E32R32P&E32N32P 3.2inch IPS ESP-IDF Demo Instructions

CONTENTS

1. Software and hardware platform description

Module: 3.2-inch ESP32-32E display module with 240x320 resolution and ST7789 screen driver IC.

Module master: ESP32-WROOM-32E module, the highest main frequency 240MHz, support 2.4G WIFI+ Bluetooth.

ESP-IDF version: 5.3.1

LVGL version: 8.3.11.

2. Pin allocation instructions

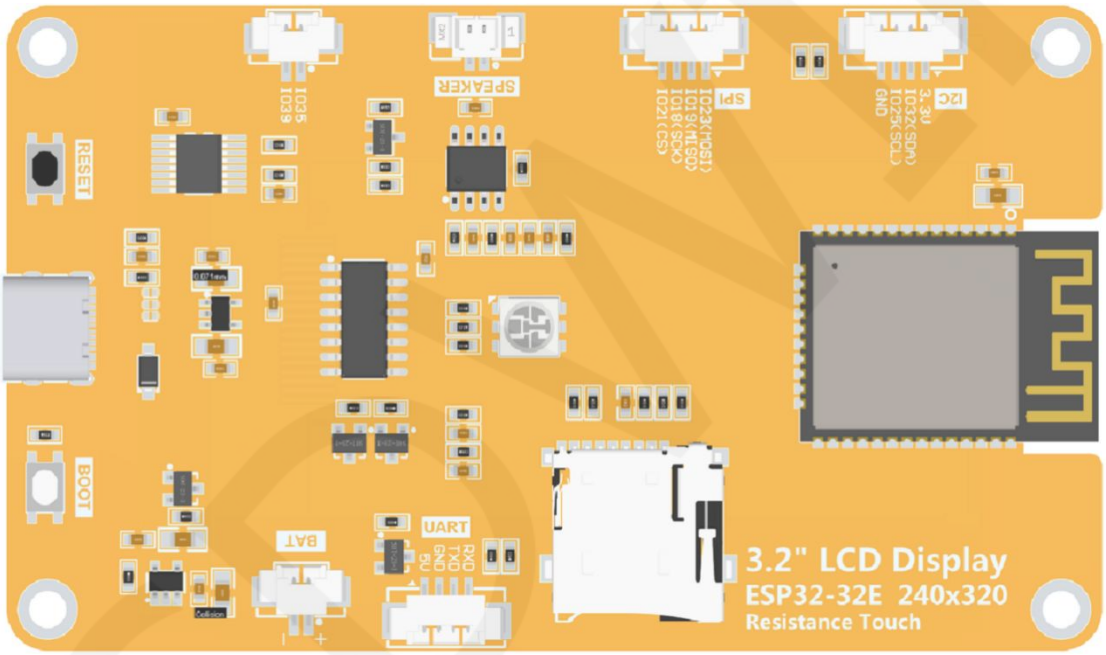

Figure 2.1 Rear view of 3.2-inch ESP32-32E display module

The main controller of the 3.2-inch ESP32 display module is ESP32-32E, and the GPIO allocation for its onboard peripherals is shown in the table below:

ESP32-32E pin allocation instructions			
On board device	**On board device pins**	**ESP32-32E connection pin**	**description**
LCD	**TFT_CS**	IO15	LCD screen chip selection control signal, low level effective
	TFT_RS	IO2	LCD screen command/data selection control signal.High level: data, low level: command

	TFT_SCK	IO14	SPI bus clock signal (shared by LCD screen and touch screen)	
	TFT_MOSI	IO13	SPI bus writes data signals (shared by LCD screen and touch screen)	
	TFT_MISO	IO12	SPI bus reading data signal (shared by LCD screen and touch screen)	
	TFT_RST	EN	LCD screen reset control signal, low level reset (shared reset pin with ESP32-32E main control)	
	TFT_BL	IO27	LCD screen backlight control signal (high level lights up the backlight, low level turns off the backlight)	
RTP	TP_SCK	IO14	SPI bus clock signal (shared by touch screen and LCD screen)	
	TP_DIN	IO13	SPI bus writes data signals (shared by touch screen and LCD screen)	
	TP_DOUT	IO12	SPI bus reading data signal (shared by touch screen and LCD screen)	
	TP_CS	IO33	Resistance touch screen chip selection control signal, low level effective	
	TP_IRQ	IO36	Resistive touch screen touch interrupt signal, when a touch is generated, input a low level to the main control	
LED	LED_RED	IO22	Red LED light	RGB tri color LED light, with a common anode, lit at low level and turned off at high level.
	LED_GREEN	IO16	Green LED light	
	LED_BLUE	IO17	Blue LED light	
SDCARD	SD_CS	IO5	SD card signal selection, low level effective	
	SD_MOSI	IO23	SD card SPI bus write data signal	
	SD_SCK	IO18	SD card SPI bus clock signal	
	SD_MISO	IO19	SD card SPI bus read data signal	
BATTERY	BAT_ADC	IO34	Battery voltage ADC value acquisition signal (input)	
Audio	Audio_ENABLE	IO4	Audio enable signal, low-level enable, high-level disable	
	Audio_DAC	IO26	Audio signal DAC output 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)
Serial Port	RX0	RXD0	ESP32-32E serial port receiving signal
	TX0	TXD0	ESP32-32E serial port sends signal
POWER	TYPE-C_POWER	/	Type-C power interface, connected to 5V voltage.

Table 2.1 Pin allocation instructions for ESP32-32E 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:

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 "**3.2inch_ESP32_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

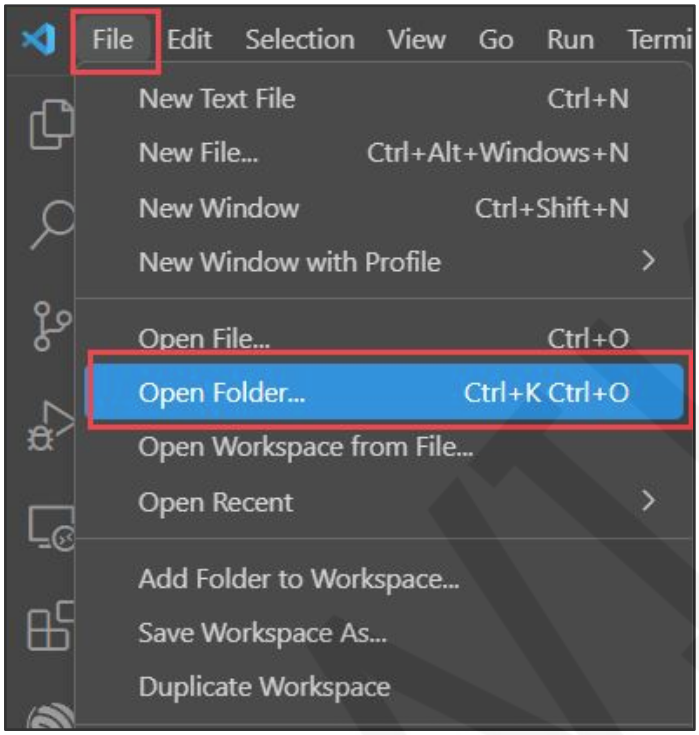

Figure 3.2 open folder

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:

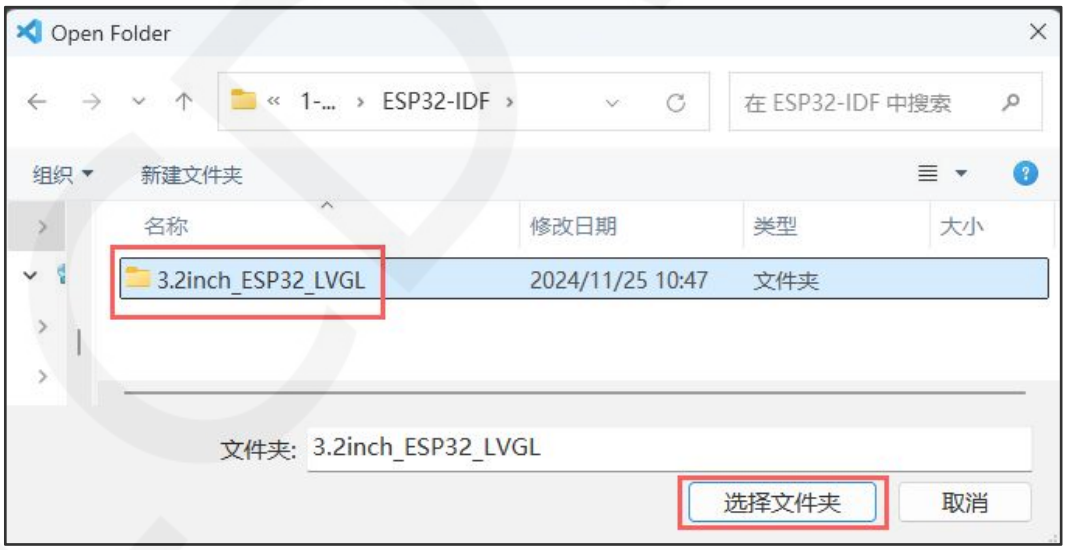

Figure 3.3 Find the sample program folder

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 that the display module has content displayed.