WAVESHARE ESP32-S3 Touch LCD 4.3 Inch User Guide

Contents

WAVESHARE ESP32-S3 Touch LCD 4.3 Inch

Specifications

- **Product Name:** ESP32-S3-Touch-LCD-4.3
- **Wireless Support:** 2.4GHz WiFi and BLE 5
- **Display:** 4.3-inch capacitive touchscreen
- **Memory:** High-capacity Flash and PSRAM

Product Overview
The ESP32-S3-Touch-LCD-4.3 is a microcontroller development board that integrates WiFi, BLE, a capacitive touch screen, and various peripheral interfaces. It is suitable for developing Human-Machine Interfaces (HMI) and other ESP32-S3 applications.

Hardware Description
The board features multiple interfaces including UART, USB, Sensor, CAN, I2C, RS485, and a battery header for efficient charging and discharging management.

Onboard Interface

- **UART Interface:** CH343P chip for USB to UART communication.
- **USB Interface:** GPIO19(DP) and GPIO20(DN) for USB communication.
- **Sensor Interface:** Connects GPIO6 as ADC for sensor integration.
- **CAN Interface:** Shared with USB interface for multiplexed function.
- **I2C Interface:** Multiple hardware I2C interfaces available.
- **RS485 Interface:** Onboard circuit for direct RS485 communication.
- **Battery Header:** Supports efficient battery charging and discharging management.

PIN Connection

Hardware Connection
Ensure proper connection of peripherals to the corresponding interfaces as described in the manual.

Environment Setting
The software framework supports CircuitPython, MicroPython, and C/C++ (Arduino, ESP-IDF) for rapid prototyping and development.

Overview

Introduction

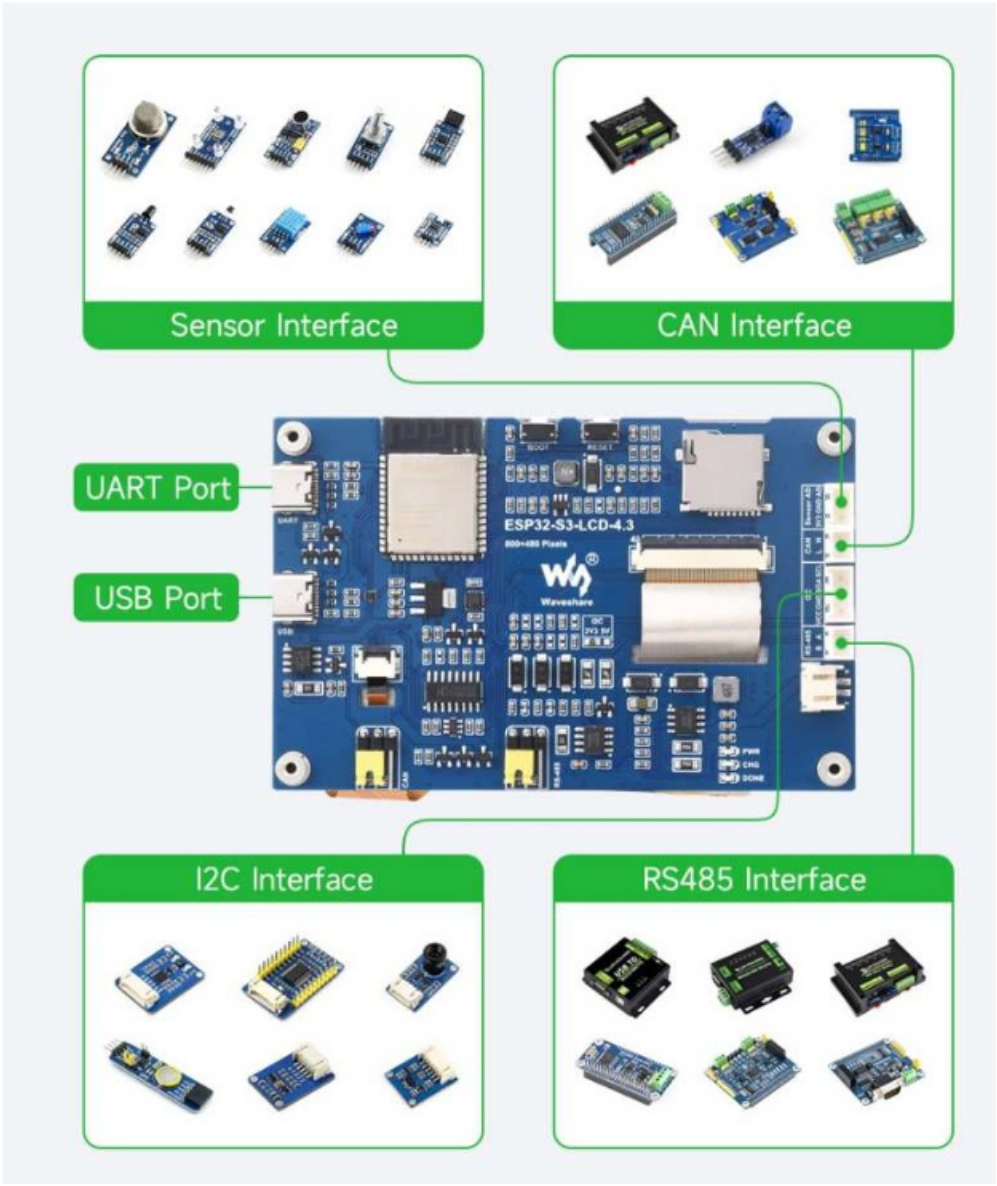
ESP32-S3-Touch-LCD-4.3 is a microcontroller development board with 2.4GHz WiFi and BLE 5 support, and integrates high-capacity Flash and PSRAM. The onboard 4.3-inch capacitive touch screen can smoothly run GUI programs such as LVGL. Combined with various peripheral interfaces, suitable for the quick development of the HMI and other ESP32-S3 applications.

Features

- Equipped with Xtensa 32-bit LX7 dual-core processor, up to 240MHz main frequency.
- Supports 2.4GHz Wi-Fi (802.11 b/g/n) and Bluetooth 5 (LE), with an onboard antenna.
- Built-in 512KB of SRAM and 384KB ROM, with onboard 8MB PSRAM and 8MB Flash.
- Onboard 4.3-inch capacitive touch display, 800×480 resolution, 65K color.
- Supports capacitive touch control via I2C interface, 5-point touch with interrupt support.
- Onboard CAN, RS485, I2C interface, and TF card slot, integrate full-speed USB port.
- Supports flexible clock, module power supply independent setting, and other controls to realize low power consumption in different scenarios.

Hardware Description

Onboard Interface



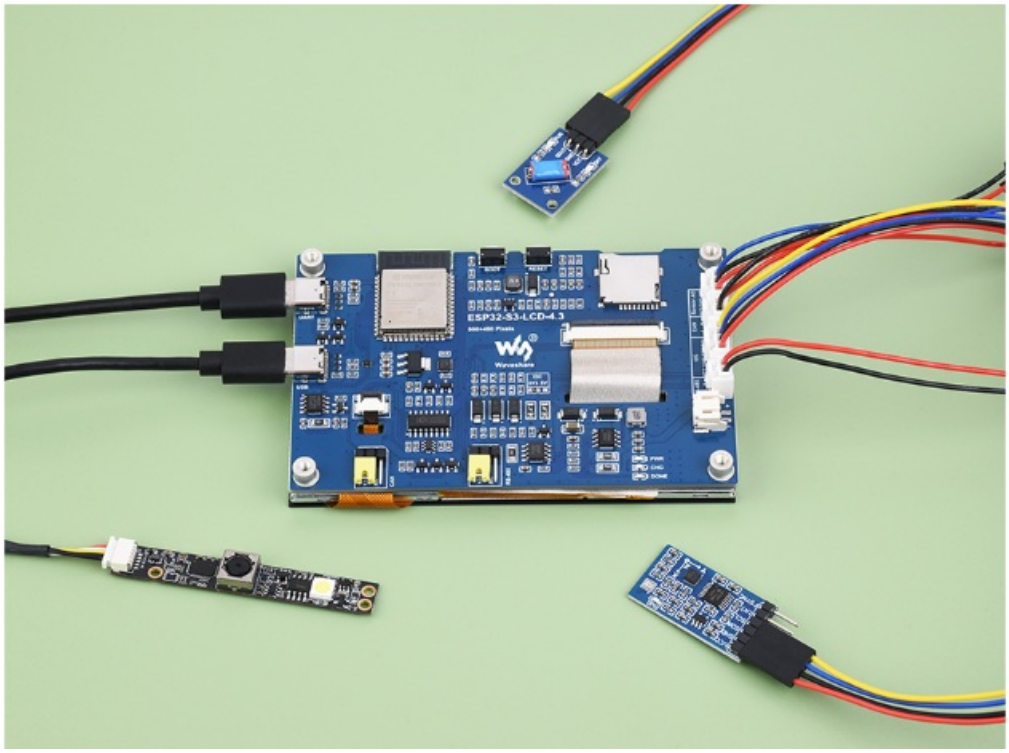
- **UART interface:** Using CH343P chip for USB to UART to connect to ESP32-S3's UART_TXD(GPIO43) and UART_RXD(GPIO44), enabling firmware burning and log printing.
- **USB interface:** GPIO19(DP) and GPIO20(DN) are the USB pins of ESP32-S3 by default, and the interface can be used for connecting cameras with protocols such as UVC. Please click here to view the UVC driver.
- **Sensor interface:** this interface is for connecting GPIO6 as ADC, and can be connected to sensors .
- **CAN interface:** The CAN interface pins and USB interface pins share a multiplexed function, utilizing the FSUSB42UMX chip for switching. By default, the USB interface is used (when the USB_SEL pin of FSUSB42UMX is set to HIGH).
- **I2C interface**: ESP32-S3 offers multiple hardware I2C interfaces. Currently, GPIO8 (SDA) and GPIO9 (SCL) pins are used as the I2C bus to connect to the IO expansion chip, touch interfaces, and other I2C peripherals.
- **RS485 interface:** The development board is equipped with an onboard RS485 interface circuit, allowing direct communication with RS485 devices. The RS485 circuit automatically switches between transmit and receive modes.
- **PH2.0 battery header:** The development board employs the efficient charging and discharging management chip CS8501, capable of boosting a single lithium battery to 5V. Currently, the charging current is set at 580mA. Users can modify the charging current by replacing the R45 resistor. For further details, please refer to the schematic diagram.

PIN Connection

ESP32-S3-WROOM-x GPIO0	LCD G3	USB	SD	UART	CAN	Sensor
GPIO1	R3					
GPIO2	R4					
GPIO3	VSYNC					
GPIO4	TP_IRQ					
GPIO5	DE					
GPIO6						AD
GPIO7	PCLK					
GPIO8	TP_SDA					
GPIO9	TP_SCL					
GPIO10	B7					
GPIO11			MOSI			
GPIO12			SCK			
GPIO13			MISO			
GPIO14	B3					
GPIO15				RS485_TX		
GPIO16				RS485_RX		
GPIO17	B6					
GPIO18	B5					
GPIO19		USB_DN			CANRX	
GPIO20		USB_DP			CANTX	
GPIO21	G7					
GPIO38	B4					
GPIO39	G2					
GPIO40	R7					
GPIO41	R6					
GPIO42	R5					
GPIO43				UART_TXD		
GPIO44				UART_RXD		
GPIO45	G4					

GPIO46	HSYNC					
GPIO47	G6					
GPIO48 CH422G	G5 –	–	–	–	–	–
EXIO1	TP_RST					
EXIO2	DISP					
EXIO3	LCD_RST					
EXIO4			SD_CS			
EXIO5		USB_SEL(HIGH)			USB_SEL(LOW)	

Hardware Connection



- ESP32-S3-Touch-LCD-4.3 comes with an onboard automatic download circuit. The Type C port, marked UART, is used for program downloads and logging. Once the program is downloaded, run it by pressing the RESET button.
- Please keep other metals or plastic material away from the PCB antenna area during use.
- The development board uses a PH2.0 connector to extend ADC, CAN, IC, and RS485 peripheral pins. Utilize a PH2.0 to 2.54mm DuPont male connector to link sensor components.
- As the 4.3-inch screen occupies most GPIO pins, you can use a CH422G chip to expand IO for functions like

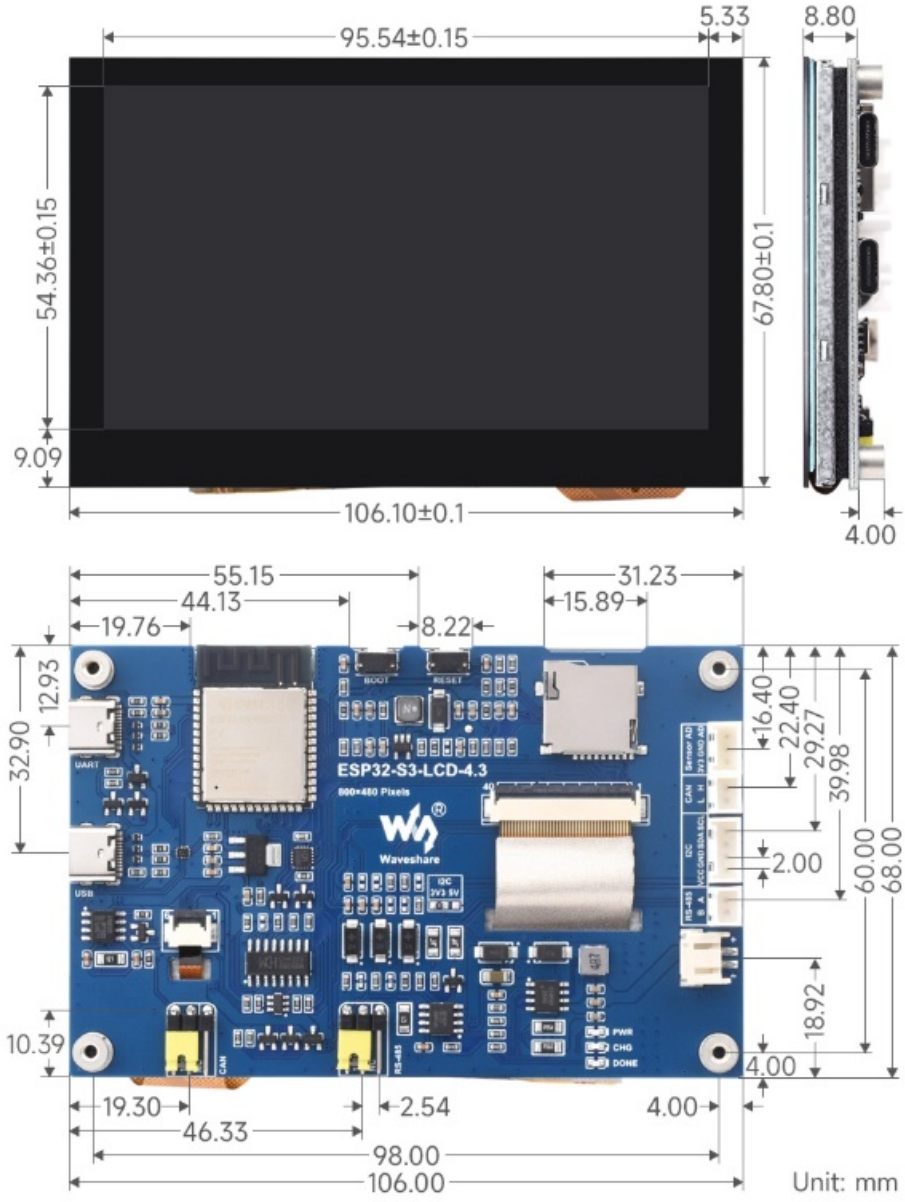
reset and backlight control.

- The CAN and RS485 peripheral interfaces connect to a 120Ohm resistor using jumper caps by default. Optionally, connect NC to cancel the termination resistor.
- The SD card employs SPI communication. Note that the SD_CS pin needs to be driven by the EXIO4 of the CH422G.

Other Notes

- The average frame rate for running the LVGL benchmark example on a single core in ESP-IDF v5.1 is 41 FPS. Before compilation, enabling 120M PSRAM is necessary.
- The PH2.0 lithium battery socket only supports a single 3.7V lithium battery. Do not use multiple sets of battery packs for charging and discharging simultaneously. It's recommended to use a single-cell battery with a capacity below 2000mAh.

Dimensions



Environment Setting

The software framework for ESP32 series development boards is completed, and you can use CircuitPython, MicroPython, and C/C+ + (Arduino, ESP-IDF) for rapid prototyping of product development. Here's a brief introduction to these three development approaches:

- CircuitPython is a programming language designed to simplify coding tests and learning on low-cost microcontroller boards. It is an open-source derivative of the MicroPython programming language, primarily aimed at students and beginners. CircuitPython development and maintenance are supported by Adafruit Industries.
    - You can refer to development documentation ® for CircuitPython-related applications development.
    - The GitHub& library for CircuitPython allows for recompilation for custom development.
- MicroPython is an efficient implementation of the Python 3 programming language. It includes a small subset of the Python standard library and has been optimized to run on microcontrollers and resource-constrained environments.
    - You can refer to development documentation & for MicroPython-related application development.
    - The GitHub library & for MicroPython allows for recompilation for custom development.
- The official libraries and support from Espressif Systems for C/C++ development make it convenient for rapid installation.
    - Users can select Arduino &
    - Visual Studio Code (ESP-IDF) as their Integrated Development Environment (IDE).
- The environment is set up under Windows 10, users can choose to use Arduino or Visual Studio Code (ESP-IDF) as IDE for development, Mac/Linux OS users please refer to the official instructions&.

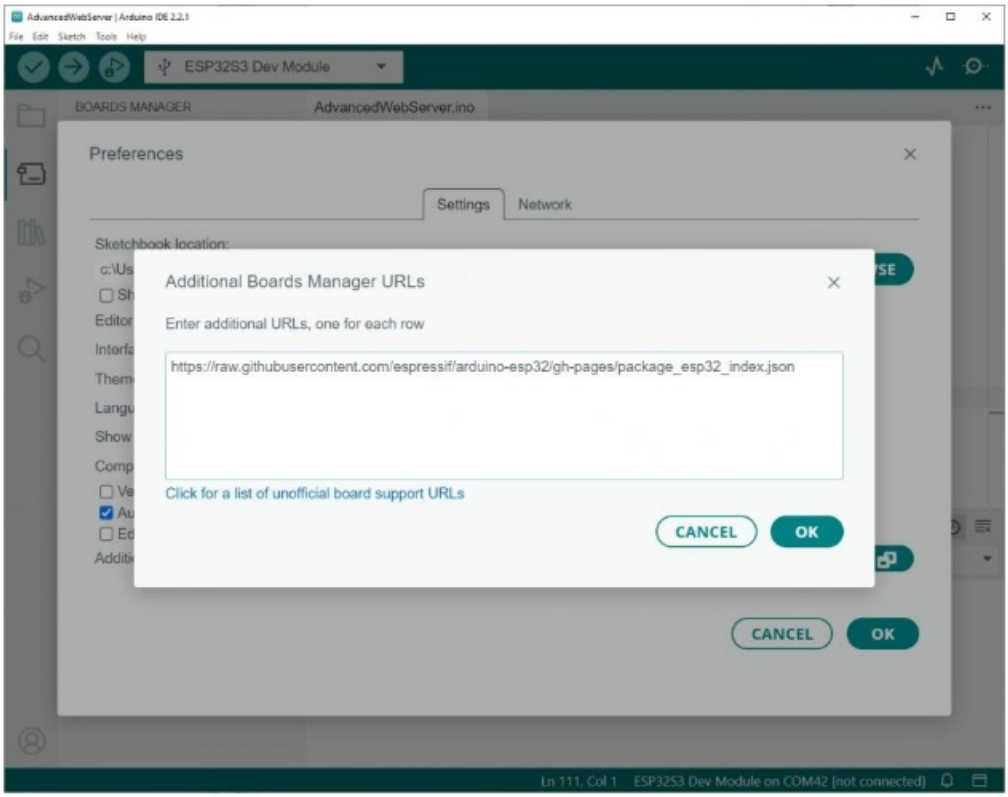
ESP-IDF

- ESP-IDF installation &

Arduino

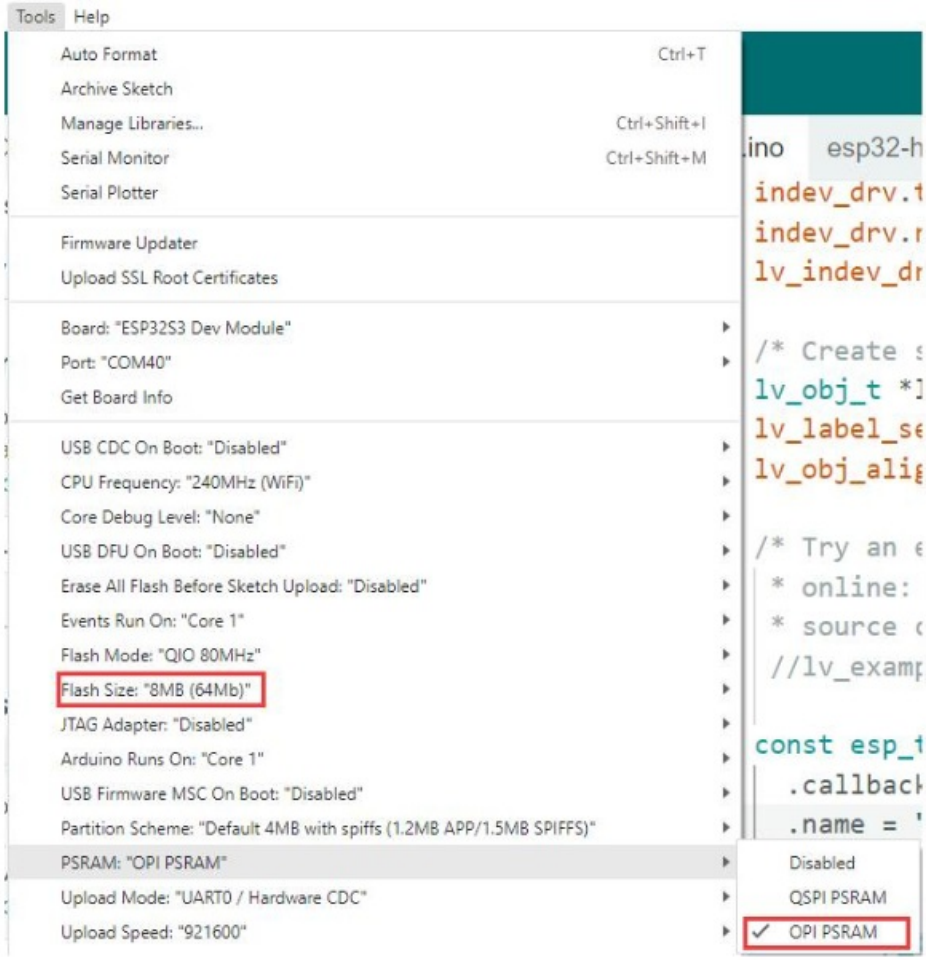
- Download and install Arduino IDE&.
- Install ESP32 on the Arduino IDE as shown below, and you can refer to this link &.
- Fill in the following link in the Additional Boards Manager URLs section of the Settings screen under File -> Preferences and save.

```
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
```



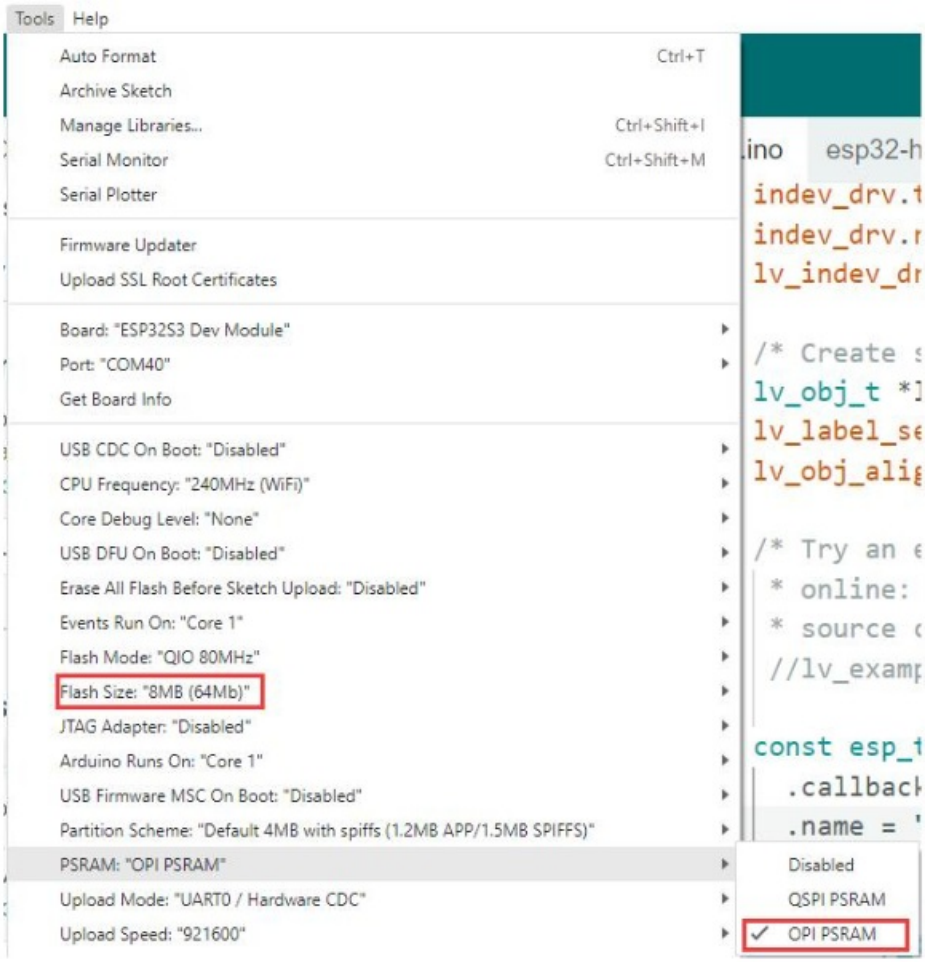
- Search esp32 on Board Manager to install, and restart Arduino IDE to take effect.

- Search esp32 on Board Manager to install, and restart Arduino IDE to take effect.



- Open the Arduino IDE and note that Tools in the menu bar selects the corresponding Flash (8MB) and enables

PSRAM (8MB OPI), as shown in the following figure.



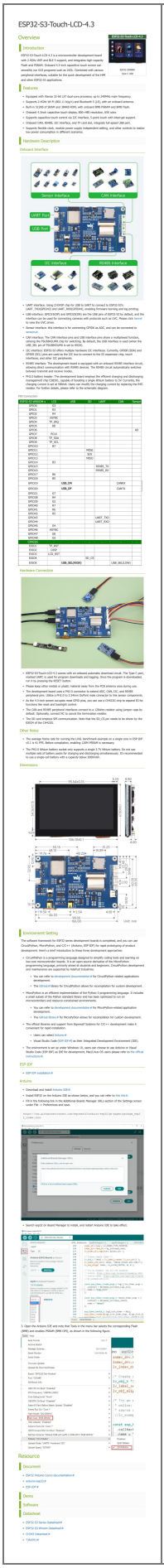
Resource

- **Document**
  - ESP32 Arduino Core’s documentation
  - Arduino-esp32
  - ESP-IDF
  - Demo
- **Software**
- **Datasheet**
  - ESP32-S3 Series Datasheet t
  - ESP32-S3 Wroom Datasheet
  - CH343 Datasheet&
  - TJA1051

FAQ

Q: Can I use multiple battery packs with the PH2.0 battery header?
A: The PH2.0 lithium battery socket only supports a single 3.7V lithium battery. Do not use multiple sets of battery packs simultaneously.

Documents / Resources



[WAVESHARE ESP32-S3 Touch LCD 4.3 Inch](#) [pdf] User Guide
ESP32-S3 Touch LCD 4.3 Inch, ESP32-S3, Touch LCD 4.3 Inch, LCD 4.3 Inch

References

- [espressif/usb_host_uvc • v1.0.4 • ESP Component Registry](#)

- Core Modules — Adafruit CircuitPython 9.1.0-beta.3 documentation
- Welcome to ESP32 Arduino Core's documentation - - — Arduino ESP32 latest documentation
- Installing - - — Arduino ESP32 latest documentation
- GitHub - adafruit/Adafruit_CircuitPython_Bundle: A bundle of useful CircuitPython libraries ready to use from the filesystem.
- GitHub - espressif/esp-idf: Espressif IoT Development Framework. Official development framework for Espressif SoCs.
- GitHub - micropython/micropython: MicroPython - a lean and efficient Python implementation for microcontrollers and constrained systems
- Software | Arduino
- Sensors Pack Tens of Different Sensors in One Pack
- Log in - Waveshare Wiki
- File:ESP32-C3-Zero -05.jpg - Waveshare Wiki
- File:ESP32-S3-Touch-LCD-4.3-Ar.jpg - Waveshare Wiki
- File:ESP32-S3-Touch-LCD-4.3-Ar02.jpg - Waveshare Wiki
- File:ESP32-S3-Touch-LCD-4.3-connection.jpg - Waveshare Wiki
- File:ESP32-S3-Touch-LCD-4.3-Dimension.jpg - Waveshare Wiki
- File:ESP32-S3-Touch-LCD-4.3-Hardware.png - Waveshare Wiki
- File:ESP32-S3-Touch-LCD-4.3.jpg - Waveshare Wiki
- Waveshare Wiki
- User Manual