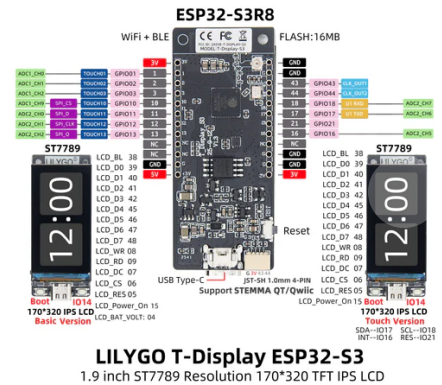
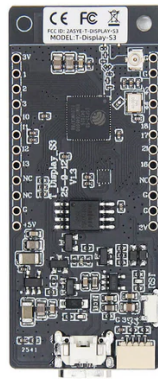


T-Display-S3



Overview

T-Display-S3 is a development board whose main control chip is ESP32-S3. It is equipped with a 1.9-inch LCD color screen and two programmable buttons. Communication using the I8080 interface Retains the same layout design as T-Display. You can directly use ESP32S3 for USB communication or programming.

Quick Start

Example Support

Example	PlatformIO/Arduino	ESP-IDF	Description
T-Display-S3	✓	✓	BLE, WIFI, SPIFFS, FFat examples

PlatformIO

1. Install Visual Studio Code and Python
2. Search for the PlatformIO plugin in the VisualStudioCode extension and install it.
3. After the installation is complete, you need to restart VisualStudioCode
4. After restarting VisualStudioCode, select File in the upper left corner of VisualStudioCode → Open Folder → select the T-Display-S3 directory
5. Wait for the installation of third-party dependent libraries to complete
6. Click on the platformio.ini file, and in the platformio column
7. Uncomment one of the lines default_envs = xxxx to make sure only one line works
8. Click the (✓) symbol in the lower left corner to compile
9. Connect the board to the computer USB
10. Click (→) to upload firmware
11. Click (plug symbol) to monitor serial output
12. If it cannot be written, or the USB device keeps flashing, please check the FAQ below

Arduino

Install Arduino IDE

1. In Arduino Preferences, on the Settings tab, enter the https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json URL in the Additional boards manager URLs input box. Please pay attention to the version. The test phase is using 2.0.14. It is not certain that versions above 2.0.14 can run. When the operation is abnormal, please downgrade to a version below 2.0.14. As of 2024/08/02, TFT_eSPI does not work on versions higher than 2.0.14, see TFT_eSPI/issue3329.
2. Download T-Display-S3 , move to Arduino library folder (e.g. C:\Users\YourName\Documents\Arduino\libraries)
3. Copy all folders in lib folder to Arduino library folder (e.g. C:\Users\YourName\Documents\Arduino\libraries)
4. Enter the downloaded T-Display-S3/examples directory

5. Select any example and double-click the any_example.ino to open it
6. Select any example and double-click the `any_example.ino` to open it
7. Open ArduinoIDE , **Tools** , Make your selection according to the table below

Arduino IDE Setting	Value
Board	ESP32S3 Dev Module
Port	Your port
USB CDC On Boot	Enable
CPU Frequency	240MHZ(WiFi)
Core Debug Level	None
USB DFU On Boot	Disable
Erase All Flash Before Sketch Upload	Disable
Events Run On	Core1
Flash Mode	QIO 80MHZ
Flash Size	16MB(128Mb)
Arduino Runs On	Core1
USB Firmware MSC On Boot	Disable
Partition Scheme	16M Flash(3M APP/9.9MB FATFS)
PSRAM	OPI PSRAM
Upload Mode	UART0/Hardware CDC
Upload Speed	921600
USB Mode	CDC and JTAG

- The options in bold are required, others are selected according to actual conditions.

8. Click **upload** , Wait for compilation and writing to complete

ESP-IDF

- [T-Display-S3](#) esp-idf version example, please jump to this [LilyGo-Display-IDF](#)

Micropython

- [russhughes/st7789s3_mpy](#)
- [Micropython](#)

Development Platforms

1. [Arduino IDE](#)
 2. [PlatformIO](#)
 3. [MicroPython](#)
-

Video

- [T-Display-S3 Arduino IDE Record](#)
-

Key Features

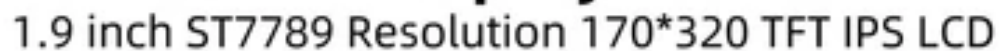
- ESP32-S3R8 Dual-core LX7 microprocessor
 - Wi-Fi 802.11, BLE 5 + BT mesh
 - 1.9" diagonal, Full-color TFT Display
 - 170(H)RGB x320(V) 8-Bit Parallel Interface
 - STEMMA QT / Qwiic
 - JST-GH 1.25mm 2PIN
-

Product Parameters

Feature	Specification
MCU	ESP32-S3R8 Dual-core LX7 microprocessor
Wireless Connectivity	Wi-Fi 802.11, BLE 5 + BT mesh
Programming Platform	Arduino-ide、 Micropython
Flash	16MB
PSRAM	8MB
Bat voltage detection	IO04
Onboard functions	Boot + Reset + IO14 Button
LCD	1.9" diagonal, Full-color TFT Display
Drive Chip	ST7789V
Resolution	170(H)RGB x320(V) 8-Bit Parallel Interface
Working power supply	3.3v
Support	STEMMA QT / Qwiic JST-SH 1.0mm 4-PIN
Connector	JST-GH 1.25mm 2PIN

Pin Diagram

FLASH:16MB



Button

Button	BOOT	User
ESP32-S3	GPIO0	GPIO14

Battery

Battery	ADC
ESP32-S3	GPIO4

I2C (QWIIC / STEMMA QT)

I2C	SDA	SCL
ESP32-S3	GPIO18	GPIO17

UART

UART	TX	RX
ESP32-S3	GPIO43	GPIO44

Dimension Diagram

- [DWG File](#)
-

Schematic

- [T-Display-S3 Schematic](#)
-

Datasheet

- [ESP32-S3 Datasheet](#)
-

Software Development

- [T-Display-S3 GitHub Repository](#)

Dependent Libraries

- [lvgl](#)
 - [Arduino_GFX](#)
 - [TFT-eSPI](#)
 - [ESP32_Host_MIDI](#)
 - [Gingoduino](#)
 - [Adafruit_MPR121](#)
-

FAQ

1. The screen does not light up when using battery?

- When T-Display-S3 is powered by battery, GPIO15 must be set to HIGH to turn on the backlight.
- Please add the following two lines at the beginning of the setup

```
void setup(){  
    //Turn on display power  
    pinMode(15, OUTPUT);  
    digitalWrite(15, HIGH);  
}
```

C++

2. The program can be written normally, but there is still no display after writing

- If you are using **TFT_eSPI**, then you can try running [Arduino_GFXDemo](#) first. If nothing is displayed after writing, you can determine that there is a problem with the hardware.

- If `Arduino_GFXDemo` is written normally, but `TFT_eSPI` is not displayed, then it can be judged that `User_Setup_Select` has been overwritten, then please read the third article of [FAQ](#) to reconfigure `TFT_eSPI`.

3. How to update `TFT_eSPI`, or confirm whether the `TFT_eSPI` pin configuration is correct?

- Search for `TFT_eSPI` in the ArduinoIDE library manager and click Update.
- Enter the default library manager installation location and open the `TFT_eSPI` folder.
The default installation location is: (e.g.
C:\Users\YourName\Documents\Arduino\libraries)
- Open `User_Setup_Select.h`, comment out `#include <User_Setup.h>` which is enabled by default, or delete it.
- Search `Setup206_LilyGo_T_Display_S3`, find it, cancel the previous comment, then save it, and finally close it, so that `TFT_eSPI` uses the pin definition of T-Display-S3 by default.

```

C++
#include <User_Setups/Setup206_LilyGo_T_Display_S3.h> // For the LilyGo T-Displa

```



4. Can't upload any sketch, Please enter the upload mode manually.

- Connect the board via the USB cable
- Press and hold the **BOOT** button , While still pressing the **BOOT** button
- Press **RST** button
- Release the **RST** button
- Release the **BOOT** button (If there is no **BOOT** button, disconnect IO0 from GND.)
- Upload sketch
- Press the **RST** button to exit download mode

5. If you use external power supply instead of USB-C, please turn off the CDC option.

This is because the board will wait for USB access when it starts.

- For Arduino IDE users, it can be turned off in the options , Please note that turning off USB CDC will turn off Serial redirection to USBC. At this time, you will not see any Serial message output when opening the port from USB-C, but output from GPIO43 and GPIO44.

Tools -> USB CDC On Boot -> Disable

- For platformio users, you can add the following compilation flags in the ini file

```
build_flags =  
    ; Enable -DARDUINO_USB_CDC_ON_BOOT will start printing and wait for terminal acc  
    ; -DARDUINO_USB_CDC_ON_BOOT=1  
  
    ; Enable -UARDUINO_USB_CDC_ON_BOOT will turn off printing and will not block whe  
    -UARDUINO_USB_CDC_ON_BOOT
```



- 6. If all the above are invalid, please flash the factory firmware for quick verification, please check [here](#)
- 7. Can I use an external 5V pin for power? Please see here [issues/205](#)
- 8. The default charging current is set at 500mA per hour. If you need to adjust the charging current, please see this [issue](#)

Version History

Version	Release Date	Update Description
---------	--------------	--------------------

[Edit this page](#)

Last updated: 8/7/26, 10:48 AM

Previous page

[Quick Start](#)

Next page

[Quick Start](#)