ESP32-C6-DEV-KIT-N8

From Waveshare Wiki
Jump to: navigation, search

Overview

Introduction

ESP32-C6-DEV-KIT-N8

(https://www.waveshare.com/esp32-c6-dev-kit-n8.htm?sku=25723)

The ESP32-C6-DEV-KIT-N8 is a compact microcontroller development board with multiple peripheral interfaces.
In terms of hardware, it adopts the ESP32-C6-WROOM-1 module with a RISC-V 32-bit single-core processor. It supports up to 160MHz clock frequency with built-in 320KB ROM, 512KB HP SRAM and 16KB LP SRAM. Onboard CH343 UART and CH334 USB HUB chips, it supports USB and UART development at the same time via a USB-C port. Compatible with the pinout of the ESP32-C6-DevKitC-1-N8 development board, it is more convenient to use and expand a variety of peripheral modules.
In terms of software, you can choose the ESP-IDF

development environment or the Arduino IED to develop, so that you can easily and quickly get started and apply it to the product.

ESP32-C6-DEV-KIT-N8 with Pre-soldered Header

(https://www.waveshare.com/esp32-c6-dev-kit-n8.htm?sku=25563)

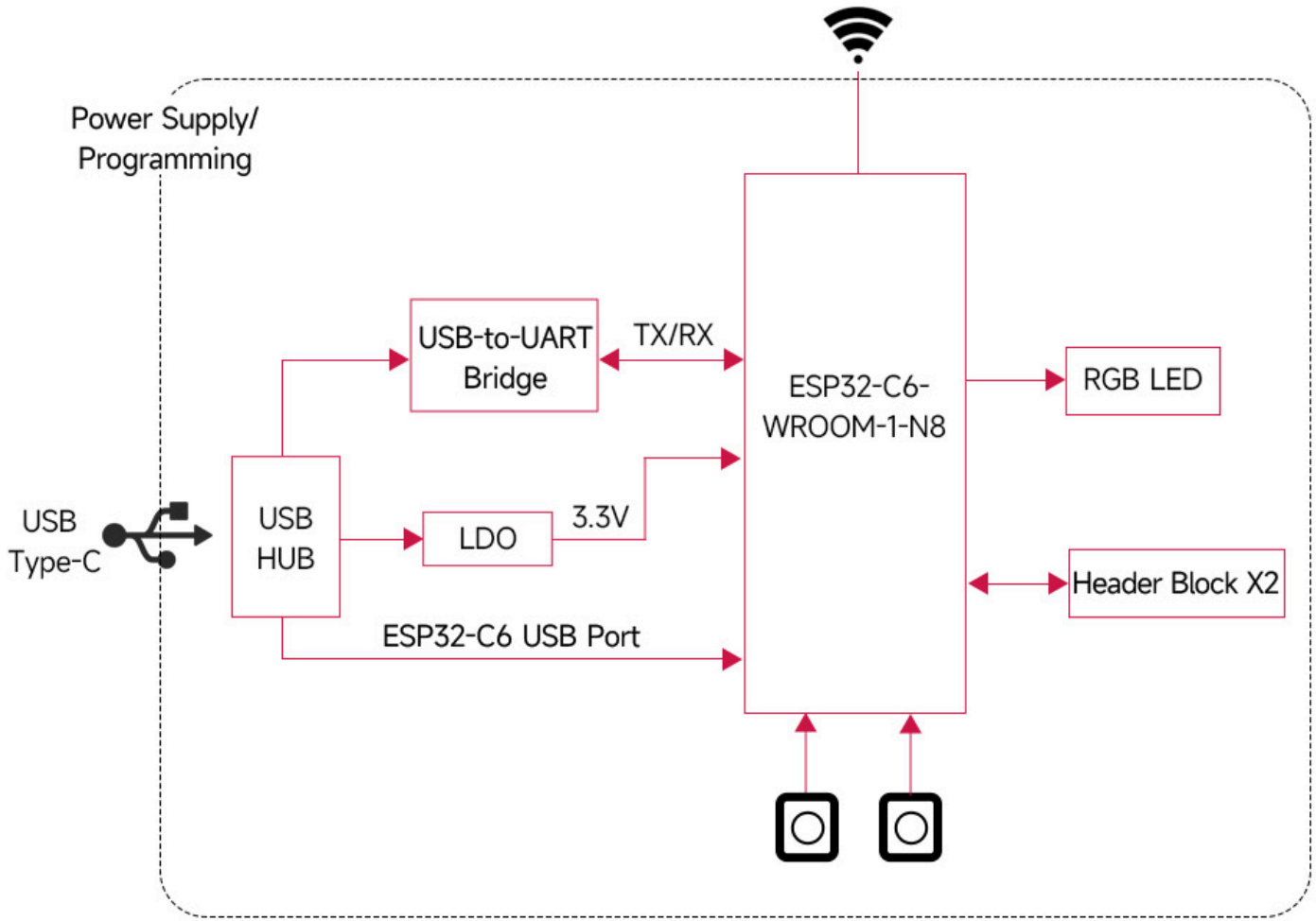
ESP32-C6
USB Type-C

Features

- Adopts ESP32-C6-WROOM-1-N8 module with RISC-V 32-bit single-core processor, up to 160MHz main frequency.
- Integrated 320KB ROM, 512KB HP SRAM, 16KB LP SRAM and 8MB Flash memory.
- Integrated 2.4GHz WiFi 6, Bluetooth Low Energy (Bluetooth LE) and IEEE 802.15.4 (Zigbee 3.0 and Thread) wireless communication, with superior RF performance.
- Type-C connector, easier to use.
- Onboard CH343 and CH334 chips can meet the needs of USB and UART development via a Type-C interface.
- Rich peripheral interfaces, compatible with the pinout of the ESP32-C6-DevKitC-1-N8 development board, offer strong compatibility and expandability.
- The castellated module allows soldering directly to carrier boards.

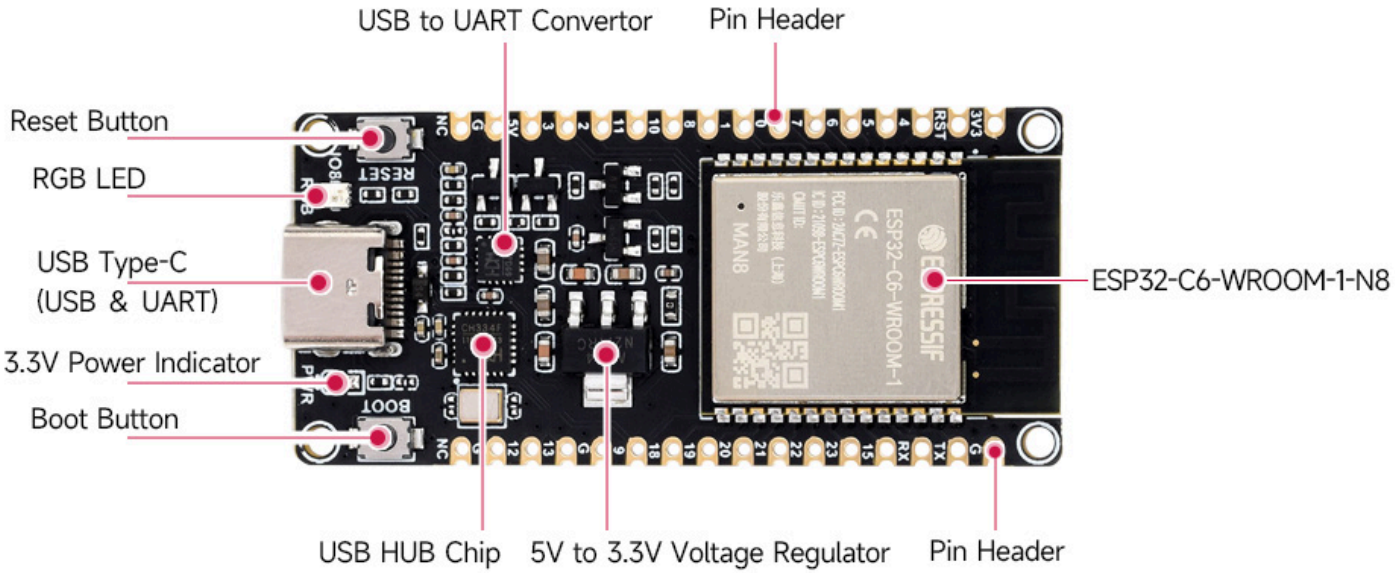
- Supports a variety of low-power operating states, adjustable balance among communication distance, data rate and power consumption, to meet the power requirements of various application scenarios.

Fuction Block Diagram



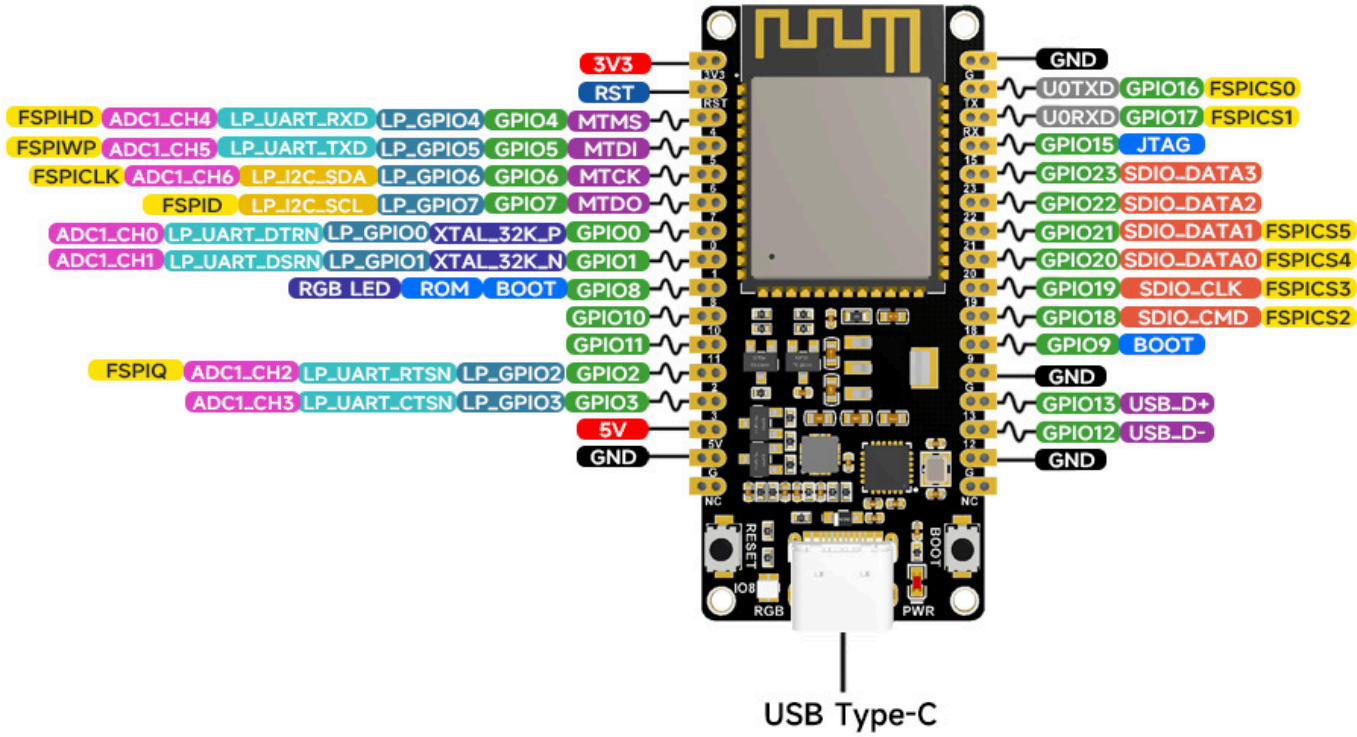
(/wiki/File:ESP32-C6-DEV-KIT-N803.jpg)

Onboard Resources



(/wiki/File:ESP32-C6-DEV-KIT-N804.jpg)

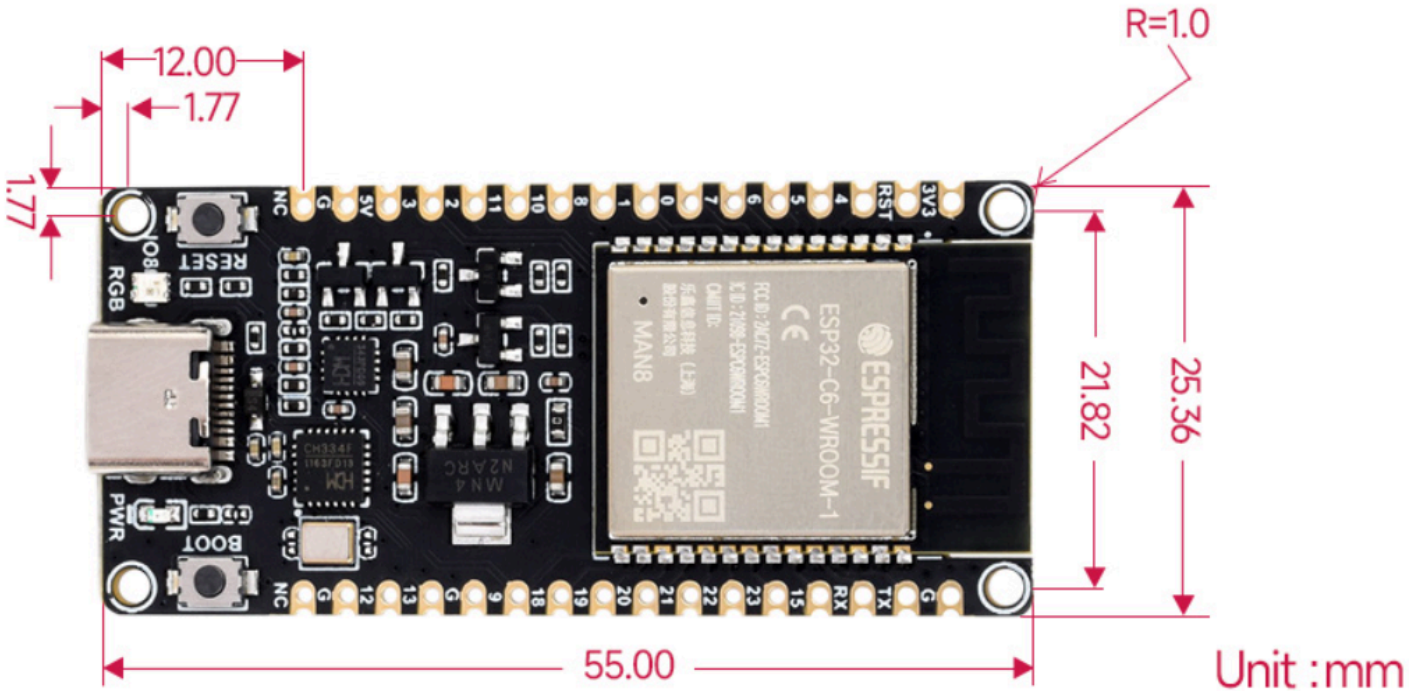
Pinout



PWM Capable Pin		FSPI	Fast SPI Functions
GPIOX	GPIO Input and Output	LP_UART	Low-Power UART Functions
LP_I2C	Low-Power I2C Functions	SDIO	SDIO Functions
OTHER	Other Related Functions	STRAP	Strapping Pin Functions
JTAG/USB	JTAG for Debugging and USB	ADCX_CH	Analog-to-Digital Converter
SERIAL	Serial for Debug/Programming	LP_GPIOX	Low-Power GPIO Functions
PWR	Power Rails (3V3 and 5V)	GND	Ground Plane

(/wiki/File:ESP32-C6-DEV-KIT-N805.jpg)

Dimensions



(/wiki/File:ESP32-C6-DEV-KIT-N806.png)

Working with ESP-IDF

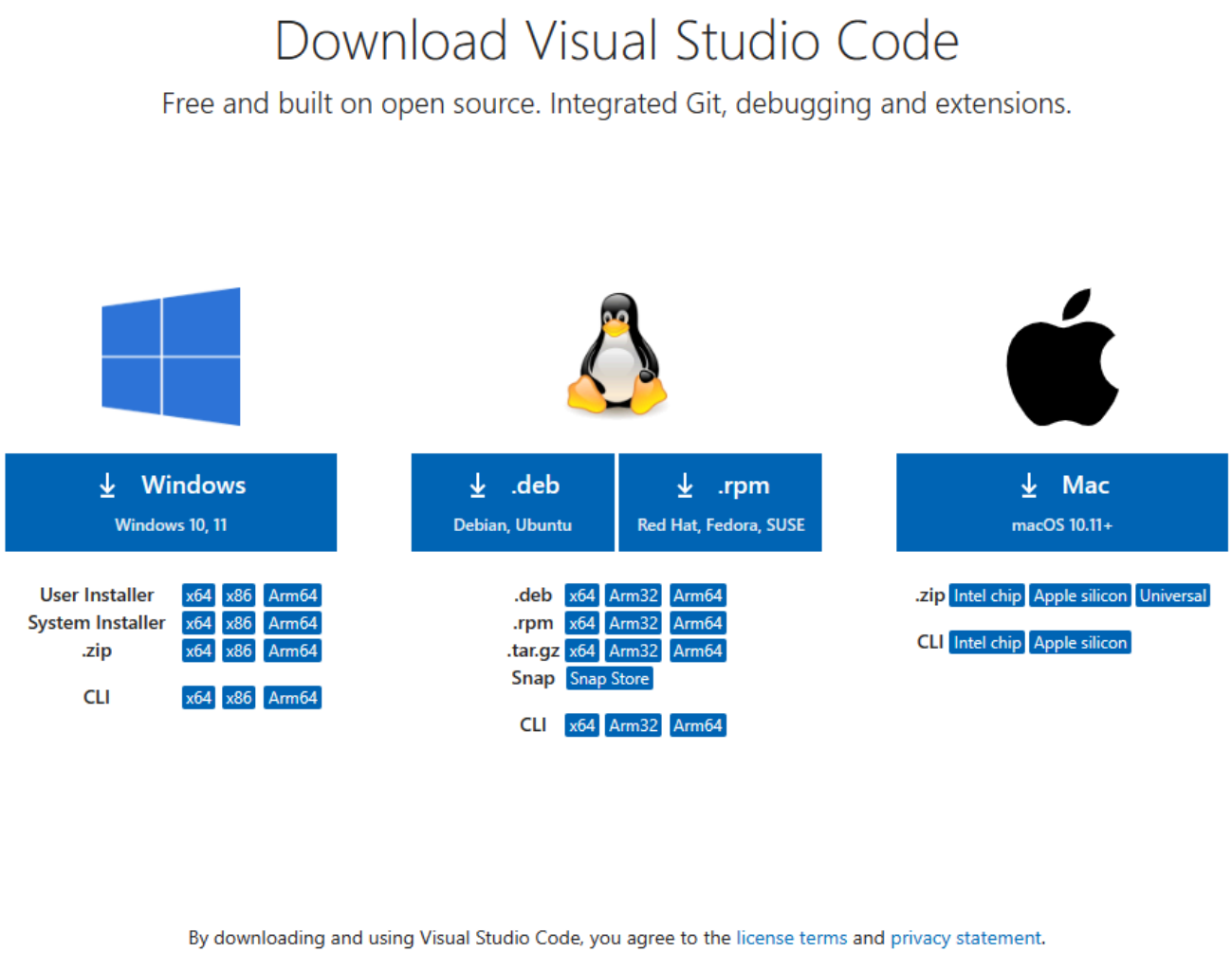
The following development system defaults to Windows, and it is recommended to use

the VSCode plug-in for development.

Develop with VSCode

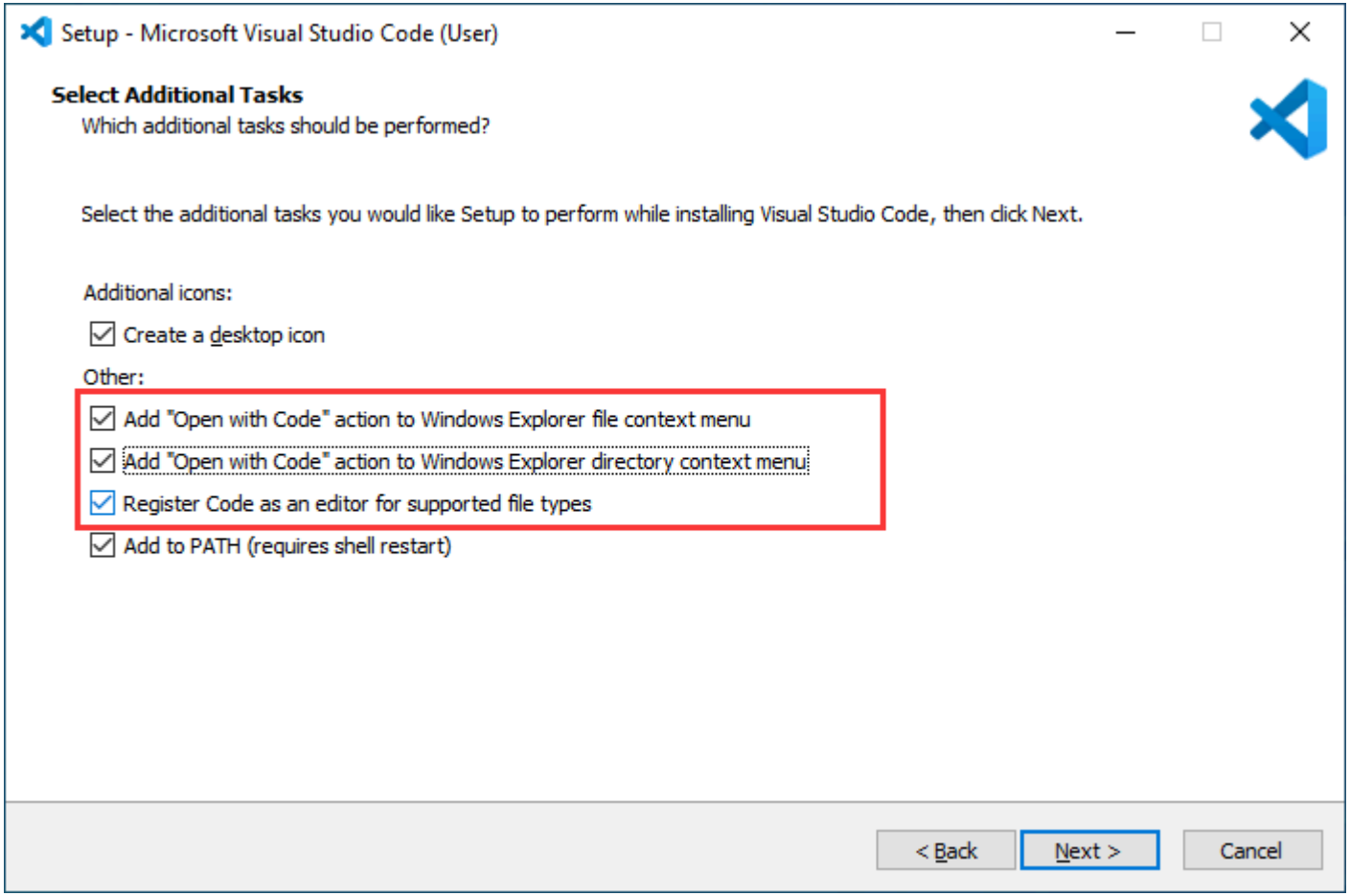
Install VSCode

- Open the VSCode website (https://code.visualstudio.com/download) to download according to the corresponding system and system bits.



(/wiki/File:ESP32-C6-DEV-KIT-N8-01.png)

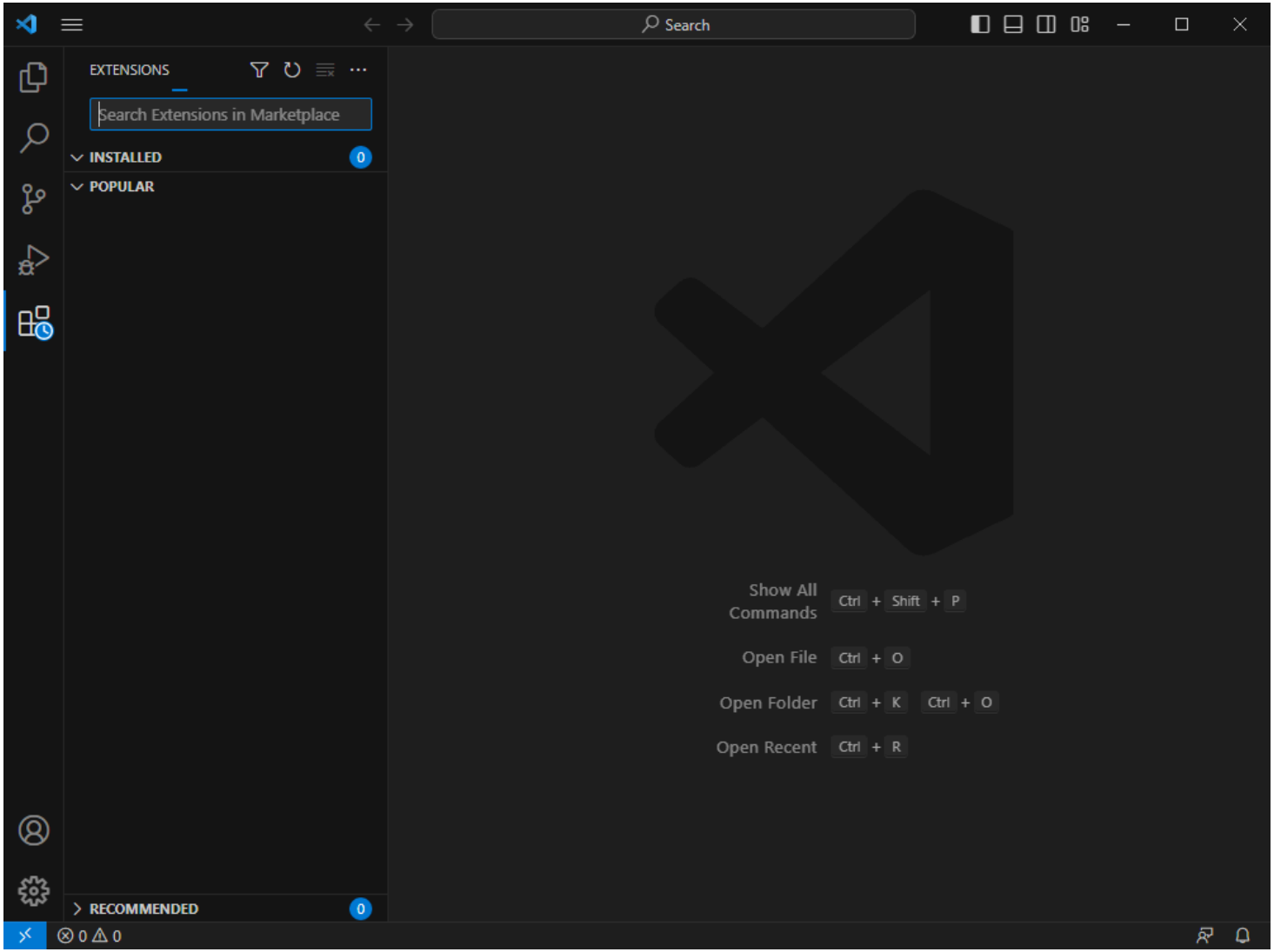
- After running the installation package, the rest can be installed by default, but here for a better experience, it is recommended to check the box here in the 1, 2 and 3 items.
  - After enabling the 1st and 2nd items, you can directly open the VScode by right-clicking the file or the directory to improve your experience.
  - After enabling the 3rd items, you can directly select VSCode when choosing how to open,



(/wiki/File:ESP32-C6-DEV-KIT-N8-02.png)

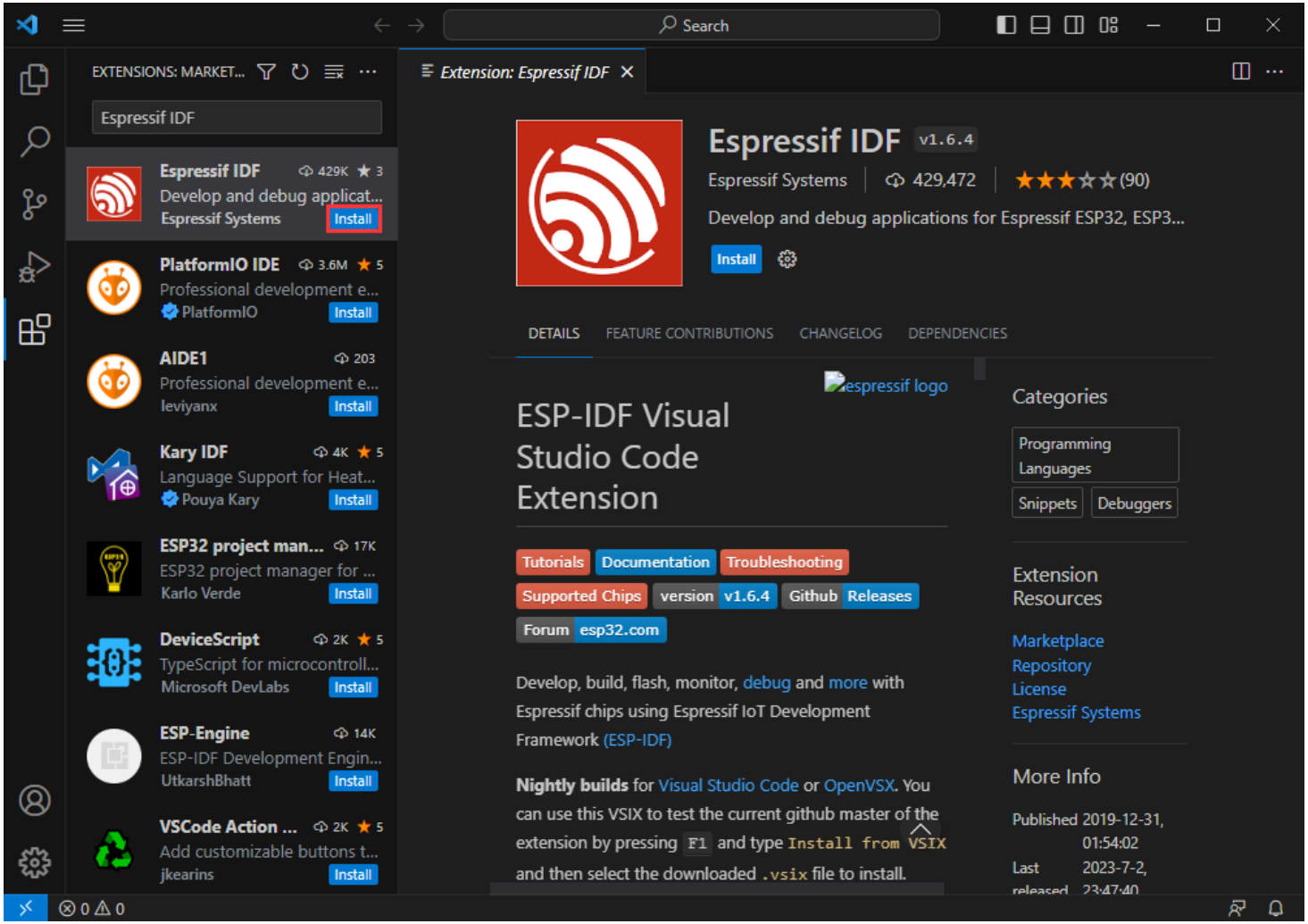
Install Espressif IDF Plug-in

- **Note: Currently the latest version of the plugin is V1.6.4, users can choose the same version as ours for a consistent experience!**
- Open VSCode, use **Shift+Ctrl+X** to enter the plug-in manager.

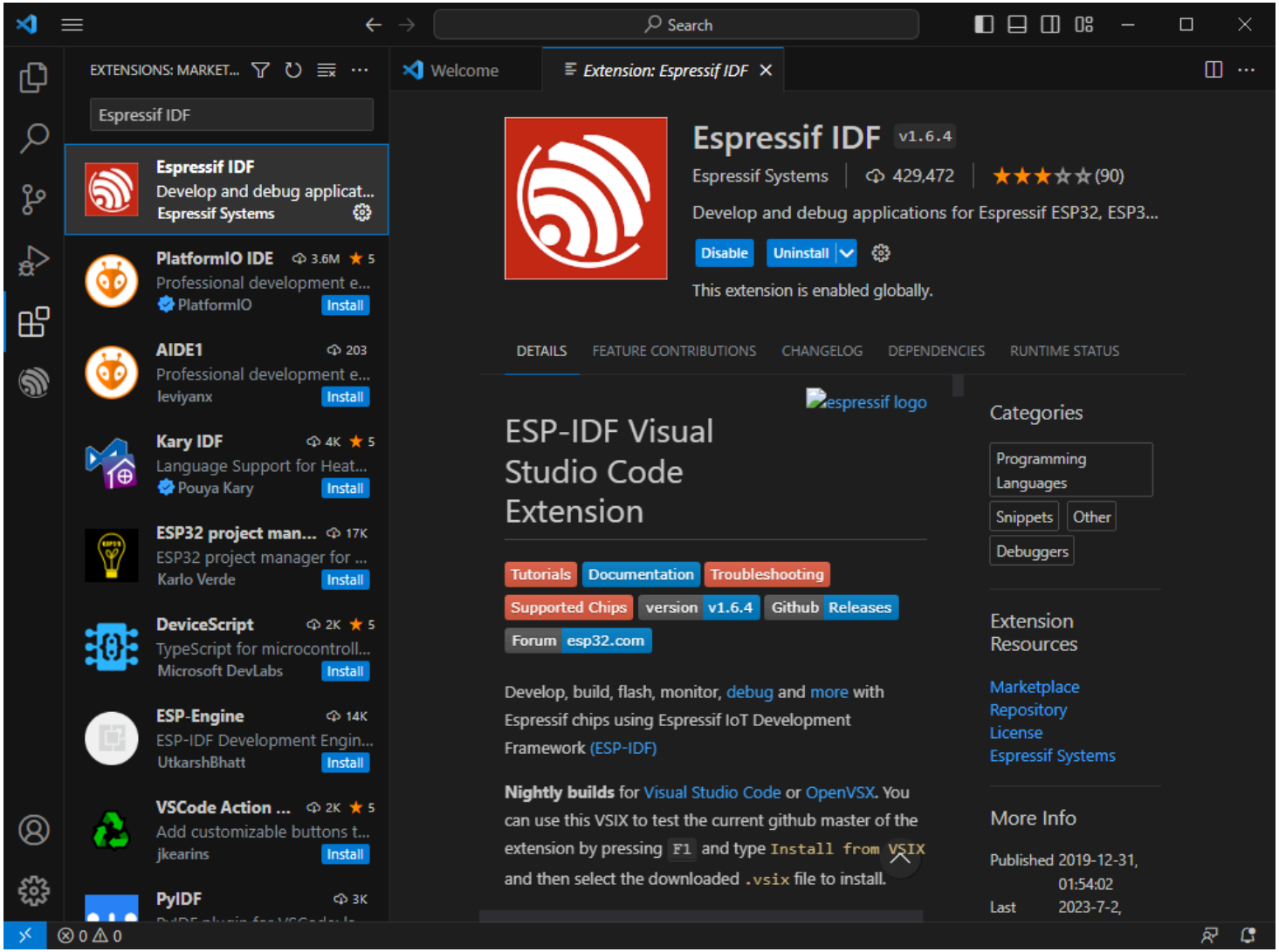


(/wiki/File:ESP32-C6-DEV-KIT-N8-03.png)

- In the search bar, enter **Espressif IDF** to select the corresponding plug-in and click "Install".



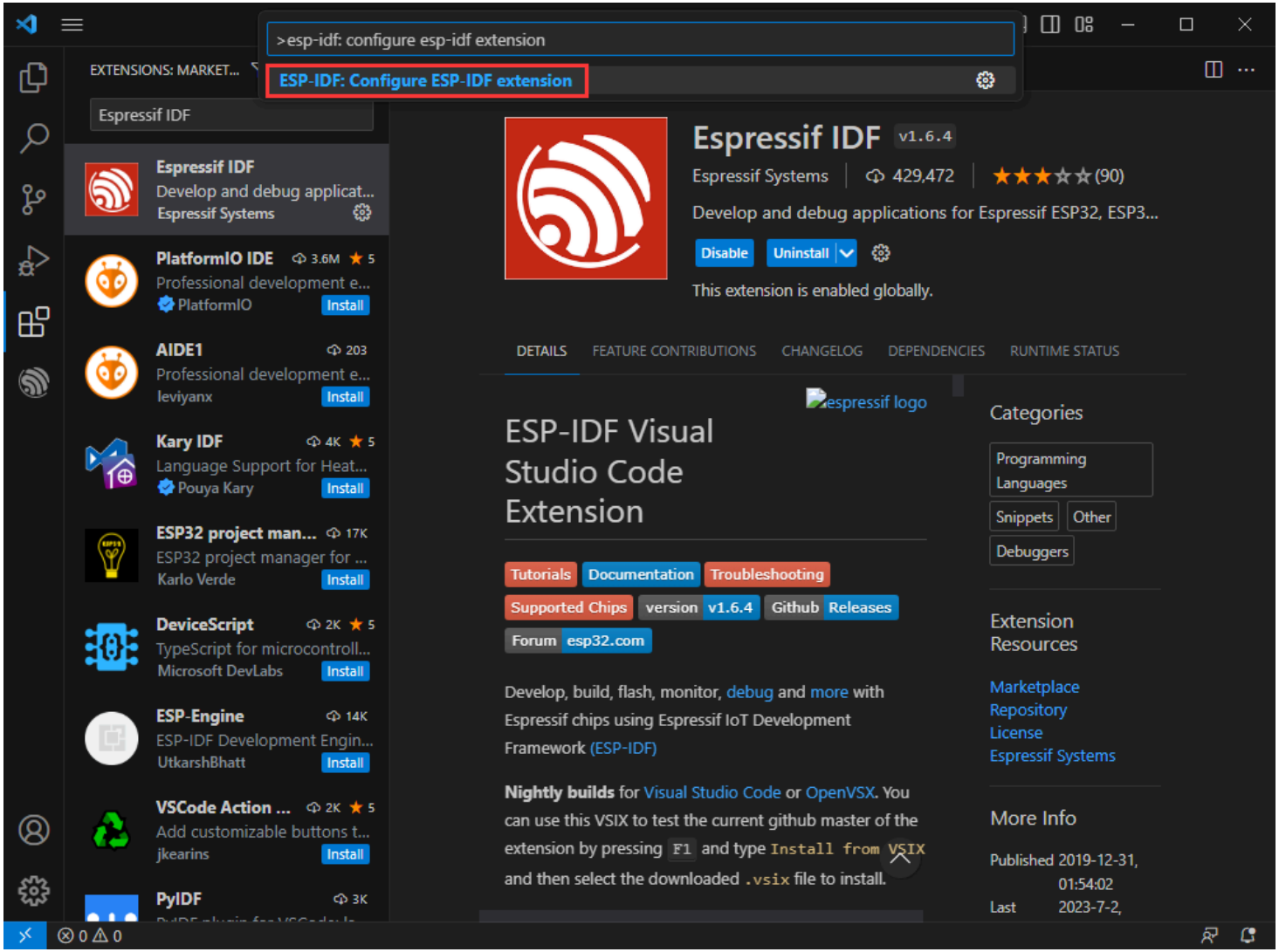
(/wiki/File:ESP32-C6-DEV-KIT-N8-04.png)



(/wiki/File:ESP32-C6-DEV-KIT-N8-05.png)

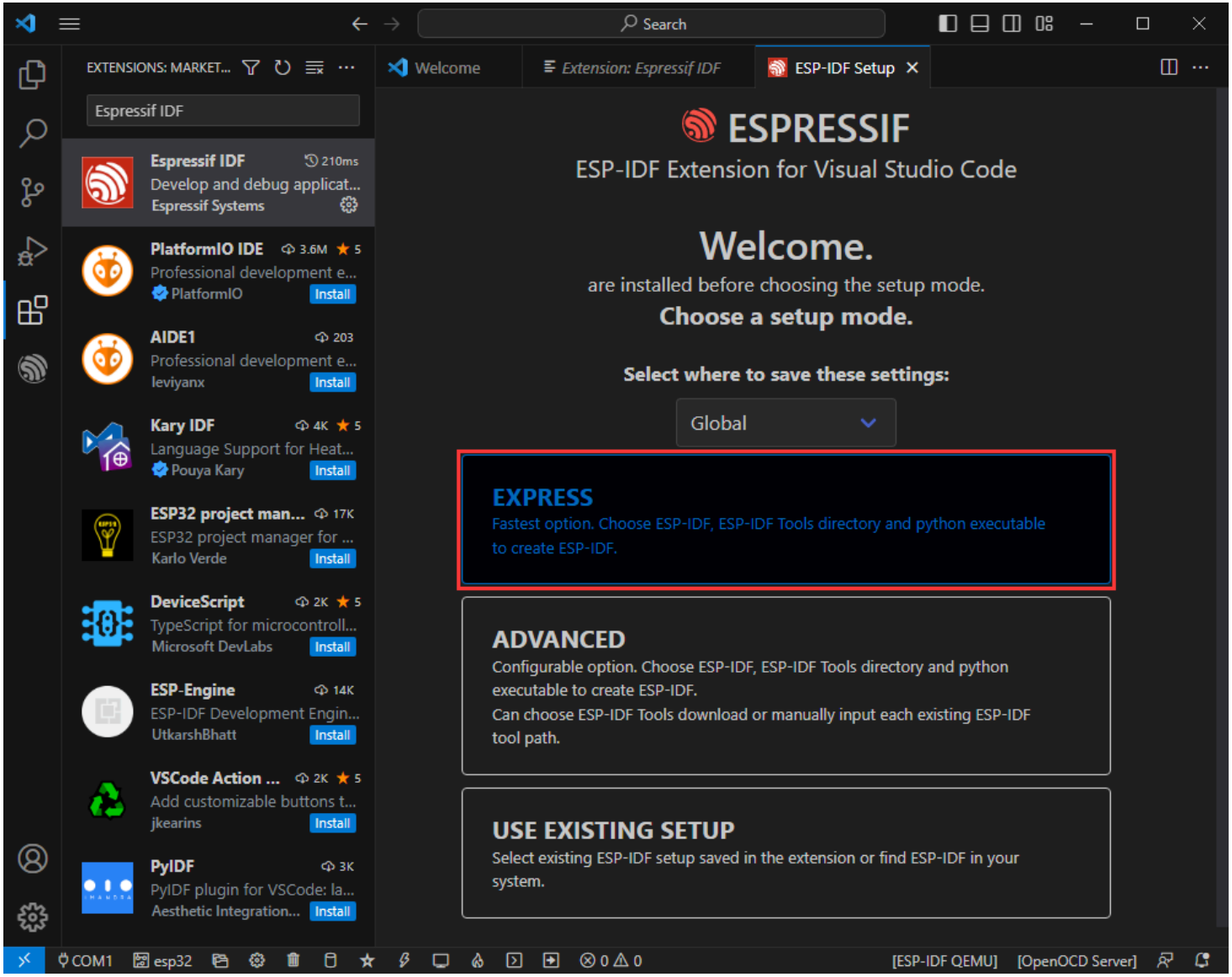
- Press **F1** to input:

```
esp-idf: configure esp-idf extension
```



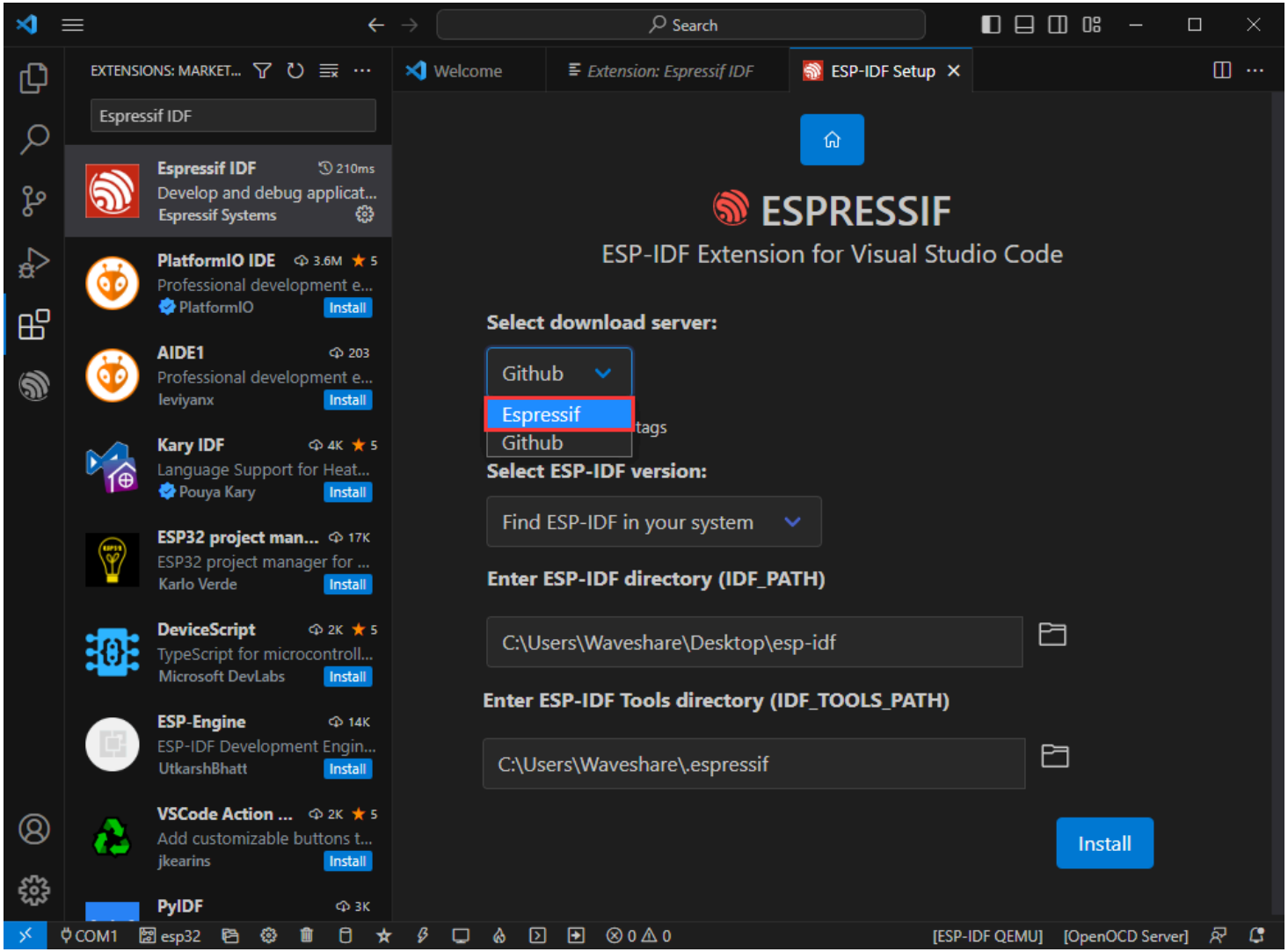
(/wiki/File:ESP32-C6-DEV-KIT-N8-06.png)

- Select express (this guide is for users who install it for the first time).



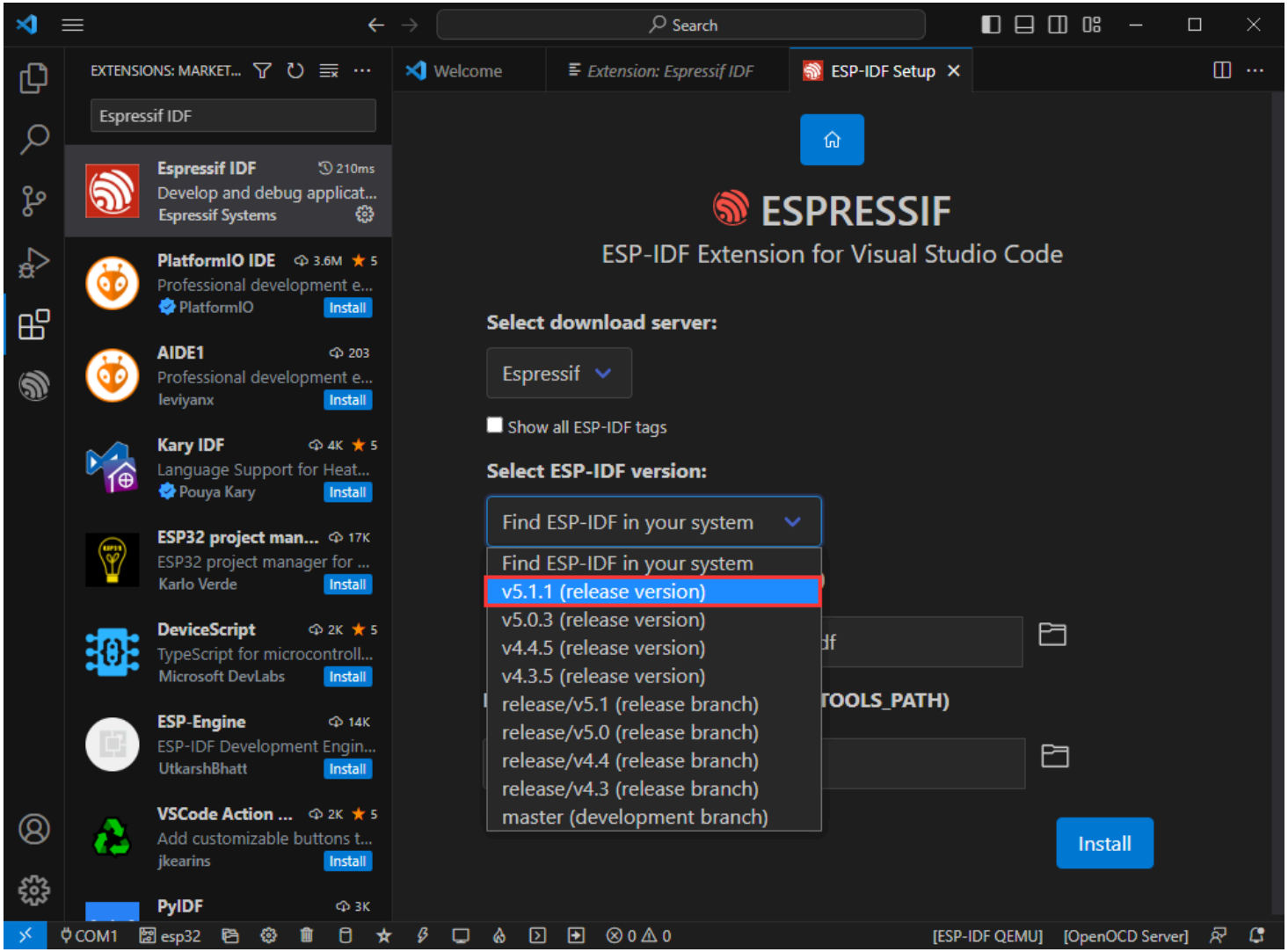
(/wiki/File:ESP32-C6-DEV-KIT-N8-07.png)

- Select download sever.



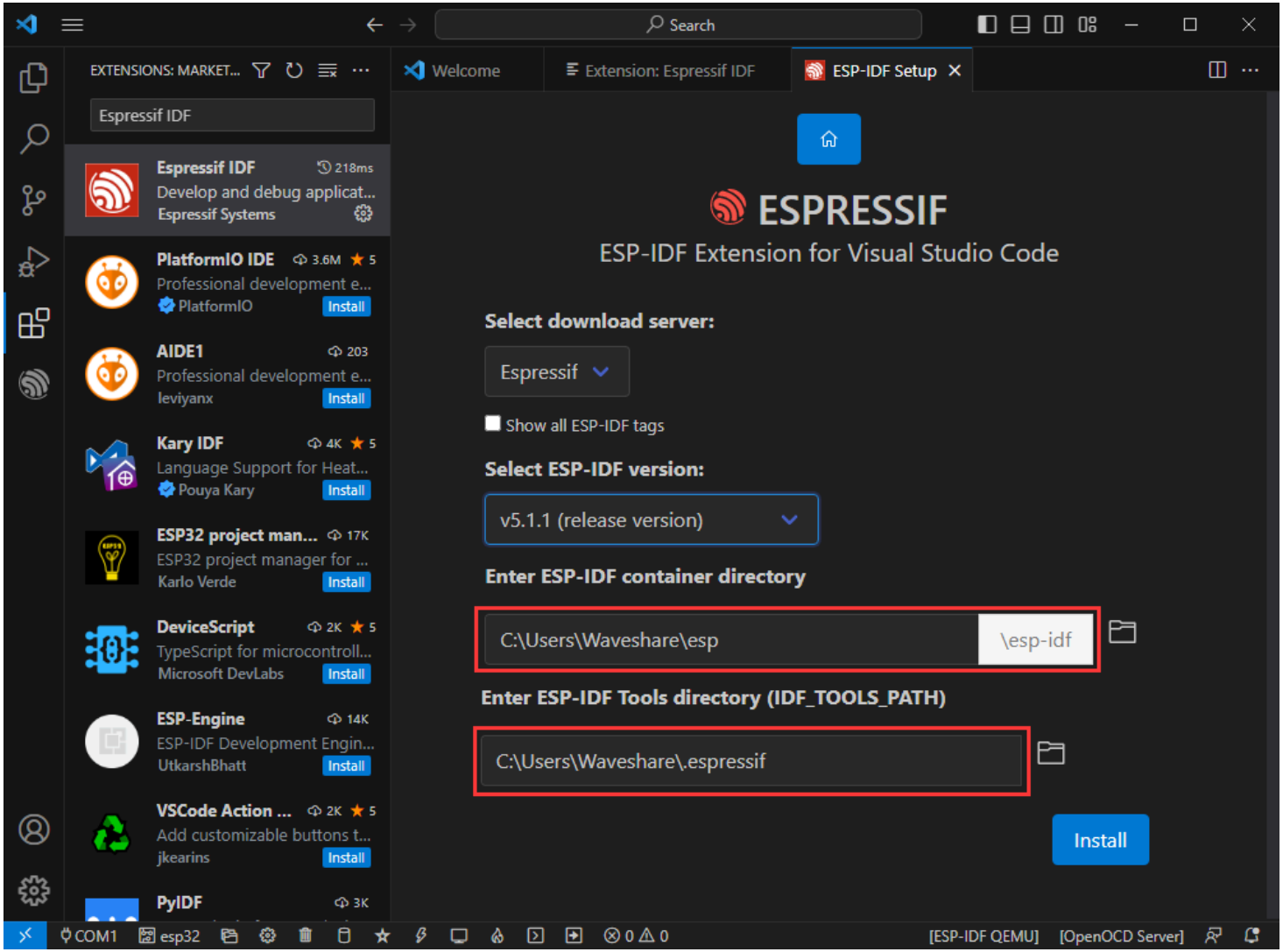
(/wiki/File:ESP32-C6-DEV-KIT-N8-08.png)

- Select the version of ESP-IDF you want to use now, we choose the latest V5.1.1 (note that only after V5.1 did ESP-IDF start to support ESP32-C6).



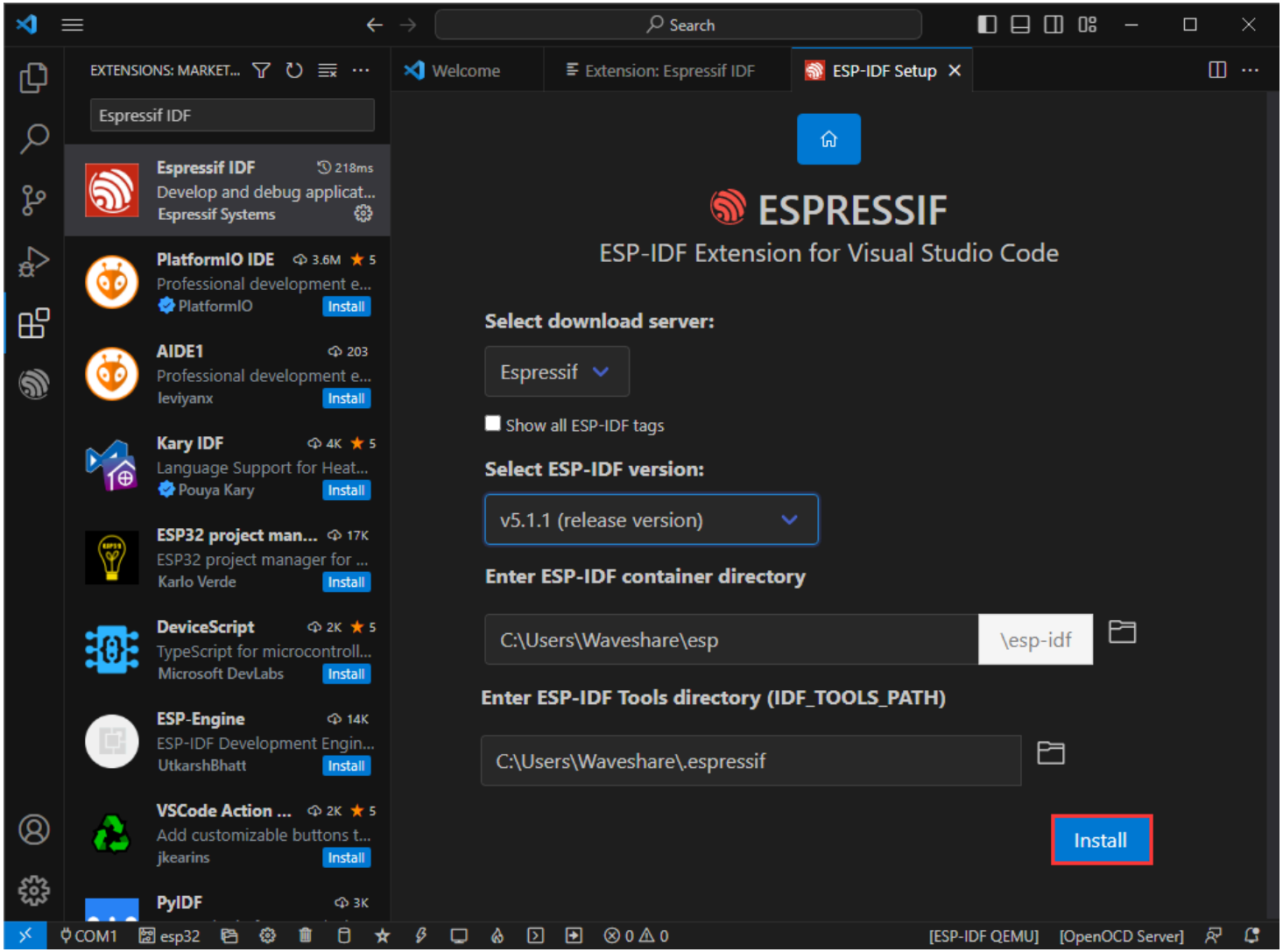
(/wiki/File:ESP32-C6-DEV-KIT-N8-09.png)

- The following two are the installation paths respectively for the ESP-IDF container directory and the ESP-IDF Tools directory.



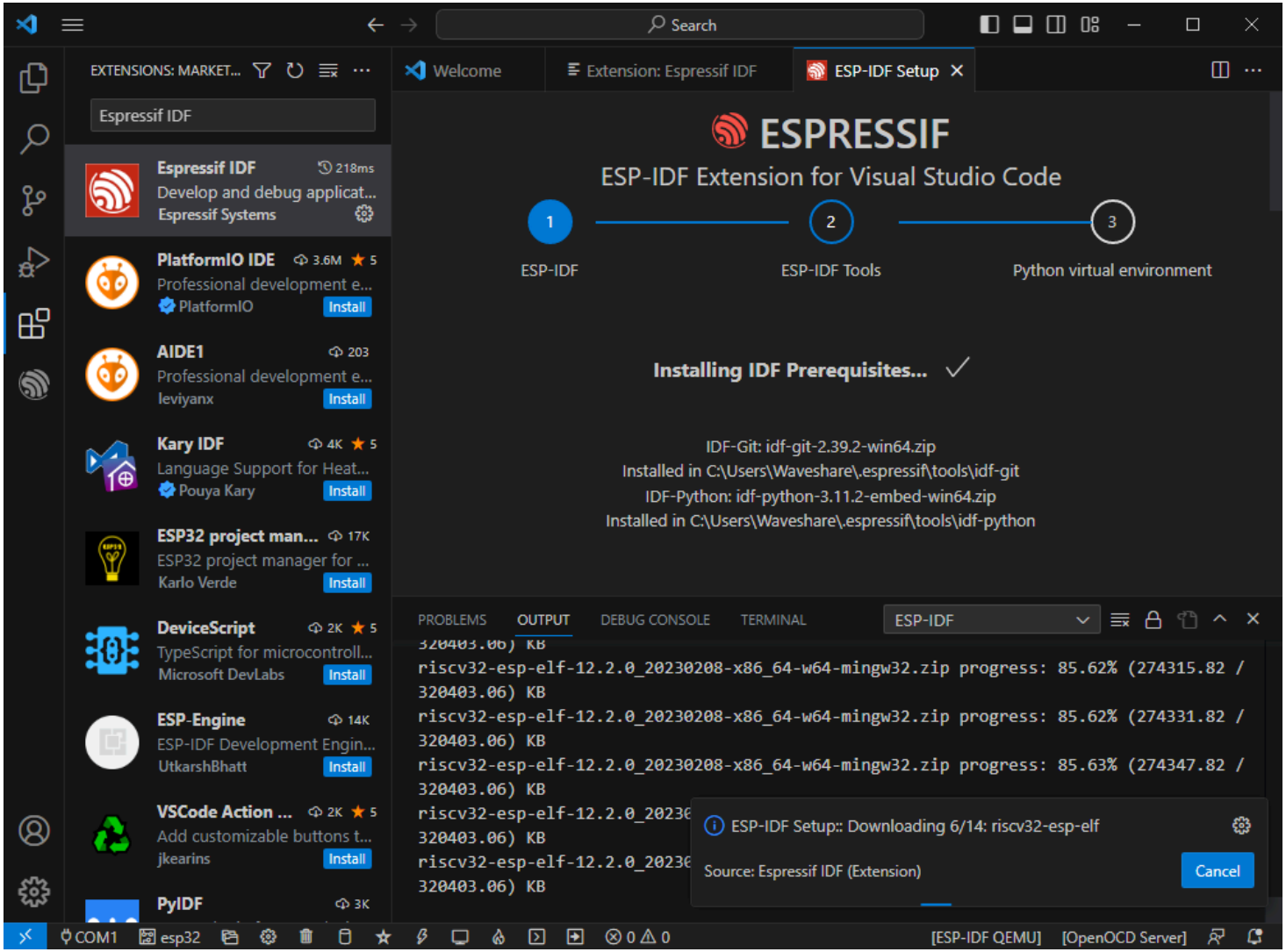
(/wiki/File:ESP32-C6-DEV-KIT-N8-10.png)

- **Note: If you have installed ESP-IDF before, or failed to do so, please be sure to delete the file completely.**
- After configuring, click "Install" to download:



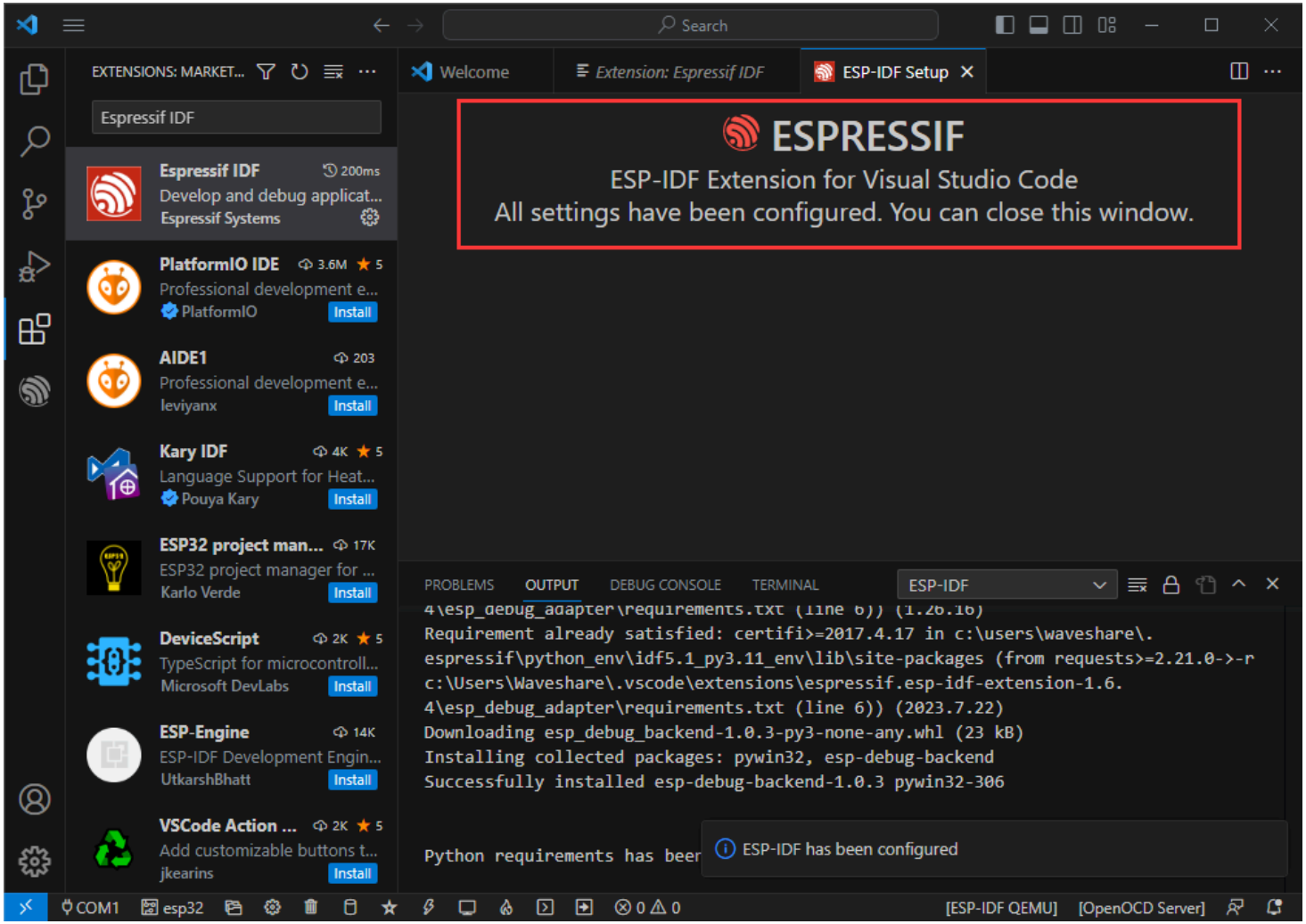
(/wiki/File:ESP32-C6-DEV-KIT-N8-19.png)

- Enter the download interface, and then it will automatically install the corresponding tools and environment, just wait for a second.



(/wiki/File:ESP32-C6-DEV-KIT-N8-11.png)

- After the installation is complete, you will enter the following interface, indicating that the installation is finished.



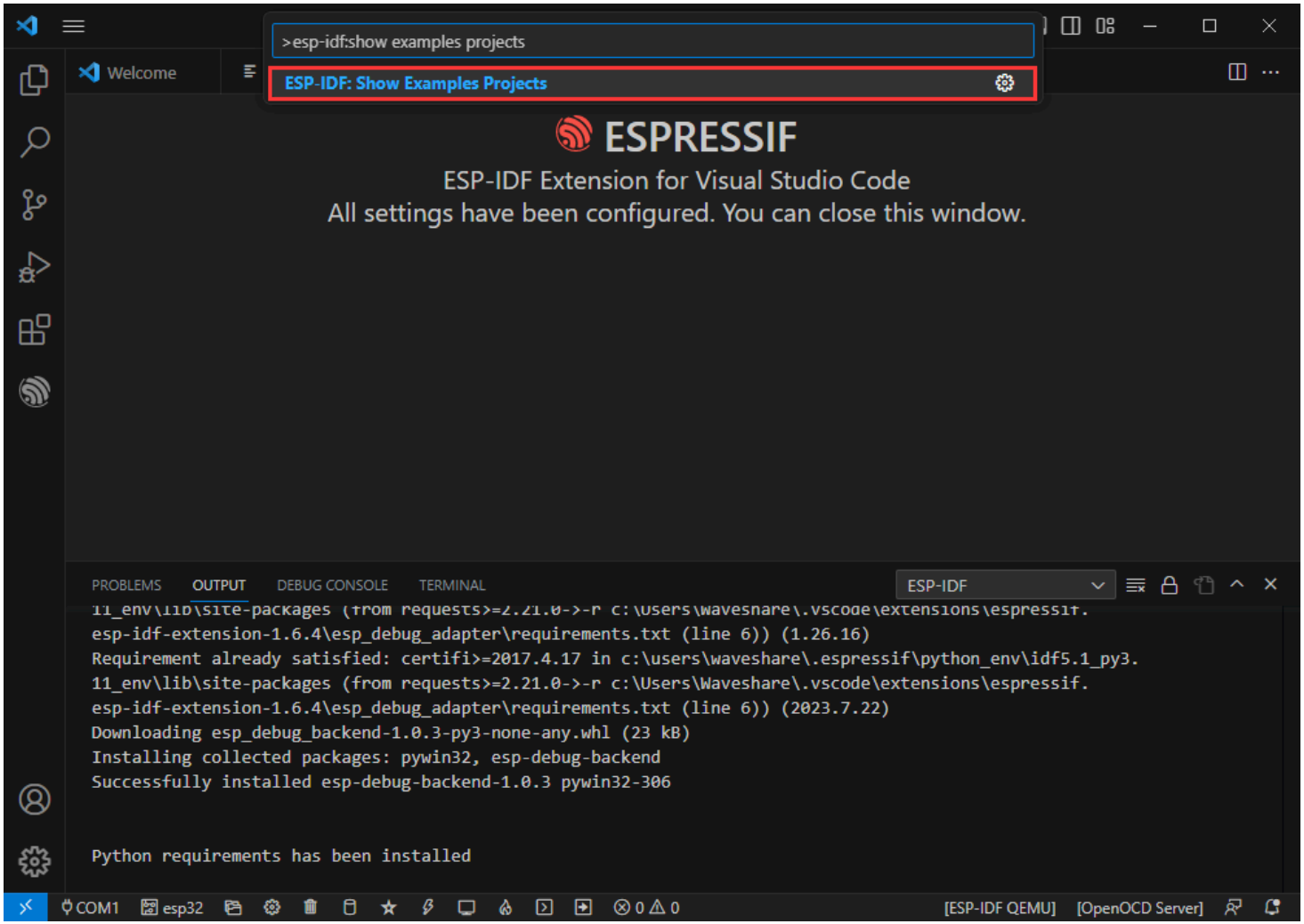
(/wiki/File:ESP32-C6-DEV-KIT-N8-12.png)

Official Demo Usage GUIDE

Create Demo (Demo Example)

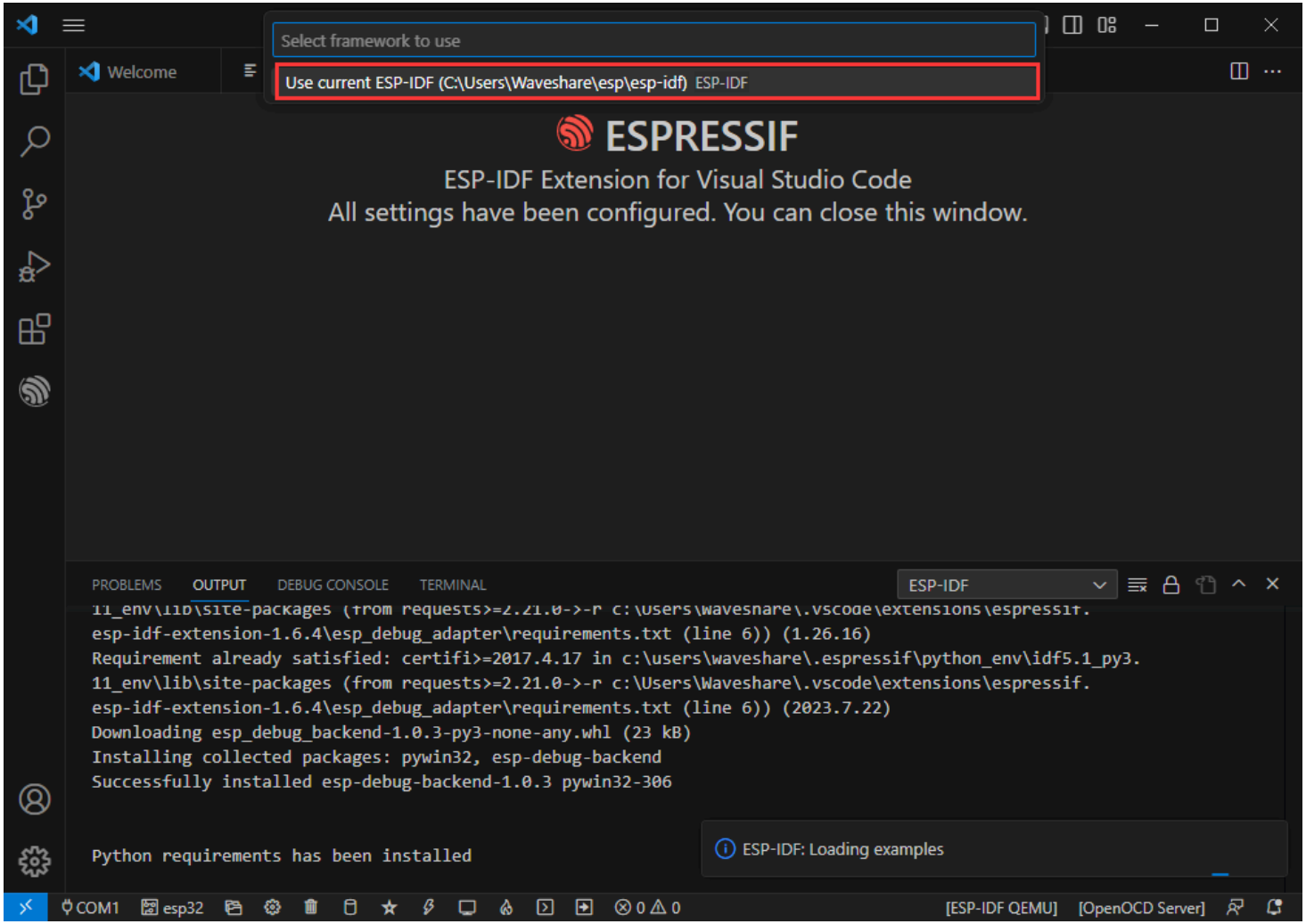
- Press **F1** to enter:

```
esp-idf:show examples projects
```



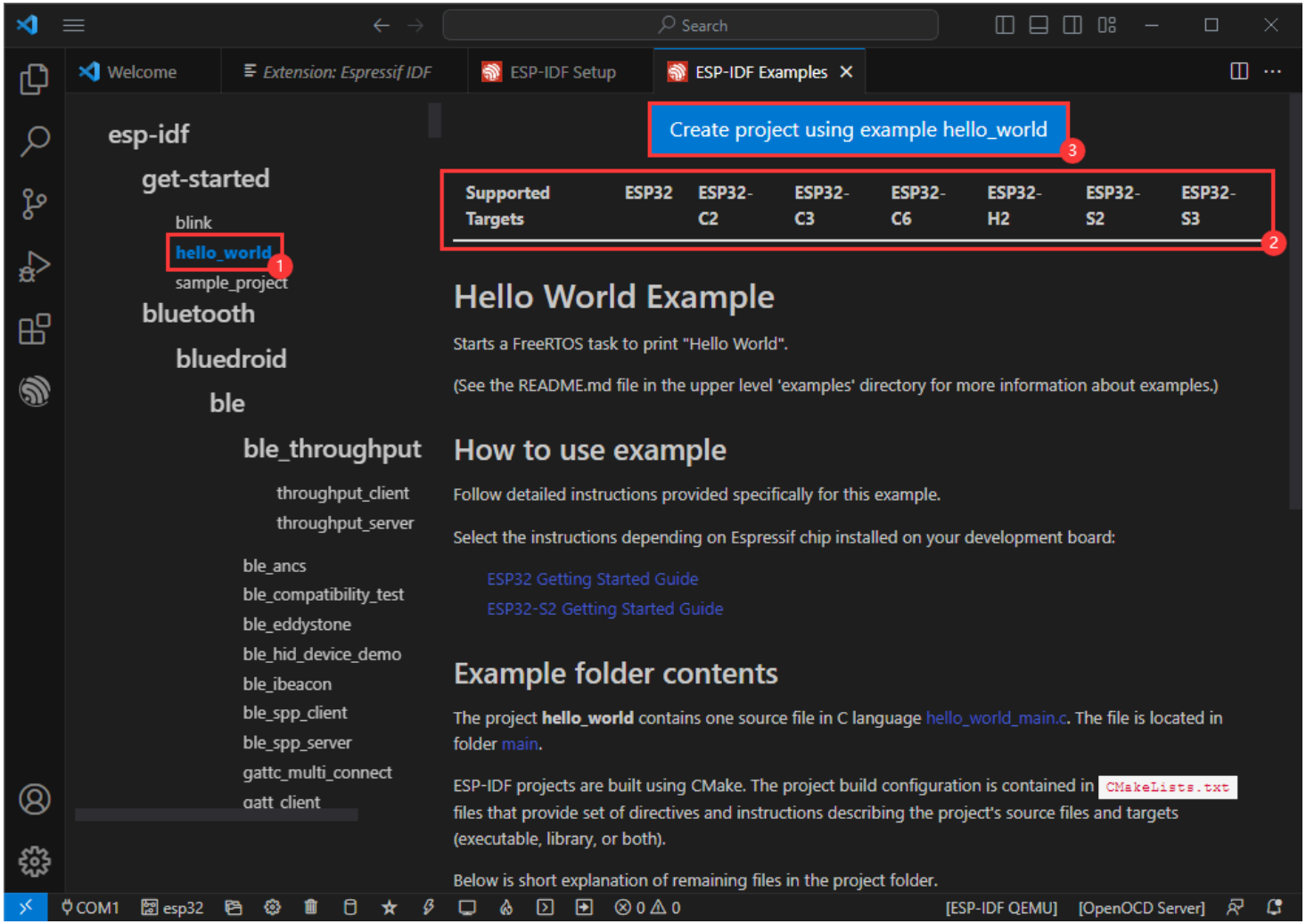
(/wiki/File:ESP32-C6-DEV-KIT-N8-13.png)

- Select the corresponding IDF version:



(/wiki/File:ESP32-C6-DEV-KIT-N8-14.png)

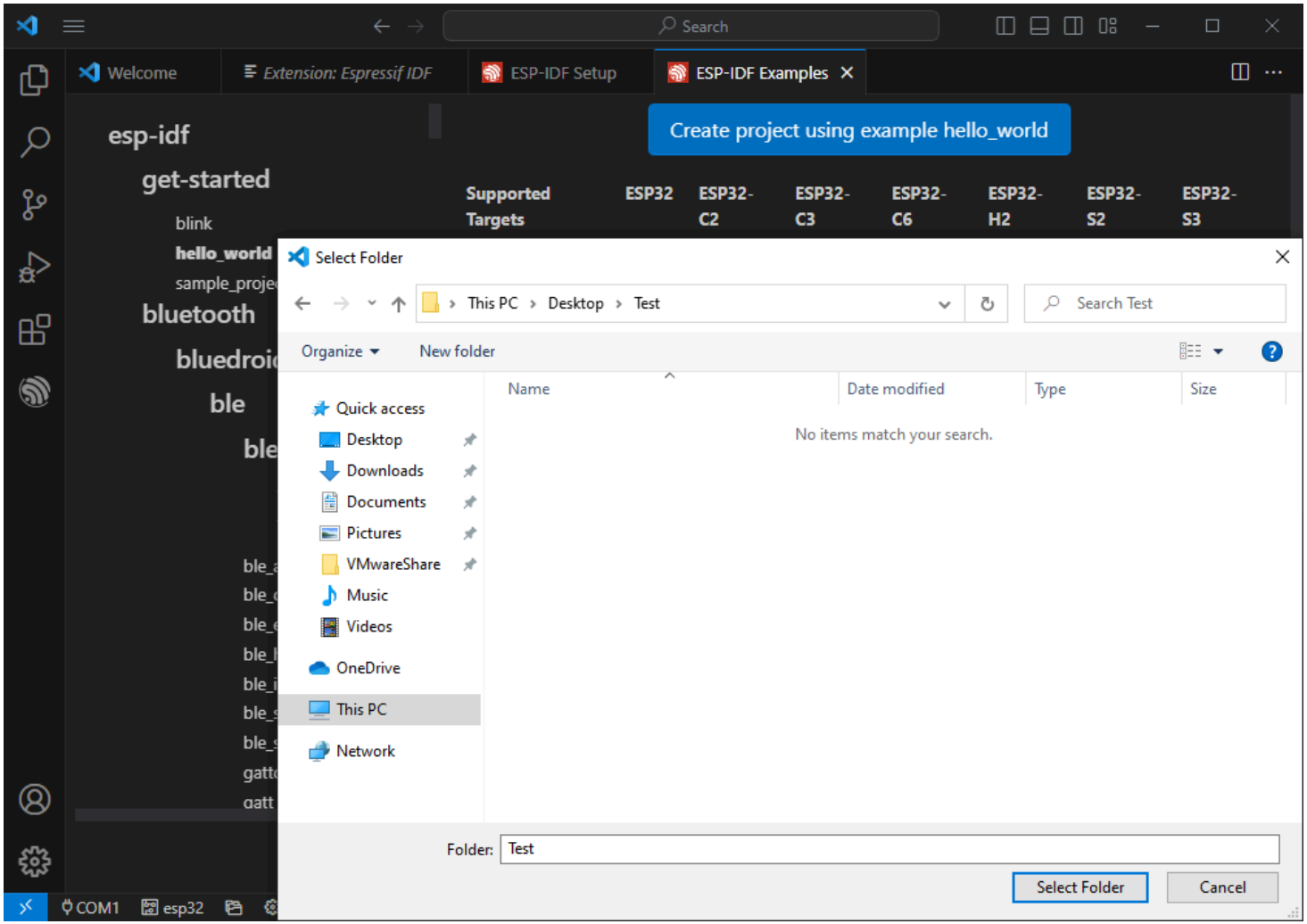
- Take the Hello World demo as an example:
  - ①Select the corresponding demo.
  - ②Its readme will state what chip the demo applies to (how the demo is used and the file structure are described below, omitted here).
  - ③Click to create the demo.



(/wiki/File:ESP32-C6-DEV-KIT-N8-15.png)

Select the path to place the demo, and the path for placing demos should not contain a folder

with the same name as the demo.

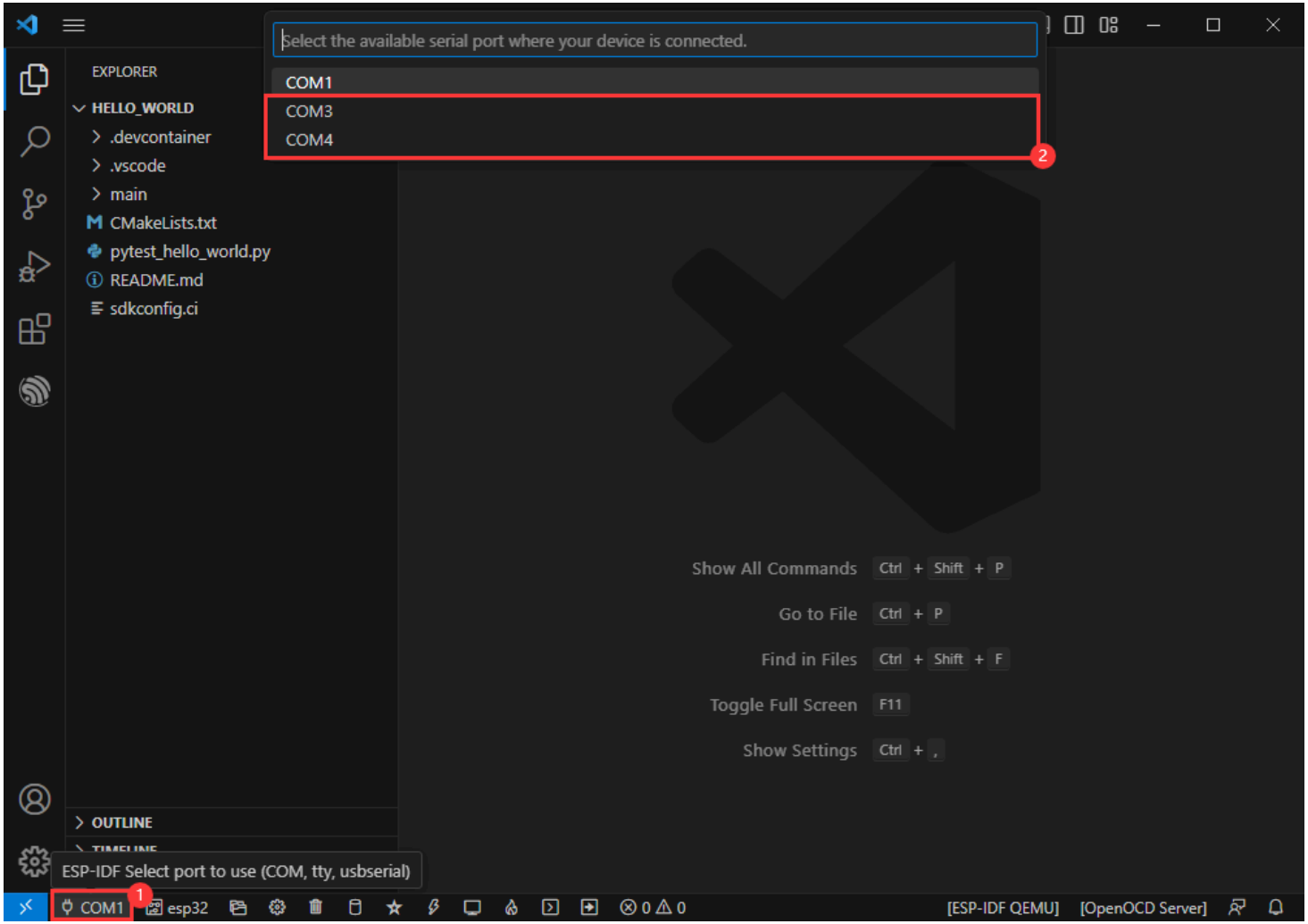


(/wiki/File:ESP32-C6-DEV-KIT-N8-16.png)

Modify COM Port

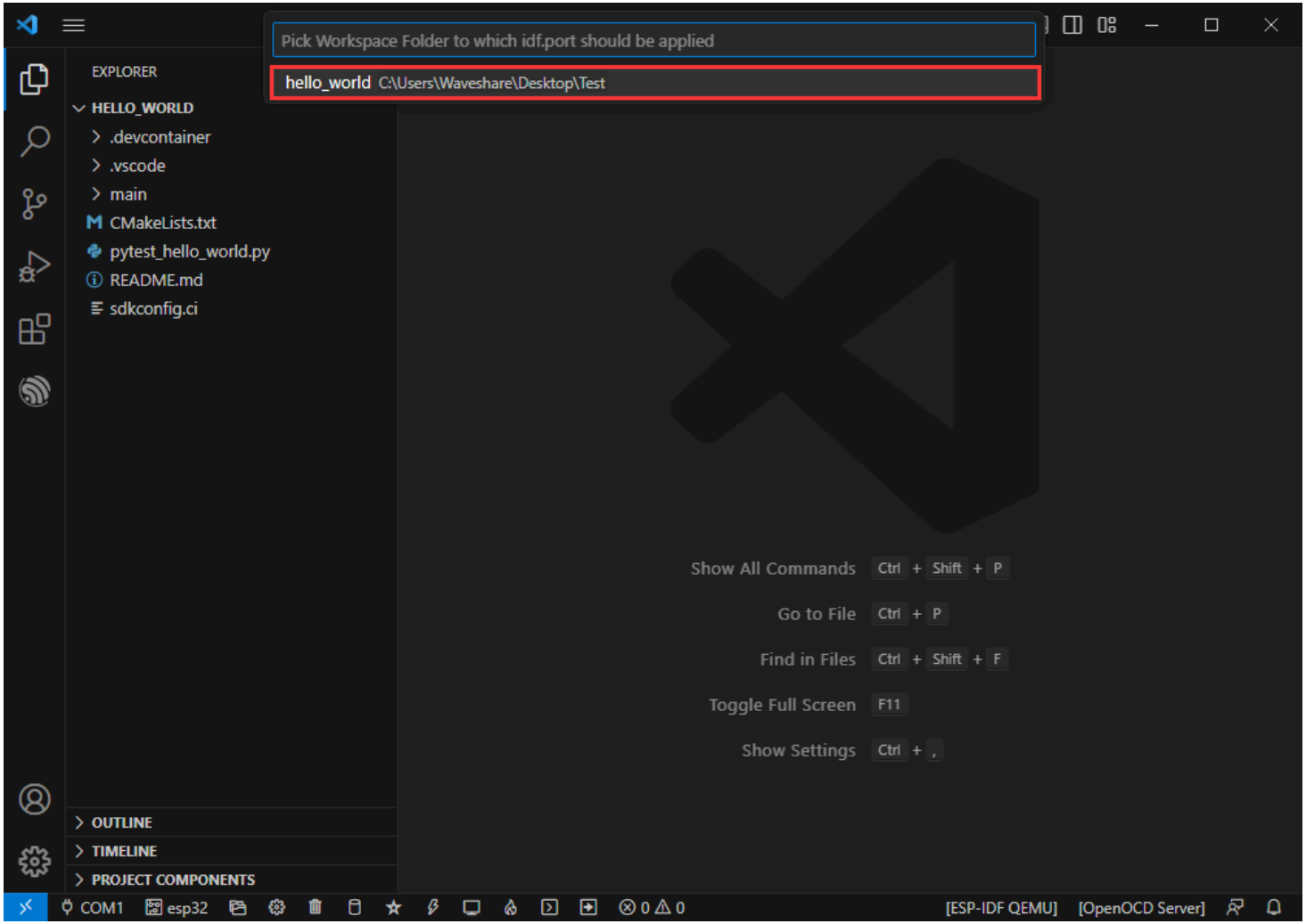
- The corresponding COM ports are shown here, click to modify them.
- Please select the COM ports according to your device. **It is recommended to use the COM port corresponding to the USB connector. (You can view it from the device manager.)**

- In case of a download failure, please press the reset button for more than 1 second and wait for the PC to recognize the device again before downloading once more.



(/wiki/File:ESP32-C6-DEV-KIT-N8-17.png)

- Select the project or demo to use:

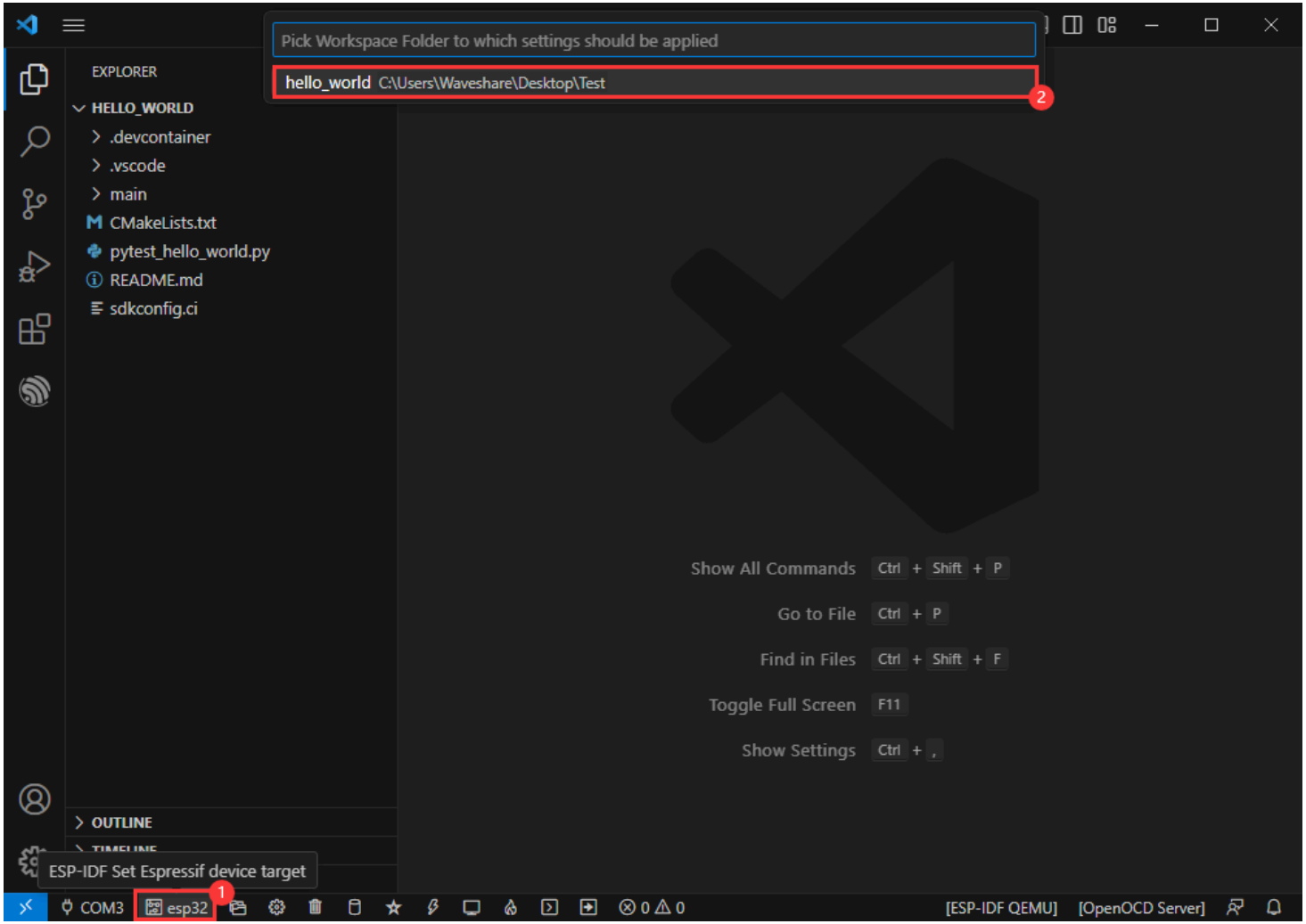


(/wiki/File:ESP32-C6-DEV-KIT-N8-18.png)

- Then we finish the modification of the COM ports.

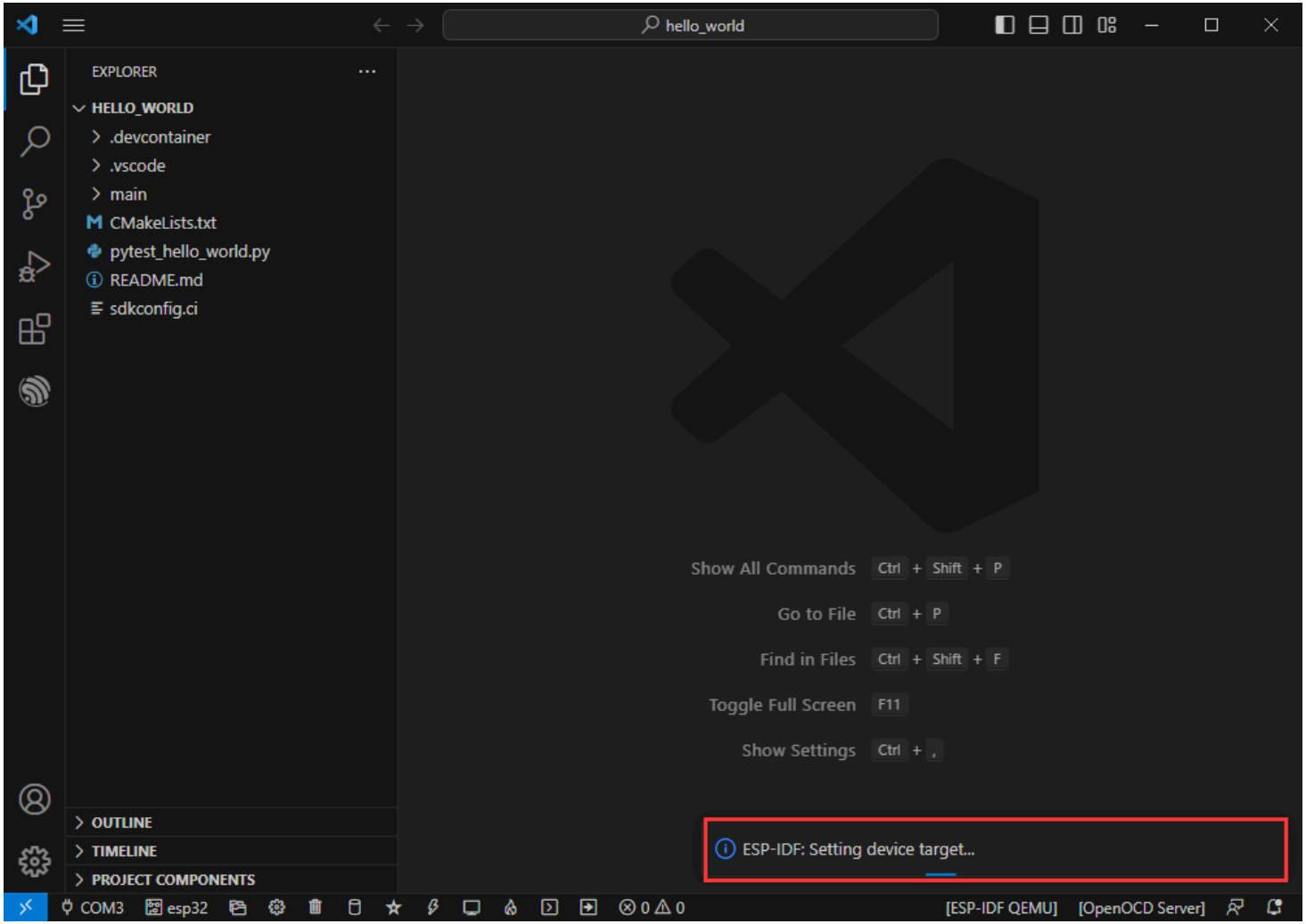
Modify the Driver Object

- The driver object is displayed here, and you can modify it by clicking on it.
- Select the project or demo to use.



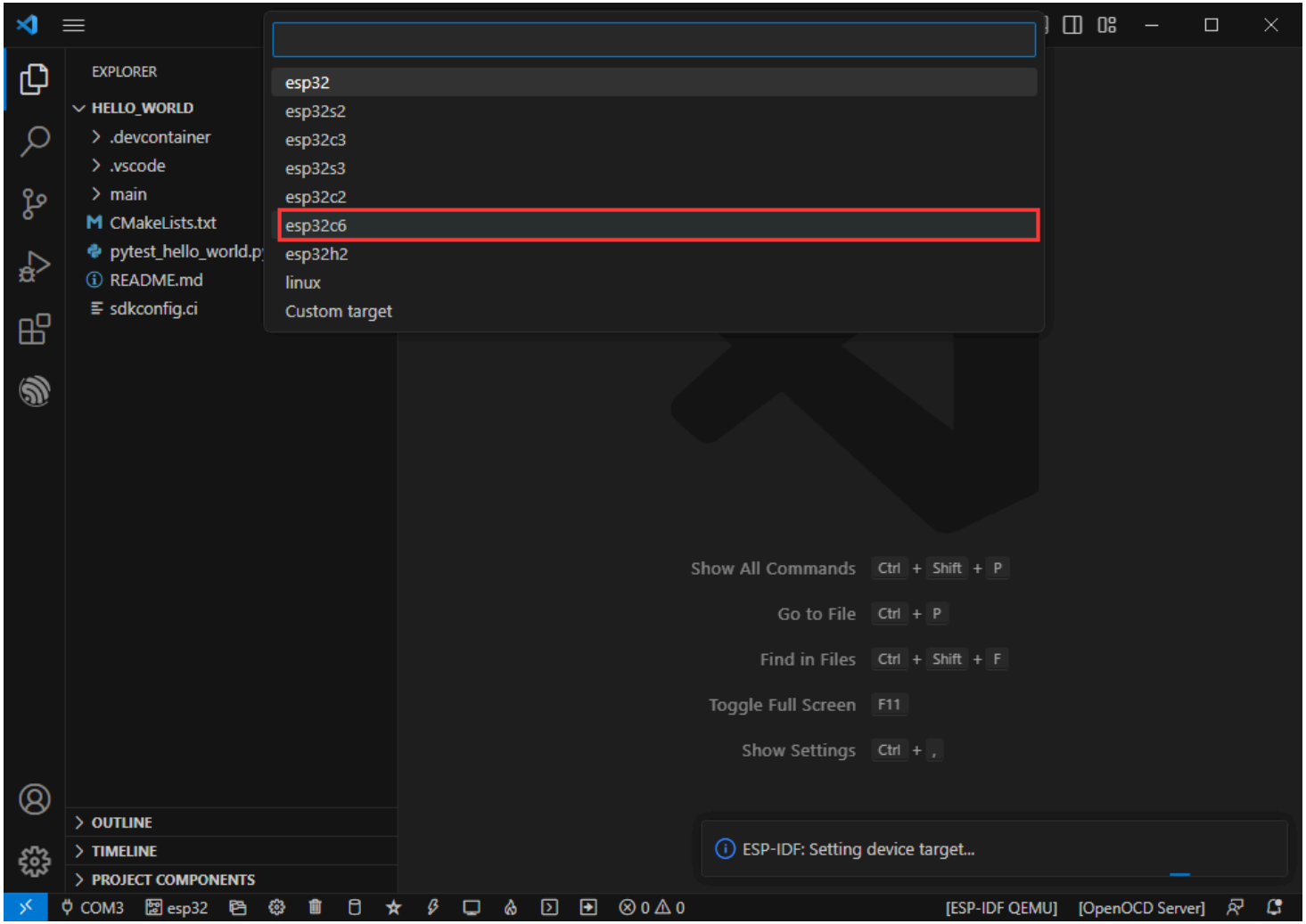
(/wiki/File:ESP32-C6-DEV-KIT-N8-20.png)

- Wait for a minute after clicking.



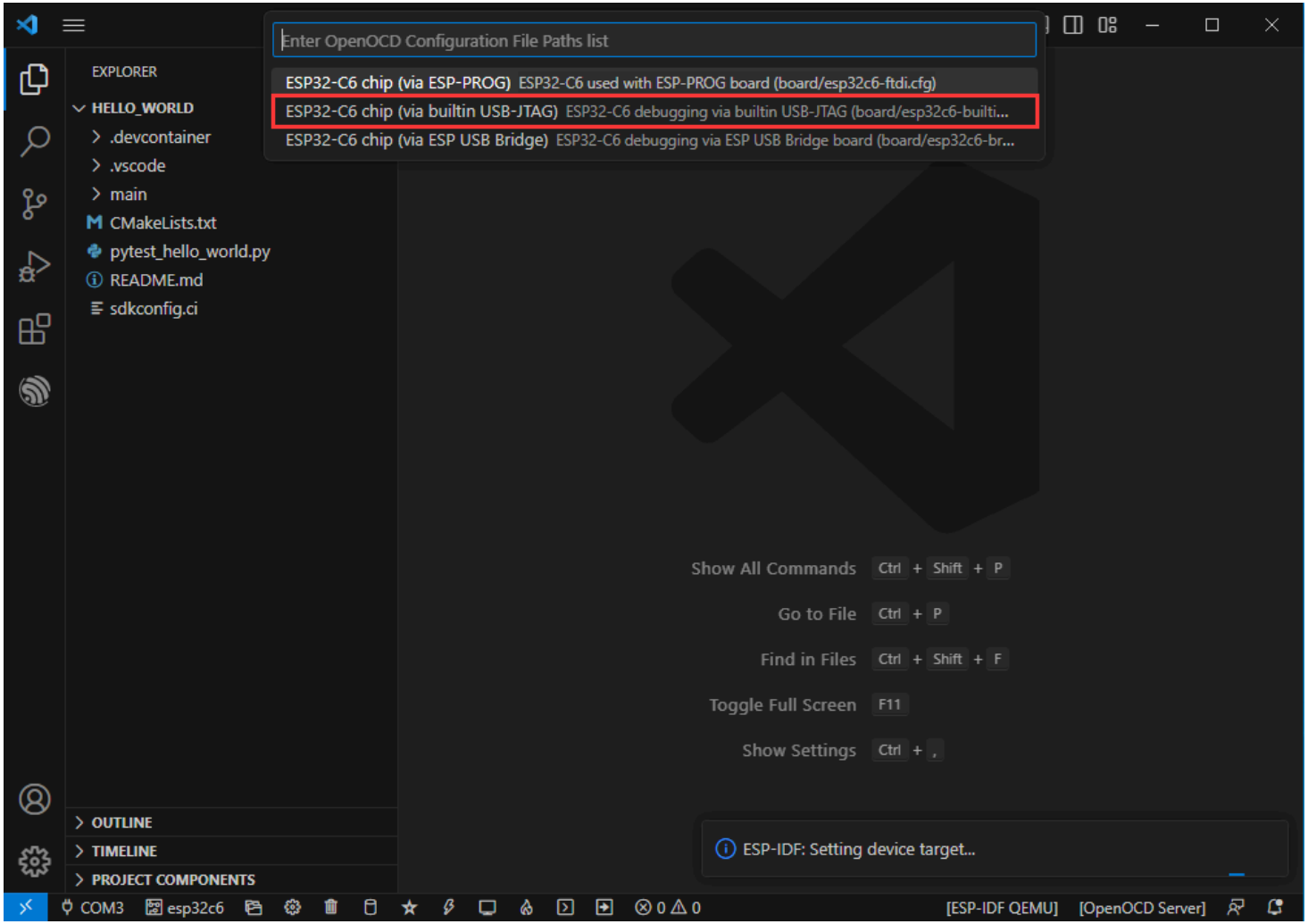
(/wiki/File:ESP32-C6-DEV-KIT-N8-21.png)

- Select the object we need to drive, which is our main chip ESP32C6.



(/wiki/File:ESP32-C6-DEV-KIT-N8-22.png)

- Choose the path to openocd, it doesn't affect us here, so let's just choose one at random.

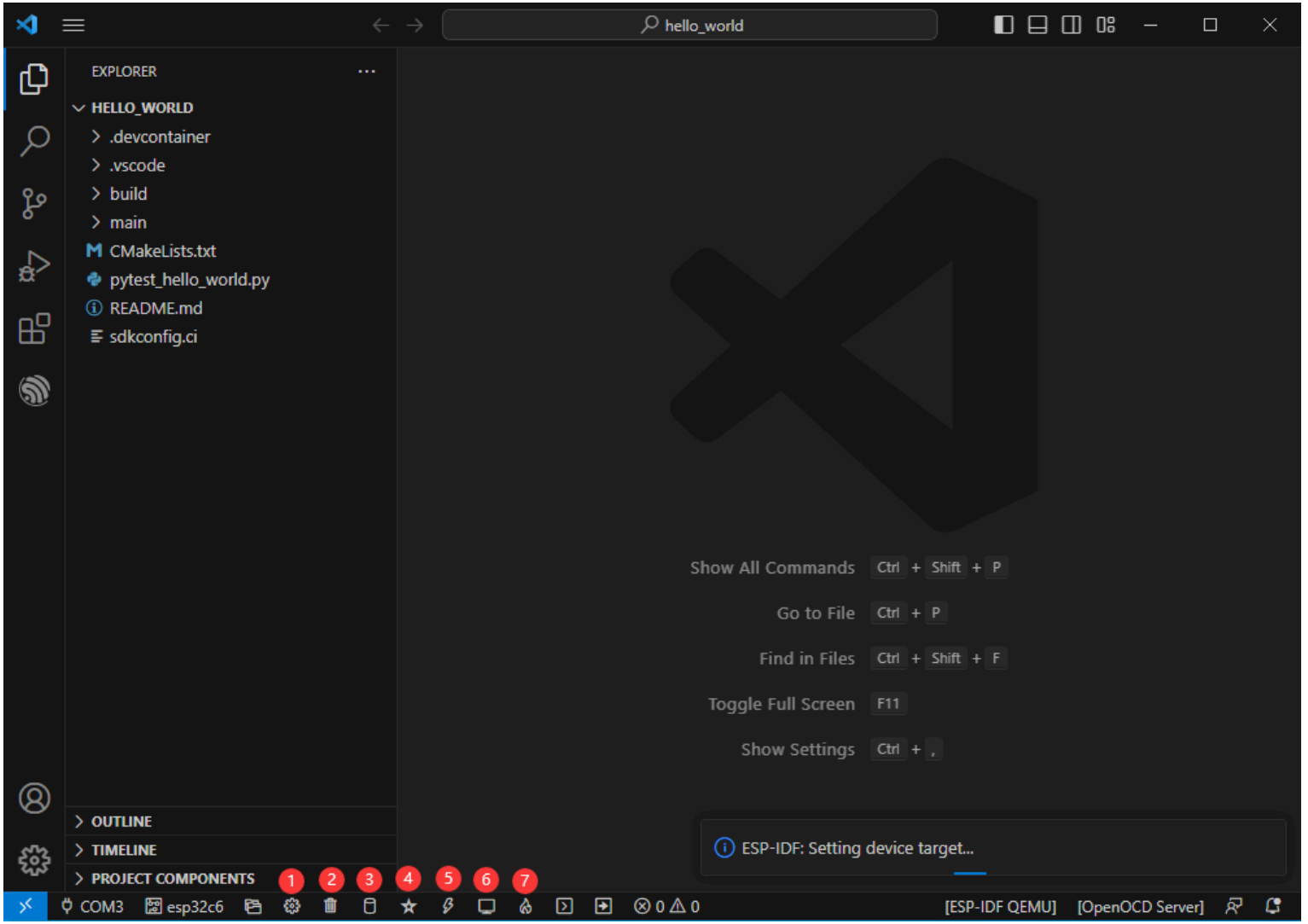


(/wiki/File:ESP32-C6-DEV-KIT-N8-23.png)

The Rest of the Status Bar

- ①SDK configuration editor, supports modifying most functions of ESP-IDF.
- ②All cleanup, and clear all compiled files.
- ③Compile.
- ④Current download mode, default is UART.
- ⑤Flash the current firmware, please do it after compiling.

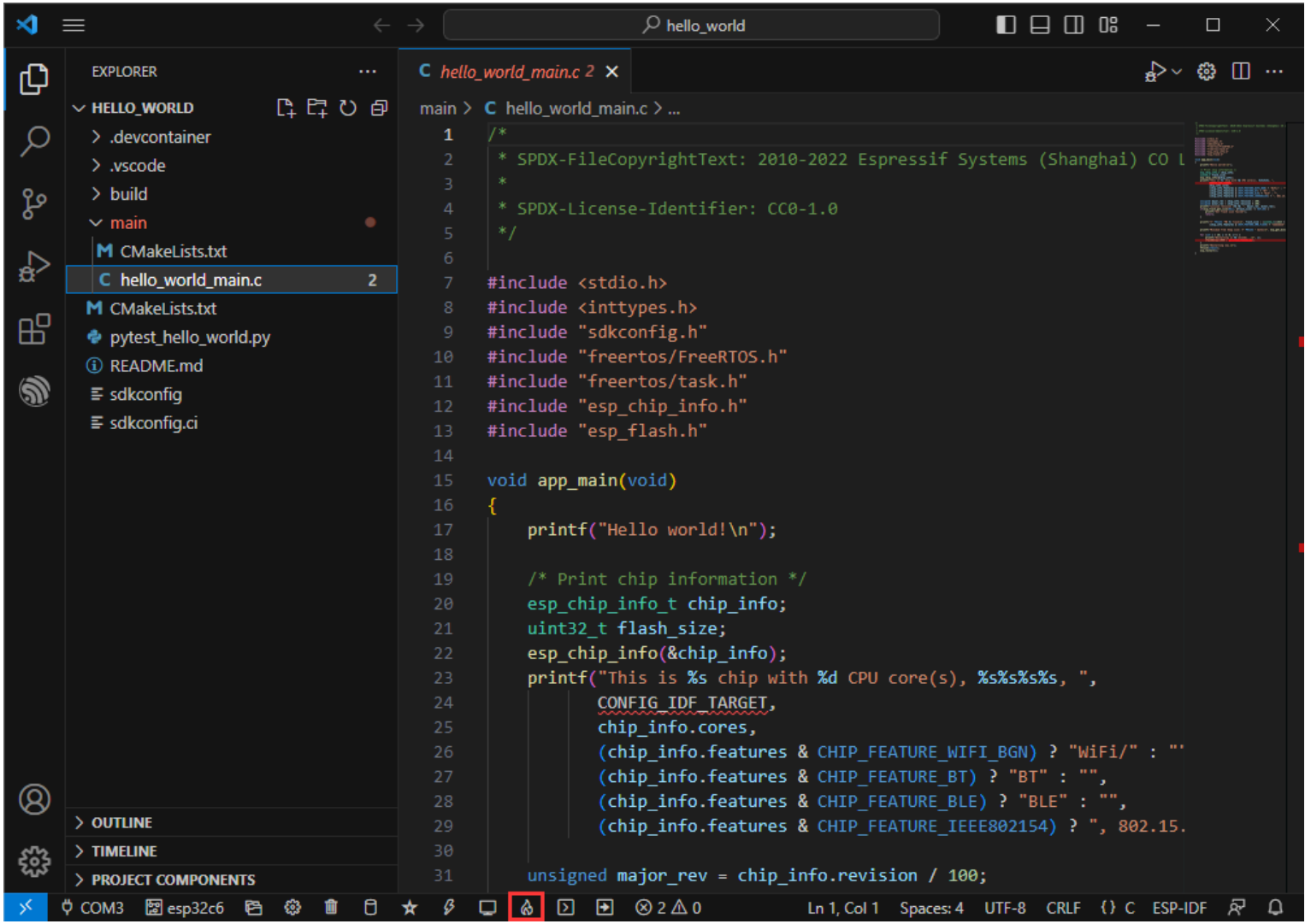
- ⑥Open the serial port monitor, which is used to view the serial port information.
- ⑦All-in-one button, compile, flash and open the serial monitor (most commonly used for debugging).



(/wiki/File:ESP32-C6-DEV-KIT-N8-24.png)

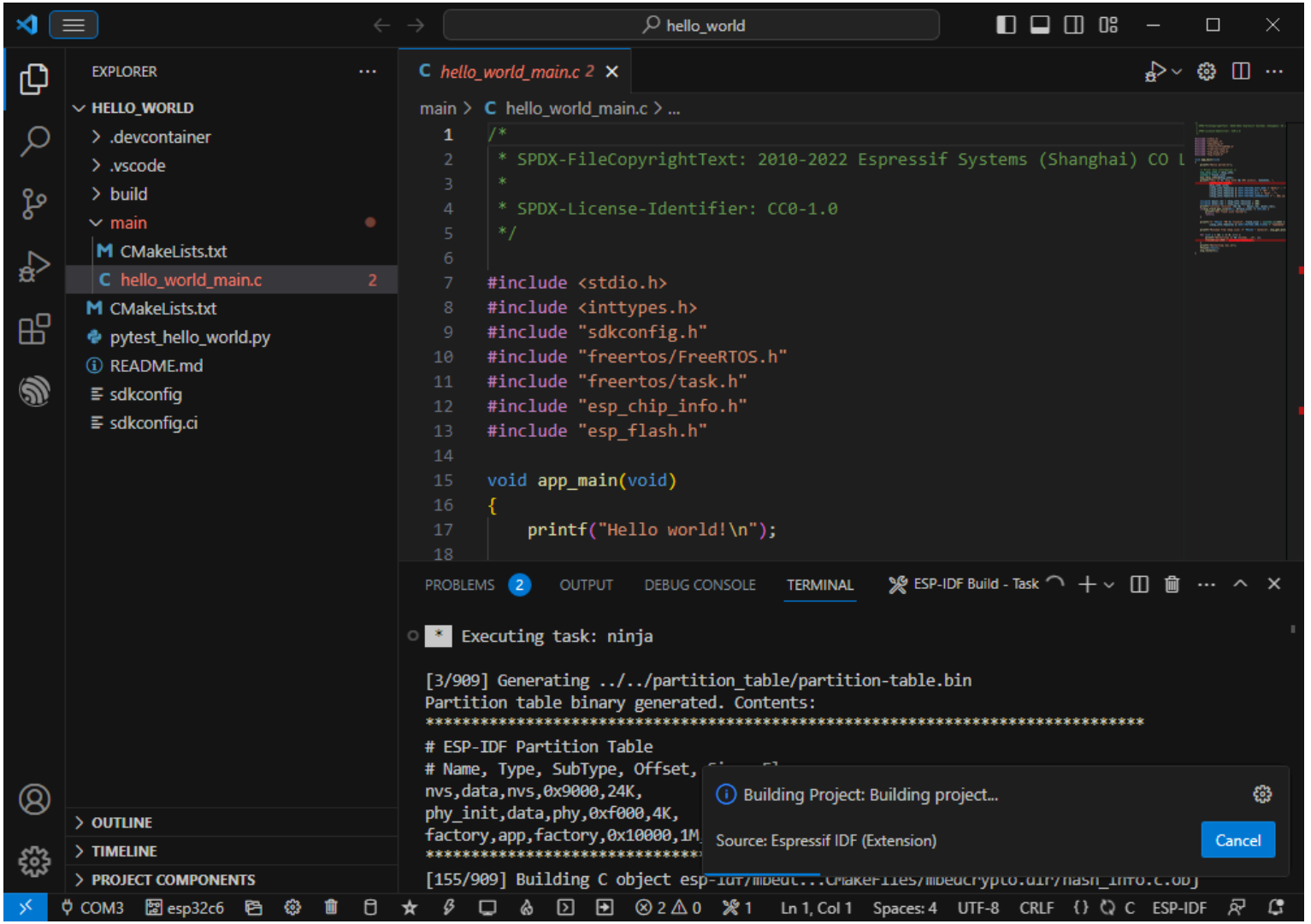
Compile, Program, Serial Port Monitoring

- Click on the all-in-one button we described before to compile, flash and open the serial port monitor.



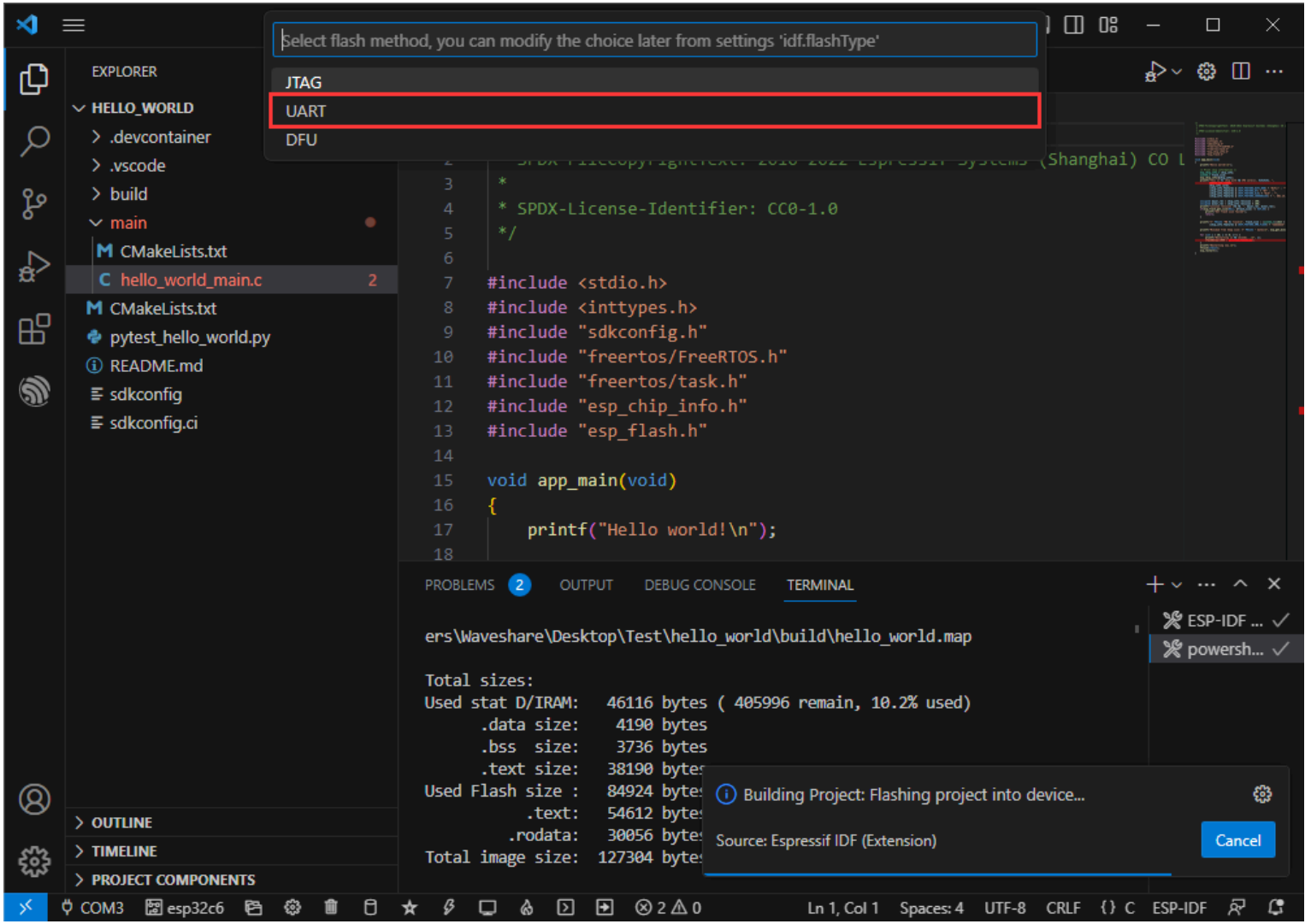
(/wiki/File:ESP32-C6-DEV-KIT-N8-25.png)

- It may take a long time to compile especially for the first time.



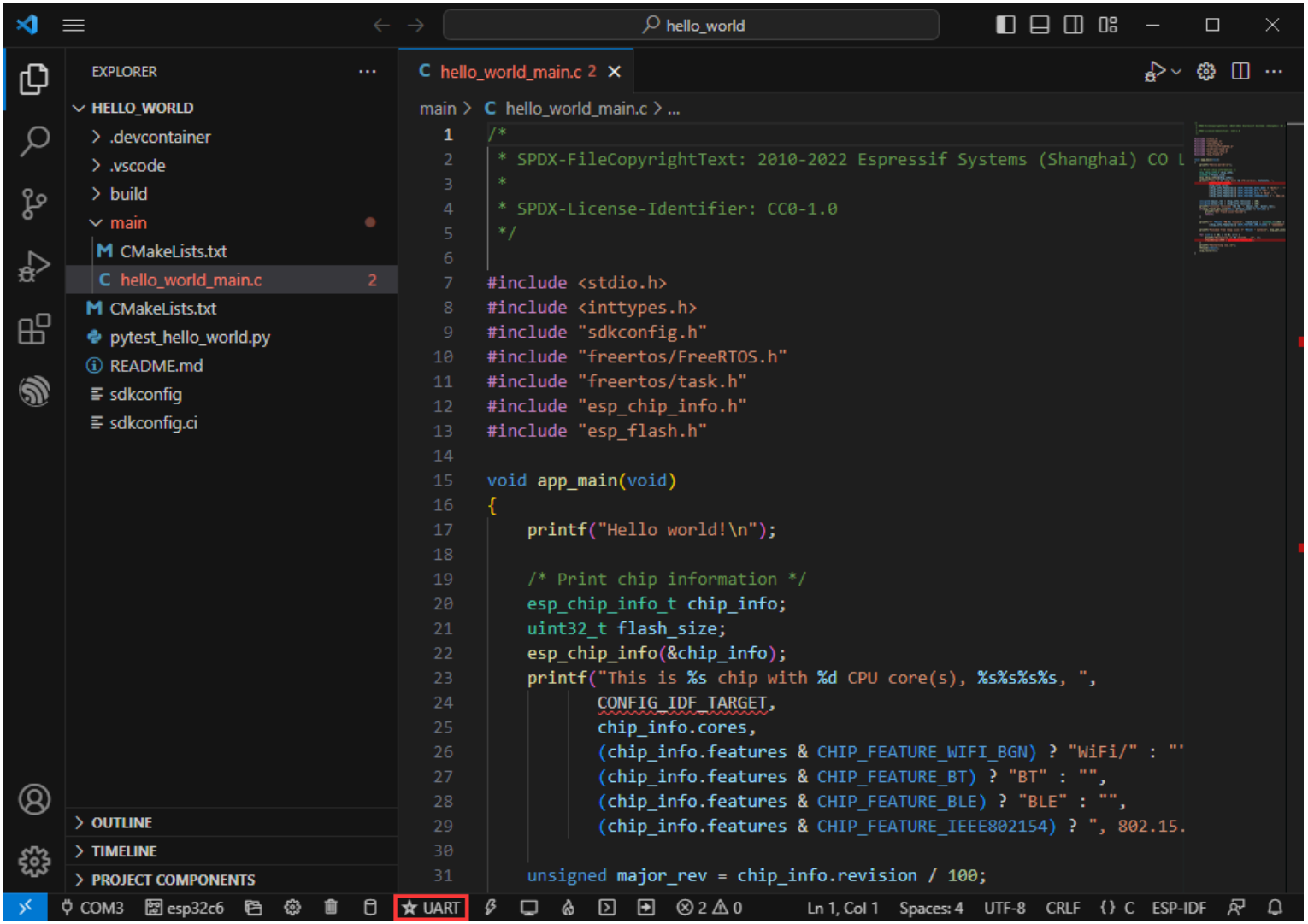
(/wiki/File:ESP32-C6-DEV-KIT-N8-26.png)

- **During this process, the ESP-IDF may take up a lot of CPU resources, so it may cause the system to lag.**
- If it is the first time to flash the program for a new project, you will need to select the download method, and select **UART**.



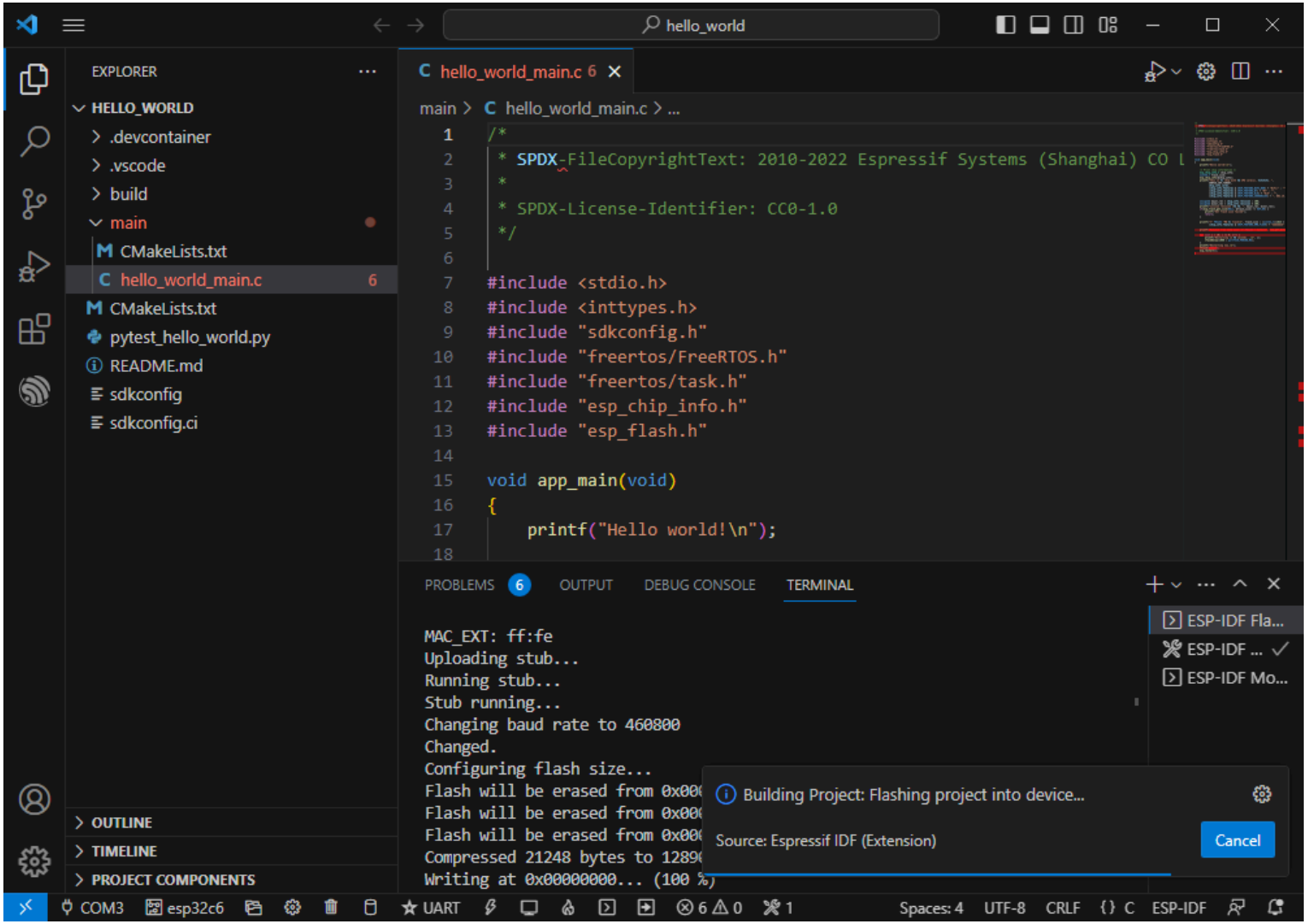
(/wiki/File:ESP32-C6-DEV-KIT-N8-27.png)

- This can also be changed later in the **Download Methods** section (click on it to bring up the options).



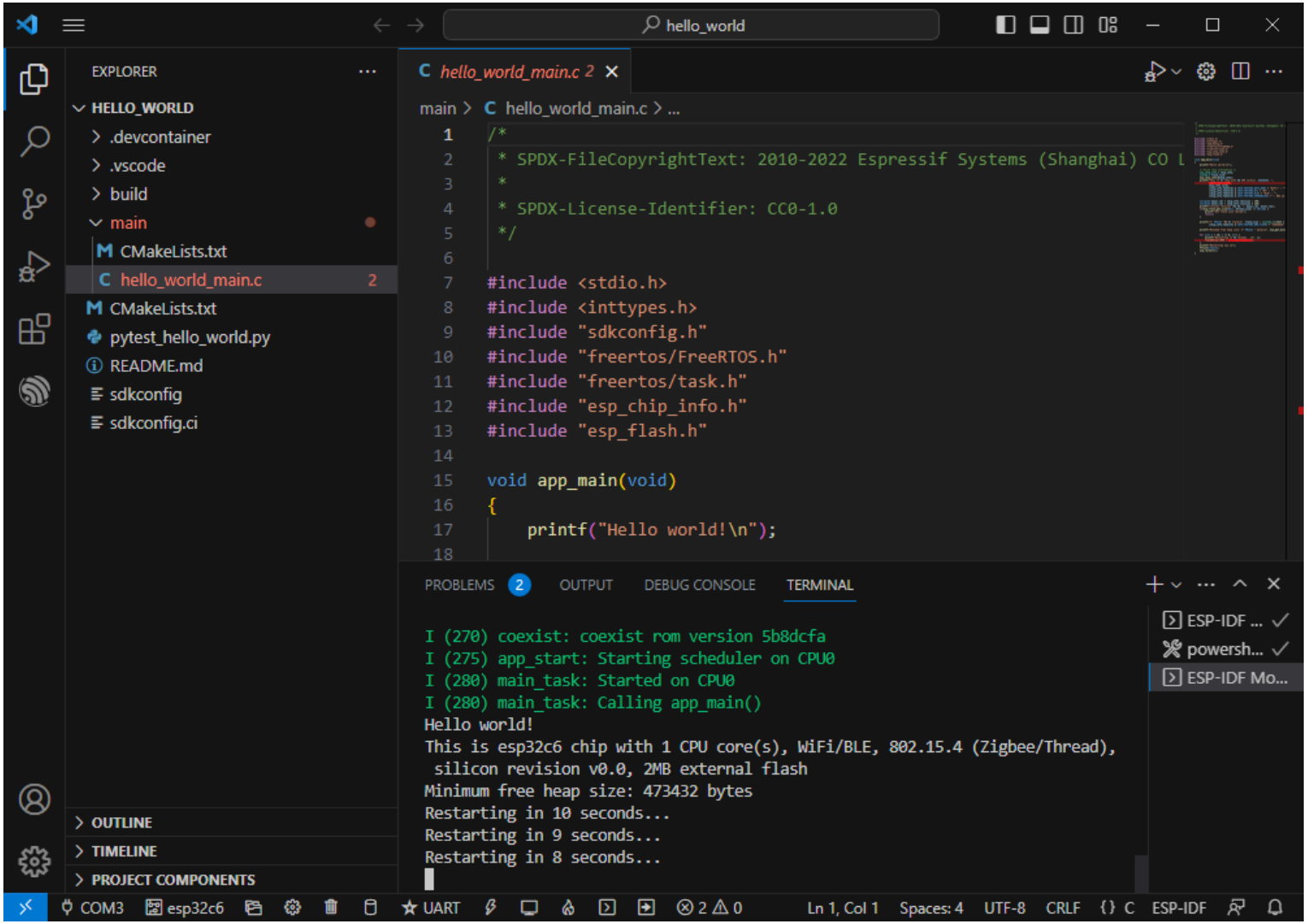
(/wiki/File:ESP32-C6-DEV-KIT-N8-28.png)

- As it comes with the onboard automatic download circuit, there is no need for manual operation to download automatically.



(/wiki/File:ESP32-C6-DEV-KIT-N8-29.png)

- After successful download, automatically enter the serial monitor, you can see the chip output the corresponding information and be prompted to restart after 10S.



(/wiki/File:ESP32-C6-DEV-KIT-N8-30.png)

Demo Example

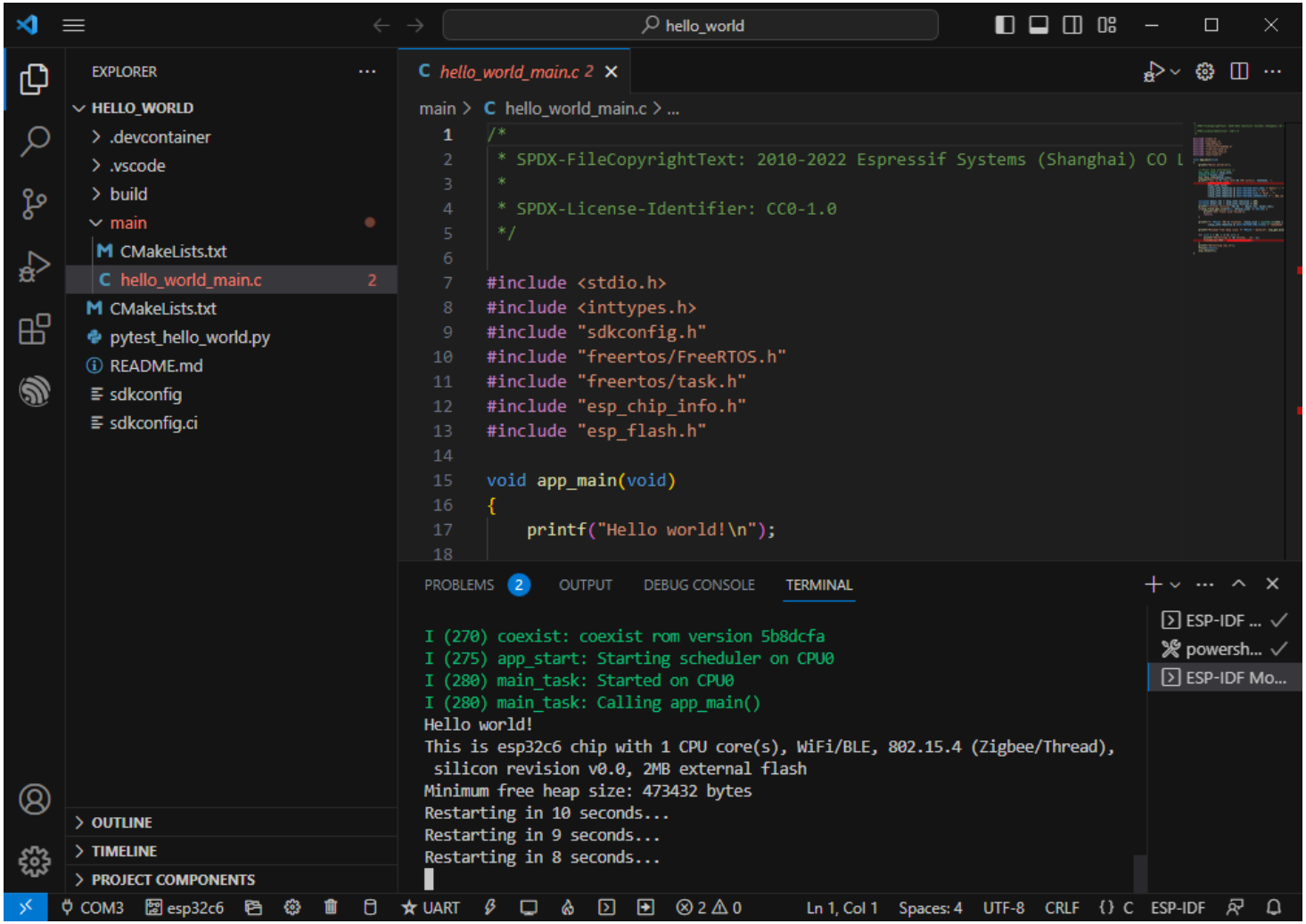
Hello World

The official example path: get-started -> hello_world.

The example effect: Output **Hello World!** on the **TERMINAL** window every 10s.

Software Operation

- Create the official example "hello_world" according to the above tutorial. (Create Example)
- The demo is compatible with ESP32-C6. and you can directly use it with no need to modify the demo.
- Modify the COM port and the driver object (**it is recommended to prioritize the COM port corresponding to the USB, which can be viewed through the device manager**), click on the compile and flash to run the demo.



(/wiki/File:ESP32-C6-DEV-KIT-N8-30.png)

GPIO

Official path: peripherals -> gpio -> generic_gpio.

Sample effect: LED blinks at 1-second intervals.

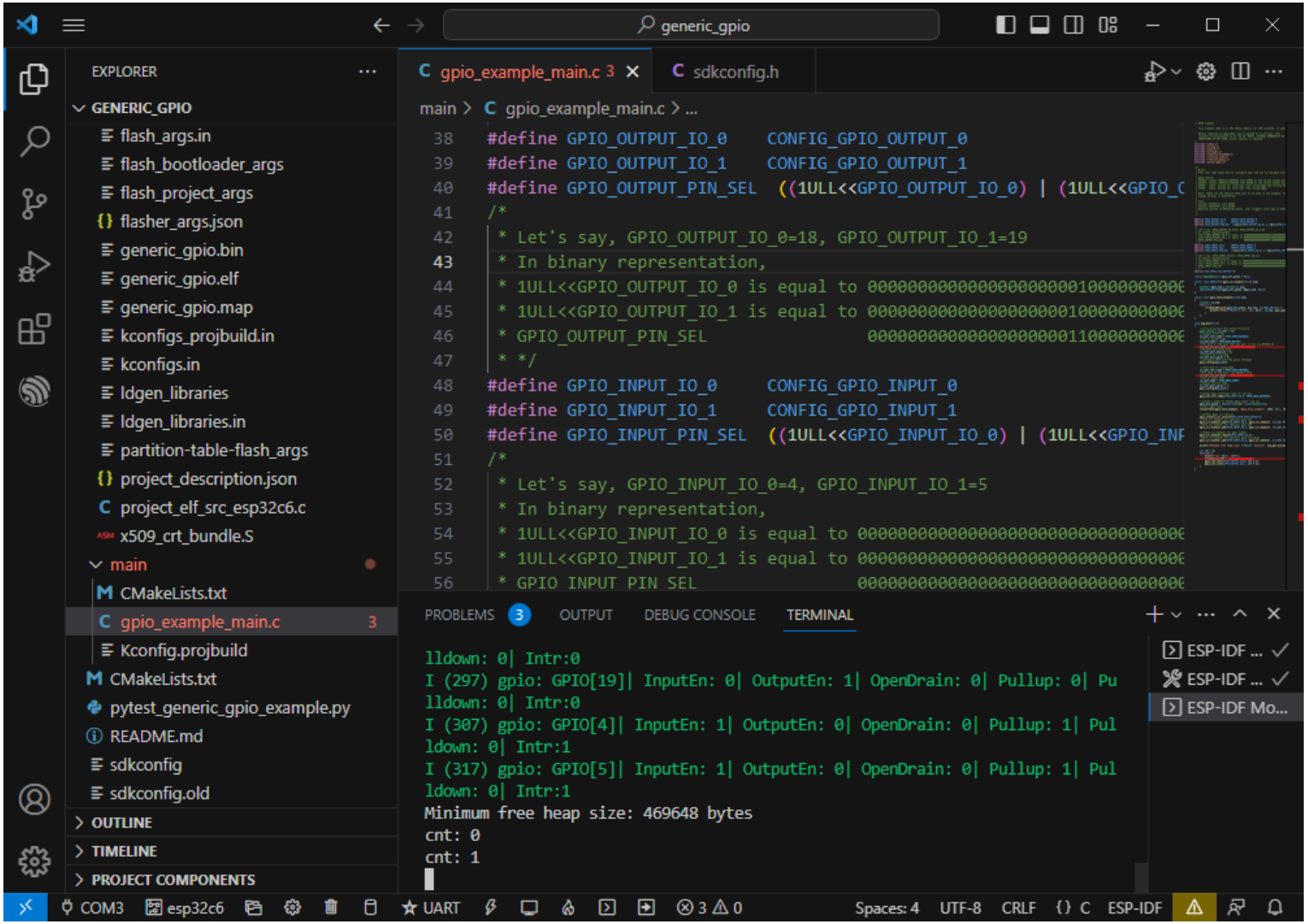
Hardware Connection

ESP32-C6	LED

GPIO18(or GPIO19)	LED+
GND	LED-

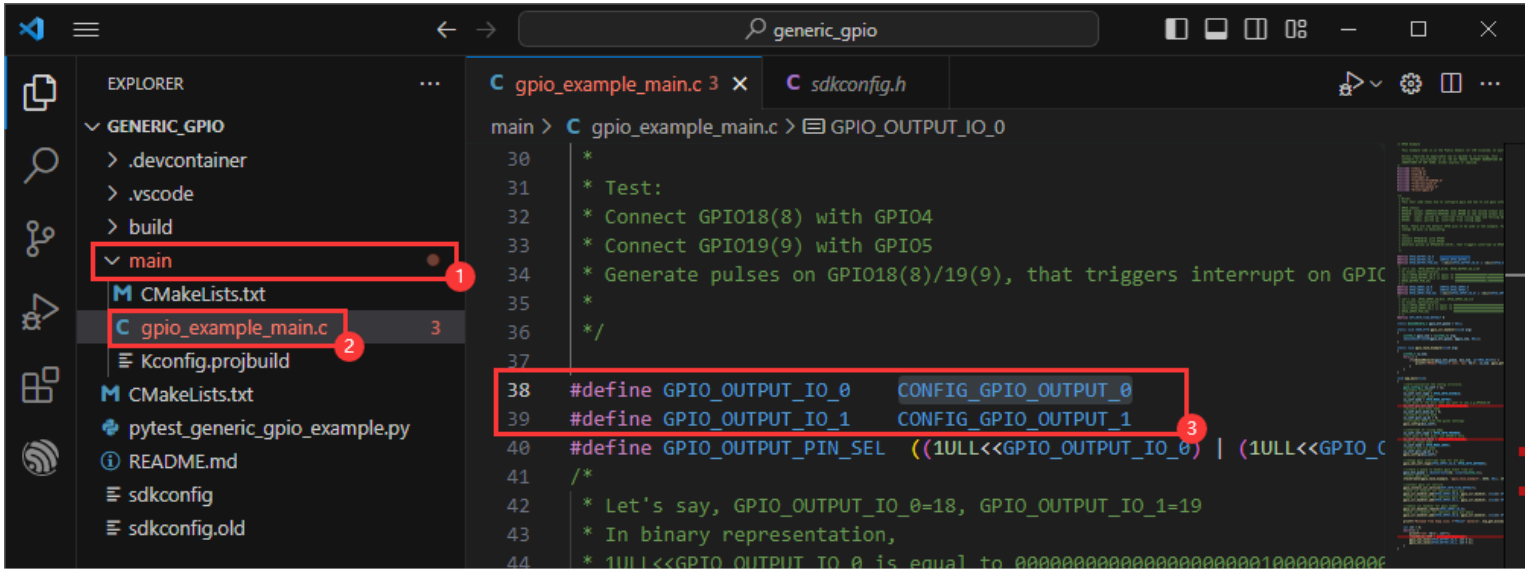
Software Operation

- Follow the tutorial above to create the official example generic_gpio. (Create Example)
- The demo is compatible with ESP32-C6 and can be used without modifying the demo content.
- Modify the COM port and the driver object (**it is recommended to prioritize the COM port corresponding to the USB, which can be viewed through the device manager**), click compile and flash to run the demo.



(/wiki/File:ESP32-C6-DEV-KIT-N8-37.png)

- Go to the demo macro definition location to see what GPIOs are handled.



(/wiki/File:ESP32-C6-DEV-KIT-N8-38.png)

- Right-click and select "Go to Declaration".

(/wiki/File:ESP32-C6-DEV-KIT-N8-39.png)

- The actual GPIOs are GPIO18, GPIO19.

(/wiki/File:ESP32-C6-DEV-KIT-N8-40.png)

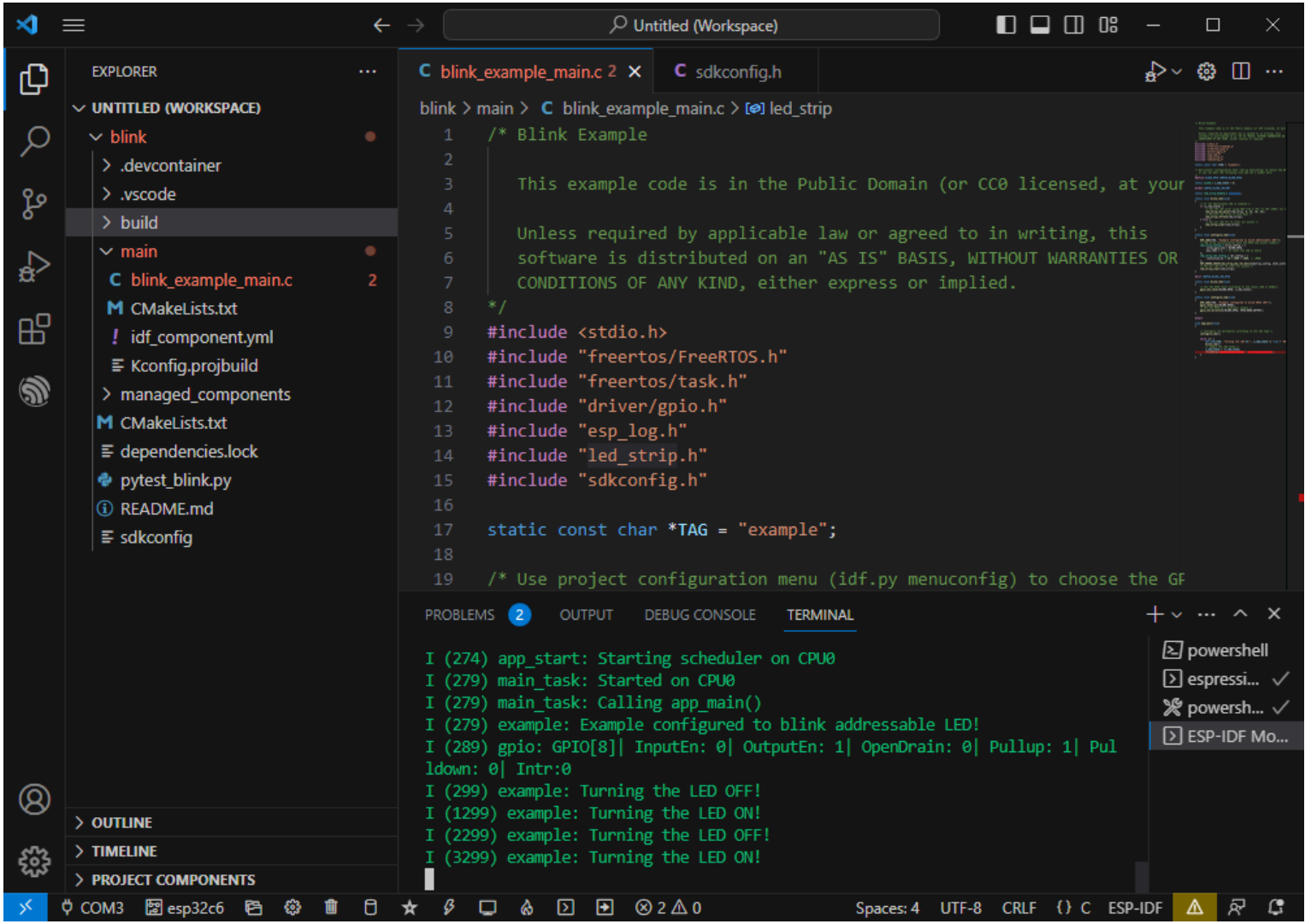
RGB

Official example path: get-started -> blink.

Example effect: onboard RGB beads blink at 1-second interval.

Software Operation

- Follow the tutorial above to create the official example blink. (Create Example)
- The demo is compatible with ESP32-C6 and can be used without modifying the demo content.
- Modify the COM port and the driver object (**it is recommended to prioritize the COM port corresponding to USB, which can be viewed through the device manager**), click compile and flash to run the demo.



(/wiki/File:ESP32-C6-DEV-KIT-N8-41.png)

UART

Official example path: peripherals -> uart-> uart_async_rxtxtasks.

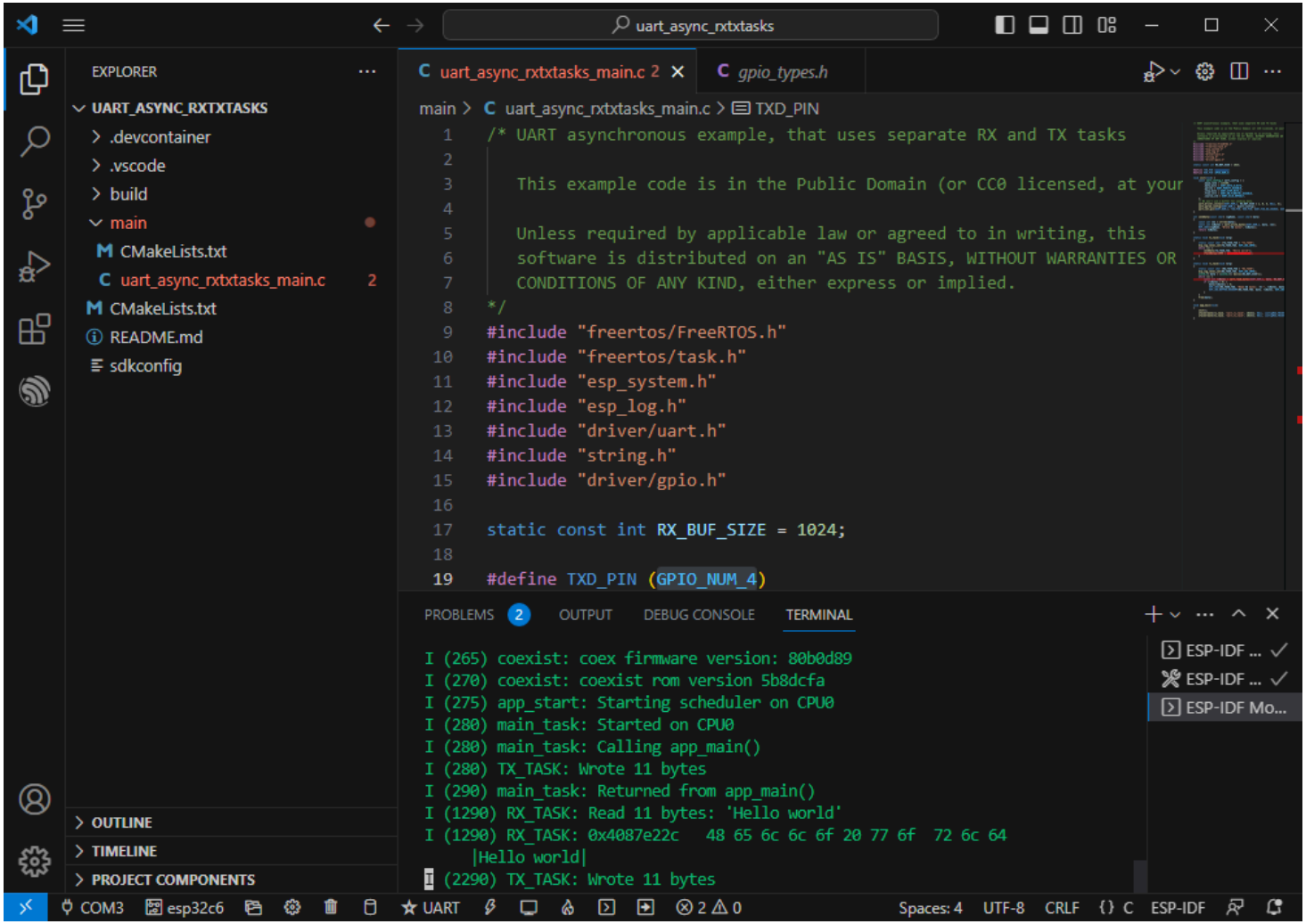
Example effect: shorting GPIO4 and GPIO5 to send/receive UART data.

Hardware Connection

ESP32-C6	ESP32-C6 (the same one)
GPIO4	GPIO5

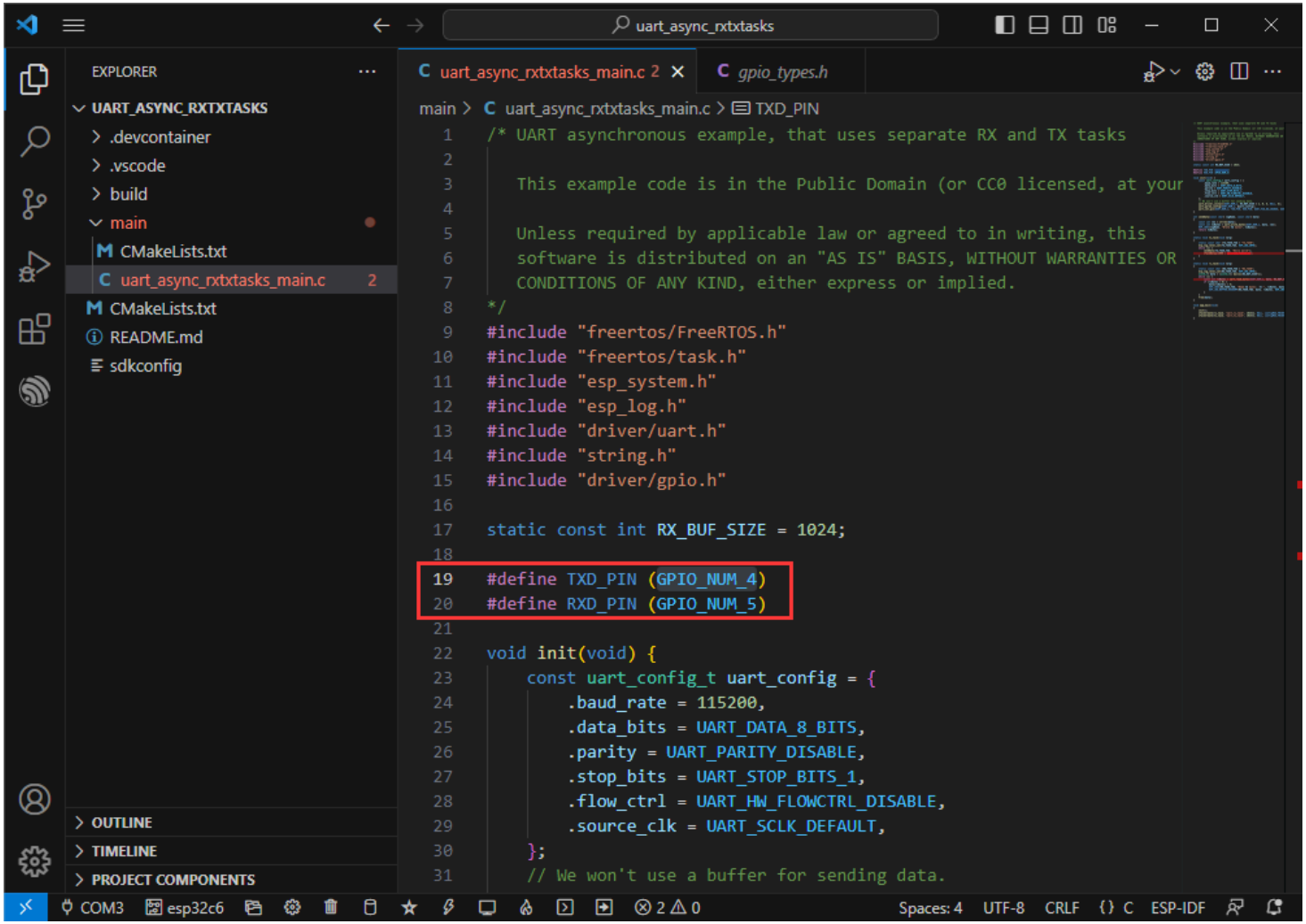
Software Operation

- Create the official example uart_async_rxtxtasks according to the tutorial above. (Create Example (https://www.waveshare.com/wiki/ESP32-C6-DEV-KIT-N8#Official_Demo_Usage_GUIDE))
- The demo is compatible with ESP32-C6 and can be used without modifying the demo content.
- Modify the COM port and driver object (**it is recommended to prioritize the COM port corresponding to USB, which can be viewed through the device manager**), click compile and flash to run the demo.



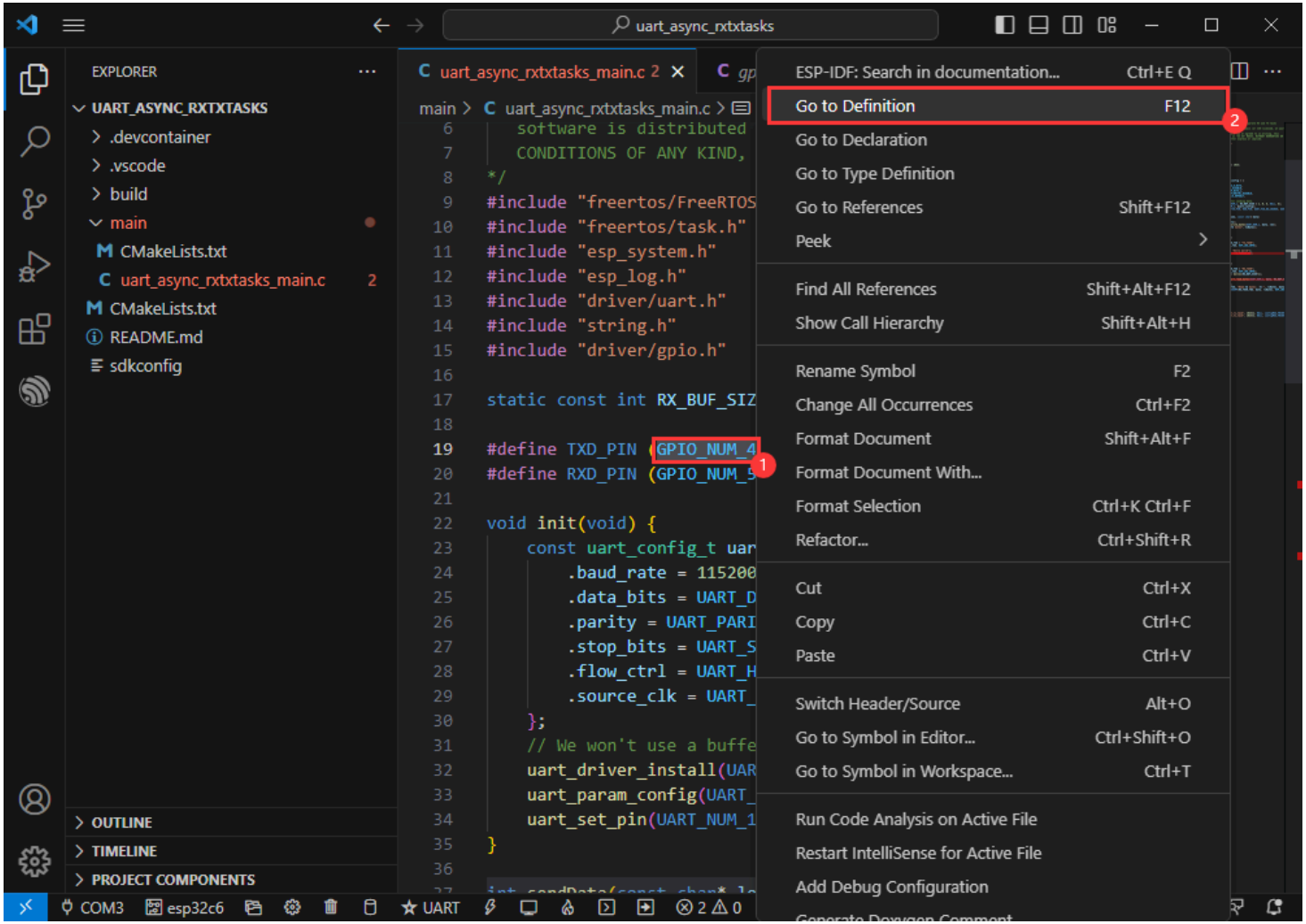
(/wiki/File:ESP32-C6-DEV-KIT-N8-77.png)

- Hardware connection according to the GPIO used.



(/wiki/File:ESP32-C6-DEV-KIT-N8-78.png)

- You can go to the definition file to see the actual GPIOs used (check **GPIO_NUM_4** -> Right click -> **Go to Definition**).



(/wiki/File:ESP32-C6-DEV-KIT-N8-79.png)

I2C

The official example path: peripherals -> lcd-> i2c_oled.

Example effect: turns on the 0.96-inch OLED (A) (https://www.waveshare.com/0.96inch-oled-a.

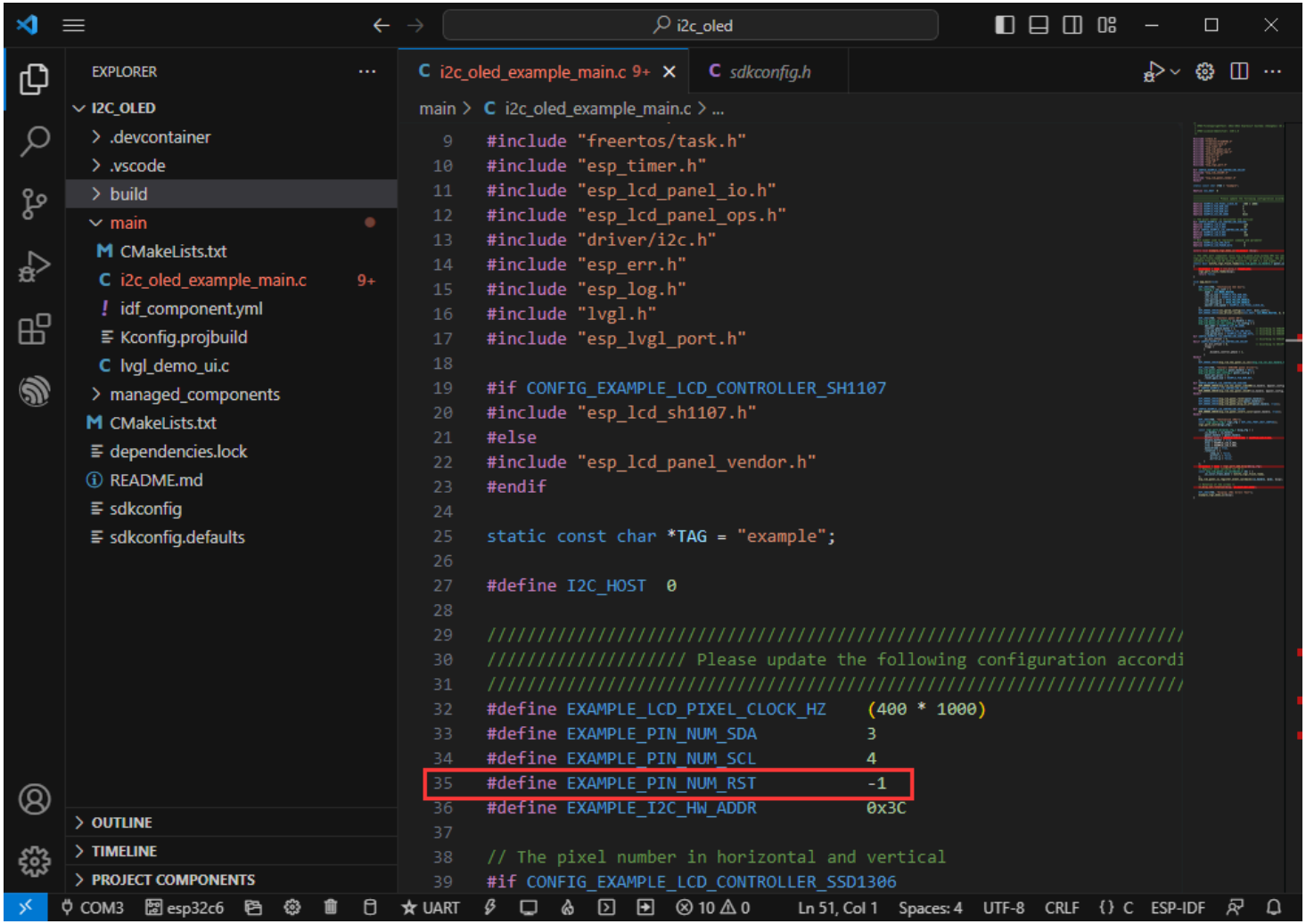
htm) and displays a character.

Hardware Connection

0.96inch OLED (A)	ESP32-C6
VCC	3V3
GND	GND
DIN	GPIO3
CLK	GPIO4
CS	GND
D/C	GND
RES	GPIO9

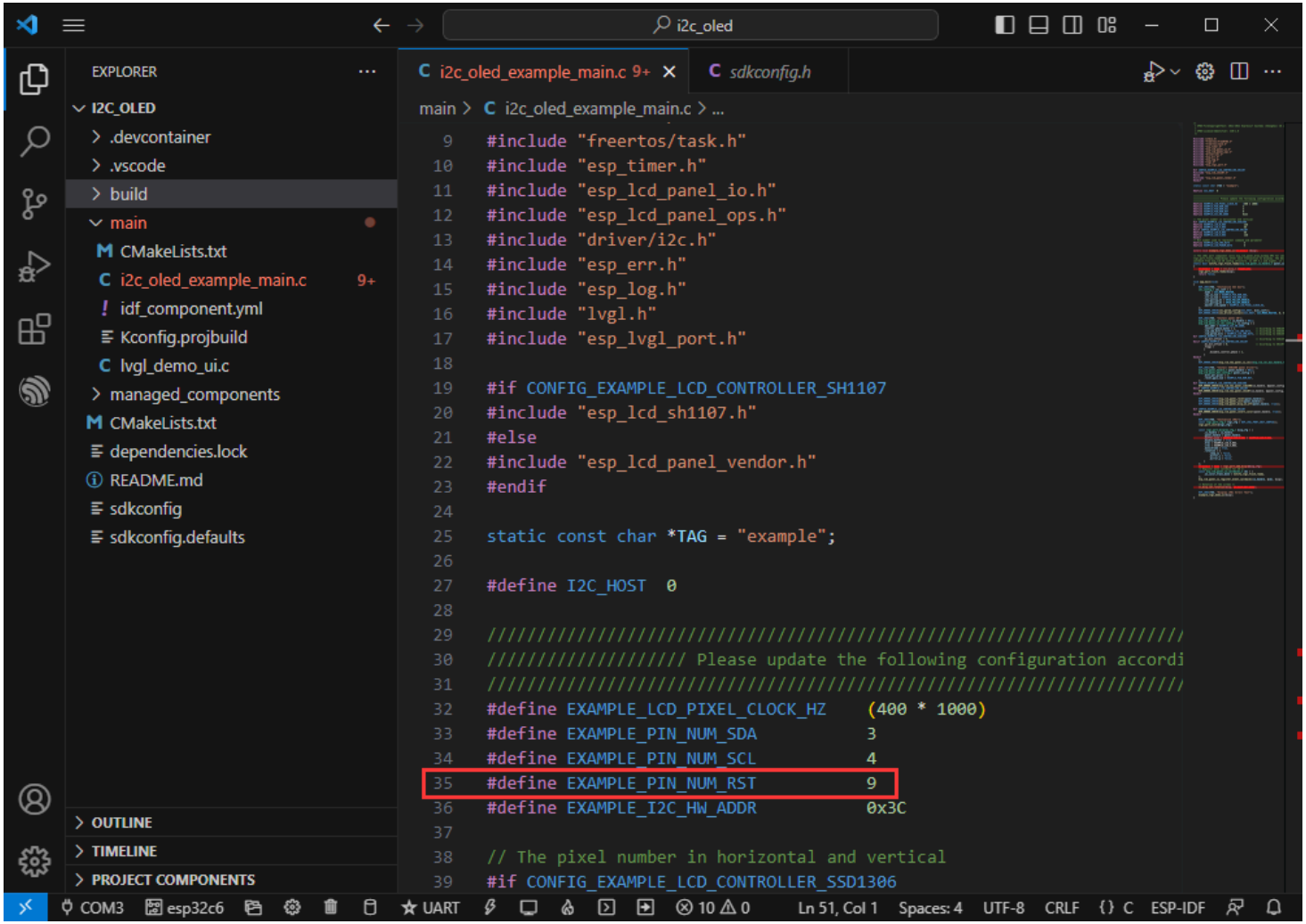
Software Operation

- Create the official example i2c_oled according to the tutorial above. (Create Example (https://www.waveshare.com/wiki/ESP32-C6-DEV-KIT-N8#Official_Demo_Usage_GUIDE))
- Modify the demo to be compatible with 0.96-inch OLED (A) (https://www.waveshare.com/0.96inch-oled-a.htm).



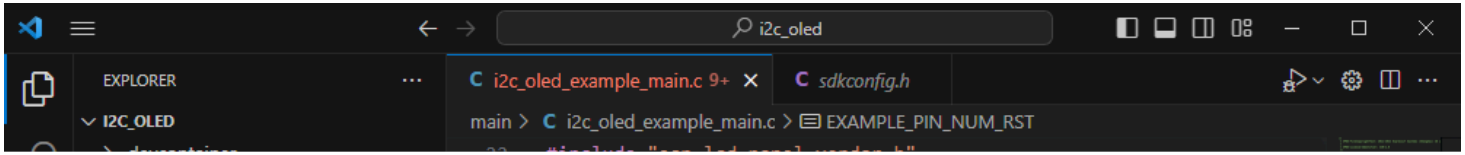
(/wiki/File:ESP32-C6-DEV-KIT-N8-31.png)

- Adapts to 0.96-inch OLED (A), defines RES pin as GPIO9.



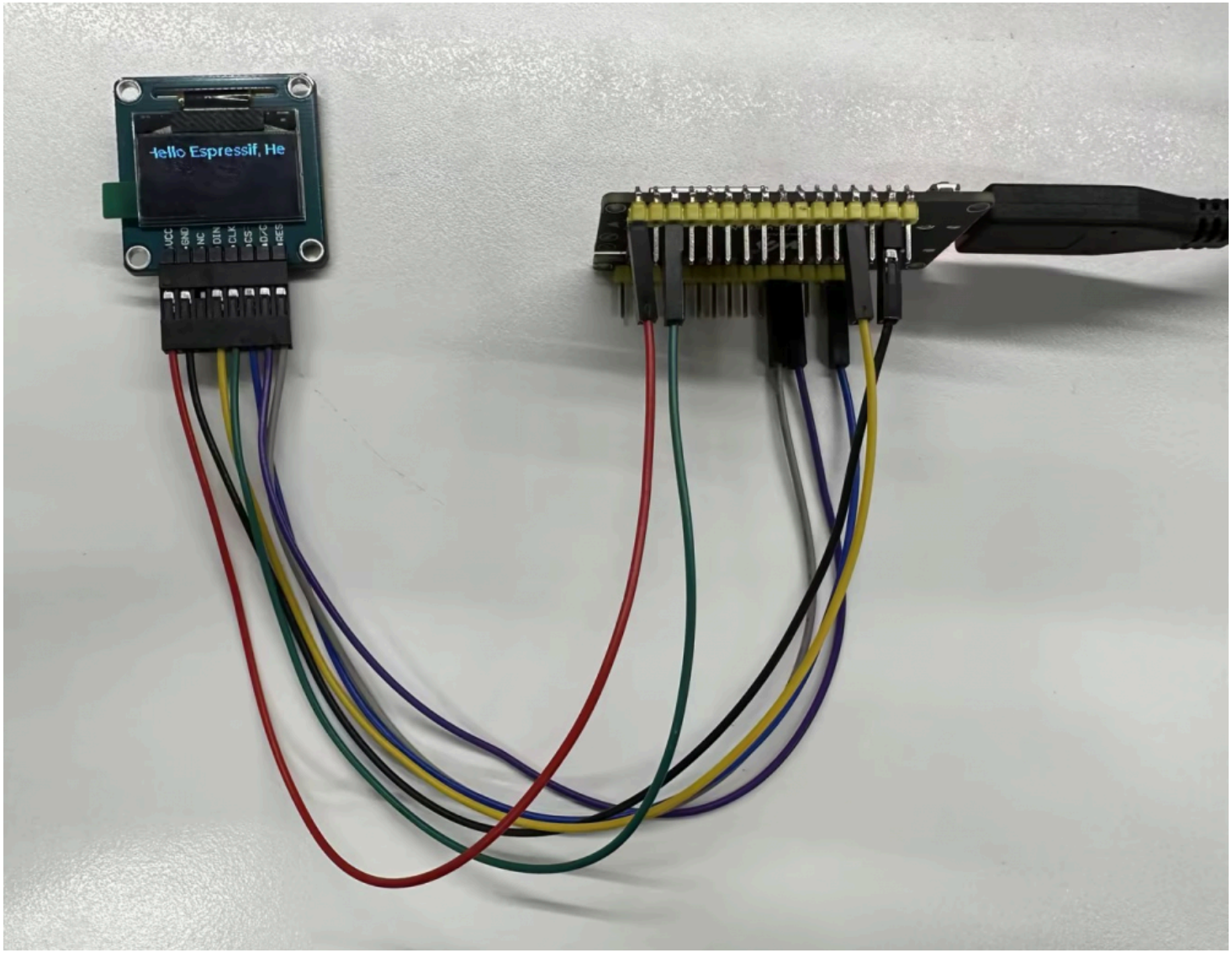
(/wiki/File:ESP32-C6-DEV-KIT-N8-32.png)

- Modify the COM port and the driver object (**it is recommended to prioritize the COM port corresponding to USB, which can be viewed through the device manager**), click compile and flash to run the demo.



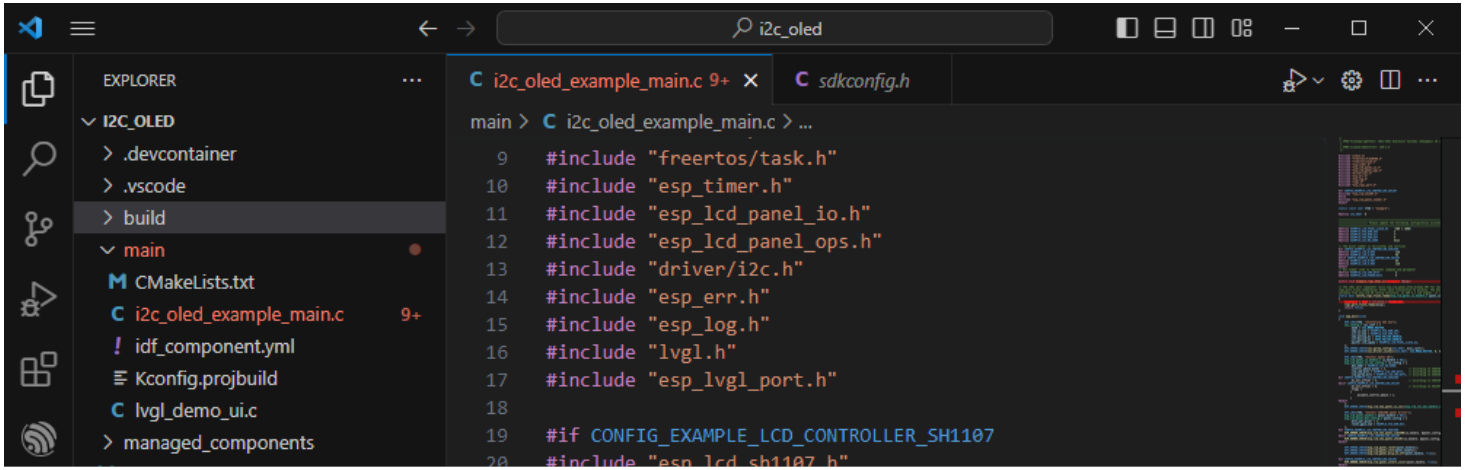
(/wiki/File:ESP32-C6-DEV-KIT-N8-35.png)

- The effect is as shown below:



(/wiki/File:ESP32-C6-DEV-KIT-N8-33.png)

- You can view the actual use of GPIO:



(/wiki/File:ESP32-C6-DEV-KIT-N8-34.png)

SPI

The path of the official example: peripherals -> spi_master-> lcd.

The example effect: dynamic displaying picture effect on the 2.4inch LCD Module (https://ww

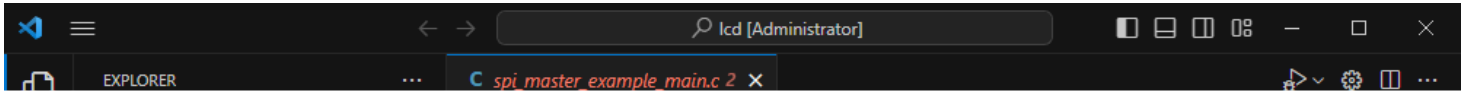
w.waveshare.com/2.4inch-lcd-module.htm).

Hardware Connection

2.4inch LCD Module	ESP32-C6
VCC	3V3
GND	GND
DIN	GPIO7
CLK	GPIO6
CS	GPIO10
D/C	GPIO11
RES	GPIO4
BL	GPIO5

Software Operation

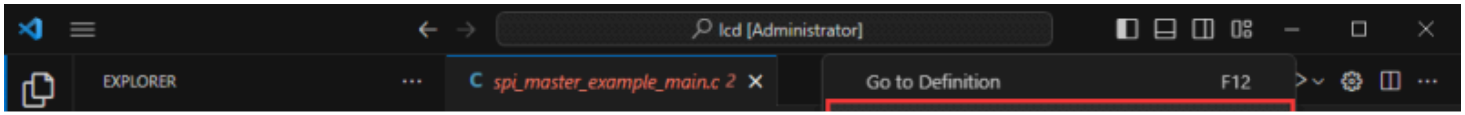
- Right-click on the VScode icon and run VScode as administrator:



(/wiki/File:ESP32-C6_TO_Sample_15.png)

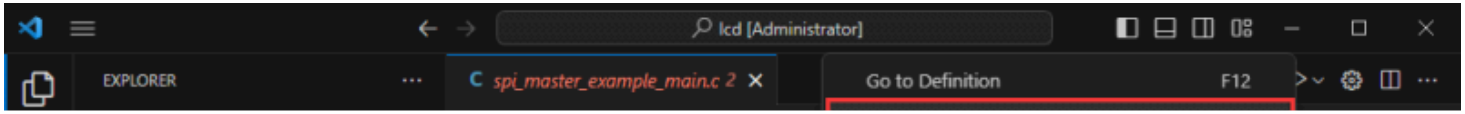
- Create the official example lcd according to the tutorial above. (Create Example (https://www.waveshare.com/wiki/ESP32-C6-DEV-KIT-N8#Official_Demo_Usage_GUIDE))

Modify the program to make it compatible with a 2.4inch LCD Module (https://www.waveshare.com/2.4inch-lcd-module.htm).



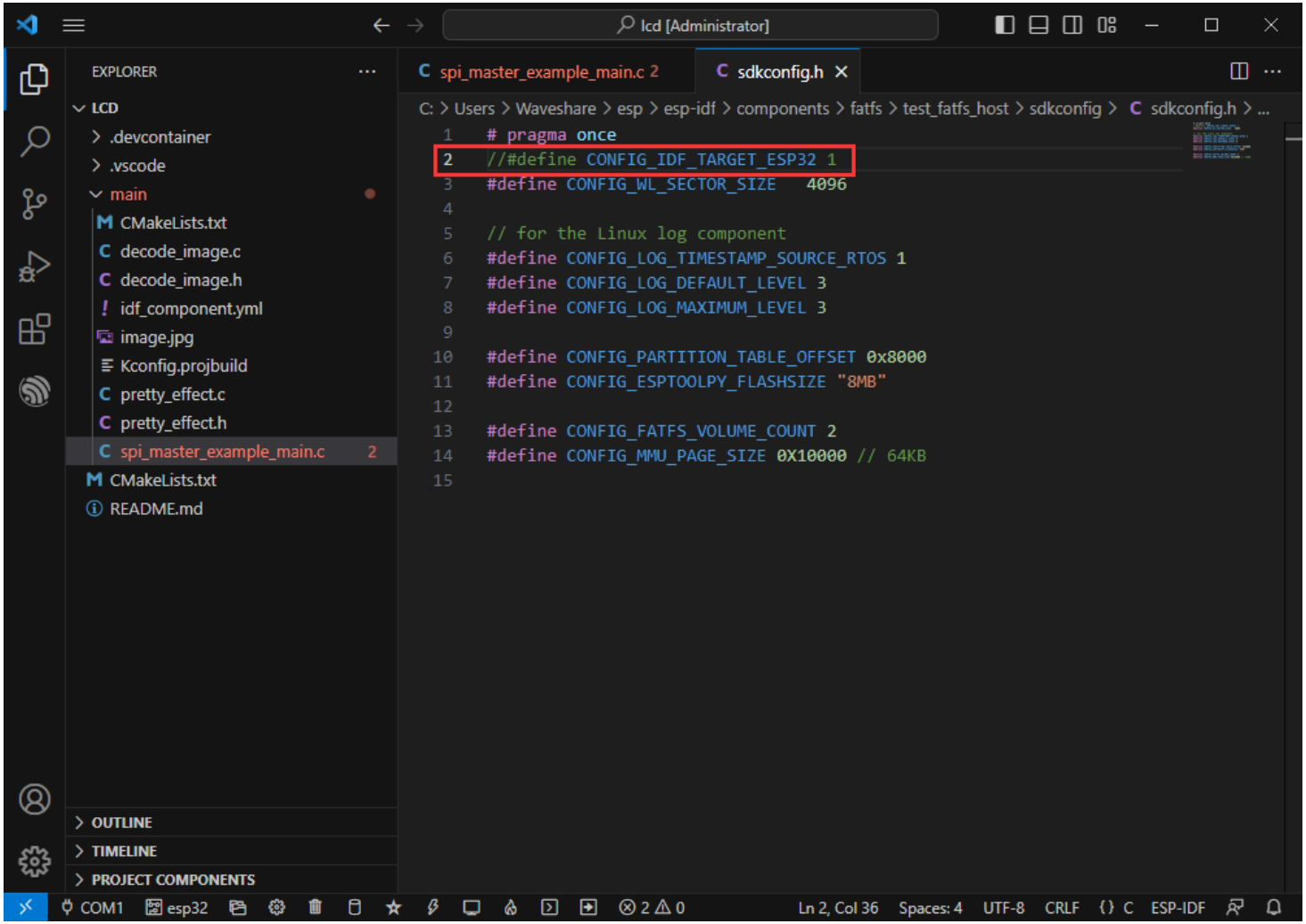
(/wiki/File:ESP32-C6-DEV-KIT-N8-36.png)

- Select "Go to Declaration".



(/wiki/File:ESP32-C6-DEV-KIT-N8-36.png)

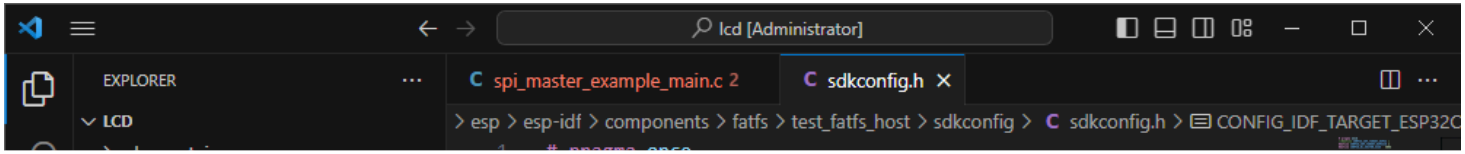
- Note that currently using "ESP32-C6", and then blocks other chip definitions.



(/wiki/File:ESP32-C6-DEV-KIT-N8-036.png)

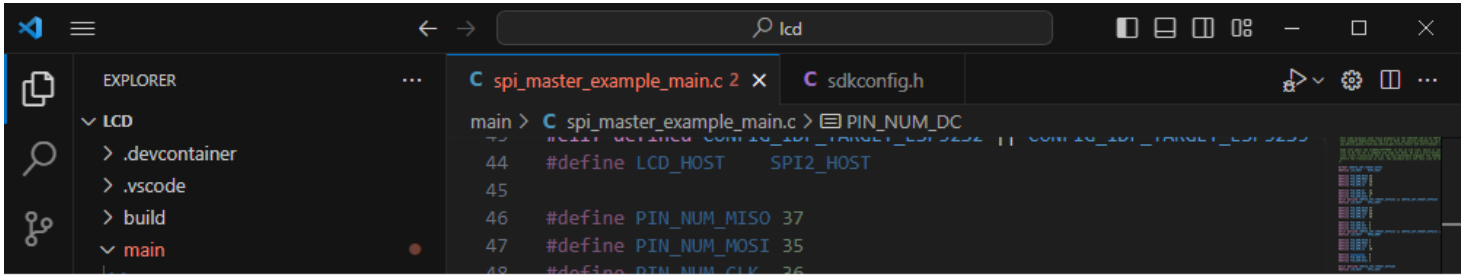
- And add macro-define ESP32-C6, **CONFIG_IDF_TARGET_ESP32C6**.

```
//#define CONFIG_IDF_TARGET_ESP32 1
#define CONFIG_IDF_TARGET_ESP32C6 1
```



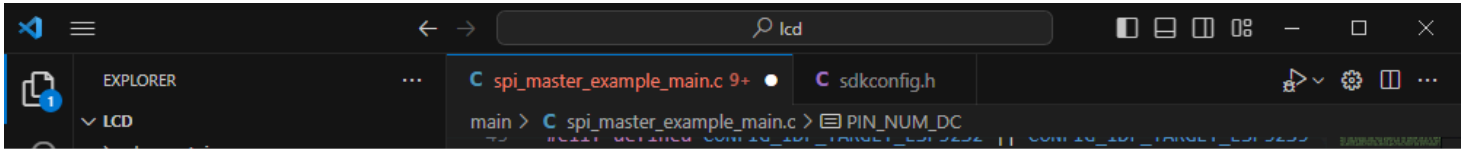
(/wiki/File:ESP32-C6-DEV-KIT-N8-42.png)

- Modify D/C using IO.
  - Enter the 62 line **spi_master_example_main.c**.



(/wiki/File:ESP32-C6-DEV-KIT-N8-43.png)

- Modify D/C using IO to avoid the download circuit (using GPIO11 instead of GPIO9).

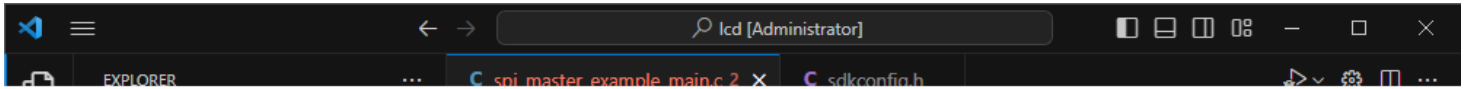


(/wiki/File:ESP32-C6-DEV-KIT-N8-44.png)

- Modify the backlight.

(/wiki/File:ESP32-C6-DEV-KIT-N8-45.png)

- Modify it as **gpio_set_level(PIN_NUM_BCKL, 1);**



(/wiki/File:ESP32-C6-DEV-KIT-N8-46.png)

- Modify the COM port and the driver object (**it is recommended to prioritize the COM port corresponding to USB, which can be viewed through the device manager**), click compile and flash to run the demo.

(/wiki/File:ESP32-C6-DEV-KIT-N8-47.png)

- The effect is as shown below:

(/wiki/File:ESP32-C6-DEV-KIT-N8-48.png)

Bluetooth

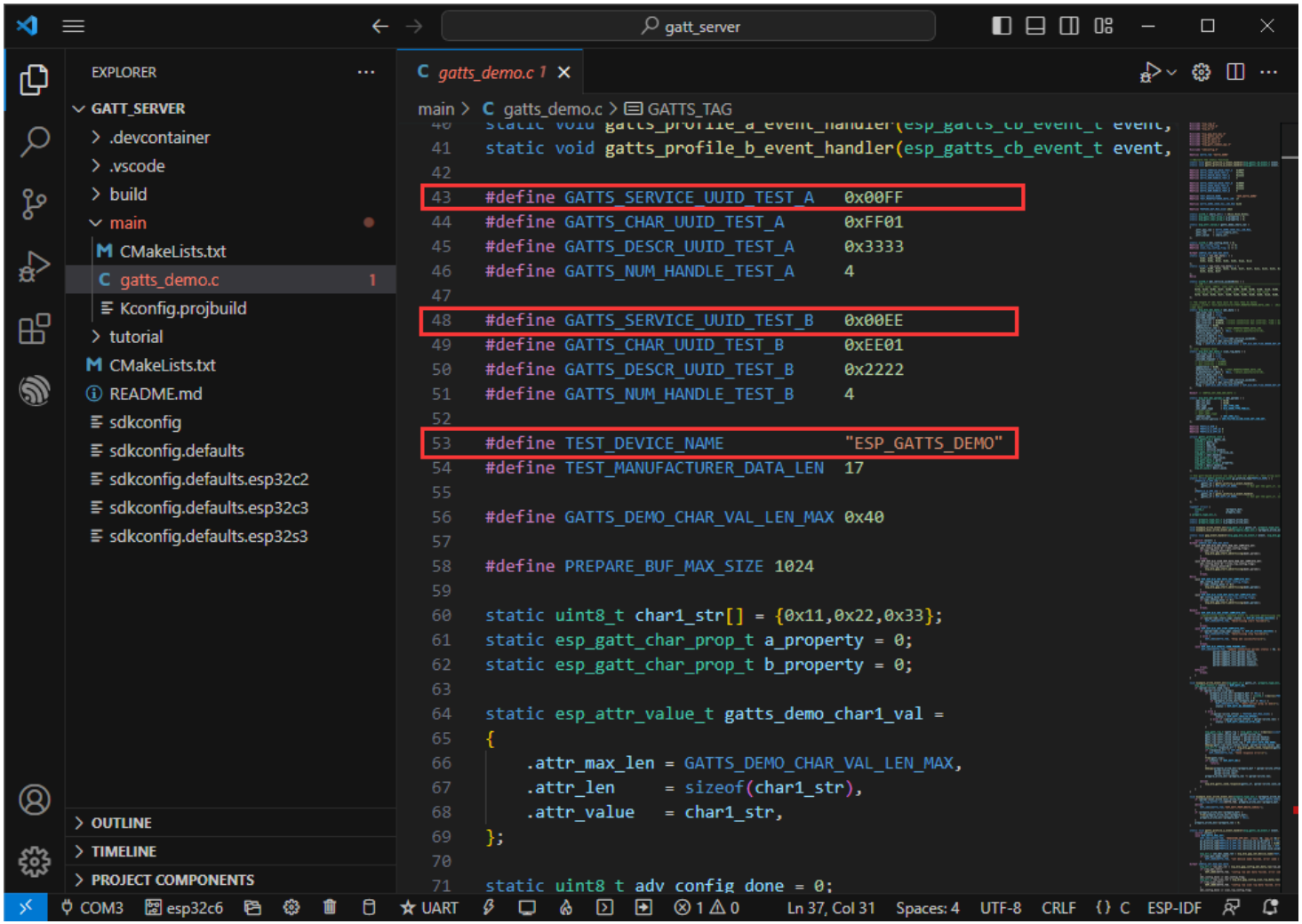
Official sample path: Bluetooth -> bluedroid -> ble -> gatt_server.

Example effect: ESP32-C6 and cell phone Bluetooth debugging assistant for data transmission.

Software Operation

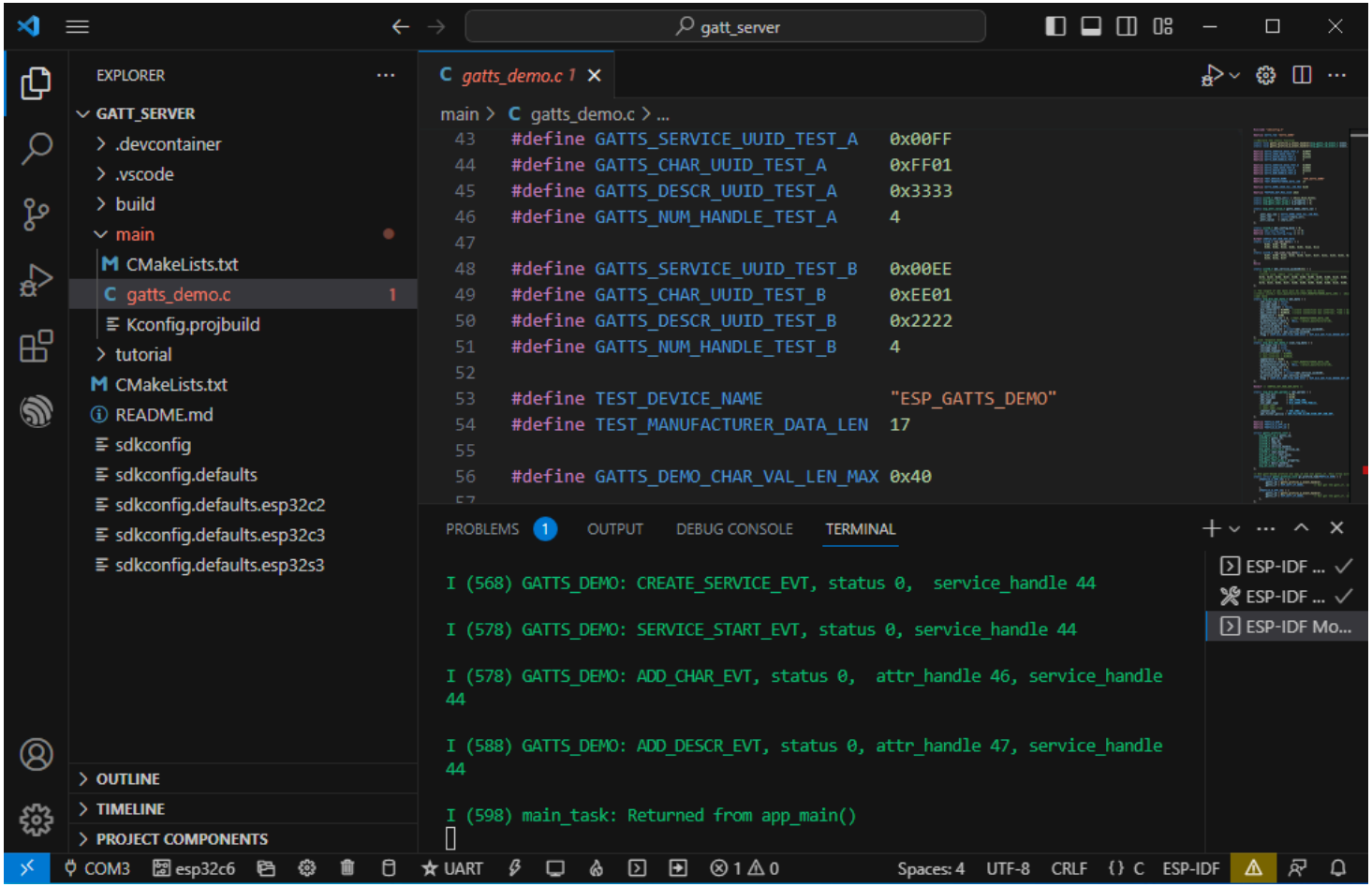
- Install the Bluetooth debugging assistant (https://files.waveshare.com/wiki/ESP32-C6-DEV-KIT-N8/ESP32-C6_TO_BLEAssist.ZIP) on your phone.
- Follow the tutorial above to create the official example gatt_server. (Create Example (https://www.waveshare.com/wiki/ESP32-C6-DEV-KIT-N8#Official_Demo_Usage_GUIDE))

- The demo is compatible with ESP32-C6 and can be used without modifying the demo content.
- Bluetooth name and UUID, Bluetooth name is **ESP_GATTS_DEMO**.



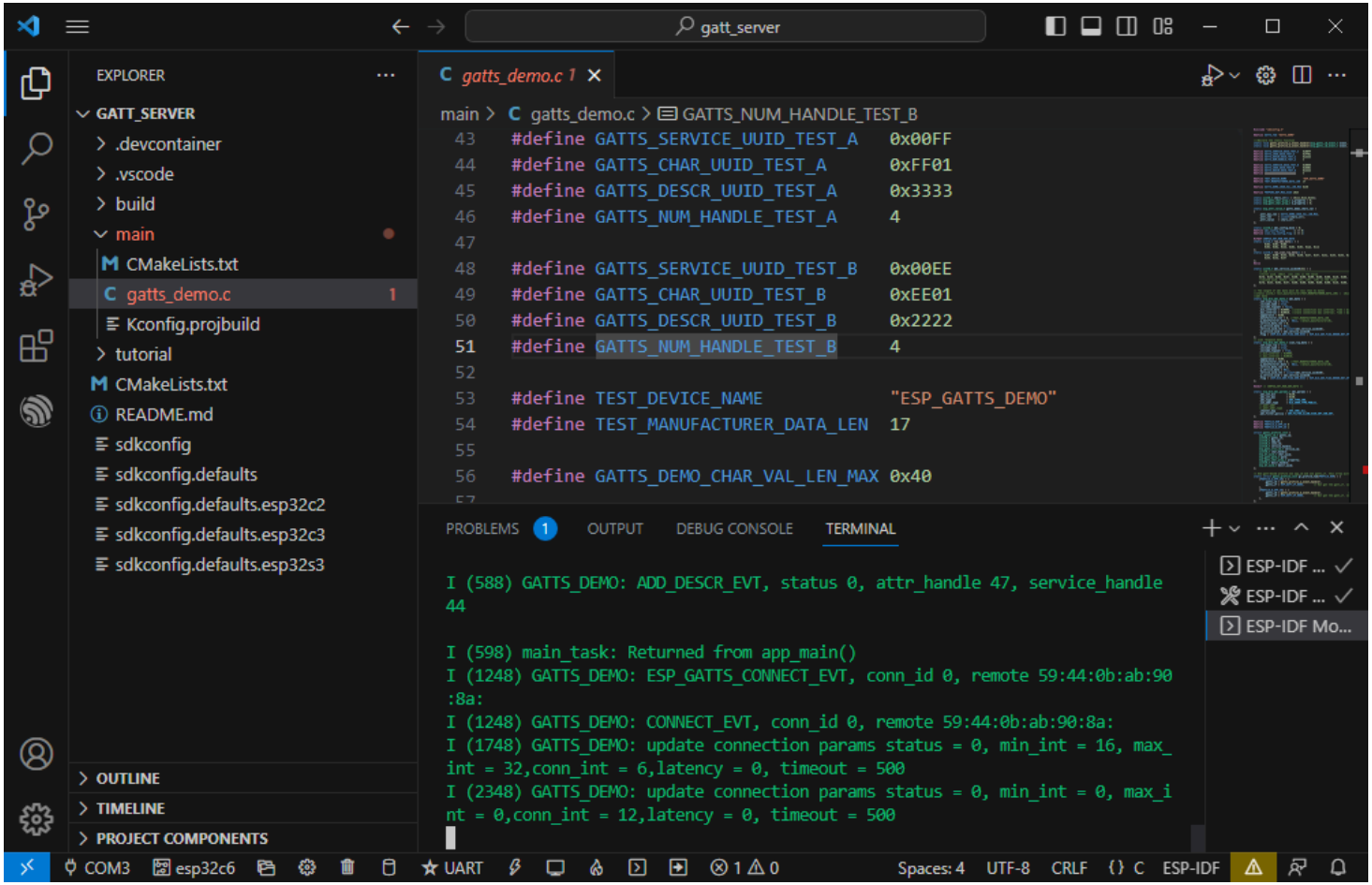
(/wiki/File:ESP32-C6-DEV-KIT-N8-50.png)

- Modify the COM port and the driver object (**it is recommended to prioritize the COM port corresponding to USB, which can be viewed through the device manager**), click compile and flash to run the demo.



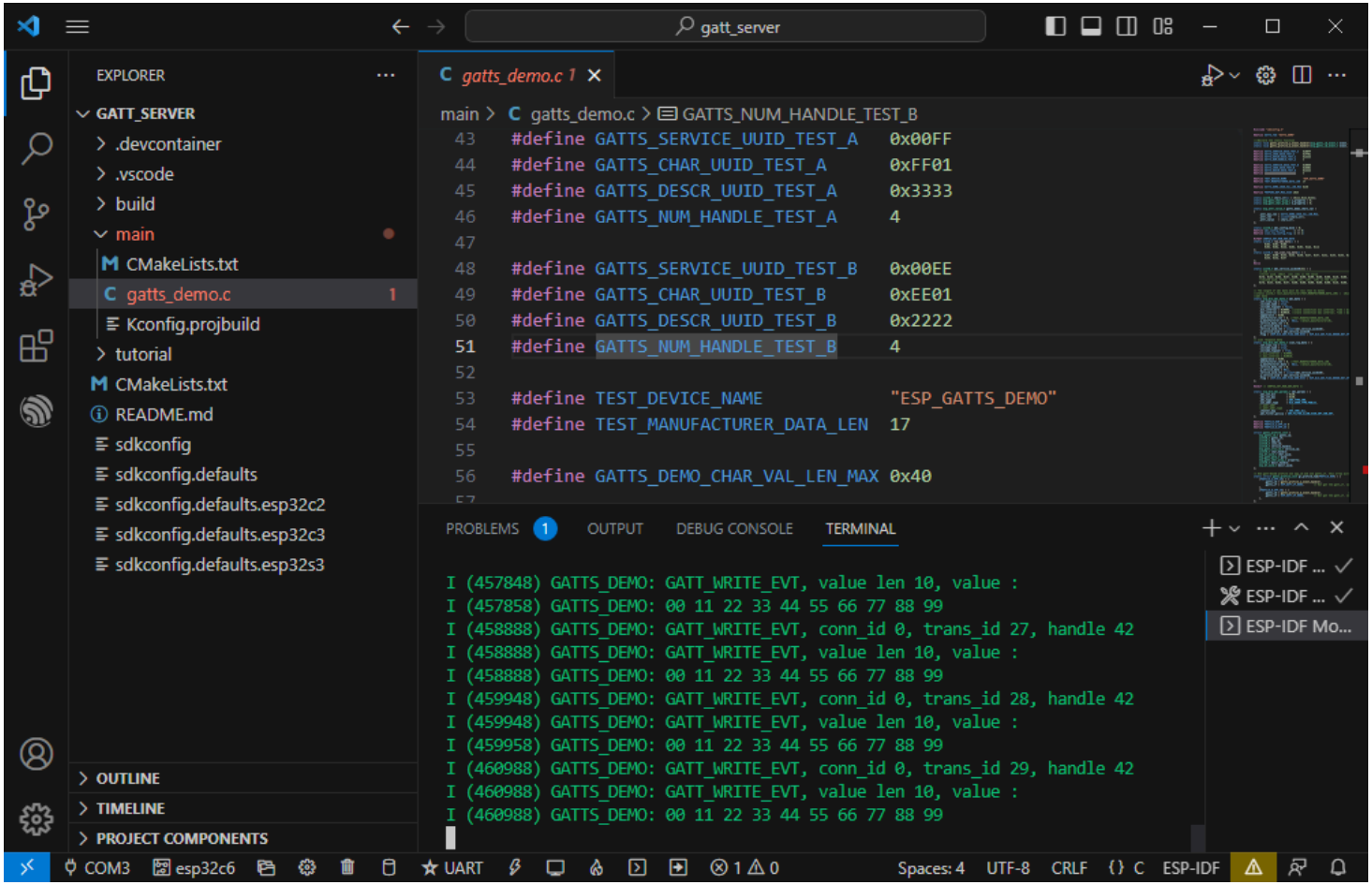
(/wiki/File:ESP32-C6-DEV-KIT-N8-51.png)

- Connect the ESP_GATTS_DEMO Bluetooth device on the phone.
- The effect of a successful connection is shown below:



(/wiki/File:ESP32-C6-DEV-KIT-N8-52.png)

- Based on the UUID value in the demo, select one of the two servers for upstream transmission.
- The ESP32-C6 receives data:



(/wiki/File:ESP32-C6-DEV-KIT-N8-54.png)

WIFI

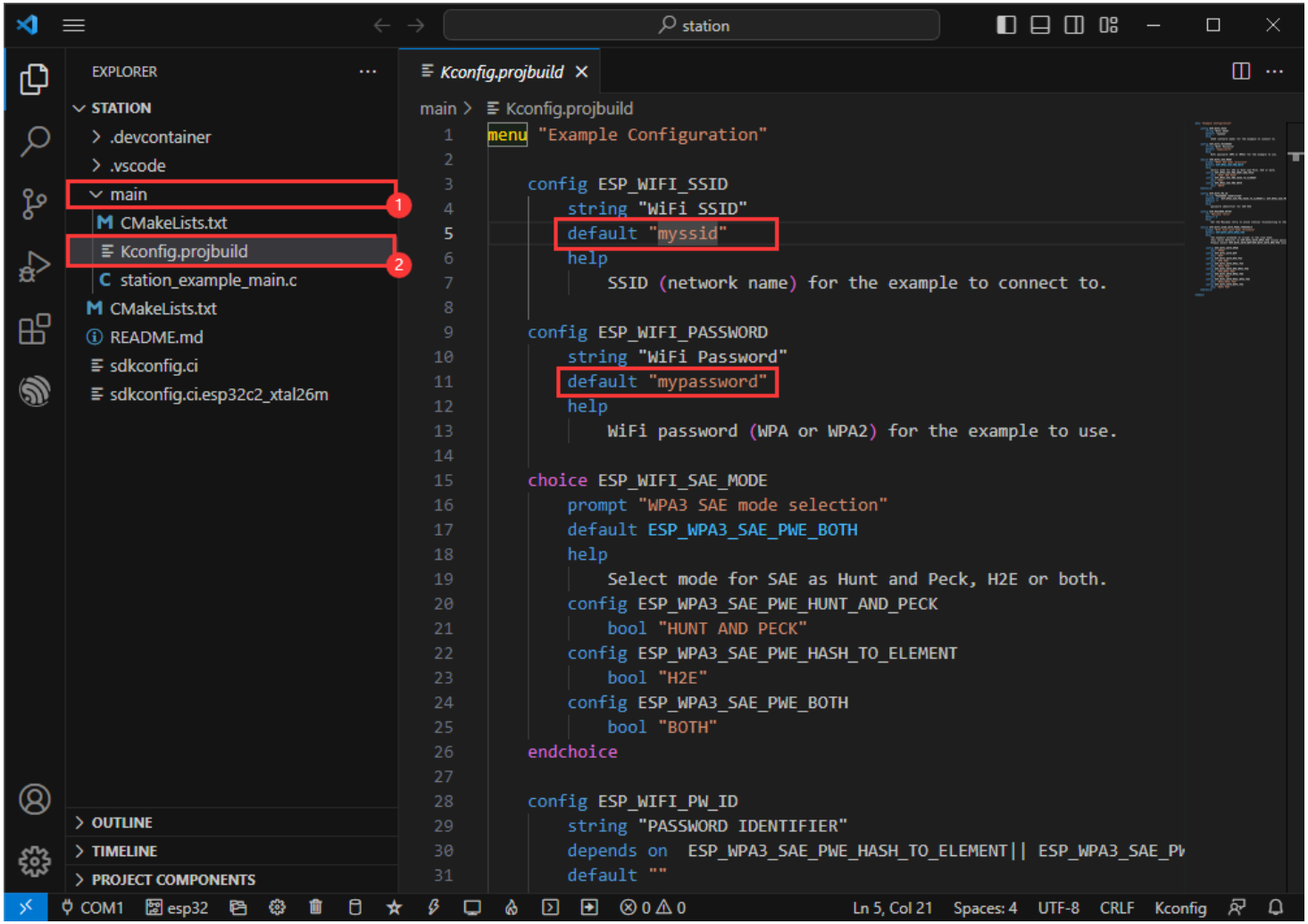
Official example path: wifi-> getting_started-> station.

Example effect: ESP32-C6 connects to WIFI.

Software Operation

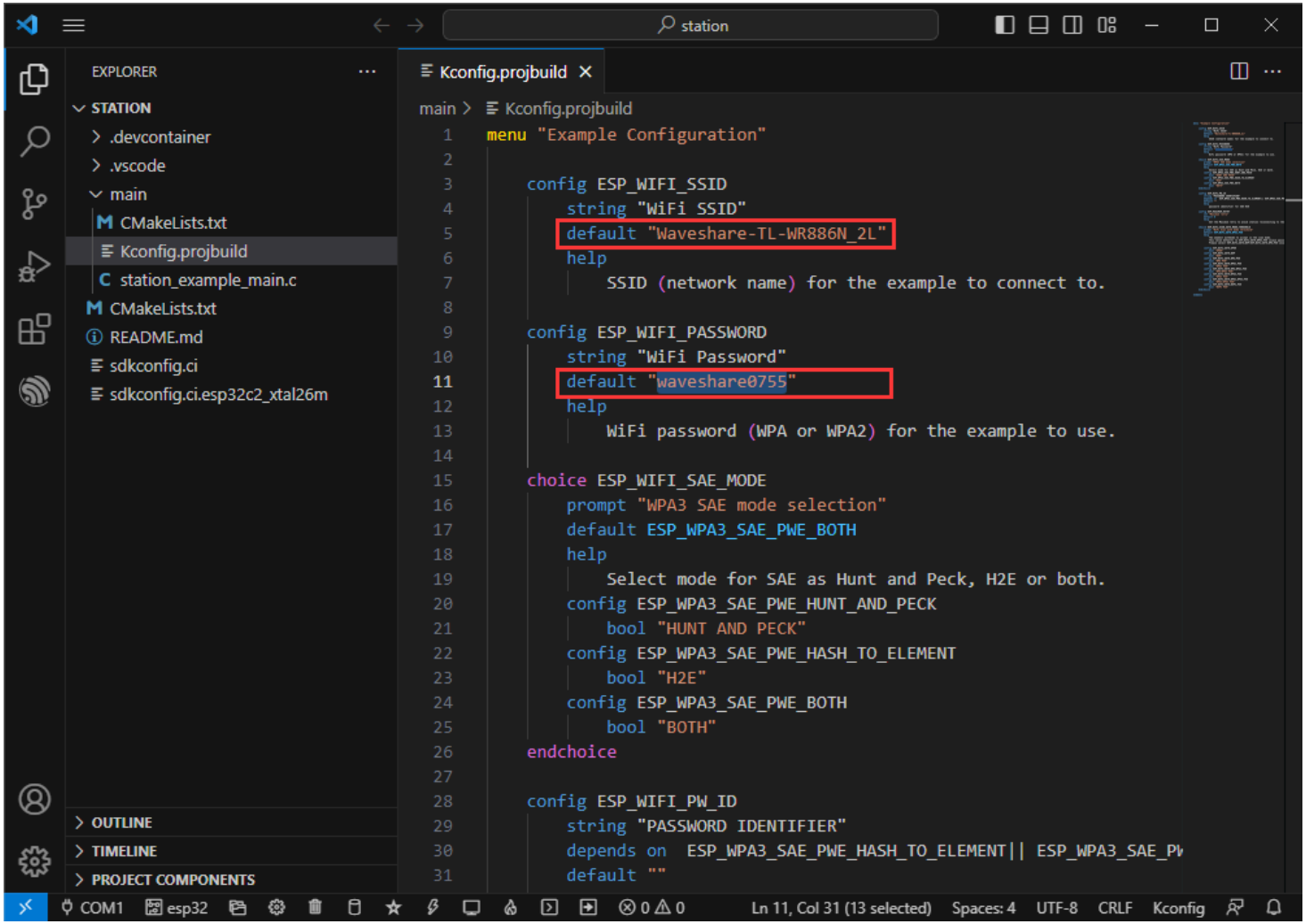
- Create the official example station according to the tutorial above. (Create Example (https://www.waveshare.com/wiki/ESP32-C6-DEV-KIT-N8#Official_Demo_Usage_GUIDE))

- Modify the contents of the demo to connect to the required WiFi.
- Go to the **Kconfig.projbuild** file.



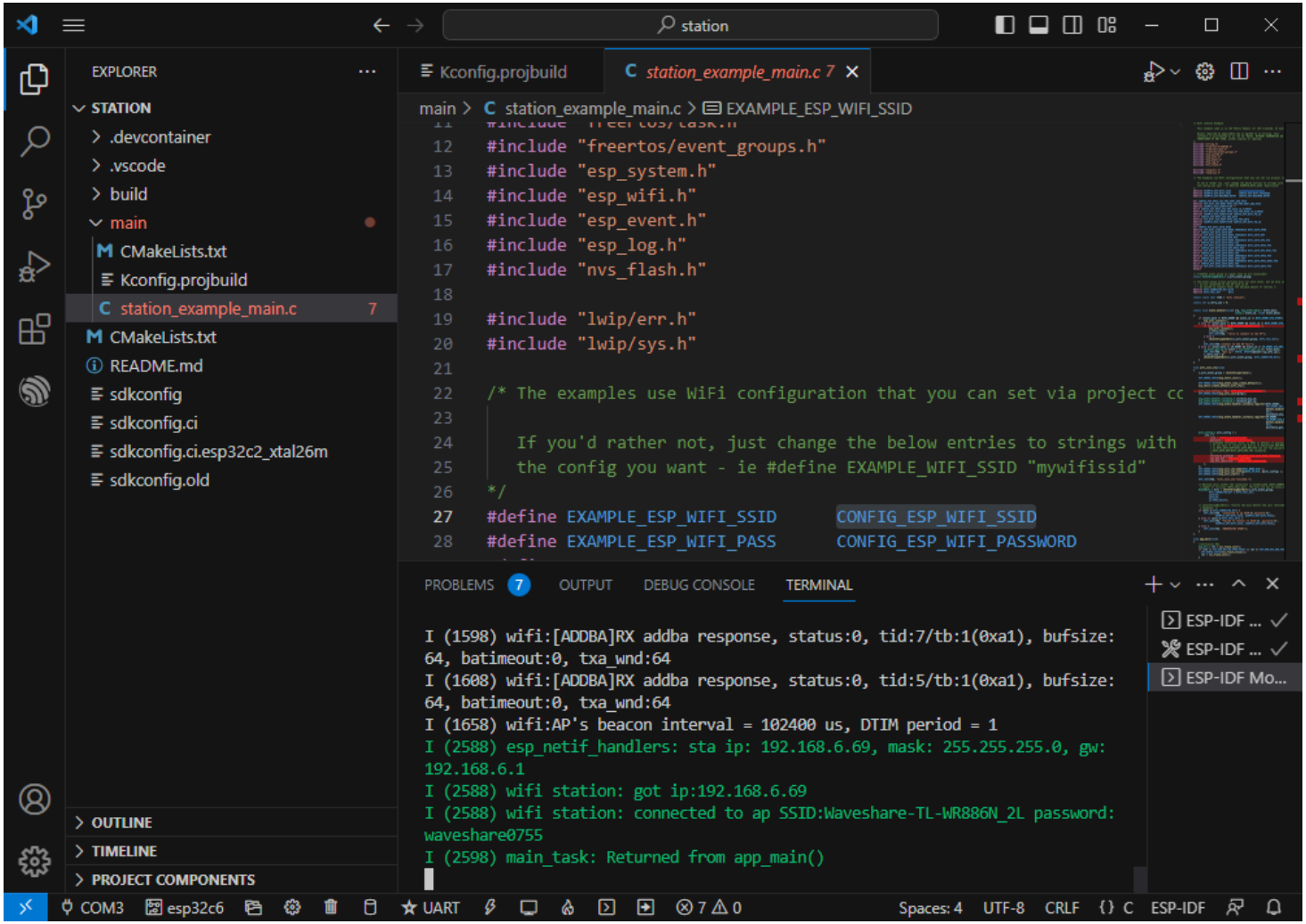
(/wiki/File:ESP32-C6-DEV-KIT-N8-60.png)

- Change the original **WiFi SSID** and **WiFi Password** to the WiFi information you want to connect to.



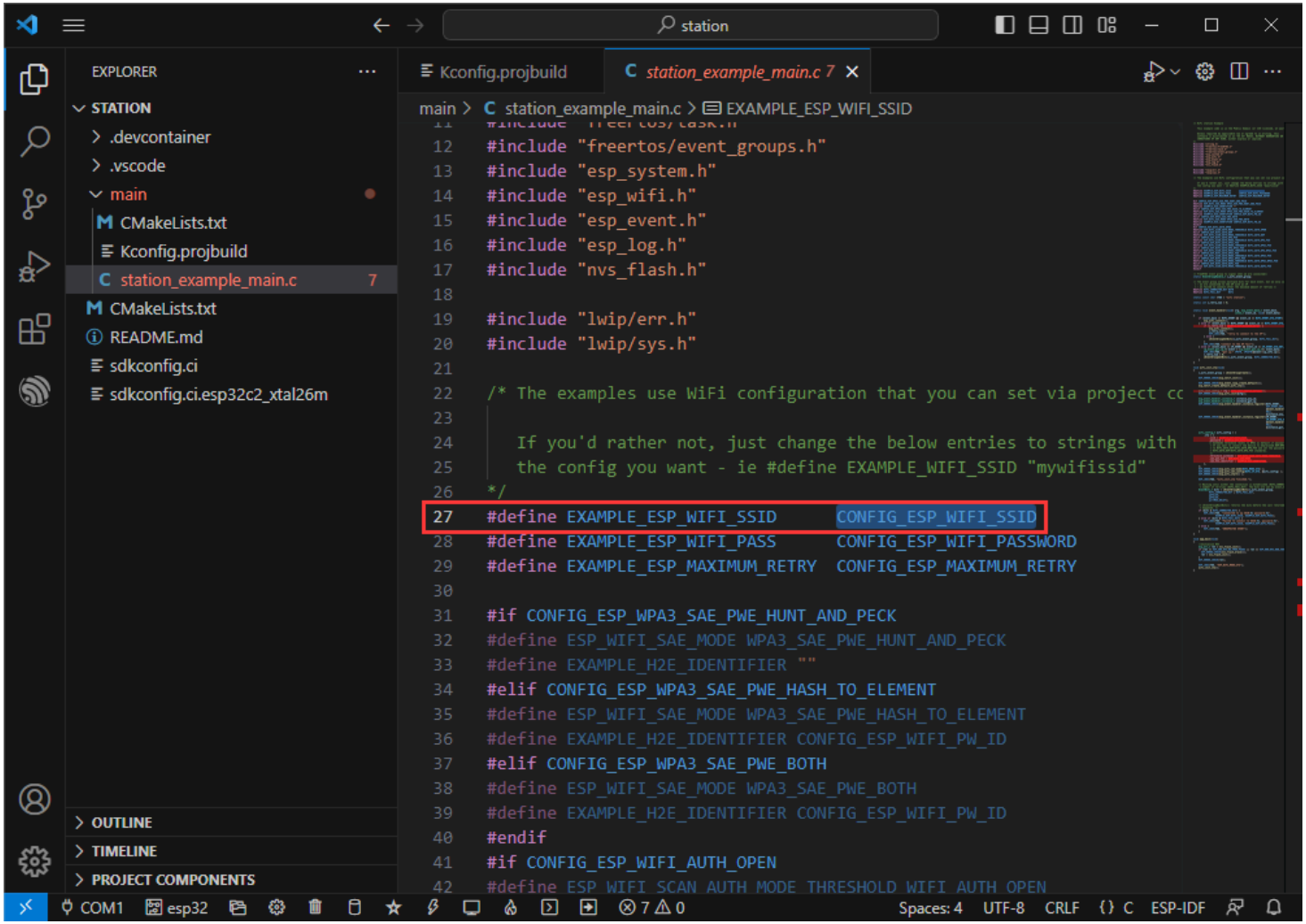
(/wiki/File:ESP32-C6-DEV-KIT-N8-61.png)

- Modify the COM port and the driver object (**it is recommended to prioritize the COM port corresponding to USB, which can be viewed through the device manager**), click compile and flash to run the demo.



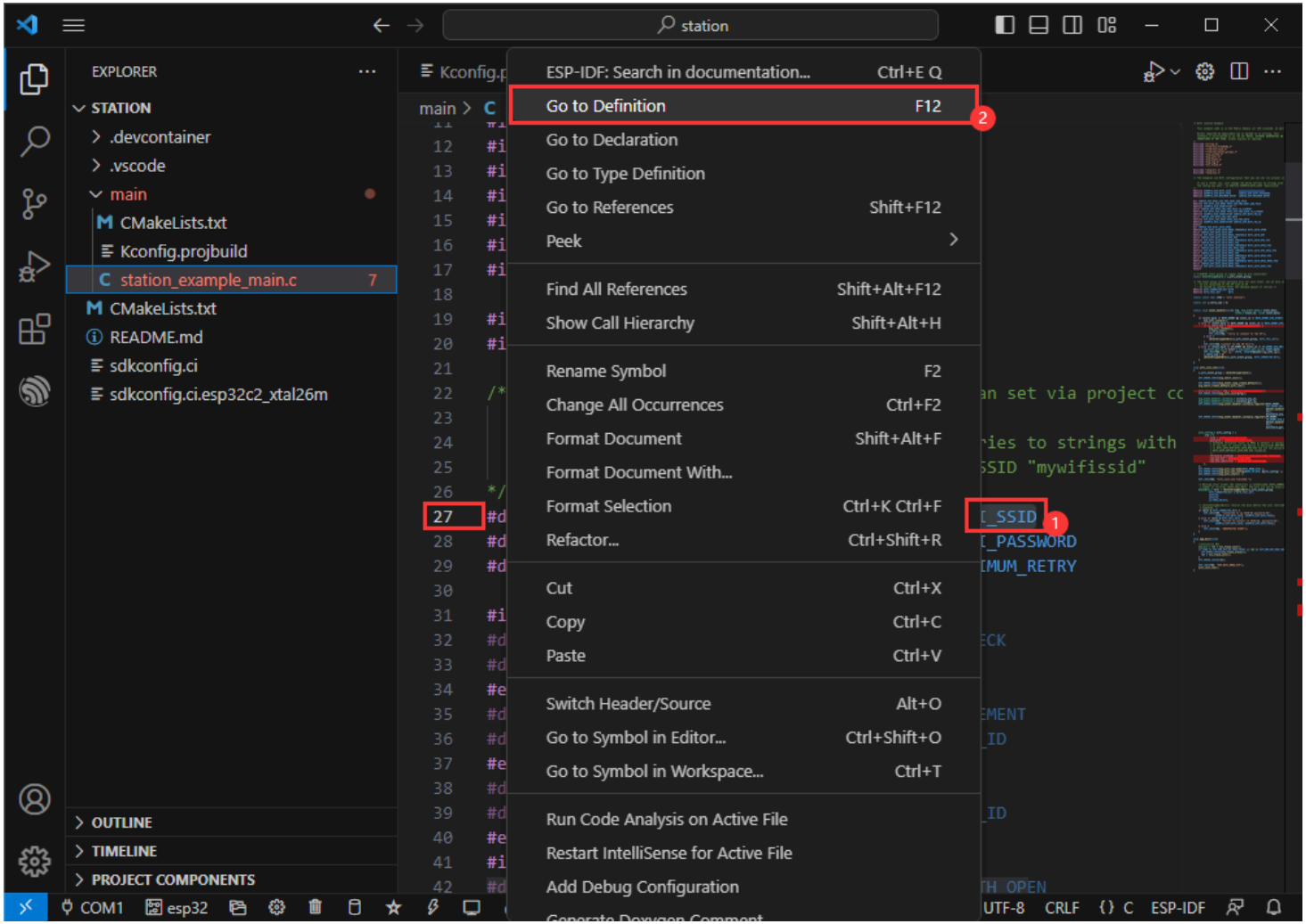
(/wiki/File:ESP32-C6-DEV-KIT-N8-62.png)

- You can check the value of **CONFIG_ESP_WIFI_SSID**.
- Go to the **station_ example_ main.c** file.



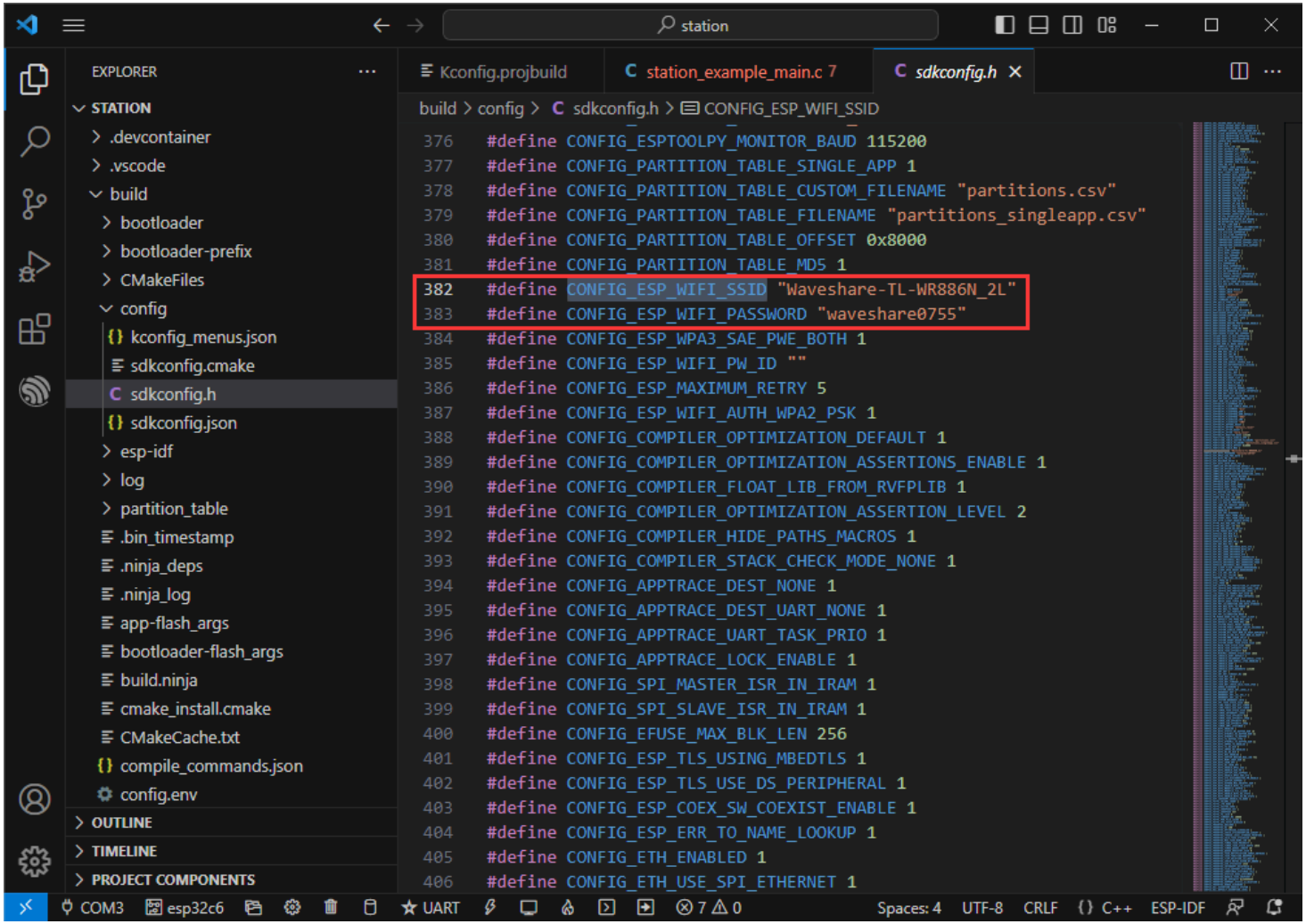
(/wiki/File:ESP32-C6-DEV-KIT-N8-63.png)

- Right-click to go to the definition.



(/wiki/File:ESP32-C6-DEV-KIT-N8-64.png)

- The previously set value can be seen as:



(/wiki/File:ESP32-C6-DEV-KIT-N8-65.png)

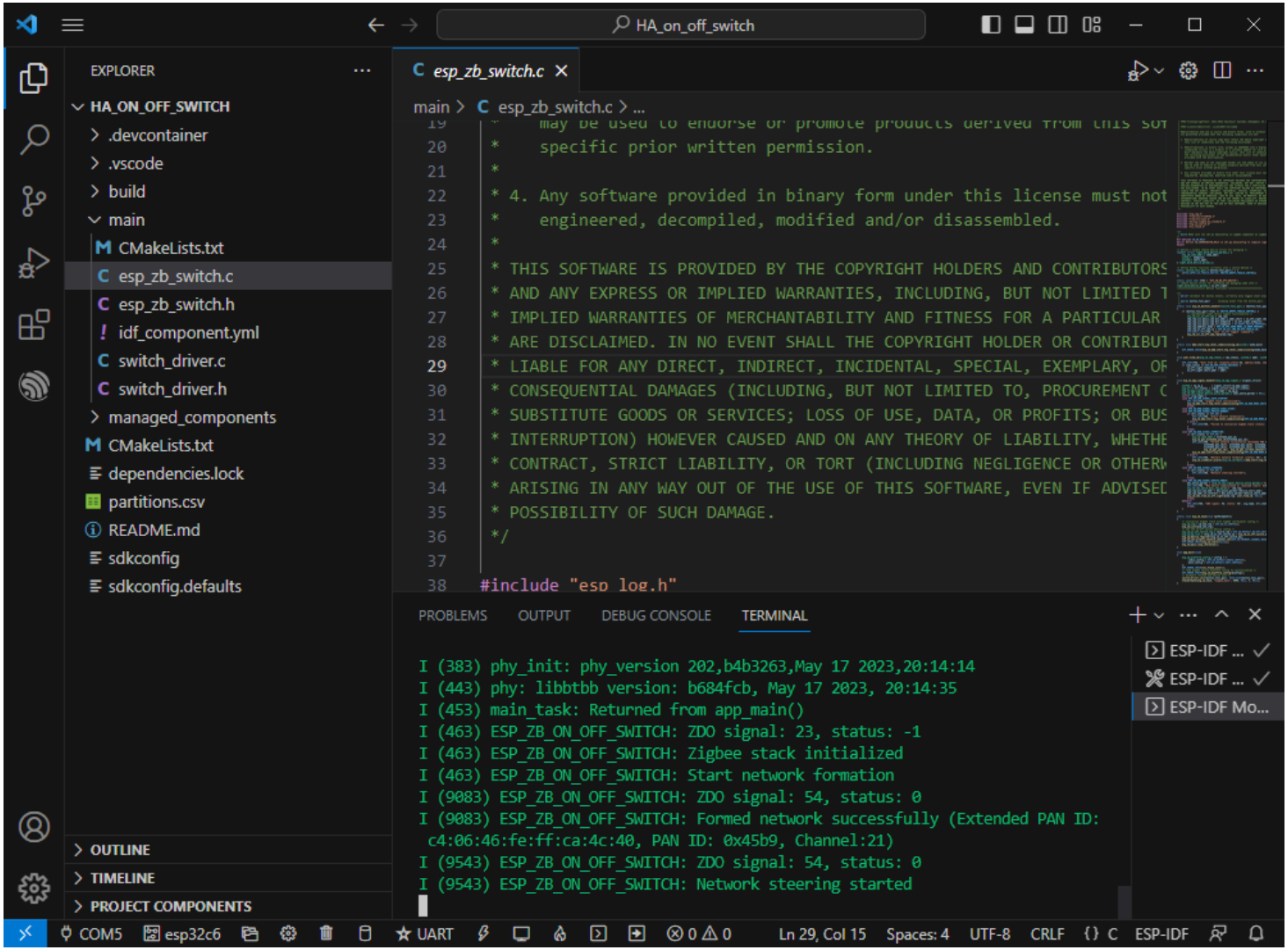
Zigbee

- Official example 1 path: Zigbee -> light_sample -> HA_on_off_switch.
- Official example 2 path: Zigbee -> light_sample -> HA_on_off_light.
- Example effect: use 2 ESP32-C6 boards, use the **BOOT** key of one ESP32-C6 board (flash HA_on_off_switch demo) to control the RGB LED ON/OFF on the other one.

- **Note: Please flash the HA_on_off_switch demo to one board first, and then flash the HA_on_off_light demo to the other board.**

Software Operation 1

- Create the official example HA_on_off_switch according to the tutorial above. (Create Example (https://www.waveshare.com/wiki/ESP32-C6-DEV-KIT-N8#Official_Demo_Usage_GUIDE))
- The demo is compatible with ESP32-C6 and can be used without modifying the demo content.
- Modify the COM port and the driver object (**it is recommended to prioritize the COM port corresponding to USB, which can be viewed through the device manager**), click compile and flash to run the demo.

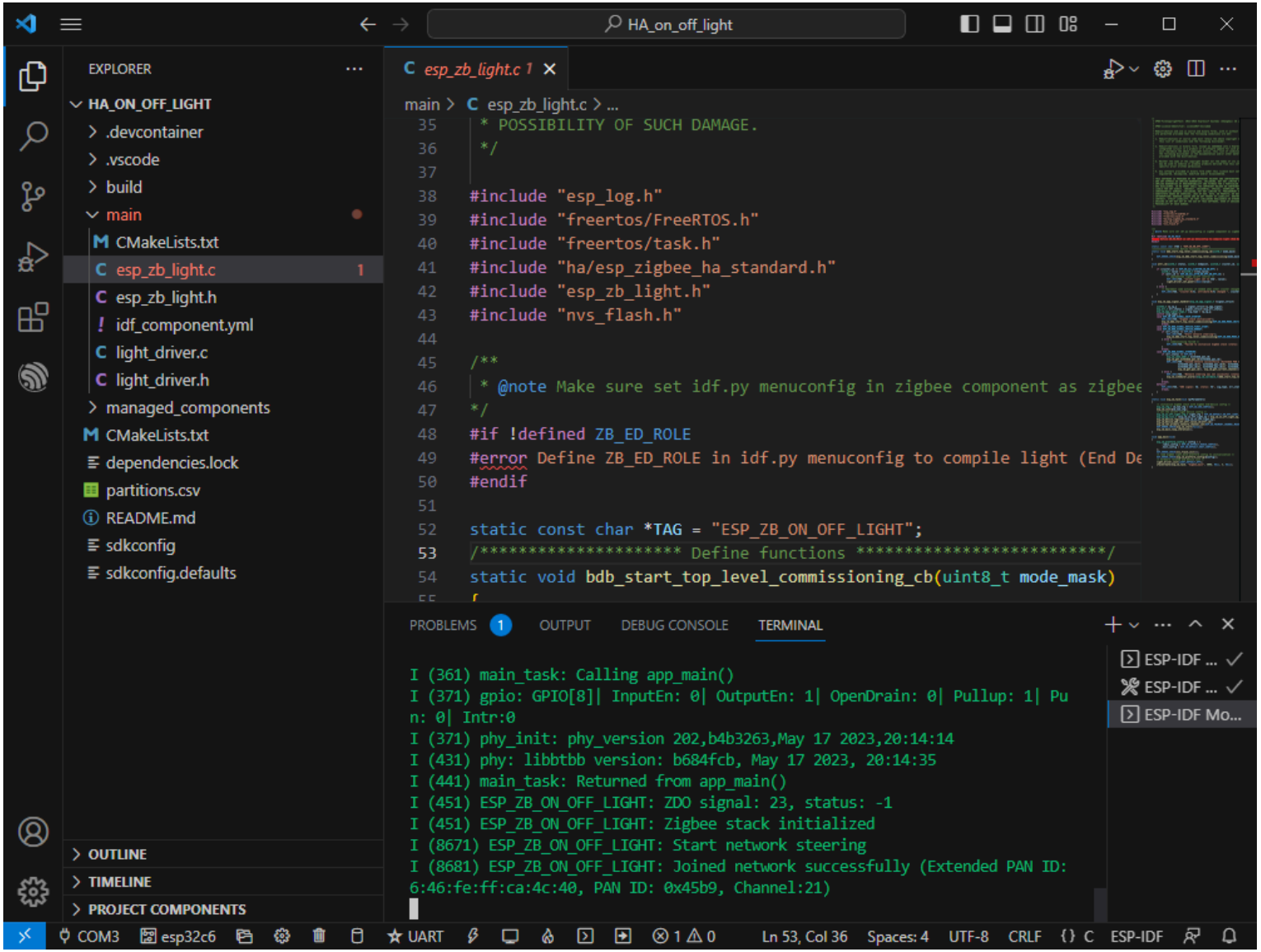


(/wiki/File:ESP32-C6-DEV-KIT-N8-66.png)

Software Operation 2

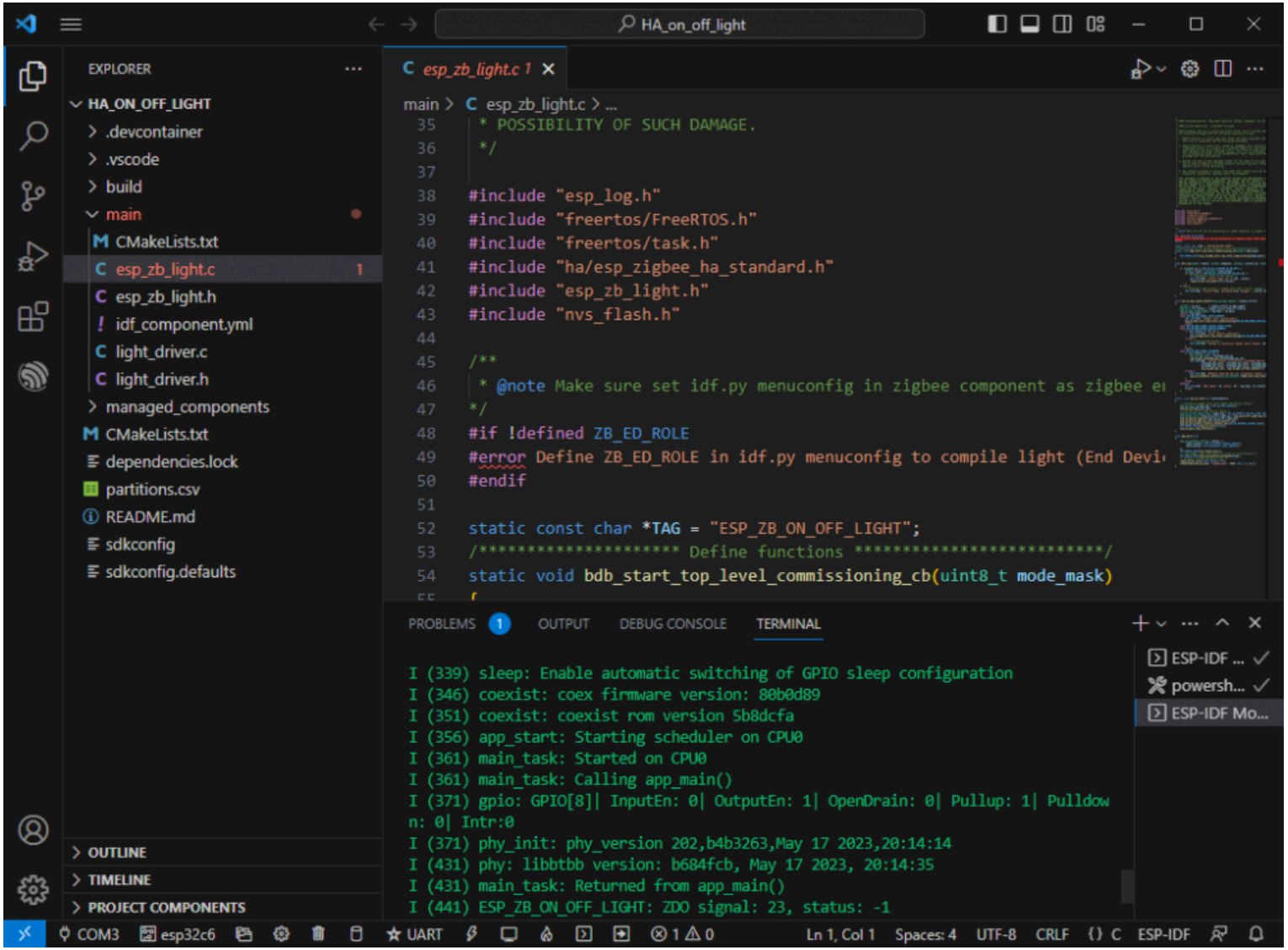
- Follow the tutorial above to create the official example HA_on_off_light. (Create Example (https://www.waveshare.com/wiki/ESP32-C6-DEV-KIT-N8#Official_Demo_Usage_GUIDE))
- The demo is compatible with ESP32-C6 and can be used without modifying the demo content.

- Modify the COM port and driver object, click compile and flash to run the demo (**you need to wait for a moment for the two chips to establish a connection**).



(/wiki/File:ESP32-C6-DEV-KIT-N8-70.png)

- **If the device remains unconnected**, it may be due to residual network information on the device, so you can erase the device information (Erase Tutorial) and reorganize the network.



(/wiki/File:ESP32-C6-DEV-KIT-N8-71.png)

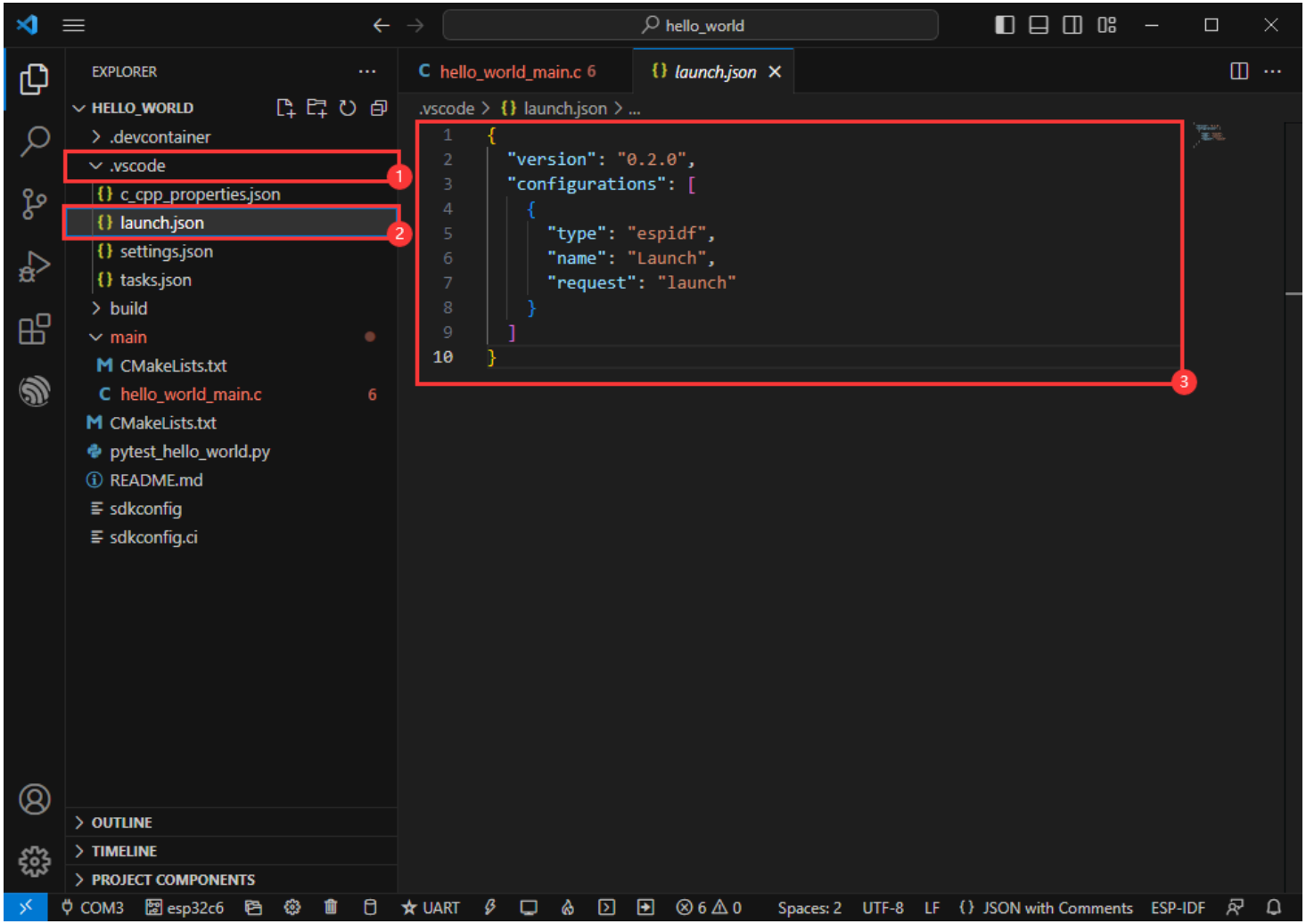
JTAG Debug

Software Operation

- Create a debugging example, this example uses the official example hello_world. (Create

Example (https://www.waveshare.com/wiki/ESP32-C6-DEV-KIT-N8#Official_Demo_Usage_GUIDE))

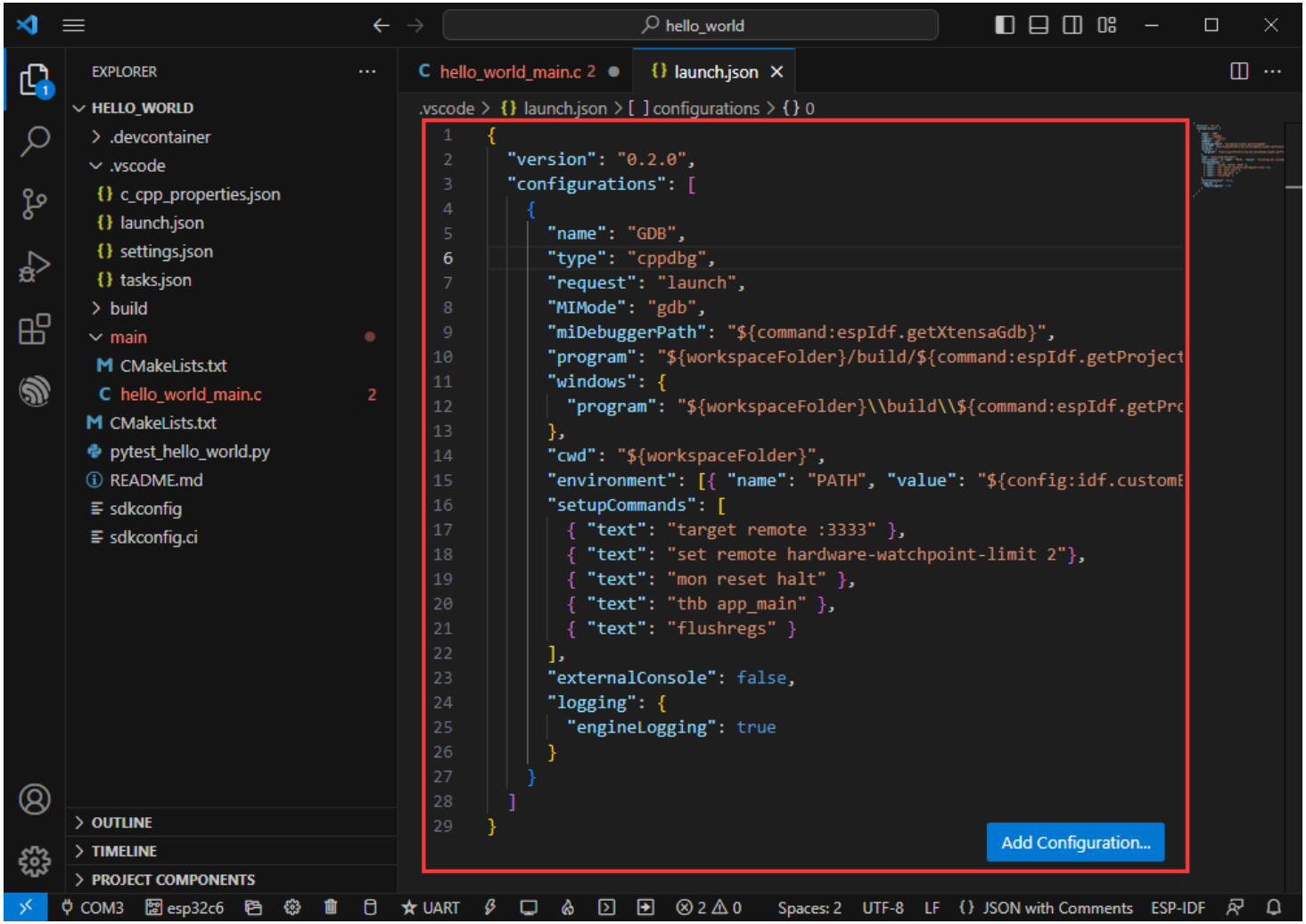
- Modify the **launch.json** file.



(/wiki/File:ESP32-C6-DEV-KIT-N8-72.png)

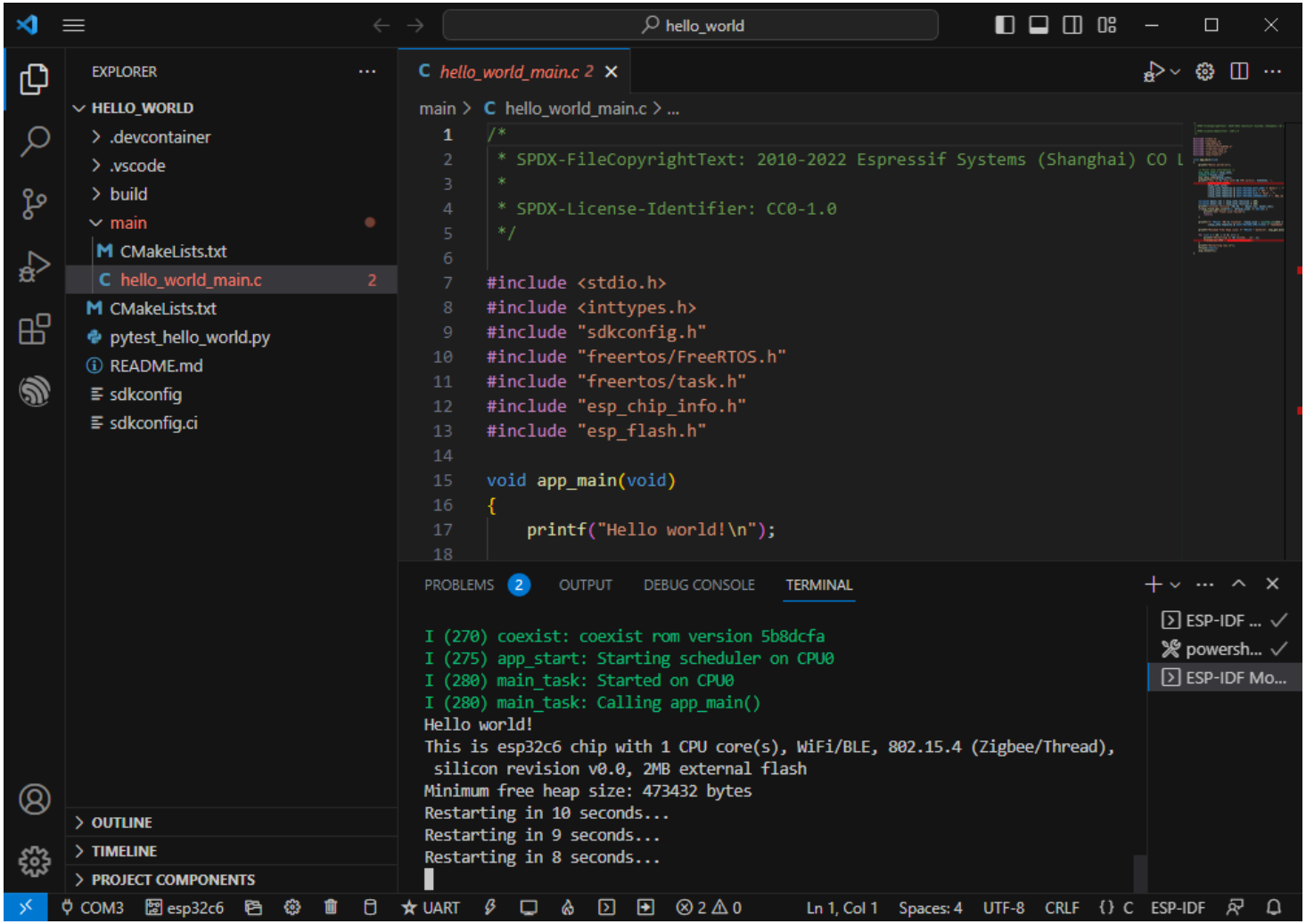
- Input the following content:

```
{
 "version": "0.2.0",
 "configurations": [
   {
     "name": "GDB",
     "type": "cppdbg",
     "request": "launch",
     "MIMode": "gdb",
     "miDebuggerPath": "${command:espIdf.getXtensaGdb}",
     "program": "${workspaceFolder}/build/${command:espIdf.getProjectName}.elf",
     "windows": {
       "program": "${workspaceFolder}\\build\\${command:espIdf.getProjectName}.elf"
     },
     "cwd": "${workspaceFolder}",
     "environment": [{ "name": "PATH", "value": "${config:idf.customExtraPaths}" }],
     "setupCommands": [
       { "text": "target remote :3333" },
       { "text": "set remote hardware-watchpoint-limit 2"},
       { "text": "mon reset halt" },
       { "text": "thb app_main" },
       { "text": "flushregs" }
     ],
     "externalConsole": false,
     "logging": {
       "engineLogging": true
     }
   }
 ]
}
```



(/wiki/File:ESP32-C6-DEV-KIT-N8-73.png)

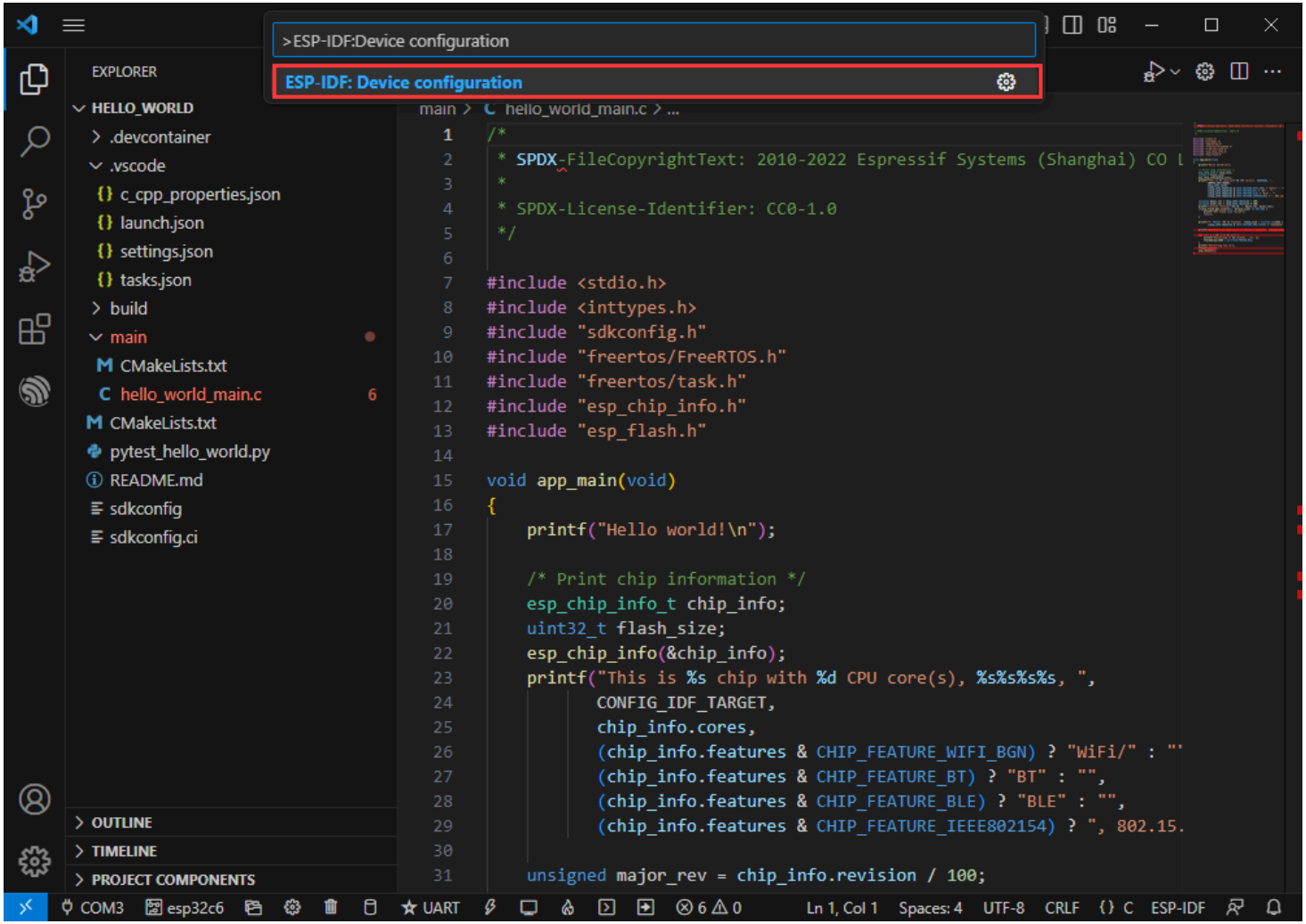
- The demo is compatible with ESP32-C6 and can be used without modifying the demo content.
- Modify the COM port and the driver object (**Please use the USB interface; the UART interface does not support JTAG debugging. The corresponding COM port can be checked through the Device Manager.**), click compile and flash to run the demo.



(/wiki/File:ESP32-C6-DEV-KIT-N8-30.png)

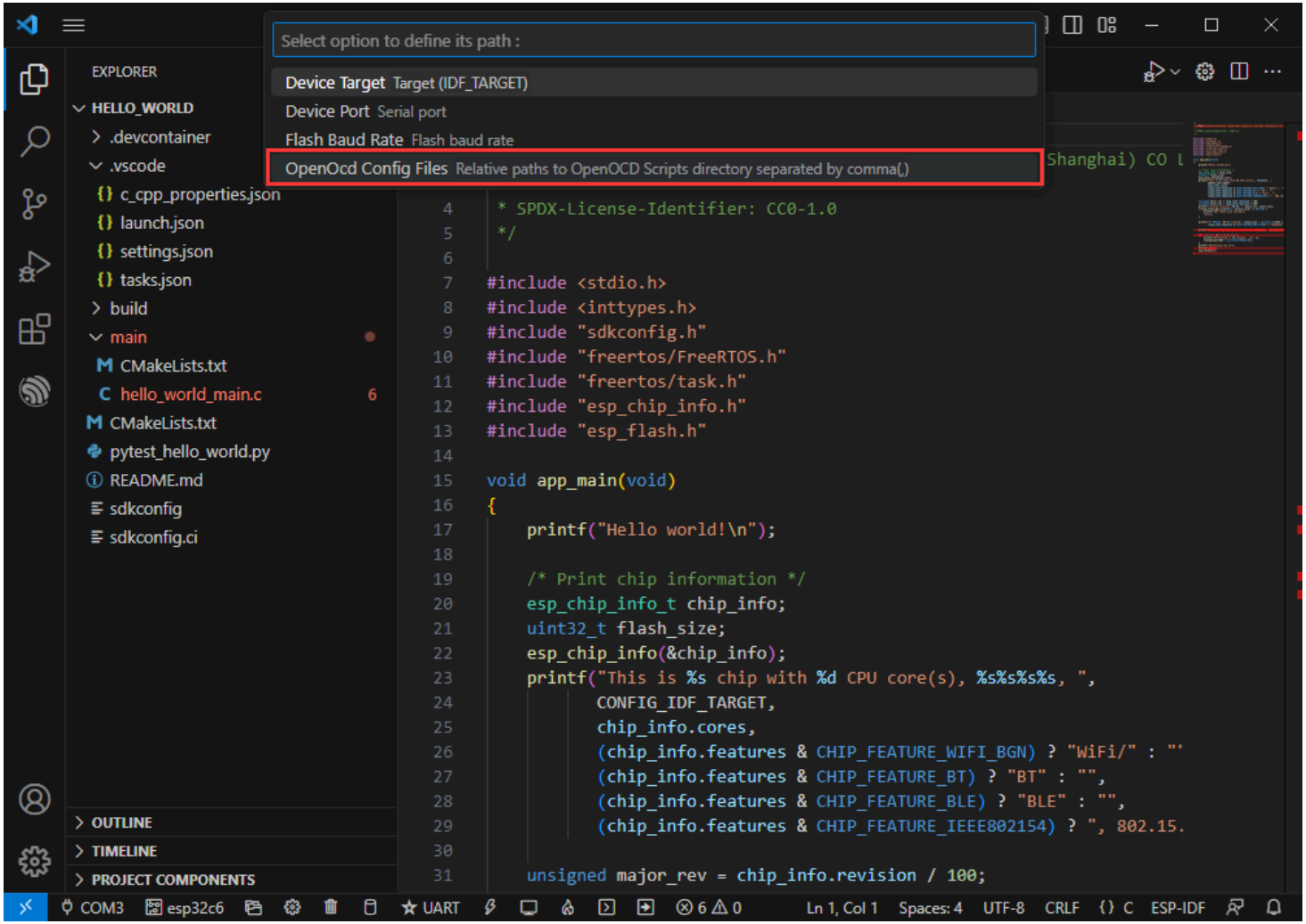
- Press F1 and input:

```
ESP-IDF:Device configuration
```



(/wiki/File:ESP32-C6-DEV-KIT-N8-75.png)

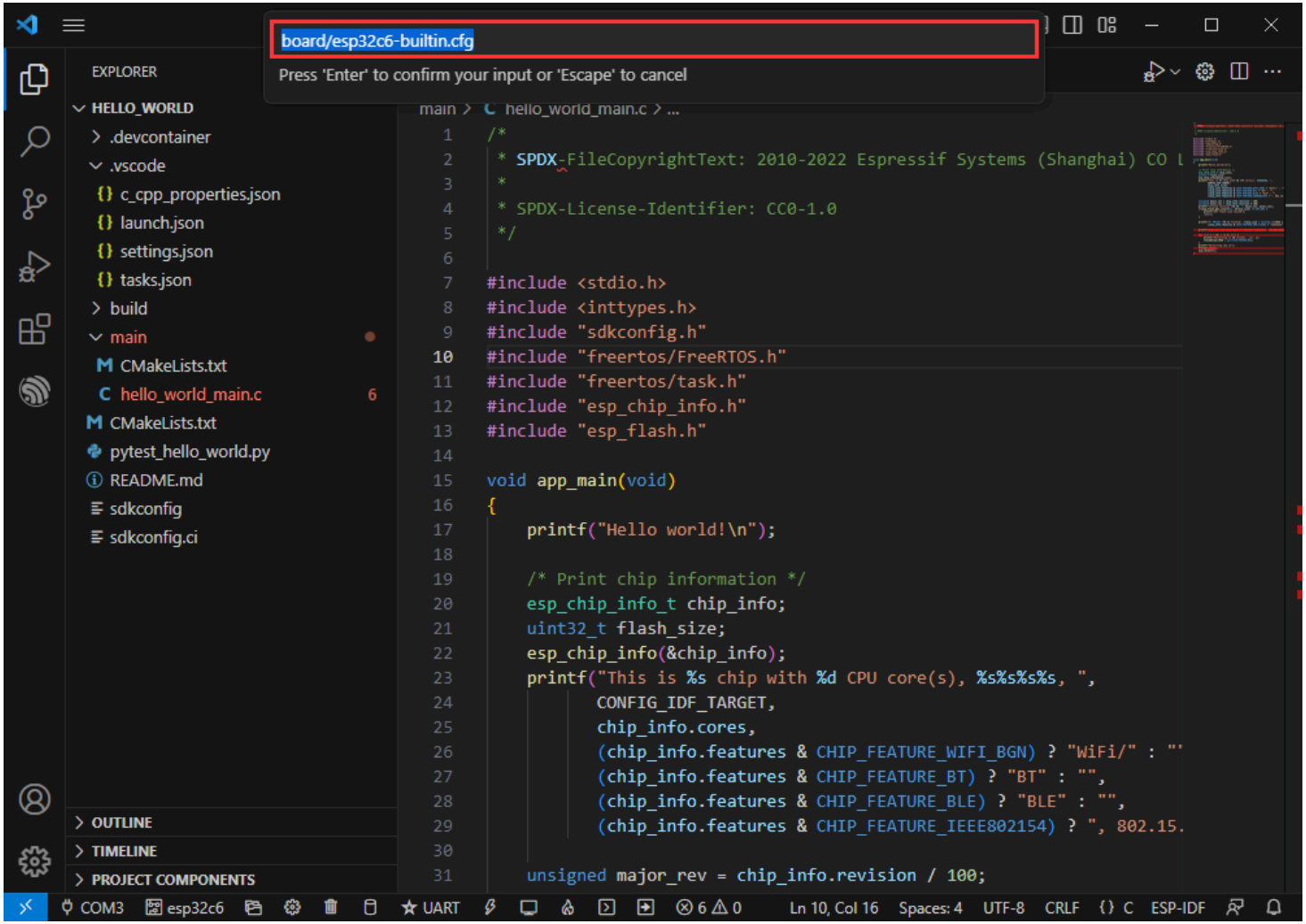
- Select **OpenOcd Config Files**.



(/wiki/File:ESP32-C6-DEV-KIT-N8-76.png)

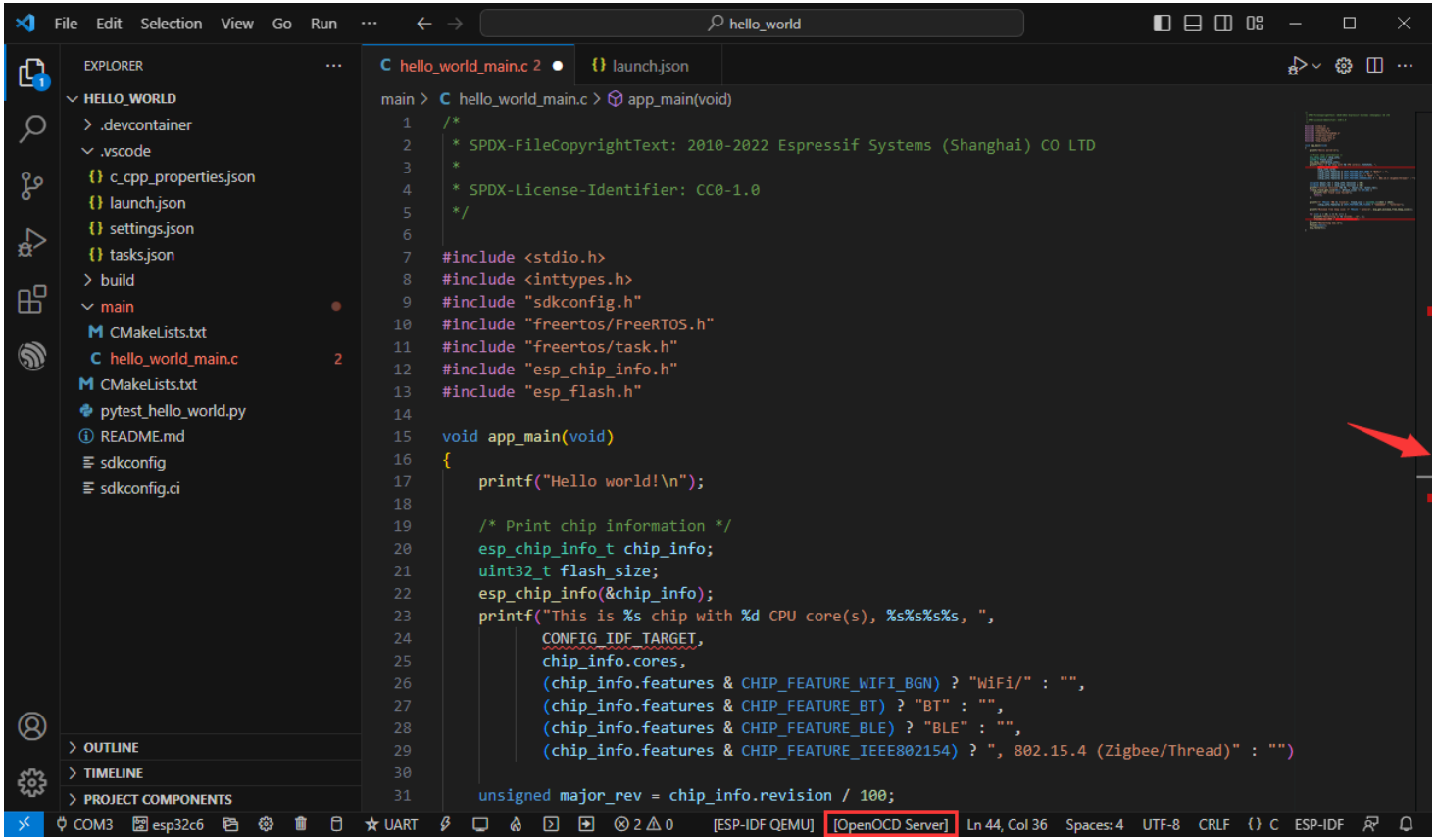
- Type **board/esp32c6-builtin.cfg** (if this is the default, just enter).

```
board/esp32c6-builtin.cfg
```



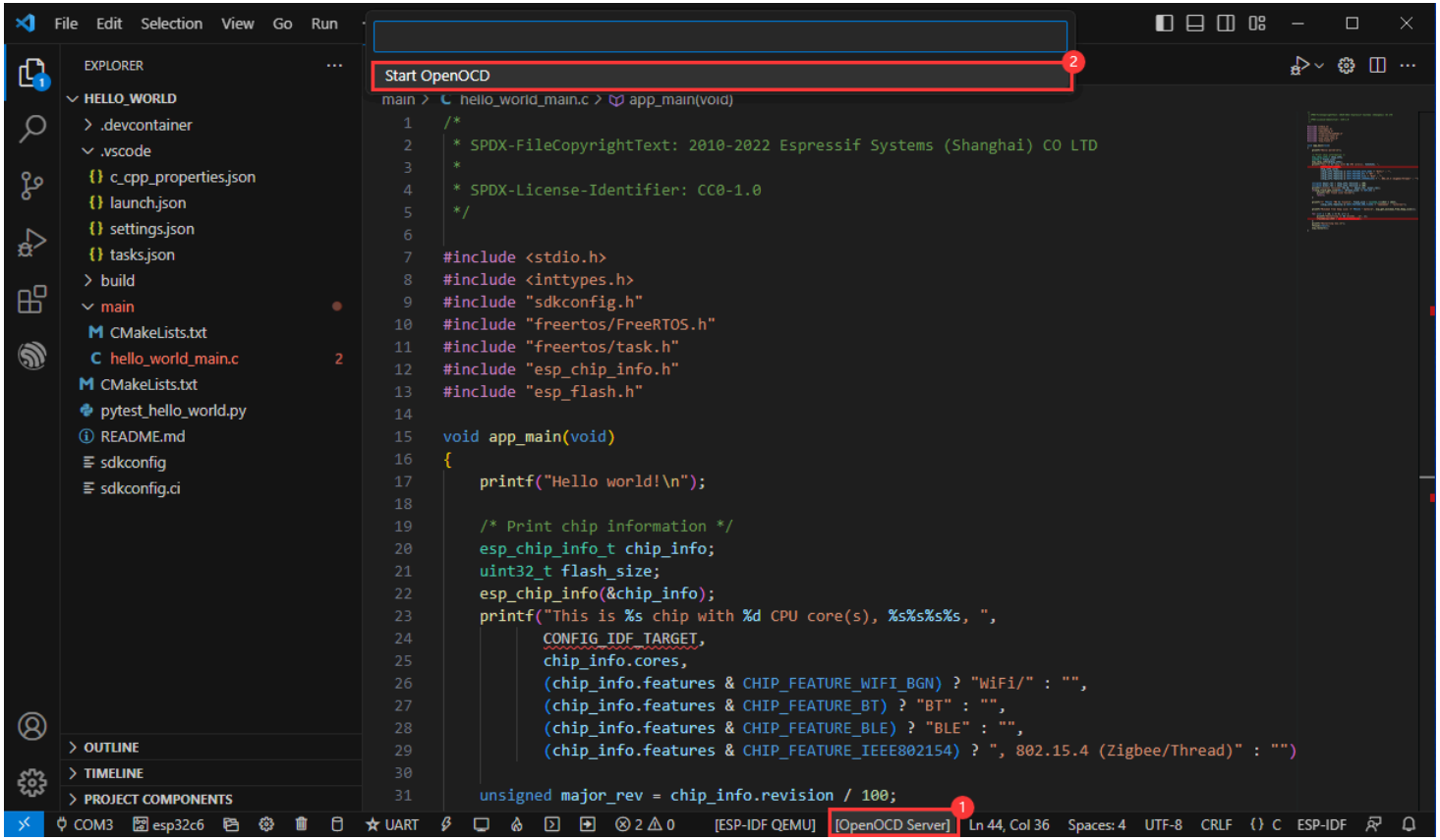
(/wiki/File:ESP32-C6-DEV-KIT-N8077.png)

- Stretch the width of the window until **[OpenOCD Server]** is displayed at the bottom.



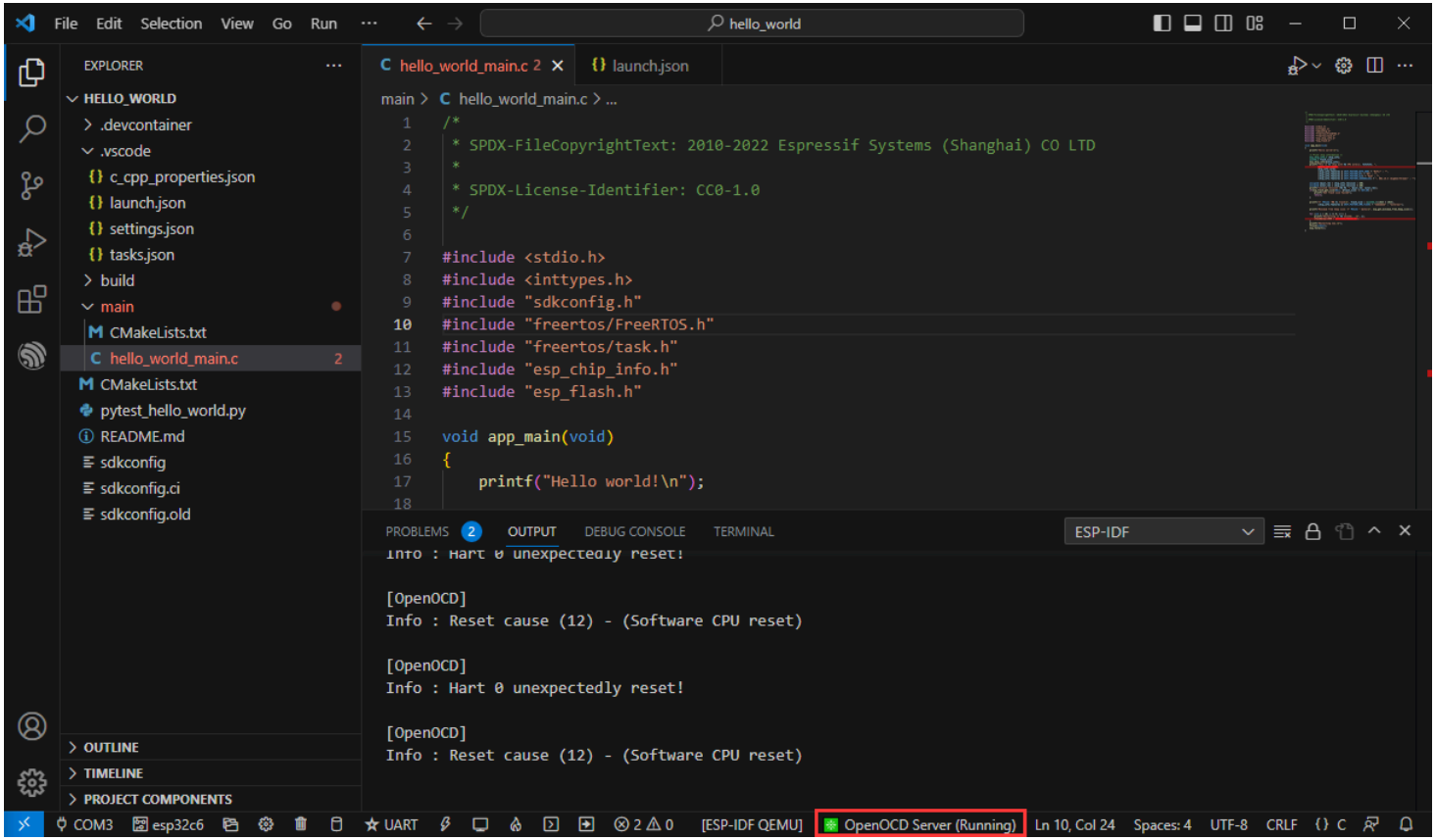
(/wiki/File:ESP32-C6-DEV-KIT-N8078.png)

- Click **[OpenOCD Server]** and select **Start OpenOCD**.



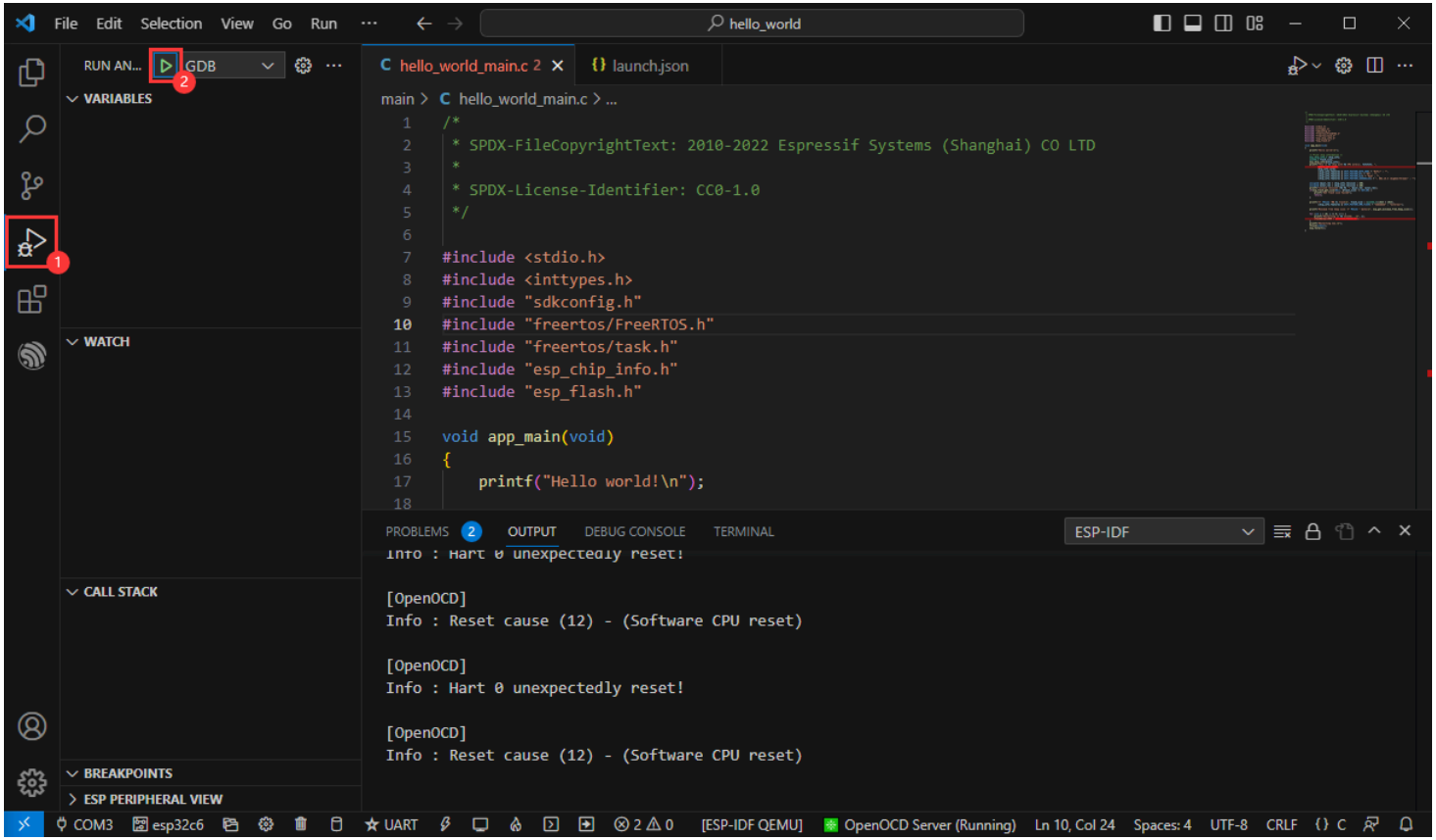
(/wiki/File:ESP32-C6-DEV-KIT-N8079.png)

- Successfully opened as follows:



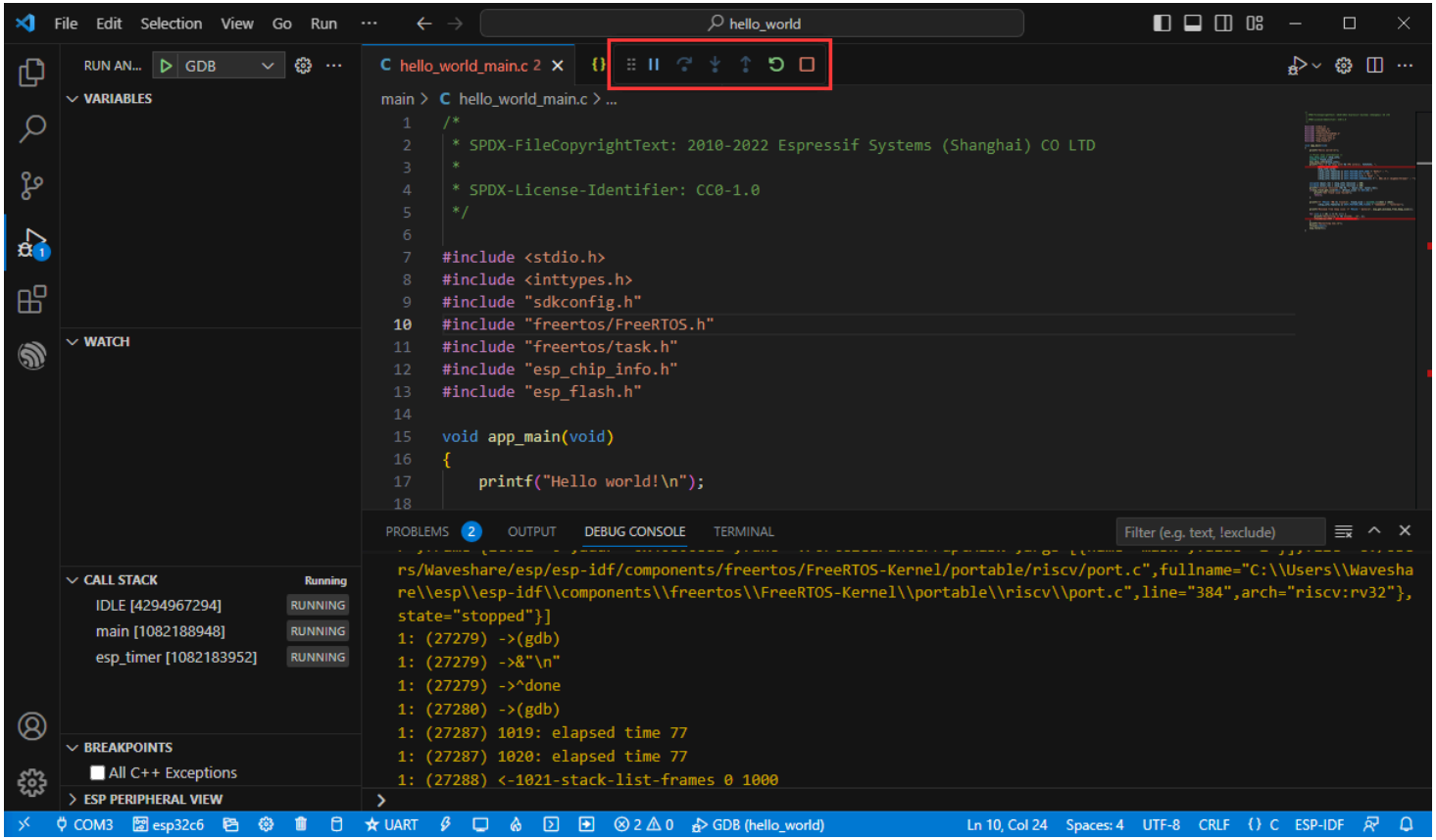
(/wiki/File:ESP32-C6-DEV-KIT-N8-80.png)

- Go to the debug function and click Debug:



(/wiki/File:ESP32-C6-DEV-KIT-N8-81.png)

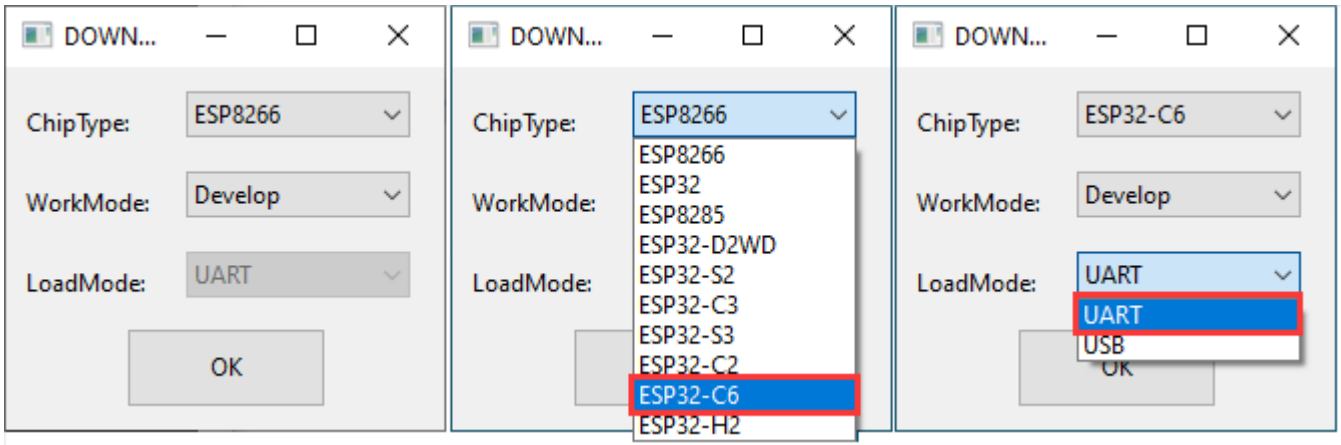
- Successfully enter the debugging interface:



(/wiki/File:ESP32-C6-DEV-KIT-N8-82.png)

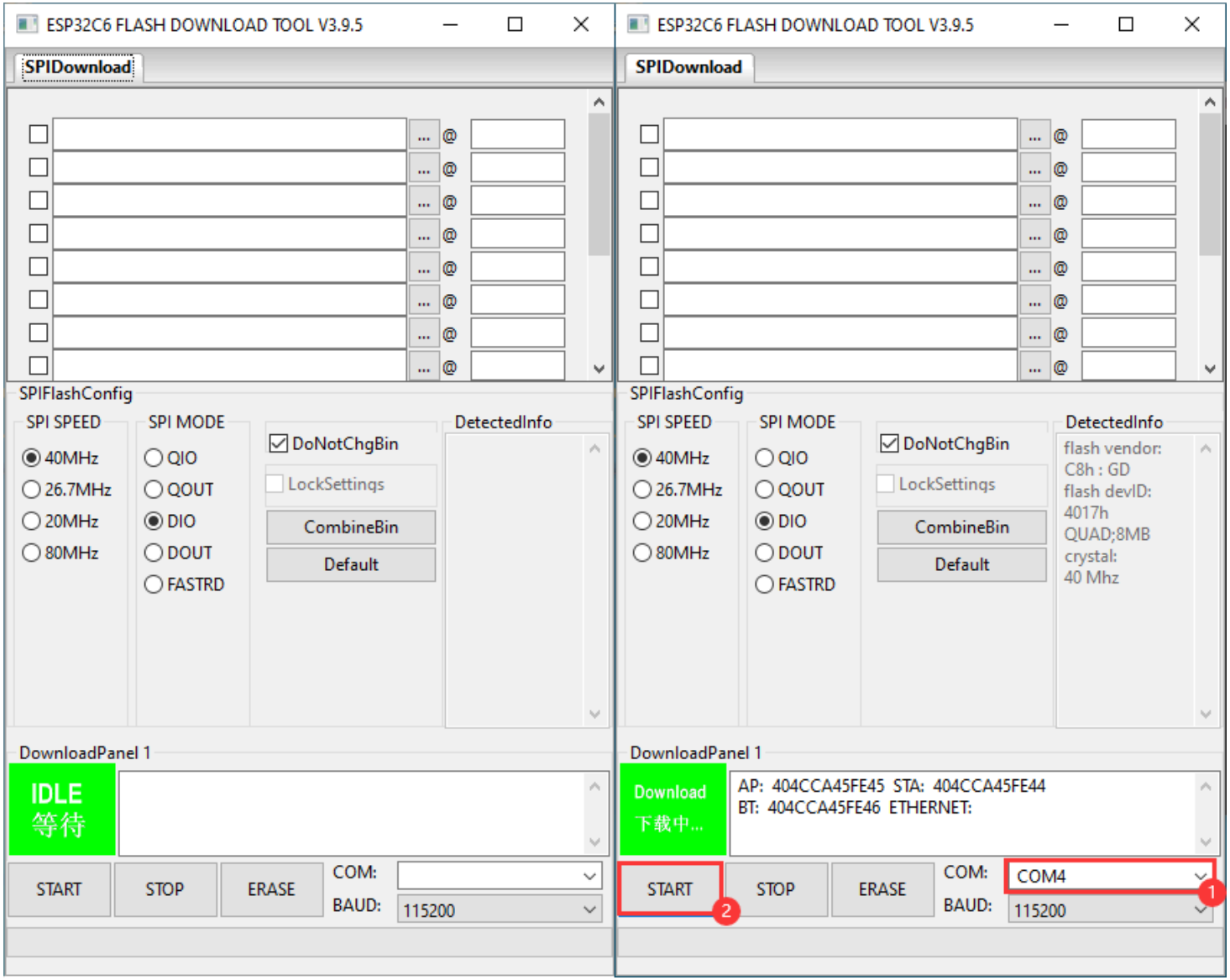
Erase Device Flash

- Unpack the software resource package (Flash debugging software (https://files.waveshare.com/wiki/ESP32-C6-DEV-KIT-N8/Flash_download_tool_3.9.5_0.zip)).
- Open **flash_download_tool_3.9.5.exe** software, select ESP32-C6 and UART.



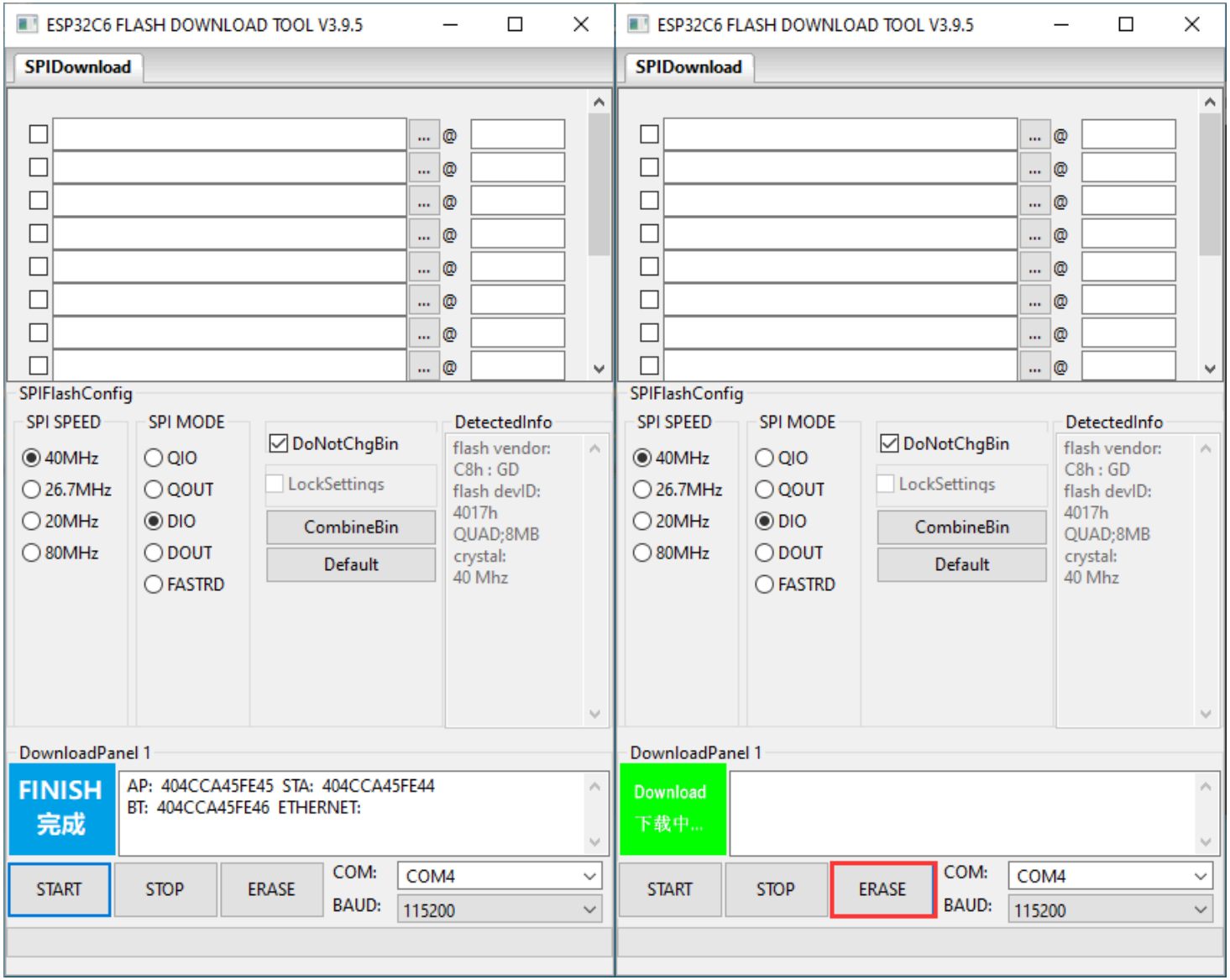
(/wiki/File:ESP32-C6-DEV-KIT-N8-83.png)

- Select the UART port number, and click **START** (not select any bin file).



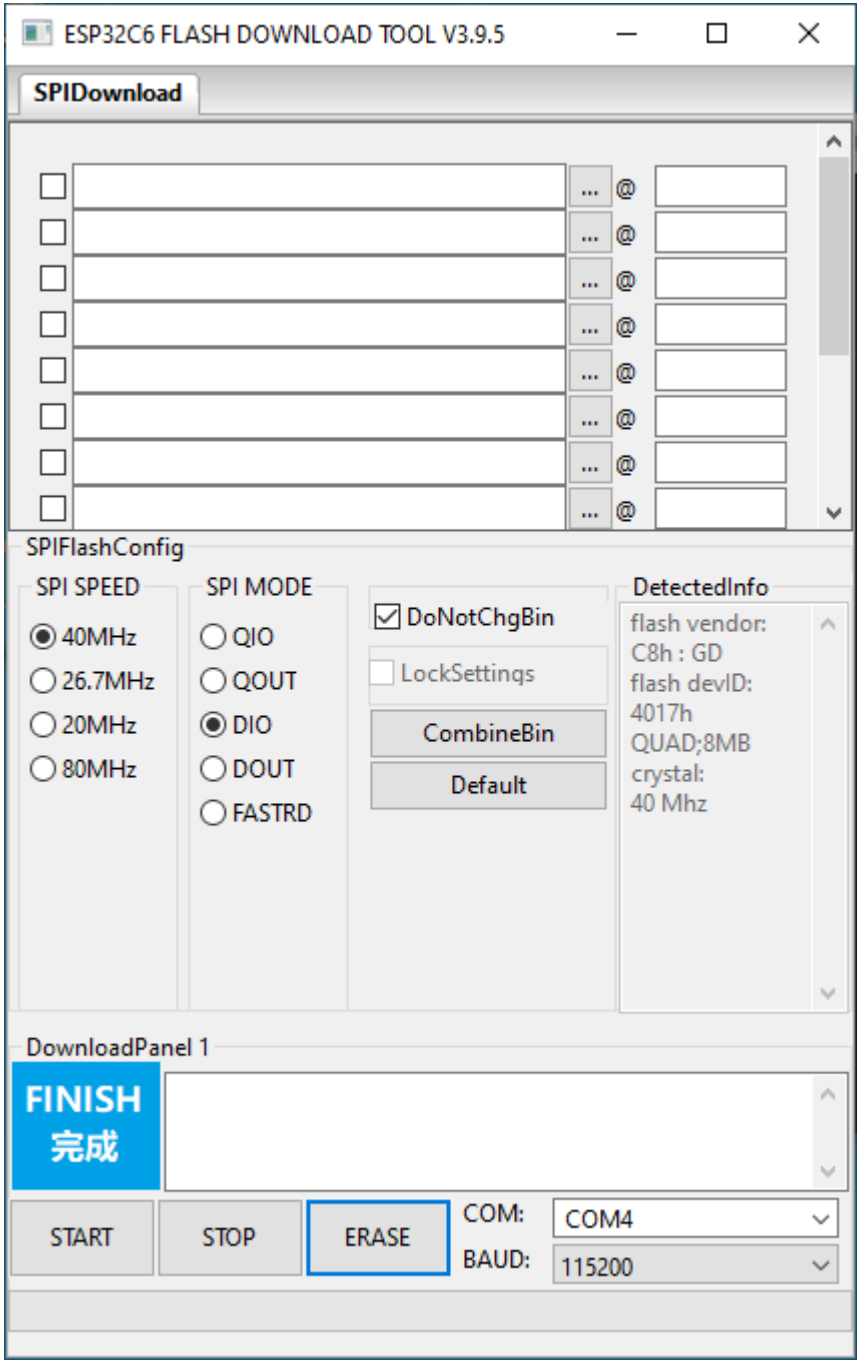
(/wiki/File:ESP32-C6-DEV-KIT-N8-84.png)

- After programming, click on "ERASE".



(/wiki/File:ESP32-C6-DEV-KIT-N8-85.png)

- Waiting for Erase to Finish.



(/wiki/File:ESP32-C6-DEV-KIT-N8-

86.png)

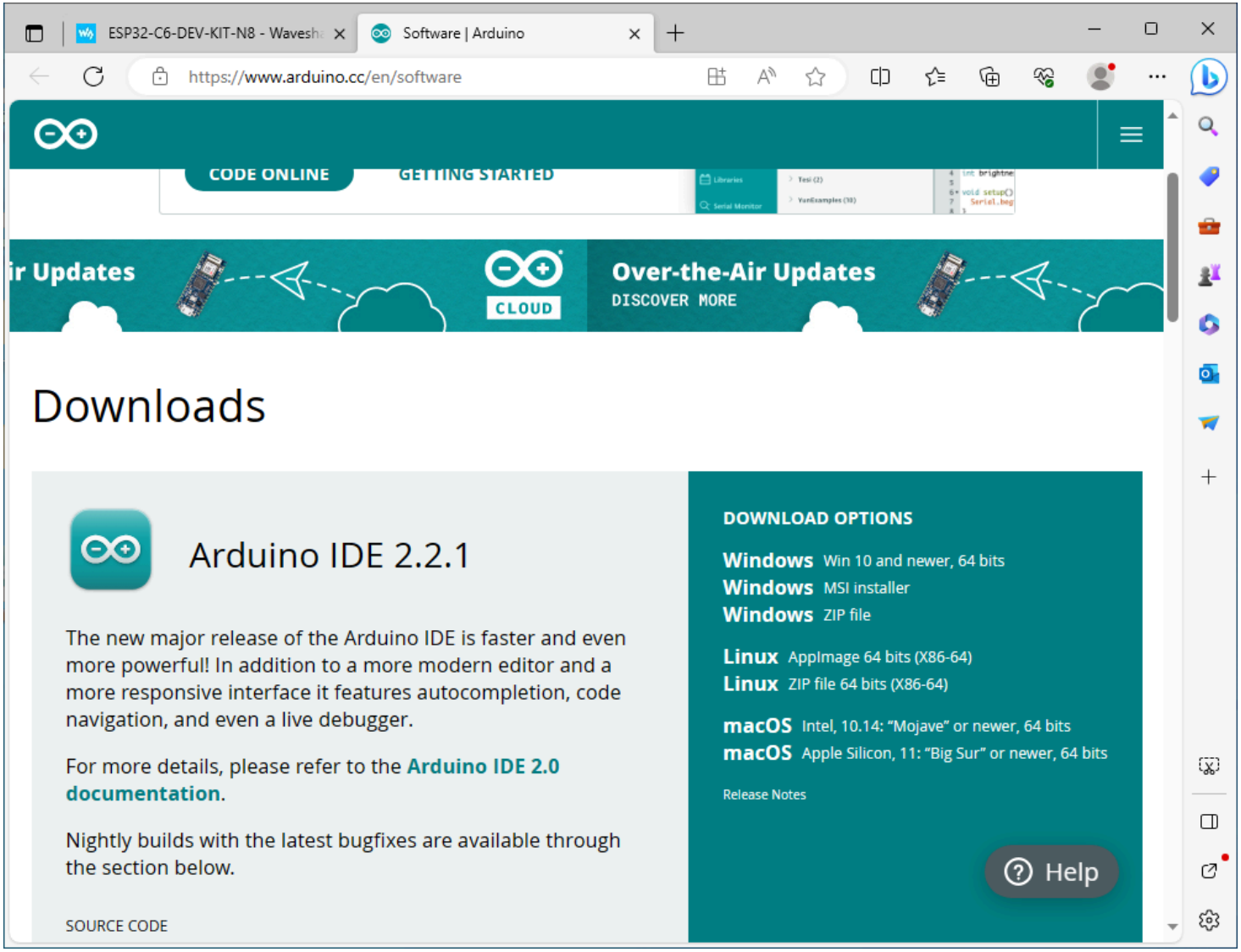
Working with Arduino

Please note that Arduino 3.0.0-alpha is based on ESP-IDF v5.1, which is quite different from the previous ESP-IDF V4.X. The original program may need to be adjusted after the following operations.

- Please do not use the computer username in Chinese as it will cause compilation errors.

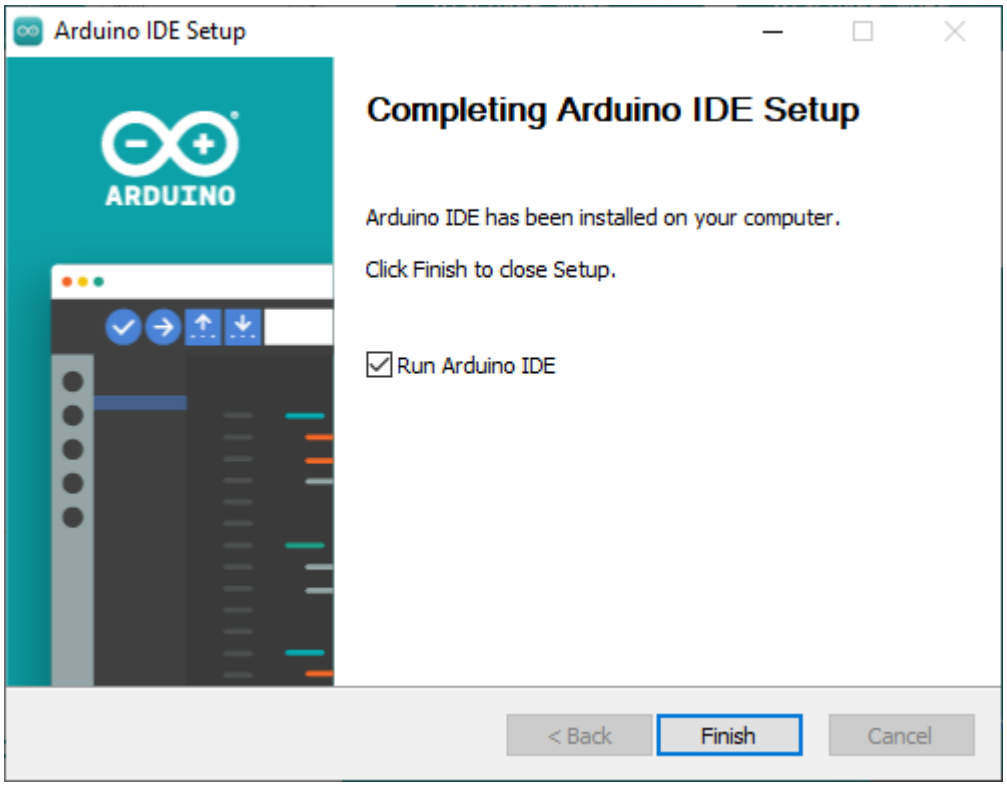
Environment Set-up

- Install Arduino IDE (https://www.arduino.cc/en/software/).



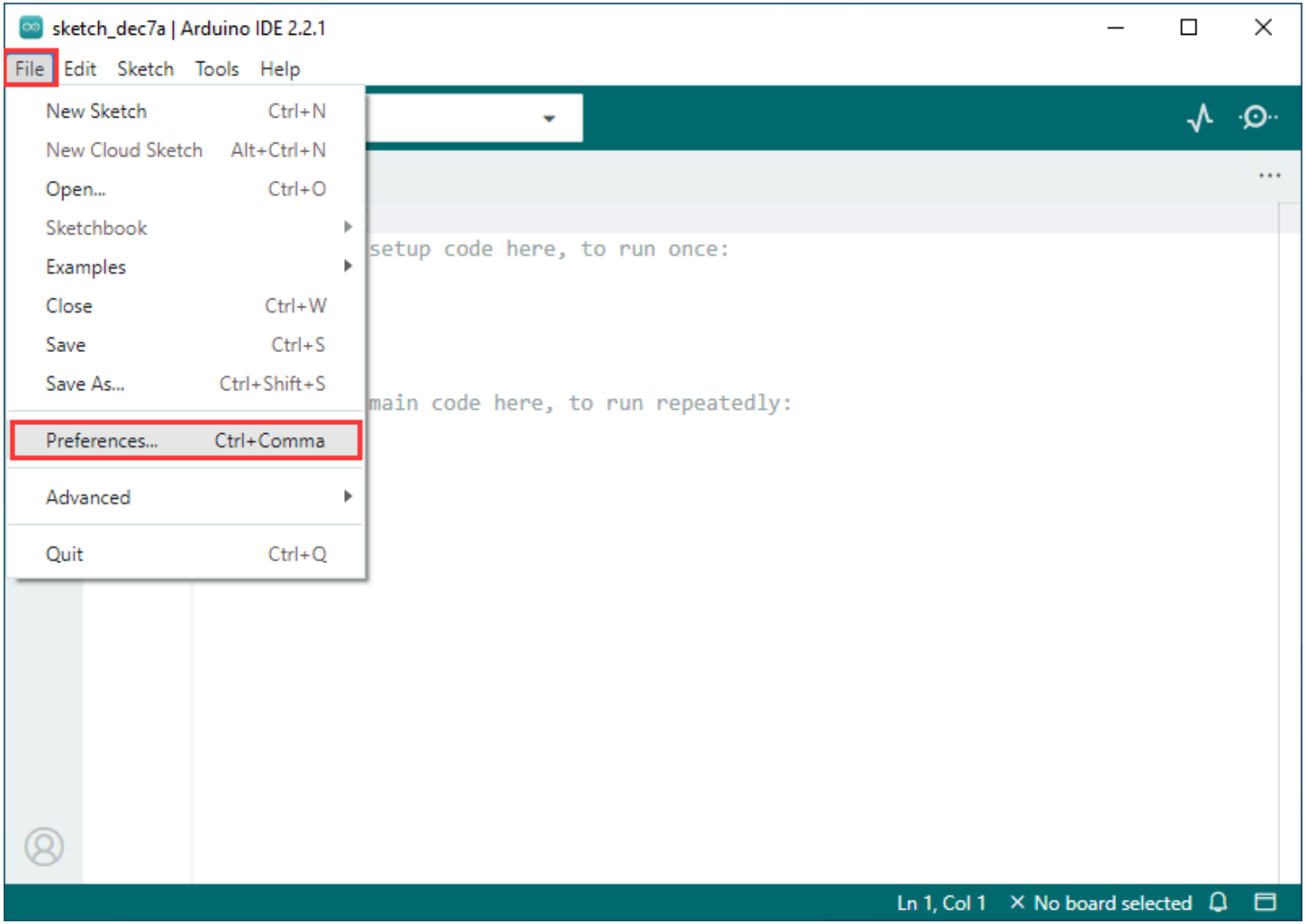
(/wiki/File:ESP32-C6-DEV-KIT-N8-Arduino01.png)

- Enter Arduino IDE after installation.



(/wiki/File:ESP32-C6-DEV-KIT-N8-Arduino02.png)

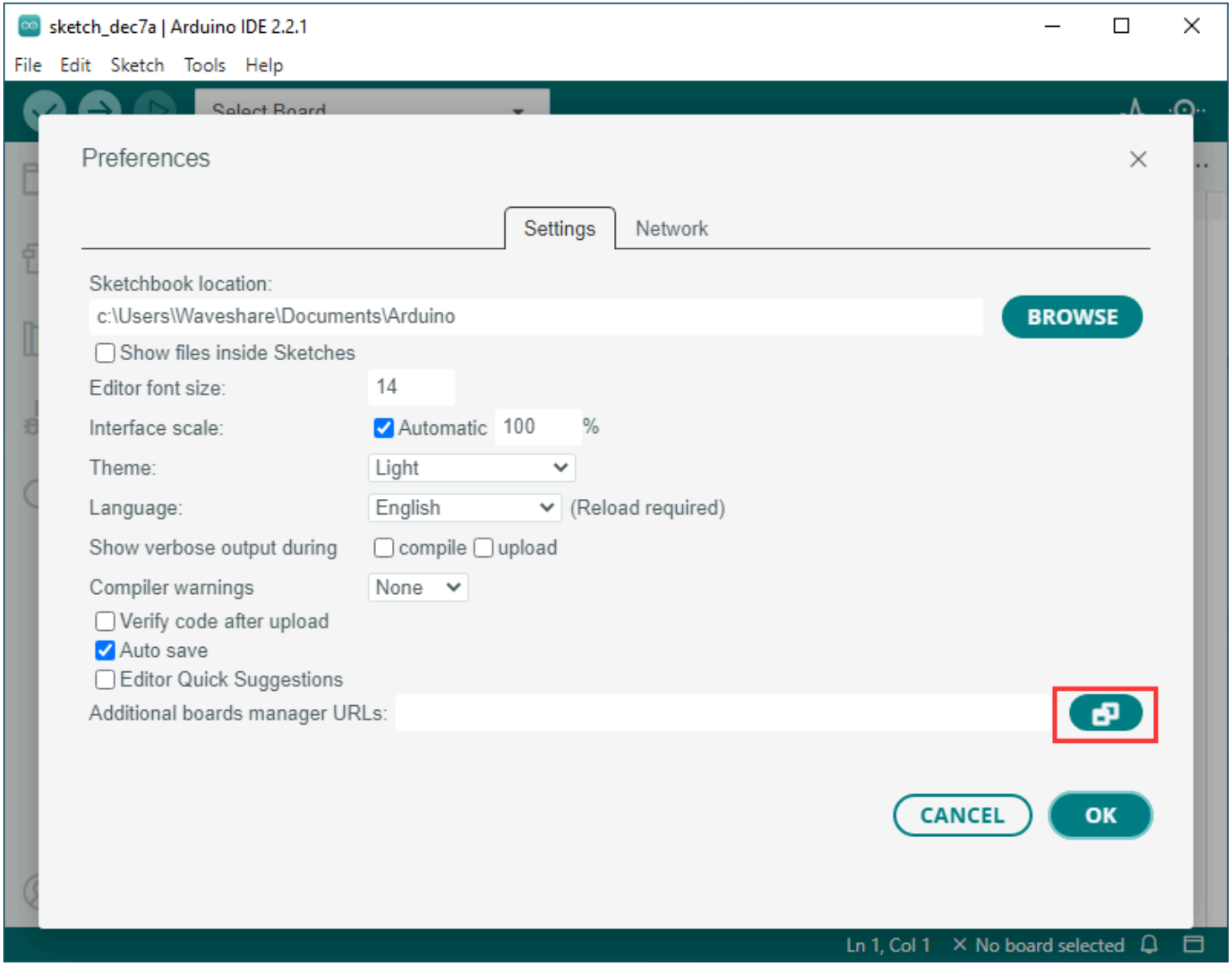
- Enter Preferences.



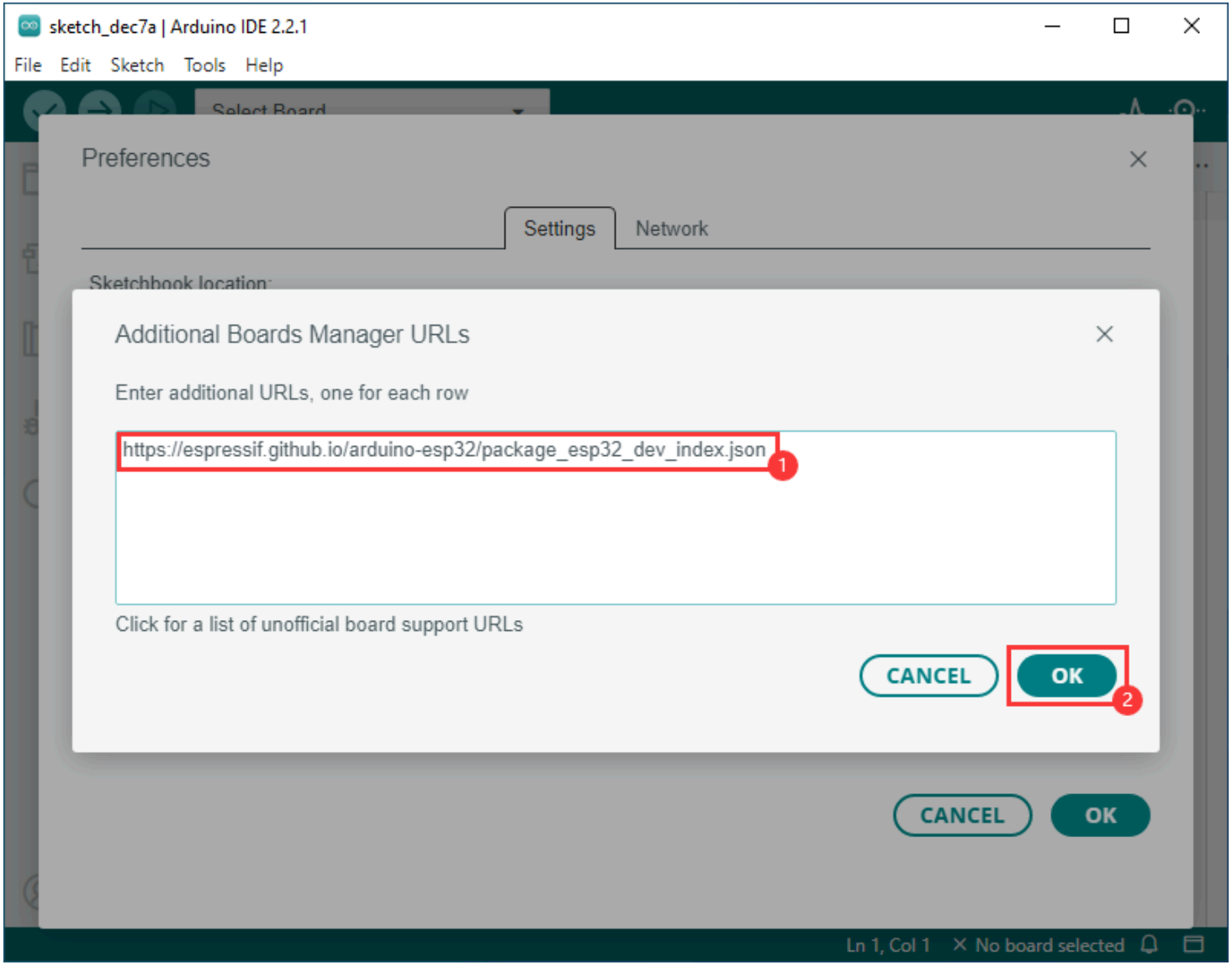
(/wiki/File:ESP32-C6-DEV-KIT-N8-Arduino03.png)

- Add JSON link:

```
https://espressif.github.io/arduino-esp32/package_esp32_dev_index.json (https://espressif.g
ithub.io/arduino-esp32/package_esp32_dev_index.json)
```

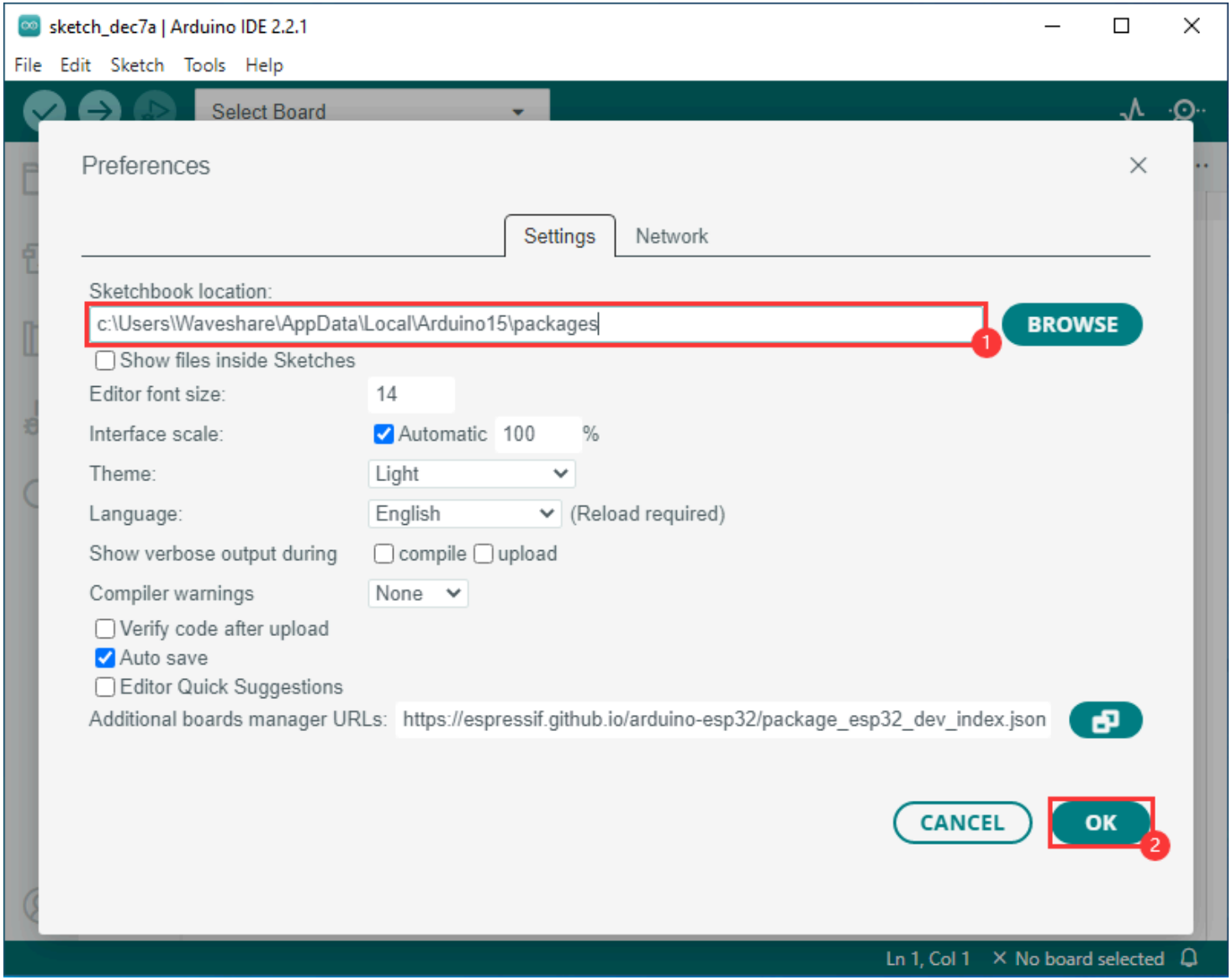


(/wiki/File:ESP32-C6-DEV-KIT-N8-Arduino04.png)



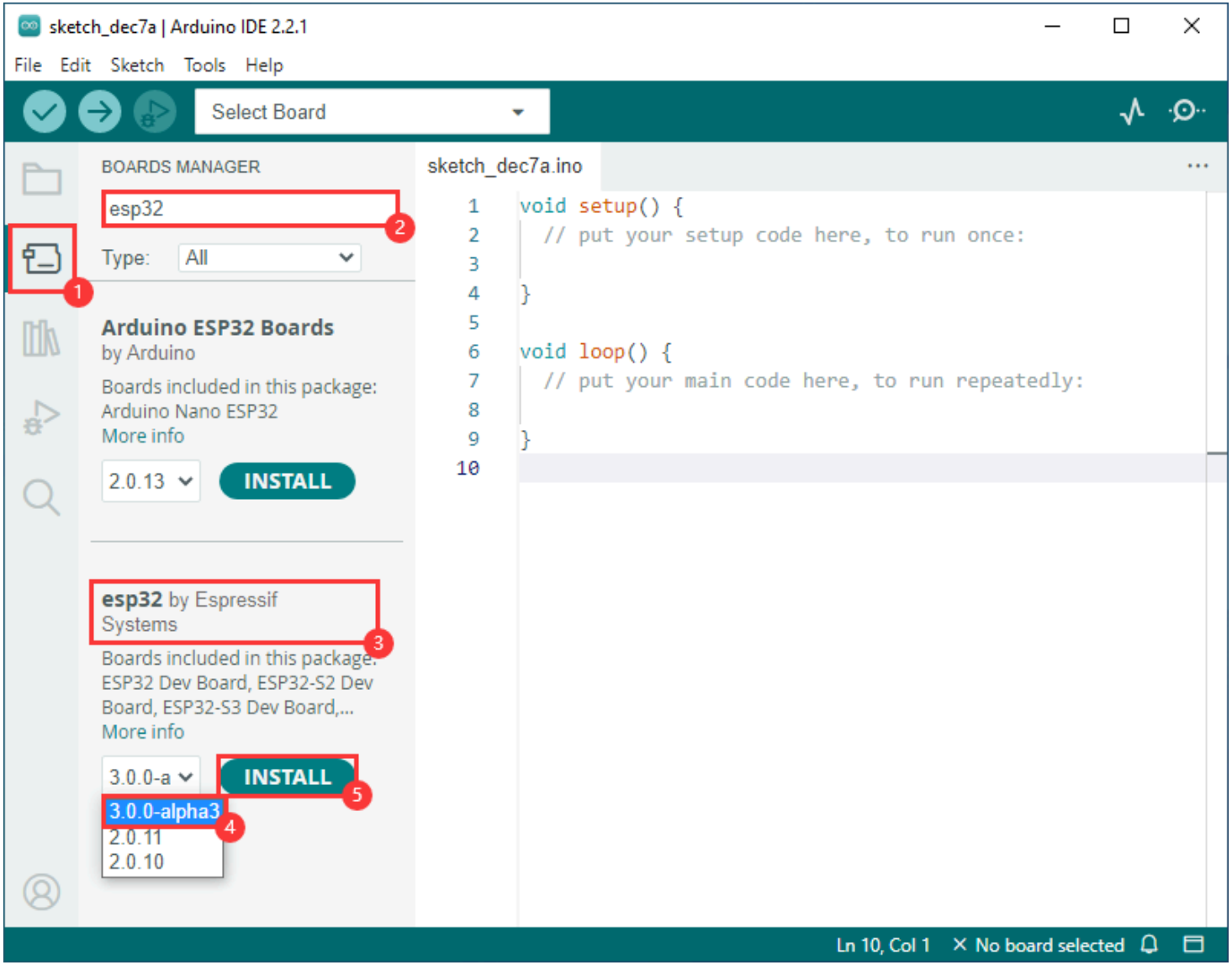
(/wiki/File:ESP32-C6-DEV-KIT-N8-Arduino05.png)

- Modify the project file folder as **C:\Users\Waveshare\AppData\Local\Arduino15\packages** (Waveshare is the username).

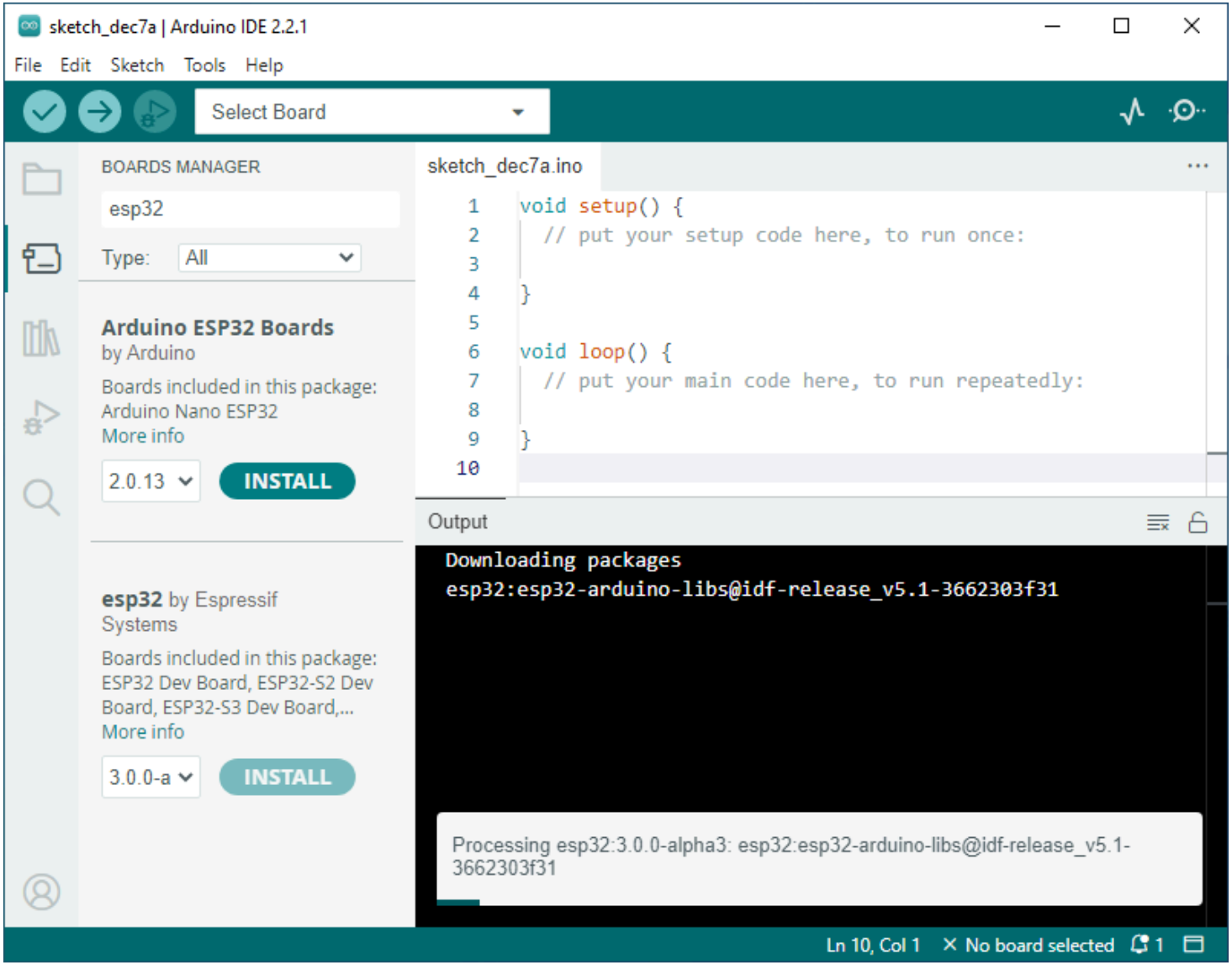


(/wiki/File:ESP32-C6-DEV-KIT-N8-Arduino06.png)

- Enter the development board manager, search for "esp32", select version 3.0.0-alpha3 under "esp32 by Espressif Systems" below, and click to install. (If installation fails, try using a mobile hotspot.)

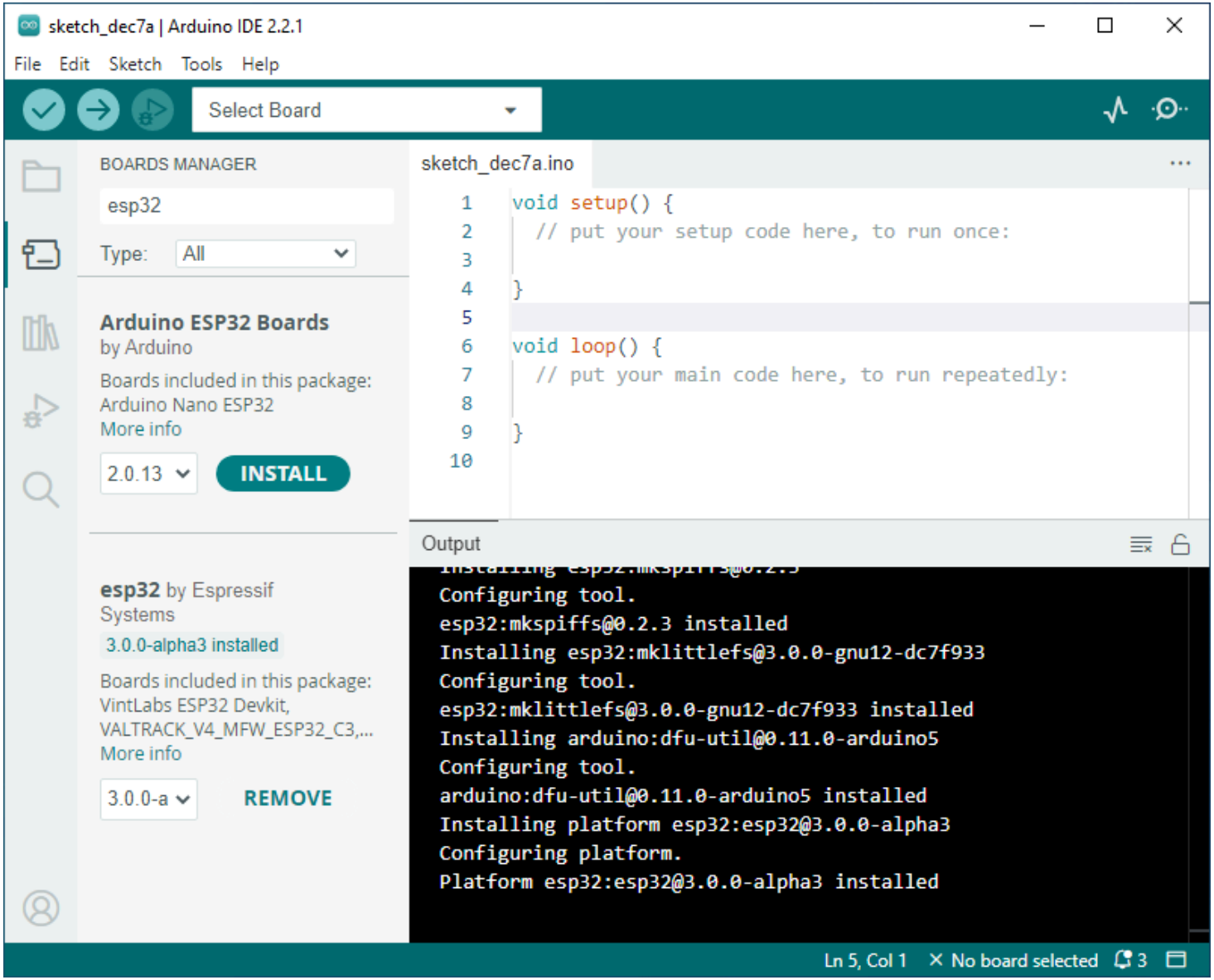


(/wiki/File:ESP32-C6-DEV-KIT-N8-Arduino07.png)



(/wiki/File:ESP32-C6-DEV-KIT-N8-Arduino08.png)

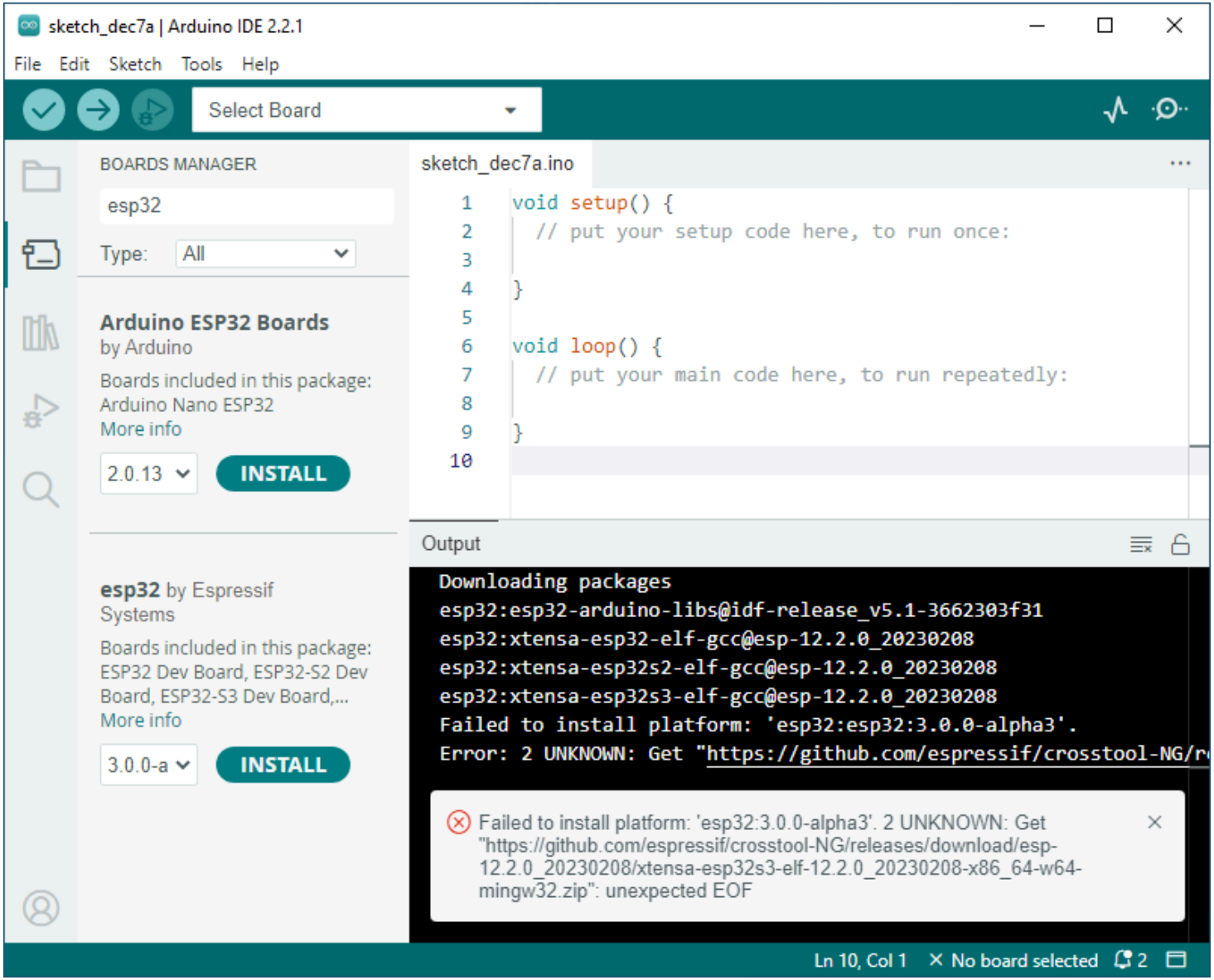
- Restart the Arduino IDE after installation, and then you can use it now.



(/wiki/File:ESP32-C6-DEV-KIT-N8-Arduino09.png)

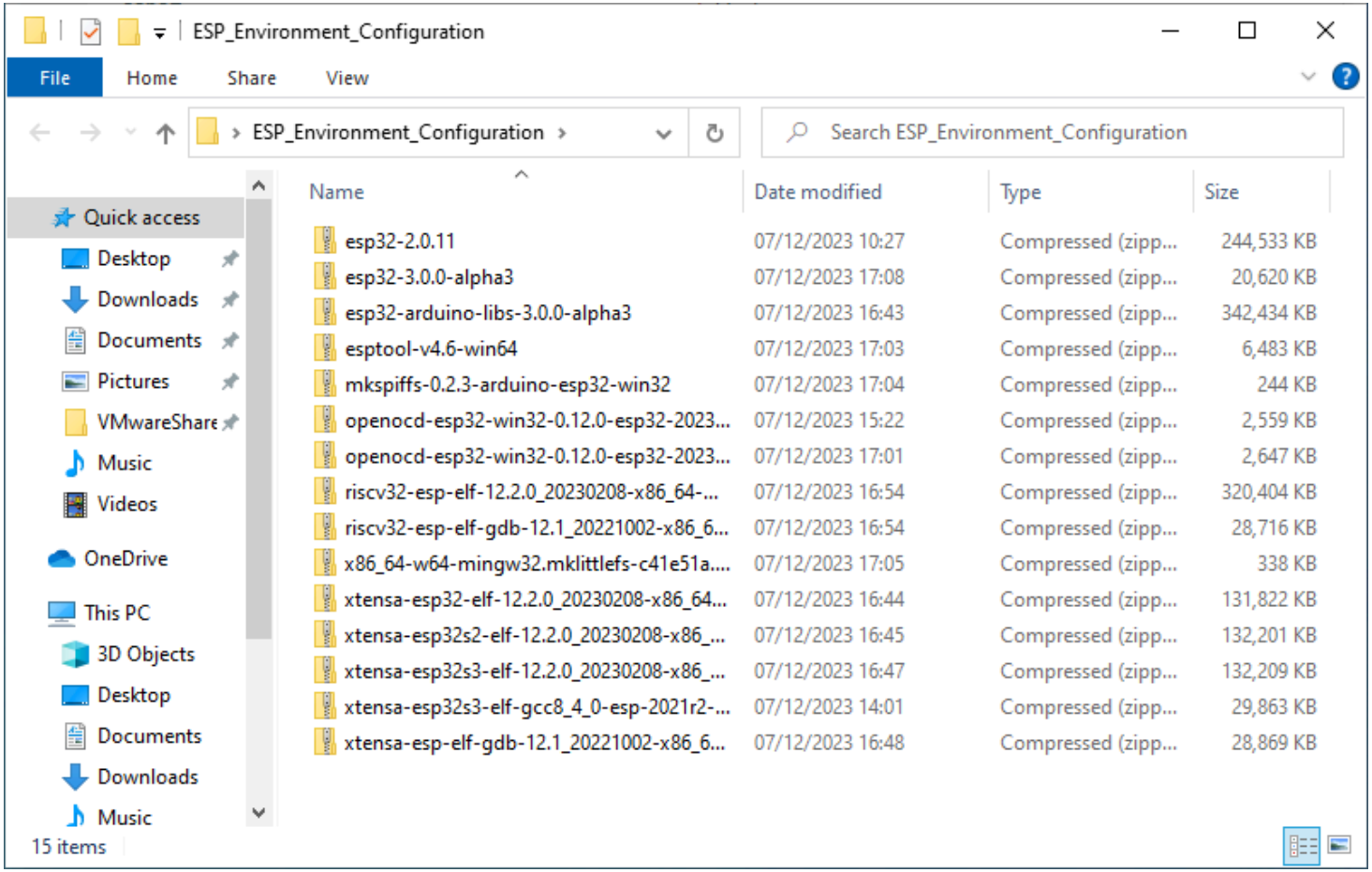
If the Installation Fails

- Failed to install version 3.0.0-alpha3:



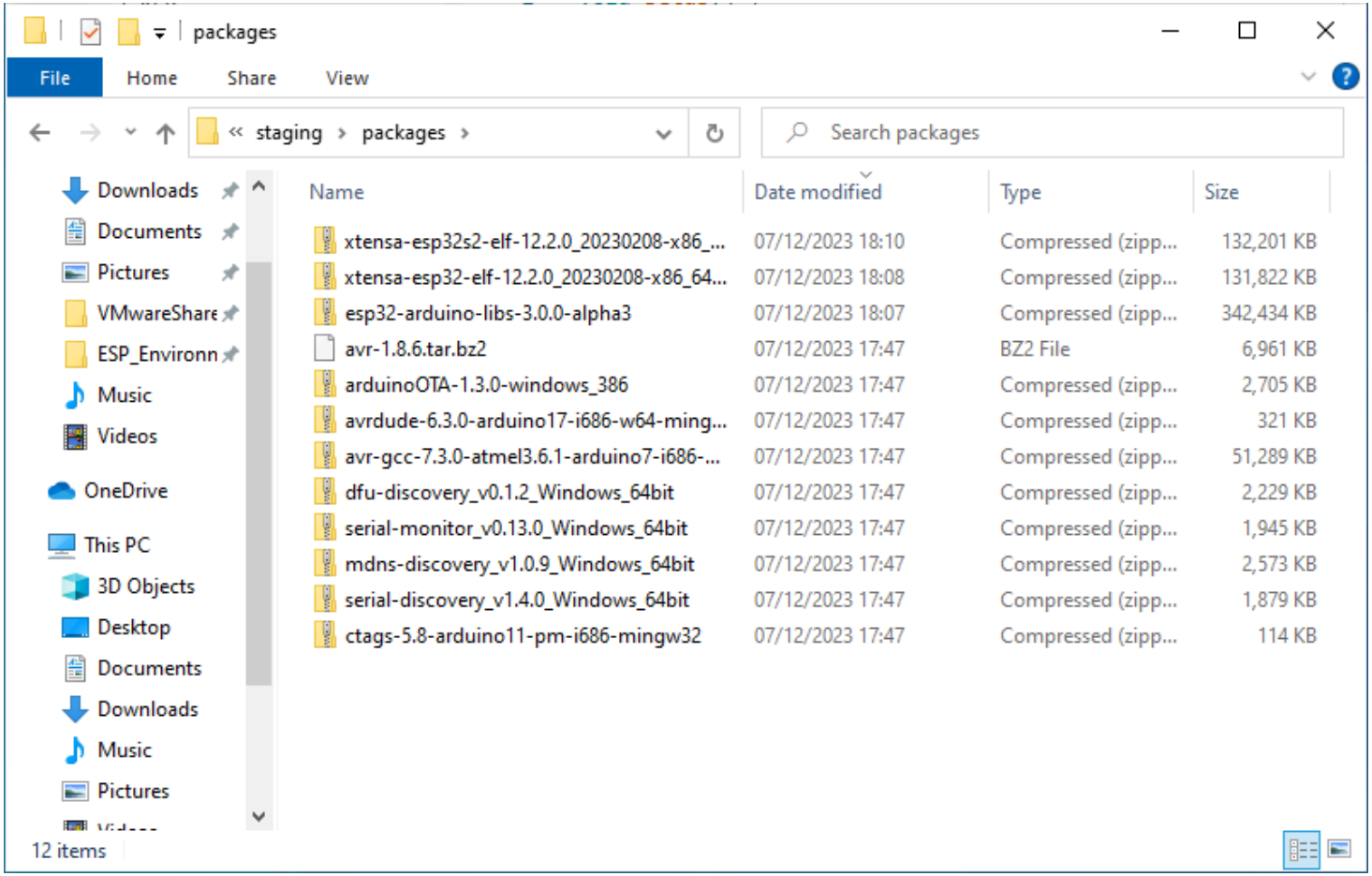
(/wiki/File:ESP32-C6-DEV-KIT-N8-install.png)

- Download the resource file (https://drive.google.com/file/d/19gMF3WHr4OreN26u2jYtGXKotRn8K16l/view?usp=sharing):



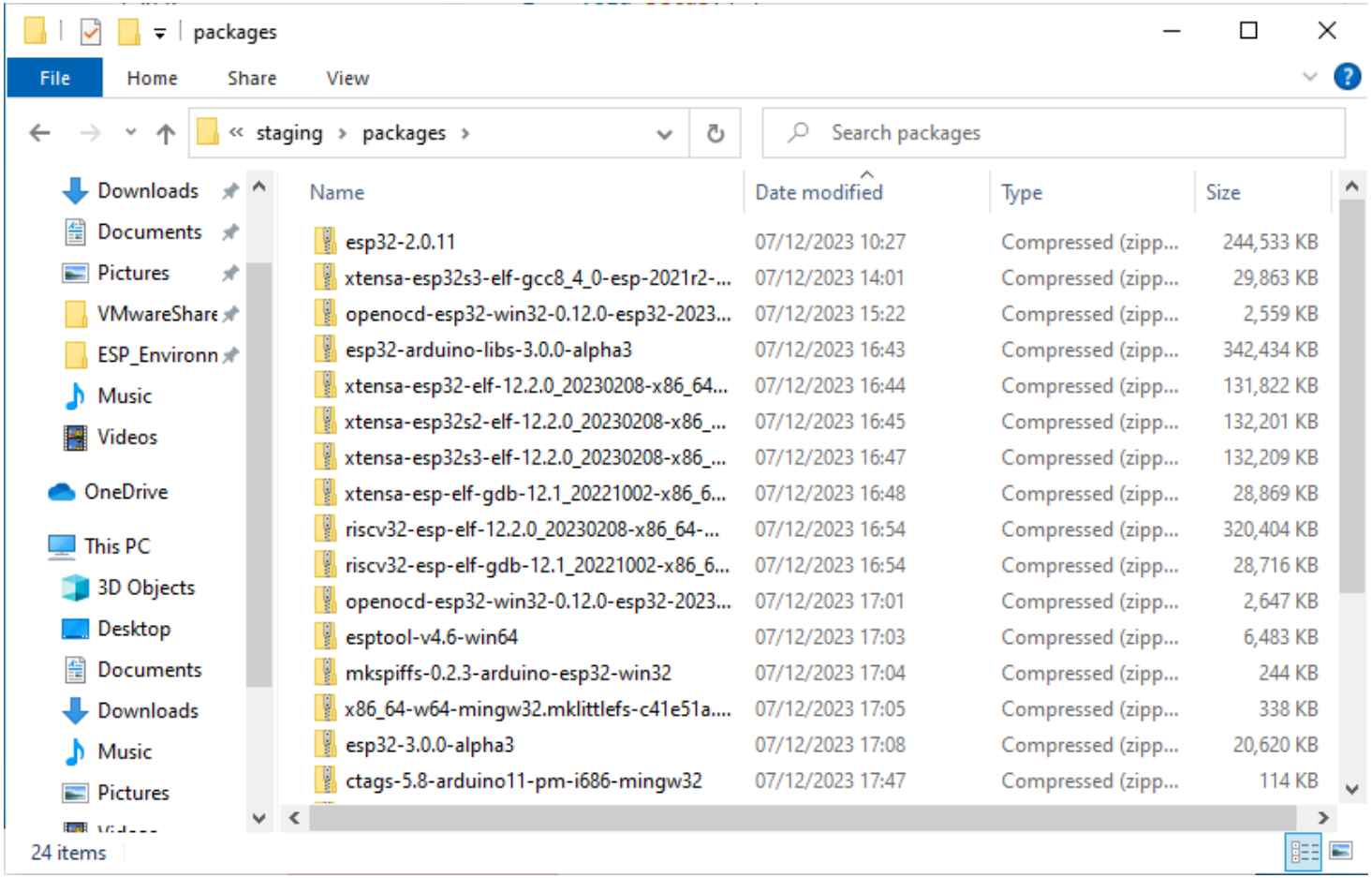
(/wiki/File:ESP32-C6-DEV-KIT-N8-install02.png)

- Click on the path "c:\Users\Waveshare\AppData\Local\Arduino15\packages" (where Waveshare is the user name of the computer, and you need to turn on Show Hidden Files).



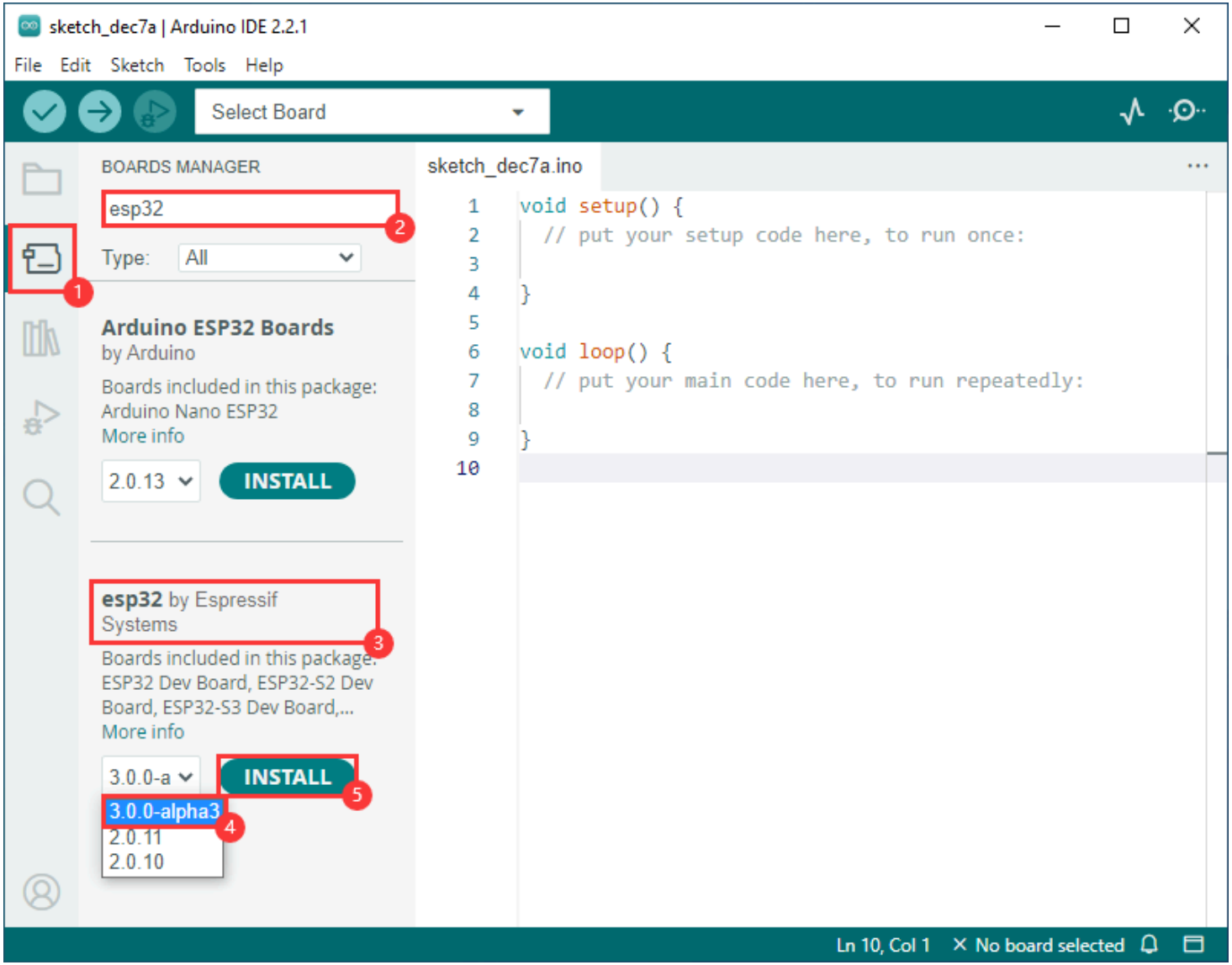
(/wiki/File:ESP32-C6-DEV-KIT-N8-install03.png)

- Unzip the downloaded files to the packages folder.



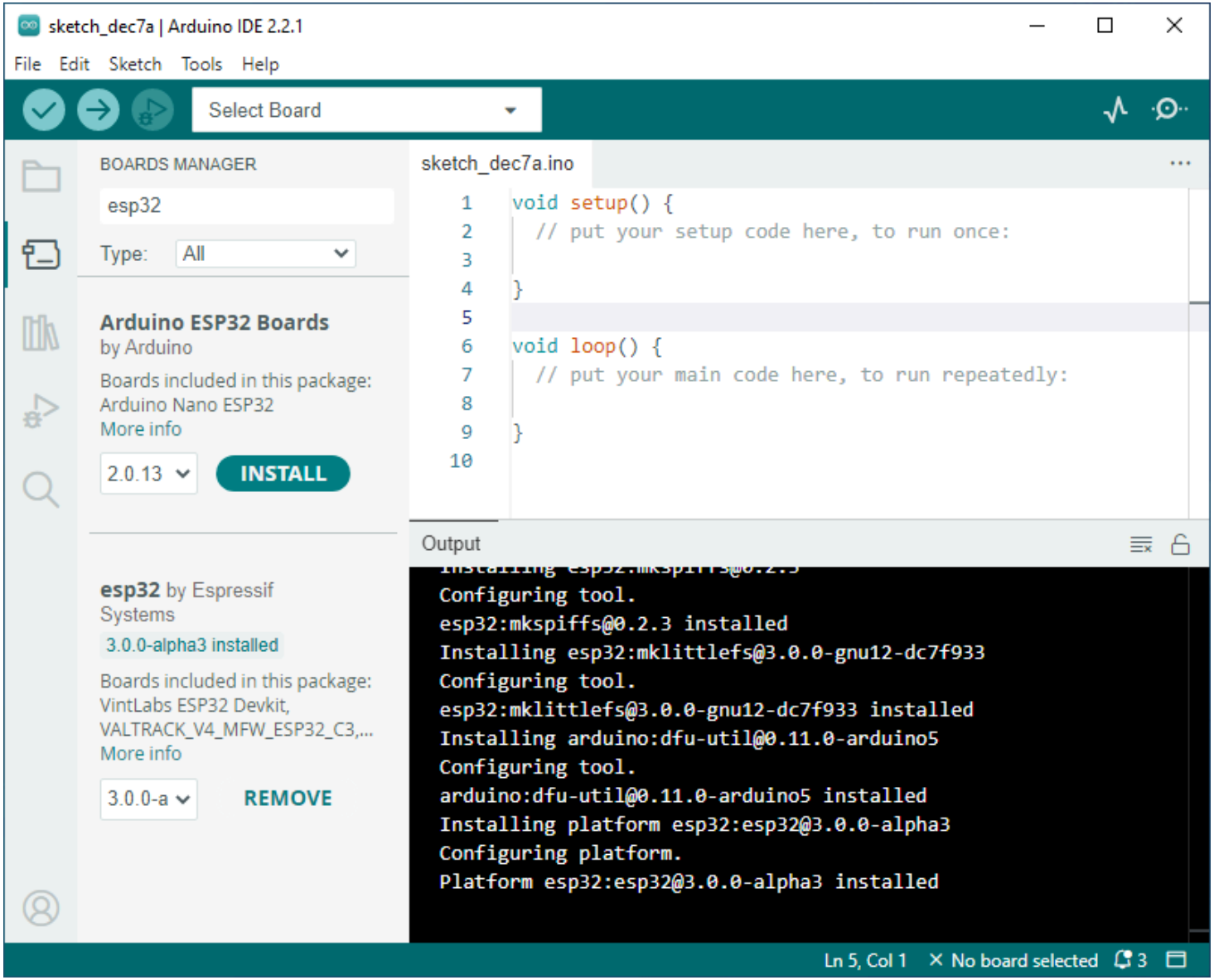
(/wiki/File:ESP32-C6-DEV-KIT-N8-install04.png)

- Install it again.



(/wiki/File:ESP32-C6-DEV-KIT-N8-Arduino07.png)

- Restart the Arduino IDE after installation and you're ready to go!

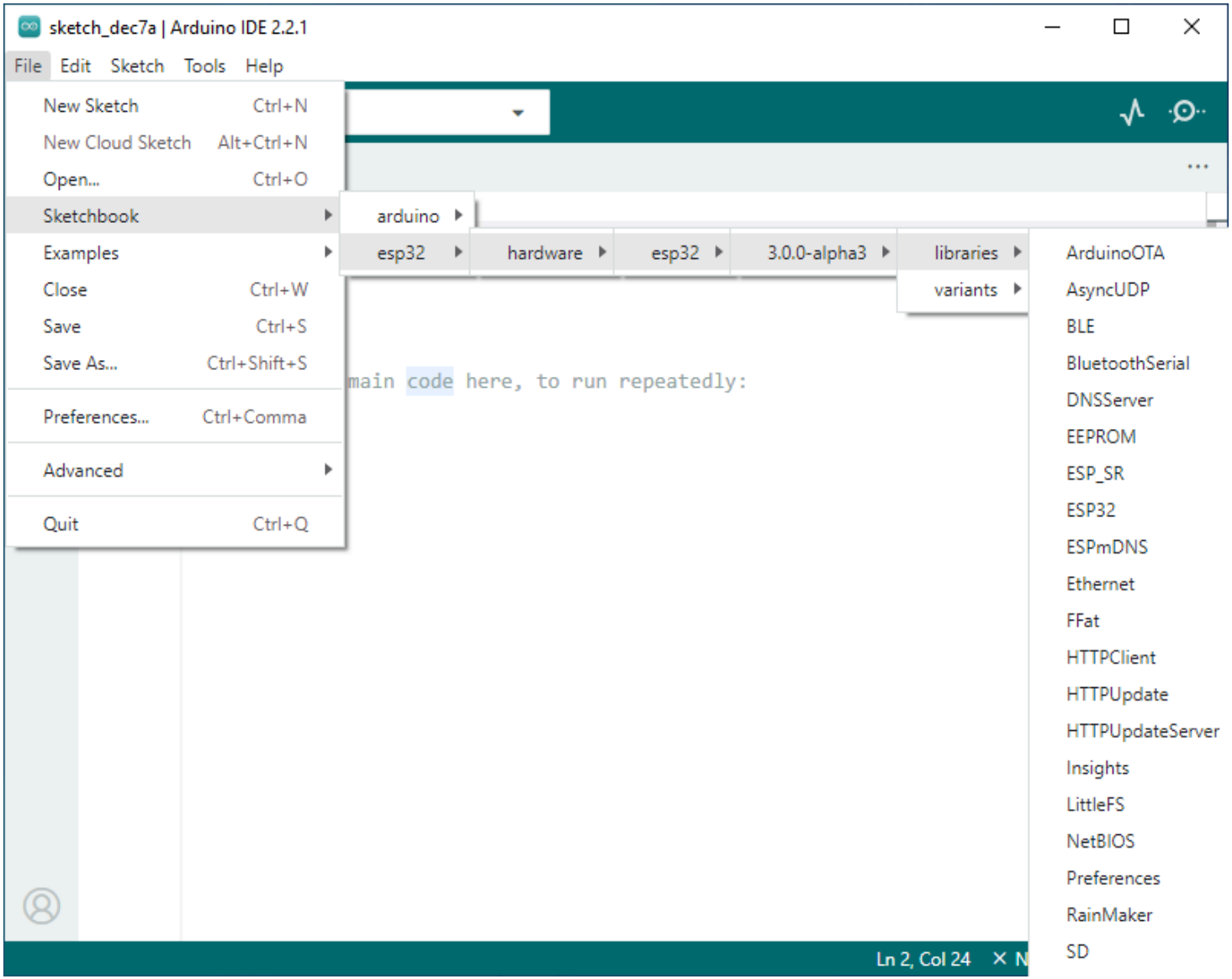


(/wiki/File:ESP32-C6-DEV-KIT-N8-Arduino09.png)

Create Example

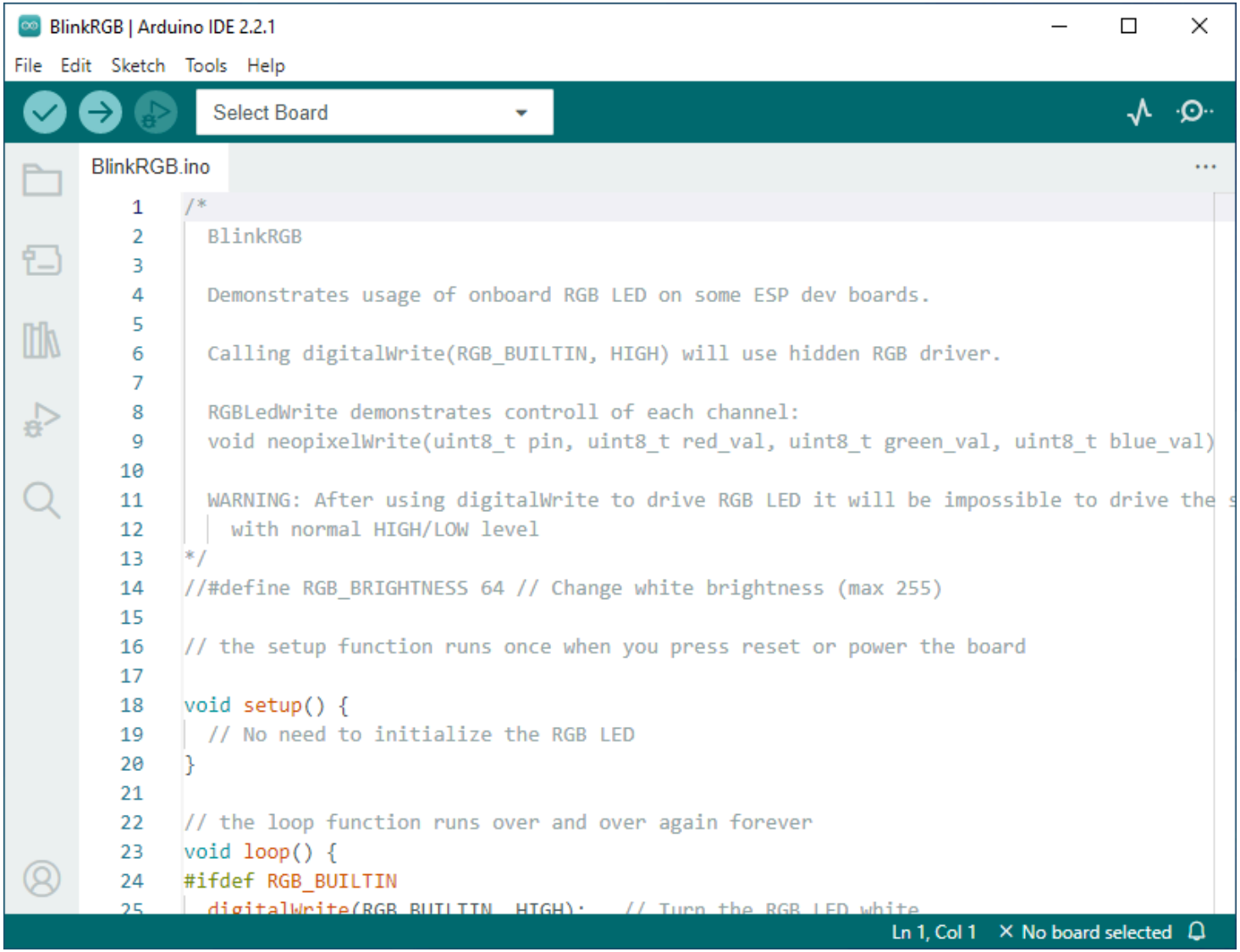
- After changing the project folder above to

c:\Users\Waveshare\AppData\Local\Arduino15\packages, you can create demos using the examples in the project folder under the File.



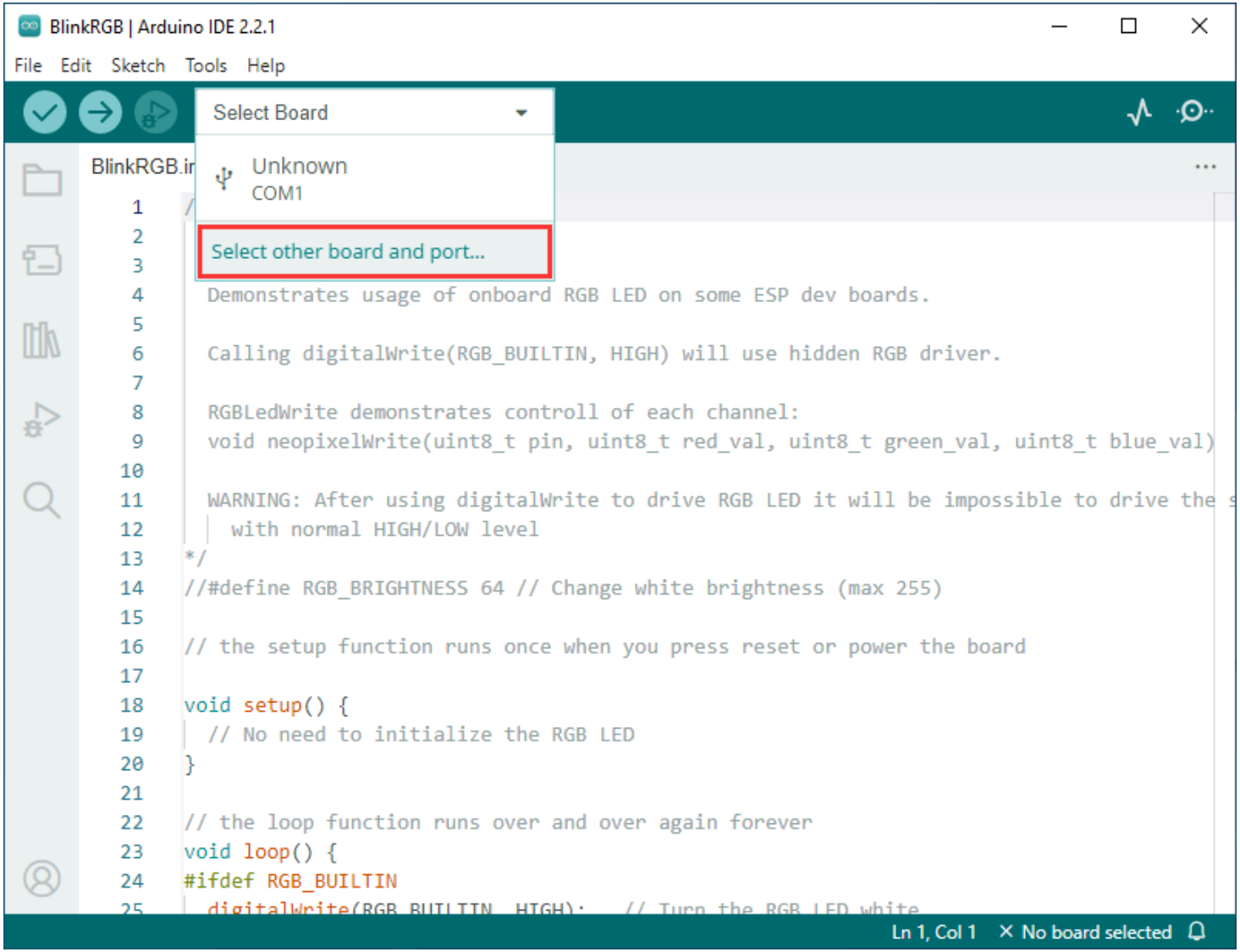
(/wiki/File:ESP32-C6-DEV-KIT-N8-Arduino10.png)

- The following is the RGB flashing example (File -> Sketchbook -> esp32 -> hardware -> esp32 -> 3.0.0-alpha3 -> libraries -> ESP32 -> examples -> BlinkRGB under GPIO).



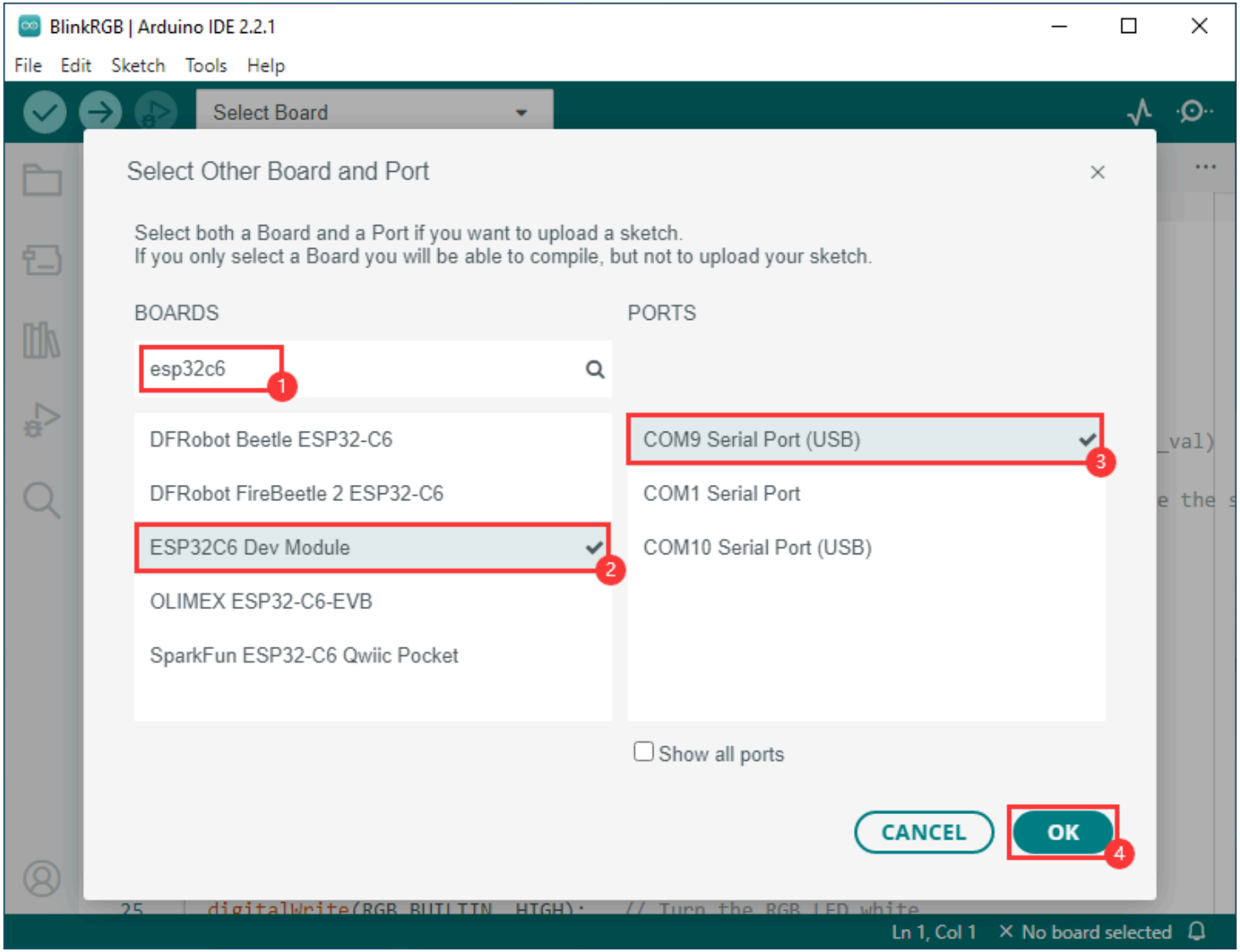
(/wiki/File:ESP32-C6-DEV-KIT-N8-Arduino11.png)

- Select the development board and port.



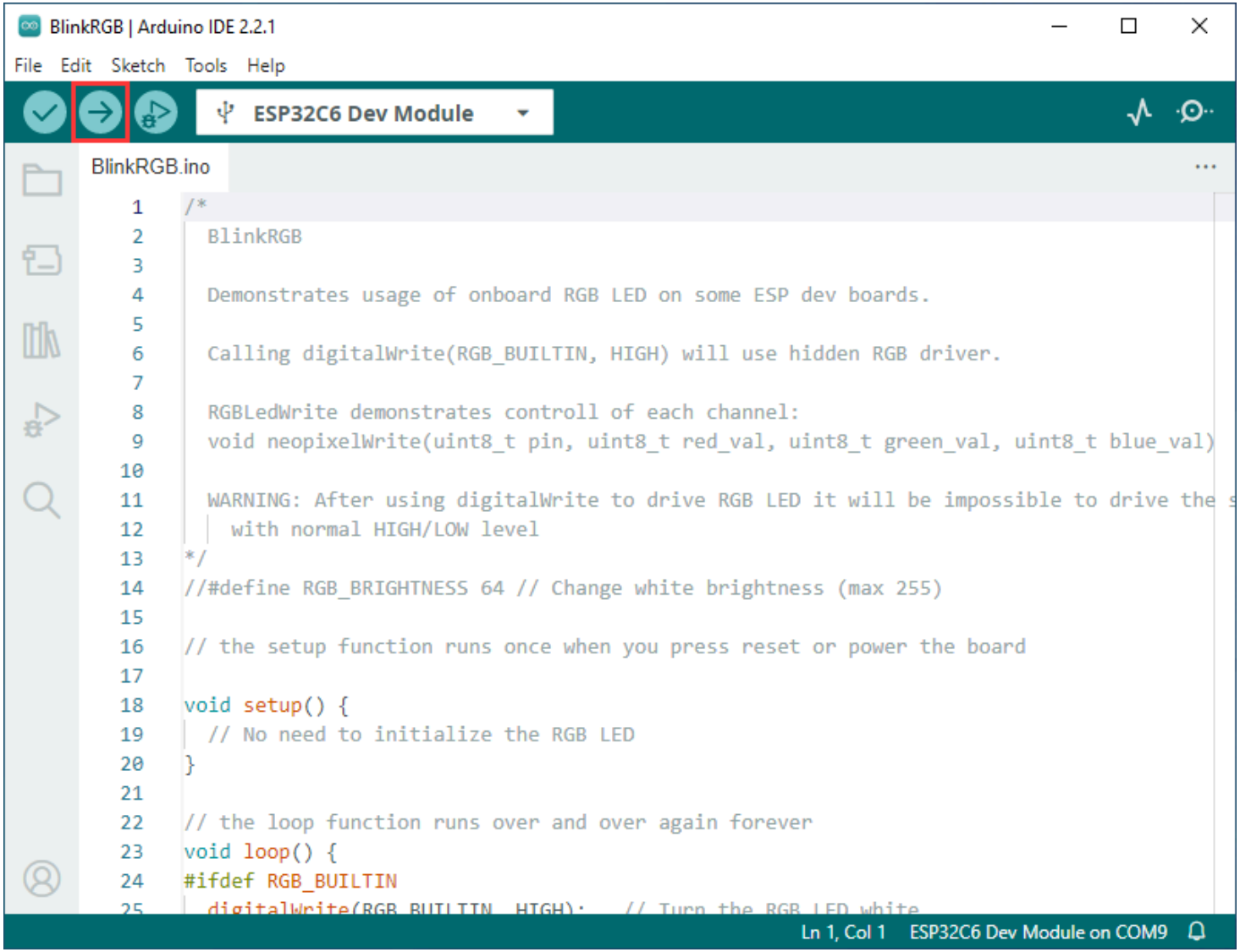
(/wiki/File:ESP32-C6-DEV-KIT-N8-Arduino12.png)

- Search for esp32c6, select ESP32C6 Dev Module and the port to download.



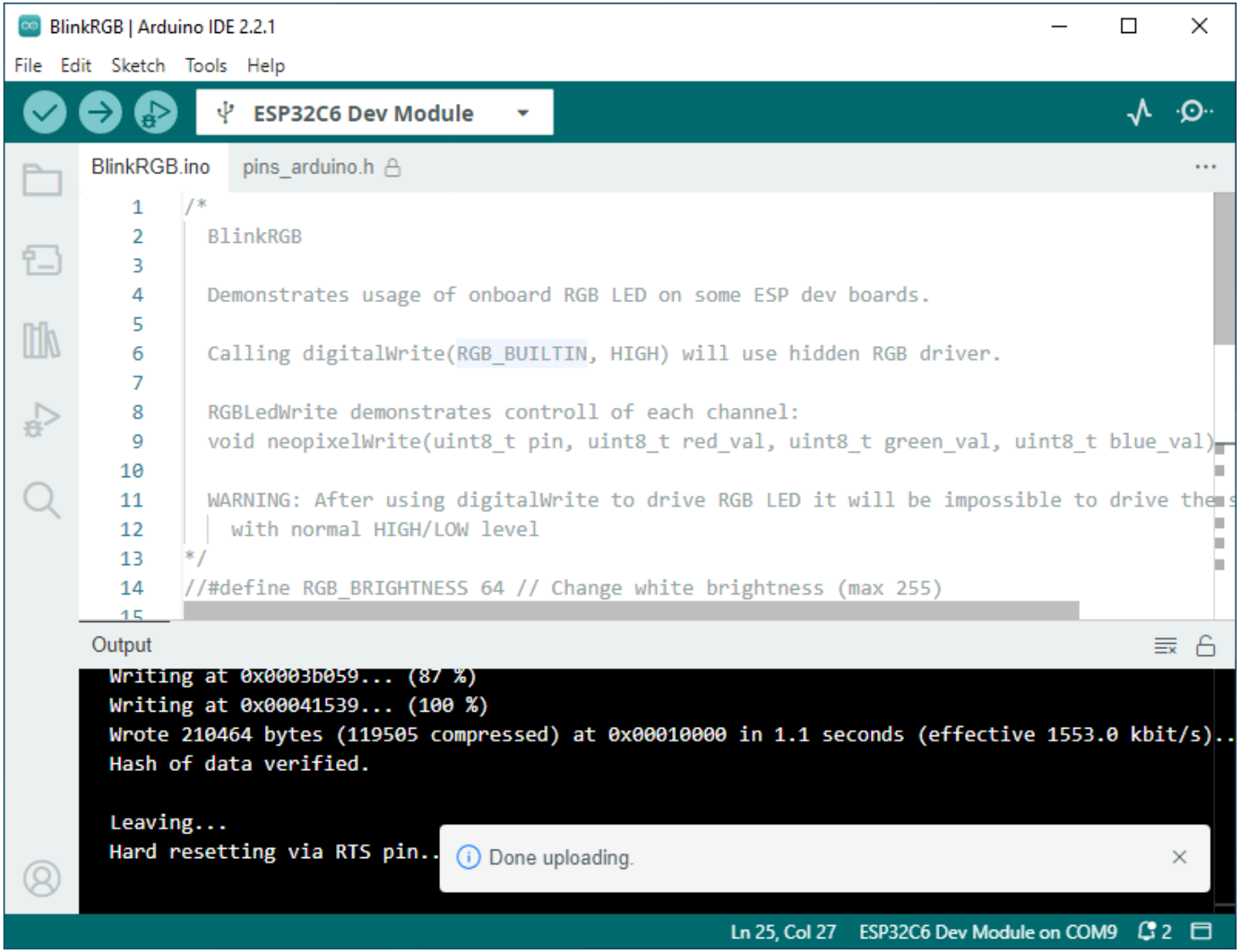
(/wiki/File:ESP32-C6-DEV-KIT-N8-Arduino13.png)

- After selecting, click to upload and Arduino IDE will start to compile and flash the demo.



(/wiki/File:ESP32-C6-DEV-KIT-N8-Arduino14.png)

- After uploading, you can see the effect on the development board.


BlinkRGB | Arduino IDE 2.2.1
File Edit Sketch Tools Help
ESP32C6 Dev Module
BlinkRGB.ino
pins_arduino.h
/*
BlinkRGB
Demonstrates usage of onboard RGB LED on some ESP dev boards.
Calling digitalWrite(RGB_BUILTIN, HIGH) will use hidden RGB driver.
RGBLedWrite demonstrates controll of each channel:
void neopixelWrite(uint8_t pin, uint8_t red_val, uint8_t green_val, uint8_t blue_val)
WARNING: After using digitalWrite to drive RGB LED it will be impossible to drive the
with normal HIGH/LOW level
*/
//#define RGB_BRIGHTNESS 64 // Change white brightness (max 255)
Output
Writing at 0x0003b059... (87 %)
Writing at 0x00041539... (100 %)
Wrote 210464 bytes (119505 compressed) at 0x00010000 in 1.1 seconds (effective 1553.0 kbit/s)..
Hash of data verified.
Leaving...
Hard resetting via RTS pin..
Done uploading.
Ln 25, Col 27 ESP32C6 Dev Module on COM9 2

(/wiki/File:ESP32-C6-DEV-KIT-N8-Arduino15.png)

Resource

Software

Compile

- VScode (https://code.visualstudio.com/download)
- Arduino IDE (https://www.arduino.cc/en/software)

CH343

- Window VCP driver (https://files.waveshare.com/upload/f/f1/CH343SER.7z)
- Android APP (https://files.waveshare.com/upload/2/22/WCHUARTDemo_V1.3.7z)

- MAC driver (https://files.waveshare.com/upload/0/04/CH34XSER_MAC.7z)

UART

- SSCOM5.13.1 (https://files.waveshare.com/wiki/LC29H(XX)-GPS-RTK-HAT/Sscom5.13.1.zip)

Flash Download Tool

- Flash Software (https://files.waveshare.com/wiki/ESP32-C6-DEV-KIT-N8/Flash_download_tool_3.9.5_0.zip)

Bluetooth Debug

- Bluetooth assist tool (https://files.waveshare.com/wiki/ESP32-C6-DEV-KIT-N8/ESP32-C6_TO_BLEAssist.ZIP)

Schematics

- Schematic (https://files.waveshare.com/wiki/ESP32-C6-DEV-KIT-N8/ESP32-C6-DEV-KIT-N8-Schematic.pdf)

Datasheet

ESP32-C6

- ESP32-C6 Series Datasheet (https://files.waveshare.com/wiki/ESP32-C6-DEV-KIT-N8/ESP32-C6_Series_Datasheet.pdf)
- ESP32-C6 Technical Reference Manual (https://files.waveshare.com/wiki/ESP32-C6-DEV-KIT-N8/ESP32-C6_Technical_Reference_Manual.pdf)

- ESP32-C6-WROOM-1 Datasheet (https://files.waveshare.com/wiki/ESP32-C6-DEV-KIT-N8/ESP32-C6-WROOM-1_Datasheet.pdf)

CH343

- CH343 Datasheet (https://www.waveshare.com/w/upload/a/a3/CH343DS1-en.PDF)

Official Document

ESP32

- ESP-IDF Official Document (https://docs.espressif.com/projects/esp-idf/en/latest/esp32c6/index.html)

FAQ

Question:After the module downloads the demo and re-downloads it, sometimes it fails to connect to the serial port, or the flashing fails?

Answer:

Method 1: Press the reset button for more than 1 second, wait for the PC to recognize the device again, and then proceed with the download.
Method 2: You can long-press the BOOT button, simultaneously press the RESET button, then release the RESET button, and finally release the BOOT button. This will put the module into download mode and can resolve most download issues.

Question:No ESP option below when setting up an environment or building a project?

Answer:

In VSCode, click the shortcut **F1**, and search for **Espressif IDF**, you will find that it is designated as an untrusted extension, set it as trusted.

Question:When the module uses UART(CH343) to download a demo, an unrecognized USB device pop-up appears?

Answer:

It has no effect on program flashing. If you want to use the corresponding COM port of USB after the pop-up of an unrecognized USB device, please disconnect the cable from the computer and reconnect the device. You can also erase the demo by software or instruction, after erasing the demo, the USB port will be recognized again.

Question:Switch to the same ESP model and encounter issues with program flashing and program execution?

Answer:

Please select the COM port and driver object again after switching ESP, then compile and flash.

Question:After powering up the module, the recognized serial devices and USB ports keep resetting and restarting?

Answer:

Check whether the power supply voltage for the USB port is less than 5V, in general, if it is 4.9V or more, the module's two USB ports can be used normally. If it is lower than 4.9V, the power supply may be insufficient and the USB port may disconnect. In this case, you should replace it with a USB port with sufficient voltage.

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/)