ES3C28P&ES3N28P 2.8inch IPS ESP-IDF Demo Instructions

CONTENTS

1. Software and hardware platform description

Module: 2.8-inch IPS ESP32-S3 display module with 240x320 resolution and ILI9341V screen driver IC.

Module master: ESP32-S3 chip, the highest main frequency 240MHz, support 2.4G WIFI+ Bluetooth.

ESP-IDF version: 5.4.1

LVGL version: 8.4.0.

2. Pin allocation instructions

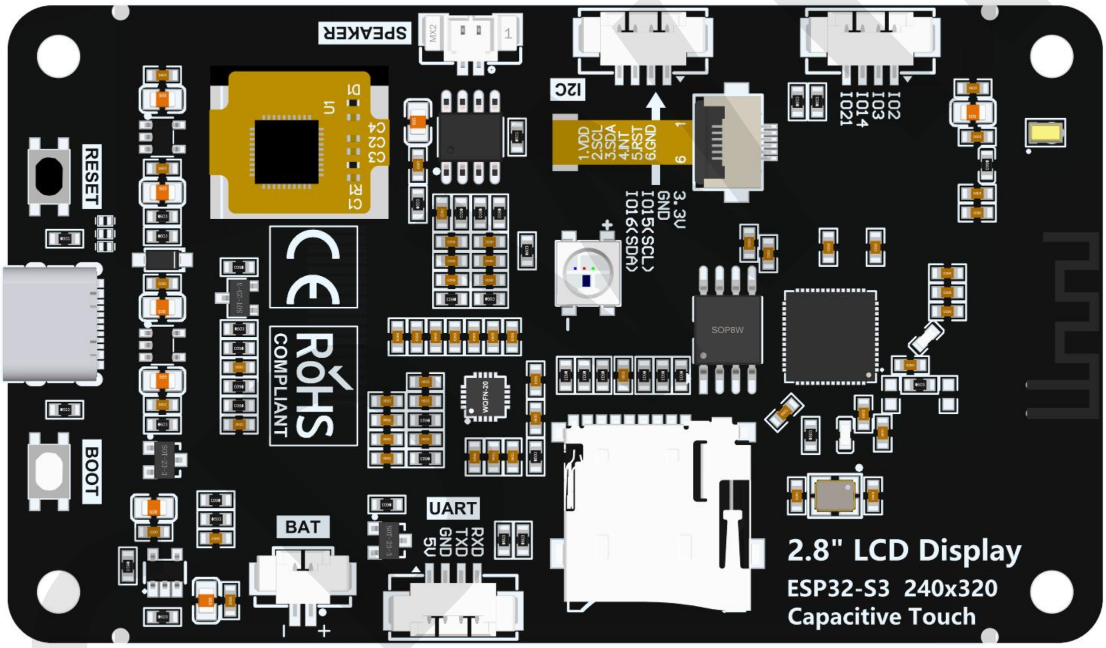


Figure 2.1 Rear view of 2.8-inch ESP32-S3 display module

The main controller of the 2.8-inch ESP32-S3 display module is ESP32-S3 chip, and the GPIO allocation for its onboard peripherals is shown in the table below:

ESP32-S3 chip pin allocation instructions			
On board device	On board device pins	ESP32-S3 connection pin	description
LCD	TFT_CS	IO10	LCD screen chip selection control signal, low level effective
	TFT_RS	IO46	LCD screen command/data selection control signal.High level: data, low level: command

	TFT_SCK	IO12	LCD SPI bus clock signal
	TFT_MOSI	IO11	LCD SPI bus writes data signals
	TFT_MISO	IO13	LCD SPI bus reading data signal
	TFT_RST	CHIP_PU	LCD screen reset control signal, low level reset (shared reset pin with ESP32-S3 main control)
	TFT_BL	IO45	LCD screen backlight control signal (high level lights up the backlight, low level turns off the backlight)
CTP	TP_SDA	IO16	Capacitive touch screen I2C bus data signal
	TP_SCL	IO15	Capacitive touch screen I2C bus clock signal
	TP_RST	IO18	Capacitive touch screen reset control signal, low level reset
	TP_INT	IO17	Capacitive touch screen interrupts the input signal, and when a touch event occurs, it inputs a low level
LED	RGB_INT	IO42	Single line RGB three color LED light, which can light up the internal red, green, and blue light beads according to different signals
SDCARD	SD_CLK	IO38	SD card SDIO bus clock signal
	SD_CMD	IO40	SD card SDIO bus command signal
	SD_D0	IO39	SD card SDIO bus data signal (DATA0~DATA3 four data lines)
	SD_D1	IO41	
	SD_D2	IO48	
	SD_D3	IO47	
BATTERY	BAT_ADC	IO9	Battery voltage ADC value acquisition input signal
Audio	Audio_EN	IO1	Audio output enable signal, low-level enable, high-level disable
	I2S_MCK	IO4	Audio I2S bus master clock signal
	I2S_SCK	IO5	Audio I2S bus bit clock signal
	I2S_DO	IO6	Audio I2S bus bit output data signal
	I2S_LRC	IO7	Audio I2S bus left and right channel

			selection signals. High level: right channel; Low level: Left channel
	I2S_DI	IO8	Audio I2S bus bit input data signal
KEY	BOOT_KEY	IO0	Download mode selection button (press and hold the button to power on, then release it to enter download mode)
	RESET_KEY	EN	ESP32-23E reset button, low level reset (shared with LCD screen reset)
USB	USB_N	IO19	USB bus differential signal data line negative pole
	USB_P	IO20	Positive pole of USB bus differential signal data line
Serial Port	TX0	IO43	ESP32-S3 serial port 0 sending signal
	RX0	IO44	ESP32-S3 serial port 0 receiving signal
POWER	TYPE-C_POWER	/	Type-C power interface, connected to 5V voltage

Table 2.1 Pin allocation instructions for ESP32-S3 onboard peripherals

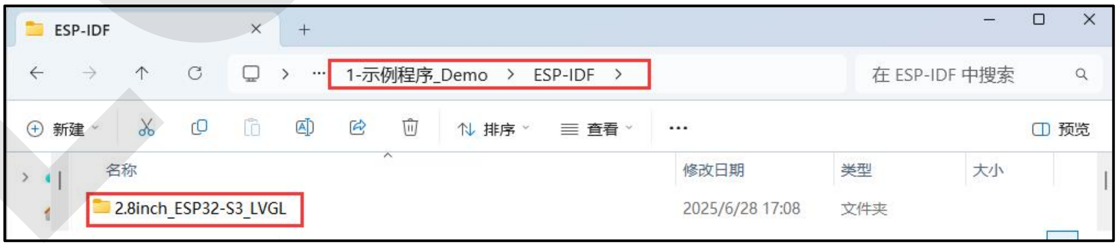
3. Instructions for the example program

3.1. Set up ESP32 IDF development environment

For detailed instructions on setting up the ESP32 IDF development environment, please refer to the "**Building an ESP-IDF environment using VS Code**" documentation in the package.

3.2. Example Program Usage Instructions

The example program is located in the "**1-示例程序_Demo\ESP32-IDF**" directory of the package, as shown in the following figure:



名称	修改日期	类型	大小
2.8inch_ESP32-S3_LVGL	2025/6/28 17:08	文件夹	

Figure 3.1 Example Program

The example program has already been ported to LVGL and the relevant program files have been modified, so it can be used directly. For LVGL porting instructions, please refer to the "**ESP-IDF_LVGL_porting_instructions**" document in the resource package. The steps to use the example program are as follows:

A. Copy the entire folder of the sample program "**2.8inch_ESP32-S3_LVGL**" **to a path named entirely in English**. Otherwise, an error will occur during compilation due to the inability to find the path.

B. Open the VS Code software, click on "**File**" ->"**Open Folder**", as shown in the following figure

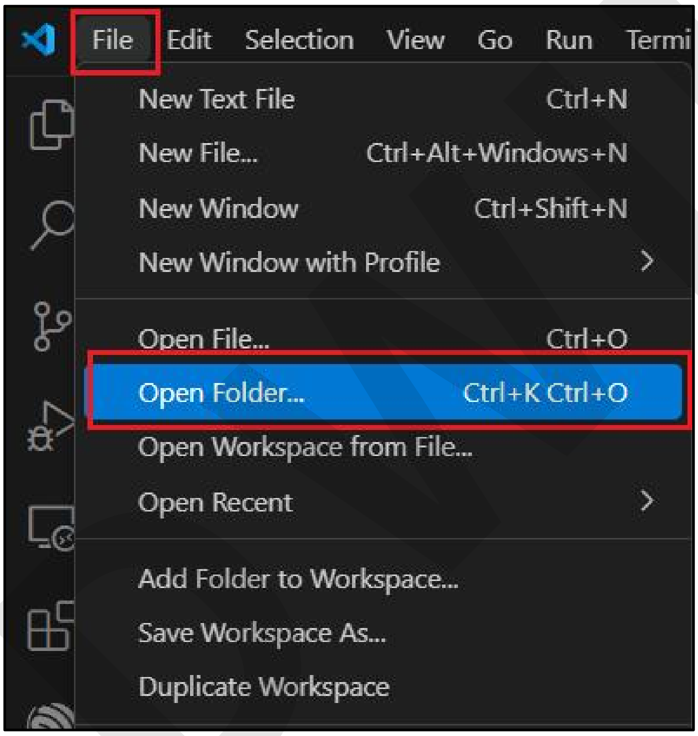


Figure 3.2 open folder

C. Find the sample program folder, click to select it, and then click the "**Select Folder**" button to open the sample program, as shown in the following figure:

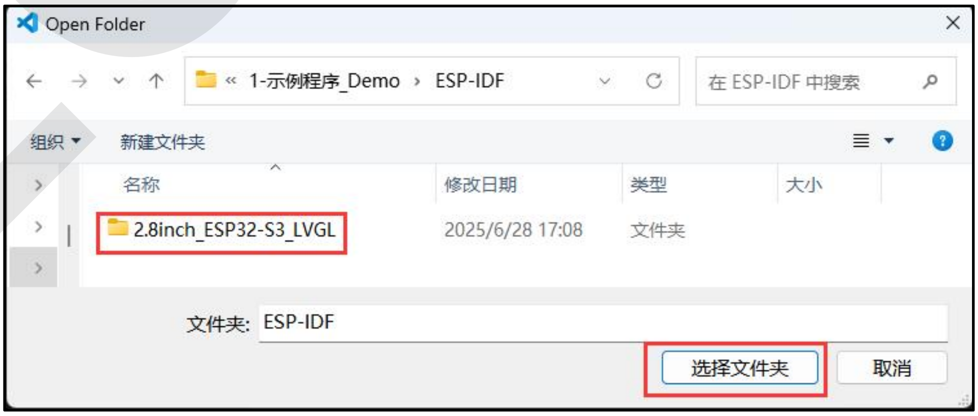


Figure 3.3 Find the sample program folder

D. Connect the ESP32 device to the computer, select the correct serial port number, chip, and download method from the bottom toolbar of VS Code, and then click the button 🔥 to compile and burn.

E. After the burning is completed, you can see the display module has displayed.