ESP32-C6-LCD-1.47

From Waveshare Wiki
Jump to: navigation, search

ESP32-C6-LCD-1.47

(https://www.waveshare.com/esp32-c6-lcd-1.47.htm)

1.47 inch, I2C SPI
172×320

Overview

[Expand]

Usage Instructions

ESP32-C6-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

(/wiki/File:180px-Arduino-IDE-logo.jpg)

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

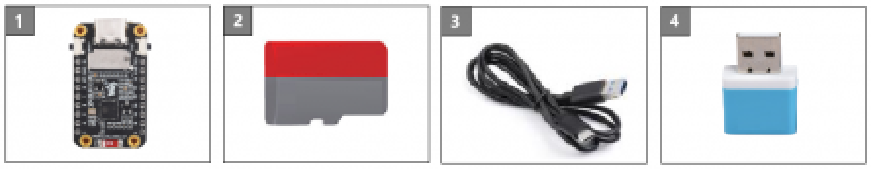
(/wiki/File:180px-ESP-IDF-logo.jpg)

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.

Components Preparation

- ESP32-C6-LCD-1.47 x1
- TF card x 1
- USB cable (Type-A to Type-C) x 1
- Card reader x1

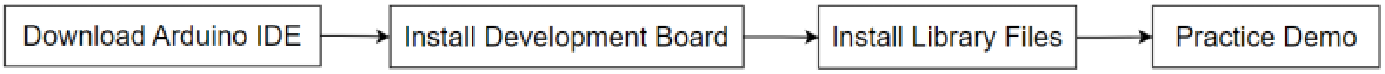


(/wiki/File:450px-ESP32-C6-LCD-1.47-uesnotes-01.png)

> 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.

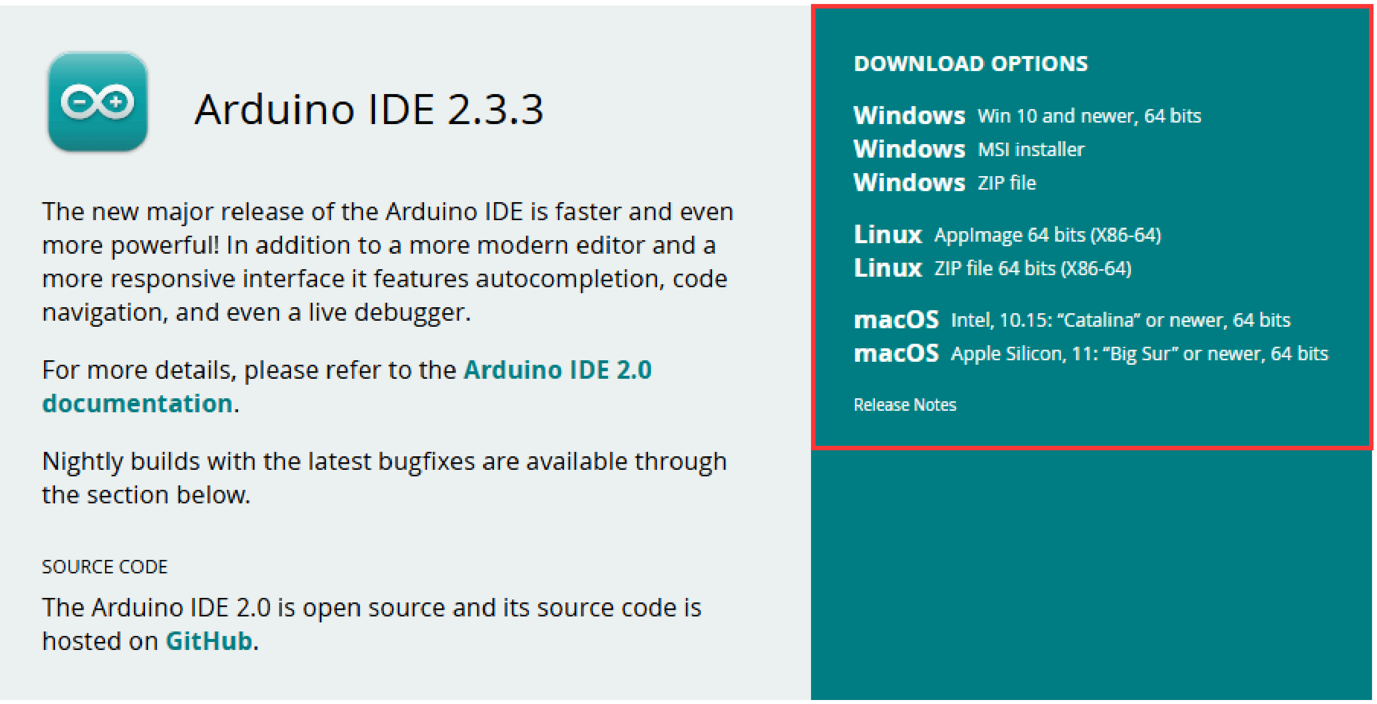


(/wiki/File:Arduino-flow-04.png)

Environment Setup

Download and install Arduino IDE

- Click to visit the Arduino official website (https://www.arduino.cc/en/software), select the corresponding system and system bit to download



Arduino IDE 2.3.3

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the **Arduino IDE 2.0 documentation**.

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on **GitHub**.

DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits
Windows MSI installer
Windows ZIP file

Linux AppImage 64 bits (X86-64)
Linux ZIP file 64 bits (X86-64)

macOS Intel, 10.15: "Catalina" or newer, 64 bits
macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits

Release Notes

(/wiki/File:ESP32-S3-AMOLED-1.91-Ar-software-01.png)

- Run the installer and install all by default

Install ESP32 development board

- Regarding ESP32-related motherboards used with the Arduino IDE, the **esp32 by Espressif Systems** library must be installed first.
- According to **Board installation requirement**, it is generally recommended to use **Install Online**. If online installation fails, use **Install Offline**
- For the installation tutorial, please refer to Arduino board manager tutorial (https://www.waveshare.com/wiki/Arduino_Board_Managers_Tutorial)
- **esp32 by Espressif Systems** development board comes with an offline package. Click here to download: esp32_package_3.0.2_arduino offline package (https://drive.google.com/file/d/1AXYTgmfmFKF6OUgjJ0BcEm2vCn9AKir2/view?usp=sharing)

- ESP32-C6-LCD-1.47 required development board installation description

Board name	Board installation requirement	Version number requirement
esp32 by Espressif Systems	"Install Offline" / "Install Online"	≥3.0.0

Install library

- 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 (https://www.waveshare.com/wiki/Arduino_Library_Manager_Tutorial)
- ESP32-C6-LCD-1.47 library file is stored in the sample program, click here to jump: ESP32-C6-LCD-1.47 Demo

- ESP32-C6-LCD-1.47 library file installation description

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

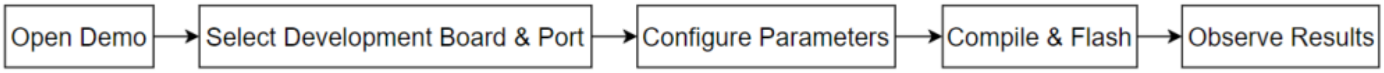
For more learning and use of LVGL, please refer to LVGL official documentation (https://docs.lvgl.io/master/intro/introduction/index.html)

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!

[Expand]

Demo

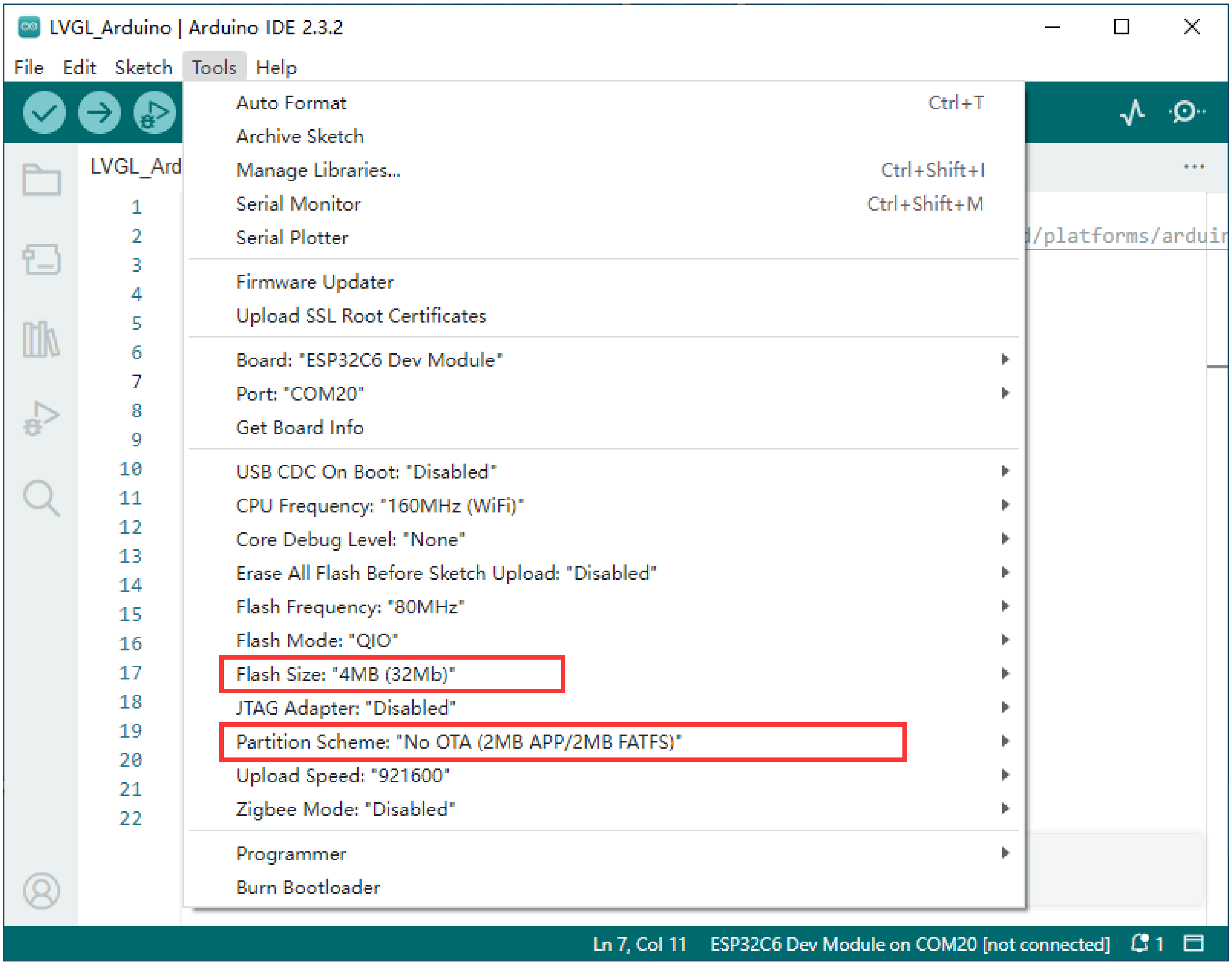


(/wiki/File:Demo-flow-01.png)

- ESP32-C6-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 setting


LVGL_Arduino | Arduino IDE 2.3.2
File Edit Sketch Tools Help
LVGL_Ard
Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
Firmware Updater
Upload SSL Root Certificates
Get Board Info
Programmer
Burn Bootloader
/platforms/arduir
Ln 7, Col 11 ESP32C6 Dev Module on COM20 [not connected] 1

Board: "ESP32C6 Dev Module"
Port: "COM20"
USB CDC On Boot: "Disabled"
CPU Frequency: "160MHz (WiFi)"
Core Debug Level: "None"
Erase All Flash Before Sketch Upload: "Disabled"
Flash Frequency: "80MHz"
Flash Mode: "QIO"
Flash Size: "4MB (32Mb)"
JTAG Adapter: "Disabled"
Partition Scheme: "No OTA (2MB APP/2MB FATFS)"
Upload Speed: "921600"
Zigbee Mode: "Disabled"

(/wiki/File:ESP32-C6-LCD-1.47_Demo_3.png)

LVGL_Arduino

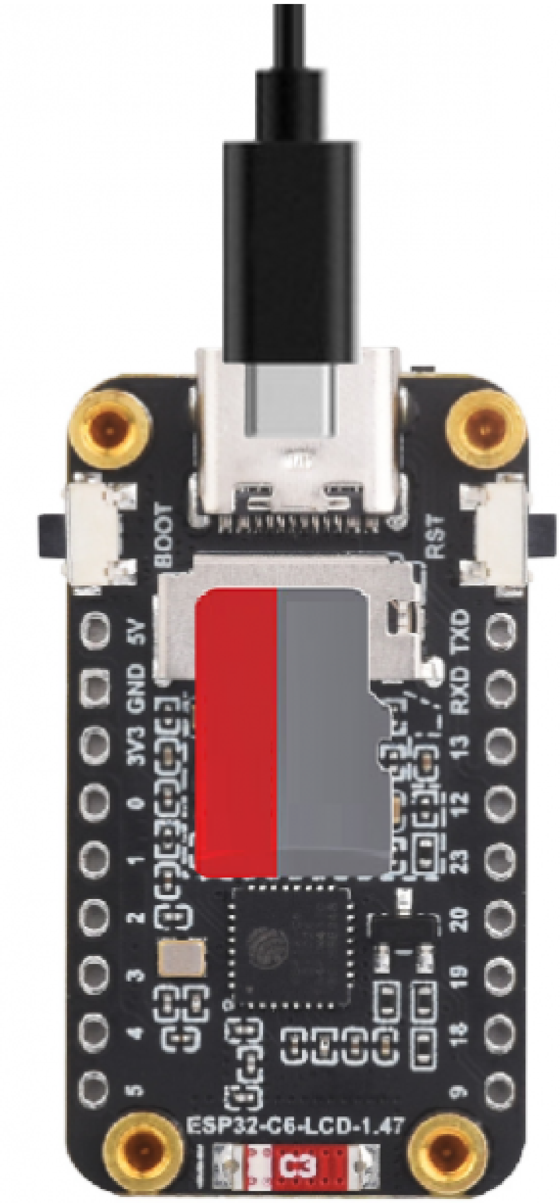
[Collapse]

Demo description

- This example demonstrates the functions of each device on the board, LCD page displays SD Card, Flash Size, Wireless scan and other parameters

Hardware connection

- Connect the development board to the computer
- Insert the TF card into the board



(/wiki/File:300px-ESP32-C6-LCD-1.47-demo-03.png)

Code analysis

- **setup()**
  - **`Flash_test()`** : Tests and prints the flash memory size information of the device
  - **`LCD_Init()`** : Initializes the display
  - **`Lvgl_Init()`** : Initializes the LVGL graphics library
  - **`SD_Init()`** : Initializes the TF card
  - **`Lvgl_Example1()`** : Calls the specific LVGL example function
  - **`Wireless_Test2()`** : Calls the test function for wireless communication

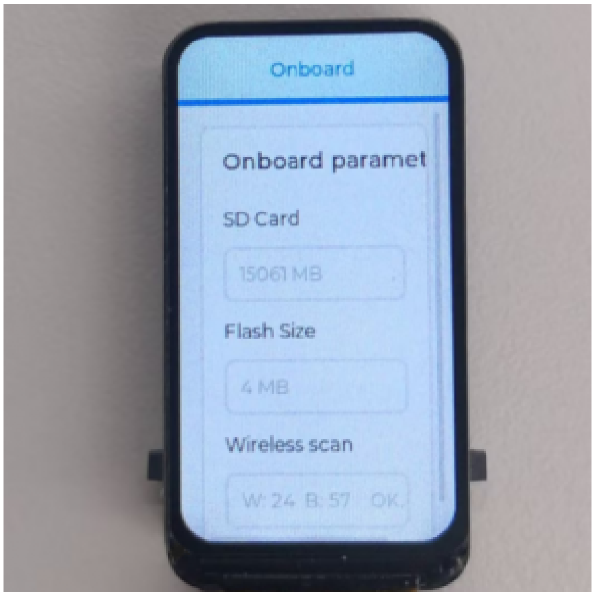
- **loop()**
    - `Timer_Loop()` : Functions that handle timer-related tasks

Demo flashing

- Select the development board ESP32S3 Dev Module and port
- Flash the demo

Result demonstration

- LCD screen display



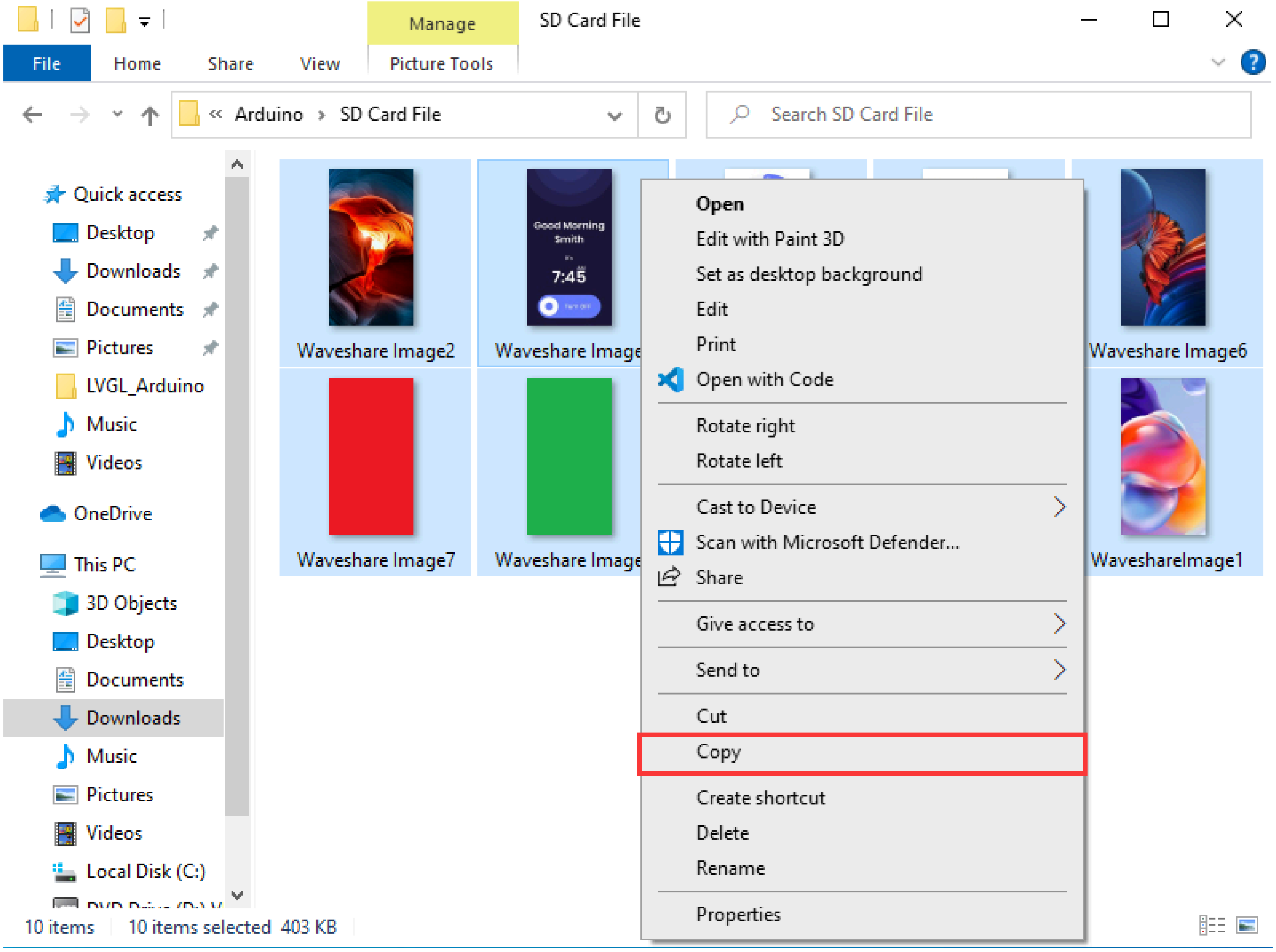
(/wiki/File:300px-ESP32-C6-LCD-1.47-demo-01.png)

> For more learning and use of LVGL, please refer to LVGL official documentation (https://docs.lvgl.io/master/intro/introduction/index.html)

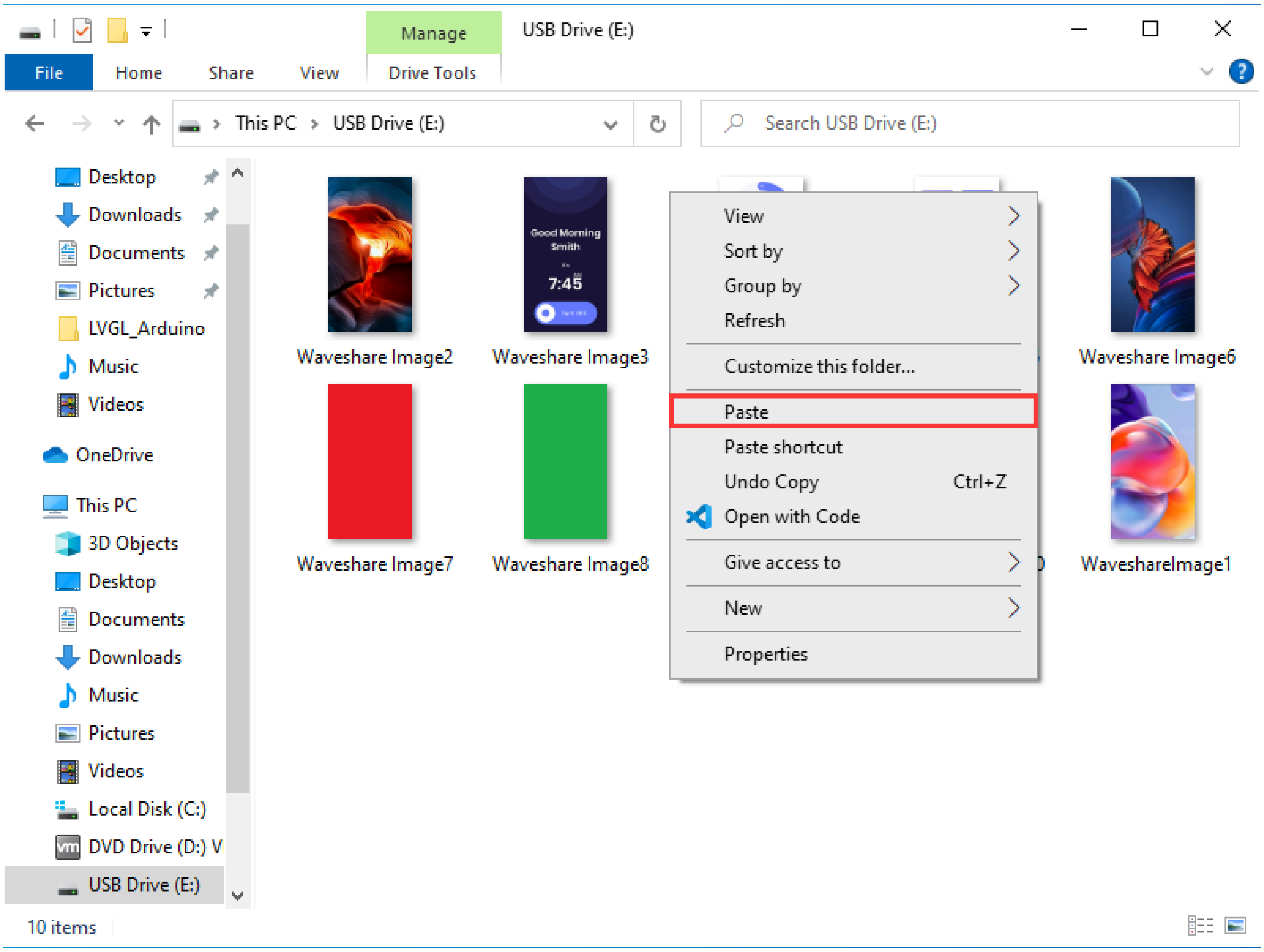
LCD_Image

TF card preparation

- Add the image examples provided by Waveshare into the TF card, and you can also add your own images (pixel ≤ 172×320)



(/wiki/File:ESP32-S3-LCD-1.47-Ar-demo-04.png)



(/wiki/File:ESP32-S3-LCD-1.47-Ar-demo-05.png)

Hardware connection

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

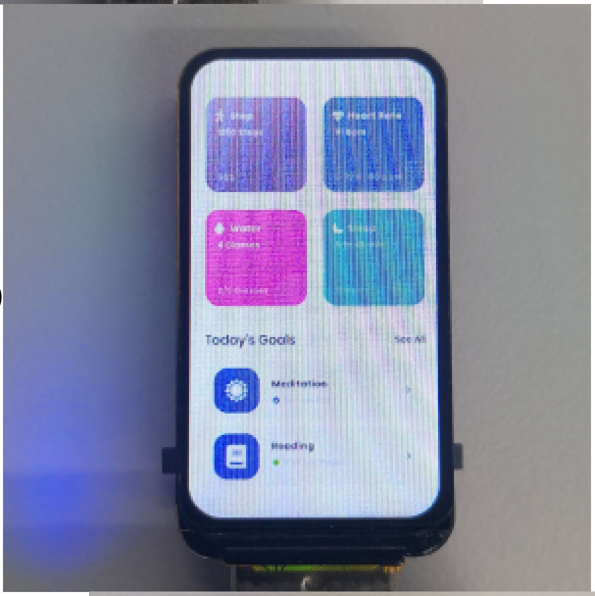
Code analysis

- **void Display_Image(const char* directory, const char* fileExtension, uint16_t ID)**
  - First, call `Search_Image(directory, fileExtension)` search for image files with the specified extension in a given directory, and store the count of found images in the global variable `Image_CNT`
  - If `Image_CNT` is not zero, one or more images have been found. Construct the full path of the image file based on whether the directory is a root directory, convert it to a C-style string for debugging purposes, and finally call the `Show_Image` function to display the image
  - If no image is found (`Image_CNT` is zero), an error message is printed indicating that a file with the specified extension was not found in the given directory

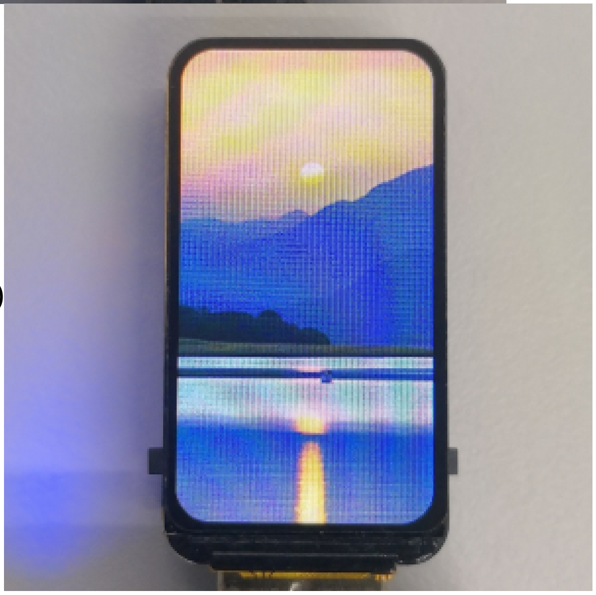
Result demonstration

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

(/wiki/File:300px-ESP32-S3-LCD-1.47-Ar-demo-06.png)



(/wiki/File:300px-ESP32-S3-LCD-1.47-Ar-demo-07.png)

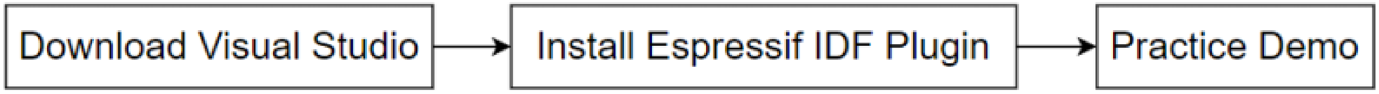


(/wiki/File:300px-ESP32-S3-LCD-1.47-

Ar-demo-08.png)

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.

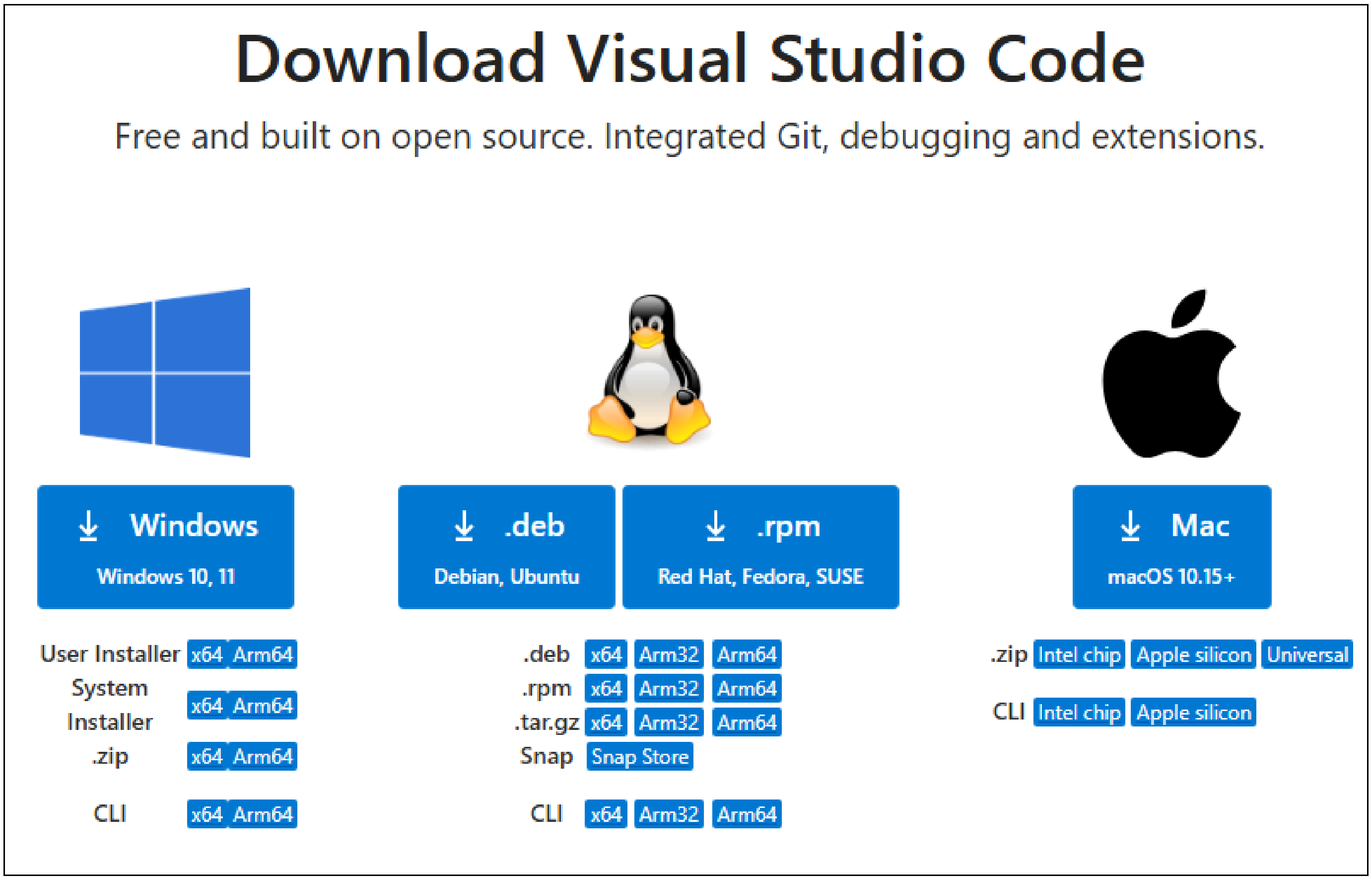


(/wiki/File:ESP-IDF-flow-01.png)

Environment Setup

Download and install Visual Studio

- Open the download page of VScode official website (https://code.visualstudio.com/download), choose the corresponding system and system bit to download

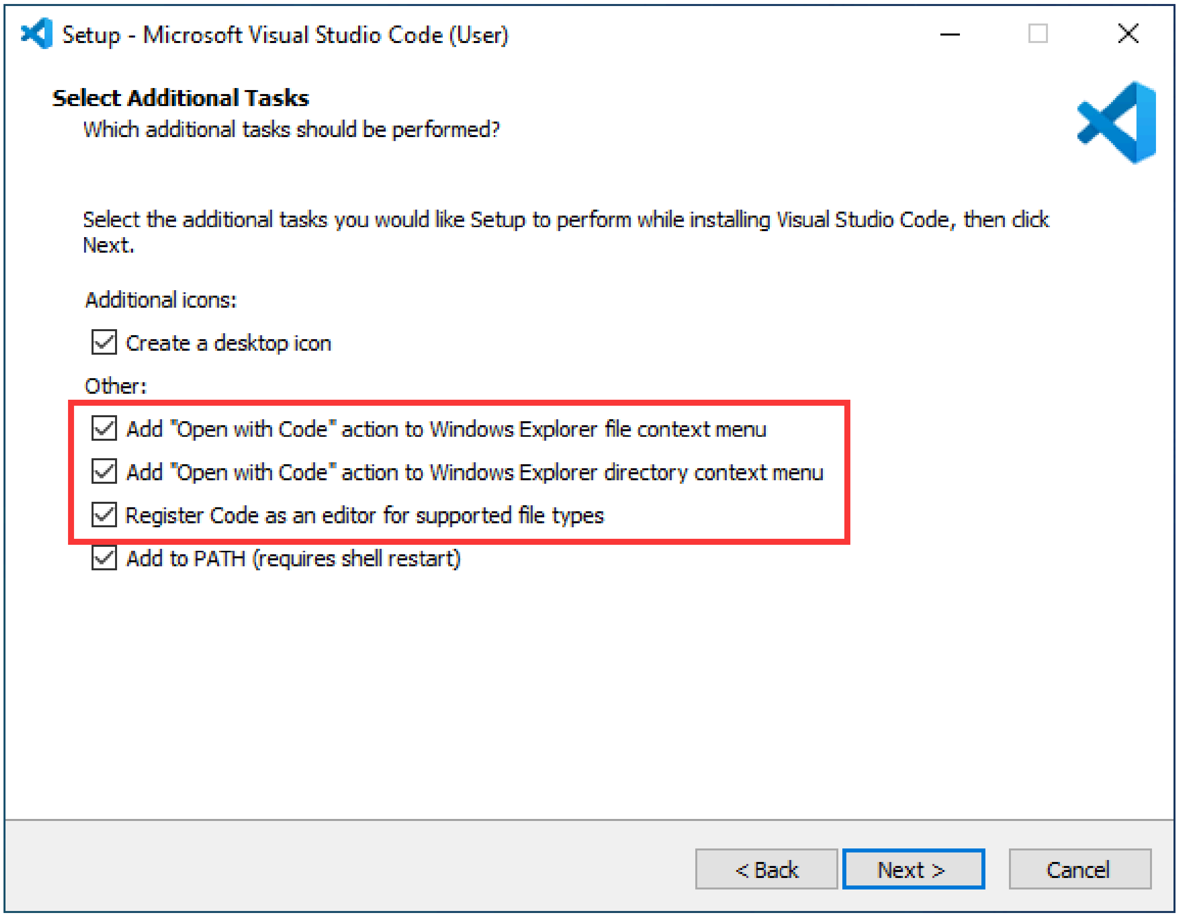


Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

(/wiki/File:ESP32-S3-AMOLED-1.91-VScode-01.png)

- 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



Setup - Microsoft Visual Studio Code (User)

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 (requires shell restart)

(/wiki/File:ESP32-S3-AMOLED-1.91-VScode-02.png)

  - 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 (https://docs.espressif.com/projects/esp-idf/en/v5.1.4/esp32s3/get-started/windows-setup.html) 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 (https://www.waveshare.com/wiki/Install_Espressif_IDF_Plugin_Tutorial)

- ESP32-C6-LCD-1.47 required Espressif IDF version description

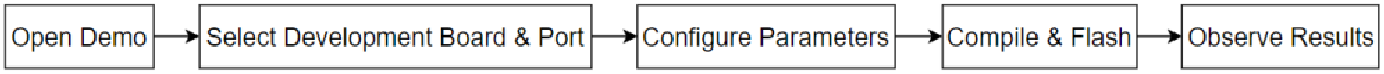
Plugin name	Plugin installation requirement	Version number requirement
Espressif IDF	"Install Offline" / "Install Online"	≥5.3.1

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!

[Expand]

Demo



(/wiki/File:Demo-flow-01.png)

- ESP32-C6-LCD-1.47 demos

Demo	Basic Description	Dependency Library
ESP32-C6-LCD-1.47-Test	Test onboard device functionality	LVGL

ESP32-C6-LCD-1.47-Test

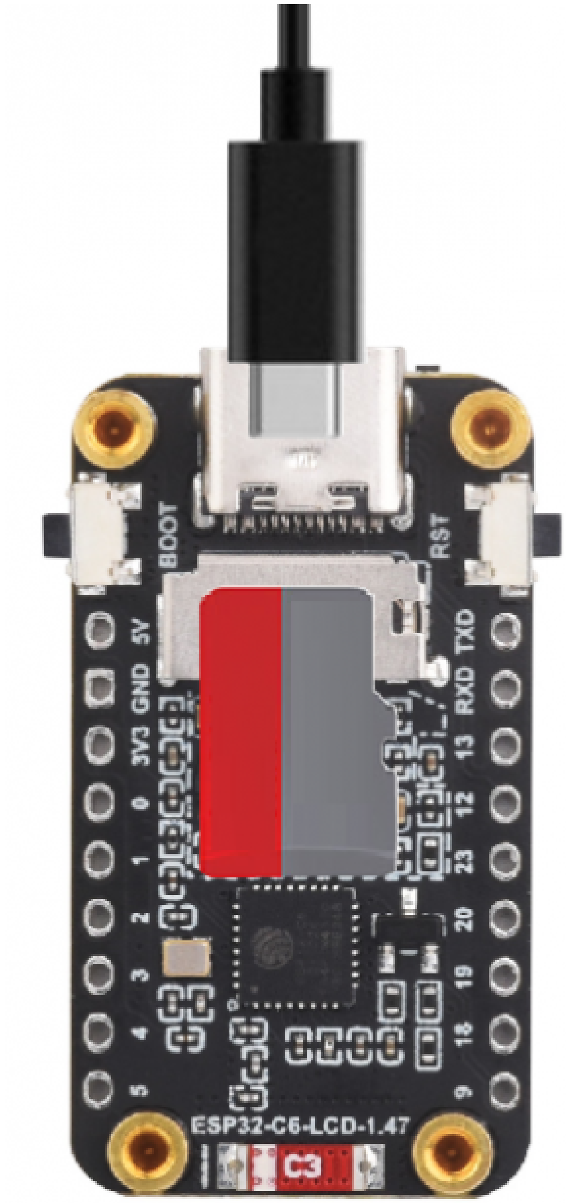
[Collapse]

Demo description

- This example demonstrates the functions of each device on the board, LCD page displays SD Card, Flash Size, Wireless scan parameters and CPU utilization

Hardware connection

- Connect the development board to the computer
- Insert the TF card into the board



(/wiki/File:300px-ESP32-C6-LCD-1.47-demo-03.png)

Code analysis

- **setup()**
  - `Wireless_Init()` : Initializes the wireless communication module
  - `Flash_Searching()` : Tests and prints the flash memory size information of the device
  - `RGB_Init()` : Initializes RGB-related functions
  - `RGB_Example()` : Displays example functions of RGB
  - `SD_Init()` : Initializes the TF card
  - `LCD_Init()` : Initializes the display
  - `BK_Light(50)` : Sets the backlight brightness to 50
  - `LVGL_Init()` : Initializes the LVGL graphics library
  - `Lvgl_Example1()` : Calls the specific LVGL example function
- **while(1)**
  - `vTaskDelay(pdMS_TO_TICKS(10))` : Short delay, every 10 milliseconds

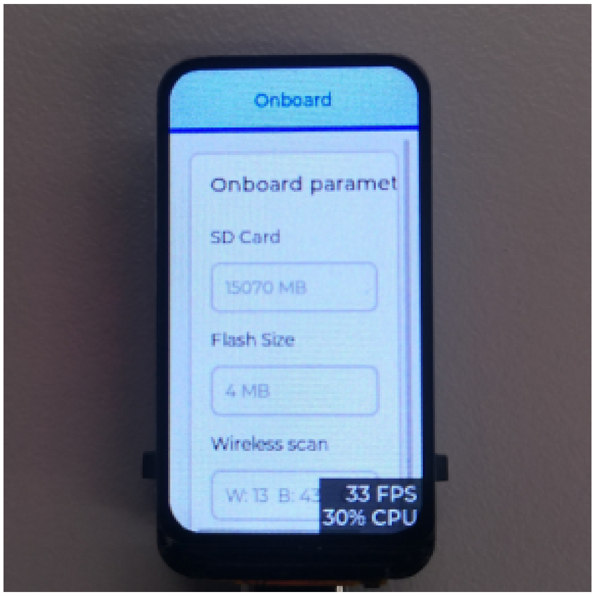
- `lv_timer_handler()` : Timer handling function for LVGL, used to handle events and animations related to time

Demo flashing

- Select the development board model ESP32C6 and port
- UART download
- Flash the demo

Result demonstration

- LCD displays onboard parameters:



(/wiki/File:300px-ESP32-C6-LCD-1.47-demo-02.png)

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-C6-LCD-1.47-Demo\Firmware
```

Flash firmware flashing and erasing (https://www.waveshare.com/wiki/Flash_Firmware_Flashin g_and_Erasing) for reference

Resources

Schematic diagram

- ESP32-C6-LCD-1.47 Schematic diagram (https://files.waveshare.com/wiki/ESP32-C6-LCD-1.47/ESP32-C6-LCD-1.47_schemetics.pdf)

Demo

- ESP32-C6-LCD-1.47 Demo (https://files.waveshare.com/wiki/ESP32-C6-LCD-1.47/ESP32-C6-LCD-1.47-Demo.zip)

Project Diagram

- ESP32-C6-LCD-1.47 3D Drawing (https://files.waveshare.com/wiki/ESP32-C6-LCD-1.47/ESP32-C6-LCD-1.47-3D_Drawing.rar)

Application Examples

- Waveshare 1.47 Boards ESP32 C6 (https://www.youtube.com/watch?v=A8wgj1fwak4)
- ESP32-C6-LCD-1.47 LVGL UI Tutorial (https://www.youtube.com/watch?v=KEcr22qZAVE&t=76s)
- Waveshare ESP32-C6-LCD-1.47 (https://www.youtube.com/watch?v=r2-nN8fWAxs)

Datasheets

ESP32-C6

- ESP32-C6 Technical Reference Manual (https://files.waveshare.com/wiki/common/ESP32-C6_Technical_Reference_Manual.pdf)

- ESP32-C6 Series Datasheet (https://files.waveshare.com/wiki/common/ESP32-C6_Series_Datasheet.pdf)

Other Components

- 1.47inch LCD Datasheet (https://files.waveshare.com/wiki/ESP32-C6-LCD-1.47/1.47inch_LCD_Datasheet.pdf)

Software Tools

Arduino

- Arduino IDE Official download link (https://www.arduino.cc/en/software/)
- ESP32_Arduino offline package (https://drive.google.com/drive/folders/1Pcs_A4FKWvdSHnz9lEBYqOpr-noTMbIv?usp=sharing)

VScode

- VScode official website (https://code.visualstudio.com/download)

Debugging Tool

- Flash_download_tool (https://dl.espressif.com/public/flash_download_tool.zip)

Other Resource Links

- ESP32-Arduino official documentation (https://docs.espressif.com/projects/arduino-esp32/en/latest/index.html)
- LVGL official documentation (https://docs.lvgl.io/master/intro/introduction/index.html)

Project Resources

This section features third - party project resources. We merely provide links and bear no responsibility for content updates or maintenance. Thank you for your understanding.

Volos Projects-How to Interact with Your UI - LVGL and Squareline Studio

- Youtube: https://www.youtube.com/watch?v=KEcr22qZAVE (https://www.youtube.com/watch?v=KEcr22qZAVE)
- Github: https://github.com/VolosR/WaveShareC6lvglexample (https://github.com/VolosR/WaveShareC6lvglexample)

nishad2m8- Flux Capacitor UI on 1.47" ESP32C6 | EEZ Studio & LVGL

- Youtube: https://www.youtube.com/watch?v=pDeoPXndeUQ (https://www.youtube.com/watch?v=pDeoPXndeUQ)
- Github: https://github.com/nishad2m8/Squareline-OBP/tree/master/WaveShare/esp32-c6-1.47 (https://github.com/nishad2m8/Squareline-OBP/tree/master/WaveShare/esp32-c6-1.47)

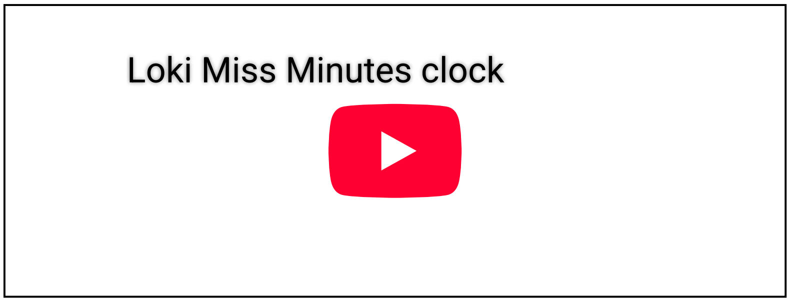
The Last Outpost Workshop- SD Card Video Player & Crypto Price Monitor tutorial

- Youtube : https://www.youtube.com/watch?v=JqQEG0eipic (https://www.youtube.com/watch?v=JqQEG0eipic)
- Github :

https://github.com/thelastoutpostworkshop/ESP32-C6-LCD-1.47_LVGL9_Crypto_Monitor (https://github.com/thelastoutpostworkshop/ESP32-C6-LCD-1.47_LVGL9_Crypto_Monitor)

https://github.com/thelastoutpostworkshop/ESP32-C6-LCD-1.47_video_player (https://github.com/thelastoutpostworkshop/ESP32-C6-LCD-1.47_video_player)

moononournation - Loki Miss Minutes clock



- Youtube : https://www.youtube.com/watch?v=e-Z9bq2xT2E (https://www.youtube.com/watch?v=e-Z9bq2xT2E)
- Guide: https://www.instructables.com/GIF-Animation-Clock/ (https://www.instructables.com/GIF-Animation-Clock/)
- Github : https://github.com/moononournation/animation_clock (https://github.com/moononournation/animation_clock)

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 (https://files.waveshare.com/wiki/common/CH34XSER_MAC.7z) and flash again.

Question: Why is there no output after successfully flashing 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 (https://www.waveshare.com/wiki/Waveshare_SquareLine_Studio)

Support

Technical Support

If you need technical support or have any feedback/review, please click the **Submit Now** button to submit a ticket, Our support team will check and reply to you within 1 to 2 working days. Please be patient as we make every effort to help you to resolve the issue.
Working Time: 9 AM - 6 PM GMT+8 (Monday to Friday)

Submit Now (https://service.waveshare.com/)

Retrieved from "https://www.waveshare.com/w/index.php?title=ESP32-C6-LCD-1.47&oldid=108132 (https://www.waveshare.com/w/index.php?title=ESP32-C6-LCD-1.47&oldid=108132)"