ESP32-S3-Touch-LCD-4.3B

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.

Features

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

Hardware Description

Onboard Interface

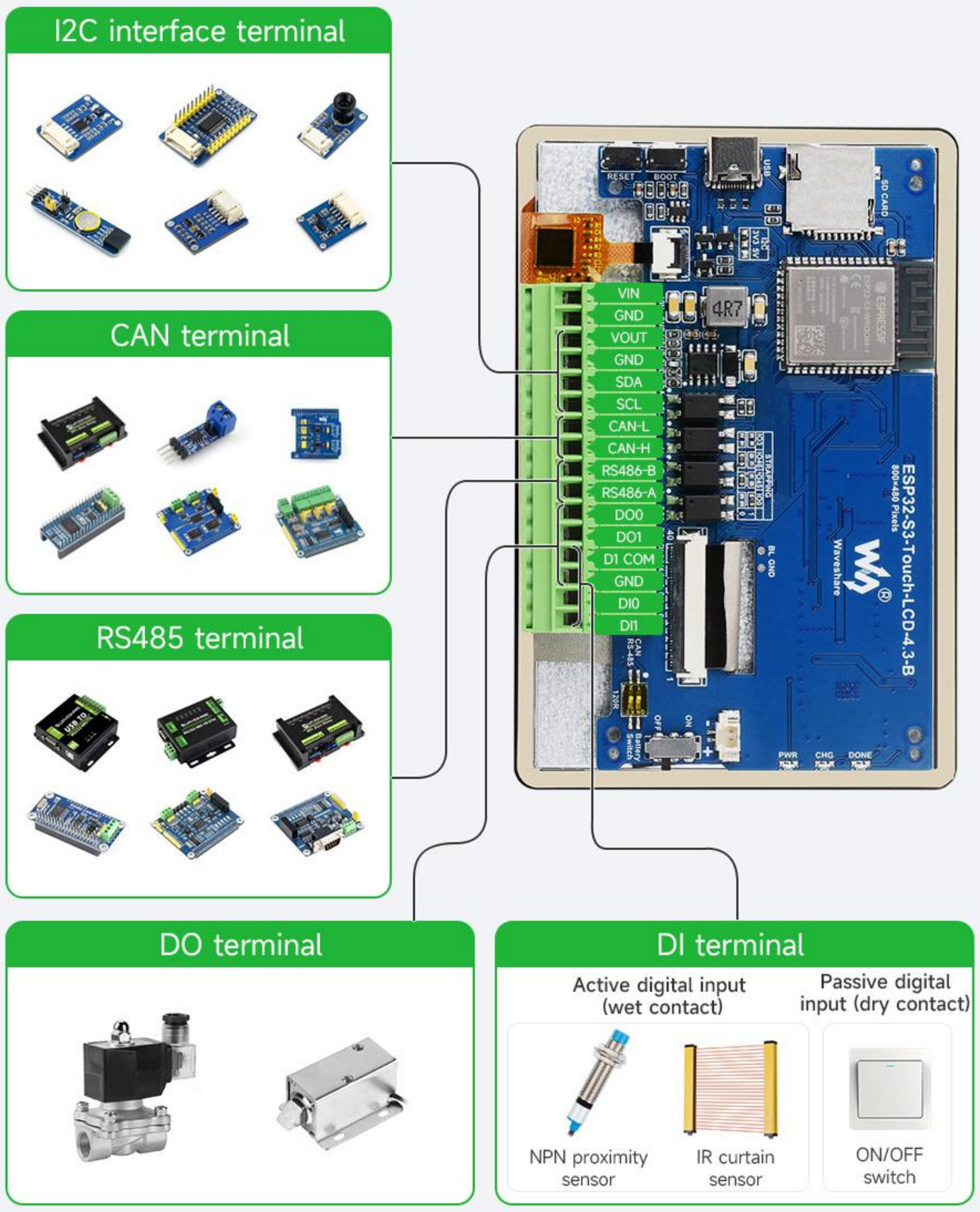
- CAN interface: Transceiver control, data analysis, acquisition and monitoring of CAN bus networks.
- I2C interface: ESP32-S3 provides multi-lane hardware I2C, currently uses GPIO8(SDA) and GPIO9(SCL) pins as I2C bus for loading IO expansion chip, touch interface and I2C interface.
- RS485 interface: 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.
- MX1.25 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.

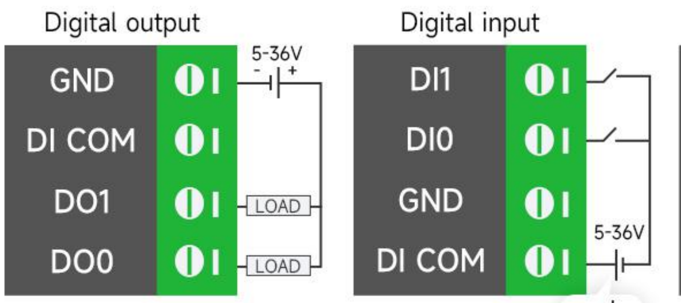
PIN Connection

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

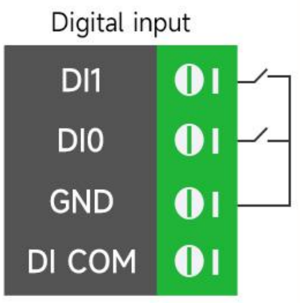
Hardware Connection





Digital output connection diagram

Digital input wet contact connection diagram



Digital input dry contact connection diagram

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 (LV
GL 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
```

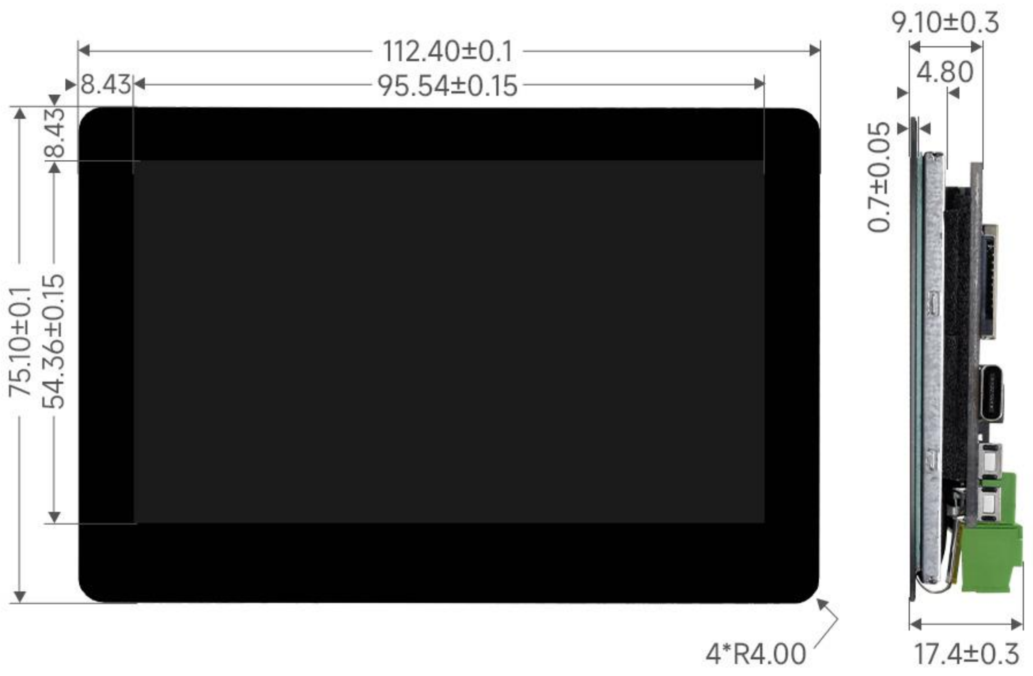
For more details about LCD and LVGL performance, you can refer to this document.

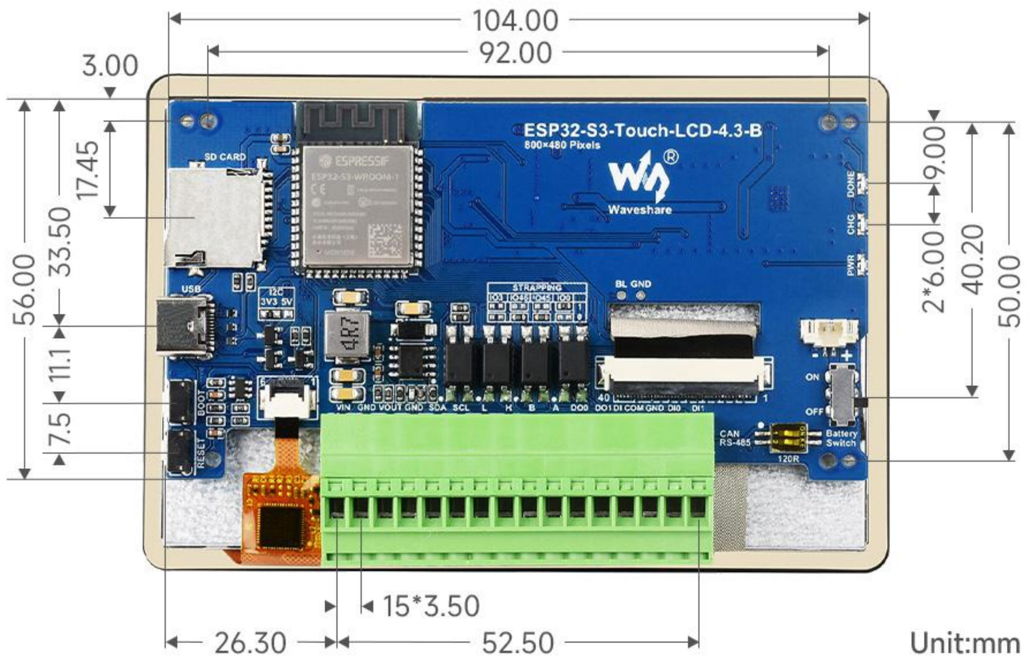
The PH2.0 lithium battery socket only supports single-cell 3.7V lithium batteries. Do not use multiple battery packs for charging and discharging simultaneously. It is recommended to use a single-cell battery with a capacity of below 2000mAh.

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:

```
     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





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 development tutorial.

ESP32 series ESP-IDF 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.

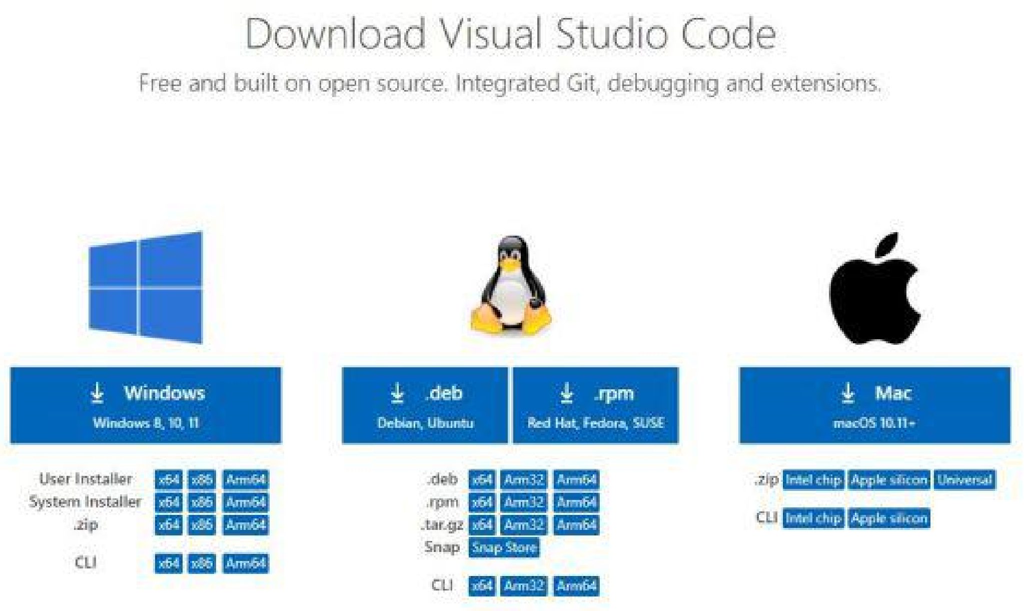
ESP-IDF

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

Develop with VSCode Plug-in

Install VSCode

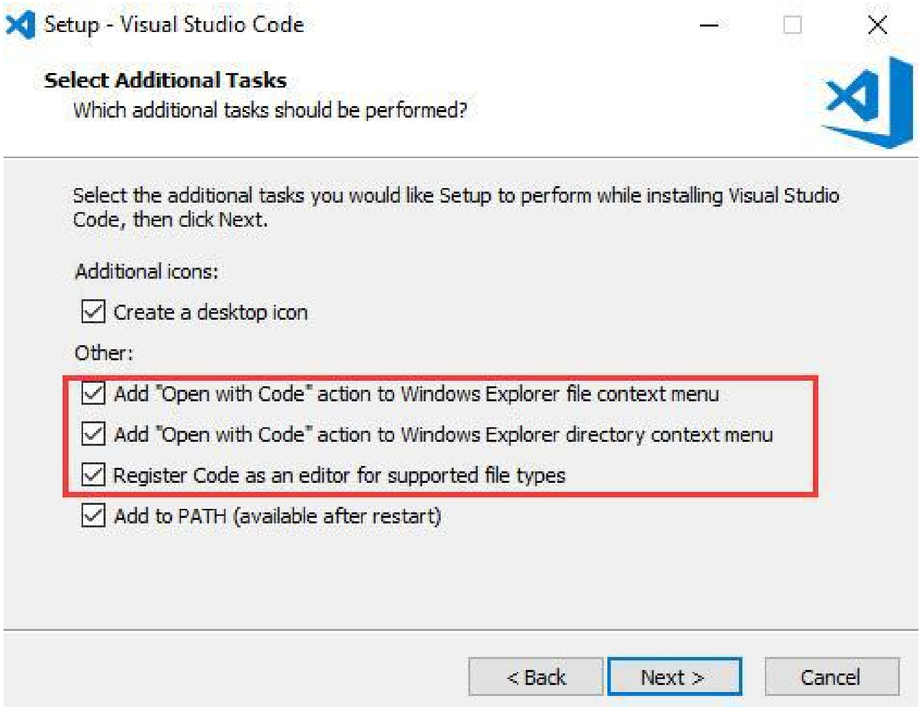
1. Open the download page of the official VSCode website, and select the corresponding system and system bit to download.


Download Visual Studio Code
Free and built on open source. Integrated Git, debugging and extensions.
Windows
Windows 8, 10, 11
.deb
Debian, Ubuntu
.rpm
Red Hat, Fedora, SUSE
Mac
macOS 10.11+
User Installer x64 x86 Arm64
System Installer x64 x86 Arm64
.zip x64 x86 Arm64
CLI x64 x86 Arm64
.deb x64 Arm32 Arm64
.rpm x64 Arm32 Arm64
.tar.gz x64 Arm32 Arm64
Snap Snap Store
CLI x64 Arm32 Arm64
.zip Intel chip Apple silicon Universal
CLI Intel chip Apple silicon

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.


Setup - Visual Studio Code

Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.

Additional icons:

- [✓] Create a desktop icon

Other:

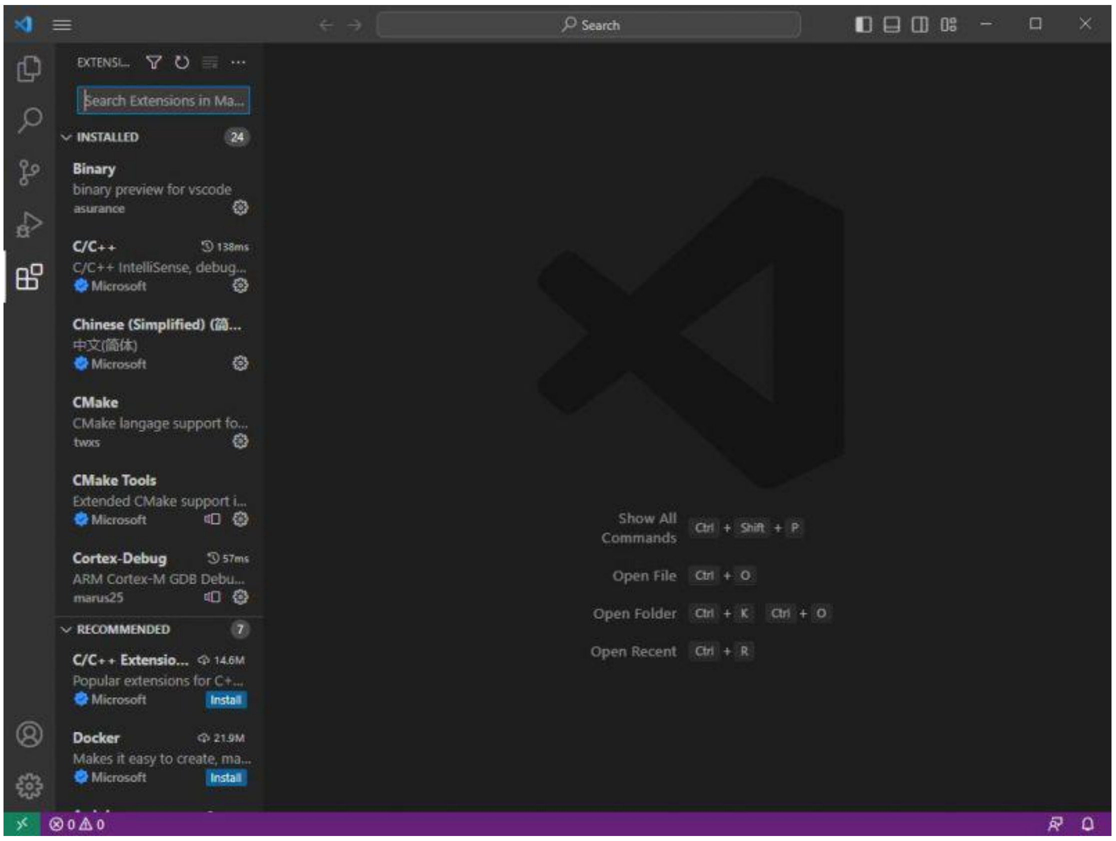
- [✓] Add "Open with Code" action to Windows Explorer file context menu
- [✓] Add "Open with Code" action to Windows Explorer directory context menu
- [✓] Register Code as an editor for supported file types
- [✓] Add to PATH (available after restart)

< Back | Next > | Cancel

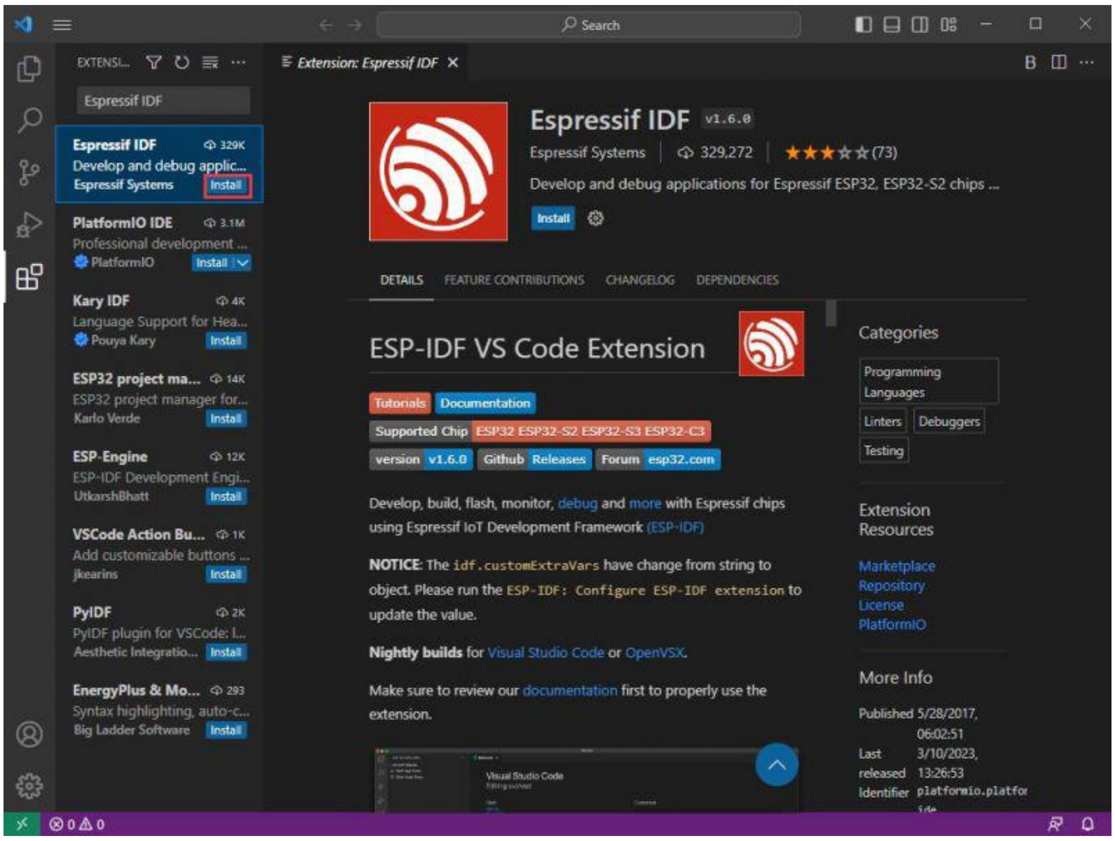
Install Espressif IDF Plug-in

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.


EXTENSI...
Search Extensions in Ma...
INSTALLED 24
Binary
binary preview for vscode
asurance
C/C++ 138ms
C/C++ IntelliSense, debug...
Microsoft
Chinese (Simplified) (简...
中文(简体)
Microsoft
CMake
CMake langage support fo...
twxs
CMake Tools
Extended CMake support i...
Microsoft
Cortex-Debug 57ms
ARM Cortex-M GDB Debu...
marus25
RECOMMENDED 7
C/C++ Extensio... 14.6M
Popular extensions for C+...
Microsoft
Install
Docker 21.9M
Makes it easy to create, ma...
Microsoft
Install
Show All Commands Ctrl + Shift + P
Open File Ctrl + O
Open Folder Ctrl + K Ctrl + O
Open Recent Ctrl + R

2. In the search bar, type Espressif IDF, select the corresponding plug-in, and click install.



Develop and debug applications for Espressif ESP32, ESP32-S2 chips ...

ESP-IDF VS Code Extension

Develop, build, flash, monitor, debug and more with Espressif chips using Espressif IoT Development Framework (ESP-IDF)

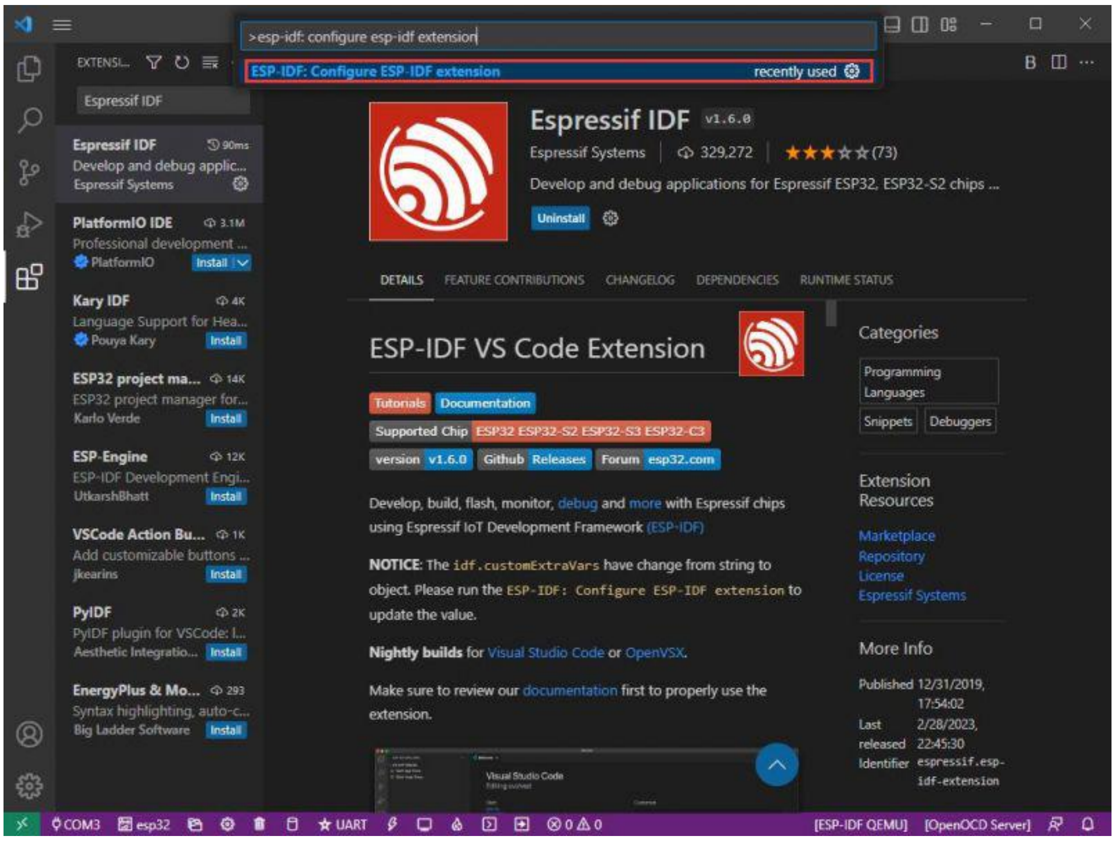
NOTICE: The `idf.customExtraVars` have change from string to object. Please run the `ESP-IDF: Configure ESP-IDF extension` to update the value.

Nightly builds for Visual Studio Code or OpenVSX.

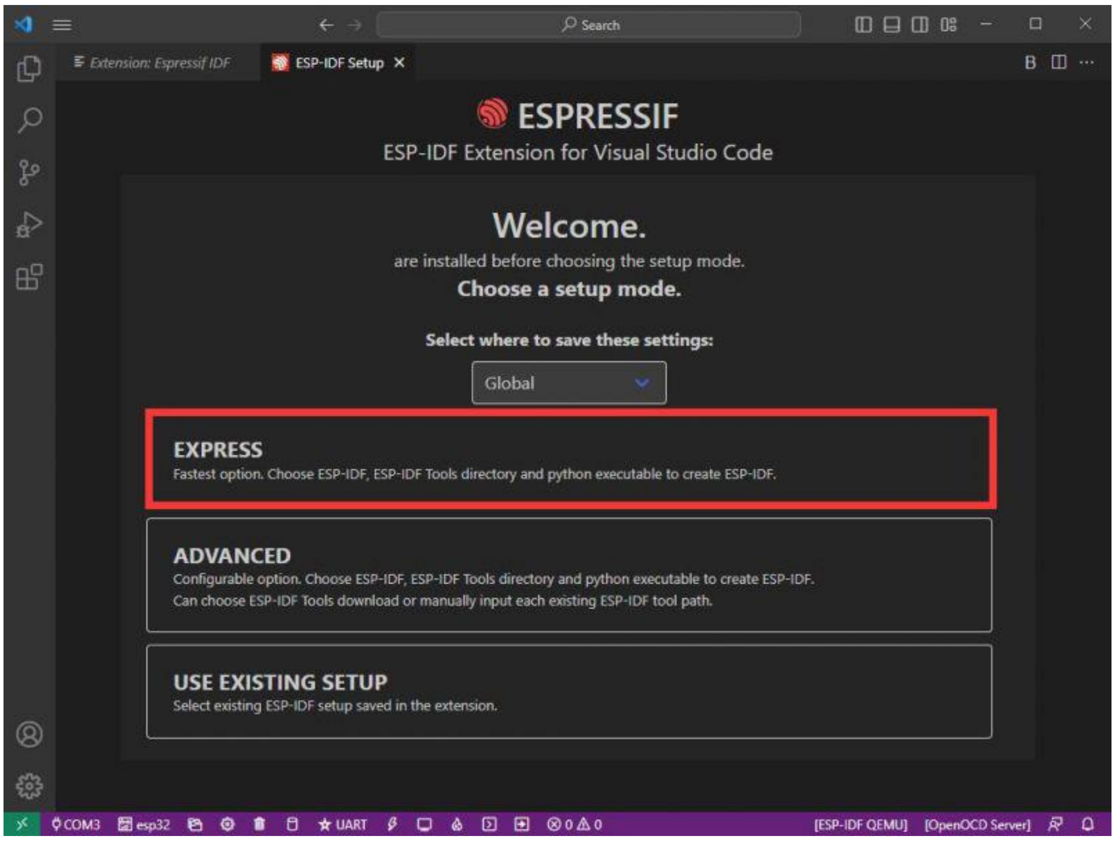
Make sure to review our documentation first to properly use the extension.

3. Press F1 to enter:

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


>esp-idf: configure esp-idf extension
ESP-IDF: Configure ESP-IDF extension
recently used
Espressif IDF v1.6.0
Uninstall
RUNTIME STATUS
Categories
Programming Languages
Snippets
Debuggers
Espressif Systems
Published 12/31/2019, 17:54:02
Last released 2/28/2023, 22:45:30
Identifier espressif.esp-idf-extension
COM3
esp32
UART
[ESP-IDF QEMU]
[OpenOCD Server]

4. Choose express (This tutorial is for first-time users, so only the first general installation tutorial is covered.)



ESPRESSIF

ESP-IDF Extension for Visual Studio Code

Welcome.

are installed before choosing the setup mode.

Choose a setup mode.

Select where to save these settings:

Global

EXPRESS

Fastest option. Choose ESP-IDF, ESP-IDF Tools directory and python executable to create ESP-IDF.

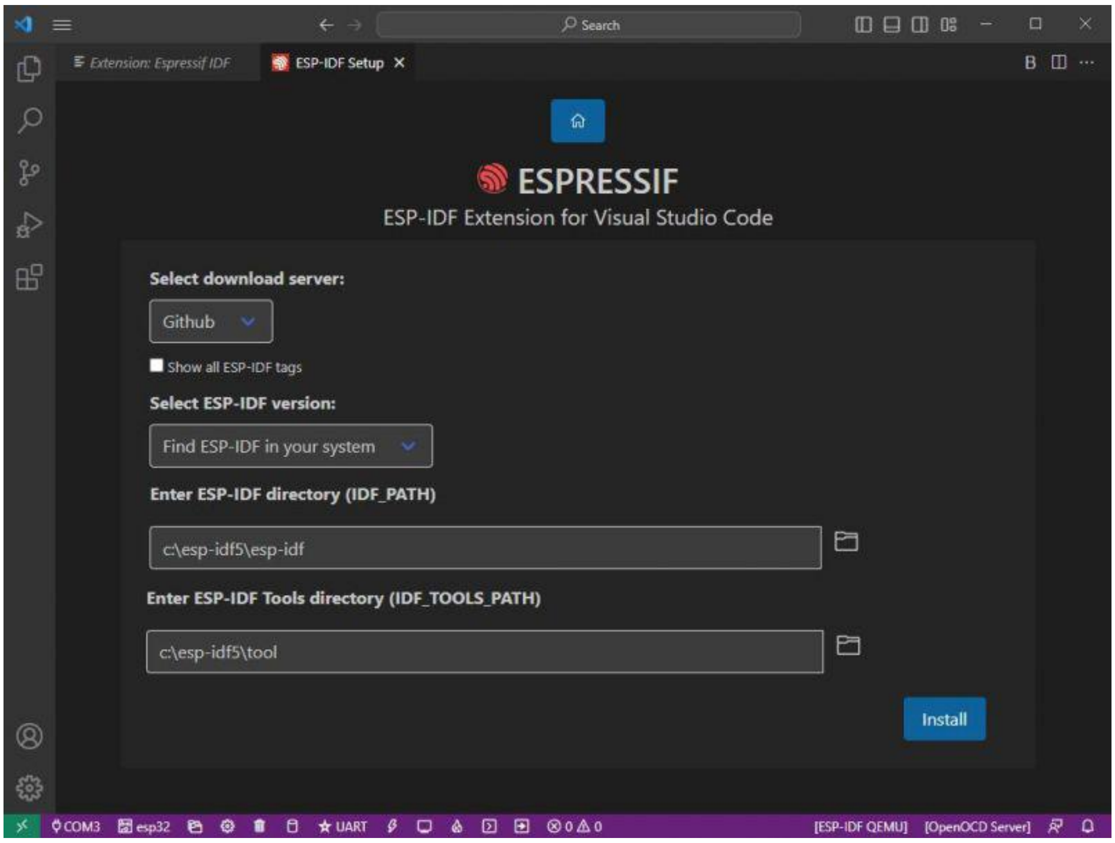
ADVANCED

Configurable option. Choose ESP-IDF, ESP-IDF Tools directory and python executable to create ESP-IDF.
Can choose ESP-IDF Tools download or manually input each existing ESP-IDF tool path.

USE EXISTING SETUP

Select existing ESP-IDF setup saved in the extension.

5. Open and display this screen.



ESPRESSIF

ESP-IDF Extension for Visual Studio Code

Select download server:

Github

Show all ESP-IDF tags

Select ESP-IDF version:

Find ESP-IDF in your system

Enter ESP-IDF directory (IDF_PATH)

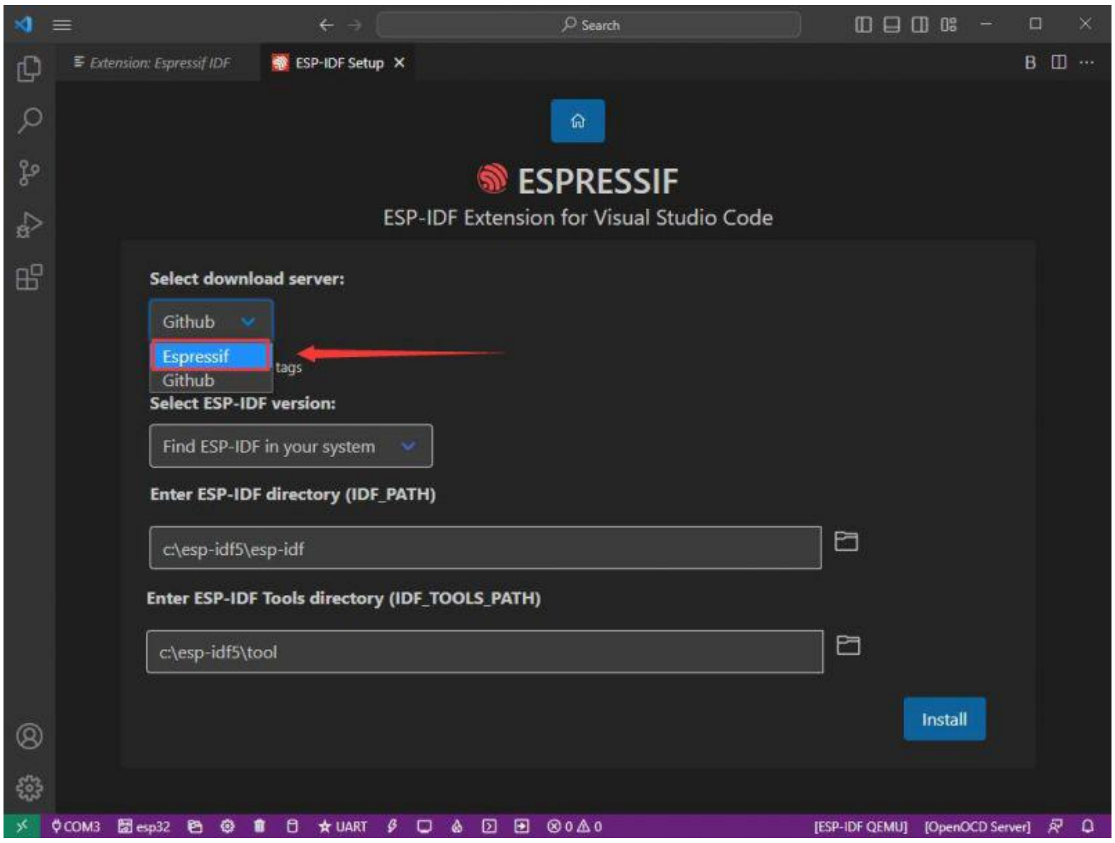
c:\esp-idf5\esp-idf

Enter ESP-IDF Tools directory (IDF_TOOLS_PATH)

c:\esp-idf5\tool

Install

6. Choose a server to download.


Select download server:

Github

Espressif
Github

Select ESP-IDF version:

Find ESP-IDF in your system

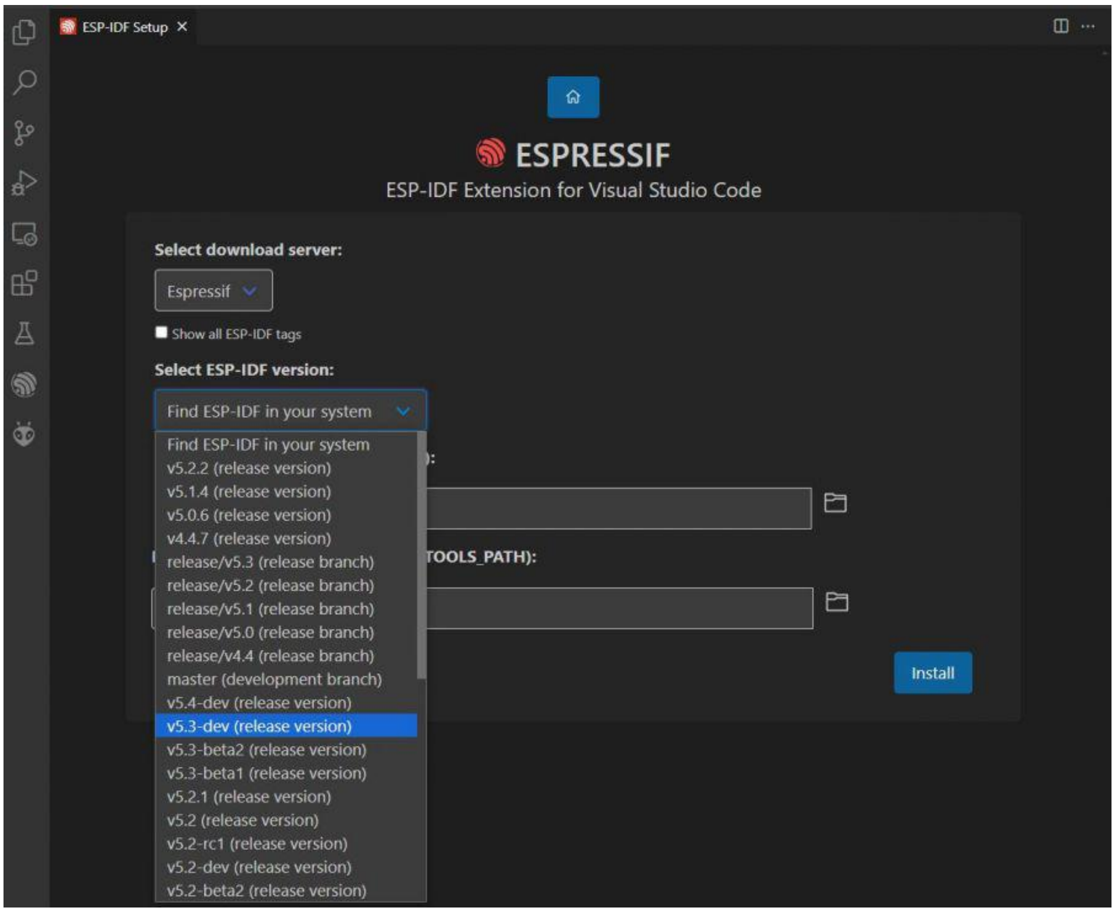
Enter ESP-IDF directory (IDF_PATH)

c:\esp-idf5\esp-idf

Enter ESP-IDF Tools directory (IDF_TOOLS_PATH)

c:\esp-idf5\tool

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).


Select download server:

Espressif

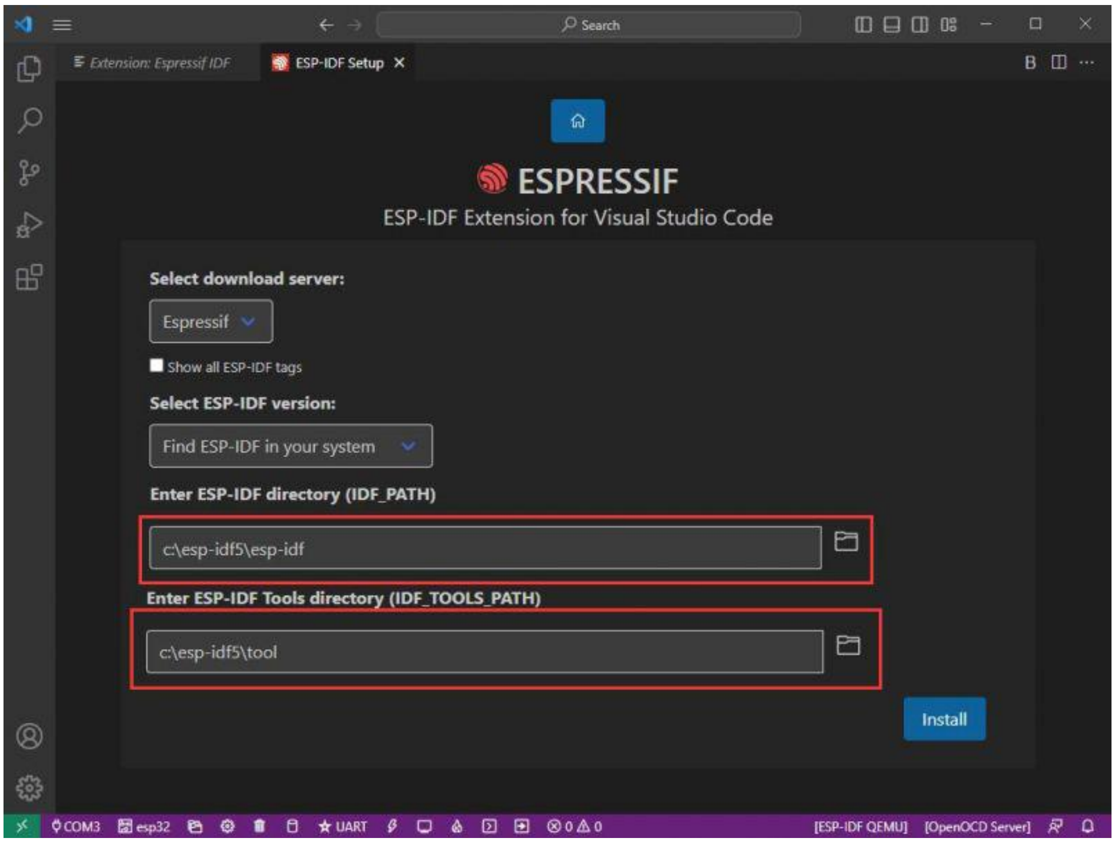
Show all ESP-IDF tags

Select ESP-IDF version:

Find ESP-IDF in your system

Find ESP-IDF in your system
v5.2.2 (release version)
v5.1.4 (release version)
v5.0.6 (release version)
v4.4.7 (release version)
release/v5.3 (release branch)
release/v5.2 (release branch)
release/v5.1 (release branch)
release/v5.0 (release branch)
release/v4.4 (release branch)
master (development branch)
v5.4-dev (release version)
v5.3-dev (release version)
v5.3-beta2 (release version)
v5.3-beta1 (release version)
v5.2.1 (release version)
v5.2 (release version)
v5.2-rc1 (release version)
v5.2-dev (release version)
v5.2-beta2 (release version)

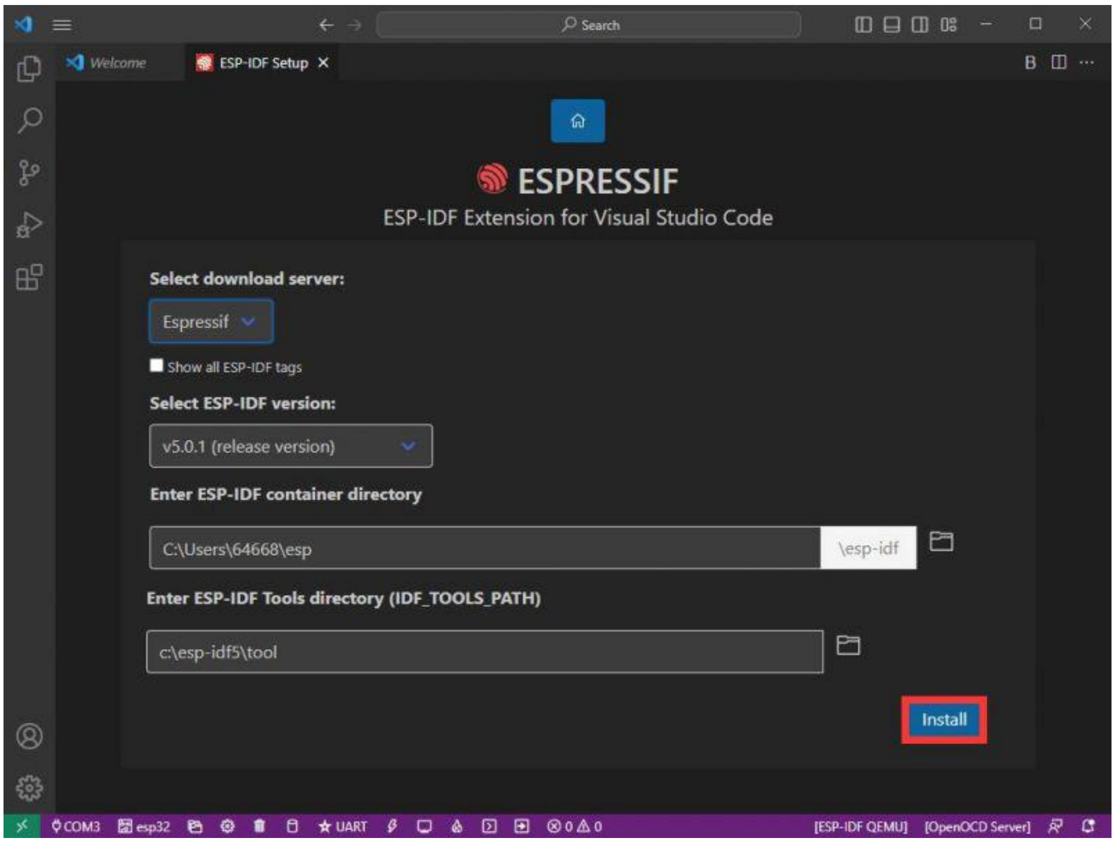
8. The following two are the ESP-IDF directory installation address and the ESP-IDF required tools installation address respectively.



Select download server:
Espressif
Show all ESP-IDF tags
Select ESP-IDF version:
Find ESP-IDF in your system
Enter ESP-IDF directory (IDF_PATH)
c:\esp-idf5\esp-idf
Enter ESP-IDF Tools directory (IDF_TOOLS_PATH)
c:\esp-idf5\tool

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.


ESPRESSIF

ESP-IDF Extension for Visual Studio Code

Select download server: Espressif

Show all ESP-IDF tags

Select ESP-IDF version: v5.0.1 (release version)

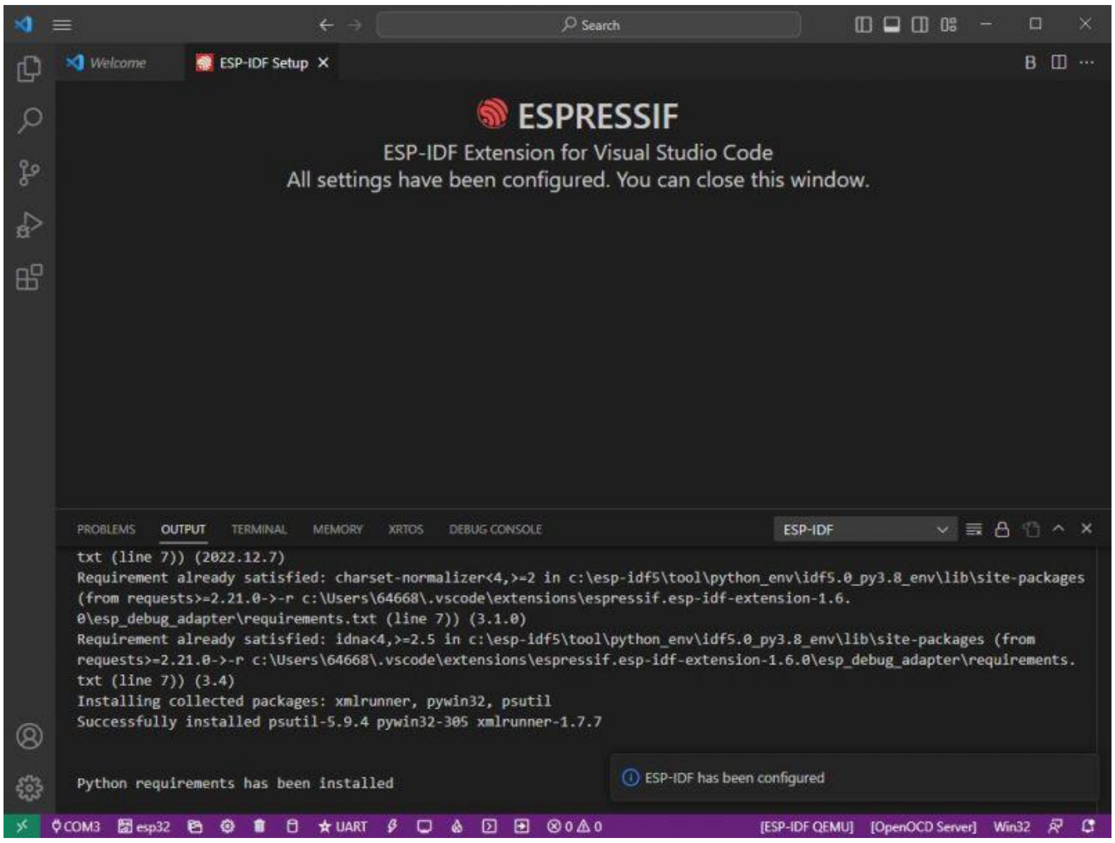
Enter ESP-IDF container directory

C:\Users\64668\esp \esp-idf

Enter ESP-IDF Tools directory (IDF_TOOLS_PATH)

c:\esp-idf5\tool

10. After the installation is completed, it will enter the following screen, indicating that the installation is finished.


ESPRESSIF

ESP-IDF Extension for Visual Studio Code

All settings have been configured. You can close this window.

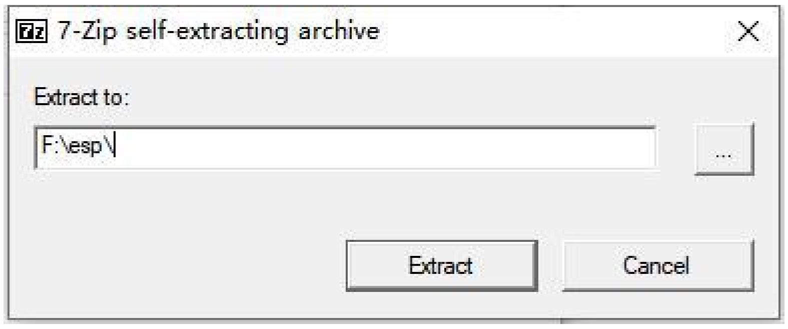
```
txt (line 7)) (2022.12.7)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\esp-idf5\tool\python_env\idf5.0_py3.8_env\lib\site-packages
(from requests>=2.21.0->-r c:\Users\64668\.vscode\extensions\espressif.esp-idf-extension-1.6.
0\esp_debug_adapter\requirements.txt (line 7)) (3.1.0)
Requirement already satisfied: idna<4,>=2.5 in c:\esp-idf5\tool\python_env\idf5.0_py3.8_env\lib\site-packages (from
requests>=2.21.0->-r c:\Users\64668\.vscode\extensions\espressif.esp-idf-extension-1.6.0\esp_debug_adapter\requirements.
txt (line 7)) (3.4)
Installing collected packages: xmlrunner, pywin32, psutil
Successfully installed psutil-5.9.4 pywin32-305 xmlrunner-1.7.7


Python requirements has been installed
```

ESP-IDF has been configured

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:

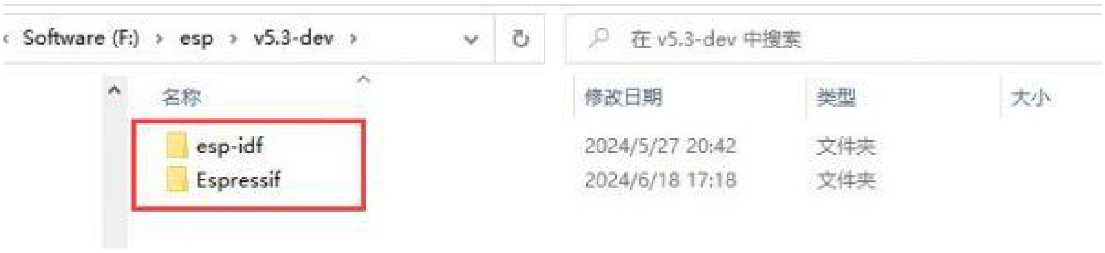


7-Zip self-extracting archive

Extract to:

F:\esp\

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".

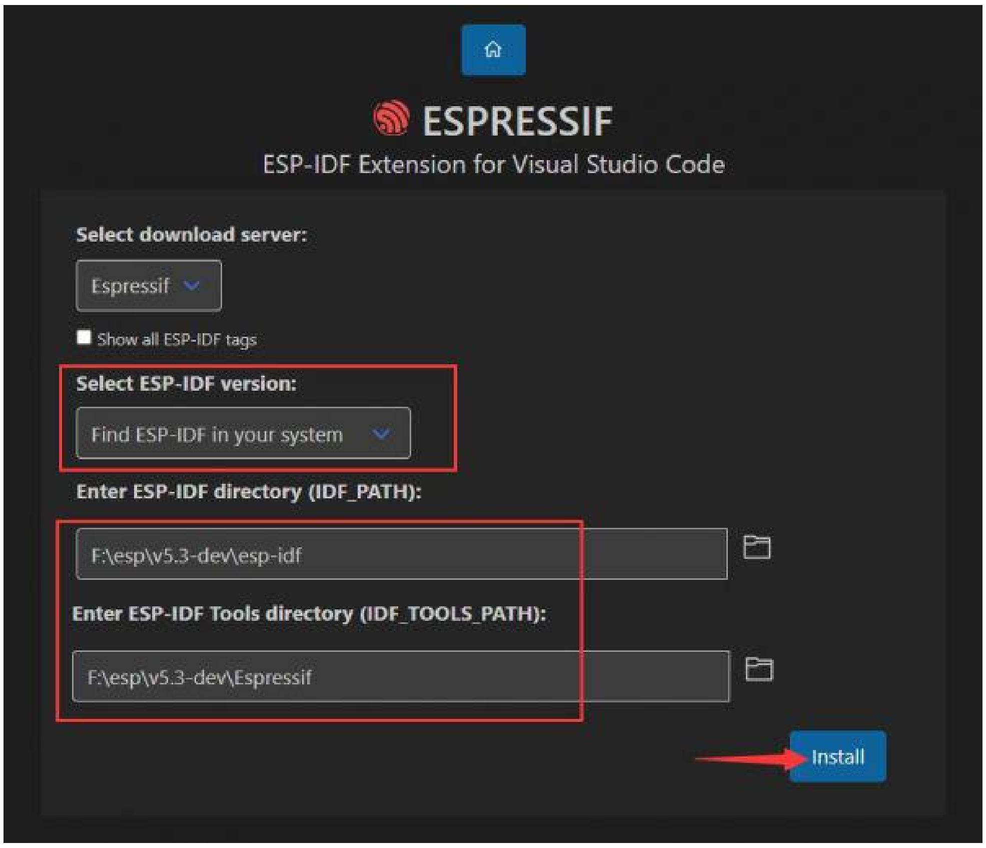


Software (F:) › esp › v5.3-dev ›

在 v5.3-dev 中搜索

名称	修改日期	类型	大小
esp-idf	2024/5/27 20:42	文件夹	
Espressif	2024/6/18 17:18	文件夹	

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":



ESPRESSIF

ESP-IDF Extension for Visual Studio Code

Select download server:

Espressif

Show all ESP-IDF tags

Select ESP-IDF version:

Find ESP-IDF in your system

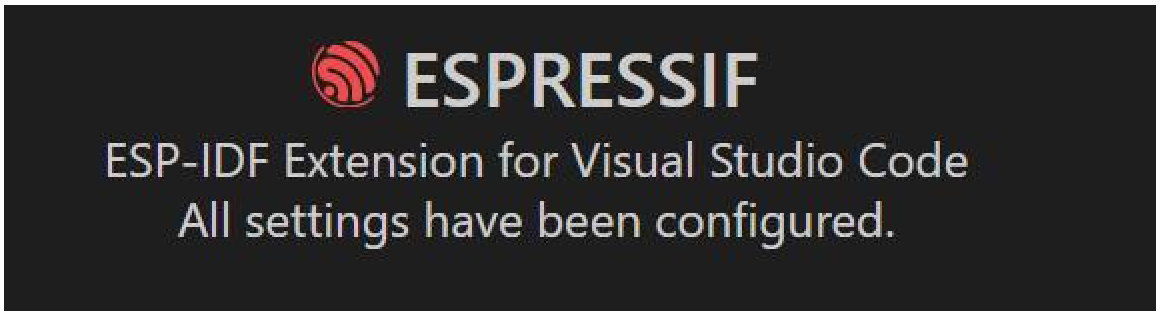
Enter ESP-IDF directory (IDF_PATH):

F:\esp\v5.3-dev\esp-idf

Enter ESP-IDF Tools directory (IDF_TOOLS_PATH):

F:\esp\v5.3-dev\Espressif

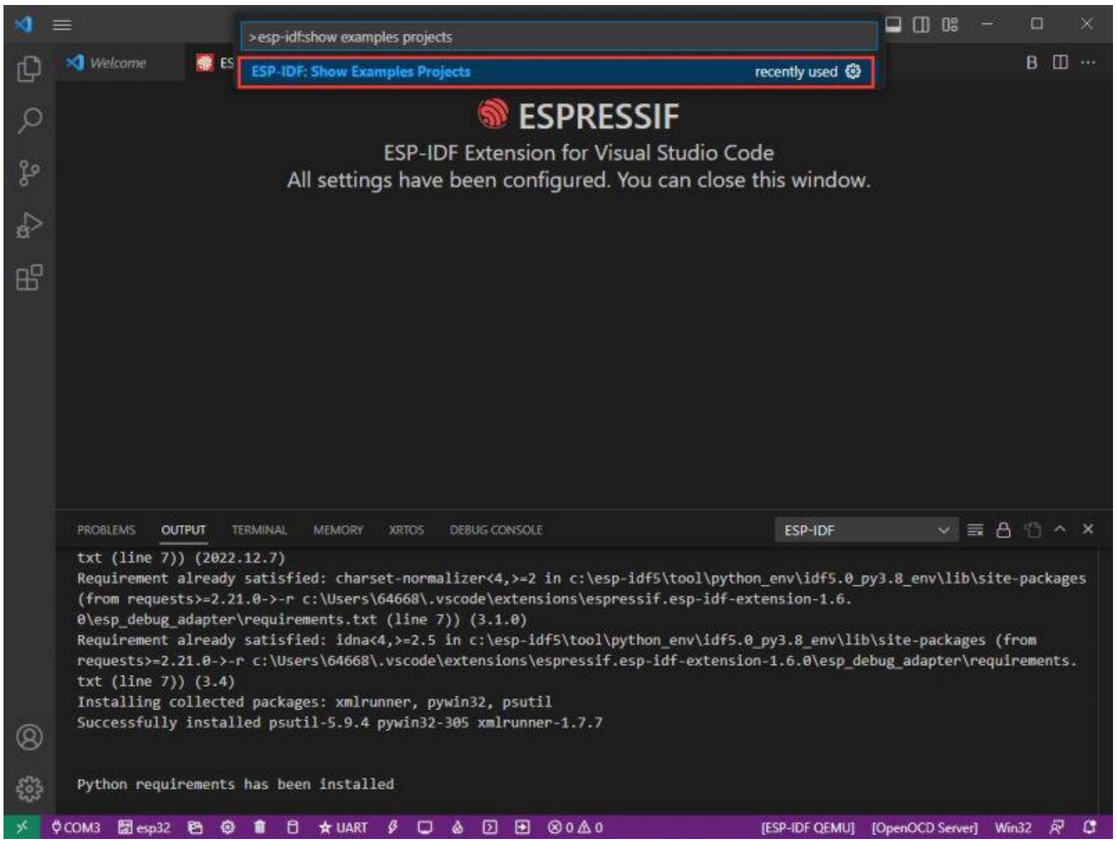
4. It is successfully installed as shown below:


ESPRESSIF
ESP-IDF Extension for Visual Studio Code
All settings have been configured.

Official Demo

Create Demo

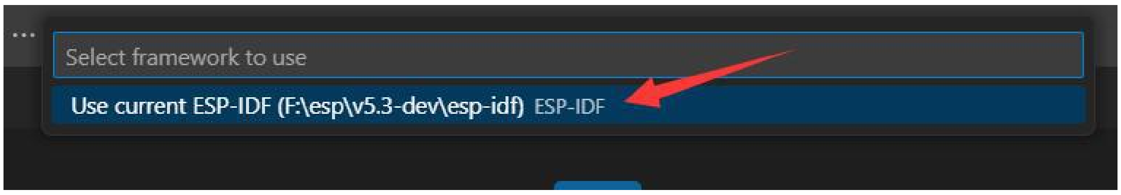
1. Using the shortcut F1, type:
esp-idf:show examples projects



ESPRESSIF
ESP-IDF Extension for Visual Studio Code
All settings have been configured. You can close this window.

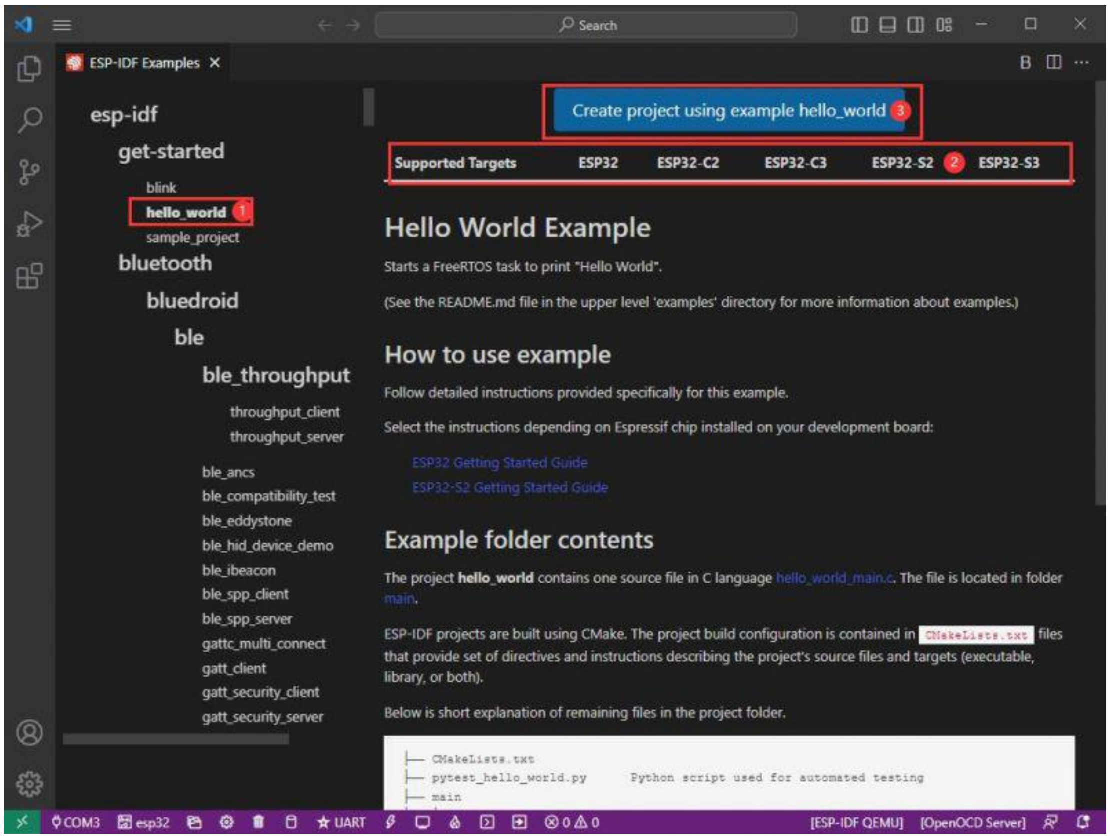
```
txt (line 7)) (2022.12.7)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\esp-idf5\tool\python_env\idf5.0_py3.8_env\lib\site-packages
(from requests>=2.21.0->-r c:\Users\64668\.vscode\extensions\espressif.esp-idf-extension-1.6.
0\esp_debug_adapter\requirements.txt (line 7)) (3.1.0)
Requirement already satisfied: idna<4,>=2.5 in c:\esp-idf5\tool\python_env\idf5.0_py3.8_env\lib\site-packages (from
requests>=2.21.0->-r c:\Users\64668\.vscode\extensions\espressif.esp-idf-extension-1.6.0\esp_debug_adapter\requirements.
txt (line 7)) (3.4)
Installing collected packages: xmlrunner, pywin32, psutil
Successfully installed psutil-5.9.4 pywin32-305 xmlrunner-1.7.7


Python requirements has been installed
```

2. Choose your current IDF version:



Use current ESP-IDF (F:\esp\v5.3-dev\esp-idf) ESP-IDF

3. Take "Hello World" as an example:



Hello World Example

Starts a FreeRTOS task to print "Hello World".

(See the README.md file in the upper level 'examples' directory for more information about examples.)

How to use example

Follow detailed instructions provided specifically for this example.

Select the instructions depending on Espressif chip installed on your development board:

ESP32 Getting Started Guide
ESP32-S2 Getting Started Guide

Example folder contents

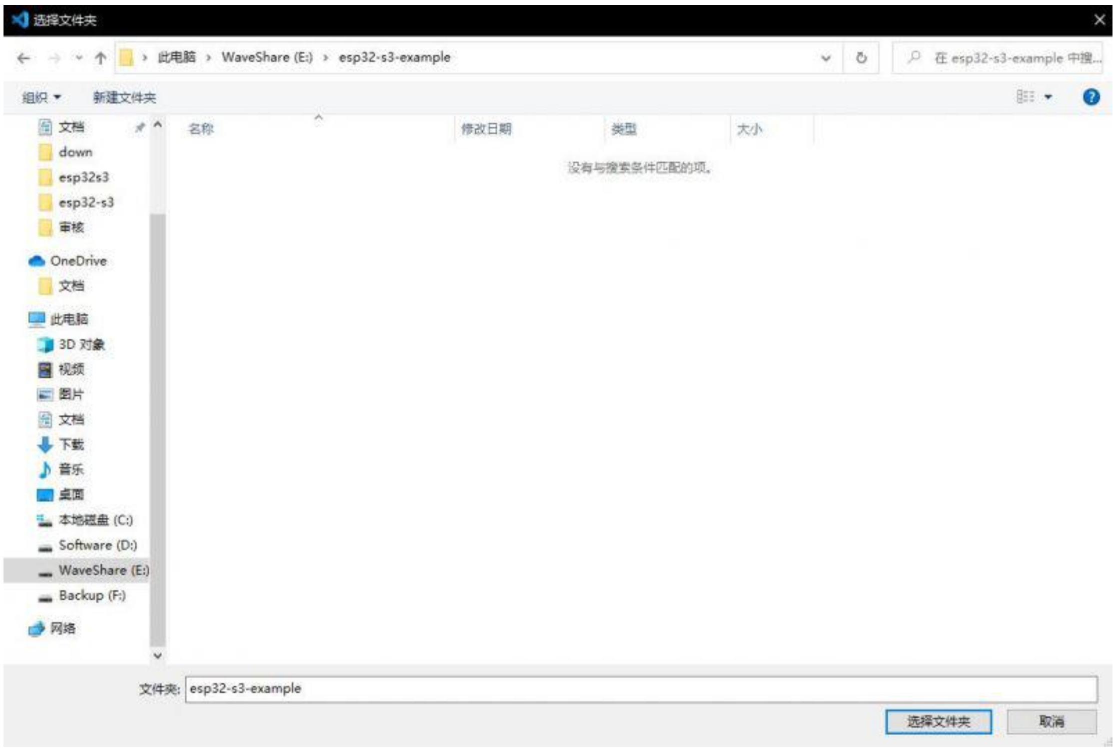
The project **hello_world** contains one source file in C language hello_world_main.c. The file is located in folder main.

ESP-IDF projects are built using CMake. The project build configuration is contained in `CMakeLists.txt` files that provide set of directives and instructions describing the project's source files and targets (executable, library, or both).

Below is short explanation of remaining files in the project folder.

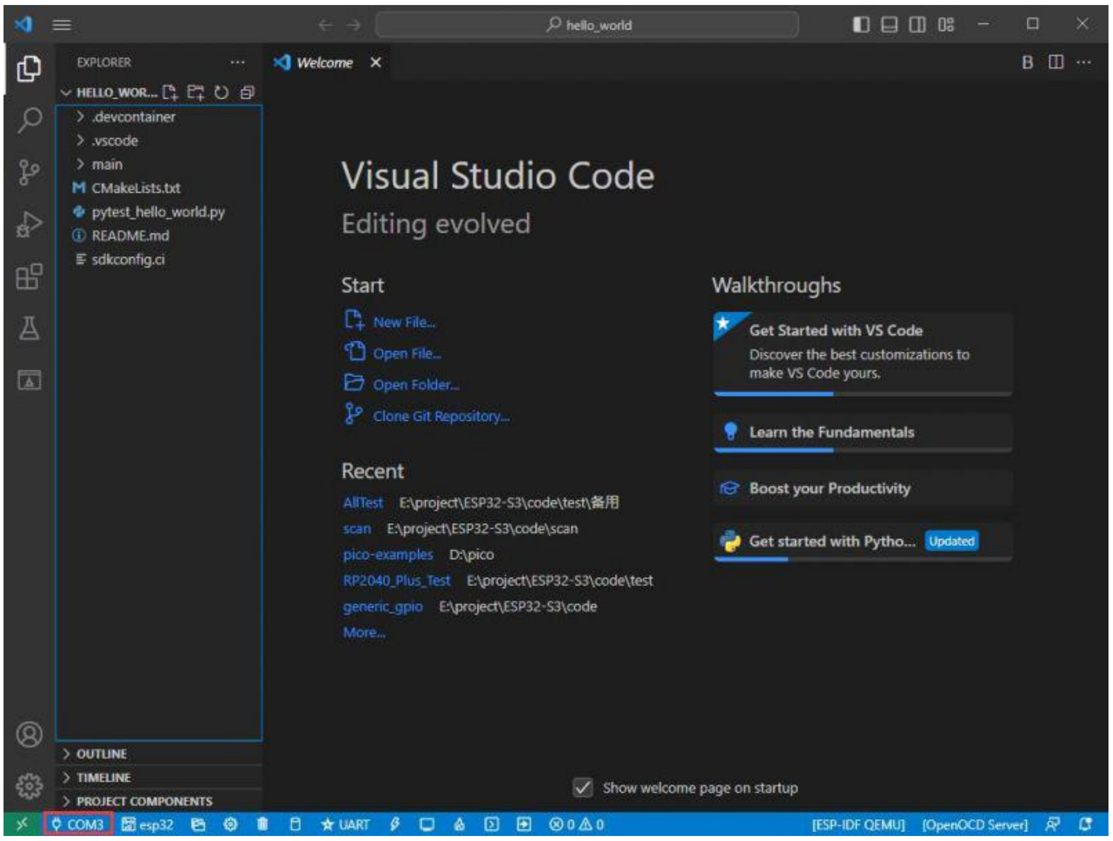
```
├── CMakeLists.txt
├── pytest_hello_world.py      Python script used for automated testing
├── main
```

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.

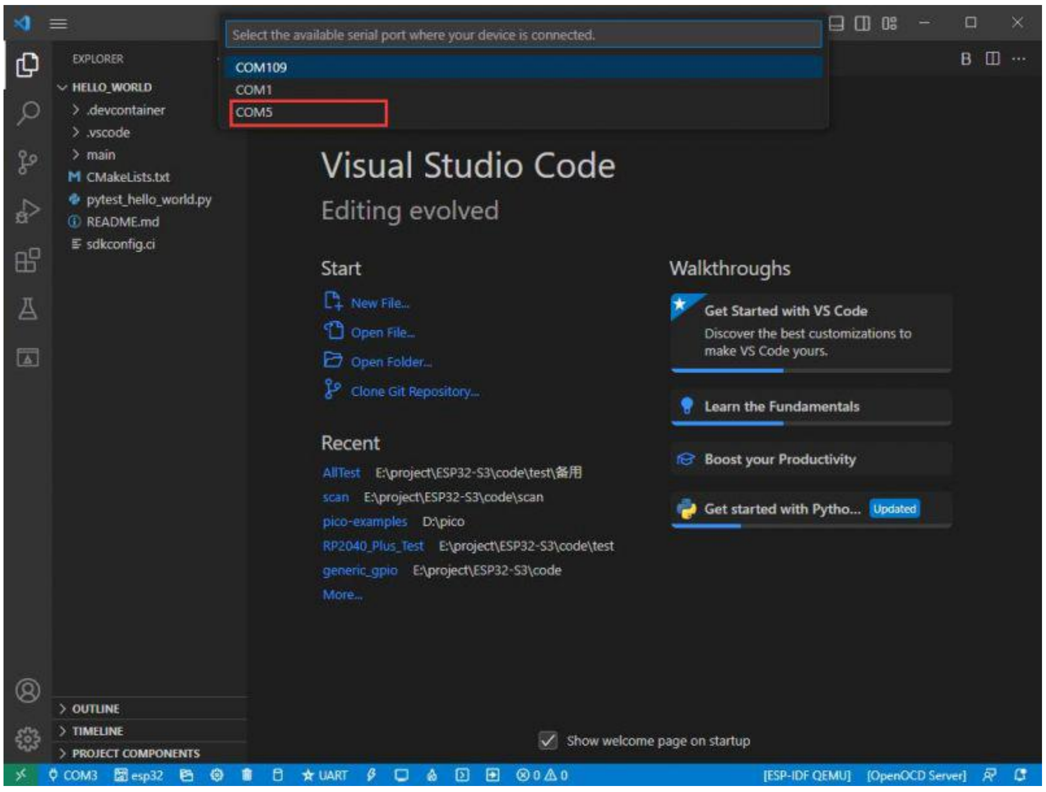


Modify COM Port

1. The corresponding COM port is displayed here, click on it to modify.

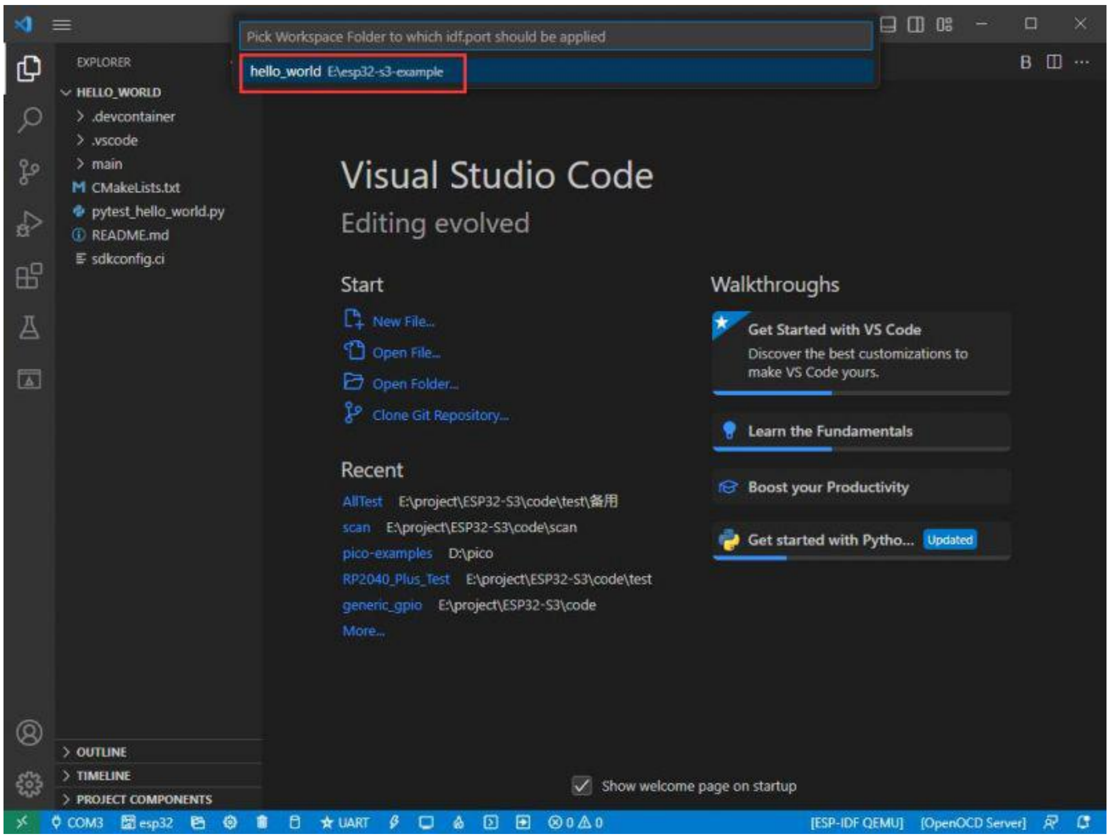


2. We check the device manager COM port, and select COM5, please select your corresponding COM port:


Select the available serial port where your device is connected.

COM109
COM1
COM5

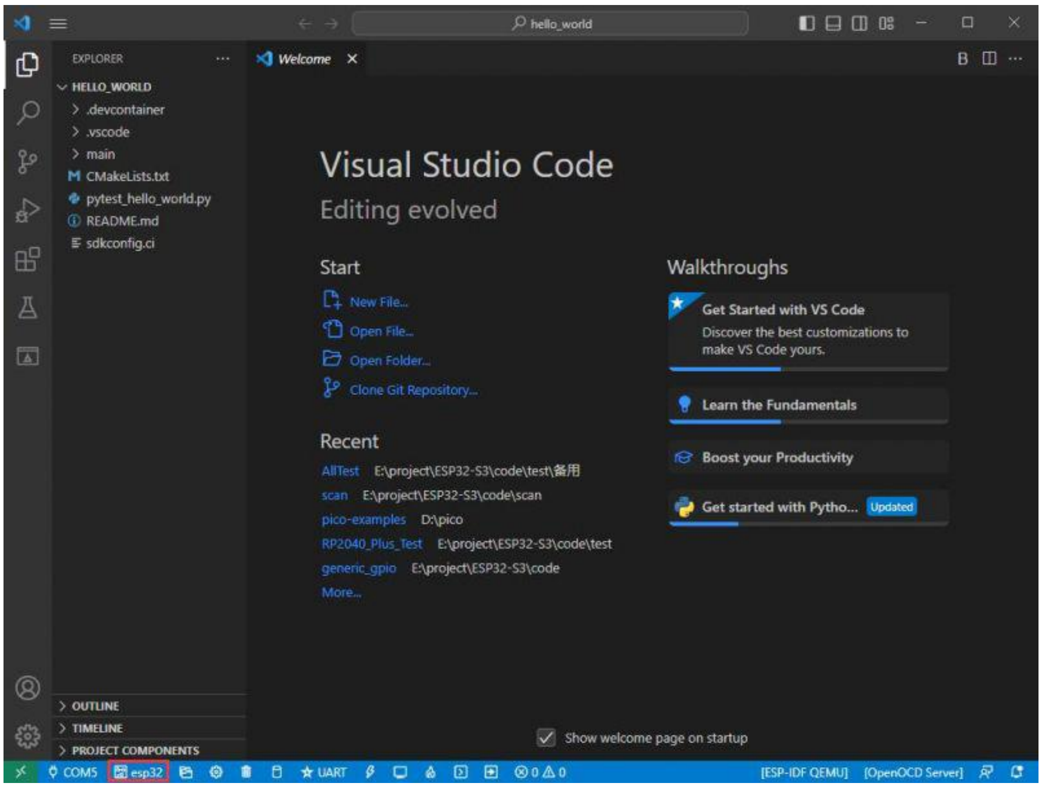
3. Choose the project and demo.


Pick Workspace Folder to which idf.port should be applied
hello_world E:\esp32-s3-example
EXPLORER
HELLO_WORLD
Visual Studio Code
Editing evolved
COM3
esp32

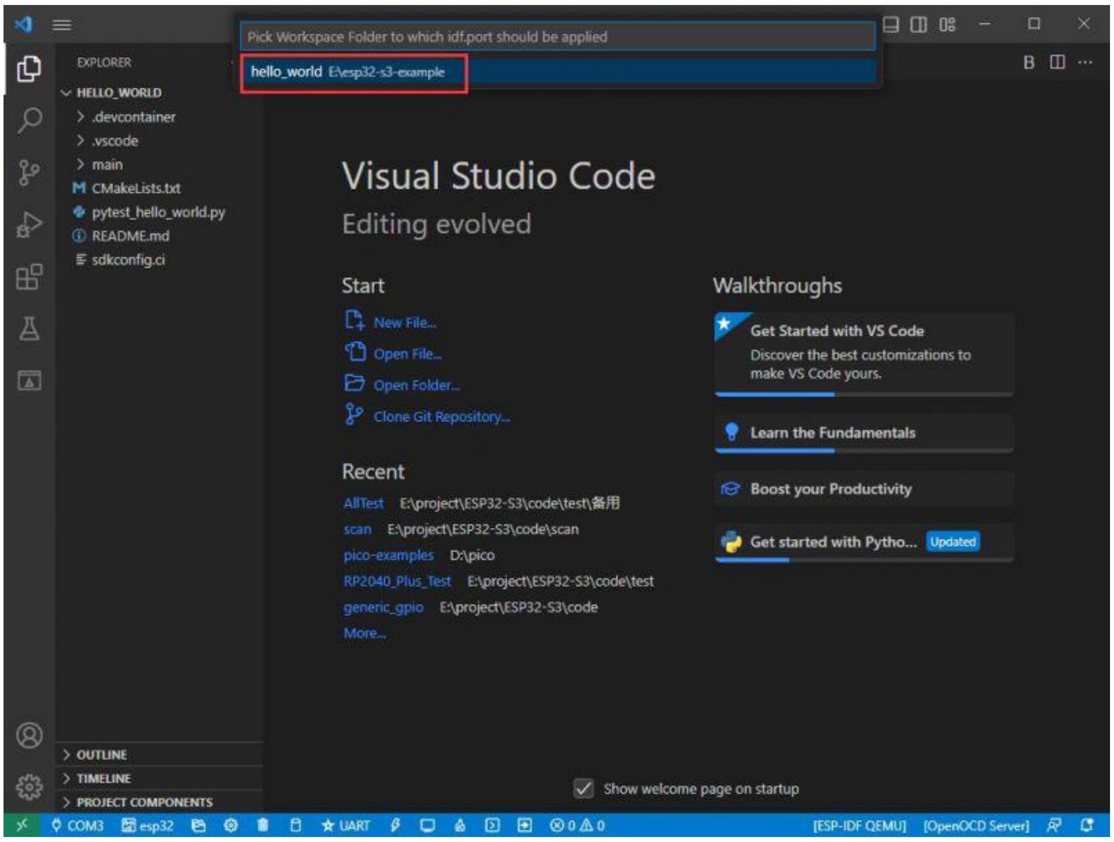
4. Then the COM port is modified.

Modify the Driver

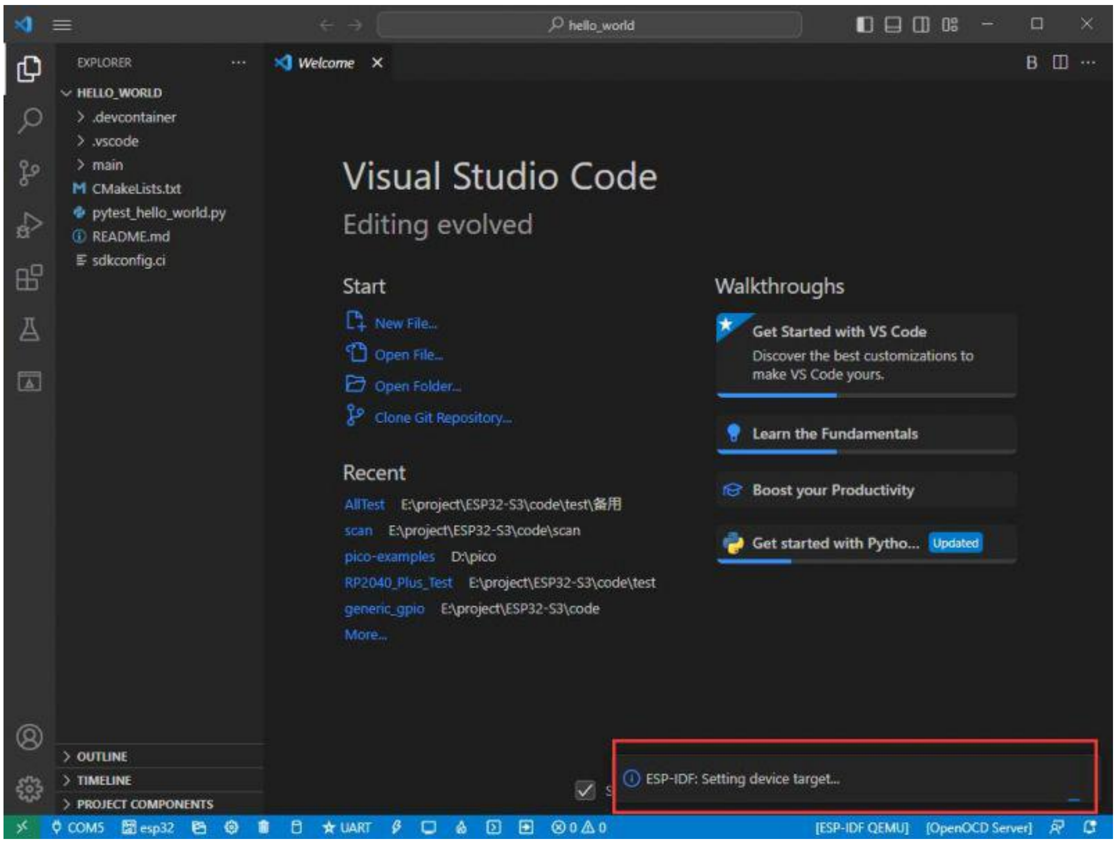
1. Here shows the driver used, click here to modify the corresponding driver:



2. Choose the project or demo:

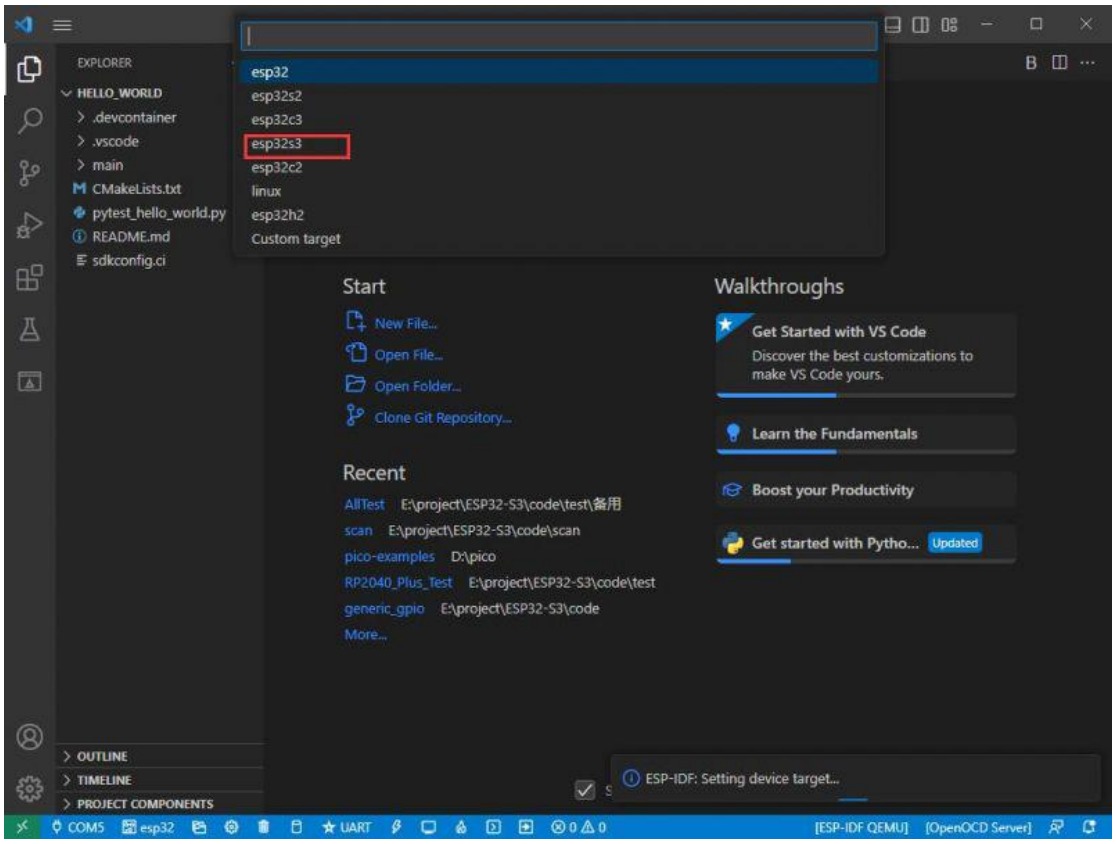


3. Wait for a few seconds after clicking.



ESP-IDF: Setting device target...

4. Choose the driver we need, that is, the main chip ESP32S3.

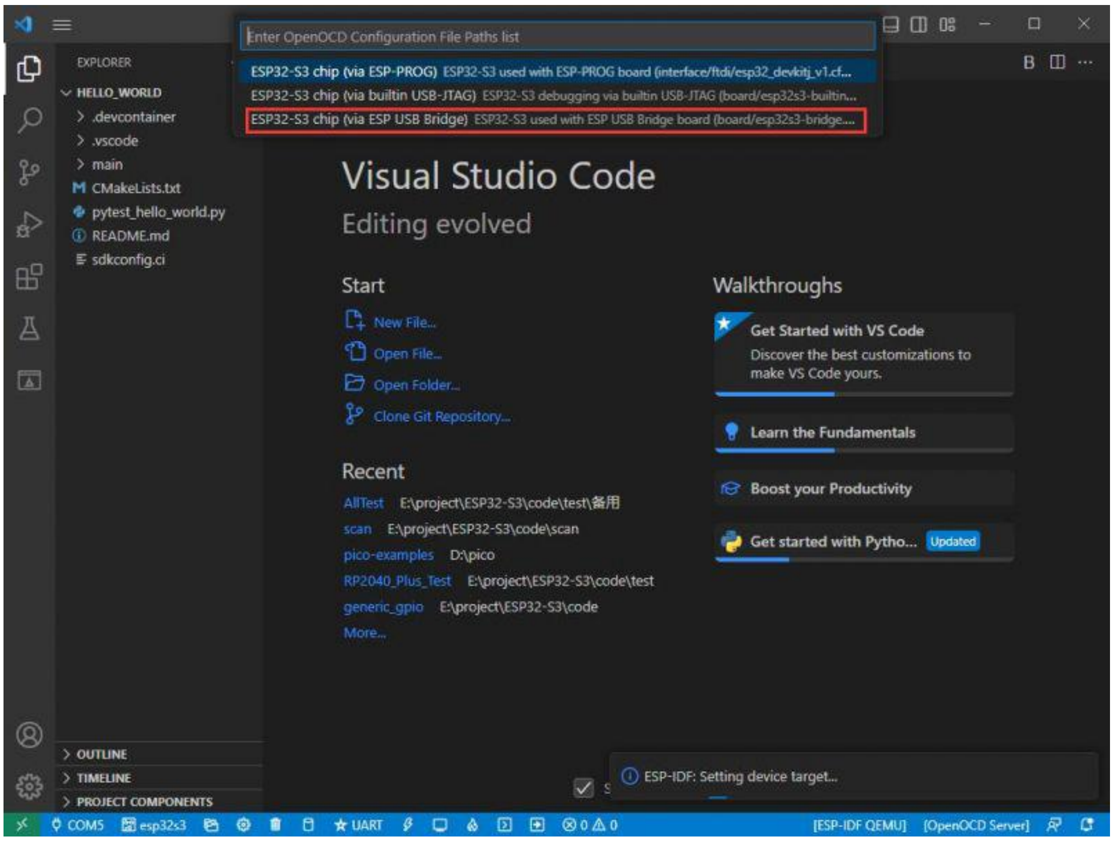


AllTest E:\project\ESP32-S3\code\test\备用
scan E:\project\ESP32-S3\code\scan
pico-examples D:\pico
RP2040_Plus_Test E:\project\ESP32-S3\code\test
generic_gpio E:\project\ESP32-S3\code
More...

Discover the best customizations to make VS Code yours.

ESP-IDF: Setting device target...

5. Choose the openocd path, we can just choose one at random as it doesn't matter.



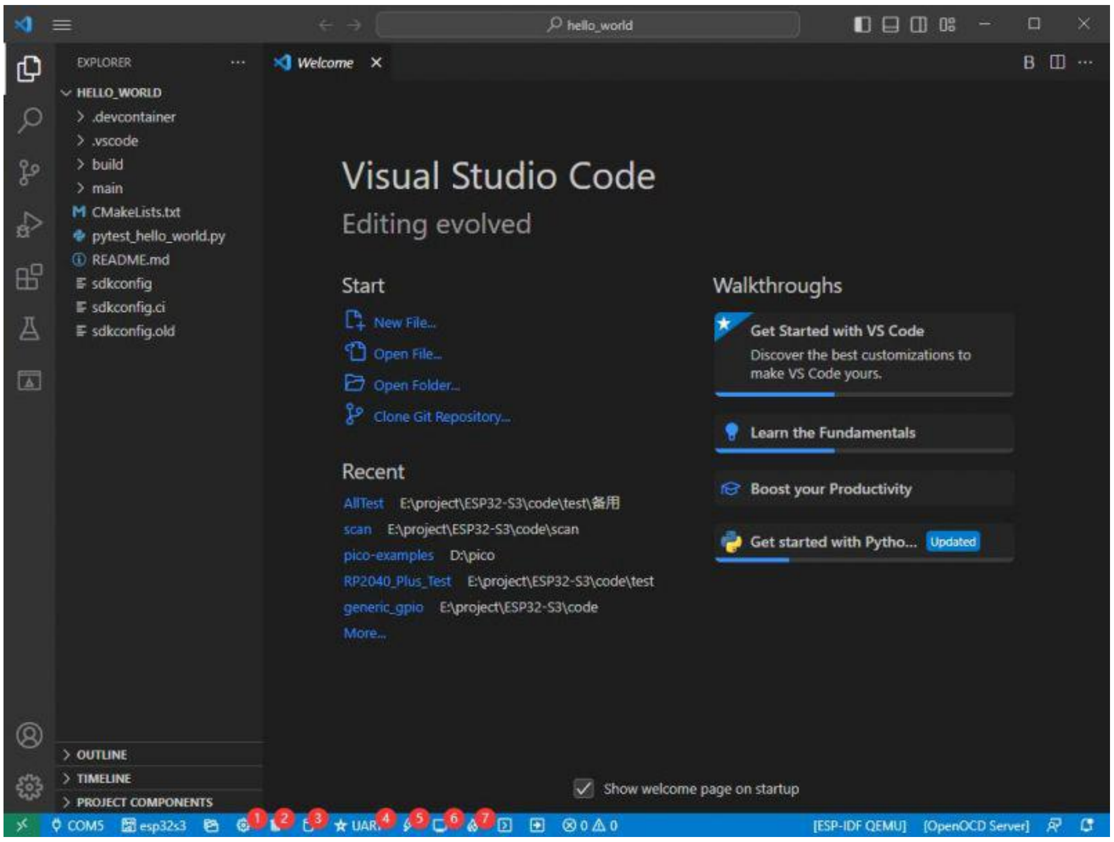
ESP32-S3 chip (via ESP-PROG) ESP32-S3 used with ESP-PROG board (interface/ftdi/esp32_devkitj_v1.cf...
ESP32-S3 chip (via builtin USB-JTAG) ESP32-S3 debugging via builtin USB-JTAG (board/esp32s3-builtin...
ESP32-S3 chip (via ESP USB Bridge) ESP32-S3 used with ESP USB Bridge board (board/esp32s3-bridge...

AllTest E:\project\ESP32-S3\code\test\备用
scan E:\project\ESP32-S3\code\scan
pico-examples D:\pico
RP2040_Plus_Test E:\project\ESP32-S3\code\test
generic_gpio E:\project\ESP32-S3\code
More...

Discover the best customizations to make VS Code yours.

ESP-IDF: Setting device target...

The Rest of the Status Bar Introduction



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.

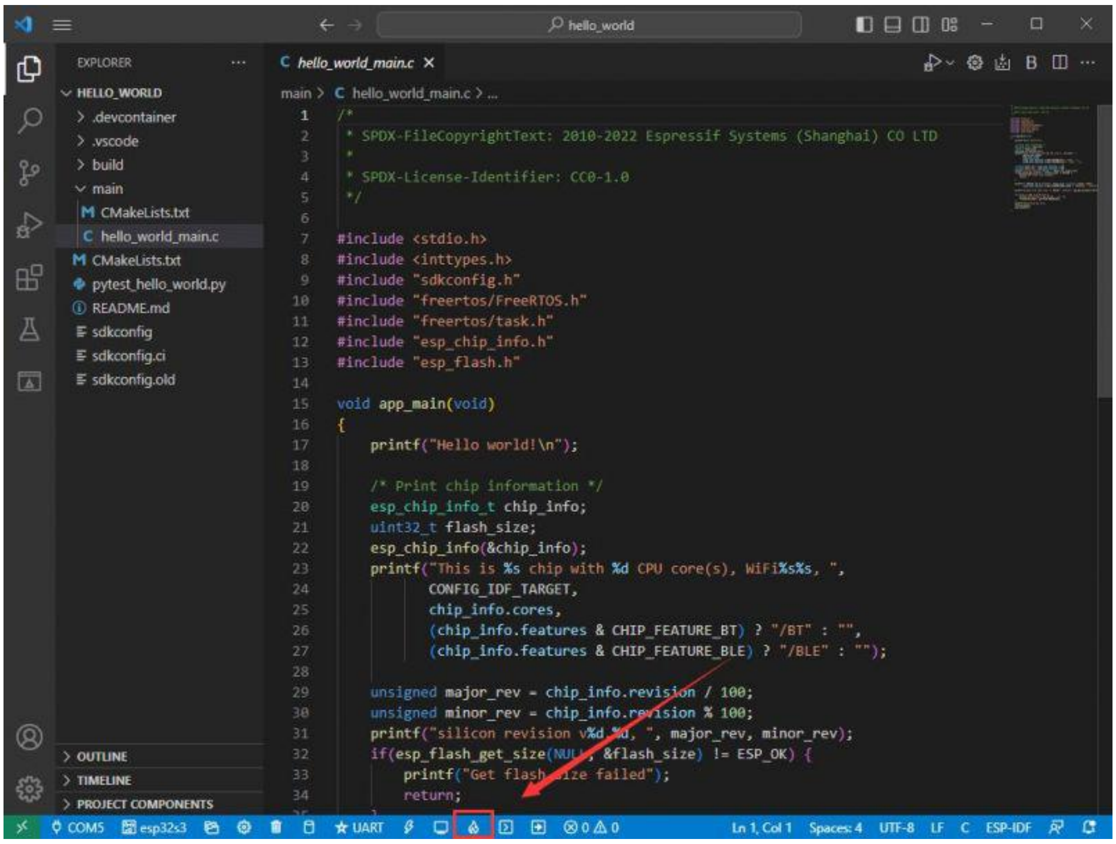
⑤ Program the current firmware, please do it after compiling.

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).

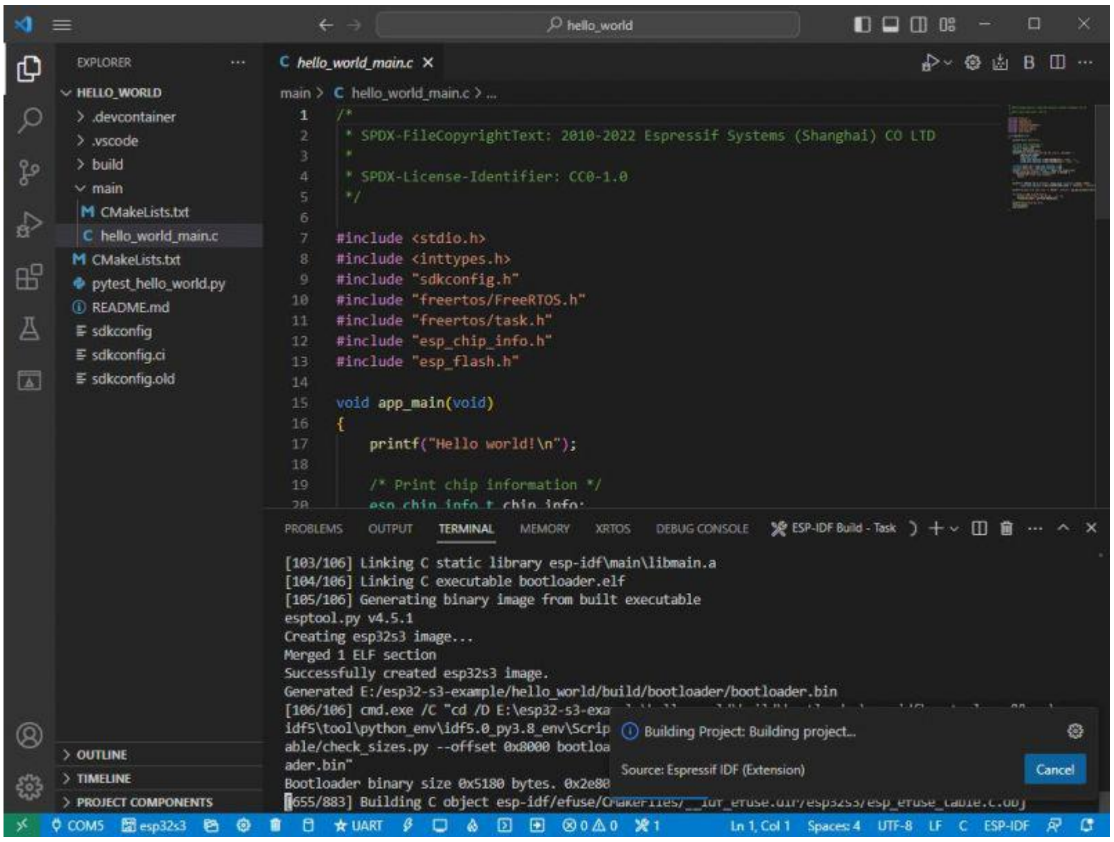
Compile, Program, and Serial Port Monitoring

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



```c
/*
 * SPDX-FileCopyrightText: 2010-2022 Espressif Systems (Shanghai) CO LTD
 *
 * SPDX-License-Identifier: CC0-1.0
 */

#include <stdio.h>
#include <inttypes.h>
#include "sdkconfig.h"
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "esp_chip_info.h"
#include "esp_flash.h"

void app_main(void)
{
    printf("Hello world!\n");

    /* Print chip information */
    esp_chip_info_t chip_info;
    uint32_t flash_size;
    esp_chip_info(&chip_info);
    printf("This is %s chip with %d CPU core(s), WiFi%s%s, ",
           CONFIG_IDF_TARGET,
           chip_info.cores,
           (chip_info.features & CHIP_FEATURE_BT) ? "/BT" : "",
           (chip_info.features & CHIP_FEATURE_BLE) ? "/BLE" : "");

    unsigned major_rev = chip_info.revision / 100;
    unsigned minor_rev = chip_info.revision % 100;
    printf("silicon revision v%d.%d, ", major_rev, minor_rev);
    if(esp_flash_get_size(NULL, &flash_size) != ESP_OK) {
        printf("Get flash size failed");
        return;
```

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



```c
/*
 * SPDX-FileCopyrightText: 2010-2022 Espressif Systems (Shanghai) CO LTD
 *
 * SPDX-License-Identifier: CC0-1.0
 */

#include <stdio.h>
#include <inttypes.h>
#include "sdkconfig.h"
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "esp_chip_info.h"
#include "esp_flash.h"

void app_main(void)
{
    printf("Hello world!\n");

    /* Print chip information */
    esp_chip_info_t chip_info;
```

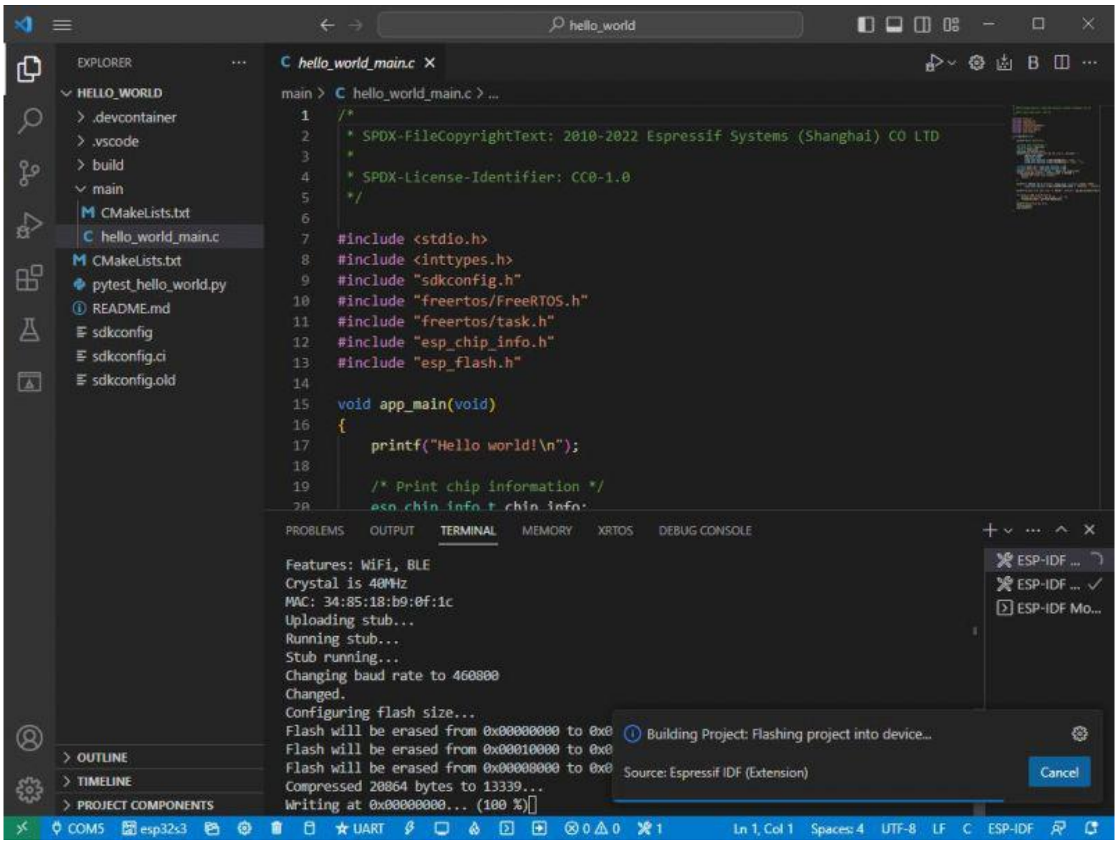
```
[103/106] Linking C static library esp-idf\main\libmain.a
[104/106] Linking C executable bootloader.elf
[105/106] Generating binary image from built executable
esptool.py v4.5.1
Creating esp32s3 image...
Merged 1 ELF section
Successfully created esp32s3 image.
Generated E:/esp32-s3-example/hello_world/build/bootloader/bootloader.bin
[106/106] cmd.exe /C "cd /D E:\esp32-s3-exa
idf5\tool\python_env\idf5.0_py3.8_env\Scrip
able/check_sizes.py --offset 0x8000 bootloa
ader.bin"
Bootloader binary size 0x5180 bytes. 0x2e80
[655/883] Building C object esp-idf/efuse/CMakeFiles/__idf_efuse.dir/esp32s3/esp_efuse_table.c.obj
```

Building Project: Building project...

Source: Espressif IDF (Extension)

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.



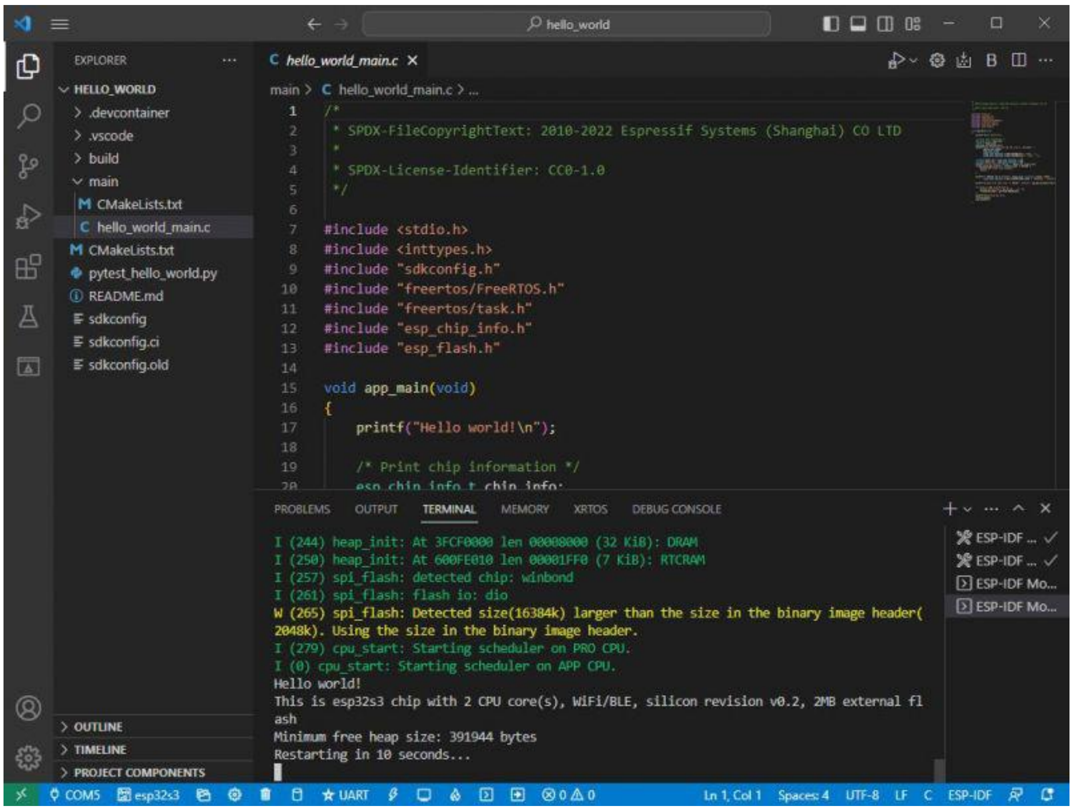
```c
/*
 * SPDX-FileCopyrightText: 2010-2022 Espressif Systems (Shanghai) CO LTD
 *
 * SPDX-License-Identifier: CC0-1.0
 */

#include <stdio.h>
#include <inttypes.h>
#include "sdkconfig.h"
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "esp_chip_info.h"
#include "esp_flash.h"

void app_main(void)
{
    printf("Hello world!\n");

    /* Print chip information */
    esp_chip_info_t chip_info;
```

```
Features: WiFi, BLE
Crystal is 40MHz
MAC: 34:85:18:b9:0f:1c
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
Configuring flash size...
Flash will be erased from 0x00000000 to 0x0
Flash will be erased from 0x00010000 to 0x0
Flash will be erased from 0x00008000 to 0x0
Compressed 20864 bytes to 13339...
Writing at 0x00000000... (100 %)
```

Building Project: Flashing project into device...

Source: Espressif IDF (Extension)

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.

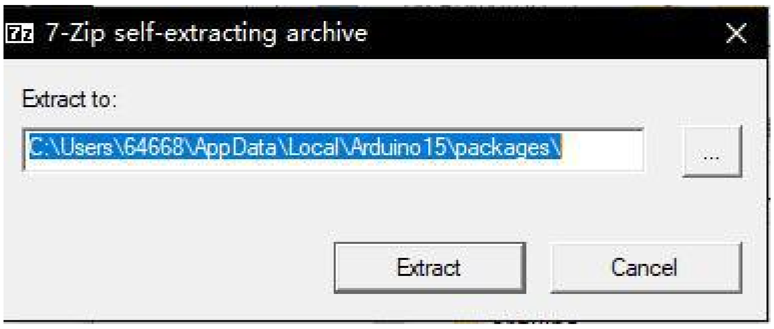


```c
/*
 * SPDX-FileCopyrightText: 2010-2022 Espressif Systems (Shanghai) CO LTD
 *
 * SPDX-License-Identifier: CC0-1.0
 */

#include <stdio.h>
#include <inttypes.h>
#include "sdkconfig.h"
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "esp_chip_info.h"
#include "esp_flash.h"

void app_main(void)
{
    printf("Hello world!\n");

    /* Print chip information */
    esp_chip_info_t chip_info;
```

```
I (244) heap_init: At 3FCF0000 len 00008000 (32 KiB): DRAM
I (250) heap_init: At 600FE010 len 00001FF0 (7 KiB): RTCRAM
I (257) spi_flash: detected chip: winbond
I (261) spi_flash: flash io: dio
W (265) spi_flash: Detected size(16384k) larger than the size in the binary image header(
2048k). Using the size in the binary image header.
I (279) cpu_start: Starting scheduler on PRO CPU.
I (0) cpu_start: Starting scheduler on APP CPU.
Hello world!
This is esp32s3 chip with 2 CPU core(s), WiFi/BLE, silicon revision v0.2, 2MB external fl
ash
Minimum free heap size: 391944 bytes
Restarting in 10 seconds...
```

Arduino

Download and install Arduino IDE.
Install ESP32 on the Arduino IDE as shown below, and you can refer to this link.
Download it from this link.



C:\Users\64668\AppData\Local\Arduino15\packages\

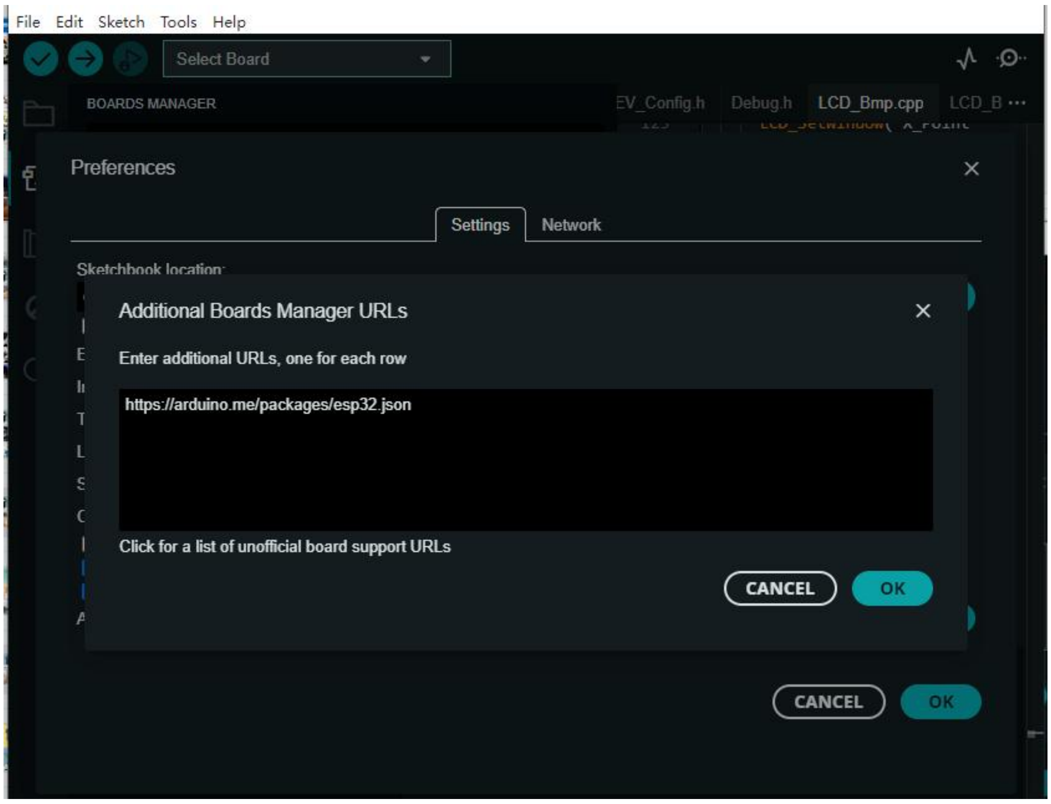
```
C:\Users\{username}\AppData\Local\Arduino15\packages\
```

Take username "waveshare" as an example:

```
C:\Users\waveshare\AppData\Local\Arduino15\packages\
```

After installation, open Arduino IDE, open File -> Preferences -> Seetting, input the following link at Additional boards manager URLs and save it:

```
https://arduino.me/packages/esp32.json
```



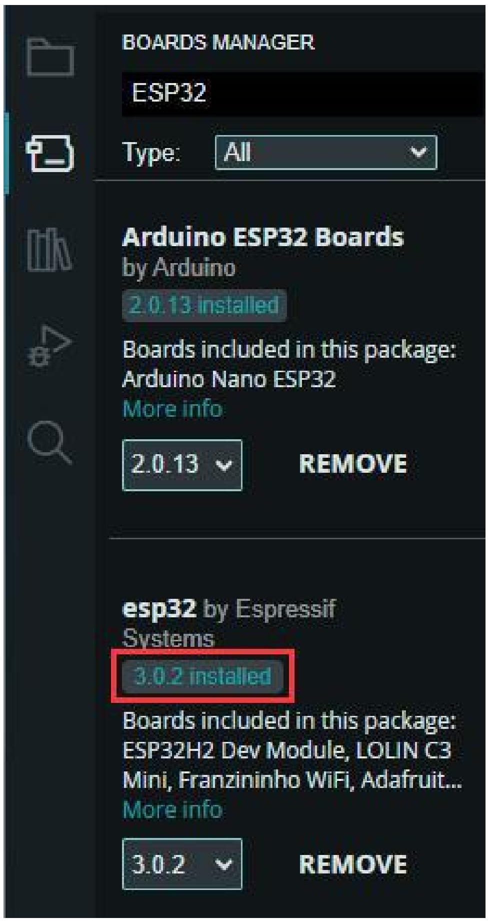
Additional Boards Manager URLs

Enter additional URLs, one for each row

https://arduino.me/packages/esp32.json

Click for a list of unofficial board support URLs

Search esp32 on Board Manager to install, if 3.0.2 is installed, the offline package is installed.



ESP32

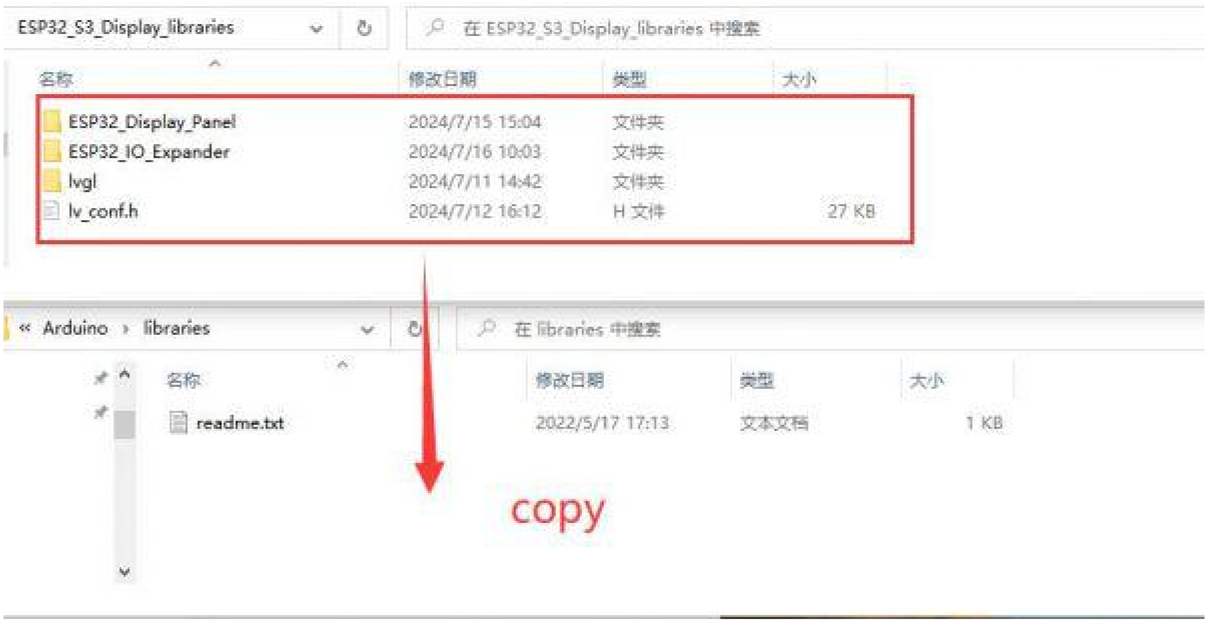
Type: All

Arduino ESP32 Boards
by Arduino
2.0.13 installed
Boards included in this package:
Arduino Nano ESP32
More info
2.0.13 REMOVE

esp32 by Espressif Systems
3.0.2 installed
Boards included in this package:
ESP32H2 Dev Module, LOLIN C3 Mini, Franzininho WiFi, Adafruit...
More info
3.0.2 REMOVE

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_Display_libraries to "C:\Users\xxxx\Documents\Arduino\libraries". Please note that "xxxx" represents your computer username.

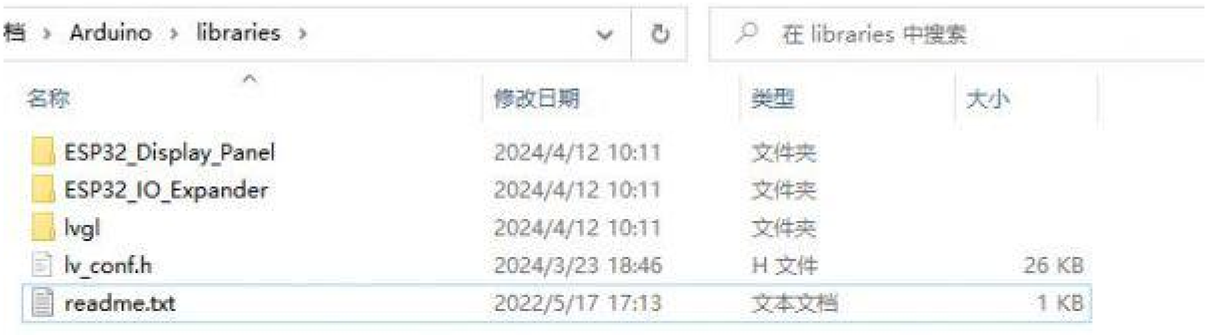

ESP32_S3_Display_libraries — 在 ESP32_S3_Display_libraries 中搜索

名称	修改日期	类型	大小
ESP32_Display_Panel	2024/7/15 15:04	文件夹	
ESP32_IO_Expander	2024/7/16 10:03	文件夹	
lvgl	2024/7/11 14:42	文件夹	
lv_conf.h	2024/7/12 16:12	H 文件	27 KB

« Arduino › libraries — 在 libraries 中搜索

名称	修改日期	类型	大小
readme.txt	2022/5/17 17:13	文本文档	1 KB

After copying:


档 › Arduino › libraries › — 在 libraries 中搜索

名称	修改日期	类型	大小
ESP32_Display_Panel	2024/4/12 10:11	文件夹	
ESP32_IO_Expander	2024/4/12 10:11	文件夹	
lvgl	2024/4/12 10:11	文件夹	
lv_conf.h	2024/3/23 18:46	H 文件	26 KB
readme.txt	2022/5/17 17:13	文本文档	1 KB

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
```

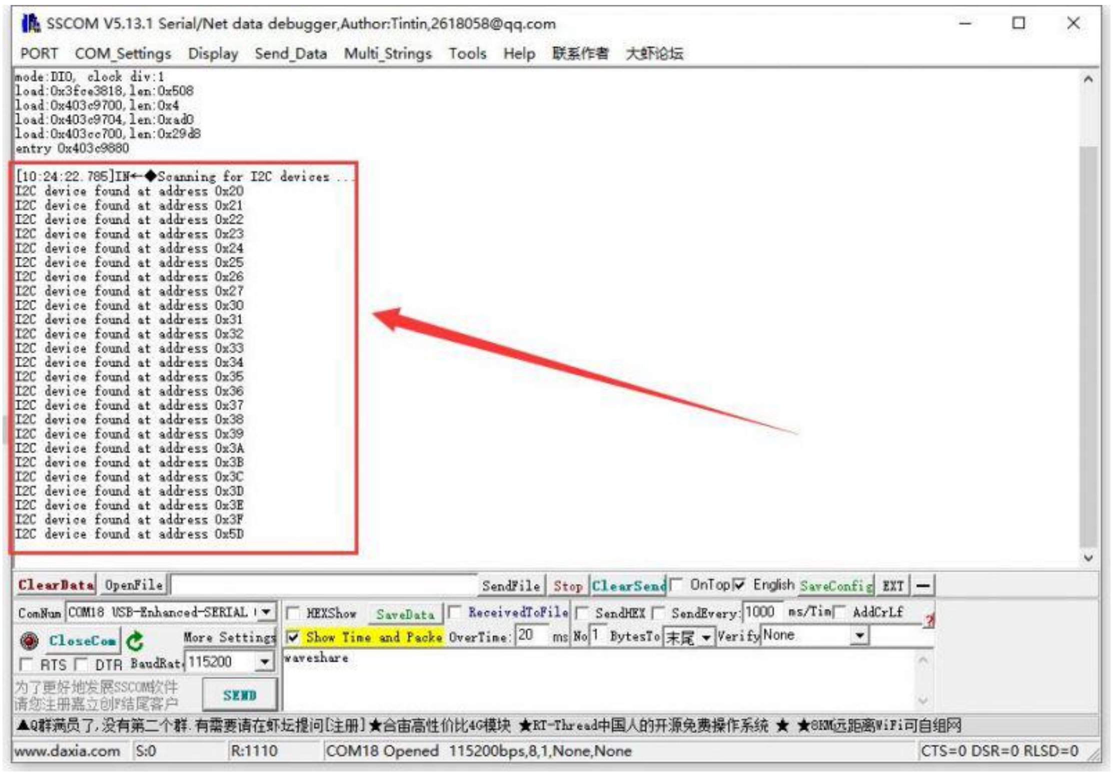
Before using the Arduino example, the software library should be set as shown below:

1. Using the library file we provided.
2. Install v3.0.2-h for Arduino esp32.
3. 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.


SSCOM V5.13.1 Serial/Net data debugger,Author:Tintin,2618058@qq.com
PORT COM_Settings Display Send_Data Multi_Strings Tools Help 联系作者 大虾论坛
ClearData OpenFile SendFile Stop ClearSend OnTop English SaveConfig EXT
ComNum COM18 USB-Enhanced-SERIAL HEXShow SaveData ReceivedToFile SendHEX SendEvery: 1000 ms/Tim AddCrLf
CloseCom More Settings Show Time and Packe OverTime: 20 ms No 1 BytesTo 末尾 Verify None
RTS DTR BaudRat 115200
为了更好地发展SSCOM软件 请您注册嘉立创F结尾客户
SEND
▲Q群满员了,没有第二个群.有需要请在虾坛提问[注册]★合宙高性价比4G模块 ★RT-Thread中国人的开源免费操作系统 ★ ★8KM远距离WiFi可自组网
www.daxia.com S:0 R:1110 COM18 Opened 115200bps,8,1,None,None CTS=0 DSR=0 RLSD=0

```
mode:DIO, clock div:1
load:0x3fce3818,len:0x508
load:0x403c9700,len:0x4
load:0x403c9704,len:0xad0
load:0x403cc700,len:0x29d8
entry 0x403c9880

[10:24:22.785]IN←◆Scanning for I2C devices ...
I2C device found at address 0x20
I2C device found at address 0x21
I2C device found at address 0x22
I2C device found at address 0x23
I2C device found at address 0x24
I2C device found at address 0x25
I2C device found at address 0x26
I2C device found at address 0x27
I2C device found at address 0x30
I2C device found at address 0x31
I2C device found at address 0x32
I2C device found at address 0x33
I2C device found at address 0x34
I2C device found at address 0x35
I2C device found at address 0x36
I2C device found at address 0x37
I2C device found at address 0x38
I2C device found at address 0x39
I2C device found at address 0x3A
I2C device found at address 0x3B
I2C device found at address 0x3C
I2C device found at address 0x3D
I2C device found at address 0x3E
I2C device found at address 0x3F
I2C device found at address 0x5D
```

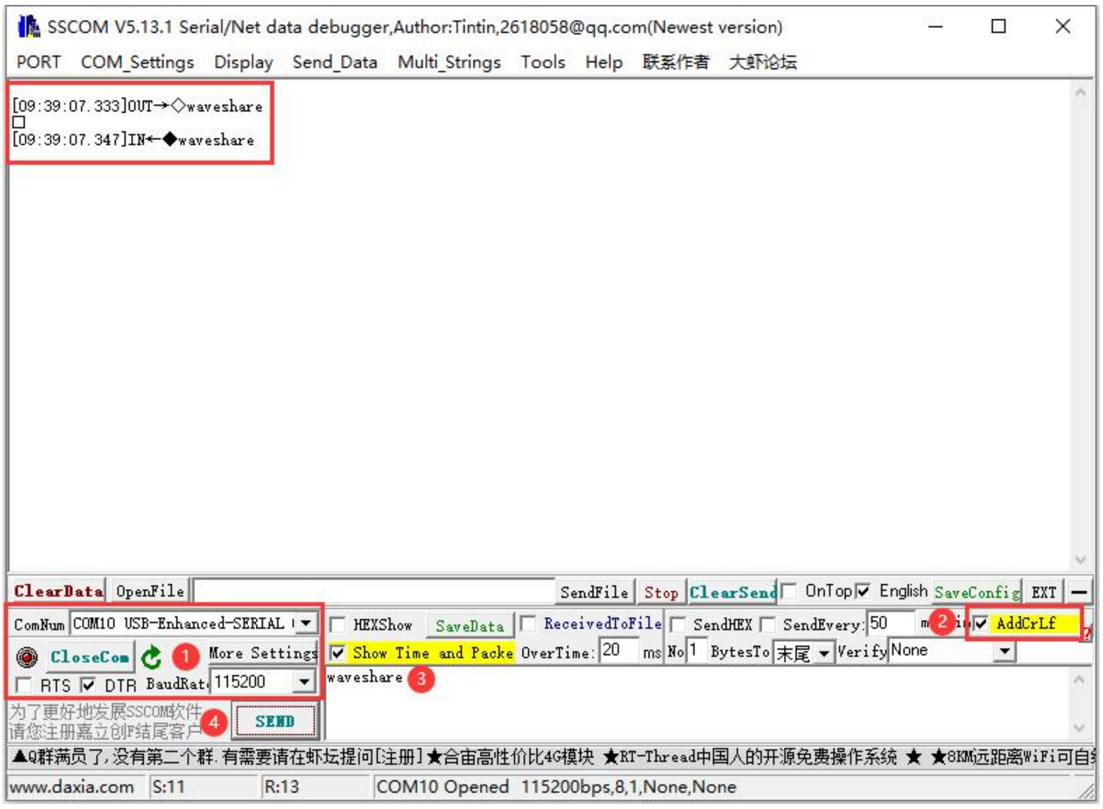
```
waveshare
```

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 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, 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.


SSCOM V5.13.1 Serial/Net data debugger,Author:Tintin,2618058@qq.com(Newest version)
PORT COM_Settings Display Send_Data Multi_Strings Tools Help 联系作者 大虾论坛
ClearData OpenFile SendFile Stop ClearSend OnTop English SaveConfig EXT
CloseCom More Settings HEXShow SaveData ReceivedToFile SendHEX SEND
为了更好地发展SSCOM软件 请您注册嘉立创F结尾客户
▲Q群满员了,没有第二个群.有需要请在虾坛提问[注册]★合宙高性价比4G模块 ★RT-Thread中国人的开源免费操作系统 ★ ★8KM远距离WiFi可自
1
2
3
4

```
[09:39:07.333]OUT→◇waveshare
□
[09:39:07.347]IN←◆waveshare
```

ComNum: COM10 USB-Enhanced-SERIAL

☐ RTS ☑ DTR BaudRate: 115200

☑ Show Time and Packe OverTime: 20 ms No 1 BytesTo 末尾 Verify None

SendEvery: 50 ms ☑ AddCrLf

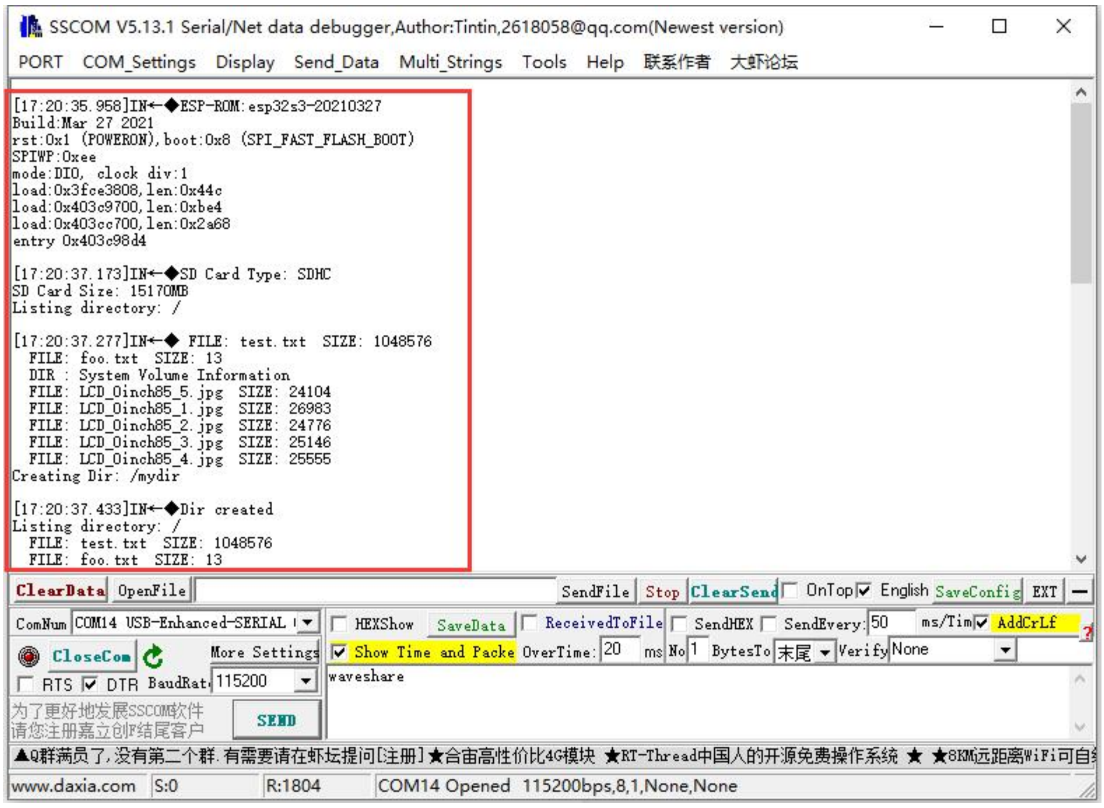
waveshare

www.daxia.com S:11 R:13 COM10 Opened 115200bps,8,1,None,None

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.

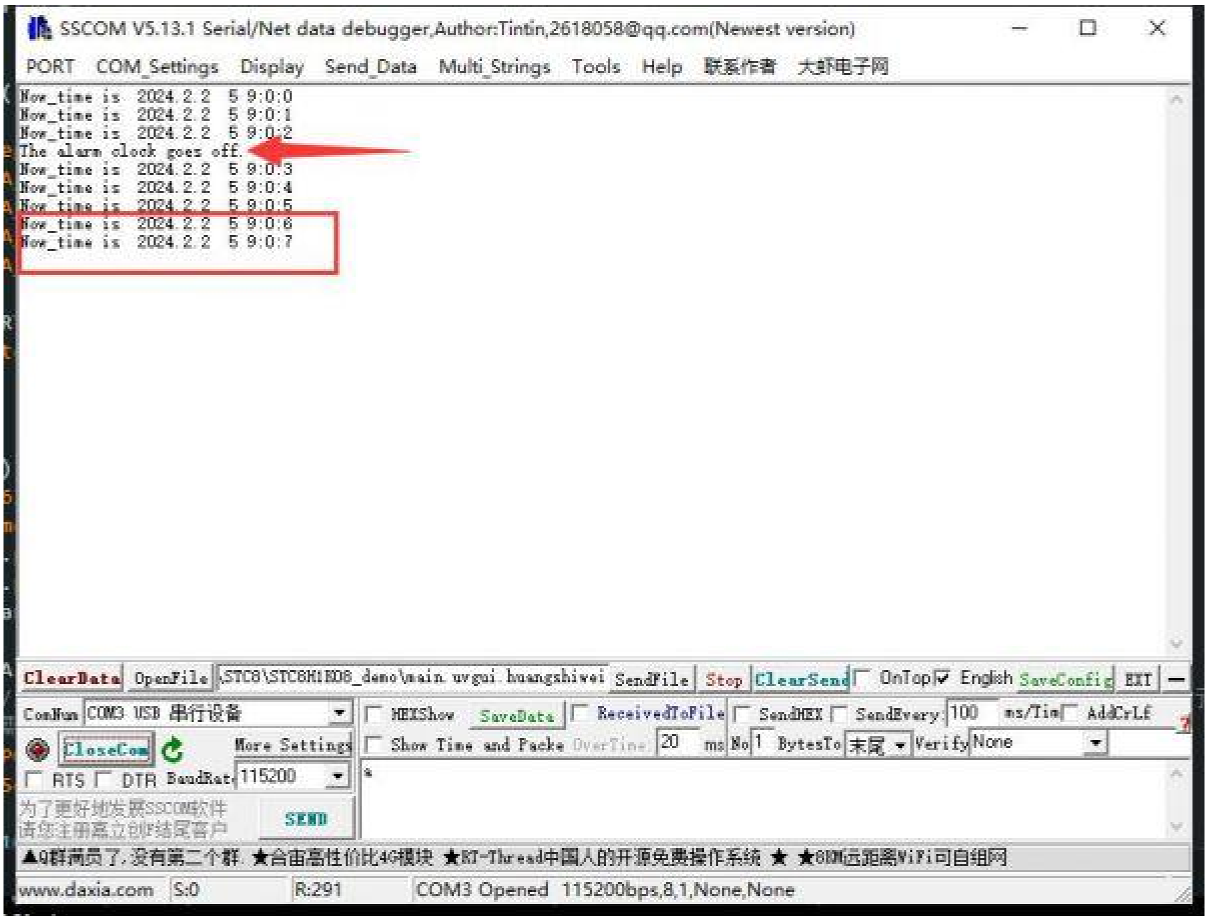
After uploading the demo, ESP32-S3-Touch-LCD-4.3B will recognize the type and size of the SD card, and then you can operate the files on the SD card:


SSCOM V5.13.1 Serial/Net data debugger,Author:Tintin,2618058@qq.com(Newest version)
PORT COM_Settings Display Send_Data Multi_Strings Tools Help 联系作者 大虾论坛
ClearData OpenFile SendFile Stop ClearSend OnTop English SaveConfig EXT
ComNum COM14 USB-Enhanced-SERIAL HEXShow SaveData ReceivedToFile SendHEX SendEvery: 50 ms/Tim AddCrLf
CloseCom More Settings Show Time and Packe OverTime: 20 ms No 1 BytesTo 末尾 Verify None
RTS DTR BaudRat 115200 SEND
www.daxia.com S:0 R:1804 COM14 Opened 115200bps,8,1,None,None

```
[17:20:35.958]IN←◆ESP-ROM:esp32s3-20210327
Build:Mar 27 2021
rst:0x1 (POWERON),boot:0x8 (SPI_FAST_FLASH_BOOT)
SPIWP:0xee
mode:DIO, clock div:1
load:0x3fce3808,len:0x44c
load:0x403c9700,len:0xbe4
load:0x403cc700,len:0x2a68
entry 0x403c98d4

[17:20:37.173]IN←◆SD Card Type: SDHC
SD Card Size: 15170MB
Listing directory: /

[17:20:37.277]IN←◆ FILE: test.txt  SIZE: 1048576
  FILE: foo.txt  SIZE: 13
  DIR : System Volume Information
  FILE: LCD_0inch85_5.jpg  SIZE: 24104
  FILE: LCD_0inch85_1.jpg  SIZE: 26983
  FILE: LCD_0inch85_2.jpg  SIZE: 24776
  FILE: LCD_0inch85_3.jpg  SIZE: 25146
  FILE: LCD_0inch85_4.jpg  SIZE: 25555
Creating Dir: /mydir

[17:20:37.433]IN←◆Dir created
Listing directory: /
  FILE: test.txt  SIZE: 1048576
  FILE: foo.txt  SIZE: 13
```

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.



```
Now_time is  2024.2.2  5 9:0:0
Now_time is  2024.2.2  5 9:0:1
Now_time is  2024.2.2  5 9:0:2
The alarm clock goes off.
Now_time is  2024.2.2  5 9:0:3
Now_time is  2024.2.2  5 9:0:4
Now_time is  2024.2.2  5 9:0:5
Now_time is  2024.2.2  5 9:0:6
Now_time is  2024.2.2  5 9:0:7
```

SSCOM V5.13.1 Serial/Net data debugger,Author:Tintin,2618058@qq.com(Newest version)
PORT COM_Settings Display Send_Data Multi_Strings Tools Help 联系作者 大虾电子网
ClearData OpenFile SendFile Stop ClearSend OnTop English SaveConfig EXT
ComNum COM3 USB 串行设备
HEXShow SaveData ReceivedToFile SendHEX SendEvery 100 ms/Tim AddCrLf
CloseCom More Settings Show Time and Packe OverTime 20 ms No 1 BytesTo 末尾 Verify None
RTS DTR BaudRat 115200
SEND
www.daxia.com S:0 R:291 COM3 Opened 115200bps,8,1,None,None

IO_Test

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

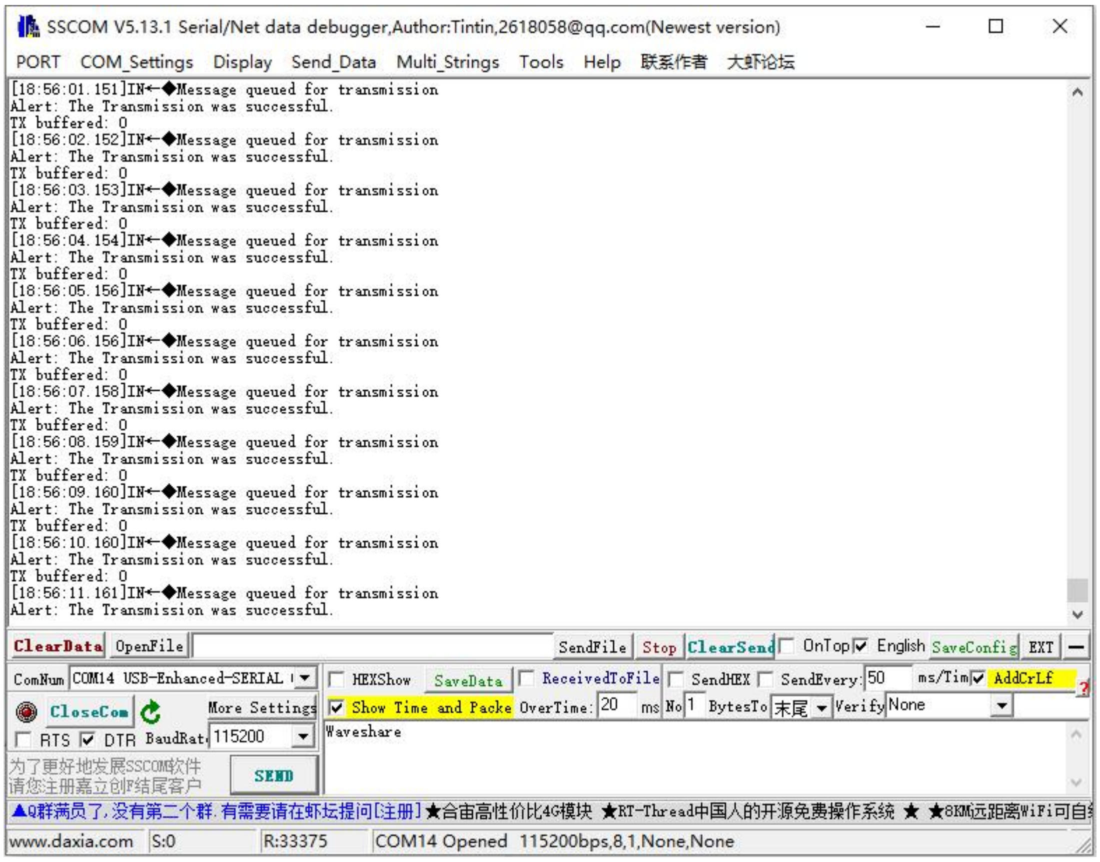
After burning the code, the test passes with a green screen, and the test fails with a red screen.

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.

Once you open the serial port debugging assistant, you can observe that the ESP32-S3-Touch-LCD-4.3B has started sending CAN messages.



```
[18:56:01.151]IN←◆Message queued for transmission
Alert: The Transmission was successful.
TX buffered: 0
[18:56:02.152]IN←◆Message queued for transmission
Alert: The Transmission was successful.
TX buffered: 0
[18:56:03.153]IN←◆Message queued for transmission
Alert: The Transmission was successful.
TX buffered: 0
[18:56:04.154]IN←◆Message queued for transmission
Alert: The Transmission was successful.
TX buffered: 0
[18:56:05.156]IN←◆Message queued for transmission
Alert: The Transmission was successful.
TX buffered: 0
[18:56:06.156]IN←◆Message queued for transmission
Alert: The Transmission was successful.
TX buffered: 0
[18:56:07.158]IN←◆Message queued for transmission
Alert: The Transmission was successful.
TX buffered: 0
[18:56:08.159]IN←◆Message queued for transmission
Alert: The Transmission was successful.
TX buffered: 0
[18:56:09.160]IN←◆Message queued for transmission
Alert: The Transmission was successful.
TX buffered: 0
[18:56:10.160]IN←◆Message queued for transmission
Alert: The Transmission was successful.
TX buffered: 0
[18:56:11.161]IN←◆Message queued for transmission
Alert: The Transmission was successful.
```

SSCOM V5.13.1 Serial/Net data debugger,Author:Tintin,2618058@qq.com(Newest version)

ComNum: COM14 USB-Enhanced-SERIAL
BaudRate: 115200
[✓ DTR] [✓ Show Time and Packe] OverTime: 20 ms No 1 BytesTo 末尾 Verify None [✓ English] SendEvery: 50 ms/Tim [✓ AddCrLf]

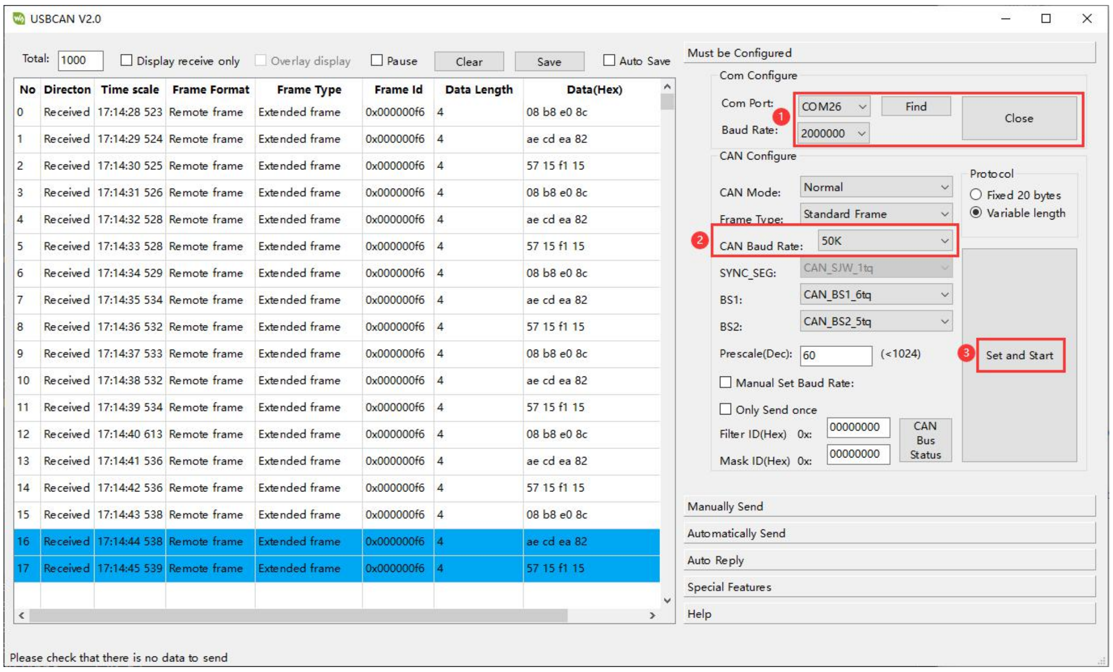
Waveshare

为了更好地发展SSCOM软件
请您注册嘉立创F结尾客户

▲Q群满员了,没有第二个群.有需要请在虾坛提问[注册]★合宙高性价比4G模块 ★RT-Thread中国人的开源免费操作系统 ★ ★8KM远距离WiFi可自

www.daxia.com S:0 R:33375 COM14 Opened 115200bps,8,1,None,None

Connect the USB-CAN-A to the computer and open the USB-CAN-A_TOOL_2.0. 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.



USBCAN V2.0

Total: 1000

No	Directon	Time scale	Frame Format	Frame Type	Frame Id	Data Length	Data(Hex)
0	Received	17:14:28 523	Remote frame	Extended frame	0x000000f6	4	08 b8 e0 8c
1	Received	17:14:29 524	Remote frame	Extended frame	0x000000f6	4	ae cd ea 82
2	Received	17:14:30 525	Remote frame	Extended frame	0x000000f6	4	57 15 f1 15
3	Received	17:14:31 526	Remote frame	Extended frame	0x000000f6	4	08 b8 e0 8c
4	Received	17:14:32 528	Remote frame	Extended frame	0x000000f6	4	ae cd ea 82
5	Received	17:14:33 528	Remote frame	Extended frame	0x000000f6	4	57 15 f1 15
6	Received	17:14:34 529	Remote frame	Extended frame	0x000000f6	4	08 b8 e0 8c
7	Received	17:14:35 534	Remote frame	Extended frame	0x000000f6	4	ae cd ea 82
8	Received	17:14:36 532	Remote frame	Extended frame	0x000000f6	4	57 15 f1 15
9	Received	17:14:37 533	Remote frame	Extended frame	0x000000f6	4	08 b8 e0 8c
10	Received	17:14:38 532	Remote frame	Extended frame	0x000000f6	4	ae cd ea 82
11	Received	17:14:39 534	Remote frame	Extended frame	0x000000f6	4	57 15 f1 15
12	Received	17:14:40 613	Remote frame	Extended frame	0x000000f6	4	08 b8 e0 8c
13	Received	17:14:41 536	Remote frame	Extended frame	0x000000f6	4	ae cd ea 82
14	Received	17:14:42 536	Remote frame	Extended frame	0x000000f6	4	57 15 f1 15
15	Received	17:14:43 538	Remote frame	Extended frame	0x000000f6	4	08 b8 e0 8c
16	Received	17:14:44 538	Remote frame	Extended frame	0x000000f6	4	ae cd ea 82
17	Received	17:14:45 539	Remote frame	Extended frame	0x000000f6	4	57 15 f1 15

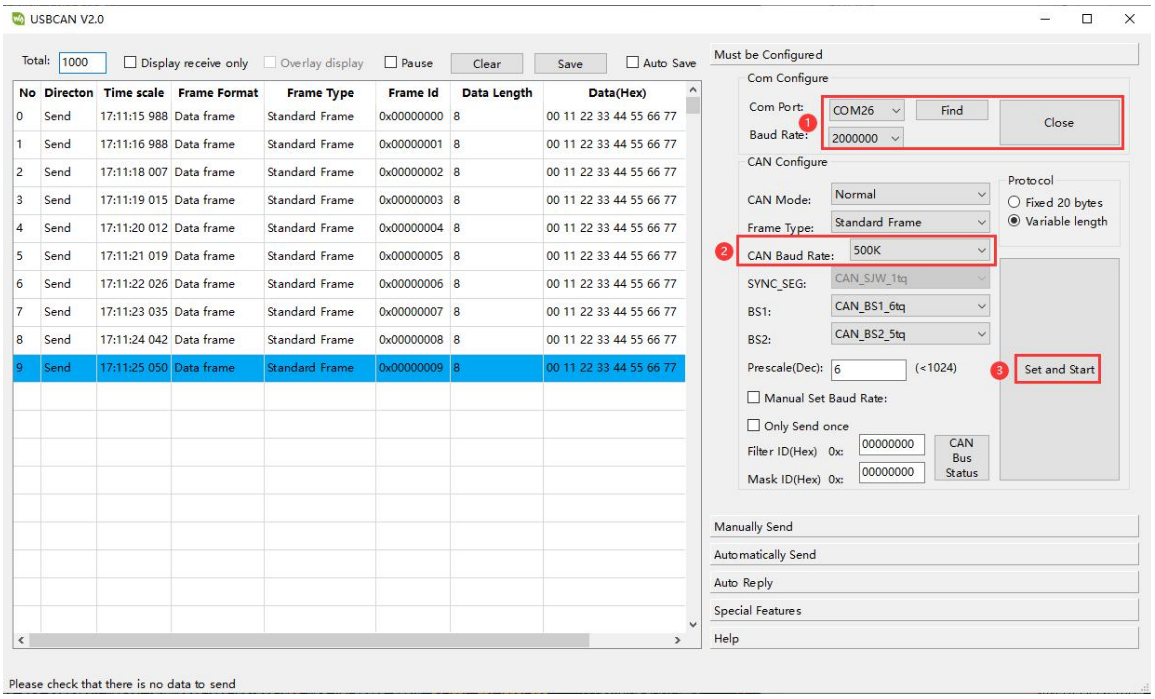
Must be Configured
Com Configure
1 Com Port: COM26
Baud Rate: 2000000
CAN Configure
CAN Mode: Normal
Frame Type: Standard Frame
2 CAN Baud Rate: 50K
SYNC_SEG: CAN_SJW_1tq
BS1: CAN_BS1_6tq
BS2: CAN_BS2_5tq
Prescale(Dec): 60 (<1024)
3 Set and Start
Protocol: Fixed 20 bytes [✓ Variable length]
Filter ID(Hex) 0x: 00000000
Mask ID(Hex) 0x: 00000000

Please check that there is no data to send

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.
Connect the USB-CAN-A to the computer and open the USB-CAN-A_TOOL_2.0. 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.


USBCAN V2.0

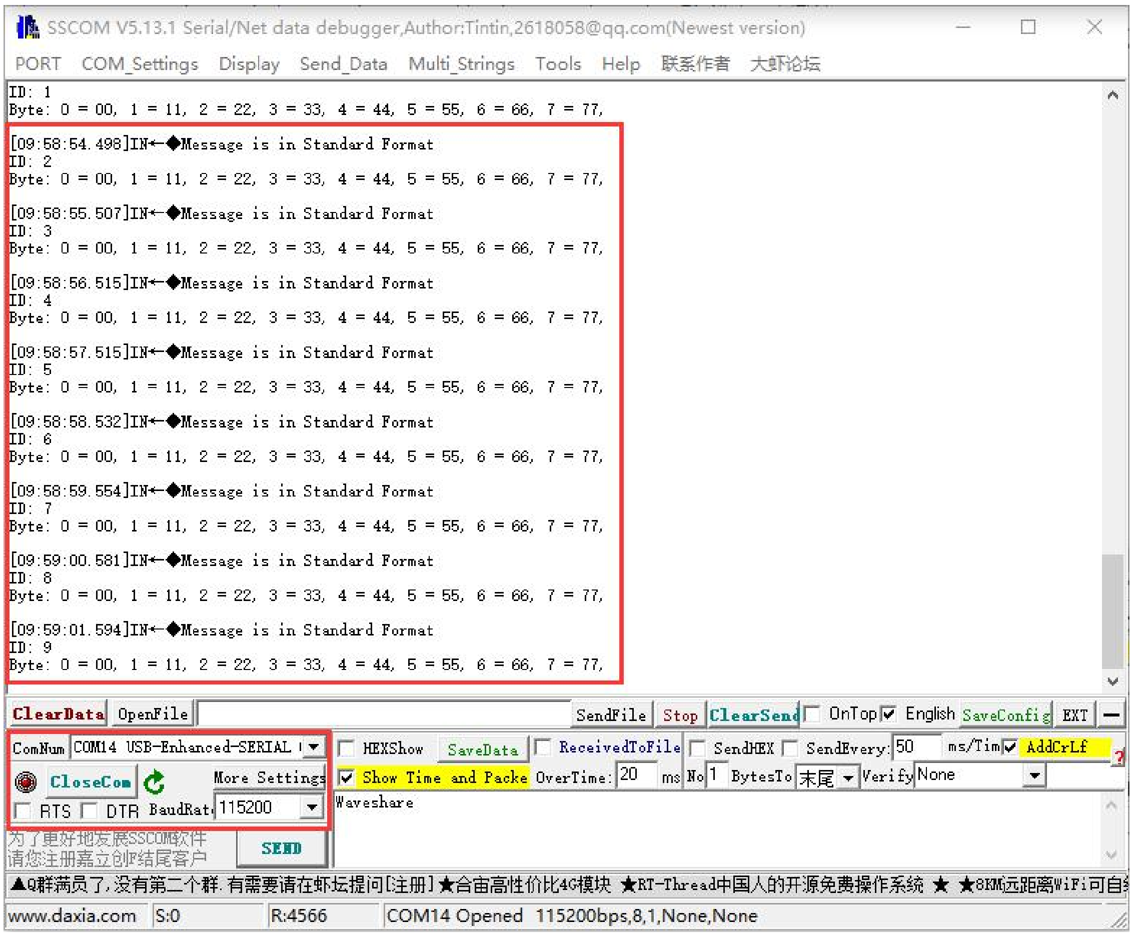
Total: 1000 ☐ Display receive only ☐ Overlay display ☐ Pause Clear Save ☐ Auto Save

No	Directon	Time scale	Frame Format	Frame Type	Frame Id	Data Length	Data(Hex)
0	Send	17:11:15 988	Data frame	Standard Frame	0x00000000	8	00 11 22 33 44 55 66 77
1	Send	17:11:16 988	Data frame	Standard Frame	0x00000001	8	00 11 22 33 44 55 66 77
2	Send	17:11:18 007	Data frame	Standard Frame	0x00000002	8	00 11 22 33 44 55 66 77
3	Send	17:11:19 015	Data frame	Standard Frame	0x00000003	8	00 11 22 33 44 55 66 77
4	Send	17:11:20 012	Data frame	Standard Frame	0x00000004	8	00 11 22 33 44 55 66 77
5	Send	17:11:21 019	Data frame	Standard Frame	0x00000005	8	00 11 22 33 44 55 66 77
6	Send	17:11:22 026	Data frame	Standard Frame	0x00000006	8	00 11 22 33 44 55 66 77
7	Send	17:11:23 035	Data frame	Standard Frame	0x00000007	8	00 11 22 33 44 55 66 77
8	Send	17:11:24 042	Data frame	Standard Frame	0x00000008	8	00 11 22 33 44 55 66 77
9	Send	17:11:25 050	Data frame	Standard Frame	0x00000009	8	00 11 22 33 44 55 66 77

Must be Configured
Com Configure
1 Com Port: COM26 Find Close
Baud Rate: 2000000
CAN Configure
CAN Mode: Normal
Frame Type: Standard Frame
2 CAN Baud Rate: 500K
SYNC_SEG: CAN_SJW_1tq
BS1: CAN_BS1_6tq
BS2: CAN_BS2_5tq
Prescale(Dec): 6 (<1024)
3 Set and Start
☐ Manual Set Baud Rate:
☐ Only Send once
Filter ID(Hex) 0x: 00000000
Mask ID(Hex) 0x: 00000000
CAN Bus Status
Protocol
○ Fixed 20 bytes
◉ Variable length
Manually Send
Automatically Send
Auto Reply
Special Features
Help

Please check that there is no data to send

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.)



```
ID: 1
Byte: 0 = 00, 1 = 11, 2 = 22, 3 = 33, 4 = 44, 5 = 55, 6 = 66, 7 = 77,

[09:58:54.498]IN←◆Message is in Standard Format
ID: 2
Byte: 0 = 00, 1 = 11, 2 = 22, 3 = 33, 4 = 44, 5 = 55, 6 = 66, 7 = 77,

[09:58:55.507]IN←◆Message is in Standard Format
ID: 3
Byte: 0 = 00, 1 = 11, 2 = 22, 3 = 33, 4 = 44, 5 = 55, 6 = 66, 7 = 77,

[09:58:56.515]IN←◆Message is in Standard Format
ID: 4
Byte: 0 = 00, 1 = 11, 2 = 22, 3 = 33, 4 = 44, 5 = 55, 6 = 66, 7 = 77,

[09:58:57.515]IN←◆Message is in Standard Format
ID: 5
Byte: 0 = 00, 1 = 11, 2 = 22, 3 = 33, 4 = 44, 5 = 55, 6 = 66, 7 = 77,

[09:58:58.532]IN←◆Message is in Standard Format
ID: 6
Byte: 0 = 00, 1 = 11, 2 = 22, 3 = 33, 4 = 44, 5 = 55, 6 = 66, 7 = 77,

[09:58:59.554]IN←◆Message is in Standard Format
ID: 7
Byte: 0 = 00, 1 = 11, 2 = 22, 3 = 33, 4 = 44, 5 = 55, 6 = 66, 7 = 77,

[09:59:00.581]IN←◆Message is in Standard Format
ID: 8
Byte: 0 = 00, 1 = 11, 2 = 22, 3 = 33, 4 = 44, 5 = 55, 6 = 66, 7 = 77,

[09:59:01.594]IN←◆Message is in Standard Format
ID: 9
Byte: 0 = 00, 1 = 11, 2 = 22, 3 = 33, 4 = 44, 5 = 55, 6 = 66, 7 = 77,
```

ComNum: COM14 USB-Enhanced-SERIAL
BaudRate: 115200
[✓ Show Time and Packe] OverTime: 20 ms No 1 BytesTo 末尾 Verify None
SendEvery: 50 ms/Tim [✓ AddCrLf]
[✓ English]

Waveshare

www.daxia.com S:0 R:4566 COM14 Opened 115200bps,8,1,None,None

SSCOM V5.13.1 Serial/Net data debugger,Author:Tintin,2618058@qq.com(Newest version)
PORT COM_Settings Display Send_Data Multi_Strings Tools Help 联系作者 大虾论坛
ClearData OpenFile SendFile Stop ClearSend OnTop SaveConfig EXT
CloseCom More Settings RTS DTR HEXShow SaveData ReceivedToFile SendHEX SEND
为了更好地发展SSCOM软件 请您注册嘉立创F结尾客户
▲Q群满员了,没有第二个群.有需要请在虾坛提问[注册]★合宙高性价比4G模块 ★RT-Thread中国人的开源免费操作系统 ★ ★8KM远距离WiFi可自

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)).

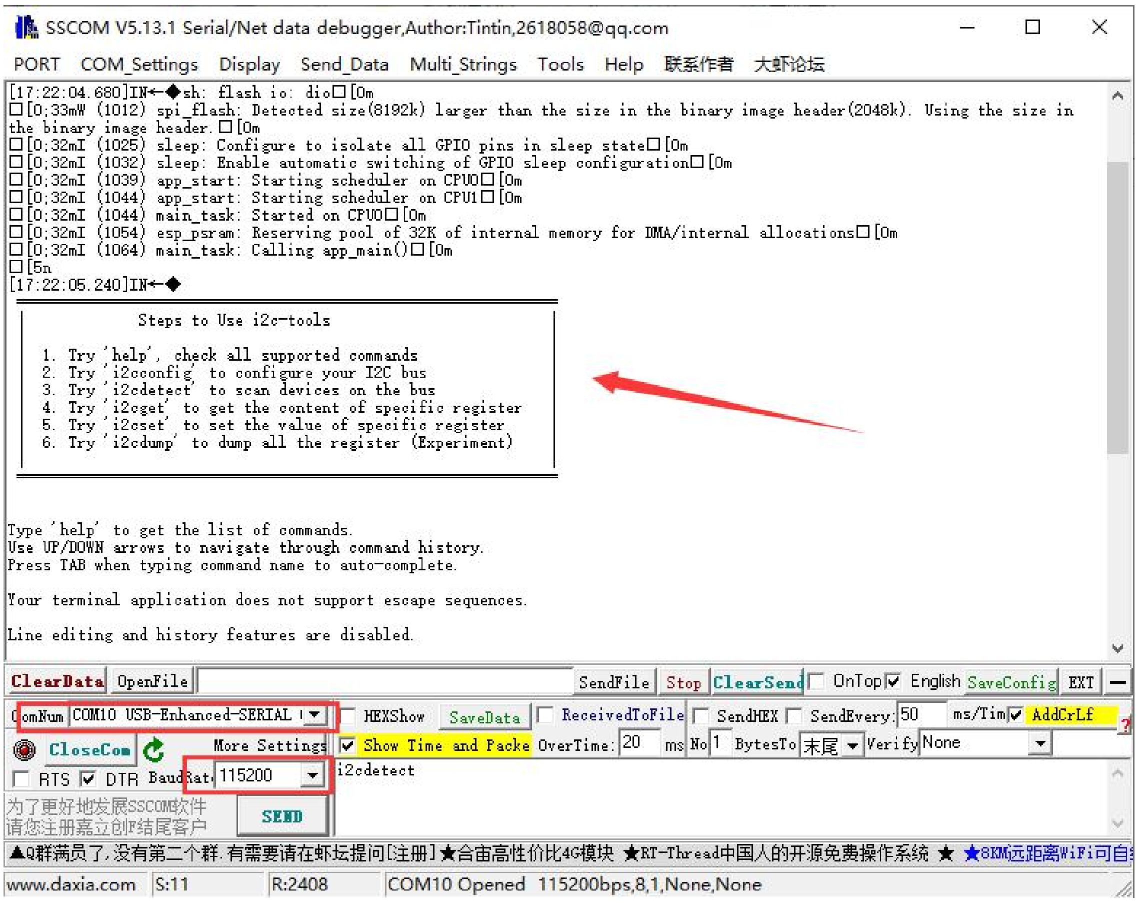
ESP-IDF

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.

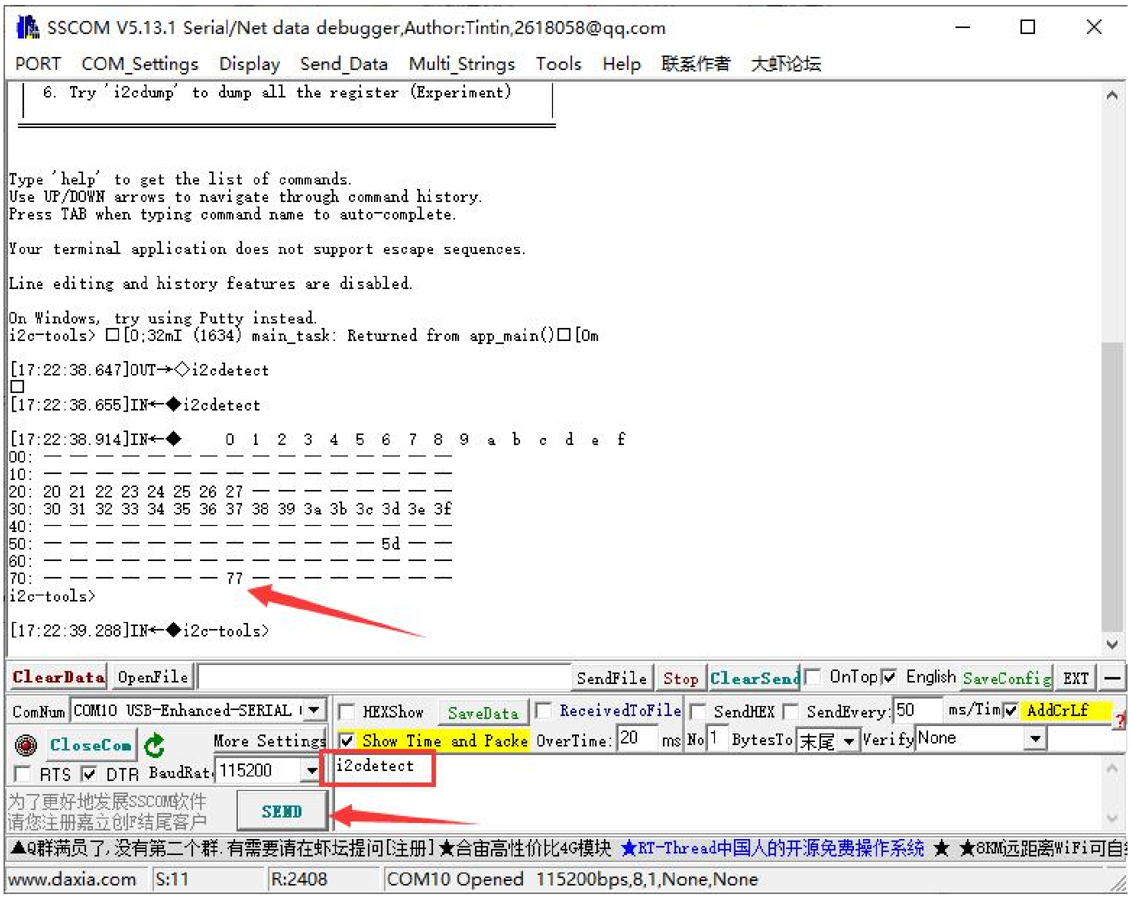
After uploading the code, connect the I2C device (in this case, using the BME680 Environmental Sensor) to the corresponding pins on the ESP32-S3-Touch-LCD-4.3B. Then, open the serial debugging assistant, select a baud rate of 115200, and choose the corresponding COM port for communication (make sure to close the ESP-IDF's COM port first, as it may occupy the COM port to prevent the serial port from opening).


SSCOM V5.13.1 Serial/Net data debugger,Author:Tintin,2618058@qq.com
PORT COM_Settings Display Send_Data Multi_Strings Tools Help 联系作者 大虾论坛
ClearData OpenFile SendFile Stop ClearSend OnTop English SaveConfig EXT
ComNum COM10 USB-Enhanced-SERIAL
HEXShow SaveData ReceivedToFile SendHEX SendEvery: 50 ms/Tim AddCrLf
CloseCom More Settings Show Time and Packe OverTime: 20 ms No 1 BytesTo 末尾 Verify None
RTS DTR BaudRate 115200
为了更好地发展SSCOM软件 请您注册嘉立创F结尾客户
SEND
▲Q群满员了,没有第二个群.有需要请在虾坛提问[注册]★合宙高性价比4G模块 ★RT-Thread中国人的开源免费操作系统 ★ ★8KM远距离WiFi可自
www.daxia.com S:11 R:2408 COM10 Opened 115200bps,8,1,None,None

```
[17:22:04.680]IN←◆sh: flash io: dio□[0m
□[0;33mW (1012) spi_flash: Detected size(8192k) larger than the size in the binary image header(2048k). Using the size in
the binary image header.□[0m
□[0;32mI (1025) sleep: Configure to isolate all GPIO pins in sleep state□[0m
□[0;32mI (1032) sleep: Enable automatic switching of GPIO sleep configuration□[0m
□[0;32mI (1039) app_start: Starting scheduler on CPU0□[0m
□[0;32mI (1044) app_start: Starting scheduler on CPU1□[0m
□[0;32mI (1044) main_task: Started on CPU0□[0m
□[0;32mI (1054) esp_psram: Reserving pool of 32K of internal memory for DMA/internal allocations□[0m
□[0;32mI (1064) main_task: Calling app_main()□[0m
□[5n
[17:22:05.240]IN←◆
 ==============================================================
 |             Steps to Use i2c-tools                         |
 |                                                            |
 |  1. Try 'help', check all supported commands               |
 |  2. Try 'i2cconfig' to configure your I2C bus              |
 |  3. Try 'i2cdetect' to scan devices on the bus             |
 |  4. Try 'i2cget' to get the content of specific register   |
 |  5. Try 'i2cset' to set the value of specific register     |
 |  6. Try 'i2cdump' to dump all the register (Experiment)    |
 |                                                            |
 ==============================================================


Type 'help' to get the list of commands.
Use UP/DOWN arrows to navigate through command history.
Press TAB when typing command name to auto-complete.

Your terminal application does not support escape sequences.

Line editing and history features are disabled.
```

```
i2cdetect
```

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.



```
 6. Try 'i2cdump' to dump all the register (Experiment)

Type 'help' to get the list of commands.
Use UP/DOWN arrows to navigate through command history.
Press TAB when typing command name to auto-complete.

Your terminal application does not support escape sequences.

Line editing and history features are disabled.

On Windows, try using Putty instead.
i2c-tools> □[0;32mI (1634) main_task: Returned from app_main()□[0m

[17:22:38.647]OUT→◇i2cdetect
□
[17:22:38.655]IN←◆i2cdetect

[17:22:38.914]IN←◆     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: -- -- -- -- -- -- -- -- -- -- -- -- -- 5d -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- 77 -- -- -- -- -- -- -- --
i2c-tools>

[17:22:39.288]IN←◆i2c-tools>
```

ComNum: COM10 USB-Enhanced-SERIAL
BaudRate: 115200
Show Time and Packe ✓
AddCrLf ✓
Send: i2cdetect

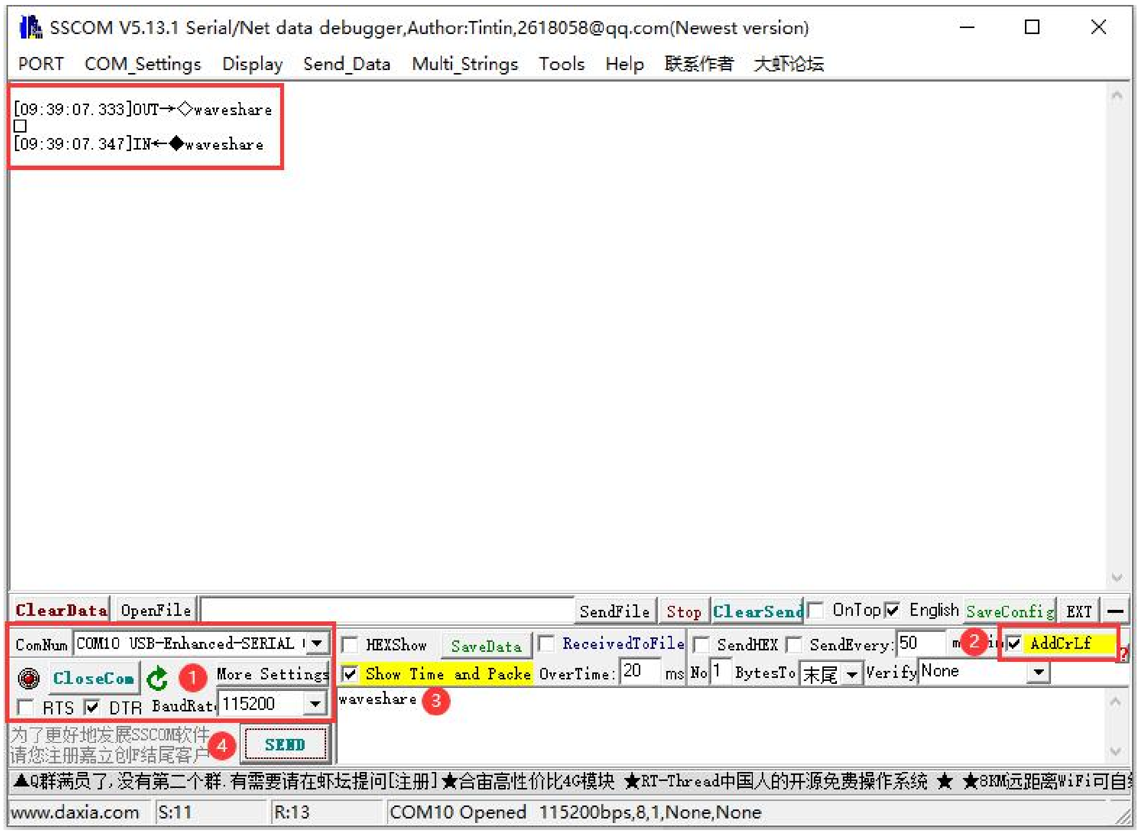
www.daxia.com S:11 R:2408 COM10 Opened 115200bps,8,1,None,None

RS485_Test

RS485_Test is for testing the RS485 socket.

After uploading the code, connect the USB to RS485 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 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.


```
[09:39:07.333]OUT→◇waveshare
□
[09:39:07.347]IN←◆waveshare
```
ComNum: COM10 USB-Enhanced-SERIAL; BaudRate: 115200; ✓ DTR; ✓ Show Time and Packe; OverTime: 20 ms; No 1; ✓ AddCrLf; SendEvery: 50 ms

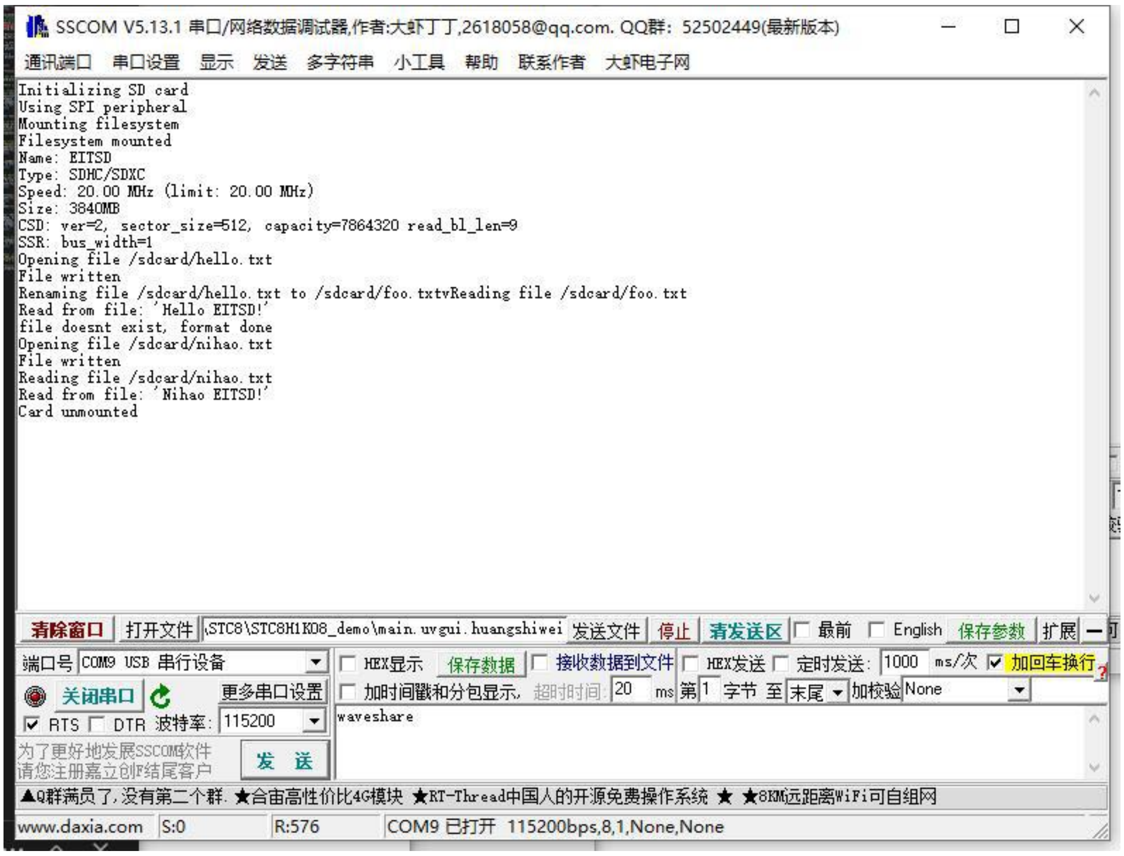
waveshare

www.daxia.com S:11 R:13 COM10 Opened 115200bps,8,1,None,None

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.


SSCOM V5.13.1 串口/网络数据调试器,作者:大虾丁丁,2618058@qq.com. QQ群: 52502449(最新版本)
通讯端口 串口设置 显示 发送 多字符串 小工具 帮助 联系作者 大虾电子网
清除窗口 打开文件 发送文件 停止 清发送区 最前 English 保存参数 扩展
端口号 COM9 USB 串行设备
HEX显示 保存数据 接收数据到文件 HEX发送 定时发送: 1000 ms/次 加回车换行
关闭串口 更多串口设置 加时间戳和分包显示, 超时时间: 20 ms 第1 字节 至 末尾 加校验 None
RTS DTR 波特率: 115200
发 送
www.daxia.com S:0 R:576 COM9 已打开 115200bps,8,1,None,None

```
Initializing SD card
Using SPI peripheral
Mounting filesystem
Filesystem mounted
Name: EITSD
Type: SDHC/SDXC
Speed: 20.00 MHz (limit: 20.00 MHz)
Size: 3840MB
CSD: ver=2, sector_size=512, capacity=7864320 read_bl_len=9
SSR: bus_width=1
Opening file /sdcard/hello.txt
File written
Renaming file /sdcard/hello.txt to /sdcard/foo.txtvReading file /sdcard/foo.txt
Read from file: 'Hello EITSD!'
file doesnt exist, format done
Opening file /sdcard/nihao.txt
File written
Reading file /sdcard/nihao.txt
Read from file: 'Nihao EITSD!'
Card unmounted
```

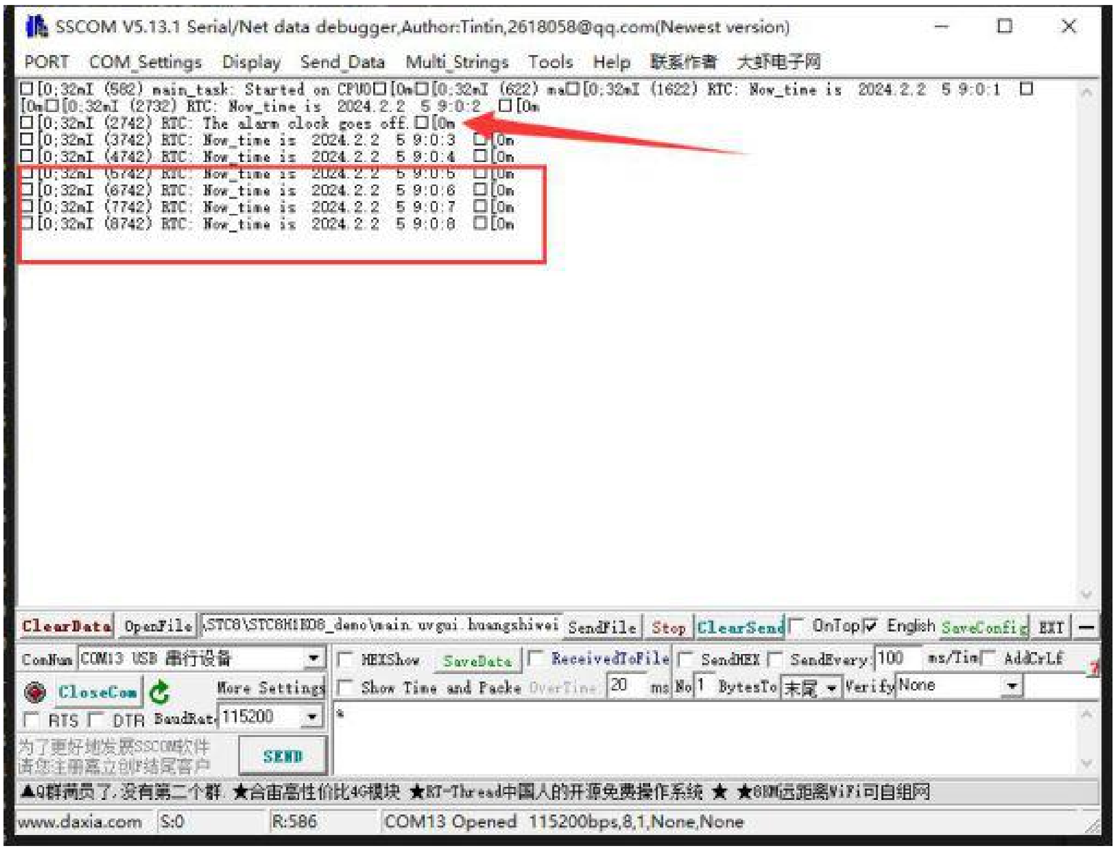
```
waveshare
```

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.



```
□[0;32mI (582) main_task: Started on CPU0□[0m□[0;32mI (622) ma□[0;32mI (1622) RTC: Now_time is  2024.2.2  5 9:0:1  □
[0m□[0;32mI (2732) RTC: Now_time is  2024.2.2  5 9:0:2  □[0m
□[0;32mI (2742) RTC: The alarm clock goes off.□[0m
□[0;32mI (3742) RTC: Now_time is  2024.2.2  5 9:0:3  □[0m
□[0;32mI (4742) RTC: Now_time is  2024.2.2  5 9:0:4  □[0m
□[0;32mI (5742) RTC: Now_time is  2024.2.2  5 9:0:5  □[0m
□[0;32mI (6742) RTC: Now_time is  2024.2.2  5 9:0:6  □[0m
□[0;32mI (7742) RTC: Now_time is  2024.2.2  5 9:0:7  □[0m
□[0;32mI (8742) RTC: Now_time is  2024.2.2  5 9:0:8  □[0m
```

SSCOM V5.13.1 Serial/Net data debugger,Author:Tintin,2618058@qq.com(Newest version)
PORT COM_Settings Display Send_Data Multi_Strings Tools Help 联系作者 大虾电子网
ClearData OpenFile SendFile Stop ClearSend OnTop English SaveConfig EXT
ComNum COM13 USB 串行设备 HEXShow SaveData ReceivedToFile SendHEX SendEvery 100 ms/Tim AddCrLf
CloseCom More Settings Show Time and Packe OverTime 20 ms No 1 BytesTo 末尾 Verify None
RTS DTR BaudRat 115200
SEND
www.daxia.com S:0 R:586 COM13 Opened 115200bps,8,1,None,None

IO_Test

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

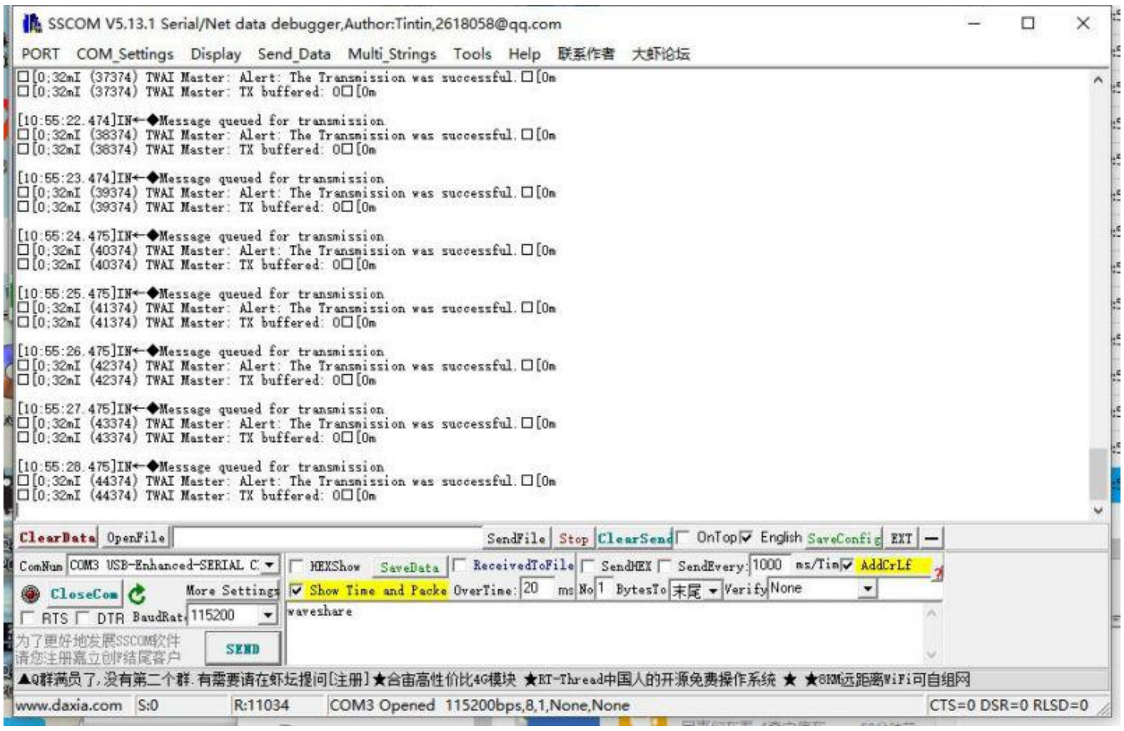
After burning the code, the test passes with a green screen, and the test fails with a red screen.

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.

Open the SSCOM, and you can see the ESP32-S3-Touch-LCD-4.3B starts to send the CAN message.



SSCOM V5.13.1 Serial/Net data debugger,Author:Tintin,2618058@qq.com

```
□[0;32mI (37374) TWAI Master: Alert: The Transmission was successful.□[0m
□[0;32mI (37374) TWAI Master: TX buffered: 0□[0m

[10:55:22.474]IN←◆Message queued for transmission
□[0;32mI (38374) TWAI Master: Alert: The Transmission was successful.□[0m
□[0;32mI (38374) TWAI Master: TX buffered: 0□[0m

[10:55:23.474]IN←◆Message queued for transmission
□[0;32mI (39374) TWAI Master: Alert: The Transmission was successful.□[0m
□[0;32mI (39374) TWAI Master: TX buffered: 0□[0m

[10:55:24.475]IN←◆Message queued for transmission
□[0;32mI (40374) TWAI Master: Alert: The Transmission was successful.□[0m
□[0;32mI (40374) TWAI Master: TX buffered: 0□[0m

[10:55:25.475]IN←◆Message queued for transmission
□[0;32mI (41374) TWAI Master: Alert: The Transmission was successful.□[0m
□[0;32mI (41374) TWAI Master: TX buffered: 0□[0m

[10:55:26.475]IN←◆Message queued for transmission
□[0;32mI (42374) TWAI Master: Alert: The Transmission was successful.□[0m
□[0;32mI (42374) TWAI Master: TX buffered: 0□[0m

[10:55:27.475]IN←◆Message queued for transmission
□[0;32mI (43374) TWAI Master: Alert: The Transmission was successful.□[0m
□[0;32mI (43374) TWAI Master: TX buffered: 0□[0m

[10:55:28.475]IN←◆Message queued for transmission
□[0;32mI (44374) TWAI Master: Alert: The Transmission was successful.□[0m
□[0;32mI (44374) TWAI Master: TX buffered: 0□[0m
```

ComNum: COM3 USB-Enhanced-SERIAL C; BaudRat: 115200; SendEvery: 1000 ms/Tim; ✓ AddCrLf; ✓ Show Time and Packe; OverTime: 20 ms; No 1; BytesTo 末尾; Verify None; ✓ OnTop; ✓ English

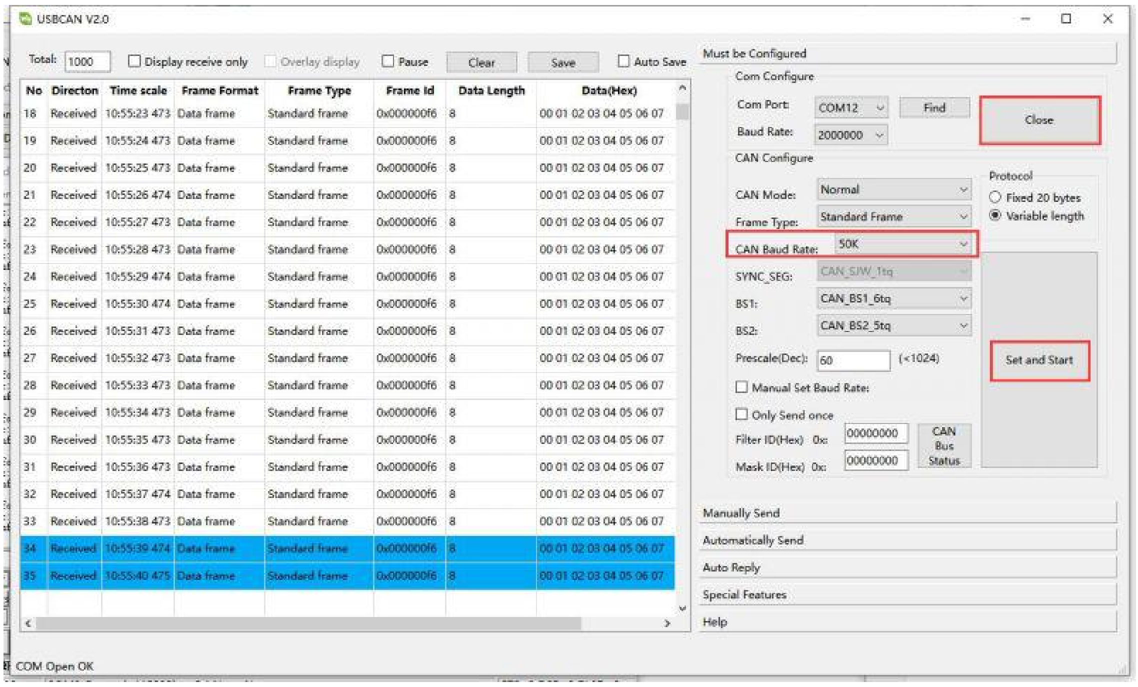
waveshare

为了更好地发展SSCOM软件 请您注册嘉立创P结尾客户

▲Q群满员了,没有第二个群.有需要请在虾坛提问[注册]★合宙高性价比4G模块 ★RT-Thread中国人的开源免费操作系统 ★ ★8KM远距离WiFi可自组网

www.daxia.com S:0 R:11034 COM3 Opened 115200bps,8,1,None,None CTS=0 DSR=0 RLSD=0

Connect the USB-CAN-A to the computer, open USB-CAN-A-Tool-2.0, 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.



USBCAN V2.0

Total: 1000

No	Directon	Time scale	Frame Format	Frame Type	Frame Id	Data Length	Data(Hex)
18	Received	10:55:23 473	Data frame	Standard frame	0x000000f6	8	00 01 02 03 04 05 06 07
19	Received	10:55:24 473	Data frame	Standard frame	0x000000f6	8	00 01 02 03 04 05 06 07
20	Received	10:55:25 473	Data frame	Standard frame	0x000000f6	8	00 01 02 03 04 05 06 07
21	Received	10:55:26 474	Data frame	Standard frame	0x000000f6	8	00 01 02 03 04 05 06 07
22	Received	10:55:27 473	Data frame	Standard frame	0x000000f6	8	00 01 02 03 04 05 06 07
23	Received	10:55:28 473	Data frame	Standard frame	0x000000f6	8	00 01 02 03 04 05 06 07
24	Received	10:55:29 474	Data frame	Standard frame	0x000000f6	8	00 01 02 03 04 05 06 07
25	Received	10:55:30 474	Data frame	Standard frame	0x000000f6	8	00 01 02 03 04 05 06 07
26	Received	10:55:31 473	Data frame	Standard frame	0x000000f6	8	00 01 02 03 04 05 06 07
27	Received	10:55:32 473	Data frame	Standard frame	0x000000f6	8	00 01 02 03 04 05 06 07
28	Received	10:55:33 473	Data frame	Standard frame	0x000000f6	8	00 01 02 03 04 05 06 07
29	Received	10:55:34 473	Data frame	Standard frame	0x000000f6	8	00 01 02 03 04 05 06 07
30	Received	10:55:35 473	Data frame	Standard frame	0x000000f6	8	00 01 02 03 04 05 06 07
31	Received	10:55:36 473	Data frame	Standard frame	0x000000f6	8	00 01 02 03 04 05 06 07
32	Received	10:55:37 474	Data frame	Standard frame	0x000000f6	8	00 01 02 03 04 05 06 07
33	Received	10:55:38 473	Data frame	Standard frame	0x000000f6	8	00 01 02 03 04 05 06 07
34	Received	10:55:39 474	Data frame	Standard frame	0x000000f6	8	00 01 02 03 04 05 06 07
35	Received	10:55:40 475	Data frame	Standard frame	0x000000f6	8	00 01 02 03 04 05 06 07

Must be Configured

Com Configure: Com Port: COM12; Baud Rate: 2000000

CAN Configure: CAN Mode: Normal; Frame Type: Standard Frame; CAN Baud Rate: 50K; SYNC_SEG: CAN_SJW_1tq; BS1: CAN_BS1_6tq; BS2: CAN_BS2_5tq; Prescale(Dec): 60 (<1024)

Protocol: Fixed 20 bytes [✓ Variable length]

Filter ID(Hex) 0x: 00000000; Mask ID(Hex) 0x: 00000000

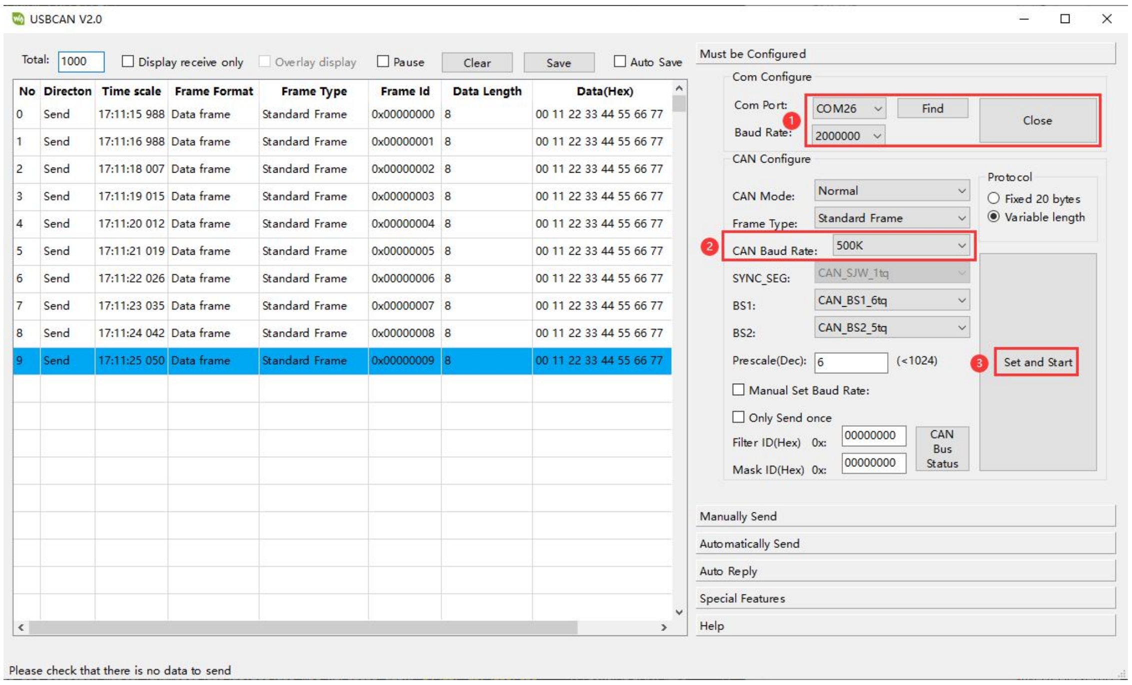
COM Open OK

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.

Connect the USB-CAN-A to the computer, open USB-CAN-A-Tool-2.0, 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.


USBCAN V2.0

Total: 1000

No	Directon	Time scale	Frame Format	Frame Type	Frame Id	Data Length	Data(Hex)
0	Send	17:11:15 988	Data frame	Standard Frame	0x00000000	8	00 11 22 33 44 55 66 77
1	Send	17:11:16 988	Data frame	Standard Frame	0x00000001	8	00 11 22 33 44 55 66 77
2	Send	17:11:18 007	Data frame	Standard Frame	0x00000002	8	00 11 22 33 44 55 66 77
3	Send	17:11:19 015	Data frame	Standard Frame	0x00000003	8	00 11 22 33 44 55 66 77
4	Send	17:11:20 012	Data frame	Standard Frame	0x00000004	8	00 11 22 33 44 55 66 77
5	Send	17:11:21 019	Data frame	Standard Frame	0x00000005	8	00 11 22 33 44 55 66 77
6	Send	17:11:22 026	Data frame	Standard Frame	0x00000006	8	00 11 22 33 44 55 66 77
7	Send	17:11:23 035	Data frame	Standard Frame	0x00000007	8	00 11 22 33 44 55 66 77
8	Send	17:11:24 042	Data frame	Standard Frame	0x00000008	8	00 11 22 33 44 55 66 77
9	Send	17:11:25 050	Data frame	Standard Frame	0x00000009	8	00 11 22 33 44 55 66 77

Must be Configured

Com Configure: Com Port: COM26; Baud Rate: 2000000

CAN Configure: CAN Mode: Normal; Frame Type: Standard Frame; CAN Baud Rate: 500K; SYNC_SEG: CAN_SJW_1tq; BS1: CAN_BS1_6tq; BS2: CAN_BS2_5tq; Prescale(Dec): 6 (<1024)

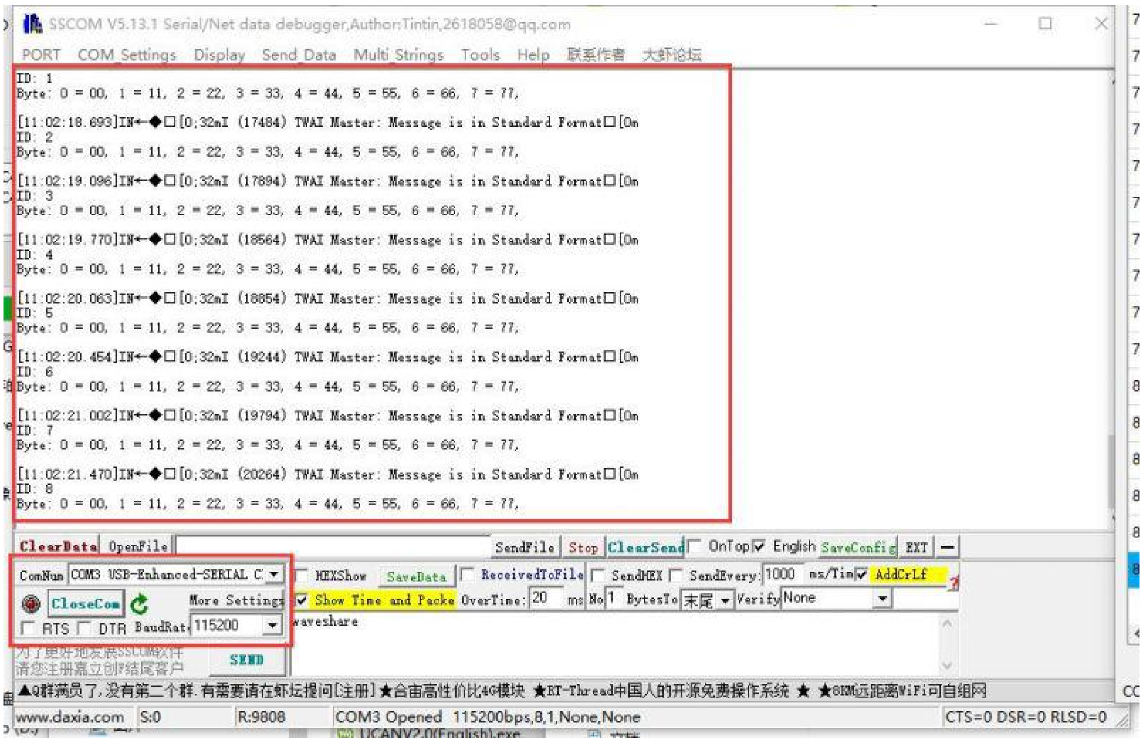
Protocol: Fixed 20 bytes [✓ Variable length]

Filter ID(Hex) 0x: 00000000

Mask ID(Hex) 0x: 00000000

Please check that there is no data to send

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.


SSCOM V5.13.1 Serial/Net data debugger,Author:Tintin,2618058@qq.com

```
ID: 1
Byte: 0 = 00, 1 = 11, 2 = 22, 3 = 33, 4 = 44, 5 = 55, 6 = 66, 7 = 77,

[11:02:18.693]IN←◆□[0;32mI (17484) TWAI Master: Message is in Standard Format□[0m
ID: 2
Byte: 0 = 00, 1 = 11, 2 = 22, 3 = 33, 4 = 44, 5 = 55, 6 = 66, 7 = 77,

[11:02:19.096]IN←◆□[0;32mI (17894) TWAI Master: Message is in Standard Format□[0m
ID: 3
Byte: 0 = 00, 1 = 11, 2 = 22, 3 = 33, 4 = 44, 5 = 55, 6 = 66, 7 = 77,

[11:02:19.770]IN←◆□[0;32mI (18564) TWAI Master: Message is in Standard Format□[0m
ID: 4
Byte: 0 = 00, 1 = 11, 2 = 22, 3 = 33, 4 = 44, 5 = 55, 6 = 66, 7 = 77,

[11:02:20.063]IN←◆□[0;32mI (18854) TWAI Master: Message is in Standard Format□[0m
ID: 5
Byte: 0 = 00, 1 = 11, 2 = 22, 3 = 33, 4 = 44, 5 = 55, 6 = 66, 7 = 77,

[11:02:20.454]IN←◆□[0;32mI (19244) TWAI Master: Message is in Standard Format□[0m
ID: 6
Byte: 0 = 00, 1 = 11, 2 = 22, 3 = 33, 4 = 44, 5 = 55, 6 = 66, 7 = 77,

[11:02:21.002]IN←◆□[0;32mI (19794) TWAI Master: Message is in Standard Format□[0m
ID: 7
Byte: 0 = 00, 1 = 11, 2 = 22, 3 = 33, 4 = 44, 5 = 55, 6 = 66, 7 = 77,

[11:02:21.470]IN←◆□[0;32mI (20264) TWAI Master: Message is in Standard Format□[0m
ID: 8
Byte: 0 = 00, 1 = 11, 2 = 22, 3 = 33, 4 = 44, 5 = 55, 6 = 66, 7 = 77,
```

ComNum: COM3 USB-Enhanced-SERIAL C; BaudRat: 115200; SendEvery: 1000 ms/Tim; OverTime: 20 ms; No 1; BytesTo 末尾; Verify None

waveshare

▲Q群满员了,没有第二个群.有需要请在虾坛提问[注册]★合宙高性价比4G模块 ★RT-Thread中国人的开源免费操作系统 ★ ★8KM远距离WiFi可自组网

www.daxia.com | S:0 | R:9808 | COM3 Opened 115200bps,8,1,None,None | CTS=0 DSR=0 RLSD=0

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.
For GT911 driver, you can refer to this link.

Resource

Document

ESP32 Arduino Core's documentation
arduino-esp32
ESP-IDF

Demo

ESP32-S3-Touch-LCD-4.3B_libraries
Sample demo

Software

Sscom5.13.1
Arduino IDE
USB-CAN-A_TOOL_1.2
USB-CAN-A_TOOL_2.0
ESP32_S3_flash_download_tool

Datasheet

ESP32-S3 Wroom Datasheet
CH343 Datasheet
TJA1051 Datasheet
GT911 datasheet
ST7262 Datasheet
CH422G Datasheet

FAQ

Question1: ESP32-S3-Touch-LCD-4.3B CAN reception failure?

Answer1:

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.

Question2: ESP32-S3-Touch-LCD-4.3B shows no response after uploading an Arduino demo for RGB screen displaying?

Answer2:

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).

Question3: ESP32-S3-Touch-LCD-4.3B fails to compile an Arduino demo for the RGB screen and shows errors?

Answer3:

Check if the "ESP32-S3-Touch-LCD-4.3B-libraries" library is installed. Please refer to installation steps.

Question4: 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.

Question5: 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
2) Run and burn original program

Question6: Why does it show "fatal error:esp_ memory_ utils.h:No such file or directory" when compiling the example with Arduino IDE?

Answer:

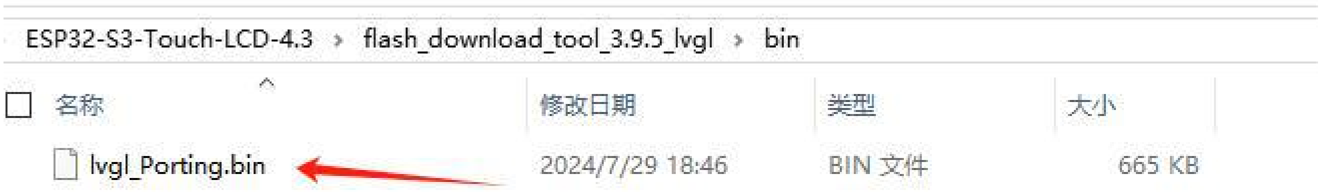
To solve this problem, please install the Arduino esp32 v3.0.2-h.

Question7: Example: lvgl_Porting; Description: It not run. If pull out the

Touch FPC and put in . It going on.

Answer:

Please try the bin file in the attachment.
https://files.waveshare.com/wiki/ESP32-S3-Touch-LCD-4.3B/flash_download_tool_3.9.5_lvgl.zip


ESP32-S3-Touch-LCD-4.3 › flash_download_tool_3.9.5_lvgl › bin

名称	修改日期	类型	大小
lvgl_Porting.bin	2024/7/29 18:46	BIN 文件	665 KB

2) If the problem still cannot be solved, please download this example and try compiling it:
https://files.waveshare.com/wiki/ESP32-S3-Touch-LCD-4.3B/ESP32-S3-Touch-LCD-4.3B_Code.zip
https://files.waveshare.com/wiki/ESP32-S3-Touch-LCD-4.3/demo/lvgl_Porting.zip