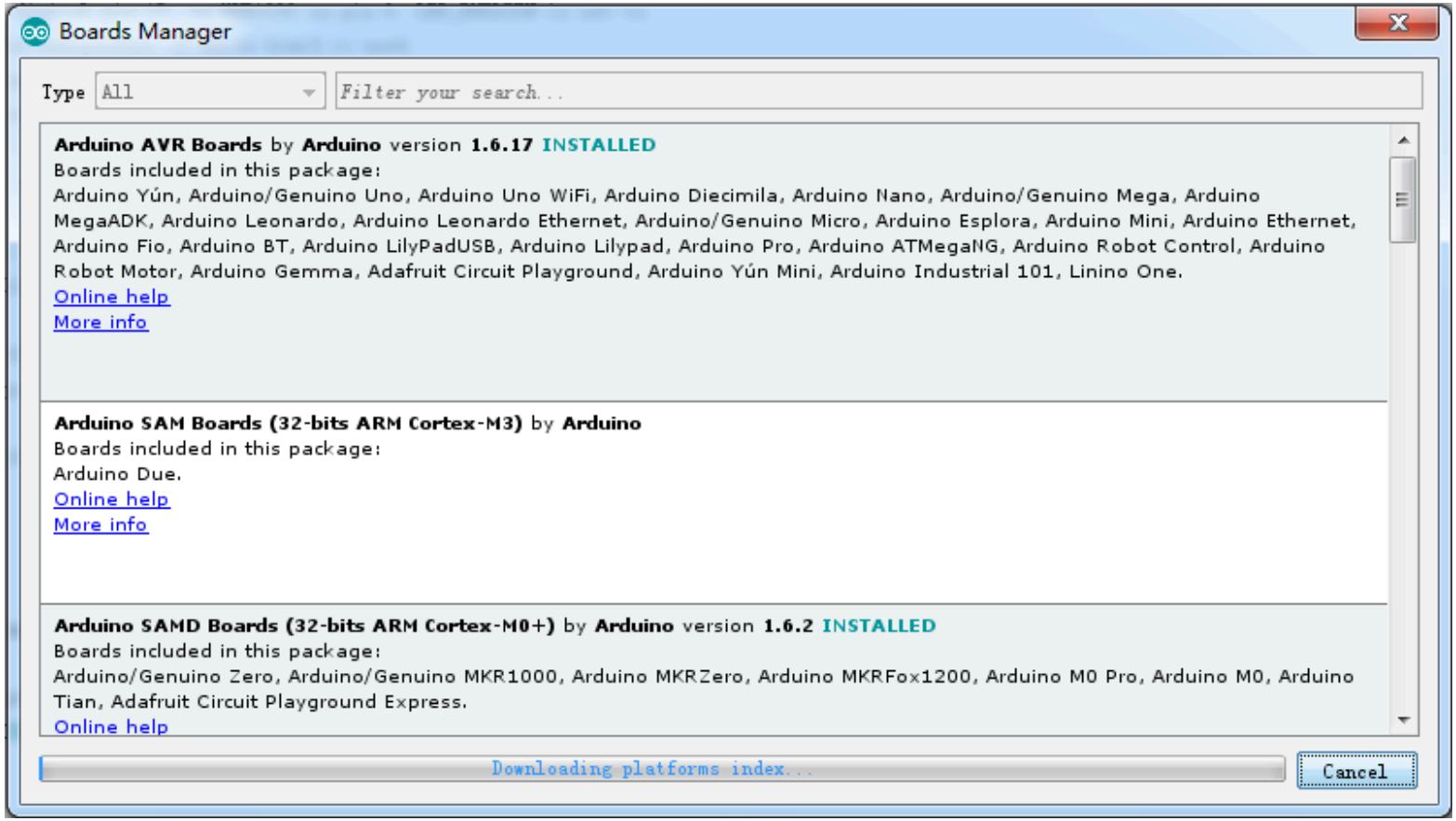
- Please choose the appropriate version according to Chip Support Situation (https://github.com/espressif/arduino-esp32#supported-chips).
- If you have installed another environment before, you can press Enter key at the beginning or end of the previous link and paste the link at a new line.



4. Click OK. Update the board. Open Tools->Board:->Boards Manager... as shown below：

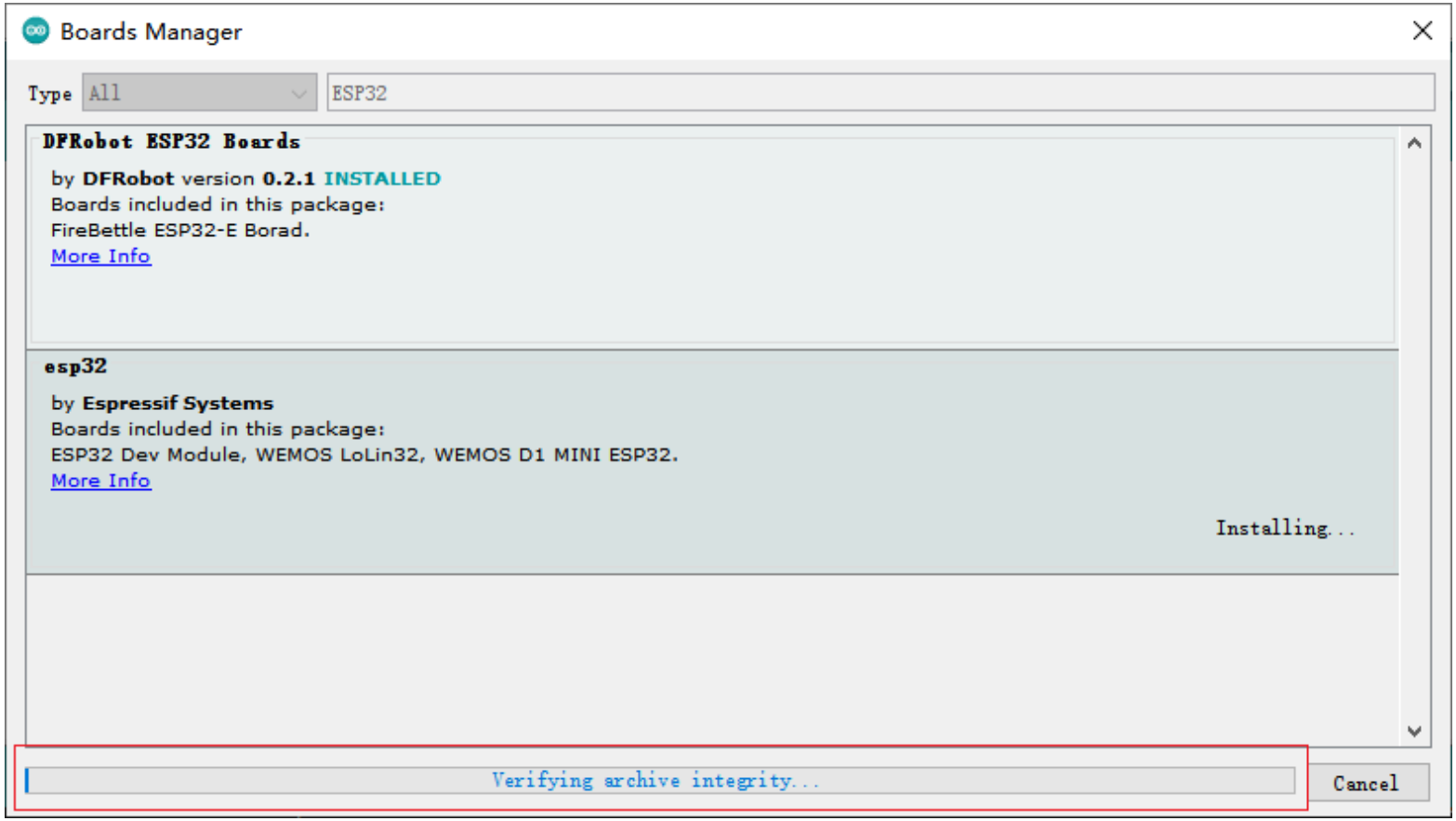


5. Boards Manager will automatically update the boards as shown below:

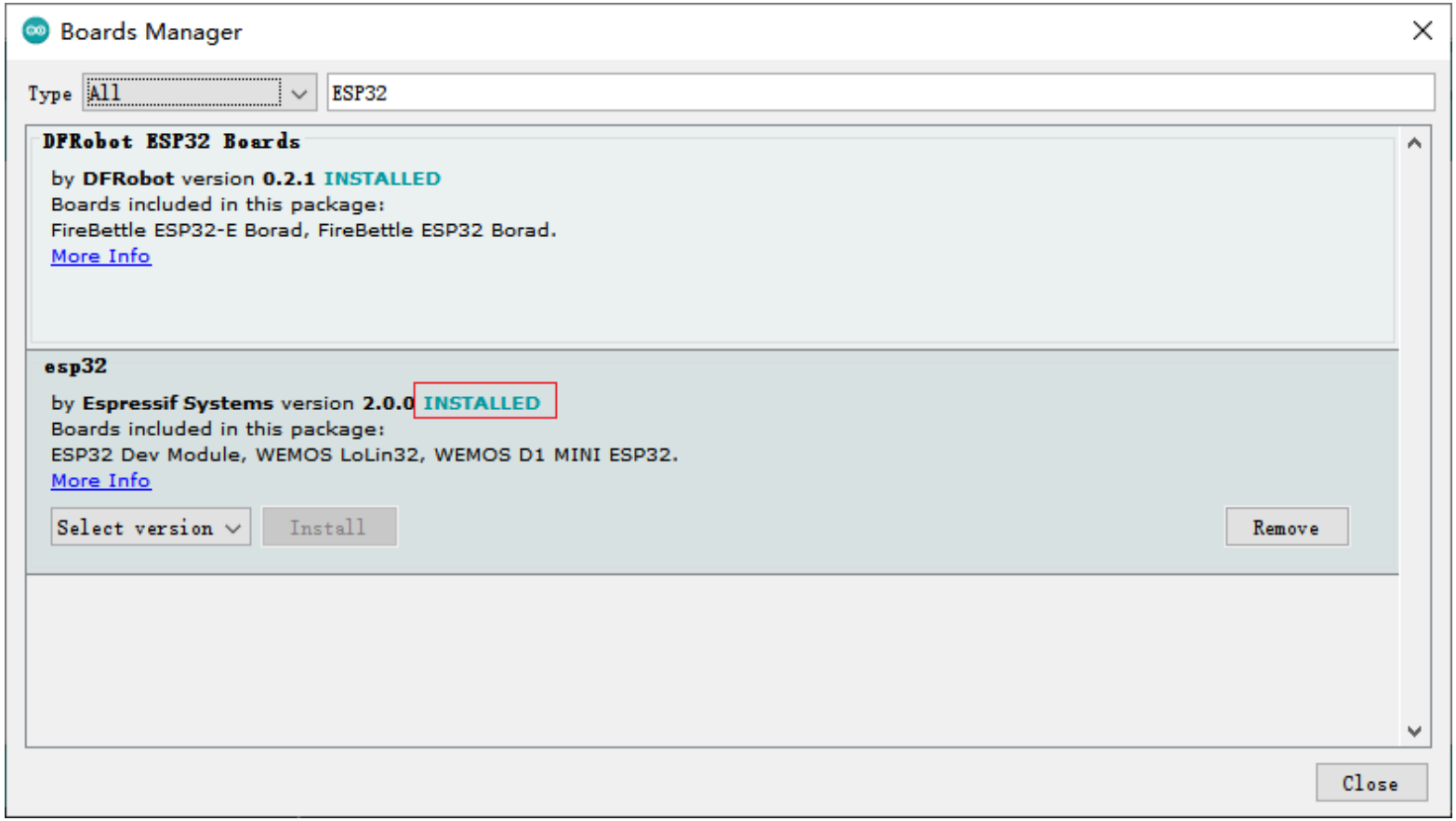


6. After completing the update, you can enter esp32 at the top, select esp32 and click install when the following occurs (It's recommended to install the latest version):

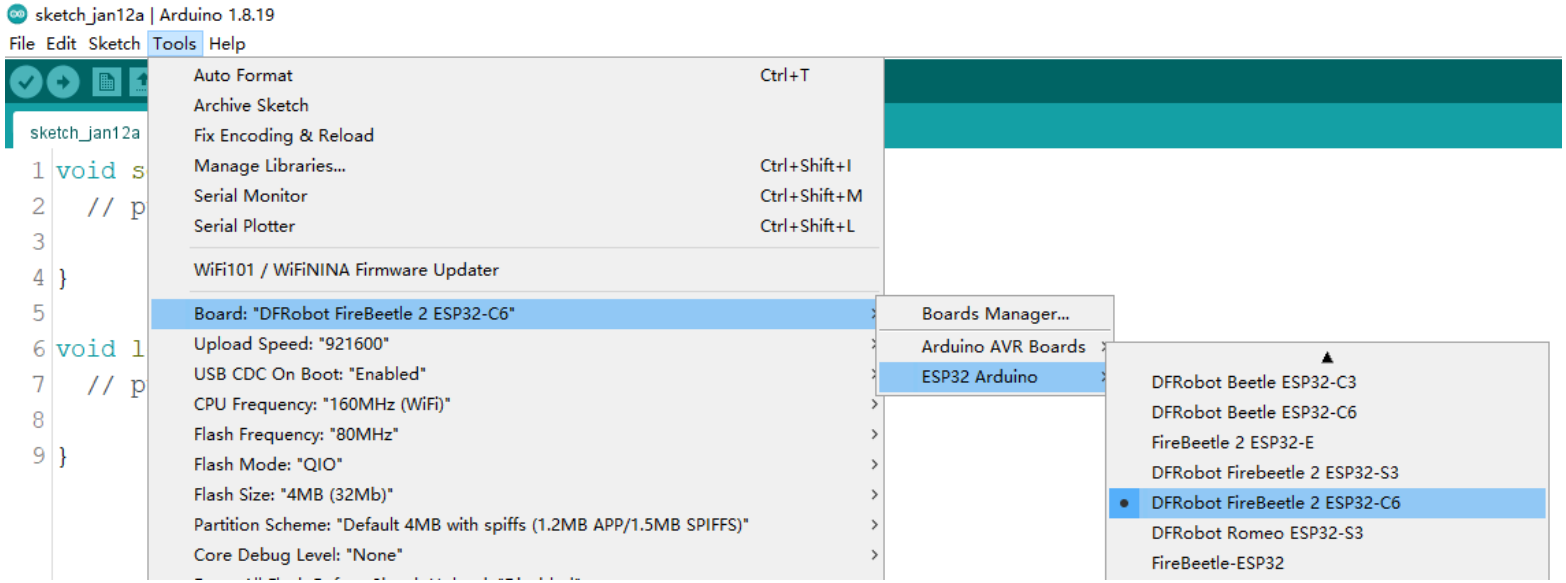
7. Wait for the end of the following progress bar:



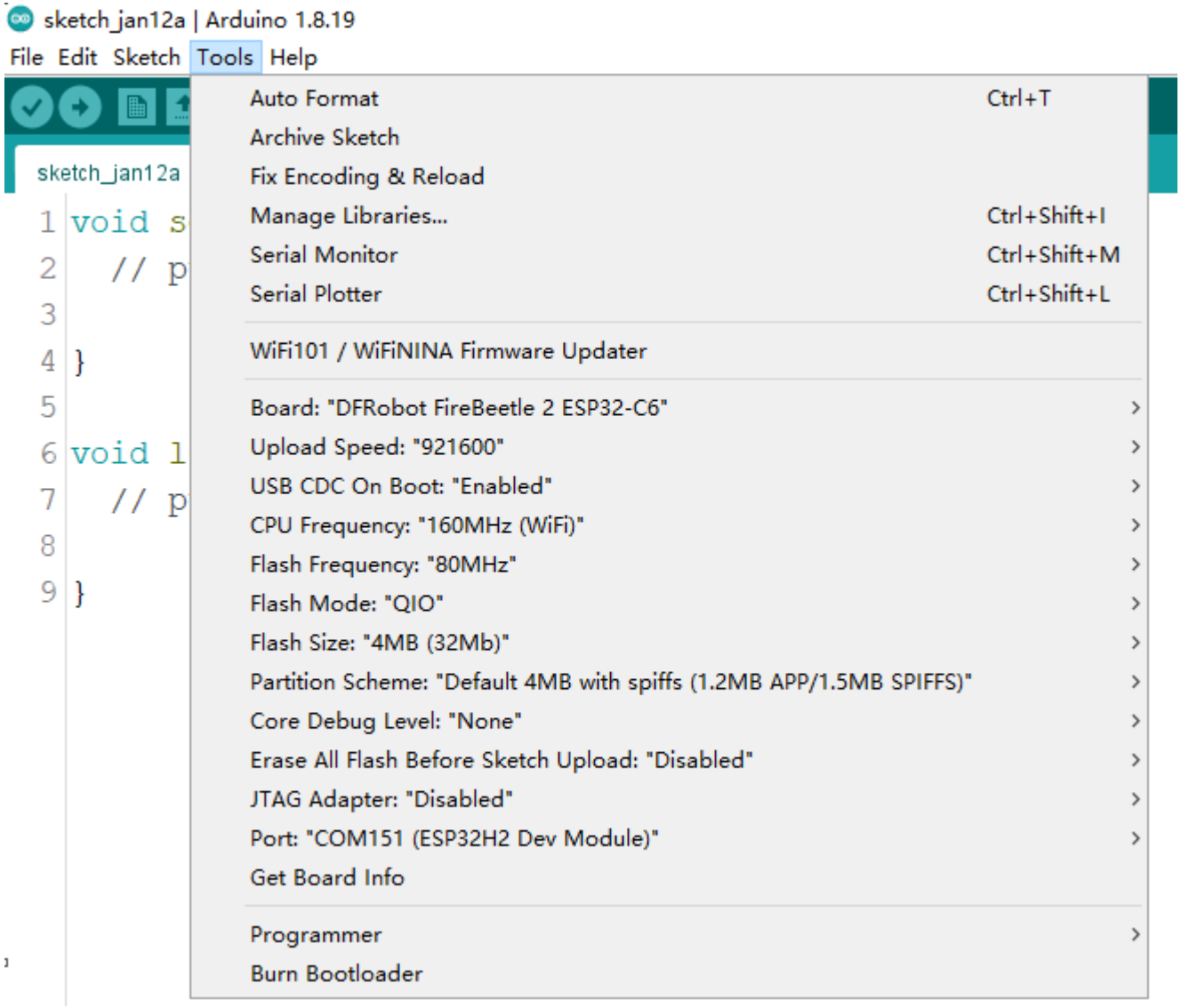
8. After completing the installation, the list will show that the esp32 has been installed, as shown below:



9. Click Tools->Board, select DFRobot FireBeetle 2 ESP32-C6.



10. Before starting, you need to configure the following settings (when you select Disabled, the serial port is RX(17), TX(16), if you need to print on the Arduino monitor via USB, you need to select Enable)



11. Click Port to select the corresponding serial port.

LED Blinking

The default pin for the onboard LED is pin 15.

Sample Code

```
int led = 15;
void setup() {
  pinMode(led,OUTPUT);
}

void loop() {
  digitalWrite(led,HIGH);
  delay(1000);
  digitalWrite(led,LOW);
  delay(1000);
}
```

- Copy the codes above to the code editing box.
- Click the arrow to complile the program and burn it into your development board.

Burning Successful

mage above shows that your codes have been successfully loaded into the board. Then, the onboard LED will start blinking.

- Burning failed? Click here. (https://wiki.dfrobot.com/SKU_DFR1075_FireBeetle_2_Board_ESP32_C6#target_9)

Basic Tutorial

The basic tutorial (https://wiki.dfrobot.com/SKU_DFR1075_FireBeetle_2_Board_ESP32_C6_Basic_Tutorial) includes the use of Battery voltage detection, PWM, interrupt, serial port, servo, and SD card.

Advanced Tutorial

The advanced tutorial (https://wiki.dfrobot.com/SKU_DFR1075_FireBeetle_2_Board_ESP32_C6_Advanced_Tutorial) domenstrates how to use screen, Bluetooth, WiFi, ESP-NOW, one-key for networking config and sample projects.

FAQ

1. What will cause burning error?

- There is no delay or too short delay in Loop.

```
A fatal error occurred: Timed out waiting for packet header
A fatal error occurred: Timed out waiting for packet header
```

- The USB cannot be recognized by the PC as some functions are incorrectlly called.

How to solve

- Press and hold BOOT, click RST, and then release the BOOT button to burn.

Principle During the initialization process, ESP32 undertakes a verification of the voltage level on the BOOT (IO9) pin. If the voltage level is determined to be high, the system proceeds with a normal startup. In contrast, if the voltage level is deemed to be low, the device enters into the programming mode. By default,

the BOOT pin maintains a high voltage level, but it transitions to a low level when a button is pressed.

2. Data cannot be printed on serial port

- Check if the USB CDC is enabled
- Check print information using other serial debugger.

For any questions, advice or cool ideas to share, please visit the **DFRobot Forum** (https://www.dfrobot.com/forum/).

More Documents

- ESP32-C6 Chip Datasheet (https://dfimg.dfrobot.com/60c1e008bddfc41c3293de80/wiki/bd5febb16dd4b86eb34cb387815080ae.pdf)
- DFR1117-Schematics.pdf (https://dfimg.dfrobot.com/60c1e008bddfc41c3293de80/wiki/68dbe234c15e4e218d4c9b54a674ea61.pdf)
- DFR1117-Dimension.pdf (https://dfimg.dfrobot.com/60c1e008bddfc41c3293de80/wiki/b6ccaa7b1c039a85403ae8ec6a33c98a.png)
- RT9080 Chip Datasheet (https://dfimg.dfrobot.com/5d57611a3416442fa39bffca/wiki/d310c343a276135955547d238c122064.pdf)

Get **Beetle ESP32-C6** (https://www.dfrobot.com/product-2778.html) from DFRobot Store or **DFRobot Distributor**. (https://www.dfrobot.com/distributor)

Turn to the Top (https://wiki.dfrobot.com/SKU_DFR1117_Beetle_ESP32_C6#target_0)