ESP32-S3-Touch-LCD-4.3B

From Waveshare Wiki
Jump to: navigation, search

Overview

<table>
<tr><td align="center">ESP32-S3-Touch-LCD-4.3B</td></tr>
</table>

Introduction

ESP32-S3-Touch-LCD-4.3B is a microcontroller development board with 2.4GHz WiFi and BLE 5 support, and integrates high-capacity Flash and PSRAM. The onboard 4.3-inch capacitive touch screen can smoothly run GUI demos such as LVGL. Combined with various peripheral interfaces (such as CAN, I2C and RS485), it is suitable for the quick development of the HMI and other ESP32-S3 applications. With a wide range of functions and interfaces, it can meet power consumption requirements in Internet of Things (IoT), mobile devices, smart home and other applications.

(https://www.waveshare.com/esp32-s3-touch-lcd-4.3b.htm)

ESP32-S3-N16R8
Type C USB

Features

- Equipped with Xtensa 32-bit LX7 dual-core processor, up to 240MHz main frequency.
- Supports 2.4GHz Wi-Fi (802.11 b/g/n) and Bluetooth 5 (LE), with an onboard antenna.
- Built-in 512KB of SRAM and 384KB ROM, with onboard 8MB PSRAM and 8MB Flash.
- Onboard 4.3inch capacitive touch display, 800×480 resolution, 65K color.
- Supports capacitive touch control via I2C interface, 5-point touch with interrupt support.
- Onboard CAN, RS485, I2C interface, and Micro SD card slot, integrate full-speed USB port.
- Supports flexible clock, module power supply independent setting, and other controls to realize low power consumption in different scenarios.

Hardware Description

Onboard Interface

- **CAN interface**: Transceiver control, data analysis, acquisition and monitoring of CAN bus networks.
- **I2C interface**: ESP32-S3 provides multi-lane hardware I2C, currently uses GPIO8(SDA) and GPIO9(SCL) pins as I2C bus for loading IO expansion chip, touch interface and I2C interface.
- **RS485 interface**: the development board onboard RS485 interface circuits for directly connecting to RS485 device communication, and support automatic switching of RS485 circuit transceiver mode.
- **Isolated IO interface**: Isolated IO is composed of digital output, digital input and input signal common terminal, IO level up to 5~36V.
- **PH2.0 battery header**: The development board utilizes the efficient charge and discharge management chip CS8501. It can boost a single-cell lithium battery to 5V. Currently, the charging current is set at 580mA, and users can modify the charging current by replacing the R45 resistor. For more details, you can refer to schematic (https://files.waveshare.com/wiki/ESP32-S3-Touch-LCD-4.3B/ESP32-S3-Touch-LCD-4.3B-Sch.pdf).

PIN Connection

ESP32-S3-WROOM-x	LCD	USB	SD	UART	CAN	RTC	DO/DO
GPIO0	G3						
GPIO1	R3						
GPIO2	R4						
GPIO3	VSYNC						
GPIO4	TP_IRQ						
GPIO5	DE						
GPIO6							
GPIO7	PCLK						
GPIO8	TP_SDA						
GPIO9	TP_SCL						
GPIO10	B7						
GPIO11			MOSI				
GPIO12			SCK				
GPIO13			MISO				
GPIO14	B3						
GPIO15					CANTX		
GPIO16					CANRX		
GPIO17	B6						
GPIO18	B5						
GPIO19		USB_DN					
GPIO20		USB_DP					
GPIO21	G7						
GPIO38	B4						
GPIO39	G2						
GPIO40	R7						
GPIO41	R6						
GPIO42	R5						
GPIO43				RS485_RX			
GPIO44				RS485_TX			
GPIO45	G4						
GPIO46	HSYNC						
GPIO47	G6						
GPIO48	G5						
CH422G	-	-	-	-	-	-	-
EXIO0							DI0
EXIO1	TP_RST						
EXIO2	DISP						
EXIO3	LCD_RST						
EXIO4			SD_CS				
EXIO5							DI1
OD0							DO0
OD1							DO1

Hardware Connection

I2C interface terminal
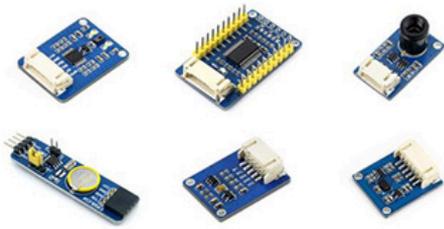
CAN terminal

RS485 terminal
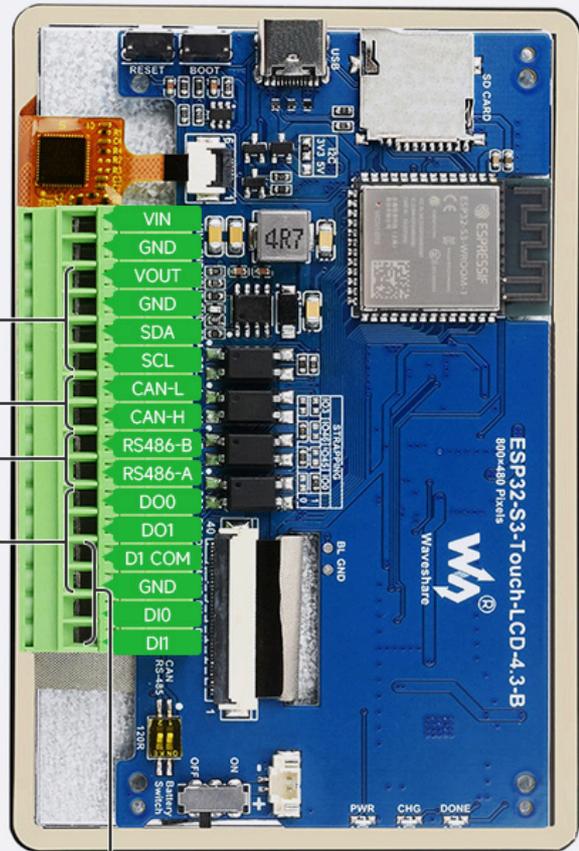
DO terminal

DI terminal

Active digital input (wet contact)

NPN proximity sensor
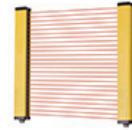
IR curtain sensor

Passive digital input (dry contact)

ON/OFF switch

VIN
GND
VOUT
GND
SDA
SCL
CAN-L
CAN-H
RS486-B
RS486-A
DO0
DO1
DI COM
GND
DI0
DI1

ESP32-S3-Touch-LCD-4.3-B
800×480 Pixels

Waveshare

Digital output

GND		5-36V − +
DI COM		
DO1		LOAD
DO0		LOAD

Digital output

Digital input

DI1		
DI0		
GND		
DI COM		5-36V

Digital input wet

or

Digital input

DI1		
DI0		
GND		
DI COM		
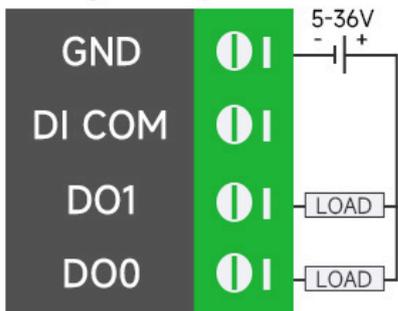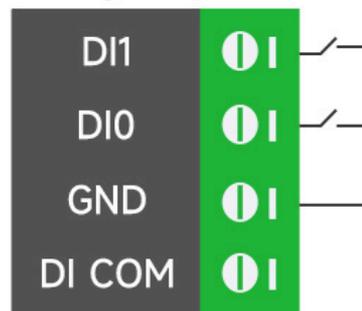
Digital input dry contact
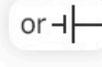
(/wiki/File:ESP32-S3-Touch-LCD-4.3B-details-9.jpg)

- The development board supports downloading the demo through USB. If the port cannot be identified, please enter the boot mode (press the boot key, connect the board to your PC and then release the boot key). After downloading the demo, press RESET key to run the demo.
- Please pay attention not to put metal or plastic materials to close the PCB antenna.
- The development board adapts the peripheral pin headers such CAN/I2C/RS485/isolated IO interface through 3.5mm screw terminal.
- The 4.3inch LCD occupies most of GPIO pin headers, and the development board adopts CH422G chip to expand IO for reset, backlight control, etc.
- CAN and RS485 peripheral interface does not connect to 120 ohm resistor through the switch by default, and you can switch it on to enable the terminal resistor connection.
- The TF card adopts the SPI connection, please note that the SD_CS pin needs to be driven by the EXIO4 of CH422G chip.
- The isolated IO is controlled by the CH422G chip. For more details, you can refer to IO_Test example. To learn the driving principle, you can refer to #Demo.

Notes

- Currently, running the LVGL benchmark example on ESP-IDF v5.3 with a single core has an average frame rate limit of 26, corresponding to an interface frame rate of 41 (PCLK 21 MHz). Before compiling, you need to configure ESP32 and LVGL through menuconfig.

```
CONFIG_FREERTOS_HZ=1000
CONFIG_ESP_DEFAULT_CPU_FREQ_MHZ_240=y
CONFIG_ESPTOOLPY_FLASHMODE_QIO=y
CONFIG_ESPTOOLPY_FLASHFREQ_120M=y [should align with PSRAM]
CONFIG_SPIRAM_MODE_OCT=y
CONFIG_IDF_EXPERIMENTAL_FEATURES=y and CONFIG_SPIRAM_SPEED_120M=y [should align with FLASH]
CONFIG_SPIRAM_FETCH_INSTRUCTIONS=y
CONFIG_SPIRAM_RODATA=y
CONFIG_ESP32S3_DATA_CACHE_LINE_64B=y
CONFIG_COMPILER_OPTIMIZATION_PERF=y
#The following LVGL configuration can improve the frame rate (LVGL v8.3):
#define LV_MEM_CUSTOM 1 or CONFIG_LV_MEM_CUSTOM=y
#define LV_MEMCPY_MEMSET_STD 1 or CONFIG_LV_MEMCPY_MEMSET_STD=y
#define LV_ATTRIBUTE_FAST_MEM IRAM_ATTR or CONFIG_LV_ATTRIBUTE_FAST_MEM=y
```

- For more details about LCD and LVGL performance, you can refer to this document (https://files.waveshare.com/wiki/ESP32-S3-Touch-LCD-7/Performance.pdf).
- The PH2.0 lithium battery socket only supports single-cell 3.7V lithium batteries. Do not use multiple battery packs for charging and discharging simultaneously. It is recommended to use a single-cell battery with a capacity of below 2000mAh.
- Please note that the CH422G of the board and touch functionality has used the following slave addresses, so do not use the I2C devices with the same slave addresses as below:

Buy at : https://hubtronics.in/esp32-s3-touch-lcd-4.3b-box

```
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
10: -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
20: 20 21 22 23 24 25 26 27 -  -  -  -  -  -  -  -
30: 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f
40: -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
50: -  51 -  -  -  -  -  -  -  -  -  -  -  5d -
60: -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
70: -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
```

Dimensions
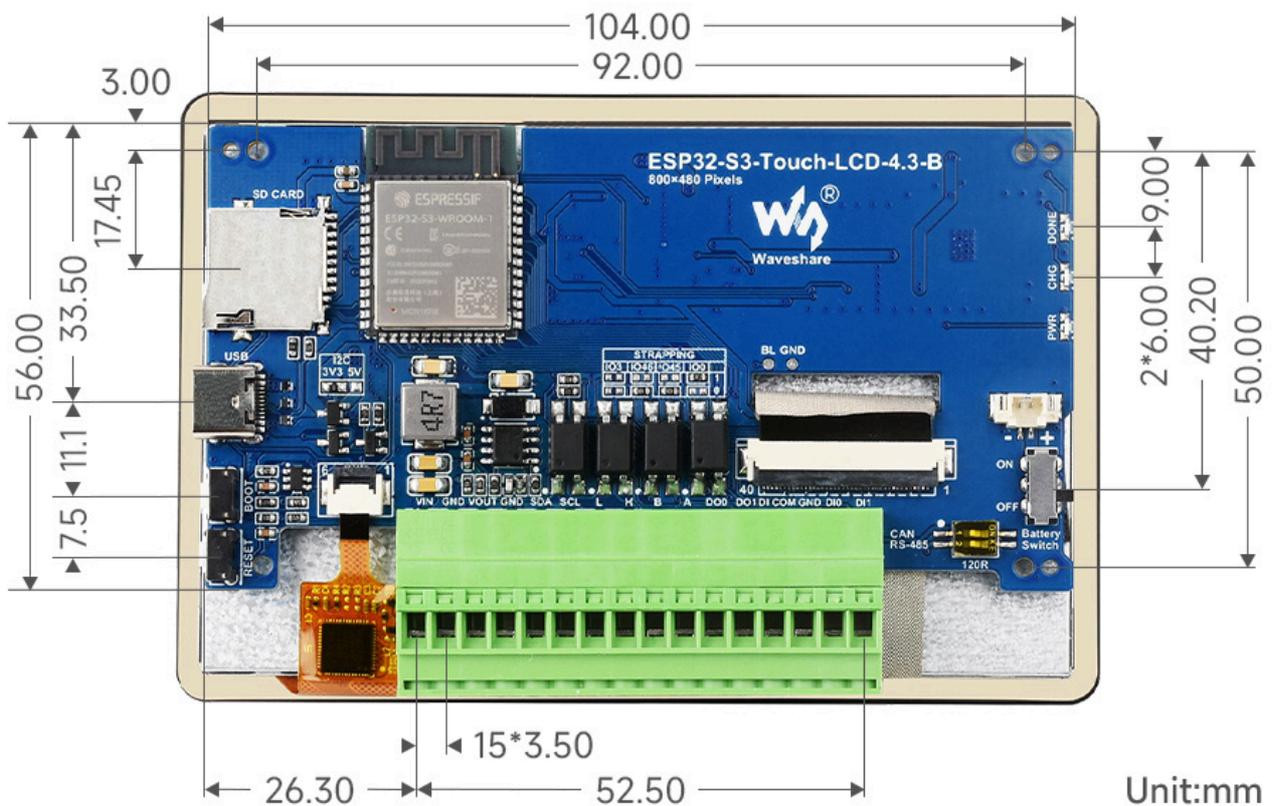
(/wiki/File:ESP32-S3-Touch-LCD-4.3B-details-size.jpg)

Environment Setting

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

these three development approaches:

- Official C/C++ library installation:

    - ESP32 series Arduino (https://docs.espressif.com/projects/arduino-esp32/en/latest/installing.html) development tutorial.

    - ESP32 series ESP-IDF (https://docs.espressif.com/projects/esp-idf/en/stable/esp32s2/get-started/index.html) development tutorial.

- Environment setting is supported on Windows 10. Users can select Arduino/Visual Studio Codes (ESP-IDF) as IDE to develop. For Mac/Linux, users can refer to official introduction (https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/index.html).

ESP-IDF

- It is recommended to develop with the VSC plug-in.

Develop with VSCode Plug-in

Install VSCode

- 1. Open the download page (https://code.visualstudio.com/download) of the official VSCode website, and select the corresponding system and system bit to download.
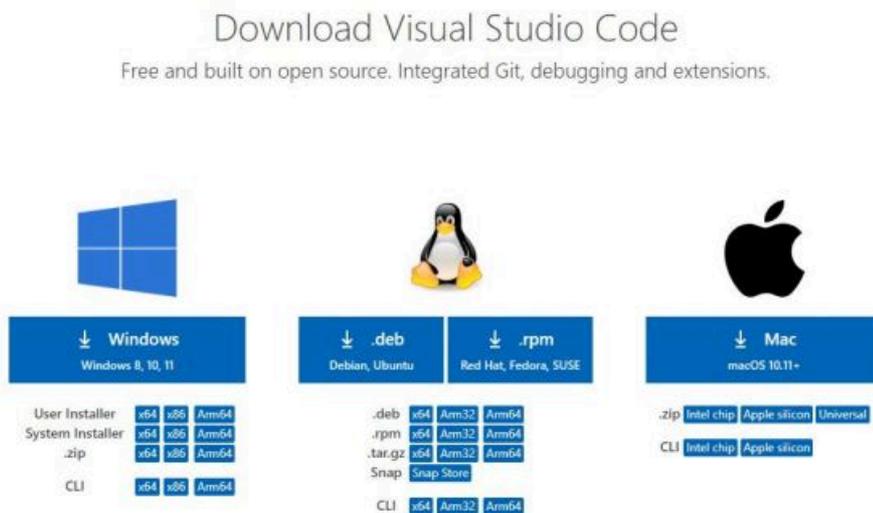
(/wiki/File:ESP32-S3-Pico_05.jpg)

- 2. After running the installation package, the rest can be installed by default, but here for the subsequent experience, it is recommended to check boxes 1, 2, and 3.

    - After the first and second items are enabled, you can open VSCode directly by right-clicking files or directories, which can improve the subsequent user experience.

    - After the third item is enabled, you can select VSCode directly when you choose how to open it.

Buy at : https://hubtronics.in/esp32-s3-touch-lcd-4.3b-box
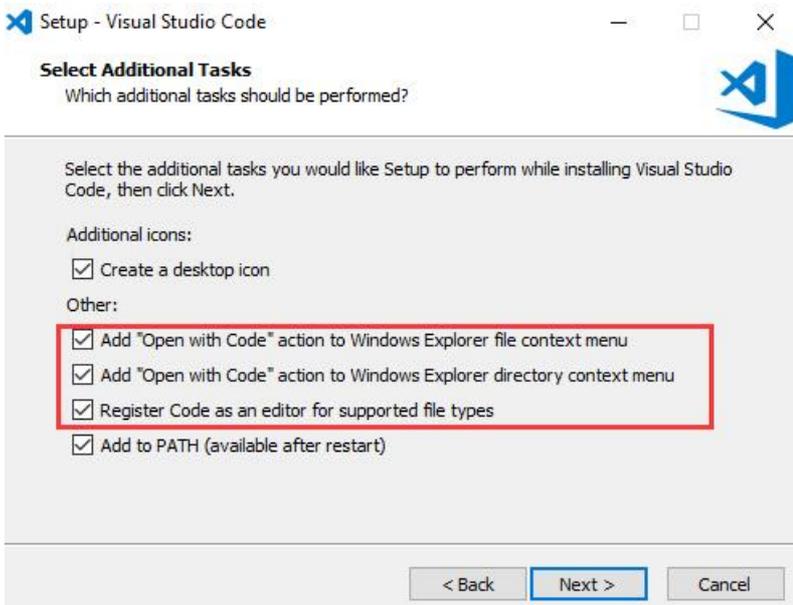
(/wiki/File:ESP32-S3-Pico_06.jpg)

Install Espressif IDF Plug-in

- Note: The latest version of the current plug-in is V1.7.1, for a consistent experience, users can choose the same version as us.

1. Open VSCode and use the shortcut key Shift + Ctrl + X to enter the plugin manager.
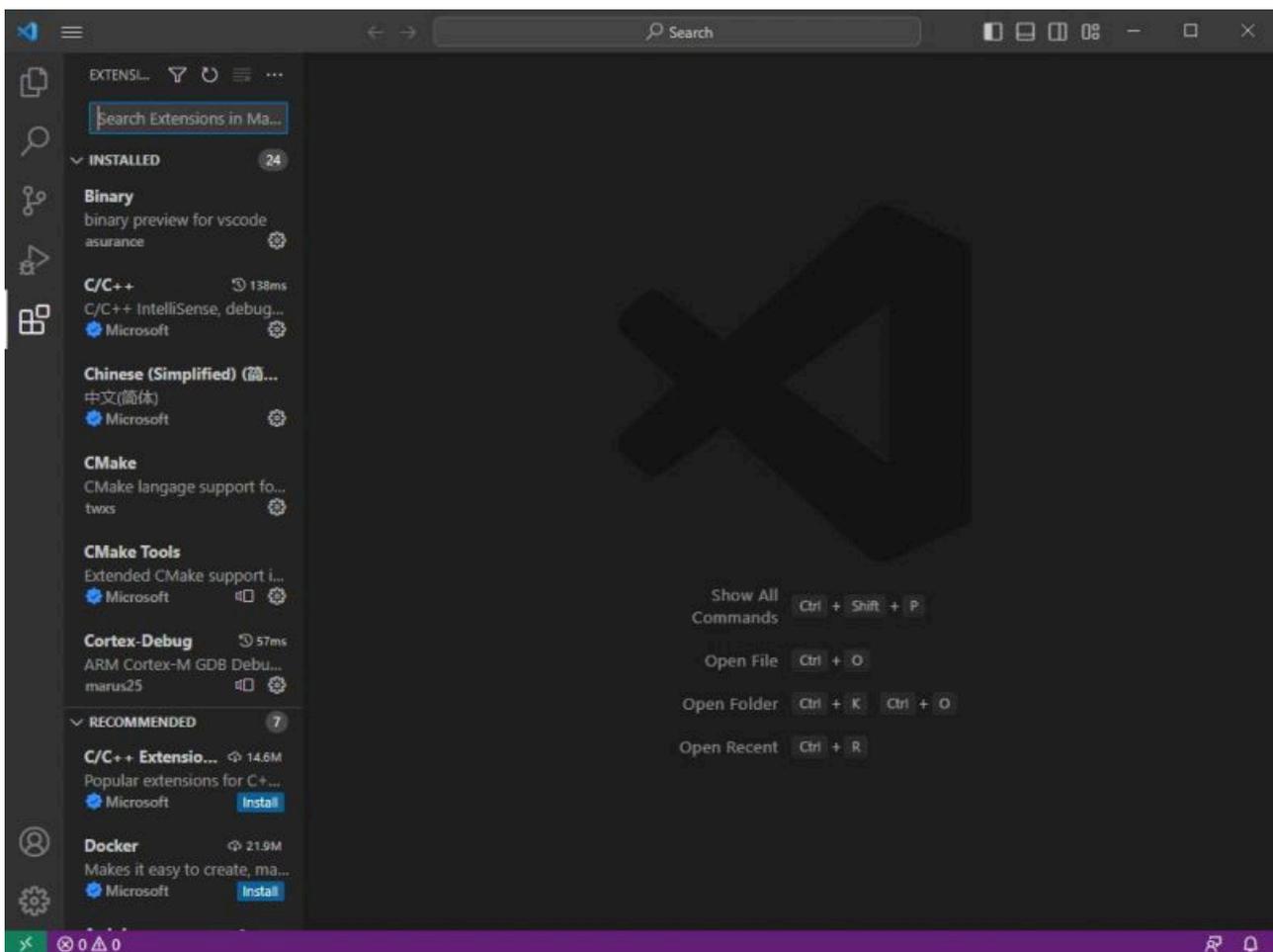
(/wiki/File:ESP32-S3-Pico_07.jpg)

2. In the search bar, type Espressif IDF, select the corresponding plug-in, and click install.
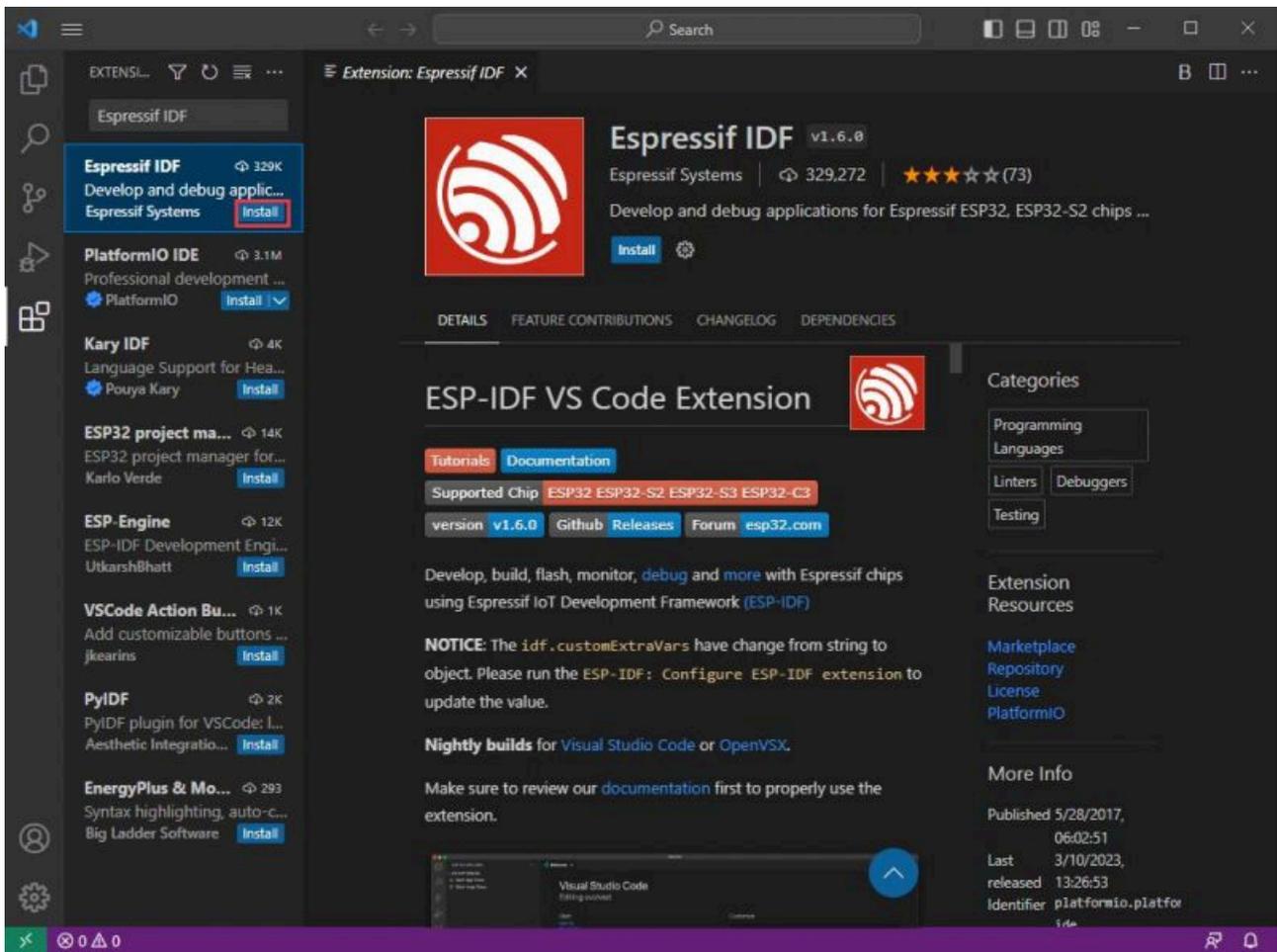
3. Press F1 to enter:

```
esp-idf: configure esp-idf extension
```
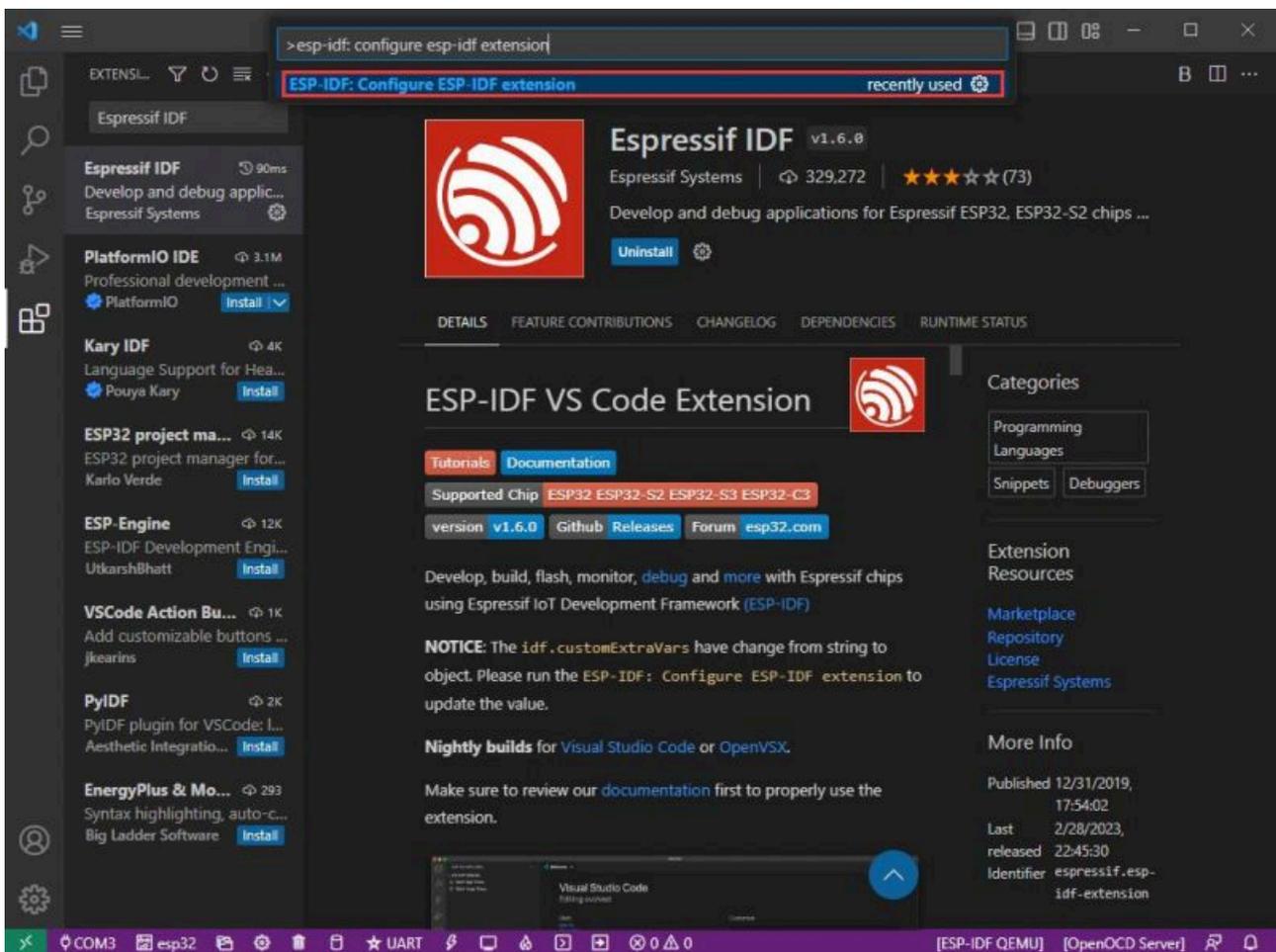
(/wiki/File:ESP32-S3-Pico_09.jpg)

4. Choose express (This tutorial is for first-time users, so only the first general installation tutorial is covered.)
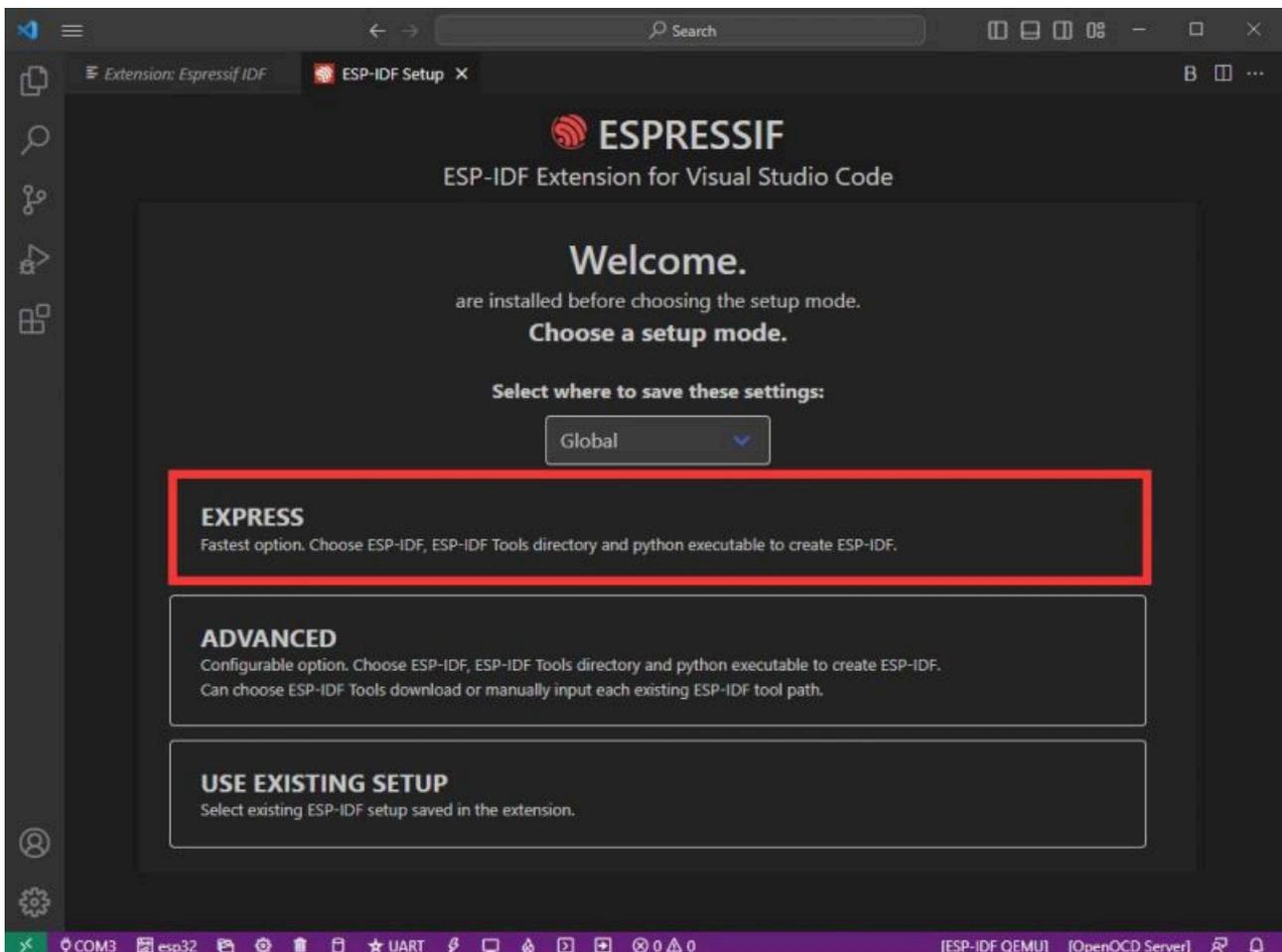
(/wiki/File:ESP32-S3-Pico_10.jpg)

5. Open and display this screen.
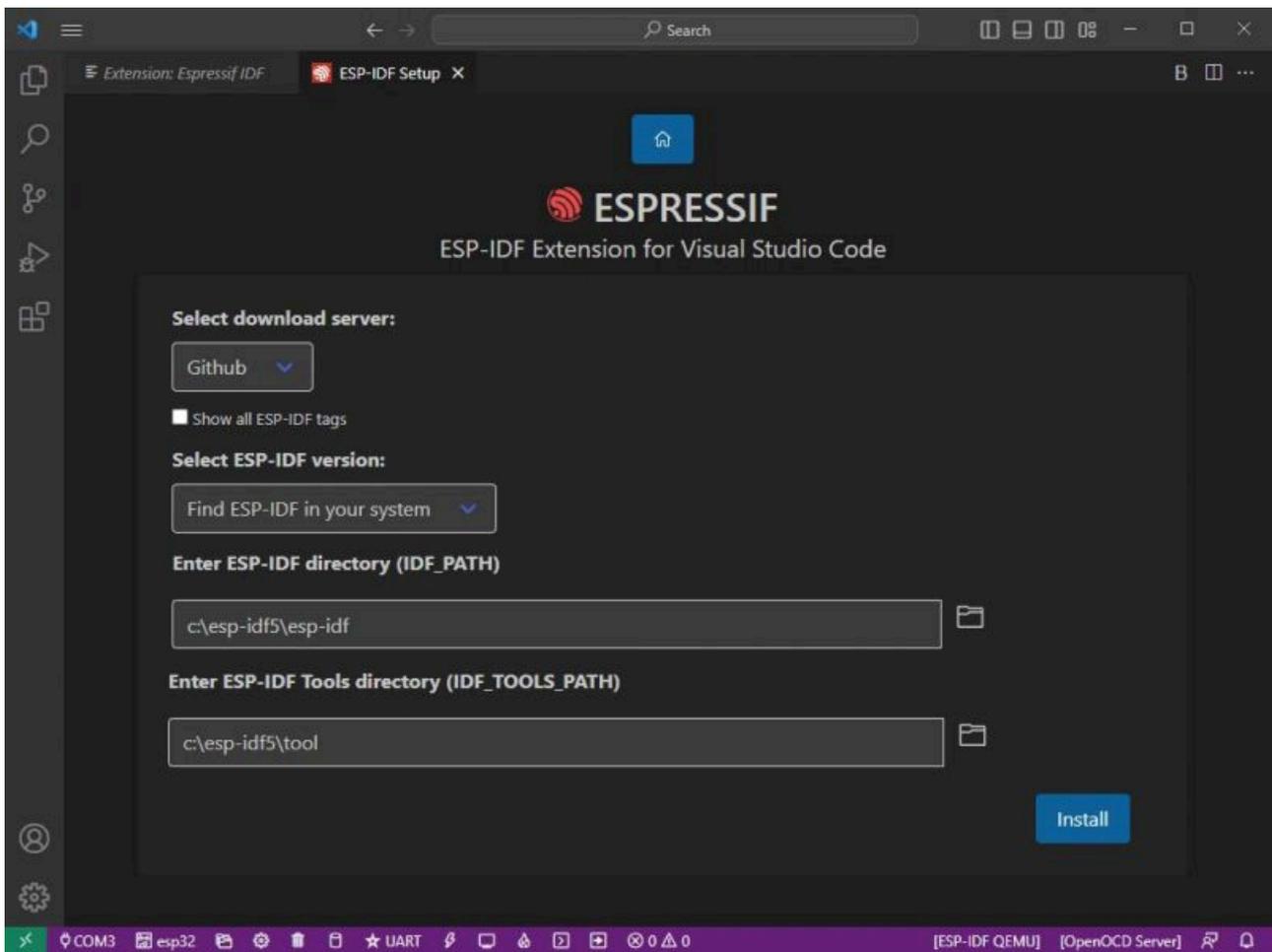
(/wiki/File:ESP32-S3-Pico_11.jpg)

6. Choose a server to download.
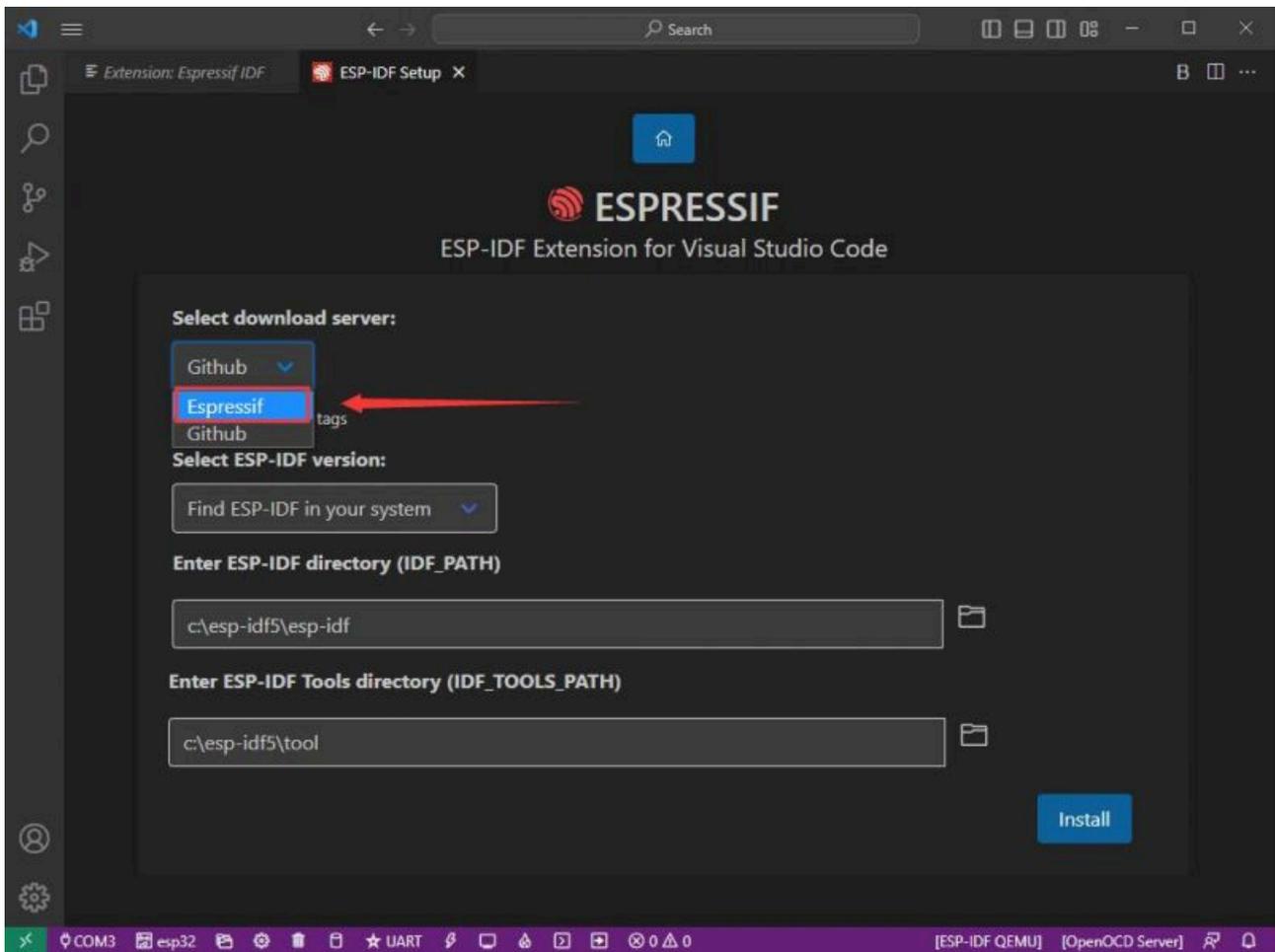
(/wiki/File:ESP32-S3-Pico_12.jpg)

7. Select the ESP-IDF version you want now, we choose the latest V5.3 (note that ESP-IDF started to support ESP32-S3 only after V4.4).
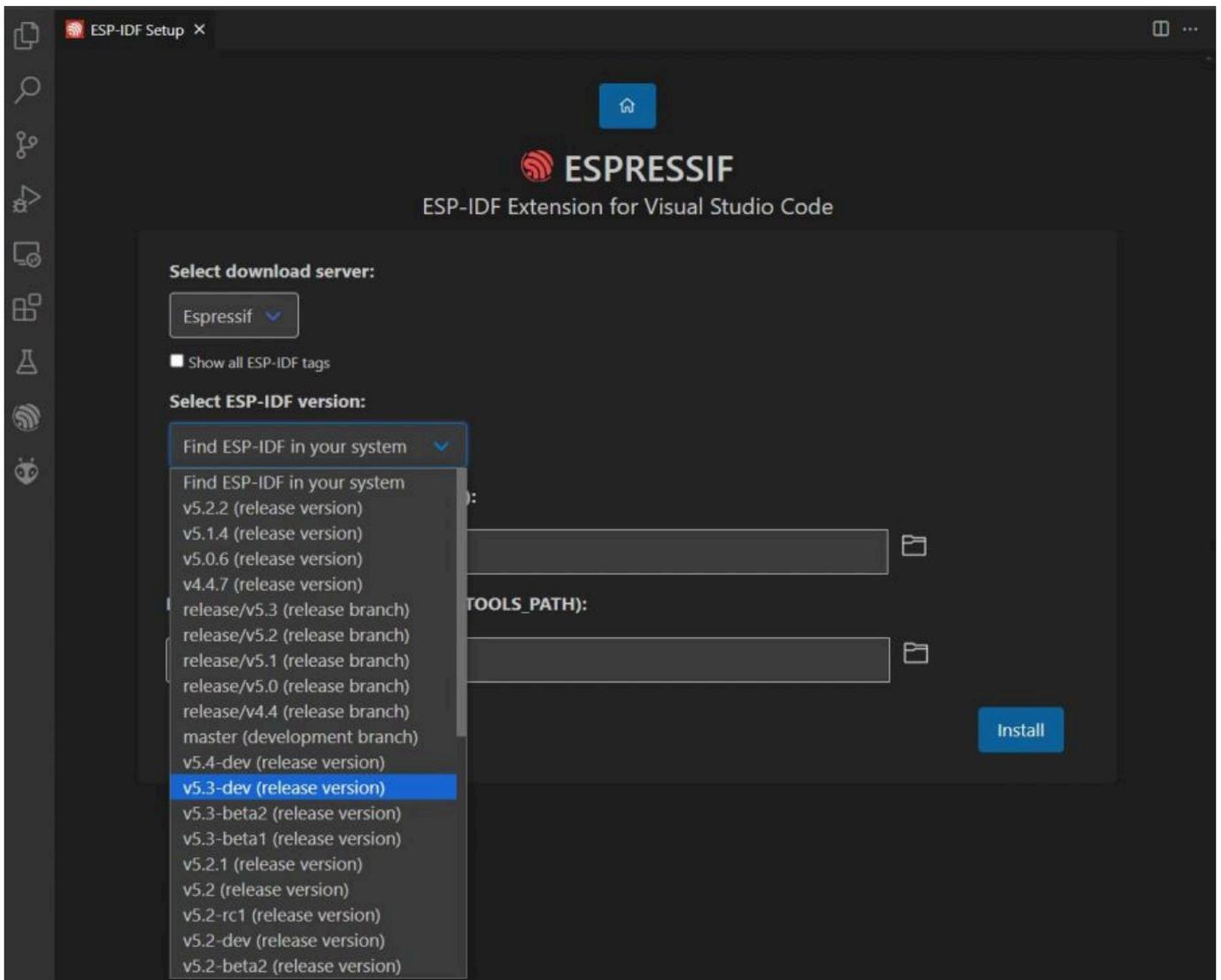
(/wiki/File:ESP32-S3-Touch-LCD-7-18.jpg)

8. The following two are the ESP-IDF directory installation address and the ESP-IDF required tools installation address respectively.
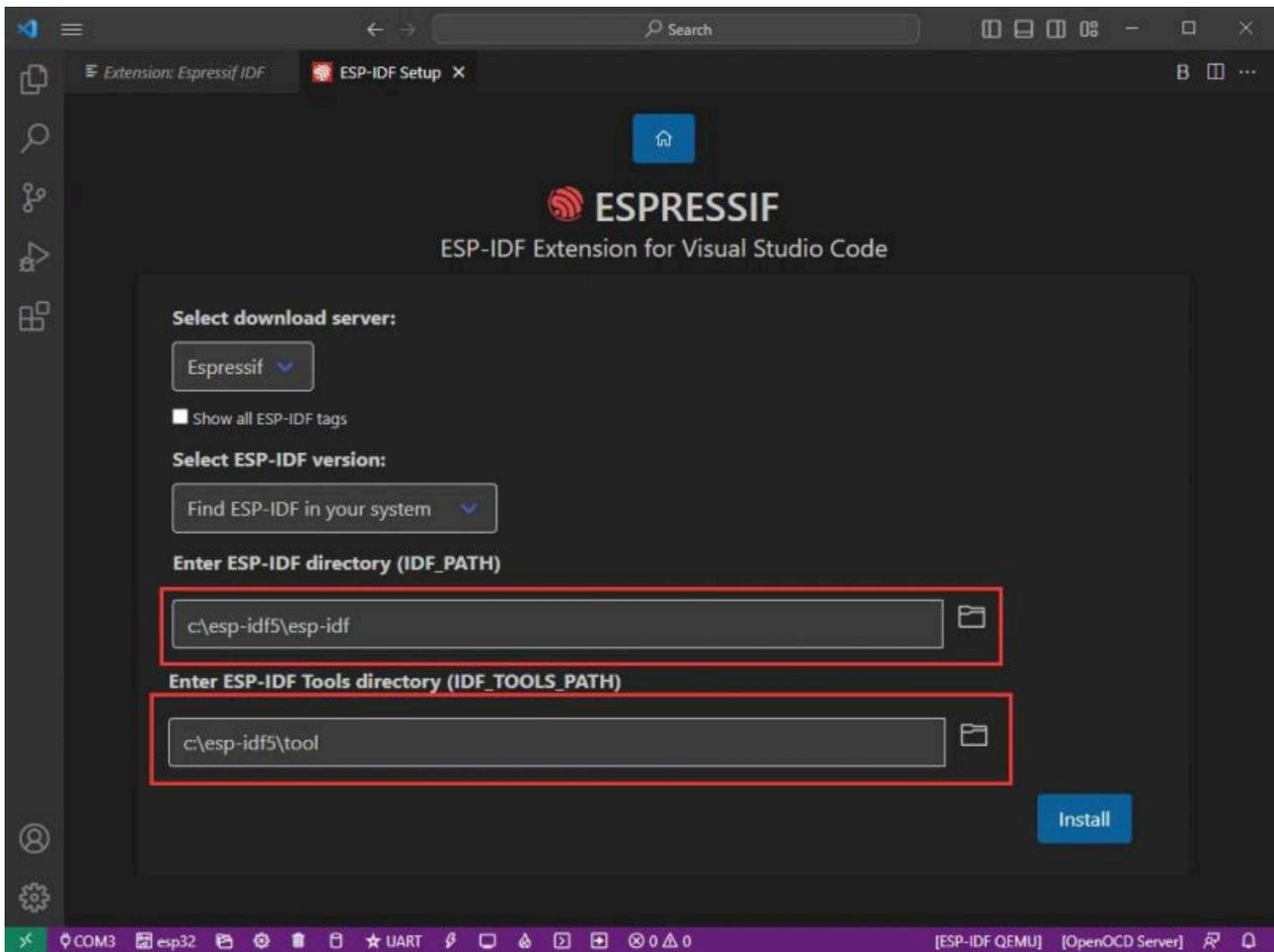
(/wiki/File:ESP32-S3-Pico_14.jpg)

- - **Note: If you have installed ESP-IDF before, or if it has failed, please make sure to delete the file completely or create a new path.**

9. Once the configuration is finished, click "install" to download.
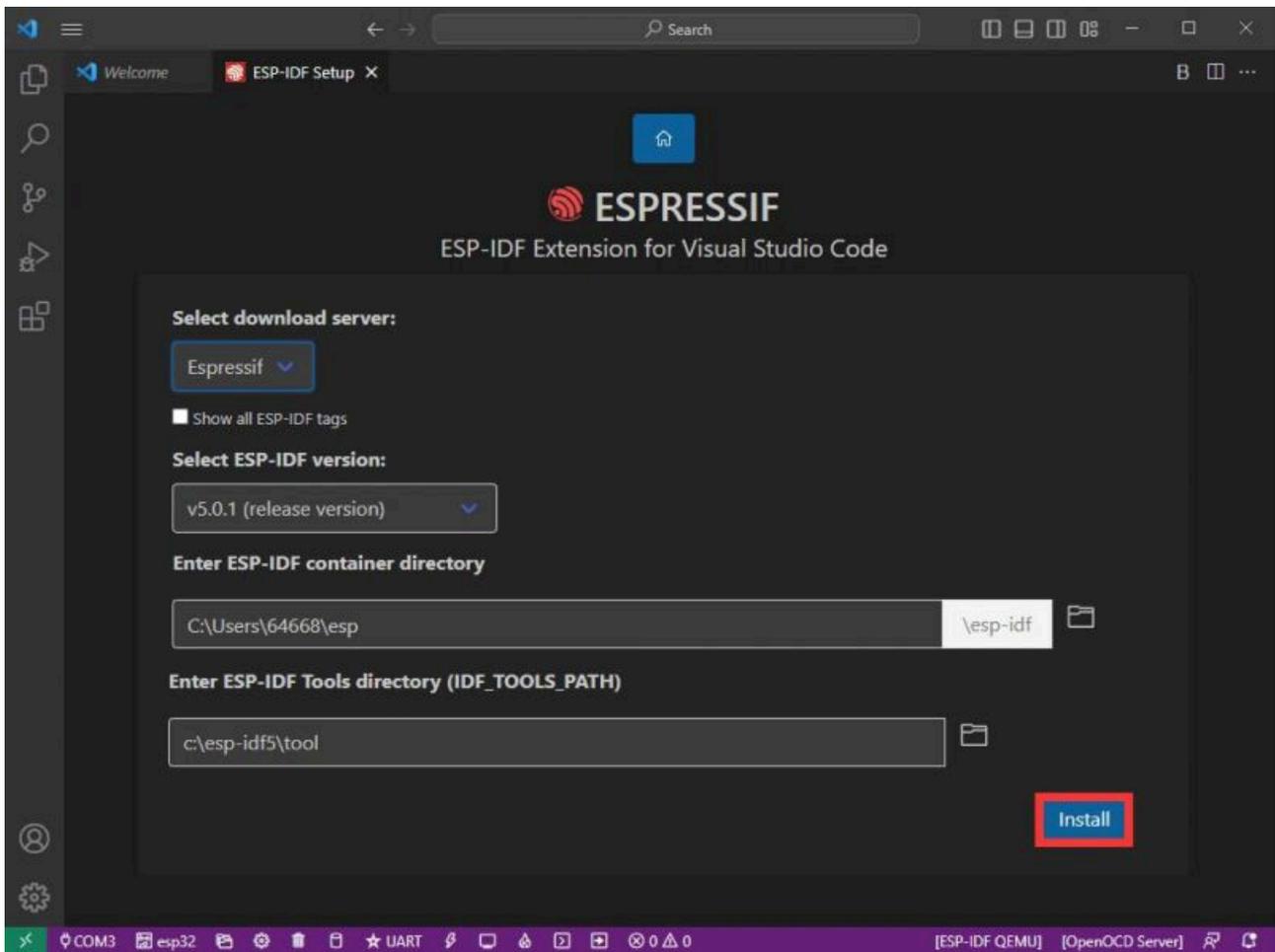
10. After the installation is completed, it will enter the following screen, indicating that the installation is finished.
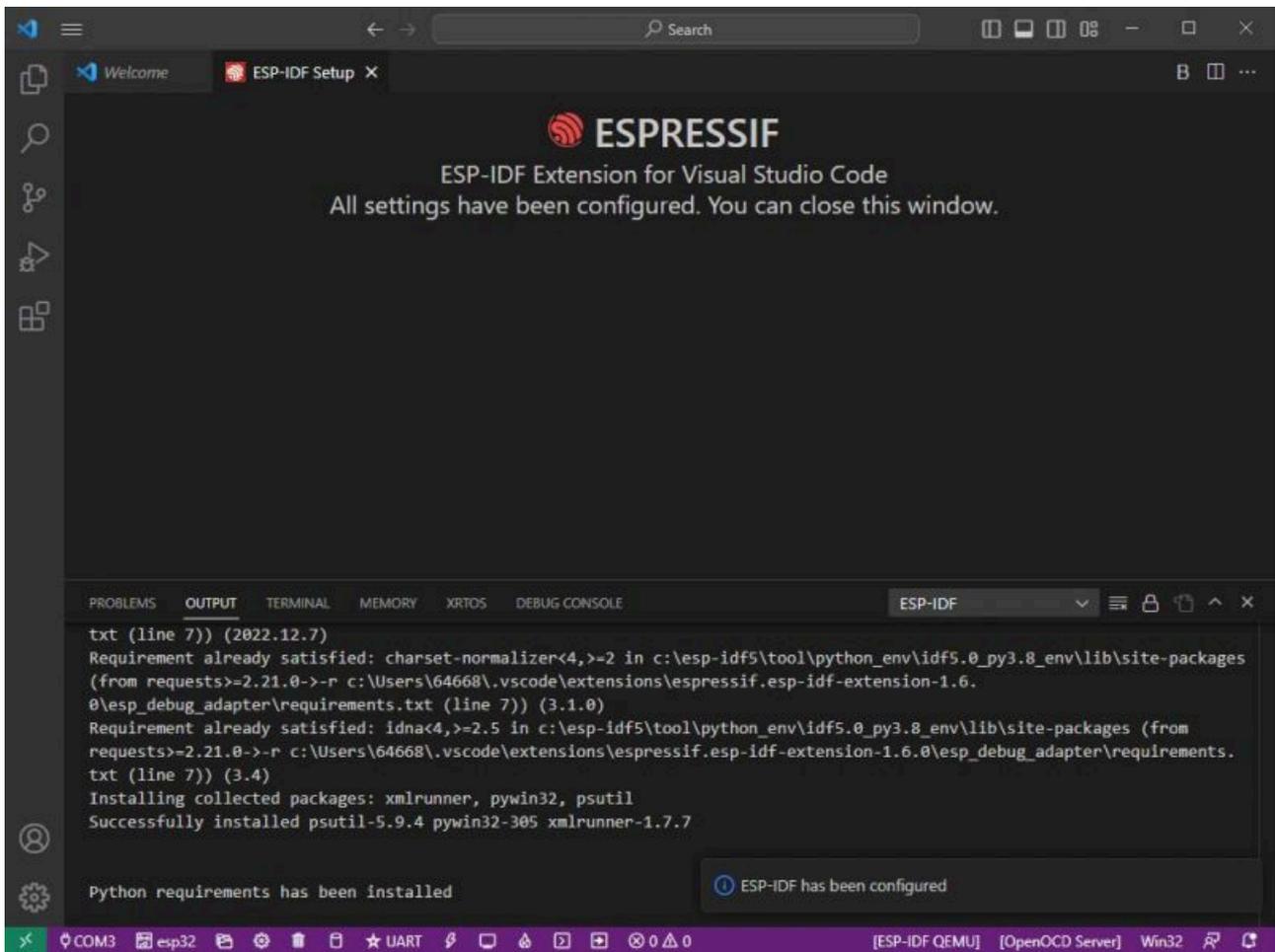
(/wiki/File:ESP32-S3-Pico_17.jpg)

Offline Package

- If it fails to download "esp-idf", you can try this offline package while the tool package should be downloaded online first.

- Double-click on it after downloading, and then type the path as shown below:
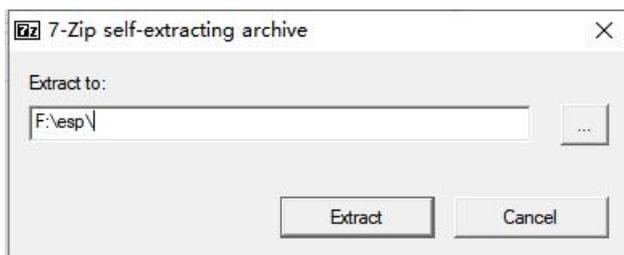
(/wiki/File:ESP32-S3-Touch-LCD-7-14.jpg)

1. Click on "Extract" to unzip the package:

2. After unzipping, create a file folder to store the compile tool in the unzipped file folder named "Espressif".
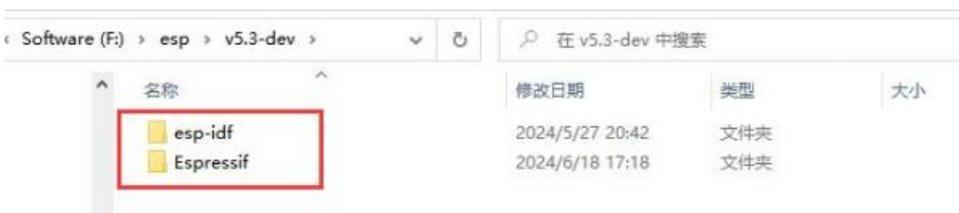
(/wiki/File:ESP32-S3-Touch-

LCD-7-15.jpg)

3. Enter the seventh STEP in the installation process of the Espressif IDF, we can set it according to the following picture and then click on "Install":
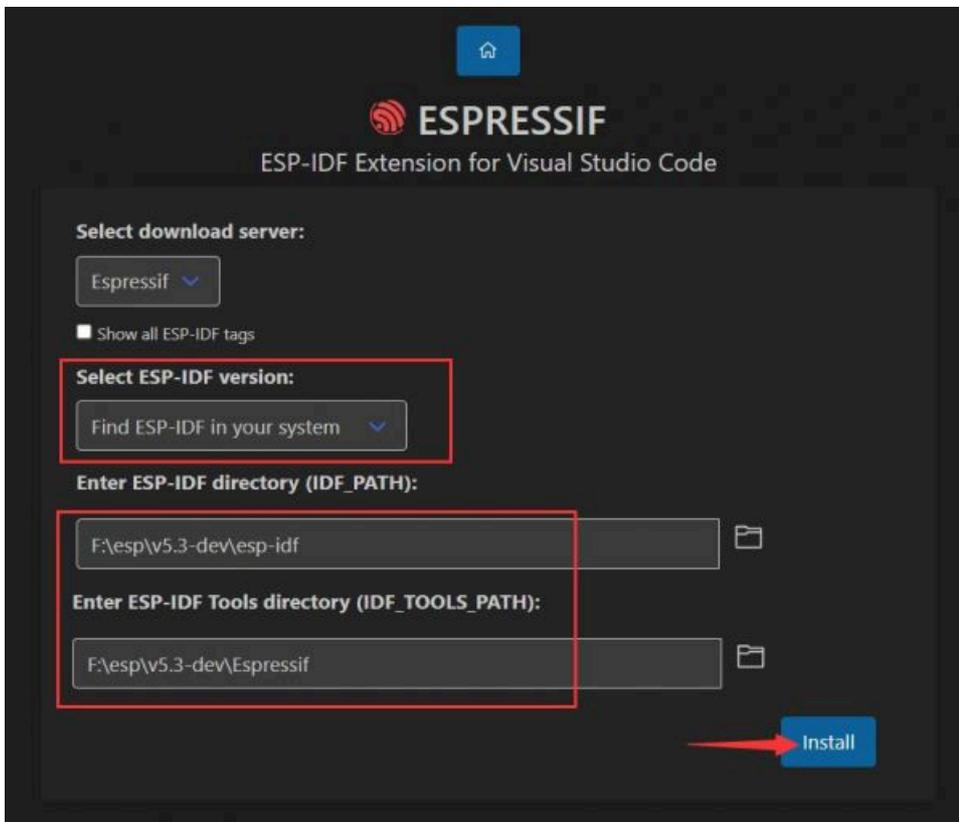
(/wiki/File:ESP32-S3-Touch-

LCD-7-16.jpg)

4. It is successfully installed as shown below:
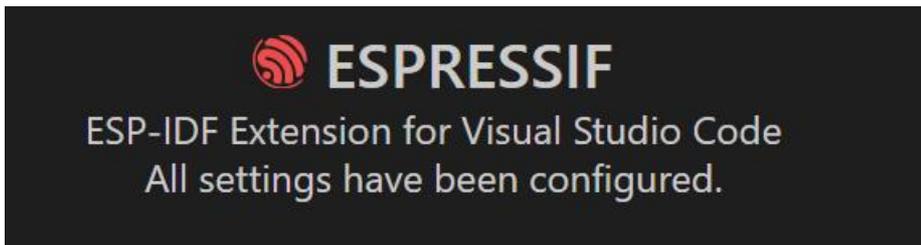
(/wiki/File:ESP32-S3-Touch-LCD-7-

17.jpg)

Official Demo

Create Demo

1. Using the shortcut F1, type:

```
esp-idf:show examples projects
```
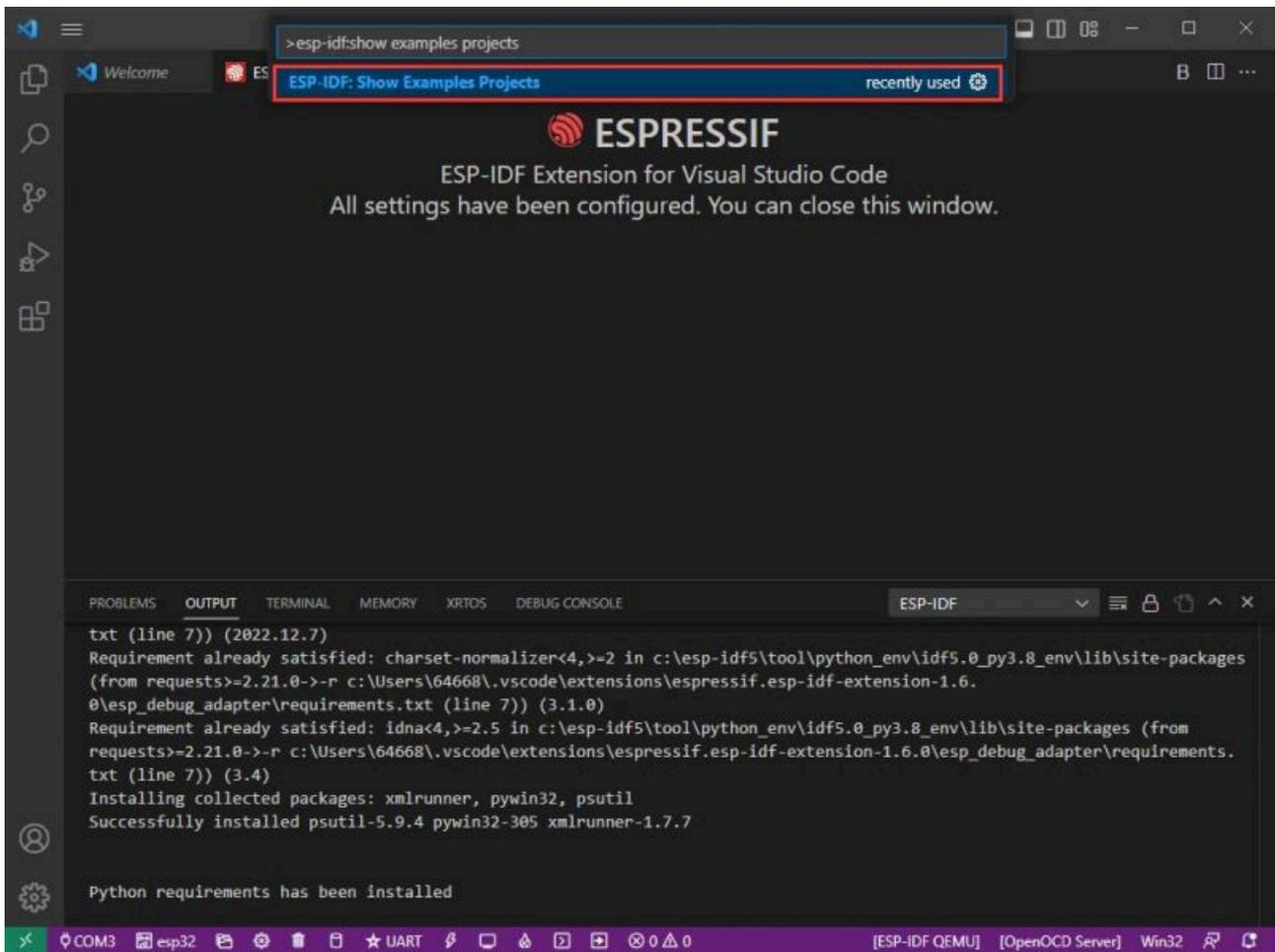
(/wiki/File:ESP32-S3-Pico_18.jpg)

2. Choose your current IDF version:

(/wiki/File:ESP32-S3-Touch-LCD-7-20.jpg)

3. Take "Hello World" as an example:
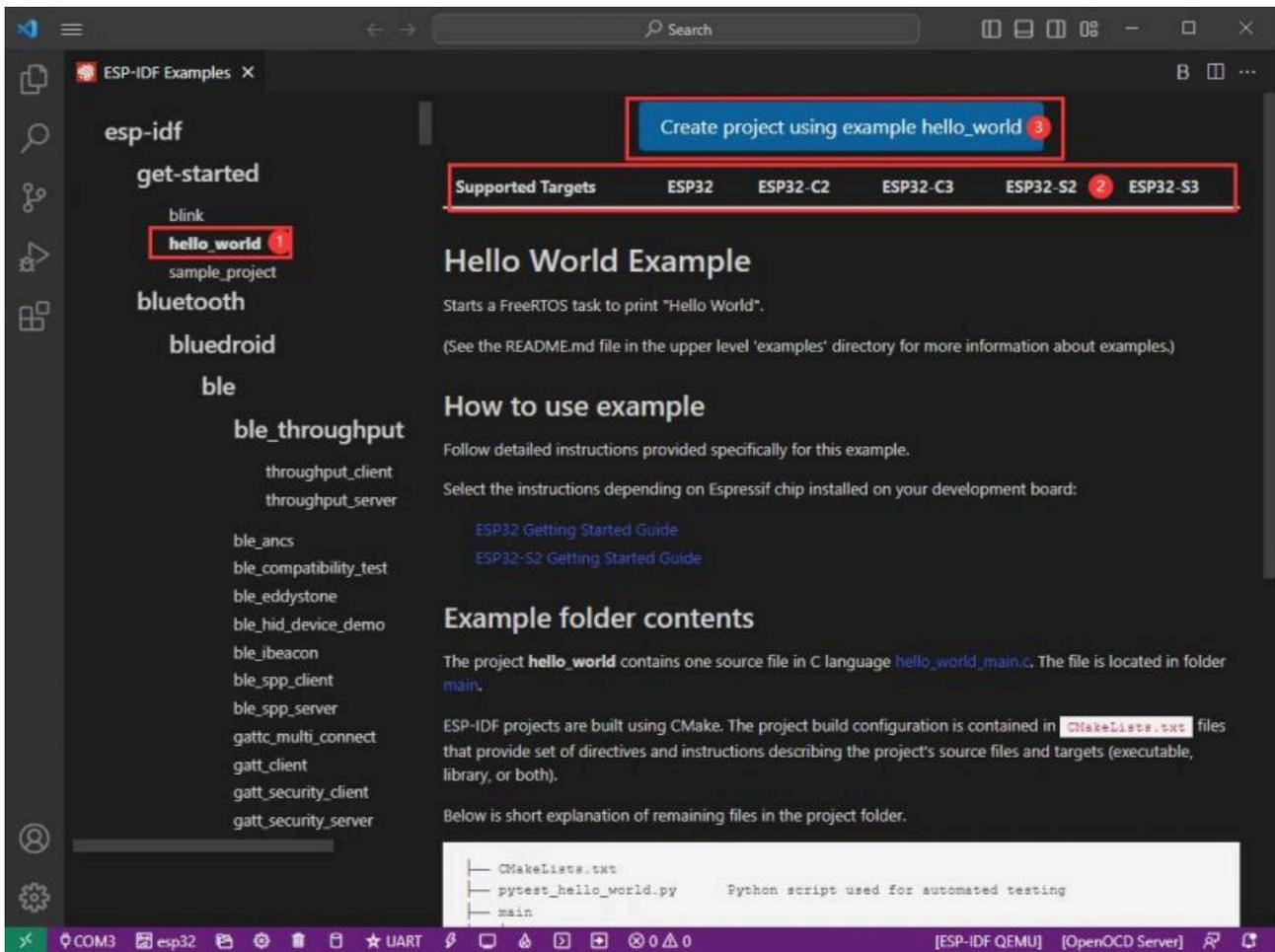
4. ① Choose the corresponding demo.

5. ② The readme file will explain which chip the demo is suitable for (the following section will introduce how to use the demo and its file structure, which is omitted here).

6. ③ Click to create the demo.

7. Choose the path to place the demo and ensure that there is no folder with the same name as the demo.
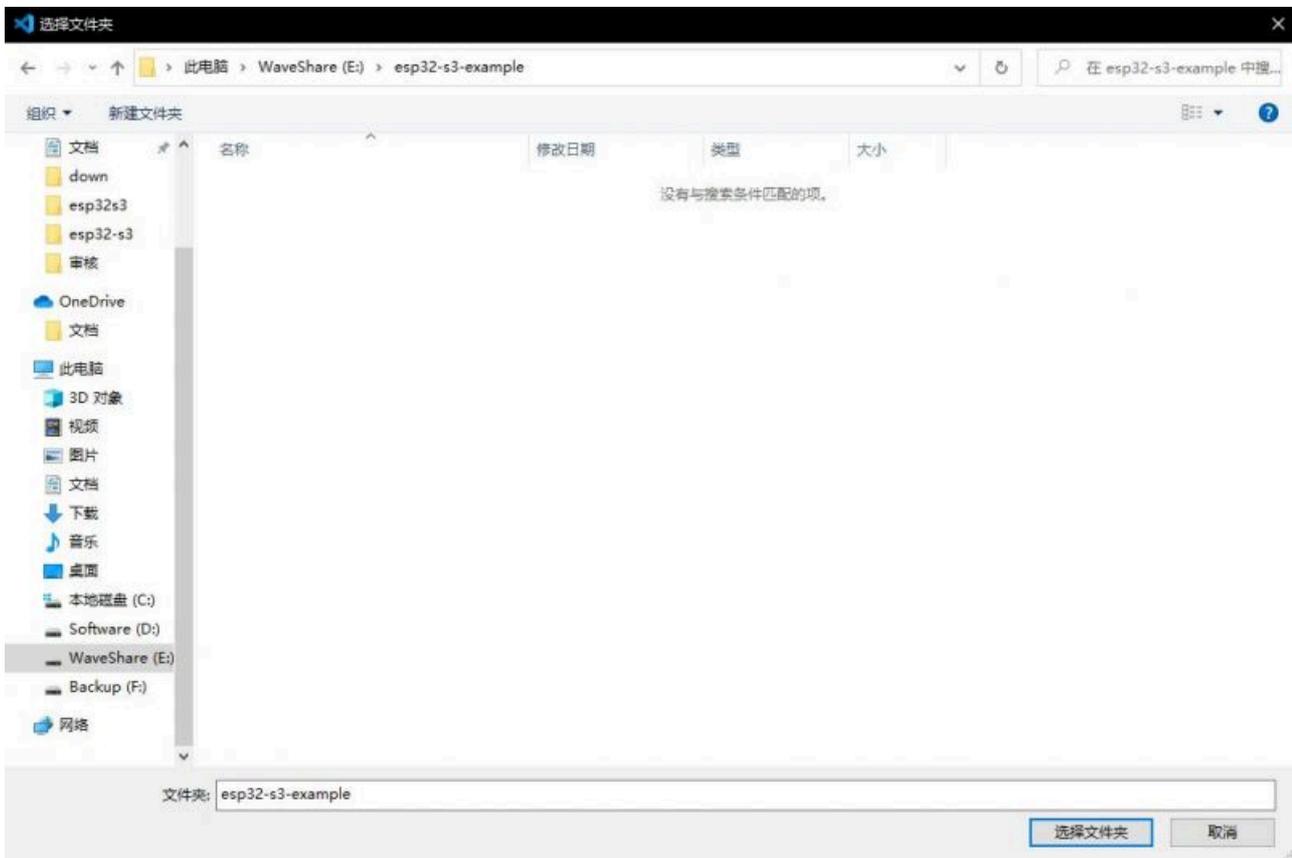
(/wiki/File:ESP32-S3-Pico_21.jpg)

Modify COM Port

1. The corresponding COM port is displayed here, click on it to modify.
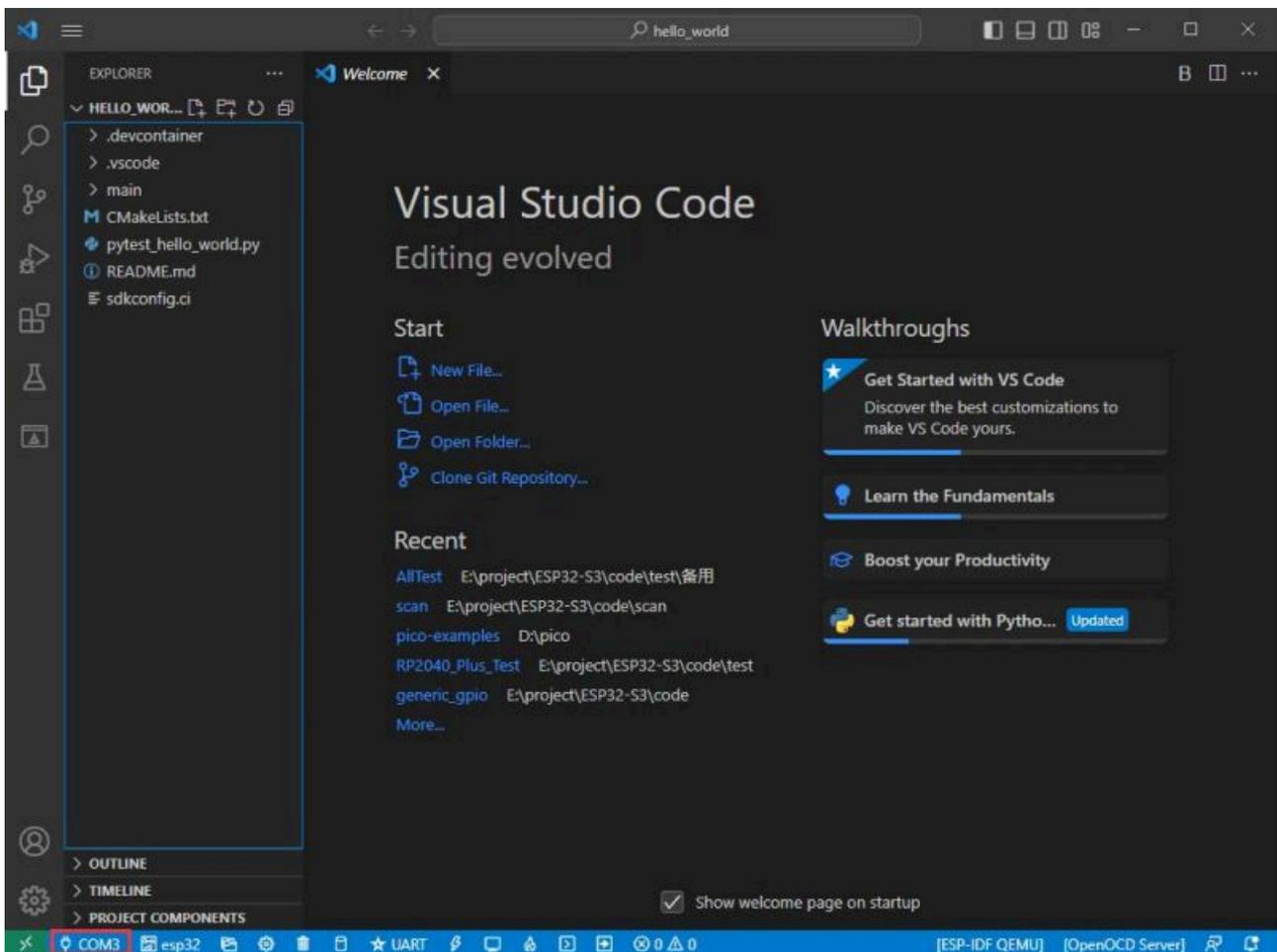
(/wiki/File:ESP32-S3-Pico_22.jpg)

2. We check the device manager COM port, and select COM5, please select your corresponding COM port:
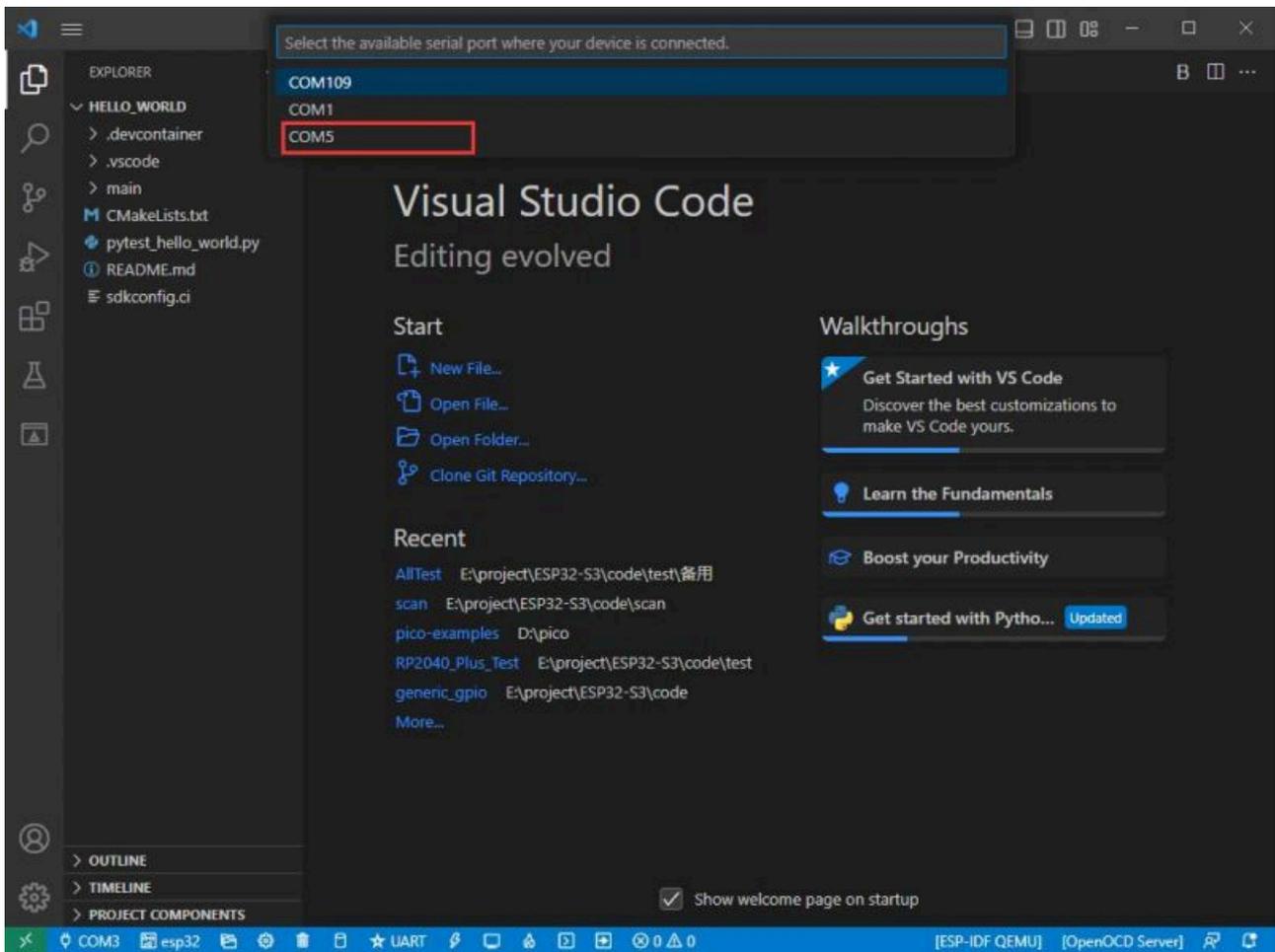
(/wiki/File:ESP32-S3-Pico_23.jpg)

3. Choose the project and demo.
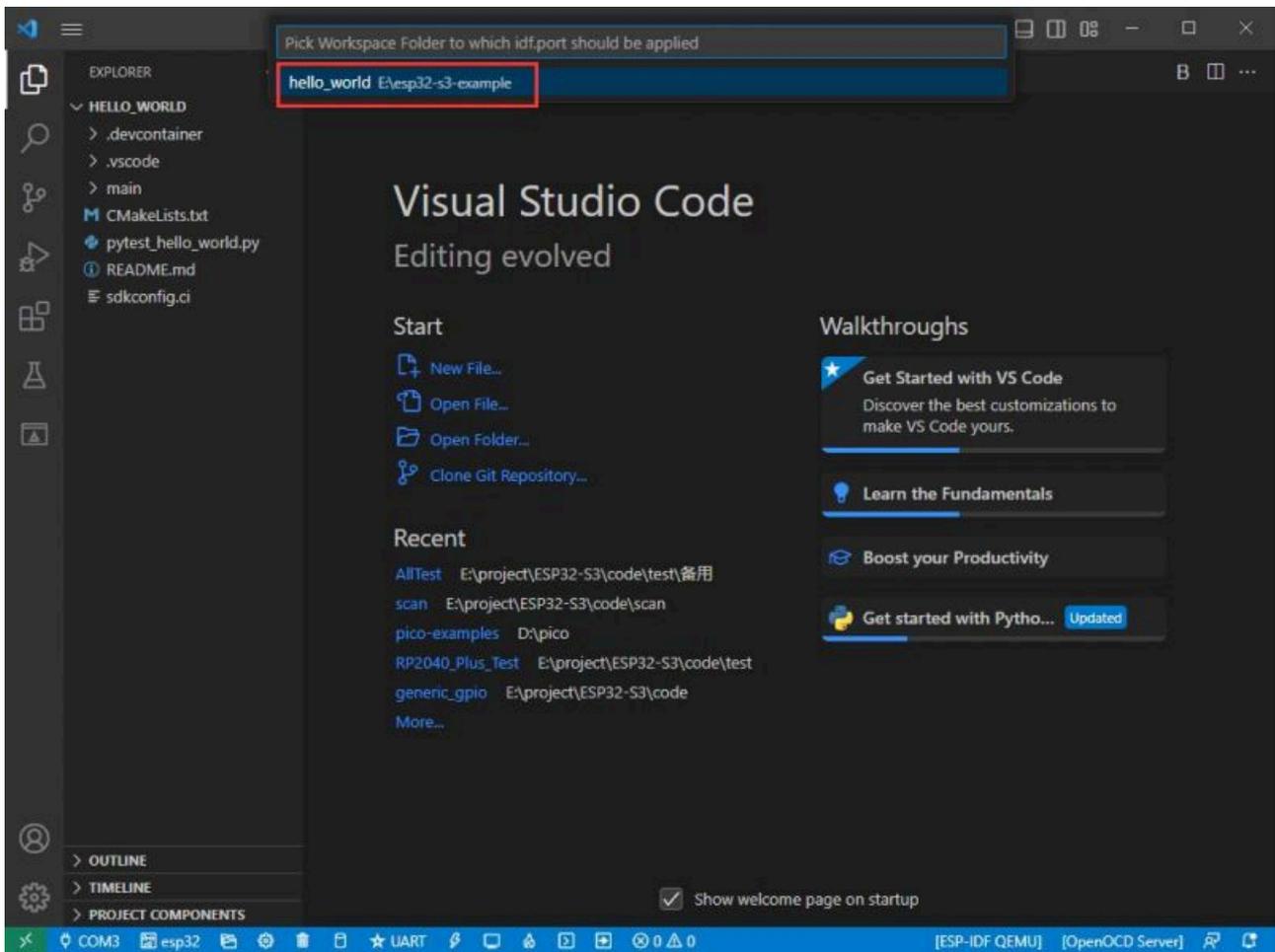
(/wiki/File:ESP32-S3-Pico_24.jpg)

4. Then the COM port is modified.

Modify the Driver

1. Here shows the driver used, click here to modify the corresponding driver:
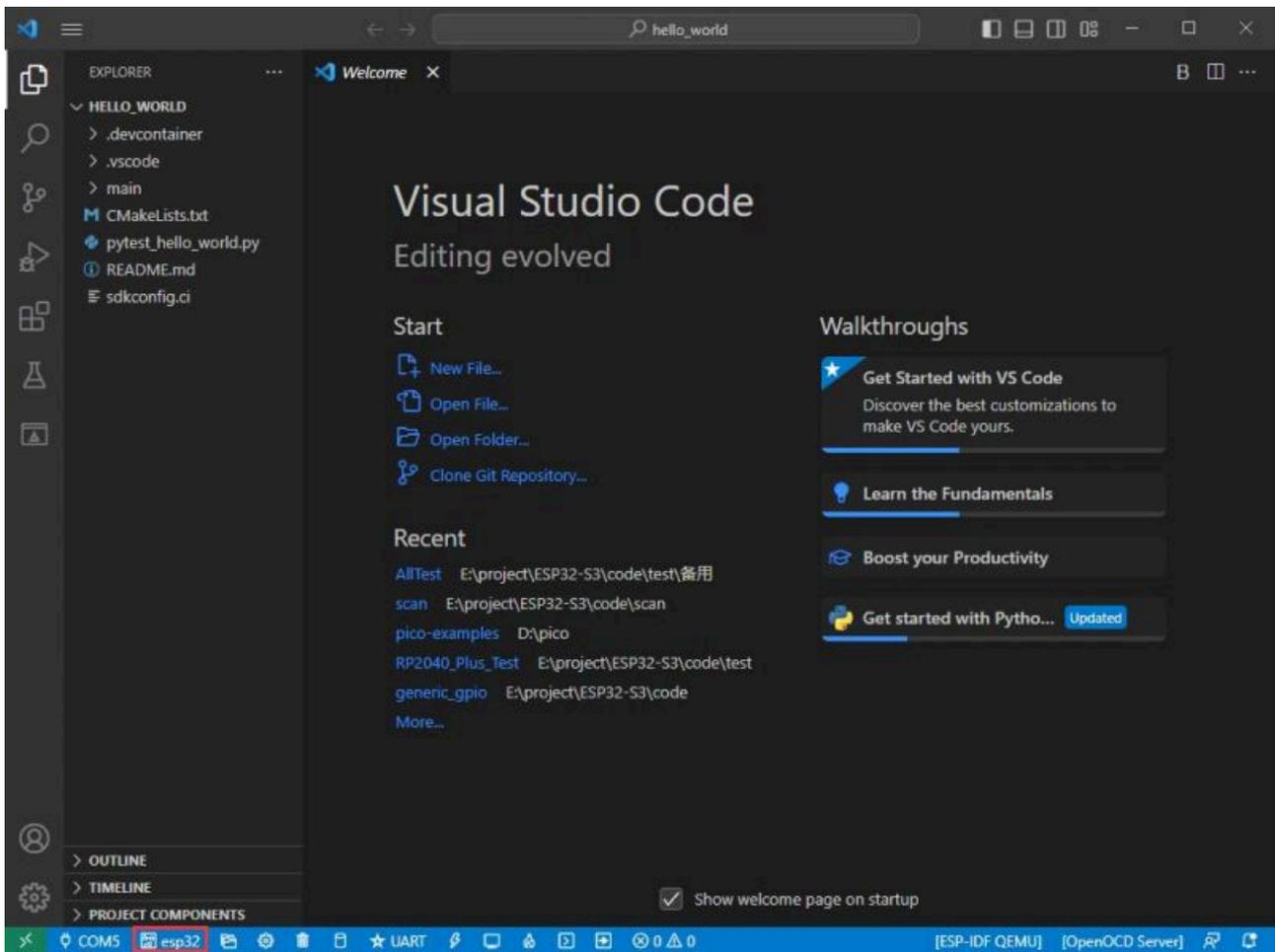
(/wiki/File:ESP32-S3-Pico_25.jpg)

2. Choose the project or demo:
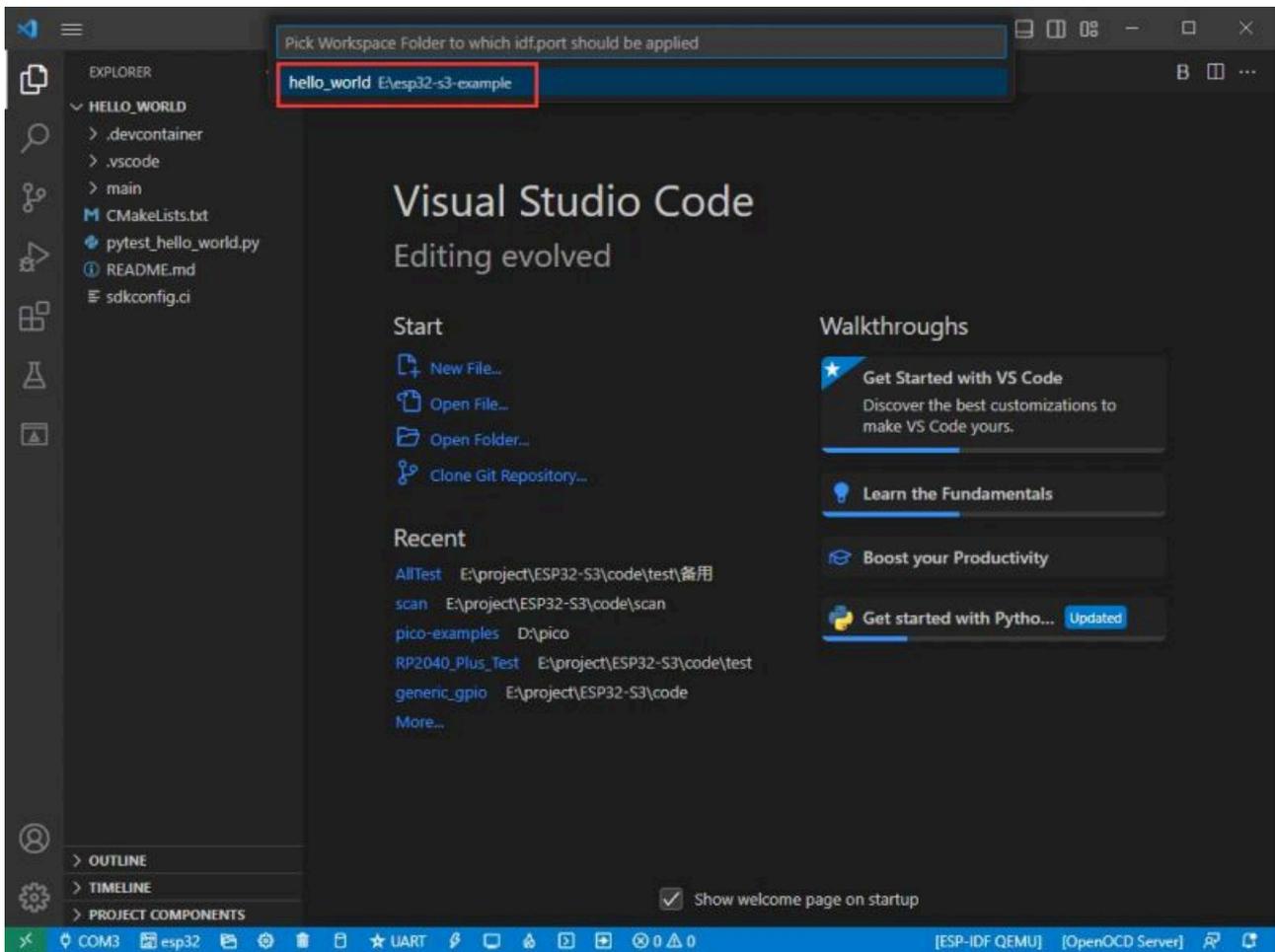
(/wiki/File:ESP32-S3-Pico_24.jpg)

3. Wait for a few seconds after clicking.
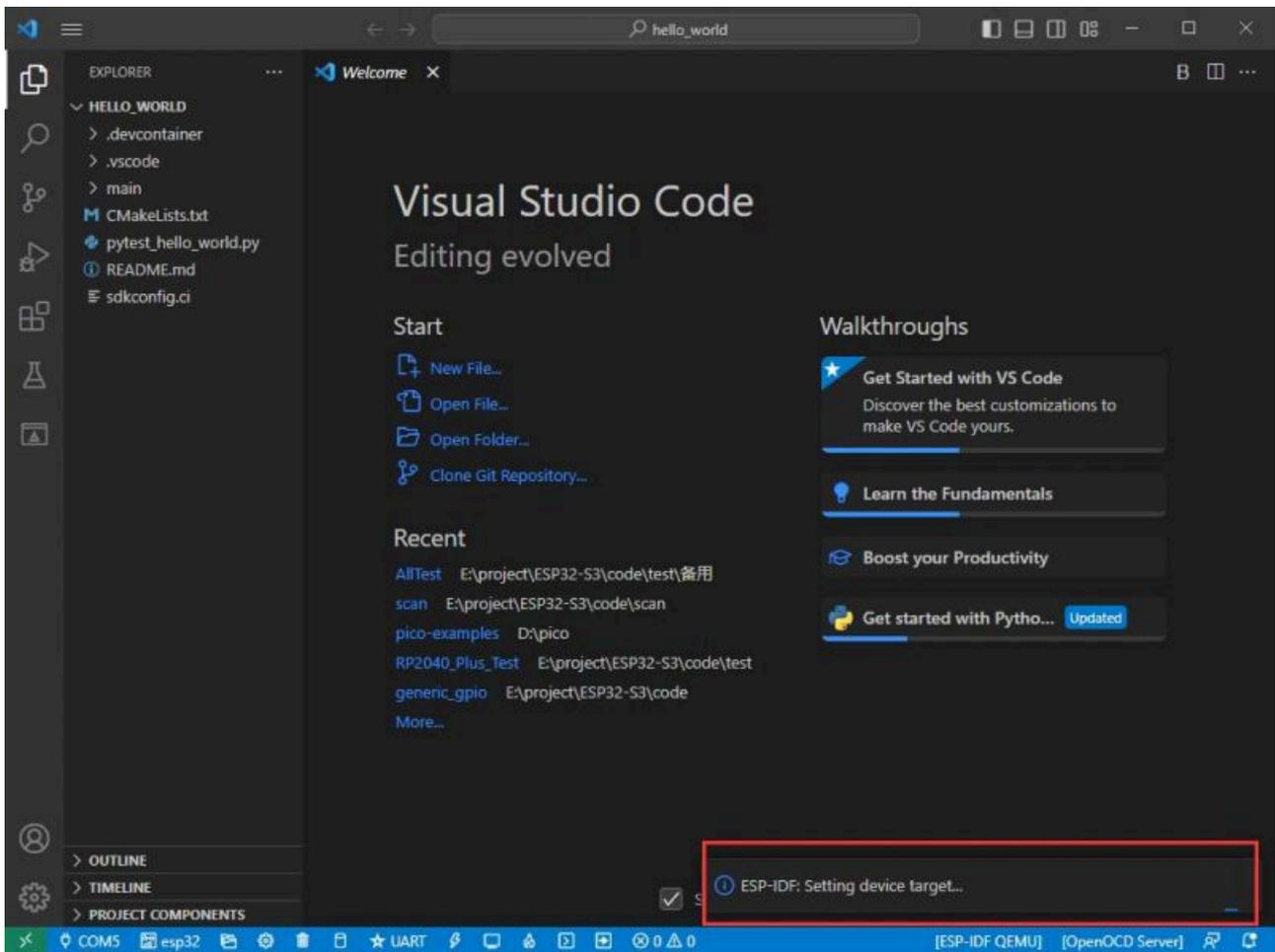
(/wiki/File:ESP32-S3-Pico_27.jpg)

4. Choose the driver we need, that is, the main chip ESP32S3.
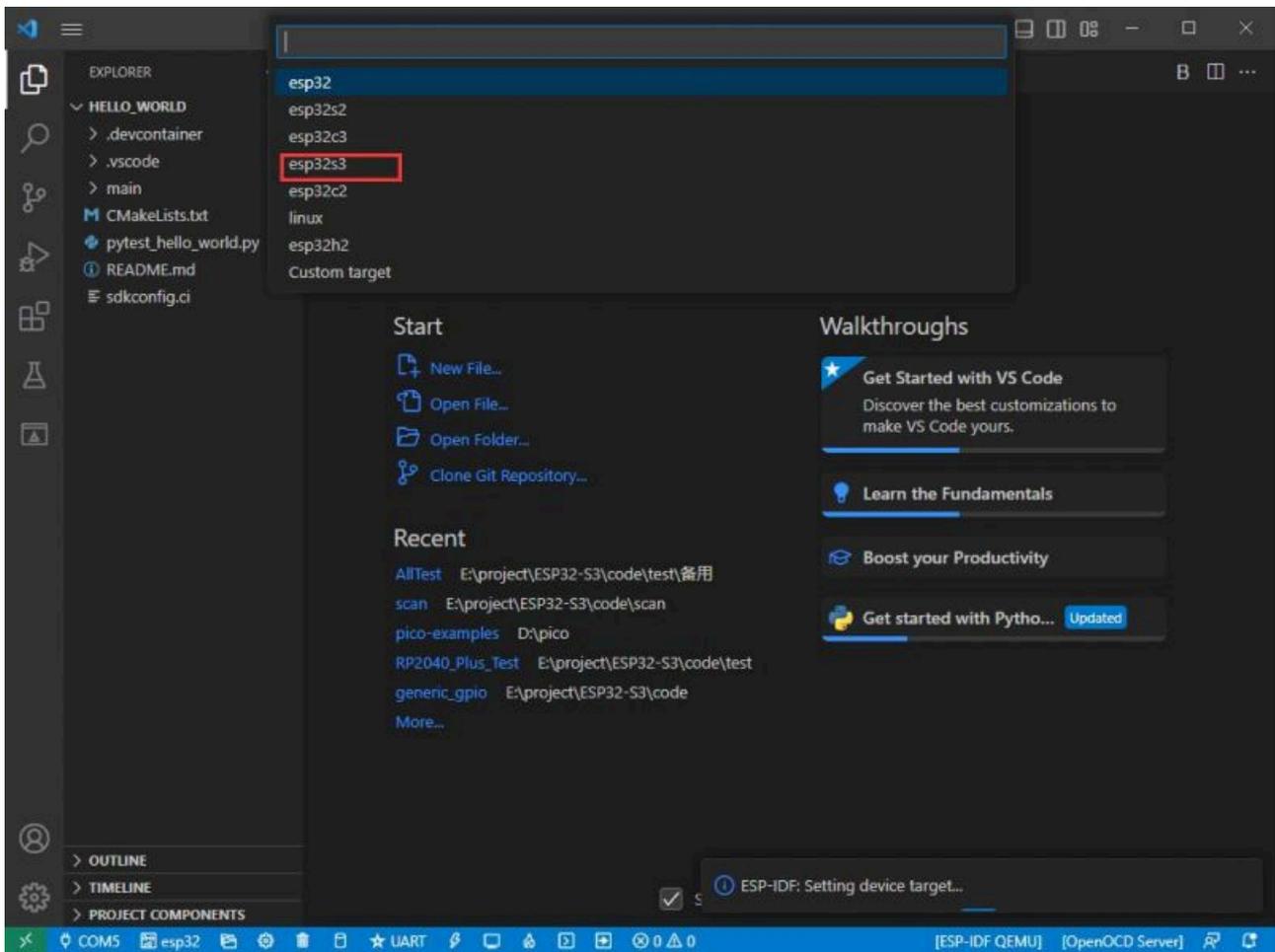
(/wiki/File:ESP32-S3-Pico_28.jpg)

5. Choose the openocd path, we can just choose one at random as it doesn't matter.
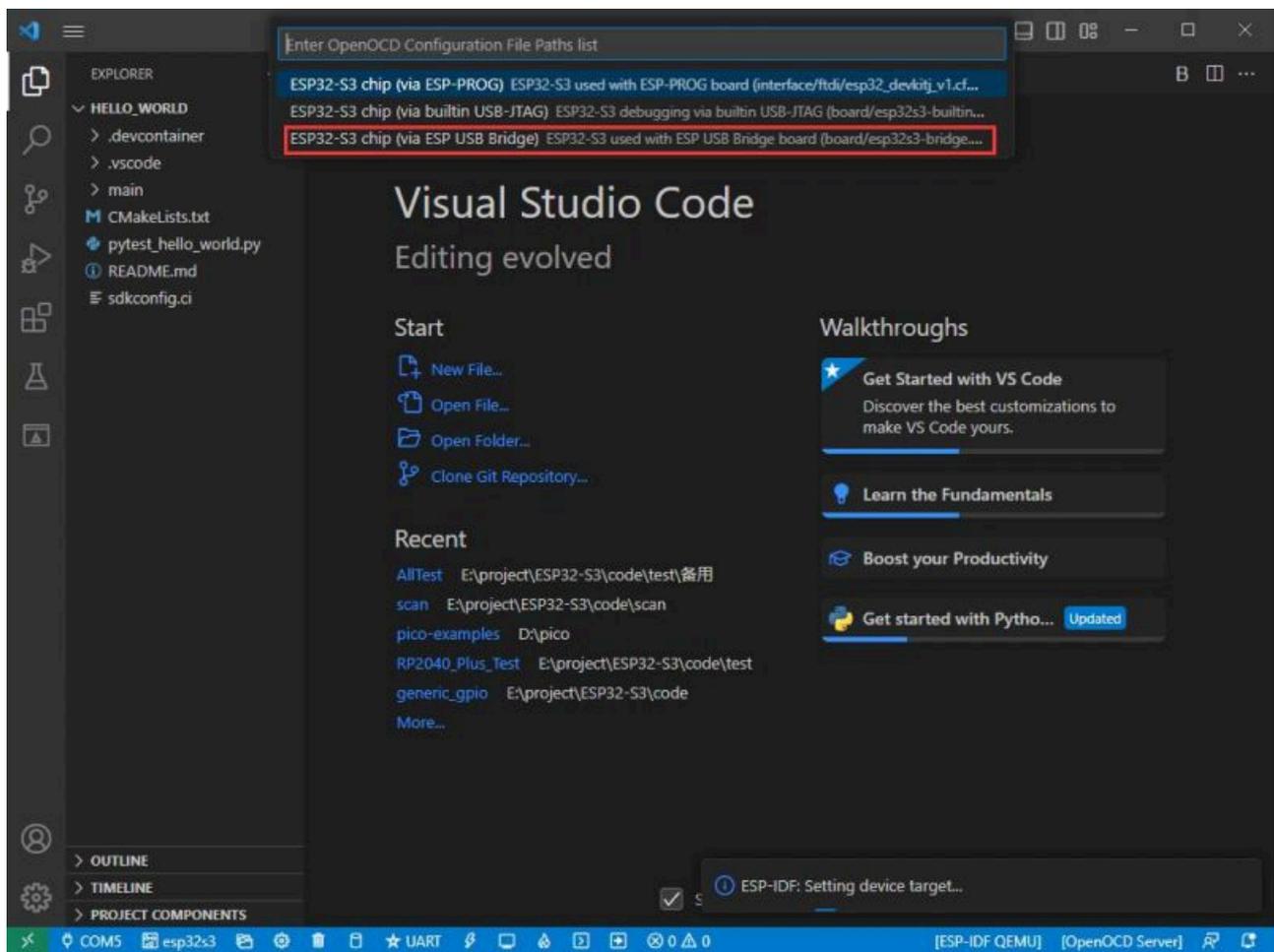
(/wiki/File:ESP32-S3-Pico_29.jpg)

The Rest of the Status Bar Introduction
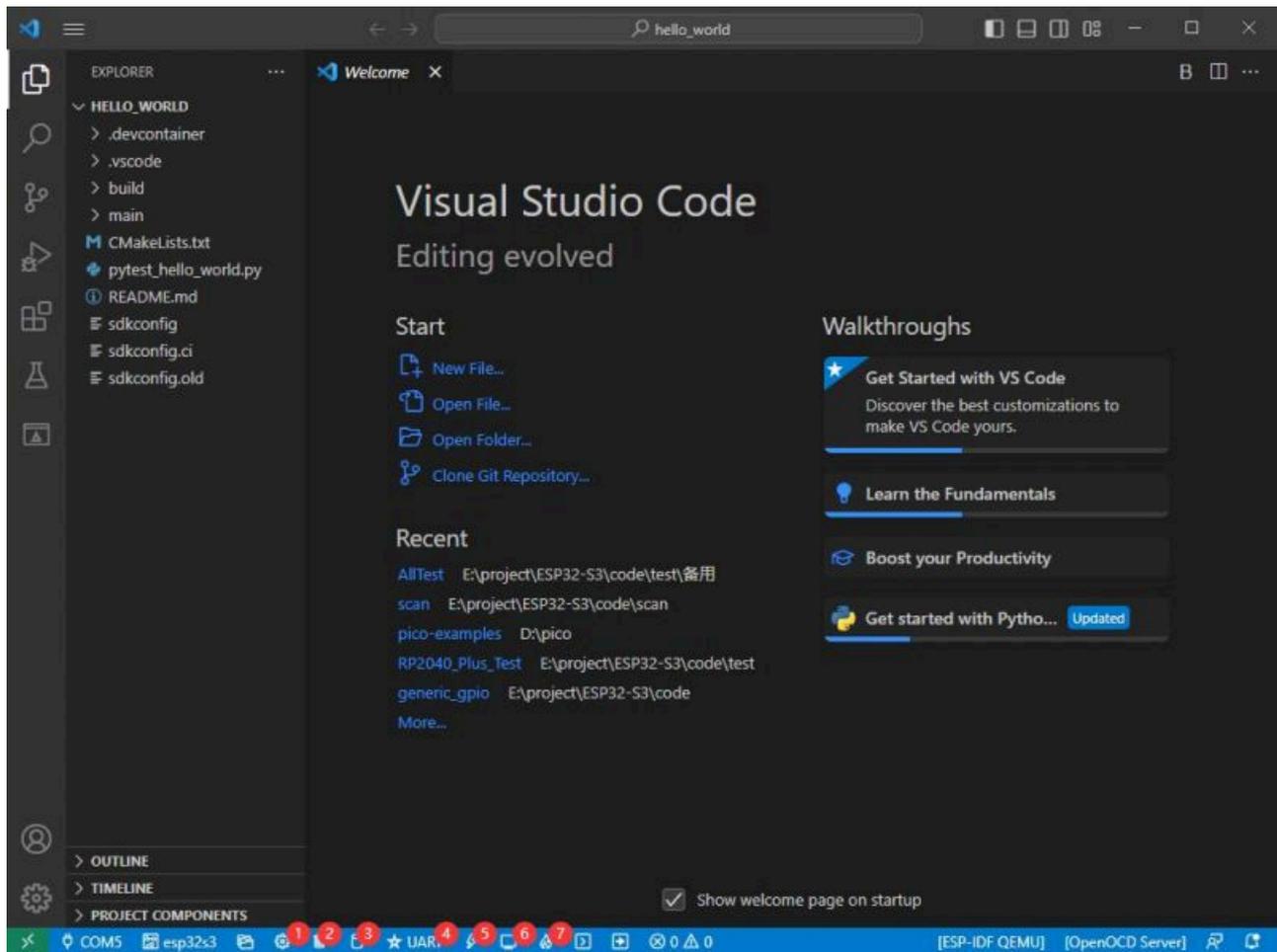
(/wiki/File:ESP32-S3-Pico_30.jpg)

- ① SDK configuration editor: many functions and configurations of ESP-IDF can be modified within it.
- ② Clean up everything and delete all compiled files.
- ③ Compile.
- ④ Current download method, default is UART.
- ⑤ Program the current firmware, please do it after compiling.
- ⑥ Open the serial monitor to view serial information.
- ⑦ Combined button for compiling, programming, and opening the serial monitor (most commonly used during debugging).

Compile, Program, and Serial Port Monitoring

1. Click on the Compile, Program, and Open Serial Monitor buttons we described earlier.

(/wiki/File:ESP32-S3-Pico_31.jpg)

2. It may take a long time to compile, especially for the first time.
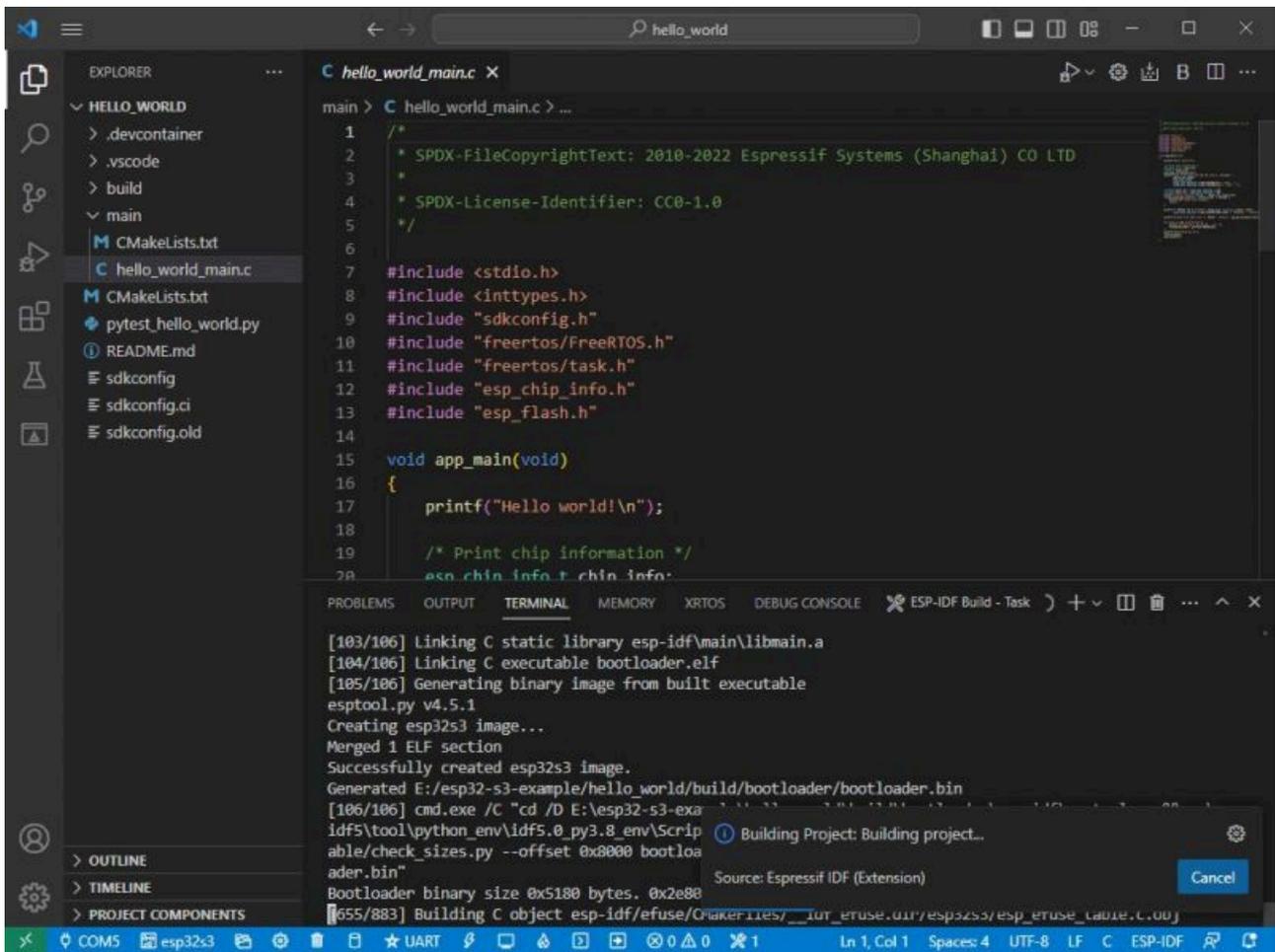
(/wiki/File:ESP32-S3-Pico_32.jpg)

- During this process, ESP-IDF may take up a lot of CPU resources and therefore may cause system lag.

3. Because we use CH343 as a USB to serial port chip, and the on-board automatic download circuit, it can be downloaded automatically without manual operation.
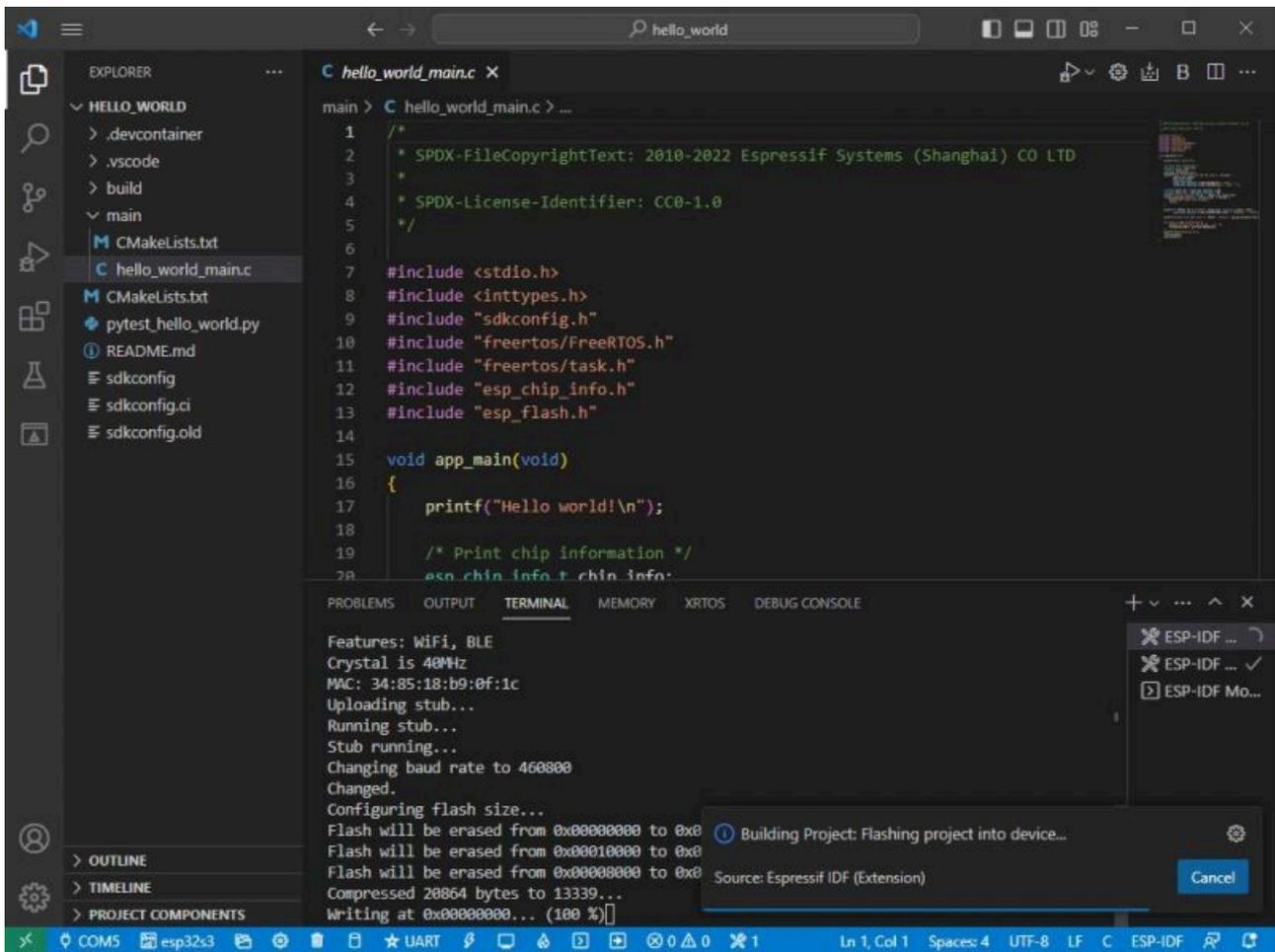
(/wiki/File:ESP32-S3-Pico_33.jpg)

4. After successful download, it will automatically enter the serial monitor, and you can see the corresponding information output from the chip and prompt to reboot after 10s.

(/wiki/File:ESP32-S3-Pico_34.jpg)

Arduino

- Download and install Arduino IDE (https://www.arduino.cc/en/software).

- 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).

- File -> Preferences -> Additional Boards Managers URLs -> Add the following links:

```
https://espressif.github.io/arduino-esp32/package_esp32_dev_index.json
```
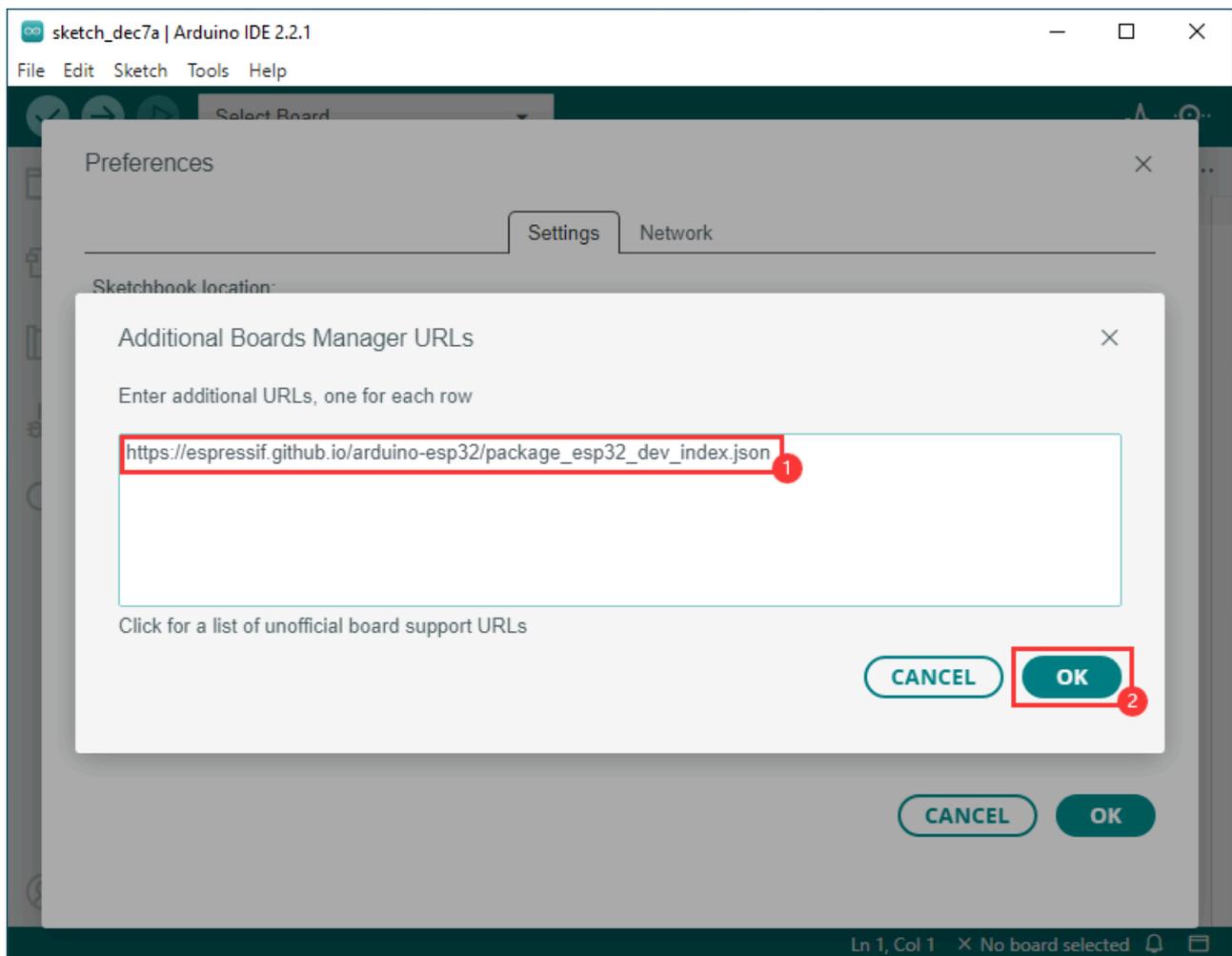
(/wiki/File:ESP32-C6-DEV-KIT-N8-Arduino05.png)

- Enter the development board manager, search for "esp32", select version 3.0.0-alpha3 under "esp32 by Espressif Systems" below, and reboot it after installation.
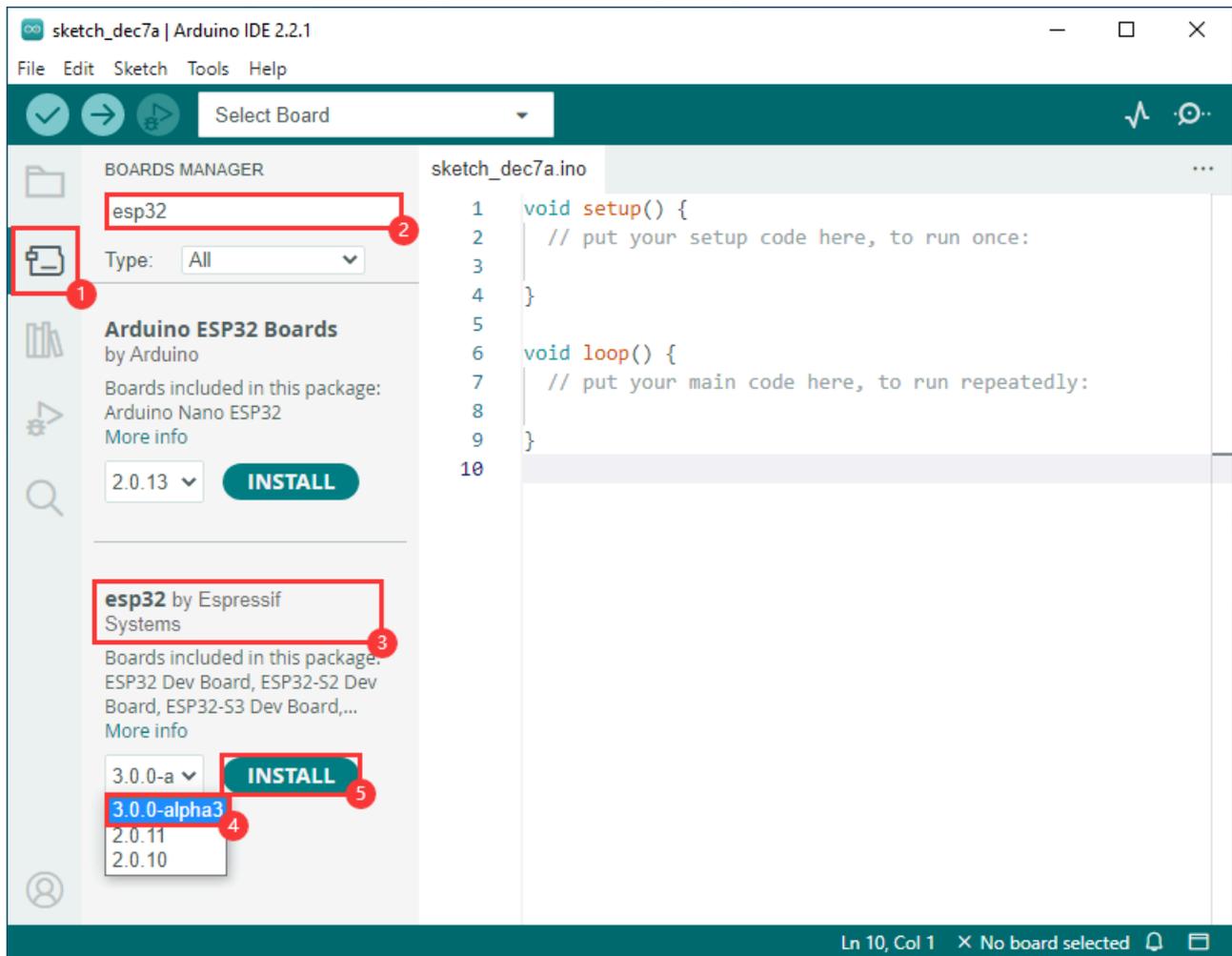
(/wiki/File:ESP32-C6-DEV-KIT-N8-Arduino07.png)

3. Open the Arduino IDE and note that Tools in the menu bar selects the corresponding **Flash (8MB)** and

enables **PSRAM (8MB OPI)**, as shown in the following figure.

 (/wiki/File:ESP32-S3-
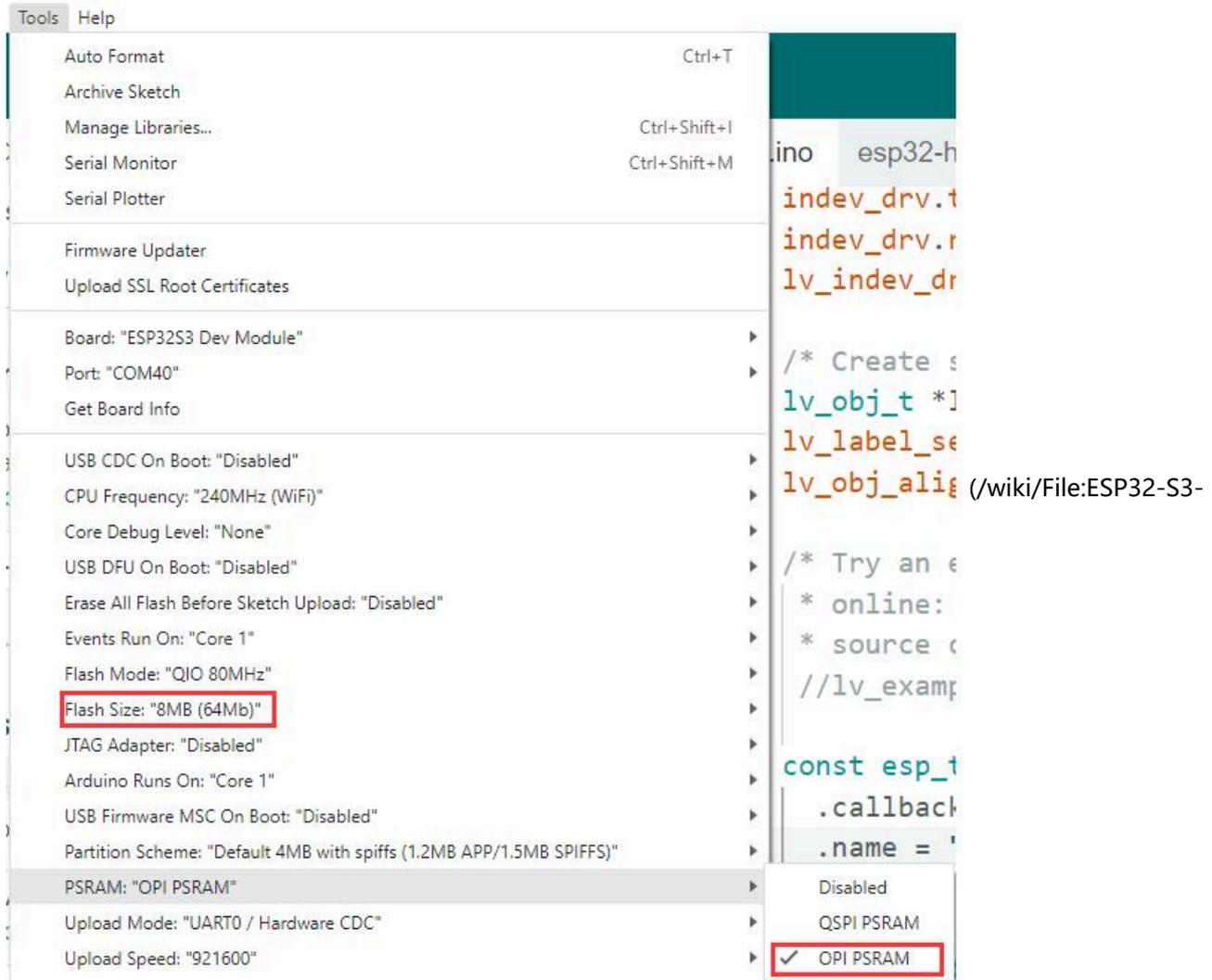
Touch-LCD-4.3-Ar02.jpg)

Library Installation

- lvgl libraries require configuration files after installation. It's recommended to directly copy the ESP32_Display_Panel, ESP32_IO_Expander, lvgl file, ESP_Panel_Conf.h, and lv_conf.h file of the ESP32-S3-Touch-LCD-4.3B_libraries (https://files.waveshare.com/wiki/ESP32-S3-Touch-LCD-4.3B/S3-4.3B-libraries.zip) to "C:\Users\xxxx\Documents\Arduino\libraries". Please note that "xxxx" represents your computer username.
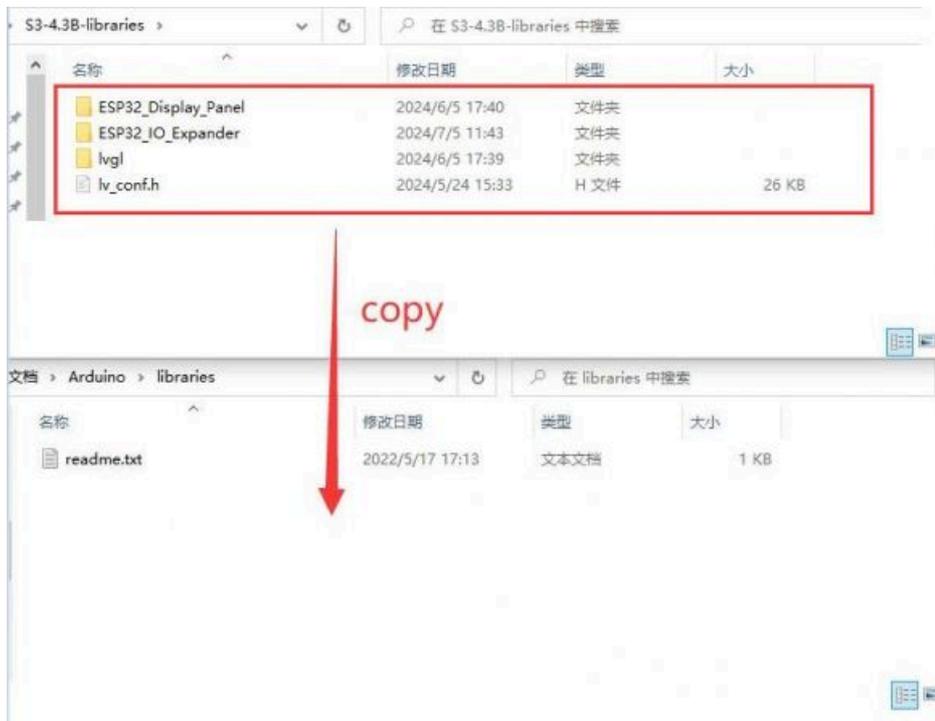
(/wiki/File:ESP32-S3-Touch-

LCD-4.3B-06.jpg)

- After copying:
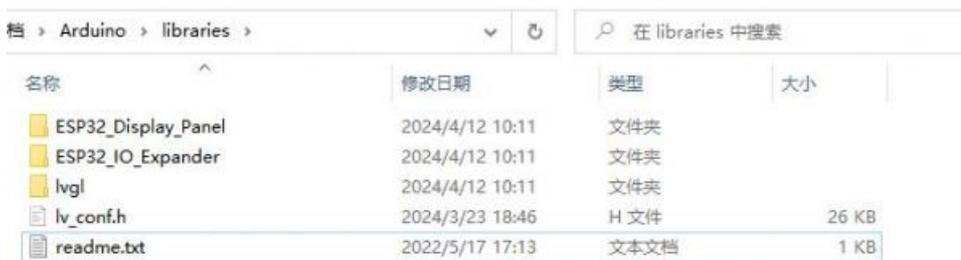
(/wiki/File:ESP32-S3-Touch-

LCD-7-07.jpg)

Sample Demo

Arduino

> **Note:** Before using the Arduino demos, please check whether the Arduino IDE environment and download settings are correctly configured, for details, please check the Arduino Configure.

- Pleas configure as shown below, otherwise the USB port will not output any information:

```
USB CDC On Boot should be set as Enabled
Flash Size should be set as 16MB(128Mb)
```

To use the LCD, you should configure as shown below:

```
USB CDC On Boot should be set as Enabled
Flash Size should be set as 16MB(128Mb)
PSRAM should be set as OPI PSRAM
```

- Before using the Arduino example, the software library should be set as shown below:

- Using the library (https://files.waveshare.com/wiki/ESP32-S3-Touch-LCD-7/ESP32_S3_Display_libraries.zip) file we provided.
- Install v3.0.2-h (https://github.com/esp-arduino-libs/arduino-esp32-sdk?tab=readme-ov-file) for Arduino esp32.
- The username of Arduino IDE must be English!

If you install a different version of Arduino esp32, there may be errors, we recommend installing the version we provide for testing and development.

I2C_Test

I2C_Test example, used to **test the I2C sockets**, this interface connects to GPIO8 (SDA) and GPIO9 (SCL) for I2C communication.

- Use this demo to scan all slave addresses of I2C devices.
- After uploading the demo, connects "HY2.0 2P to DuPont Male 4P 10cm" to the I2C socket. Then connect to the I2C device, open SSCOM, and you can see the scanned I2C addresses.
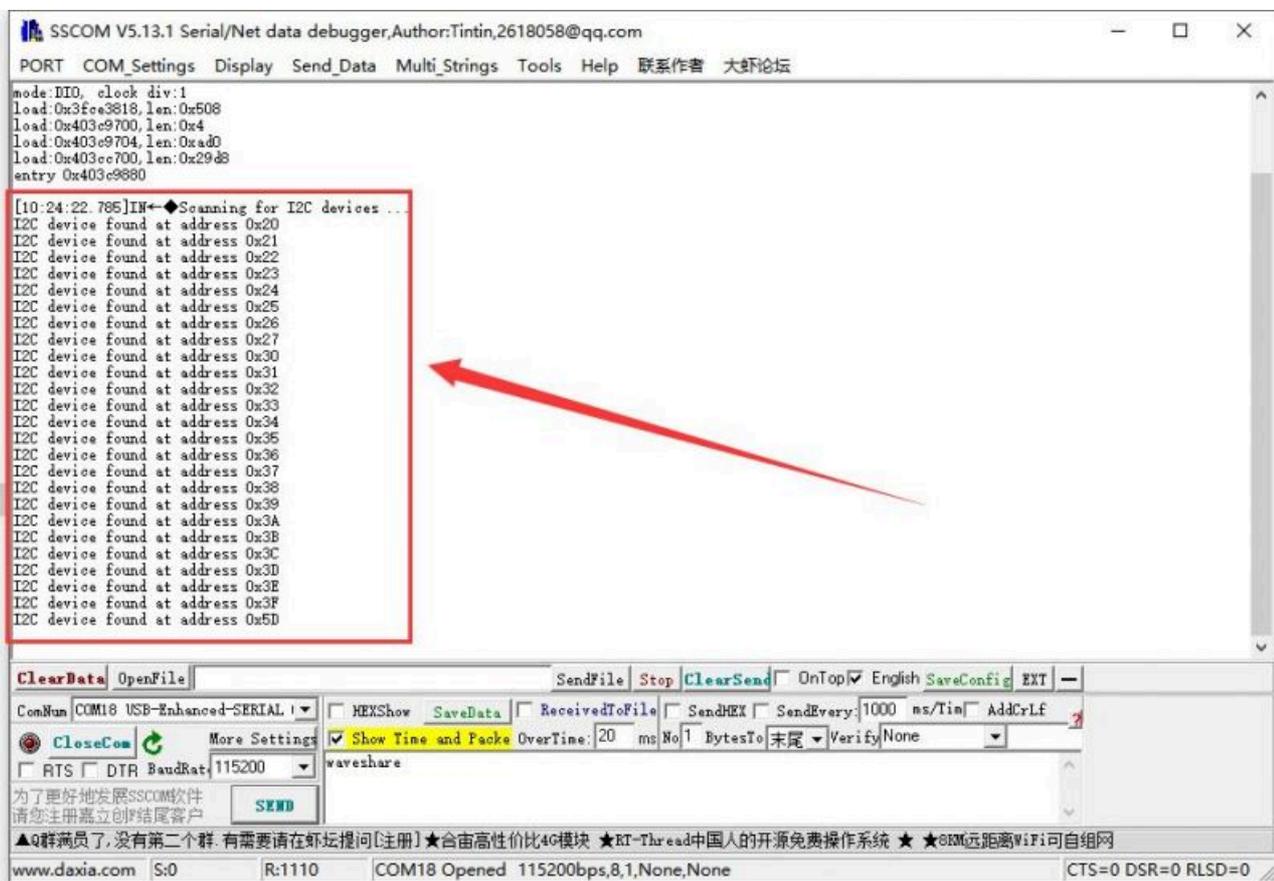
(/wiki/File:I2C_TEST.jpg)

RS485_Test

RS485_Test example is for **testing RS-485 socket**. This interface connects to GPIO44(TXD) and GPIO43(RXD) for RS485 communication.

- After uploading the code, the demo needs to use the USB TO RS485 (https://www.waveshare.com/usb-to-rs485.htm) converter, connect the RS-485 socket to the "HY2.0 2P to Dupont male 2P 10cm", and then connected to the USB to RS485 converter, USB to RS485 converter connected to the computer.
- Open the SSCOM (https://files.waveshare.com/upload/b/b3/Sscom5.13.1.zip), send RS485 messages to the ESP32-S3-Touch-LCD-4.3B, ESP32-S3-Touch-LCD-4.3B will send the received message back to the SSCOM,

pay attention to the need to select the correct COM port and baud rate, check "AddCrLf " before sending a message.
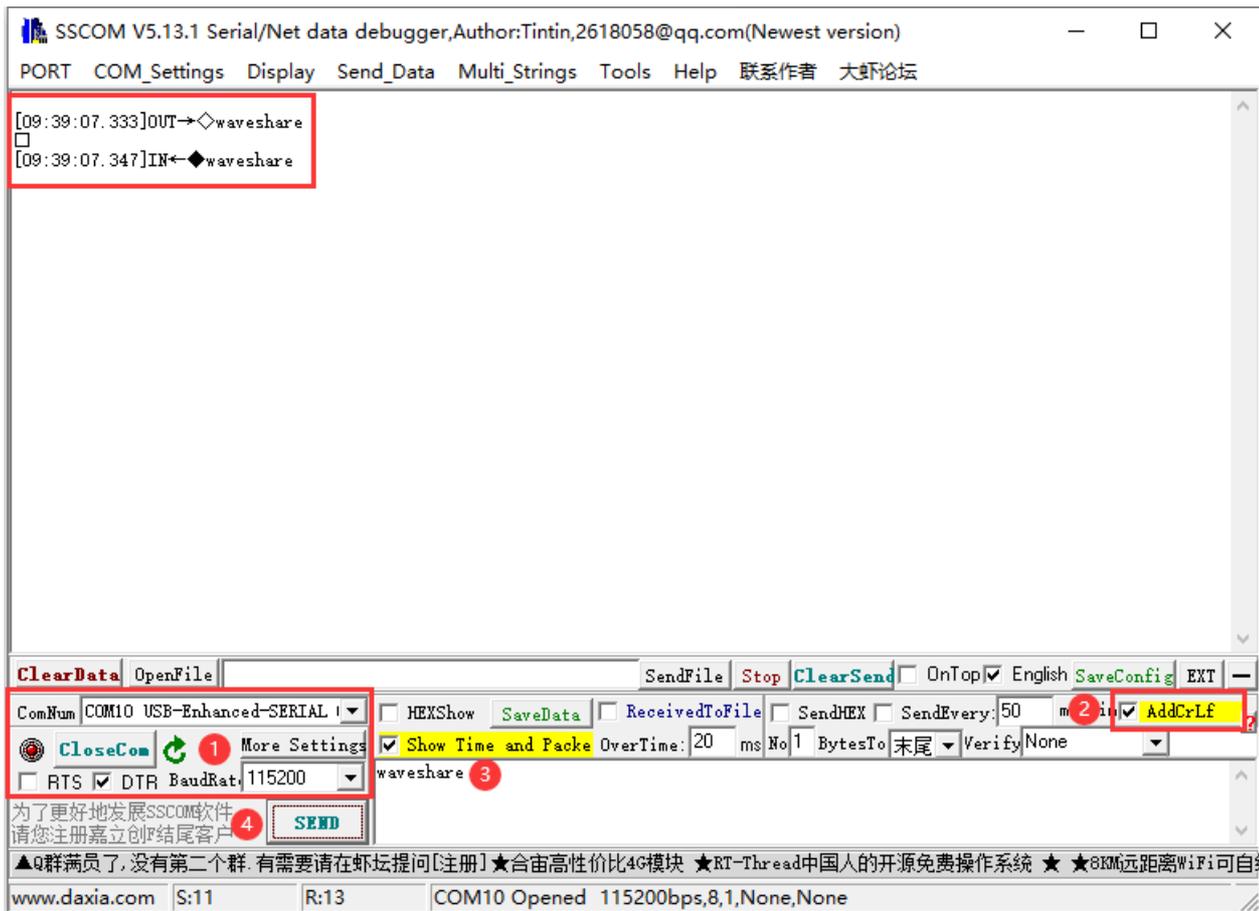
(/wiki/File:ESP32-S3-Touch-LCD-4.3-UART-test.png)

SD_Test

SD_Test is for **testing the SD card slot**. First, insert the SD card, upload the demo, and then read/write the SD card.

- After uploading the demo, ESP32-S3-Touch-LCD-4.3B will recognize the type and size of the SD card, and then you can operate the files on the SD card:
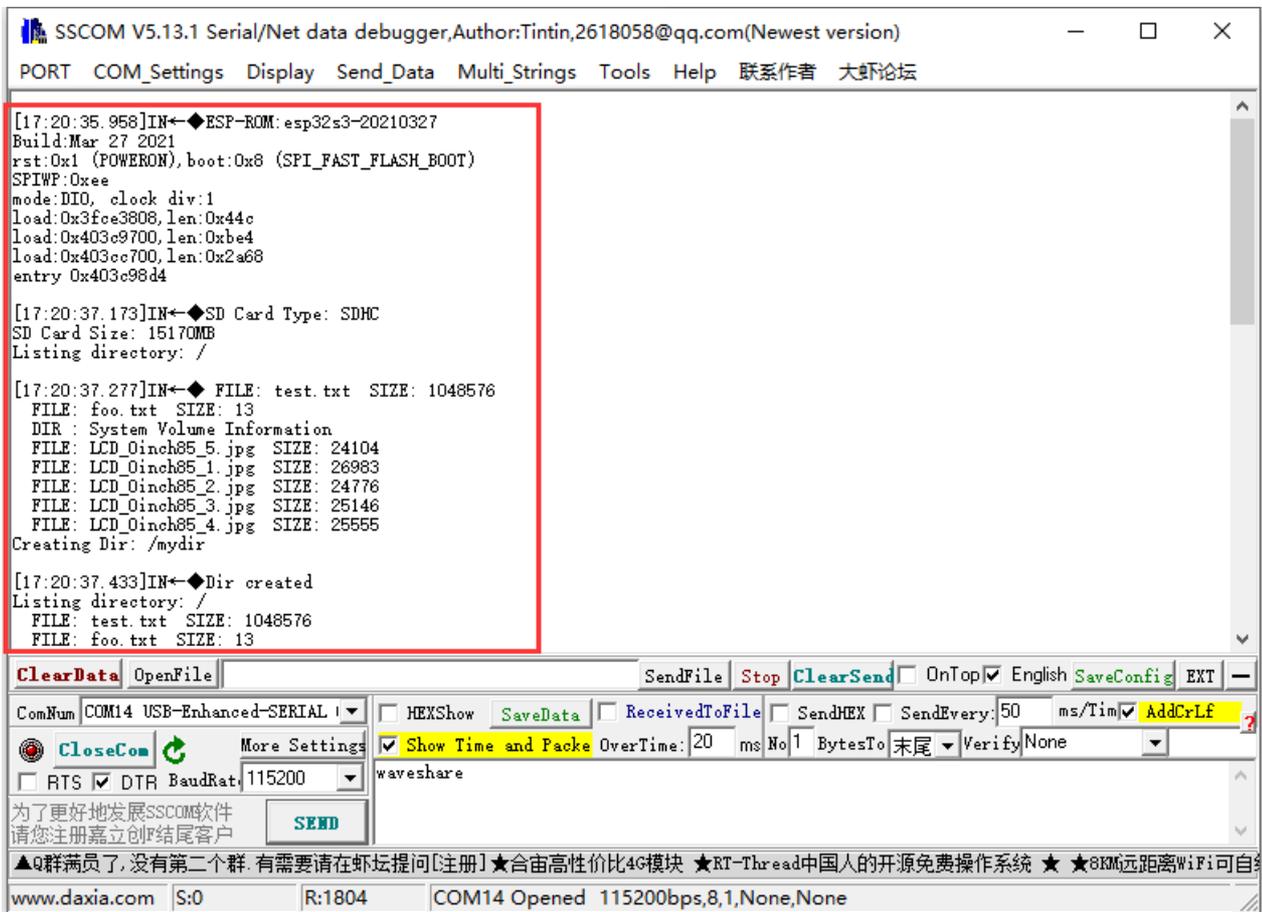
(/wiki/File:ESP32-S3-Touch-LCD-4.3-SDtest01.png)

RTC_Test

RTC_Test example for **RTC clock with RTC interrupt**.

- After burning the code, it will set the time, start the alarm, then read the current time and wait for the alarm to be entered.
- The arrow is to trigger the alarm, the red box is to read the time.

(/wiki/File:ESP32-S3-Touch-LCD-4.3B-04.jpg)

IO_Test

IO_Test example **tests the use of isolated IO**, and you need to connect DO0 with DI0, DO1 with DI1 first.

- After burning the code, the test passes with a green screen, and the test fails with a red screen.

TWAItransmit

TWAItransmit example is for **testing CAN socket**, and this interface can connect to GPIO15(TXD) and GPIO16(RXD) for CAN communication.

- After programming the code, using the "HY2.0 2P to DuPont male head 2P red-black 10cm" cable, and connect ESP32-S3-Touch-LCD-4.3B to the CAN H and CAN L pins of the USB-CAN-A (https://www.waveshare.com/usb-can-a.htm).
- Once you open the serial port debugging assistant, you can observe that the ESP32-S3-Touch-LCD-4.3B has started sending CAN messages.
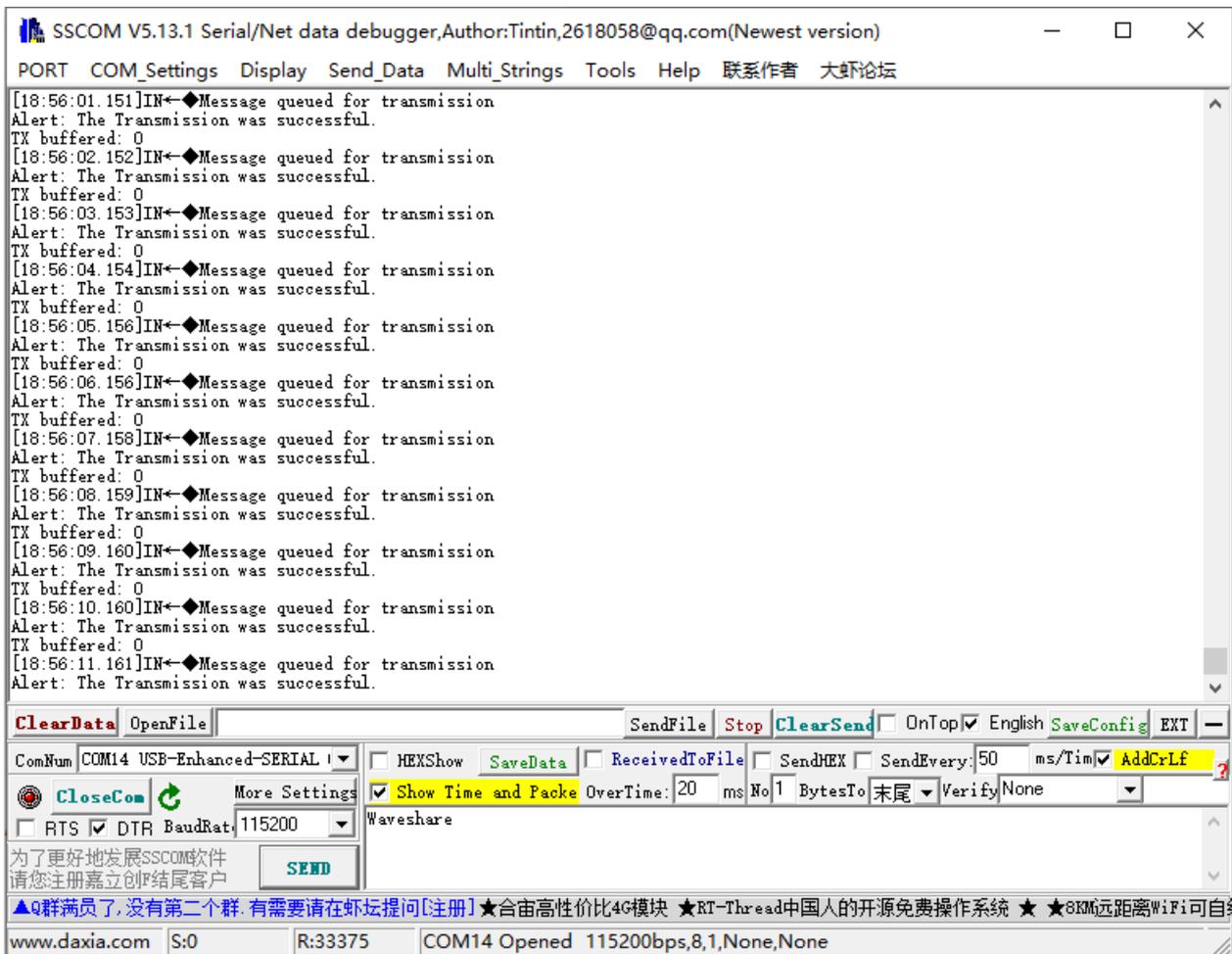
(/wiki/File:ESP32-S3-Touch-LCD-4.3-PORtest01.png)

- Connect the USB-CAN-A to the computer and open the USB-CAN-A_TOOL_2.0 (https://files.waveshare.com/wiki/USB-CAN-A/Tool/Progrom(English%EF%BC%89.zip). Select the corresponding COM port, set the baud rate to 2000000 as shown in the image, set the CAN baud rate to 50.000Kbps, and you can view the CAN messages sent by the ESP32-S3-Touch-LCD-4.3B.

Buy at : https://hubtronics.in/esp32-s3-touch-lcd-4.3b-box

(/wiki/File:ESP32-S3-Touch-LCD-4.3-PORtest102.png)

TWAIreceive

TWAIreceive example is for **testing CAN socket**, and this interface can connect to GPIO15(TXD) and GPIO16(RXD) for CAN communication.

- After uploading the code, use the "HY2.0 2P to DuPont male head 2P red-black 10cm" cable to connect the ESP32-S3-Touch-LCD-4.3B to the CAN H and CAN L pins of USB-CAN-A (https://www.waveshare.com/usb-can-a.htm).

- Connect the USB-CAN-A to the computer and open the USB-CAN-A_TOOL_2.0 (https://files.waveshare.com/wiki/USB-CAN-A/Tool/Progrom(English%EF%BC%89.zip). Select the corresponding COM port, set the port baud rate to 2000000 as indicated in the image, and set the CAN baud rate to 500.000Kbps. With these settings, you'll be able to send CAN messages to the ESP32-S3-Touch-LCD-4.3B.

(/wiki/File:ESP32-S3-Touch-LCD-4.3-PORtest03.png)

- Open the serial port debugging assistant, send data on USB-CAN-A_TOOL_2.0, and you can see ESP32-S3-Touch-LCD-4.3B starts to receive CAN messages (If there are any reception errors, try resetting the devices multiple times and restarting the software. Please be patient and allow some time for the reception process.)
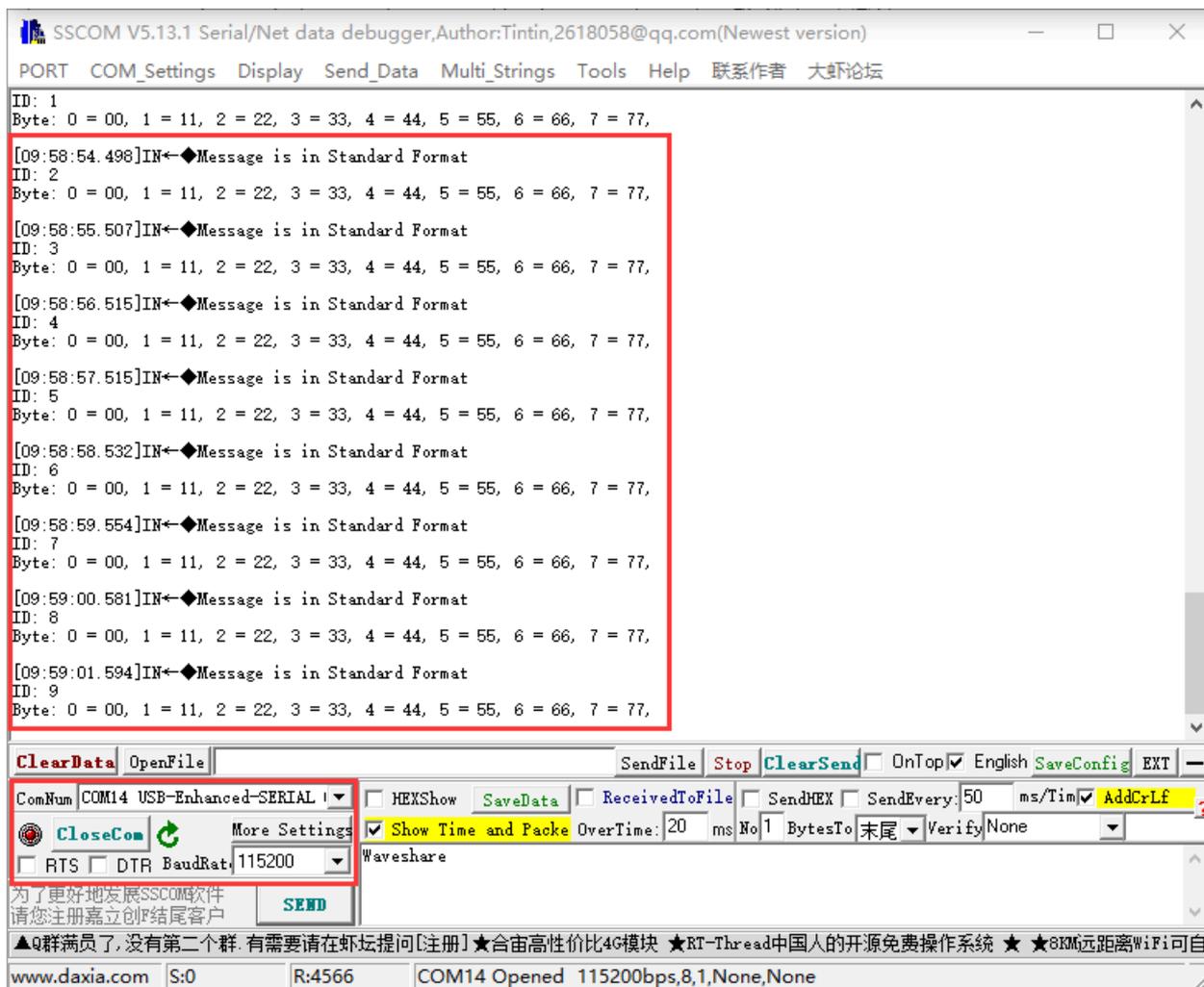
(/wiki/File:ESP32-S3-Touch-LCD-4.3-PORtest04.png)

lvgl_Porting

lvgl_Porting example is for **testing RGB touch screen**.

- After uploading the code, you can try to make a series of touch screen operation. Also, we provide LVGL porting examples for users (If there's no screen response after burning the code, check if the Arduino IDE -> Tools settings are correctly configured: choose the corresponding Flash (8MB) and enable PSRAM (8MB OPI)).

DrawColorBar

DrawColorBar example is for **testing RGB screen**.

- After uploading the code, you should observe the screen displaying bands of blue, green, and red colors. (If the screen shows no response after burning the code, check if the Arduino IDE -> Tools settings are correctly configured: choose the corresponding Flash (8MB) and enable PSRAM (8MB OPI)).

ESP-IDF

> **Note:** Before using ESP-IDF examples, please ensure that the ESP-IDF environment and download settings are correctly configured. You can refer to the ESP-IDF environment setting for specific instructions on how to check and configure them.

I2C_Test

I2C_Test example is for **testing I2C interface**, scanning all device addresses of I2C.

After uploading the code, connect the I2C device (in this case, using the BME680 Environmental Sensor (http s://www.waveshare.com/bme68x-environmental-sensor.htm)) to the corresponding pins on the ESP32-S3-Touch-LCD-4.3B. Then, open the serial debugging assistant (https://files.waveshare.com/upload/b/b3/Sscom5. 13.1.zip), select a baud rate of 115200, and choose the corresponding COM port for communication (make sure to close the ESP-IDF's COM port first, as it may occupy the COM port to prevent the serial port from opening).

(/wiki/File:ESP32-S3-Touch-LCD-4.3-110.png)

- Press the Reset button of ESP32-S3-Touch-LCD-4.3B, SSCOM will print the message, enter i2cdetect as shown in the following figure, it will print 77, and the I2C socket test passes.
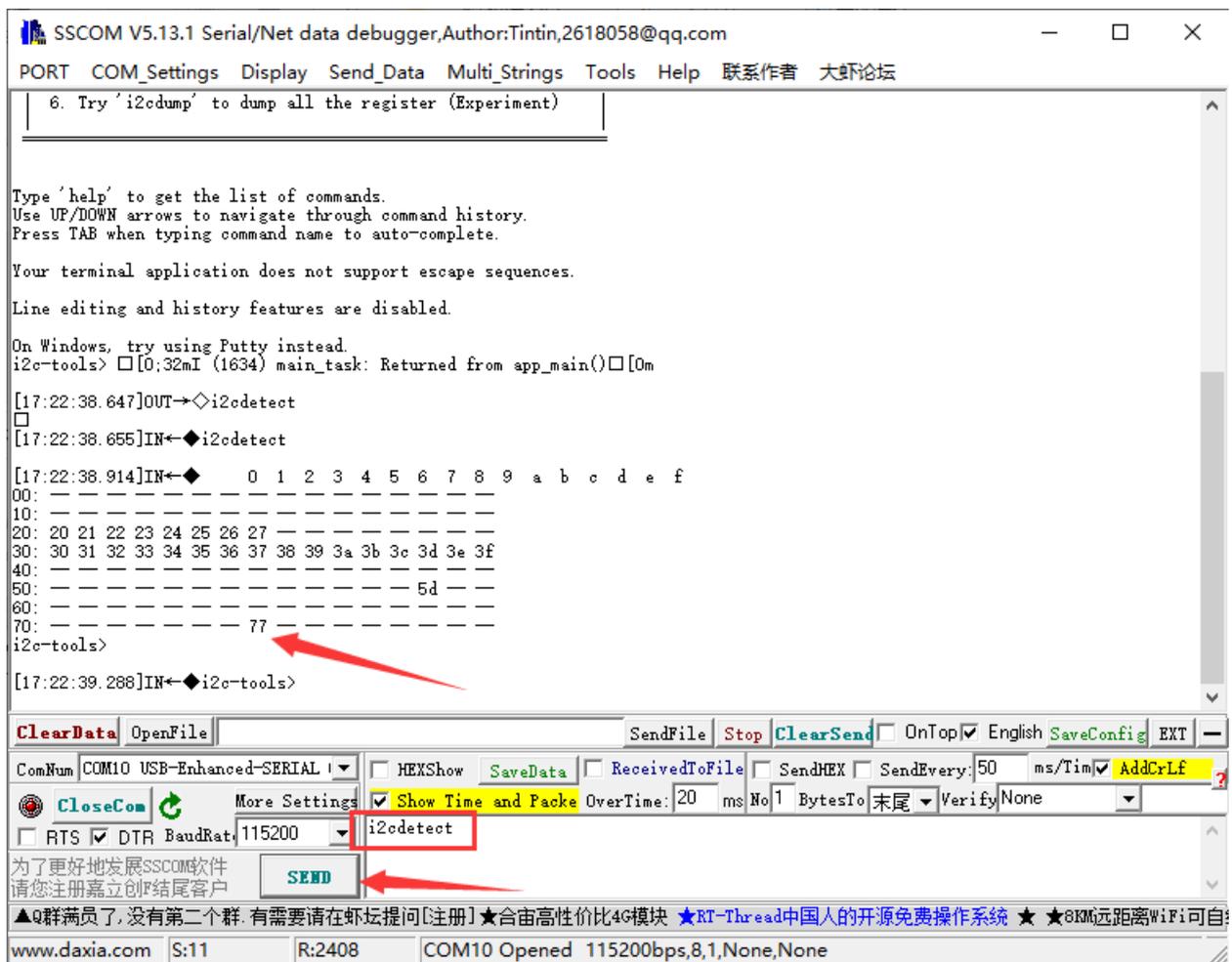
(/wiki/File:ESP32-S3-Touch-LCD-4.3-111.png)

RS485_Test

RS485_Test is for **testing the RS485 socket**.

- After uploading the code, connect the USB to RS485 (https://www.waveshare.com/usb-to-rs485.htm) to the A and B pins of the ESP32-S3-Touch-LCD-4.3B. After connecting the USB to RS485 to the computer, open the SSCOM (https://files.waveshare.com/upload/b/b3/Sscom5.13.1.zip) and select the corresponding COM port for communication.

- Choose a baud rate of 115200 as shown in the diagram below. Sending any character will result in a loopback display. Testing of the RS485 socket has passed.

(/wiki/File:ESP32-S3-Touch-LCD-4.3-UART-test.png)

SD_Test

SD_Test example, used to **test the use of SD card slot**, you need to insert the SD card first, burn the code and then read and write to the SD card.

- After burning the code, ESP32-S3-Touch-LCD-4.3b will print the information about the memory card, such as the name, type, capacity and the maximum frequency supported, then create the file, write the file, rename the file and read the renamed file.
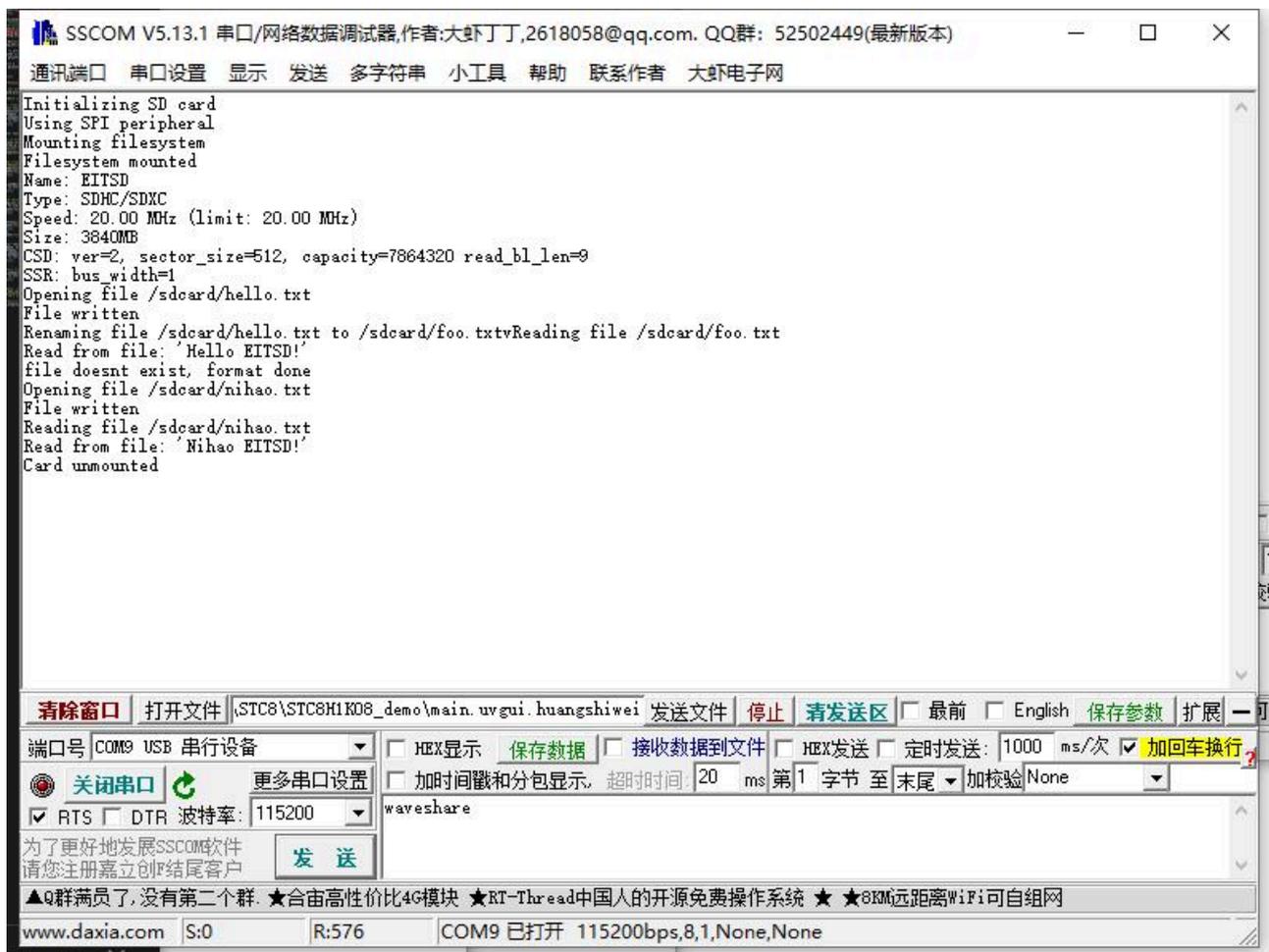
(/wiki/File:ESP32-S3-Touch-LCD-4.3B-02.jpg)

RTC_Test

RTC_Test example for **RTC clock with RTC interrupt**.

- After burning the code, it will set the time, start the alarm, then read the current time and wait for the alarm to be entered.
- The arrow is to trigger the alarm, the red box is to read the time.

(/wiki/File:ESP32-S3-Touch-

LCD-4.3B-04.jpg)

IO_Test

IO_Test example tests **the use of isolated IO**, and you need to connect DO0 with DI0, DO1 with DI1 first.

- After burning the code, the test passes with a green screen, and the test fails with a red screen.

TWAItransmit
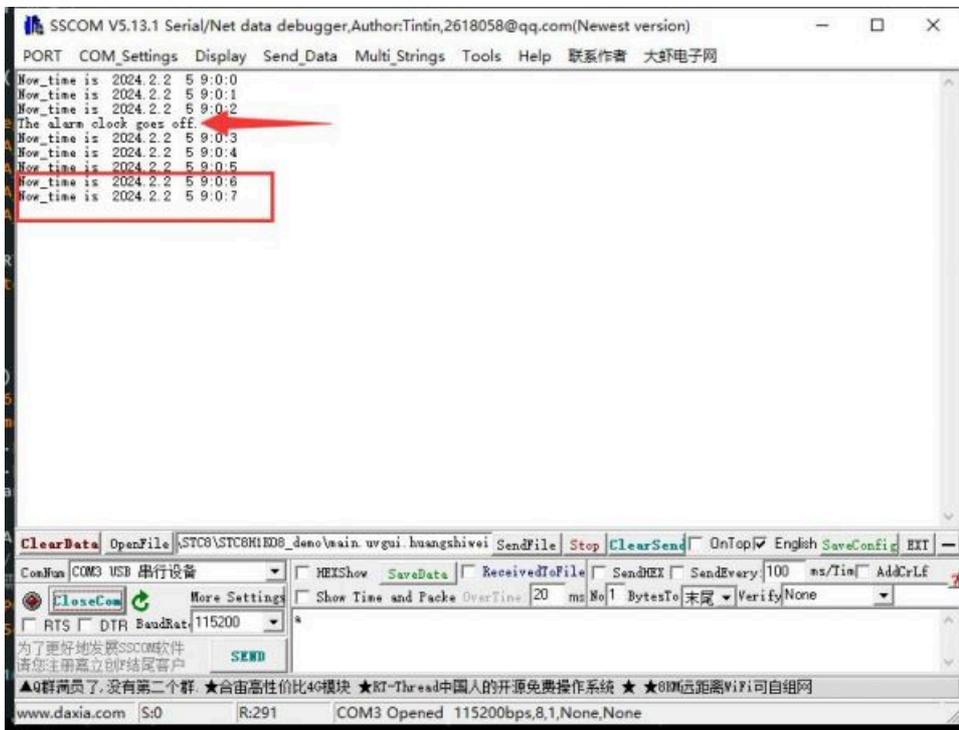
TWAItransmit example is for **testing CAN socket**. This interface can connect GPIO15(TXD) and GPIO16(RXD) for CAN communication.

- After uploading the demo, you can use the "HY2.0 2P to DuPont male head 2P red-black 10cm" cable to connect the ESP32-S3-Touch-LCD-4.3B to the CAN H and CAN L pins of USB-CAN-A (https://www.waveshar e.com/usb-can-a.htm).
- Open the SSCOM, and you can see the ESP32-S3-Touch-LCD-4.3B starts to send the CAN message.

Buy at : https://hubtronics.in/esp32-s3-touch-lcd-4.3b-box
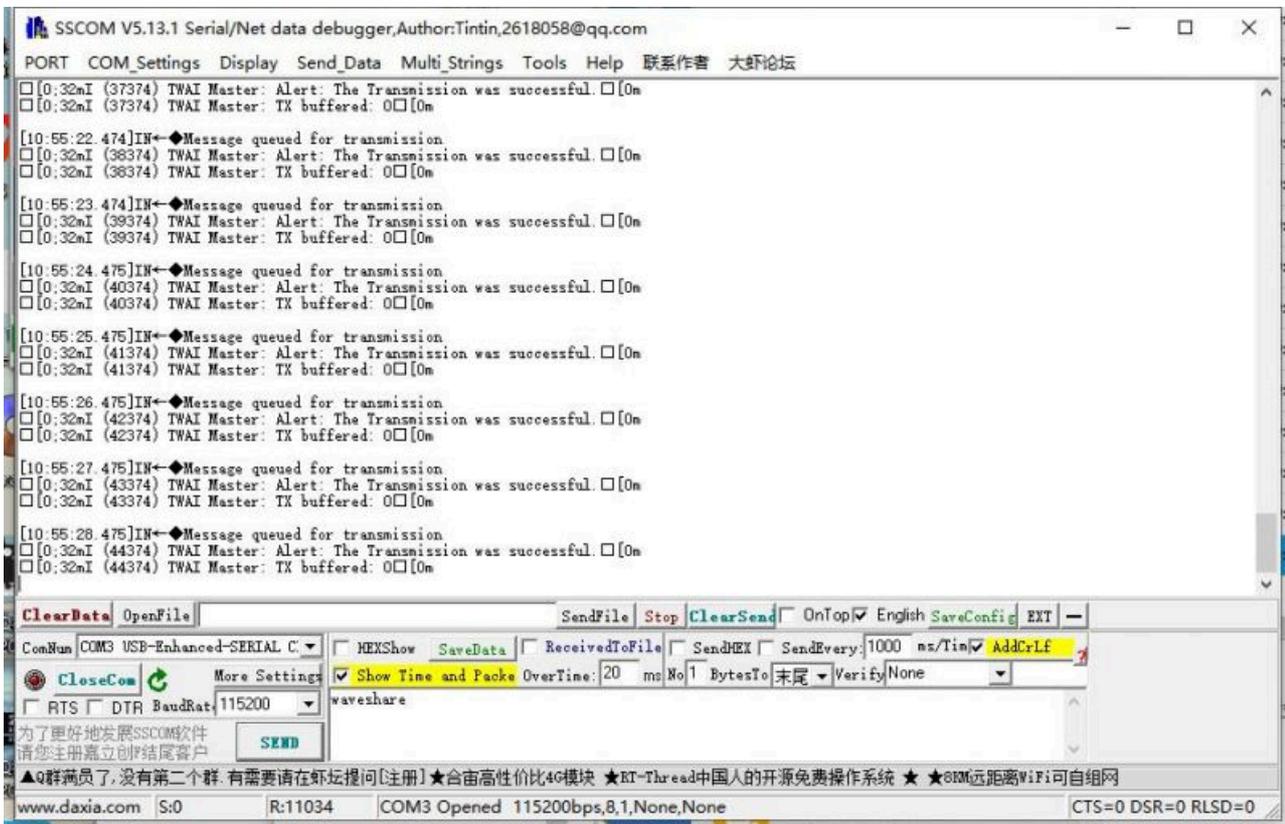
(/wiki/File:ESP32-S3-Touch-LCD-7_twa.jpg)

- Connect the USB-CAN-A to the computer, open USB-CAN-A-Tool-2.0 (https://files.waveshare.com/wiki/USB-CAN-A/Tool/Progrom(English%EF%BC%89.zip), select the corresponding COM port, 2000000 as the baud rate, 50.000Kbps as the CAN baud rate, and then you can see the CAN message sent from the ESP32-S3-Touch-LCD-4.3B.
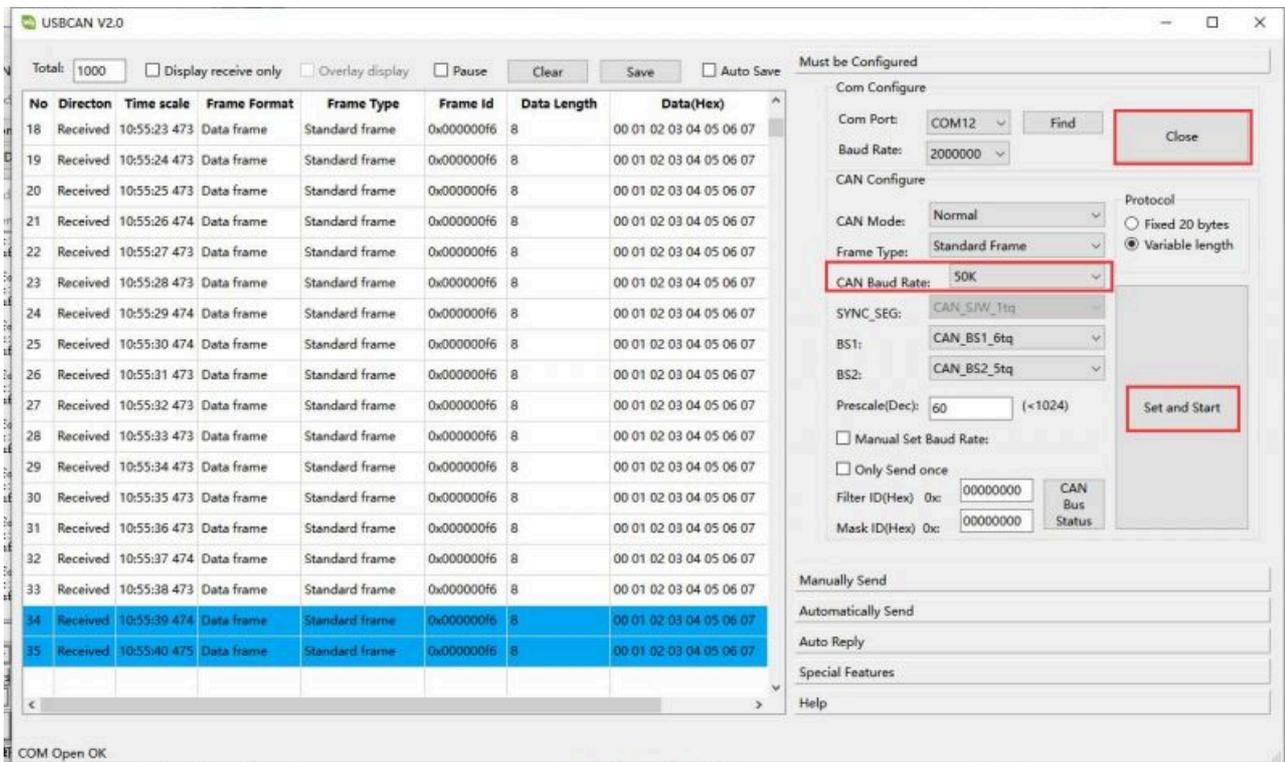
(/wiki/File:ESP32-S3-Touch-LCD-7_twa2.jpg)

TWAIreceive

TWAIreceive example is for **testing CAN socket**. This interface can connect GPIO20 (TXD) and GPIO19 (RXD) for CAN communication.

- After uploading the demo, you can use the "HY2.0 2P to DuPont male head 2P red-black 10cm" cable to connect the ESP32-S3-Touch-LCD-4.3B to the CAN H and CAN L pins of USB-CAN-A (https://www.waveshare.com/usb-can-a.htm).

- Connect the USB-CAN-A to the computer, open USB-CAN-A-Tool-2.0 (https://files.waveshare.com/wiki/USB-CAN-A/Tool/Progrom(English%EF%BC%89.zip), select the corresponding COM port, 2000000 as the baud rate, 500.000Kbps as the CAN baud rate, and then you can see the CAN message is sent to the ESP32-S3-Touch-LCD-4.3B.


(/wiki/File:ESP32-S3-Touch-LCD-4.3-PORtest03.png)

- Open the Serial Debug Assistant, where you can see that the ESP32-S3-Touch-LCD-4.3B has started receiving CAN messages. If there are reception errors, try resetting the device multiple times and restarting the software. Please be patient and wait for the process to complete.
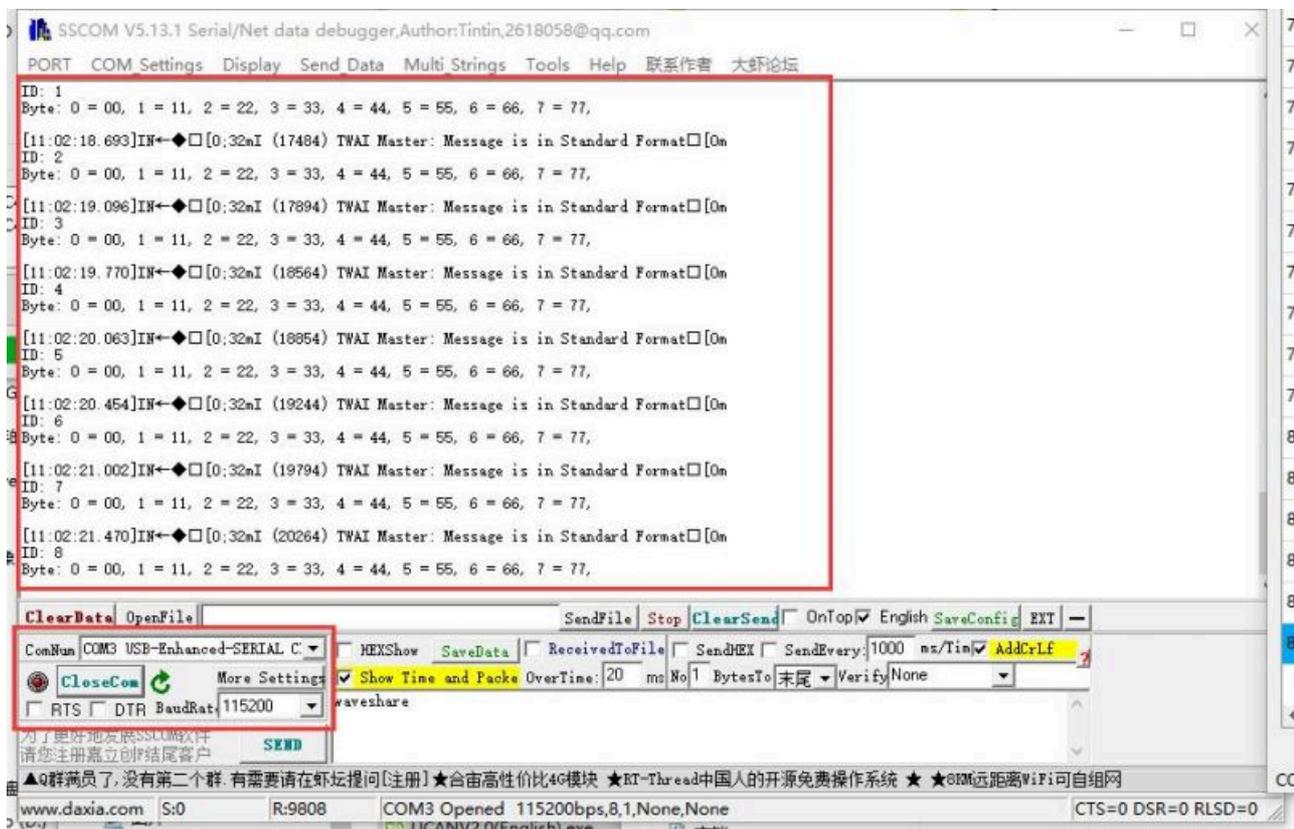
(/wiki/File:ESP32-S3-Touch-LCD_twa4.jpg)

lvgl_Porting

lvgl_Porting example is for **testing RGB touch screen**.

- After uploading the code, you can test the touching on the screen, and the demo also supports LVGL porting for users.

- For RGB LCD driver, you can refer to this link (https://docs.espressif.com/projects/esp-idf/zh_CN/release-v5. 3/esp32s3/api-reference/peripherals/lcd/rgb_lcd.html).

- For GT911 driver, you can refer to this link (https://github.com/espressif/esp-bsp/tree/master/components/l cd_touch/esp_lcd_touch_gt911).

Resource

Document

- Schematic (https://files.waveshare.com/wiki/ESP32-S3-Touch-LCD-4.3B/ESP32-S3-Touch-LCD-4.3B-Sch.pdf)
- ESP32 Arduino Core's documentation (https://docs.espressif.com/projects/arduino-esp32/en/latest/index.ht ml)
- arduino-esp32 (https://github.com/espressif/arduino-esp32)
- ESP-IDF (https://github.com/espressif/esp-idf)
- 3D Drawing (https://files.waveshare.com/wiki/ESP32-S3-Touch-LCD-4.3B/ESP32-S3-Touch-LCD-4in3B_Drawi ng.zip)

Demo

- ESP32-S3-Touch-LCD-4.3B_libraries (https://files.waveshare.com/wiki/ESP32-S3-Touch-LCD-4.3B/S3-4.3B-libr aries.zip)

- Sample demo (https://files.waveshare.com/wiki/ESP32-S3-Touch-LCD-4.3B/ESP32-S3-Touch-LCD-4.3B_Code.zip)

Software

- Sscom5.13.1 (https://files.waveshare.com/upload/b/b3/Sscom5.13.1.zip)
- Arduino IDE (https://www.arduino.cc/en/software)
- USB-CAN-A_TOOL_1.2 (https://files.waveshare.com/upload/8/8a/USB-CAN-A_TOOL_EN.zip)
- USB-CAN-A_TOOL_2.0 (https://files.waveshare.com/wiki/USB-CAN-A/Tool/Progrom(English%EF%BC%89.zip)
- ESP32_S3_flash_download_tool (https://files.waveshare.com/wiki/ESP32-S3-Touch-LCD-4.3B/ESP32_S3_flash_download_tool.zip)

Datasheet

- ESP32-S3 Wroom Datasheet (https://files.waveshare.com/wiki/ESP32-S3-Touch-LCD-4.3/Esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf)
- CH343 Datasheet (https://files.waveshare.com/upload/a/a3/CH343DS1-en.PDF)
- TJA1051 Datasheet (https://www.nxp.com/docs/en/data-sheet/TJA1051.pdf)
- GT911 datasheet (https://files.waveshare.com/upload/e/eb/GT911.pdf)

FAQ

Question:ESP32-S3-Touch-LCD-4.3B CAN reception failure?

Answer:

① Restart the COM port in UCANV2.0.exe and press the ESP32-S3-Touch-LCD-4.3B reset button multiple times.

② Uncheck **DTR** and **RTS** in the serial port debugging assistant.

Question:ESP32-S3-Touch-LCD-4.3B shows no response after uploading an Arduino demo for RGB screen displaying?

Answer:

If there's no screen response after programming the code, check whether the correct configurations are set in Arduino IDE -> Tools: Choose the corresponding Flash (8MB) and enable PSRAM (8MB OPI).

Question:ESP32-S3-Touch-LCD-4.3B fails to compile an Arduino demo for the RGB screen and shows errors?

Answer:

Check if the "ESP32-S3-Touch-LCD-4.3B-libraries" library (https://files.waveshare.com/wiki/ESP32-S3-Touch-LCD-4.3B/S3-4.3B-libraries.zip) is installed. Please refer to installation steps.

Question:Why burn lvgl program missing lv_cong.h when all libraries are installed?

Answer:

I can't retrieve the library file because the path to install the library is in Chinese.

Question:Why is the screen not displaying?

Answer:

You can refer to the following steps to run the demo for comparison:

- 1) Before running the program, please install the library (https://files.waveshare.com/wiki/ESP32-S3-Touch -LCD-4.3B/S3-4.3B-libraries.zip)
- 2) Run and burn original program (https://files.waveshare.com/wiki/ESP32-S3-Touch-LCD-4.3B/ESP32-S3-Touch-LCD-4.3B_Code.zip)

Question:Why does it show "fatal error:esp_ memory_ utils.h:No such file or directory" when compiling the example with Arduino IDE?

Answer:

To solve this problem, please install the Arduino esp32 v3.0.2-h.

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.wave share.com/)

Buy at : https://hubtronics.in/esp32-s3-touch-lcd-4.3b-box