

ESP32-C3-Zero

From Waveshare Wiki

Jump to: navigation, search

Overview

Introduction

ESP32-C3-Zero (without pin header) and ESP32-C3-Zero-M (with pin header) are tiny in size with castellated holes, making them easy to integrate into other host boards.

ESP32-C3-Zero comes with an onboard Type-C USB connector, which exposes most of the unused pins in a small form factor. It adopts ESP32-C3FH4 as a System-on-Chip (SoC) that integrates low-power WiFi and BLE5 with 4MB Flash. In addition, there is a hardware encryption accelerator, RNG, HMAC, and Digital

Signature modules to meet the safety requirements of IoT and provide peripheral interfaces. Moreover, its low-power working mode supports most application scenarios such as IoT, mobile devices, wearable electronic devices, and smart homes.

ESP32-C3-Zero



(<https://www.waveshare.com/esp32-c3-zero.htm>)

ESP32-C3FH4
Type C USB

Features

- Low-power SoC equipped with RISC-V 32-bit single-core processor, up to 160MHz main frequency.
- Supports 2.4GHz Wi-Fi (802.11 b/g/n) and Bluetooth® 5 (LE).
- 400KB of SRAM and 384KB ROM, and 4MB of onboard Flash memory.
- Castellated module and onboard ceramic antenna, allow soldering direct to carrier boards.
- Supports flexible clock, module power supply independent setting, and other controls to realize low power consumption in different scenarios.
- Integrated with USB serial port full-speed controller, 15 × GPIO pins allow flexible configuring pin functions.

- 3 × SPI, 1 × I2C, 2 × UART, 1 × I2S, 2 × ADC, etc.

Hardware Description

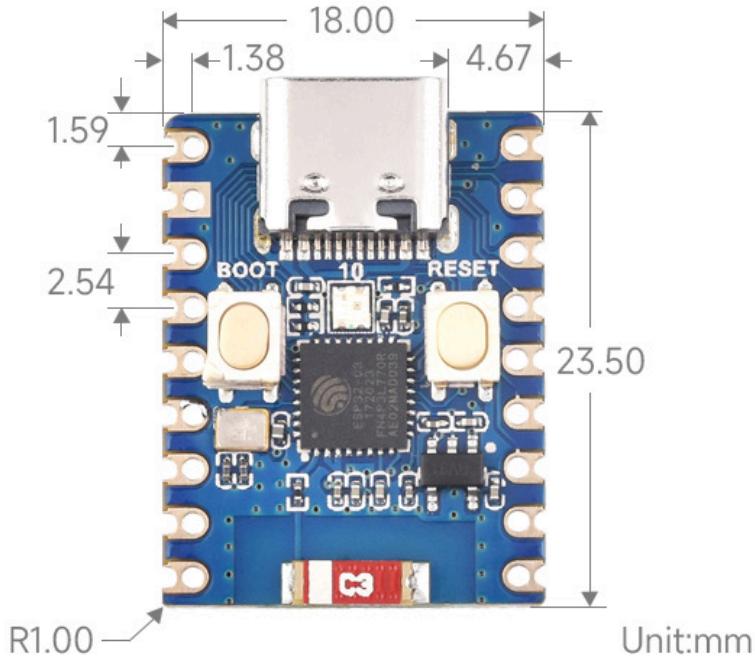
- When using ESP32-C3-Zero with daughterboards, please avoid covering the ceramic antenna with PCB boards, metal, or plastic components.
- In ESP32-C3-Zero, GPIO12 to GPIO17 pins are not exposed; these pins are used for stacking the 4MB Flash.
- ESP32-C3-Zero uses GPIO10 to connect with WS2812 RGB LED. Please refer to the documentation link ([https://files.waveshare.com/wiki/ESP32-C3-Zero/XL-0807RGBC-WS2812B%20\(1\).pdf](https://files.waveshare.com/wiki/ESP32-C3-Zero/XL-0807RGBC-WS2812B%20(1).pdf)) for WS2812 specifications.
- ESP32-C3-Zero does not employ a USB to UART chip. When flashing firmware, press and hold the BOOT button (GPIO9) and then connect the Type-C cable.
- The "TX" and "RX" markings on the board indicate the default UART0 pins for ESP32-C3-Zero. Specifically, TX is GPIO21, and RX is GPIO20.

Hardware Connection

- Press the BOOT (GPIO9) key before connecting the Type-C cable each time download the firmware.

- Input 3.7V~6V for the castellated hole with 5V silkscreen when connecting the external power.

Dimensions



(/wiki/File:ESP32-C3-Zero_02.jpg)

Environment Setting

The software framework for ESP32 series development boards is completed, and you can use CircuitPython, MicroPython, and C/C++ (Arduino, ESP-IDF) for rapid prototyping of product development. Here's a brief introduction to these three development approaches:

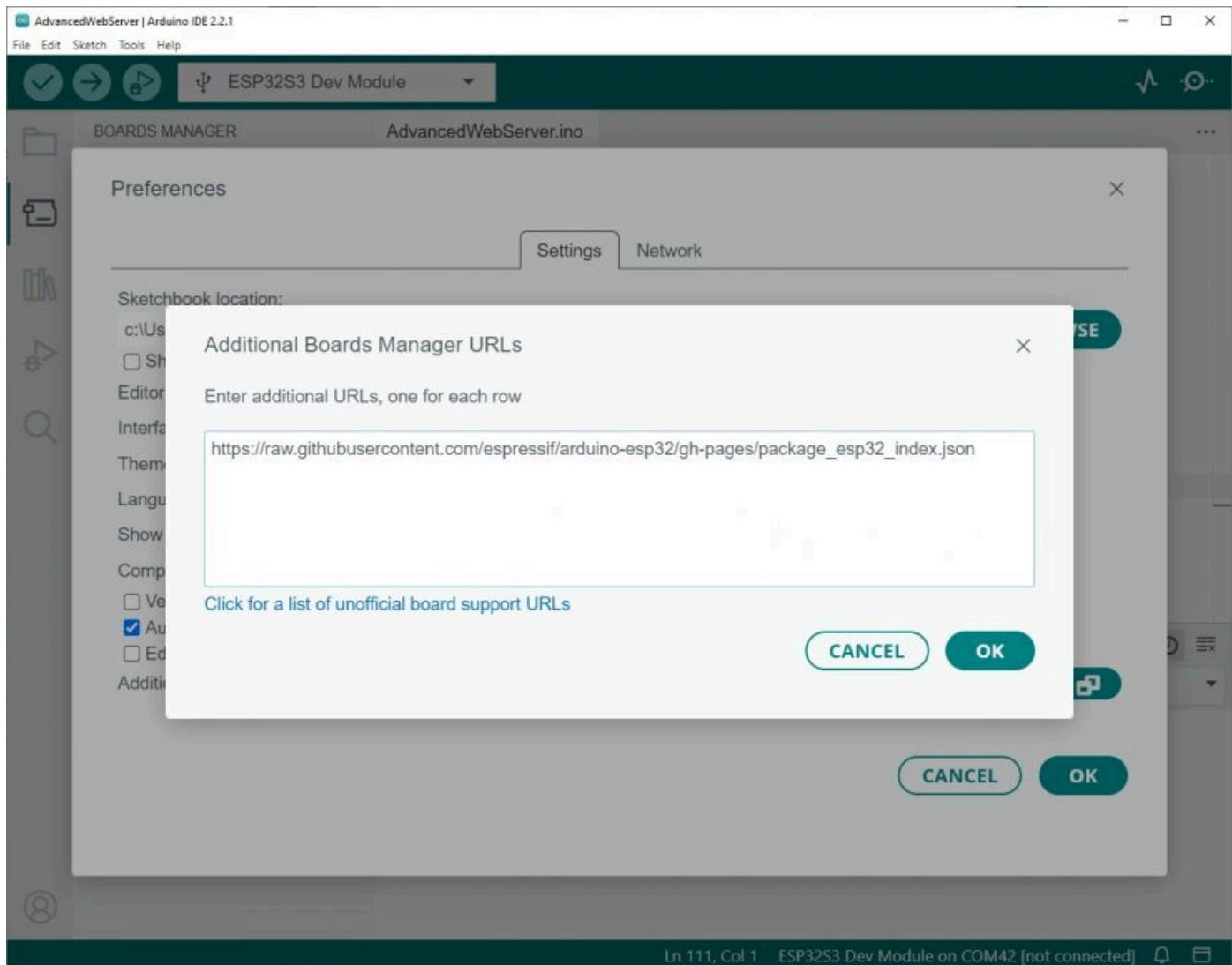
- CircuitPython is a programming language designed to simplify coding tests and learning on low-cost microcontroller boards. It is an open-source derivative of the MicroPython programming language, primarily aimed at students and beginners. CircuitPython development and maintenance are supported by Adafruit Industries.
 - You can refer to development documentation (<https://docs.circuitpython.org/en/latest/shared-bindings/index.html>) for CircuitPython-related applications development.
 - The GitHub library (https://github.com/adafruit/Adafruit_CircuitPython_Bundle) for CircuitPython allows for recompilation for custom development.
- MicroPython is an efficient implementation of the Python 3 programming language. It includes a small subset of the Python standard library and has been optimized to run on microcontrollers and resource-constrained environments.

- You can refer to development documentation (<https://docs.micropython.org/en/latest/>) for MicroPython-related application development.
- The GitHub library (<https://github.com/micropython/micropython>) for MicroPython allows for recompilation for custom development.
- The official libraries and support from Espressif Systems for C/C++ development make it convenient for rapid installation. For Windows 10 system:
 - Users can select Arduino (<https://docs.espressif.com/projects/arduino-esp32/en/latest/installing.html>)
 - Or Visual Studio Code (ESP-IDF (<https://docs.espressif.com/projects/esp-idf/en/stable/esp32s2/get-started/index.html>)) as their Integrated Development Environment (IDE).
- Mac and Linux operating system users can refer to the official instructions (<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/index.html>). Additionally, you can find development manuals for ESP32 series Arduino and ESP-IDF development to assist you in the process.

Arduino

1. Download and install Arduino IDE (<https://www.arduino.cc/en/software>).
2. Install ESP32 on the Arduino IDE as shown below, and you can refer to this link (<https://docs.espressif.com/projects/arduino-esp32/en/latest/installing.html>).
 - Fill in the following link in the Additional Boards Manager URLs section of the Settings interface under File -> Preferences and save.

```
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
```



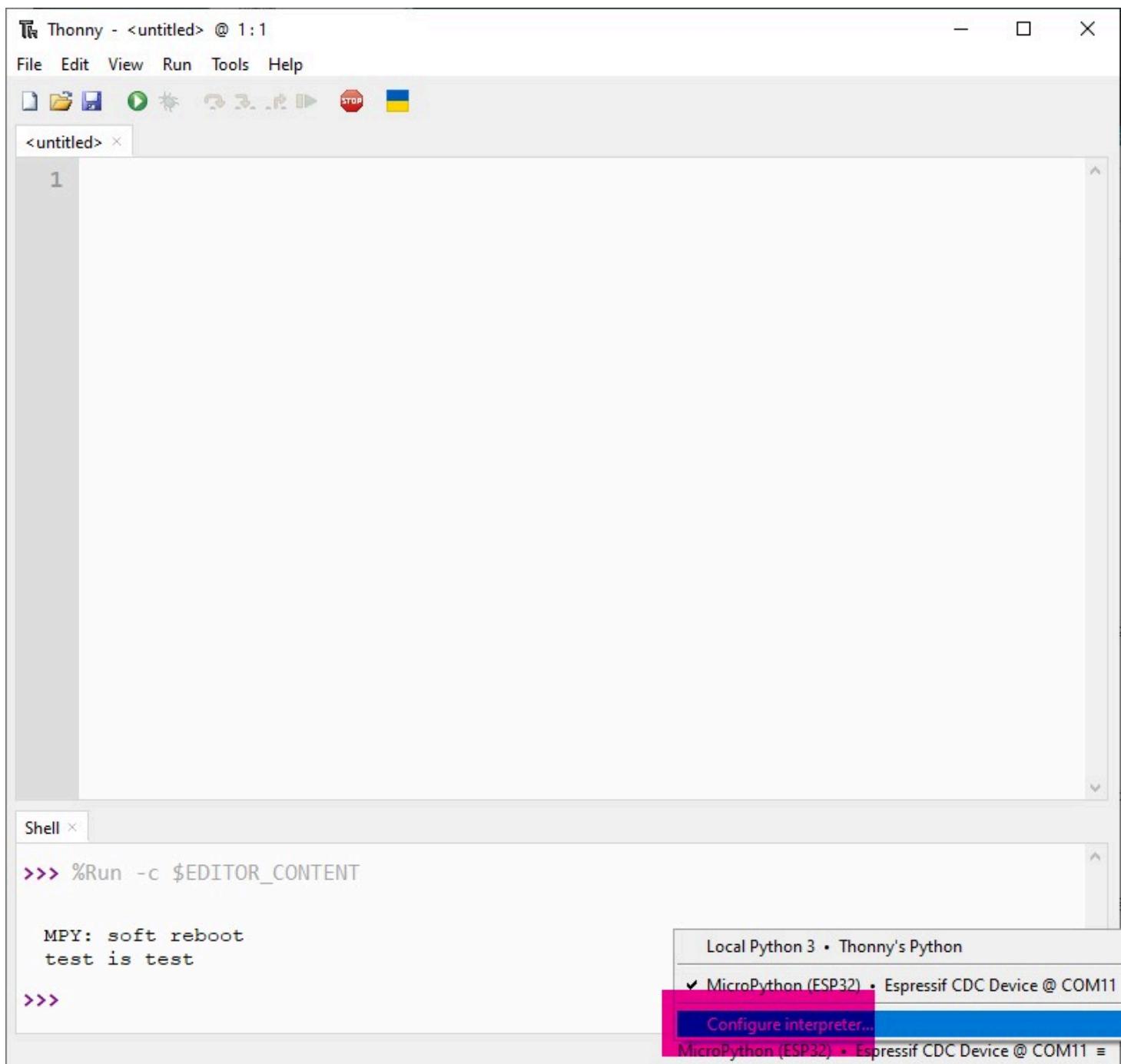
(/wiki/File:ESP32-C3-Zero_-05.jpg)

- Search esp32 on Board Manager to install, and restart Arduino IDE to take effect.
3. ESP32-C3-Zero does not have USB to UART chip mounted, you need to use the USB of ESP32-C3 as the download interface and Log print interface, and you need to enable USB CDC when using Arduino IDE.

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: "ESP32S3 Dev Module"	▶
Port: "COM42"	▶
Get Board Info	
USB CDC On Boot: "Enabled"	▶
CPU Frequency: "240MHz (WiFi)"	▶
Core Debug Level: "None"	▶
USB DFU On Boot: "Disabled"	▶ (/wiki/File:ESP32-C3-Zero_-06.jpg)
Erase All Flash Before Sketch Upload: "Disabled"	▶
Events Run On: "Core 1"	▶
Flash Mode: "QIO 80MHz"	▶
Flash Size: "4MB (32Mb)"	▶
JTAG Adapter: "Disabled"	▶
Arduino Runs On: "Core 1"	▶
USB Firmware MSC On Boot: "Disabled"	▶
Partition Scheme: "Huge APP (3MB No OTA/1MB SPIFFS)"	▶
PSRAM: "QSPI PSRAM"	▶
Upload Mode: "UART0 / Hardware CDC"	▶
Upload Speed: "921600"	▶
USB Mode: "Hardware CDC and JTAG"	▶
Programmer	▶
Burn Bootloader	

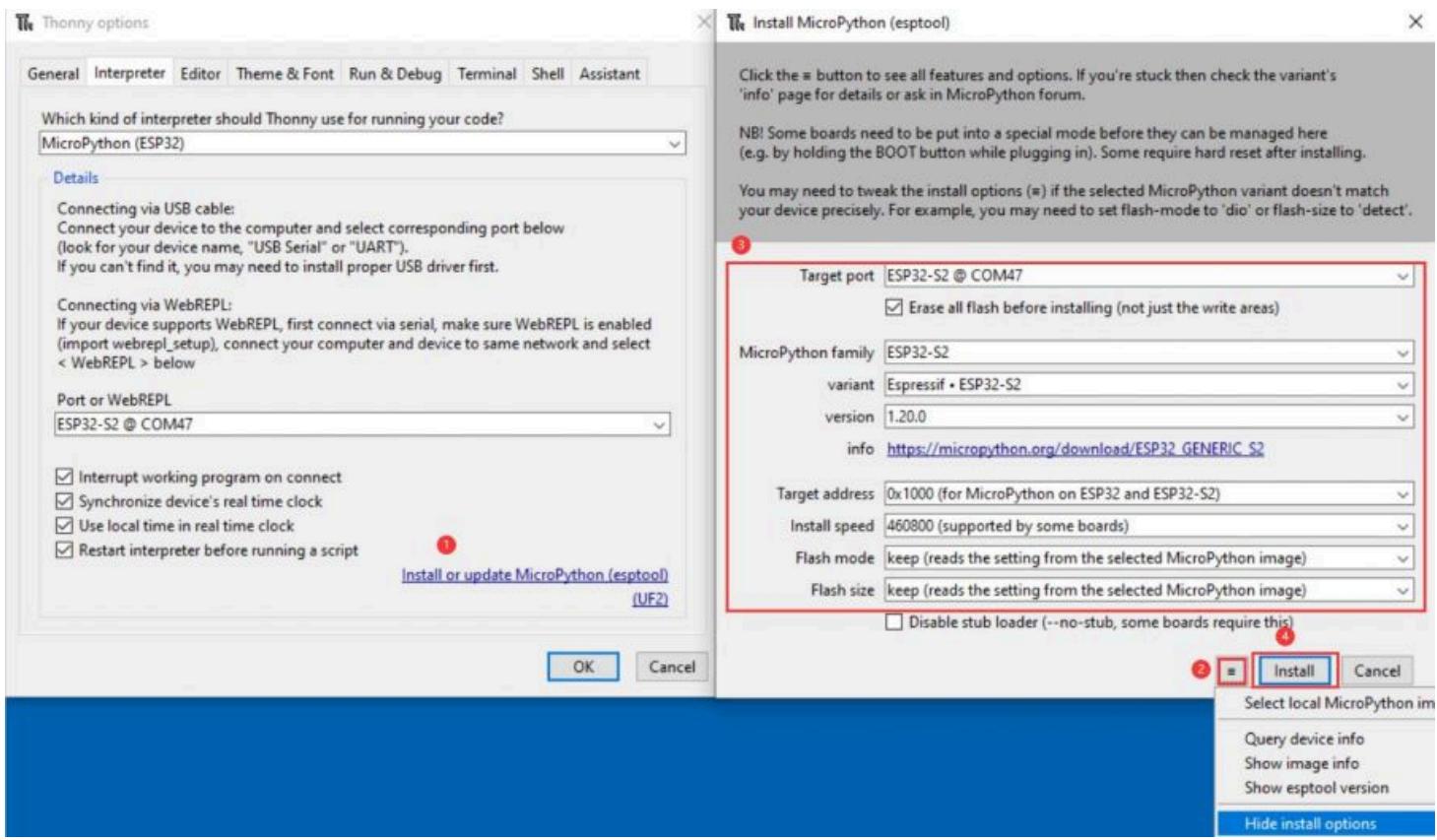
MicroPython

1. Download and install the latest Thonny (<https://thonny.org/>), open Thonny IDE -> Configure interpreter..., as shown below:



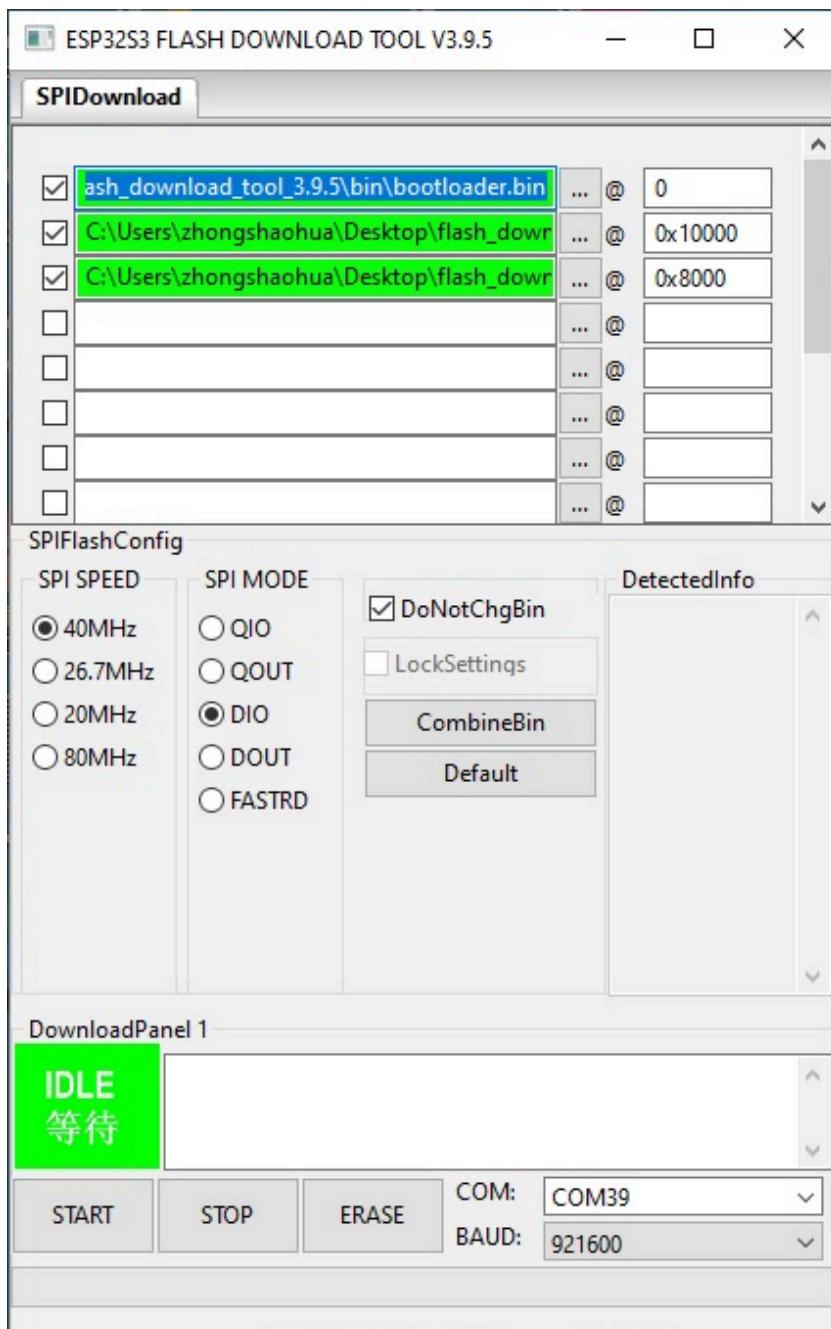
(/wiki/File:CircuitPython_Thonny06.jpg)

2. Press the BOOT key on the board, connect it to the USB cable, find the device manager or the corresponding COM port, and download or run the demo. For more details, you can see #Hardware Connection.
3. Select the ESP32 series online MPY firmware to download according to the following steps. The Flash content of the development board will be cleared before downloading, and the whole download process lasts about 1 minute.



(/wiki/File:CircuitPython_Thonny03.jpg)

4. Download the local firmware using `flash_download_tool` according to the following figure. You can choose either step 3 or step 4, but we recommend to choose step 4.



5. Please refer to MicroPython Documentation (<https://github.com/micropython/micropython/releases/tag/v1.18>) and Release Note (<https://github.com/micropython/micropython/releases/tag/v1.18>) to program.

Sample Demo

- Please refer to arduino-esp32 (<https://github.com/espressif/arduino-esp32>) for Arduino sample demo or File -> examples on the Arduino IDE.

- Please refer to MicroPython (<https://docs.micropython.org/en/latest/>) development documentation and sample demo for mpy example.

Resource

Document

- Schematic (<https://files.waveshare.com/wiki/ESP32-C3-Zero/ESP32-C3-Zero-Sch-V2.pdf>)
- MicroPython development documentation (<https://docs.micropython.org/en/latest/>)
- ESP32 Arduino Core's documentation (<https://docs.espressif.com/projects/arduino-esp32/en/latest/index.html>)
- arduino-esp32 (<https://github.com/espressif/arduino-esp32>)
- ESP-IDF (<https://github.com/espressif/esp-idf>)
- esp32-c3 programming guide (<https://docs.espressif.com/projects/esp-idf/en/latest/esp32c3/api-reference/api-conventions.html>)

Demo

- Sample demo (<https://files.waveshare.com/wiki/ESP32-C3-Zero/Esp32-c3-zero-code.zip>)

Software

- Sscom serial port assistant (<https://files.waveshare.com/upload/b/b3/Sscom5.13.1.zip>)
- Thonny Python IDE (<https://thonny.org/>)
- Arduino IDE (<https://www.arduino.cc/en/software>)

Datasheet

- ESP32-C3 Datasheet (https://files.waveshare.com/wiki/ESP32-C3-Zero/esp32-c3_datasheet_en.pdf)
- WS2812B (<https://files.waveshare.com/wiki/ESP32-C3-Zero/XL-0807RGBC-WS2812B.pdf>)

FAQ

Question: Can the ESP32-C3-Zero be powered via pins?

Answer:

You can use 5V pins and then supply it with 5V.

Question:How to flash ESP32-C3-Zero with hardware debugging, is it just plugging in the type C port and connecting it to the computer?

Answer:

That's right. If the flashing fails, press and hold the boot button to connect to the computer and then release it.

Question:Does ESP32-C3-Zero support ipv6?

Answer:

ESP32-C3 has ipv6 sample demo, it can support ipv6.

Support

Technical Support

If you need technical support or have any feedback/review, please click the **Submit Now** button to submit a ticket, Our support team will check and reply to you within 1 to 2 working days. Please be patient as we make every effort to help you to resolve the issue.

Working Time: 9 AM - 6 PM GMT+8
(Monday to Friday)

[Submit Now \(https://service.waveshare.com/\)](https://service.waveshare.com/)