

E32R24P

2.4inch ESP-IDF

Demo Instructions

CONTENTS

| | |
|--|----------|
| 1. Software and hardware platform description | 3 |
| 2. Pin allocation instructions | 3 |
| 3. Instructions for using the example program | 5 |
| 3.1. Set up ESP32 ESP-IDF development environment | 5 |
| 3.2. Example Program Usage Instructions | 5 |

| | | | | |
|---------|--------------|------|---|--|
| | TFT_SCK | IO14 | LCD SPI bus clock signal | |
| | TFT_MOSI | IO13 | LCD SPI bus writes data signals | |
| | TFT_RST | EN | LCD screen reset control signal, low level reset (shared reset pin with ESP32-32E main control) | |
| | TFT_BL | IO21 | LCD screen backlight control signal (high level lights up the backlight, low level turns off the backlight) | |
| RTP | TP_SCK | IO25 | Resistance touch screen SPI bus clock signal | |
| | TP_DIN | IO32 | Resistance touch screen SPI bus writes data signals | |
| | TP_DOUT | IO39 | Resistance touch screen SPI bus reading data signal | |
| | 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 using 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 "Setting Up ESP-IDF Environment with VS Code" documentation in the resource package.

3.2. Instruction for Using the Sample Program

The sample program is located in the "1-Sample Programs_Demo\ESP32-IDF" directory of the data package, as shown in the following figure:



Figure 3.1 Example Program

This sample program has already been ported to LVGL, and the related program files have also been modified and are ready for direct use. For the LVGL porting instructions, please refer to the "ESP-IDF_LVGL Porting Instructions" document in the reference package. The steps for using the sample program are as follows:

- A: Copy the entire folder "2.4inch_ESP32_LVGL" of the sample program to the path you have named in English, otherwise, the compilation will fail due to the inability to find the path.
- B: Open the VS Code software: Click "File" -> "Open Folder", as shown in the following figure:



Figure 3.2 open the folder

C: Locate the sample program folder, select it, then click the "Select Folder" button. This will open the sample program, as shown in the following image:

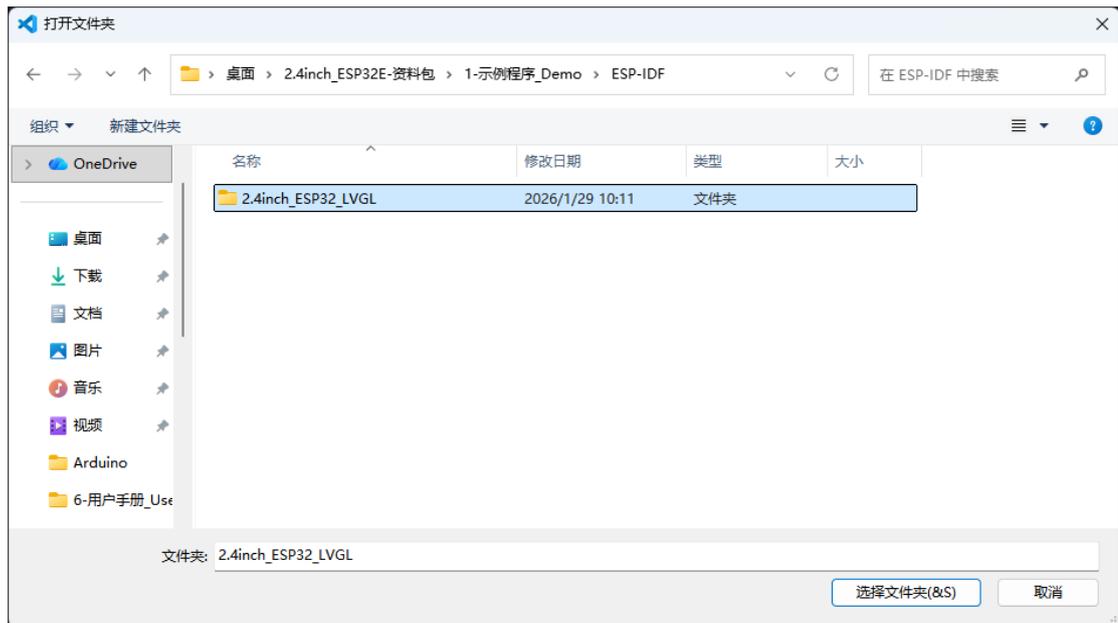


Figure 3.3 Locate the sample program folder

- D: Connect the ESP32 device to the computer. In the bottom toolbar of VS Code, select the correct serial port, chip, and download method, and then click the button  to perform compilation and burning.
- E: After the burning process is completed, you can see that the display module is showing content.