# ESP32-S3-LCD-1.47

## Usage Instructions

ESP32-S3-LCD-1.47 currently provides two development tools and frameworks, **Arduino IDE** and **ESP-IDF**, providing flexible development options, you can choose the right development tool according to your project needs and personal habits.

## Development tools

### Arduino IDE

Arduino IDE is an open source electronic prototyping platform, convenient and flexible, easy to get started. After a simple learning, you can start to develop quickly. At the same time, Arduino has a large global user community, providing an abundance of open source code, project examples and tutorials, as well as rich library resources, encapsulating complex functions, allowing developers to quickly implement various functions.

### ESP-IDF

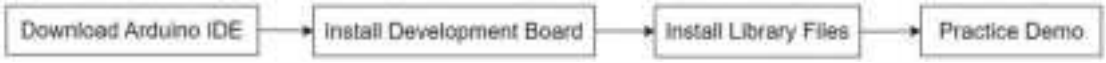
ESP-IDF, or full name Espressif IDE, is a professional development framework introduced by Espressif Technology for the ESP series chips. It is developed using the C language, including a compiler, debugger, and flashing tools, etc., and can be developed via the command lines or through an integrated development environment (such as Visual Studio Code with the Espressif IDF plugin). The plugin offers features such as code navigation, project management, and debugging.

Each of these two development approaches has its own advantages, and developers can choose according to their needs and skill levels. Arduino are suitable for beginners and non-professionals because they are easy to learn and quick to get started. ESP-IDF is a better choice for developers with a professional background or high performance requirements, as it provides more advanced development tools and greater control capabilities for the development of complex projects.

Before operating, it is recommended to browse the table of contents to quickly understand the document structure. For smooth operation, please read the FAQ carefully to understand possible problems in advance. All resources in the document are provided with hyperlinks for easy download.

# Working with Arduino

This chapter introduces setting up the Arduino environment, including the Arduino IDE, management of ESP32 boards, installation of related libraries, program compilation and downloading, as well as testing demos. It aims to help users master the development board and facilitate secondary development.




## Environment setup

### Download and install Arduino IDE

- Click to visit the official website, select the corresponding system and system bit to download.


- Run the installer and install all by default.

### Install ESP32 development board

- In order to use the ESP32-related motherboard in the Arduino IDE, the software package of the **esp32 by Espressif Systems** board must be installed
- According to **Board installation requirement**, it is generally recommended to use **Install Online**. If online installation fails, use **Install Offline**

- The **esp32 by Espressif Systems** development board comes with an offline package. Click here to download: esp32_package_3.0.2_arduino offline package

ESP32-S3-LCD-1.47 required development board installation instructions

| Board name | Board installation requirement | Version number requirement |
|---|---|---|
| esp32 by Espressif Systems | "Install Offline" / "Install Online" | ⩾3.0.2 |

## Install libraries

- When installing Arduino libraries, there are usually two ways to choose from: **Install online** and **Install offline**. **If the library installation requires offline installation, you must use the provided library file**
  For most libraries, users can easily search and install them through the online library manager of the Arduino software. However, some open-source libraries or custom libraries are not synchronized to the Arduino Library Manager, so they cannot be acquired through online searches. In this case, users can only manually install these libraries offline.
- For library installation tutorial, please refer to Arduino library manager tutorial
- ESP32-S3-LCD-1.47 library file is stored in the sample program, click here to jump: ESP32-S3-LCD-1.47 Demo

ESP32-S3-LCD-1.47 library installation description

| Library Name | Description | Version | Library Installation Requirements |
|---|---|---|---|
| LVGL | Graphical library | v8.3.10 | "Install Offline" |
| PNGdec | Decode PNG image formats | v1.0.2 | "Install Offline" |

For more learning and use of LVGL, please refer to LVGL official documentation

# Run the First Arduino Demo

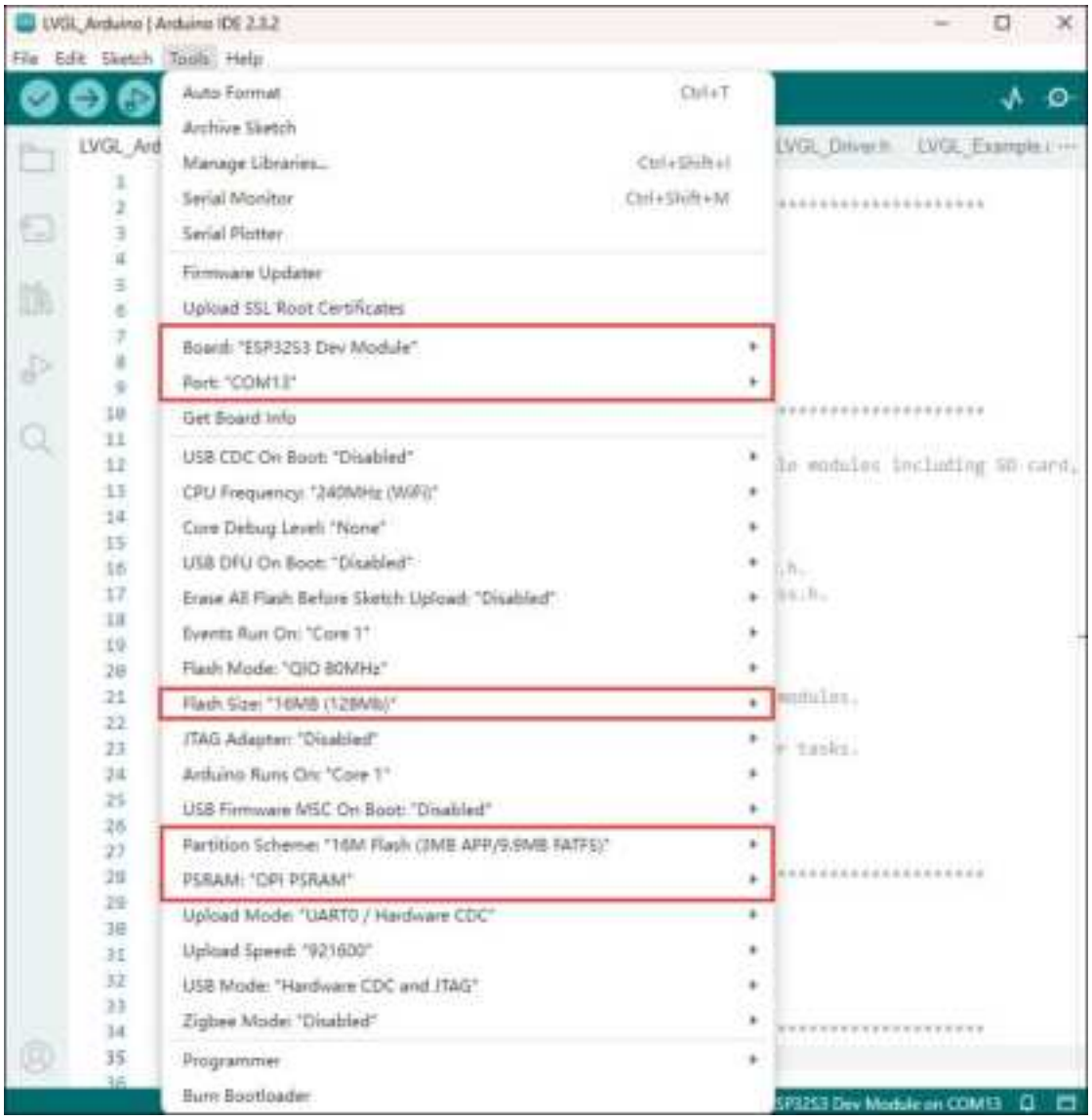
If you are just getting started with ESP32 and Arduino, and you don't know how to create, compile, flash, and run Arduino ESP32 programs, then please expand and take a look. Hope it can help you!

# Demos


ESP32-S3-LCD-1.47 demos

| Demo | Basic Description | Dependency Library |
|---|---|---|
| LVGL_Arduino | Test onboard device functionality | LVGL |
| LCD_Image | Display TF card root directory PNG file at intervals | PNGdec |

## Arduino project parameter settings




## LVGL_Arduino

**Hardware connection**

- Connect the development board to the computer

## Code analysis

---

### 1. setup()

- `Flash_test()`: Test and print the flash memory size information of the device
- `SD_Init()`: Initialize the TF card
- `LCD_Init()`: Initialize the display
- `Set_Backlight(90)`: Set the backlight brightness to 90
- `Lvgl_Init()`: Initialize the LVGL graphics library
- `Lvgl_Example1()`: Calls the specific LVGL example function
- `Wireless_Test2()`: Call the test function for wireless communication

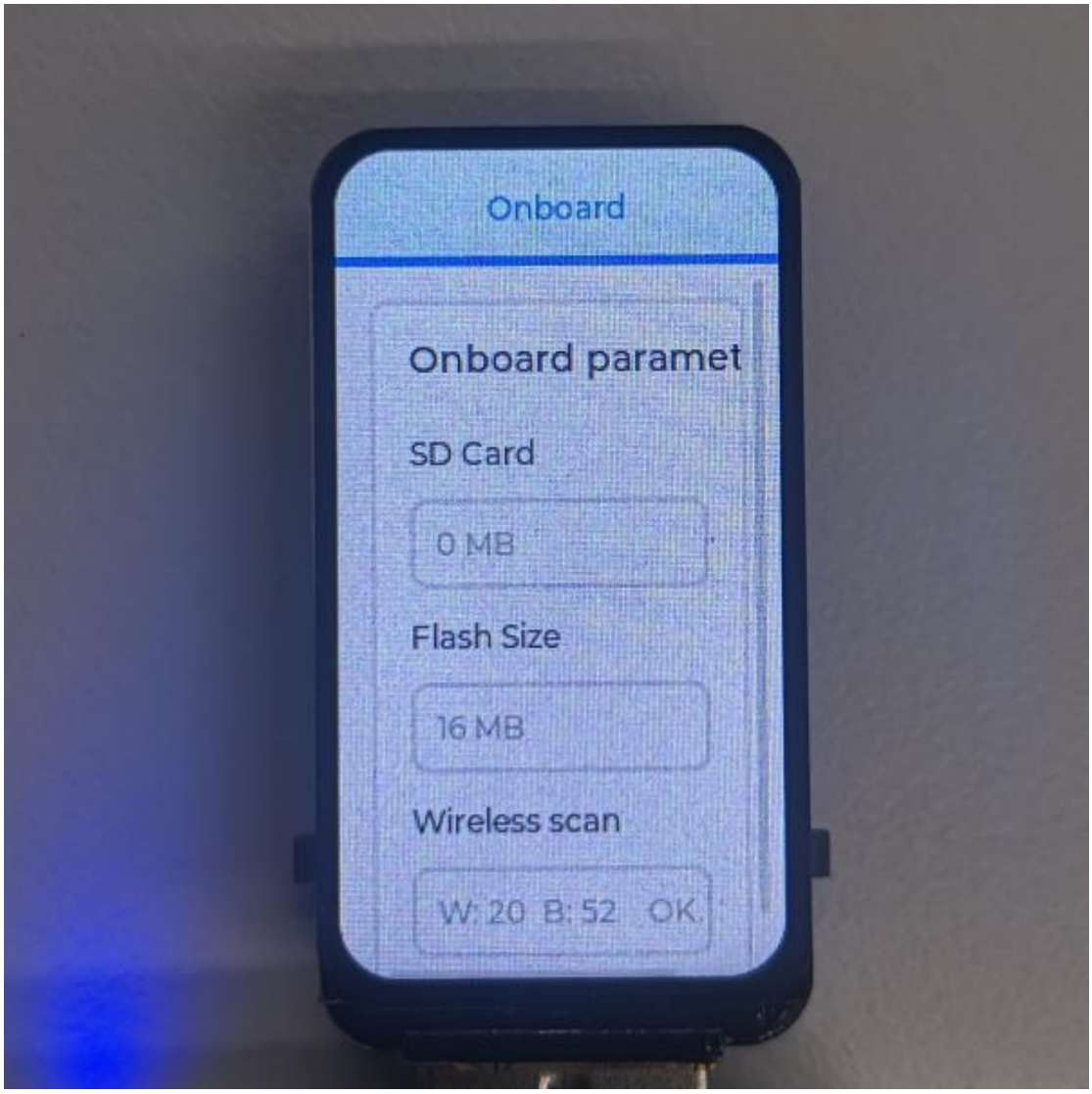
### 2. loop()

- `Timer_Loop()`: Functions that handle timer-related tasks
- `RGB_Lamp_Loop(2)`: Update the RGB light color at regular intervals

## Result demonstration

---

- LCD screen display



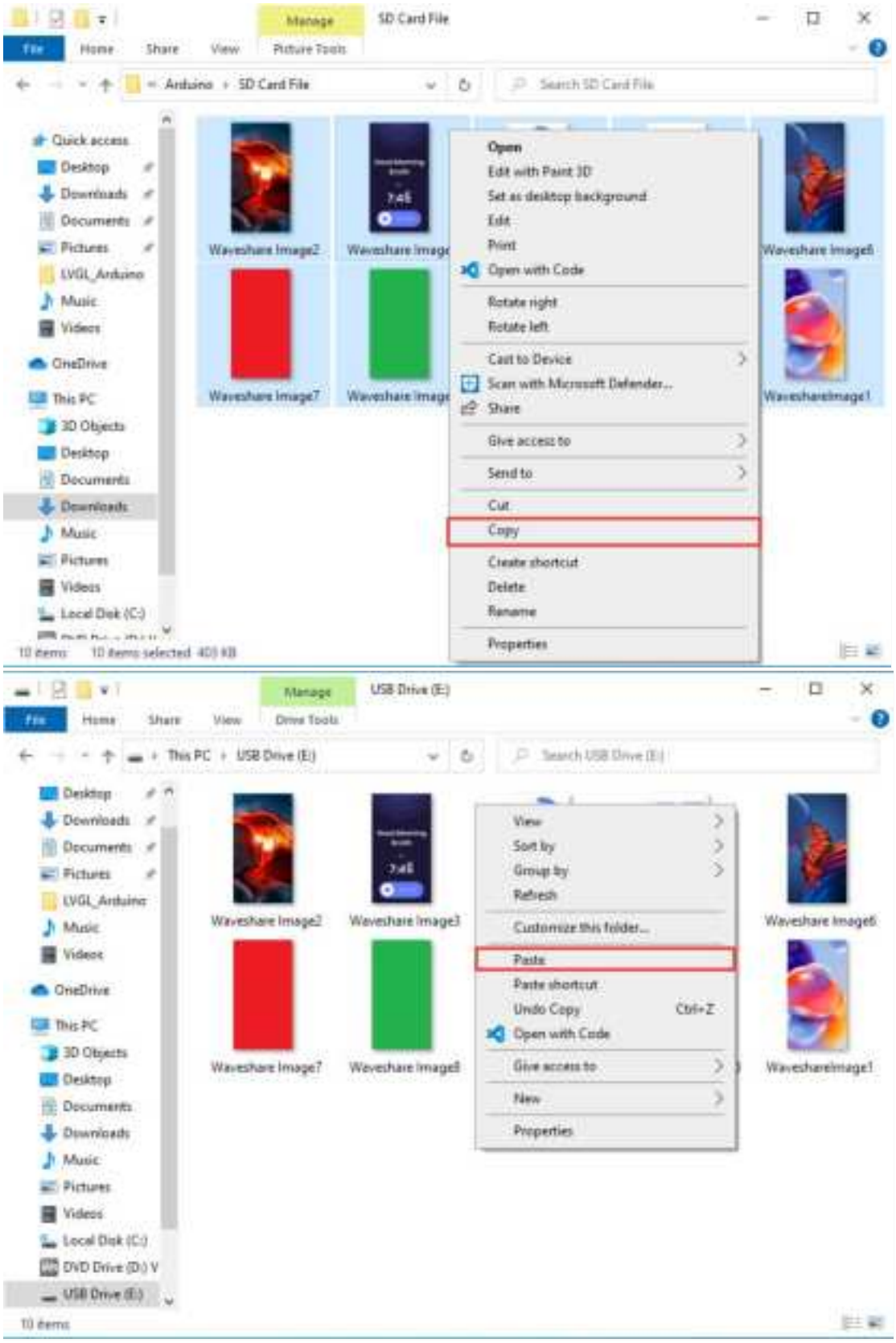

For more learning and use of LVGL, please refer to [LVGL official documentation]

## LCD_Image

## TF card preparation

---

- Add the image examples provided by Waveshare into the TF card




**Hardware connection**

- Insert the TF card containing example images into the device
- Connect the development board to the computer

**Code analysis**

---

1. **setup()**

- **`Flash_test()`**: Test and print the flash memory size information of the device
- **`SD_Init()`**: Initialize the TF card
- **`LCD_Init()`**: Initialize the display
- **`Set_Backlight(90)`**: Set the backlight brightness to 90

2. **loop()**

- **`Image_Next_Loop("/", ".png", 300)`**: Display PNG files in the TF card root directory in sequence at regular time intervals
- **`RGB_Lamp_Loop(2)`**: Update the RGB light color at regular intervals

**Result demonstration**

---

- The LCD displays PNG files in the root directory of the TF card in sequence at regular intervals

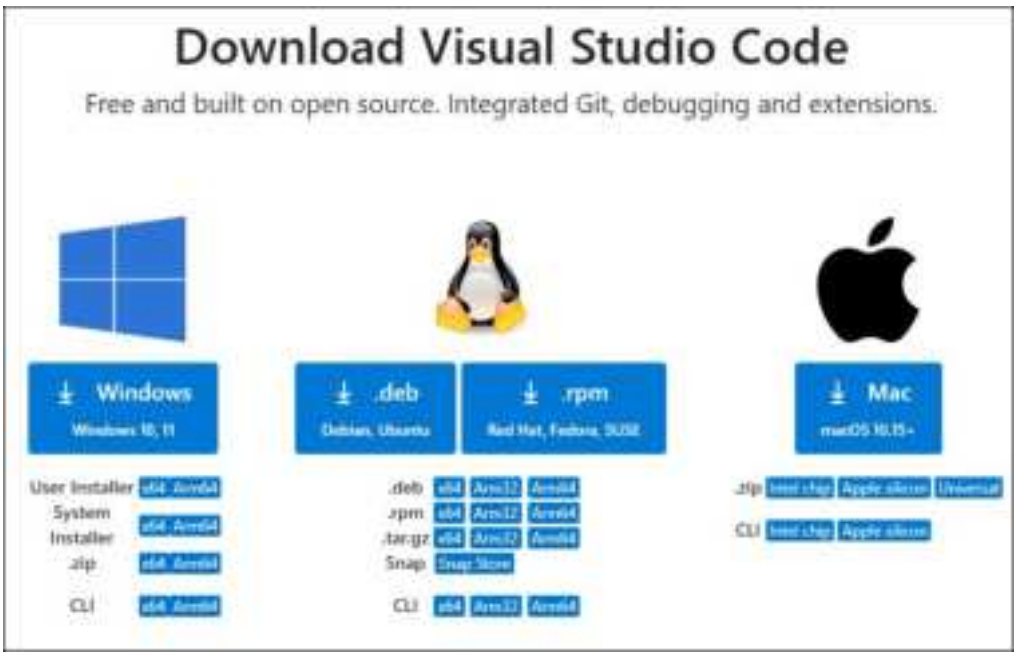

# Working with ESP-IDF

This chapter introduces setting up the ESP-IDF environment setup, including the installation of Visual Studio and the Espressif IDF plugin, program compilation, downloading, and testing of example programs, to assist users in mastering the development board and facilitating secondary development.

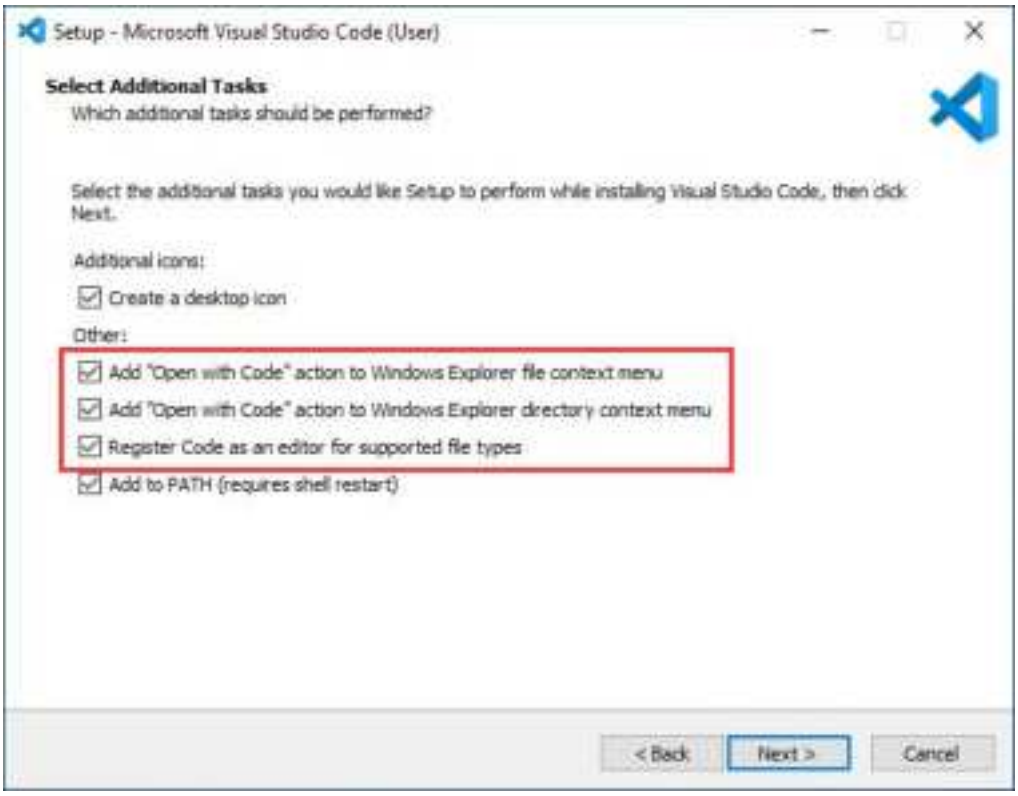
# Environment setup

## Download and install Visual Studio

- Open the download page of [VScode official website](), choose the corresponding system and system bit to download



- 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 two 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.

The environment setup is carried out on the Windows 10 system, Linux and Mac users can access ESP-IDF environment setup for reference

### Install Espressif IDF Plugin

- It is generally recommended to use **Install Online**. If online installation fails due to network factor, use **Install Offline**
- For more information about how to install the Espressif IDF plugin, see Install Espressif IDF Plugin

## Run the First ESP-IDF Demo

If you are just getting started with ESP32 and ESP-IDF, and you don't know how to create, compile, flash, and run ESP-IDF ESP32 programs, then please expand and take a look. Hope it can help you!

# Demos


ESP32-S3-LCD-1.47 demos

| Demo | Basic Description | Dependency Library |
| --- | --- | --- |
| ESP32-S3-LCD-1.47-Test | Test onboard device functionality | LVGL |

## ESP32-S3-LCD-1.47-Test

### Hardware connection

---

- Connect the development board to the computer

### Code analysis

---

1. **setup()**

- **`Wireless_Init()`**: Initialize the wireless communication module
- **`Flash_Searching()`**: Test and print the flash memory size information of the device
- **`RGB_Init()`**: Initialize RGB-related functions
- **`RGB_Example()`**: Display example functions of RGB
- **`SD_Init()`**: Initialize the TF card
- **`LCD_Init()`**: Initialize the display
- **`BK_Light(50)`**: Set the backlight brightness to 50
- **`LVGL_Init()`**: Initialize the LVGL graphics library
- **`Lvgl_Example1()`**: Calls the specific LVGL example function

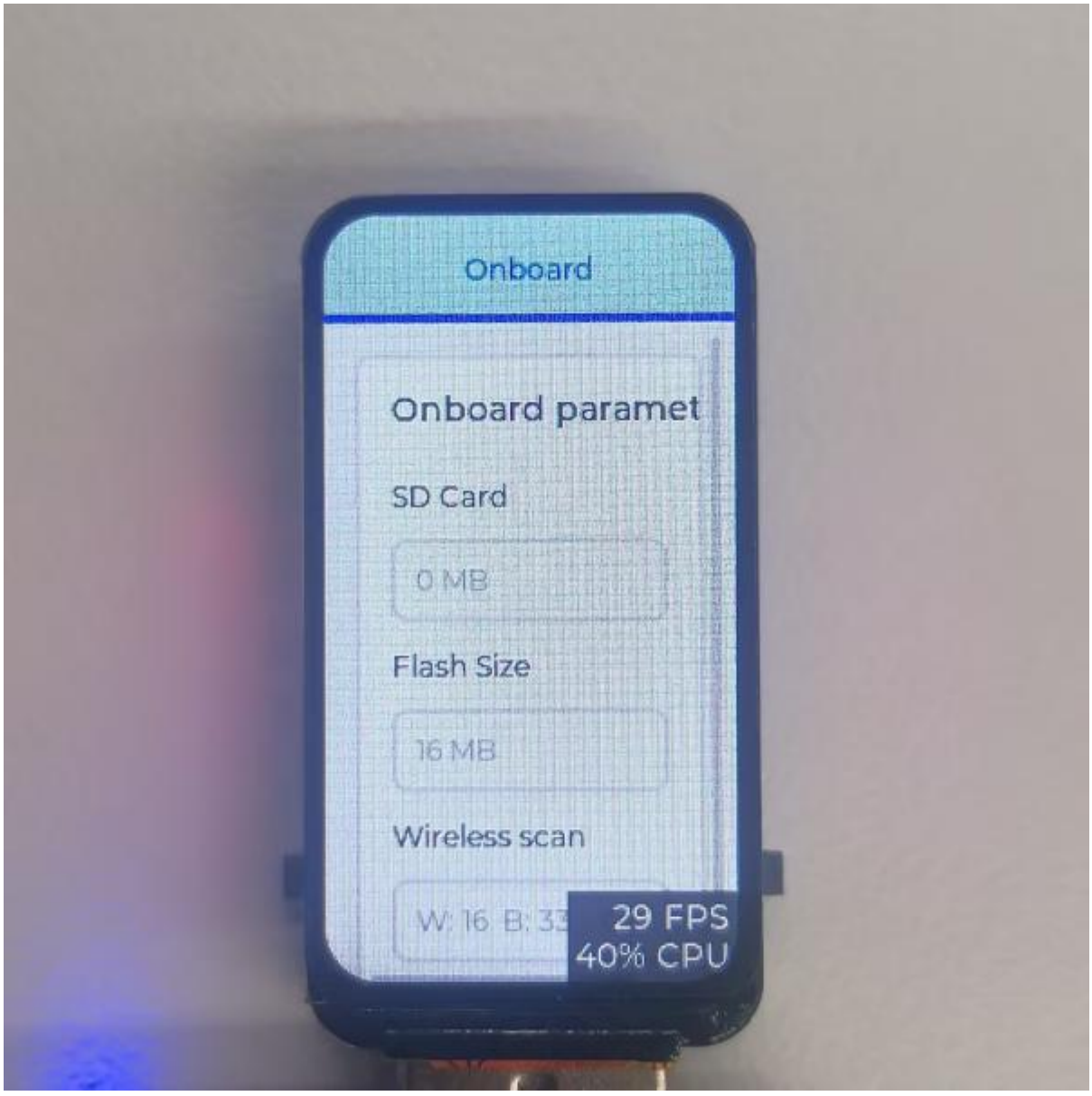
2. **while(1)**

- `vTaskDelay(pdMS_TO_TICKS(10))`: Short delay, every 10 milliseconds
- `lv_timer_handler()`: Timer handling function for LVGL, used to handle events and animations related to time

**Result demonstration**

---

- LCD displays onboard parameters:



# Flash Firmware Flashing and Erasing

---

- **The current demo provides test firmware, which can be used to test whether the onboard device functions properly by directly flashing the test firmware**
    - **bin file path:**

..\ESP32-SS-LCD-1.47-Demo\Firmware

Flash firmware flashing and erasing for reference

# Resources

## Schematic diagram

- ESP32-S3-LCD-1.47 Schematic diagram

## Demo

- ESP32-S3-LCD-1.47 Demo

## Datasheets

- 1.47inch LCD Datasheet and other files

## Software tools

### Arduino

- Arduino IDE Official download link
- esp32_package_3.0.2_arduino offline package

### VScode

- VScode official website

### Flash Download Tool

- Flash_download_tool_3.9.5_0

## Other resource links

- ESP32-Arduino official documentation
- LVGL official documentation

# FAQ

### Question: After the module downloads the demo and re-downloads it, why sometimes it can't connect to the serial port or the flashing fails?

Answer:

- Long press the BOOT button, press RESET at the same time, then release RESET, then release the BOOT button, at this time the module can enter the download mode, which can solve most of the problems that can not be downloaded.

### Question: Why does the module keep resetting and flicker when viewed the recognition status from the device manager?

Answer:

- It may be due to Flash blank and the USB port is not stable, you can long-press the BOOT button, press RESET at the same time, and then release RESET, and then release the BOOT button, at this time the module can enter the download mode to flash the firmware (demo) to solve the situation.

### Question: How to deal with the first compilation of the program being extremely slow?

Answer:

- It's normal for the first compilation to be slow, just be patient

### Question: How to handle the display "waiting for download..." on the serial port after successfully ESP-IDF flashing?

Answer:

- If there is a reset button on the development board, press the reset button; if there is no reset button, please power it on again

Question: What should I do if I can't find the AppData folder?

Answer:

- Some AppData folders are hidden by default and can be set to show.
- English system: Explorer->View->Check "Hidden items"
- Chinese system: File Explorer -> View -> Display -> Check "Hidden Items"

Question: How do I check the COM port I use?

Answer:

- Windows system:

①View through Device Manager: Press the Windows + R keys to open the "Run" dialog box; input devmgmt.msc and press Enter to open the Device Manager; expand the "Ports (COM and LPT)" section, where all COM ports and their current statuses will be listed.
②Use the command prompt to view: Open the Command Prompt (CMD), enter the "mode" command, which will display status information for all COM ports.
③Check hardware connections: If you have already connected external devices to the COM port, the device usually occupies a port number, which can be determined by checking the connected hardware.

- Linux system:

①Use the dmesg command to view: Open the terminal.
①Use the ls command to view: Enter ls /dev/ttyS* or ls /dev/ttyUSB* to list all serial port devices.
③Use the setserial command to view: Enter setserial -g /dev/ttyS* to view the configuration information of all serial port devices.

Question: Why does the program flashing fail when using a MAC device?

Answer:

- Install MAC Driver and flash again.

Question: Why is there no output after successfully burning the code with no issues?

Answer:

- Check the schematic diagram for different development boards with Type-C interfaces, and handle the output accordingly:

    - For development boards with direct USB output, printf function is supported for printing output. If you want to support output via the Serial function, you will need to enable the USB CDC On Boot feature or declare HWCDC.
    - For development boards with UART to USB conversion, both printf and Serial functions are supported for printing output, and there is no need to enable USB CDC On Boot.

## Question: How to use SquareLine Studio to design interfaces?

### Answer:

- Refer to SquareLine Studio tutorial