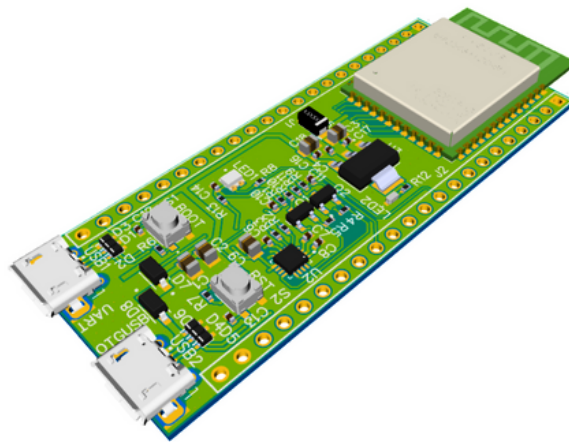
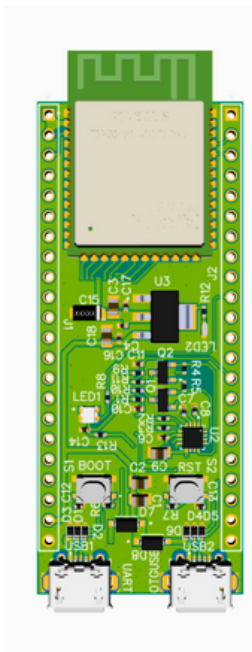


ESP32-S3-DEV-MODULE Datasheet

Revision: 1.0 – August 2025

Table of contents

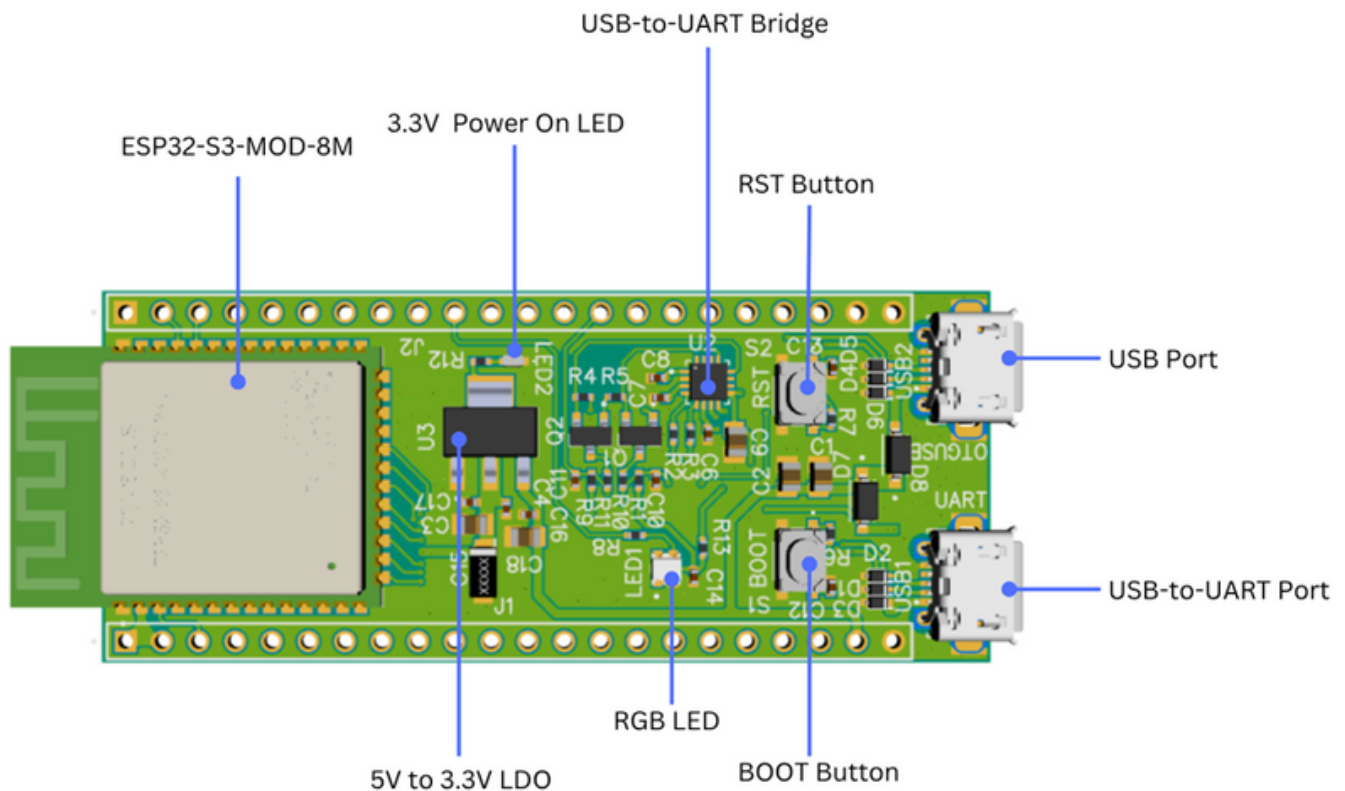
1. Overview	-----	1
2. Features	-----	1
3. Applications	-----	2
4. Block Diagram	-----	2
5. Functional Description	-----	2
6. Pin Description	-----	3
7. Electrical Characteristics	-----	4
8. Usage Guide	-----	4 - 10
9. Mechanical Dimensions	-----	11
10. Ordering Information	-----	11
11. References	-----	11

1. Overview

The **ESP32-S3-DEV-MODULE** is a general-purpose development board based on the custom ESP32-S3-MOD-8M module. Designed similarly to the ESP32-S3-DevKitC-1, it includes USB-to-UART and USB-OTG interfaces, boot and reset buttons, a UART bridge, and an onboard RGB LED.

2. Features

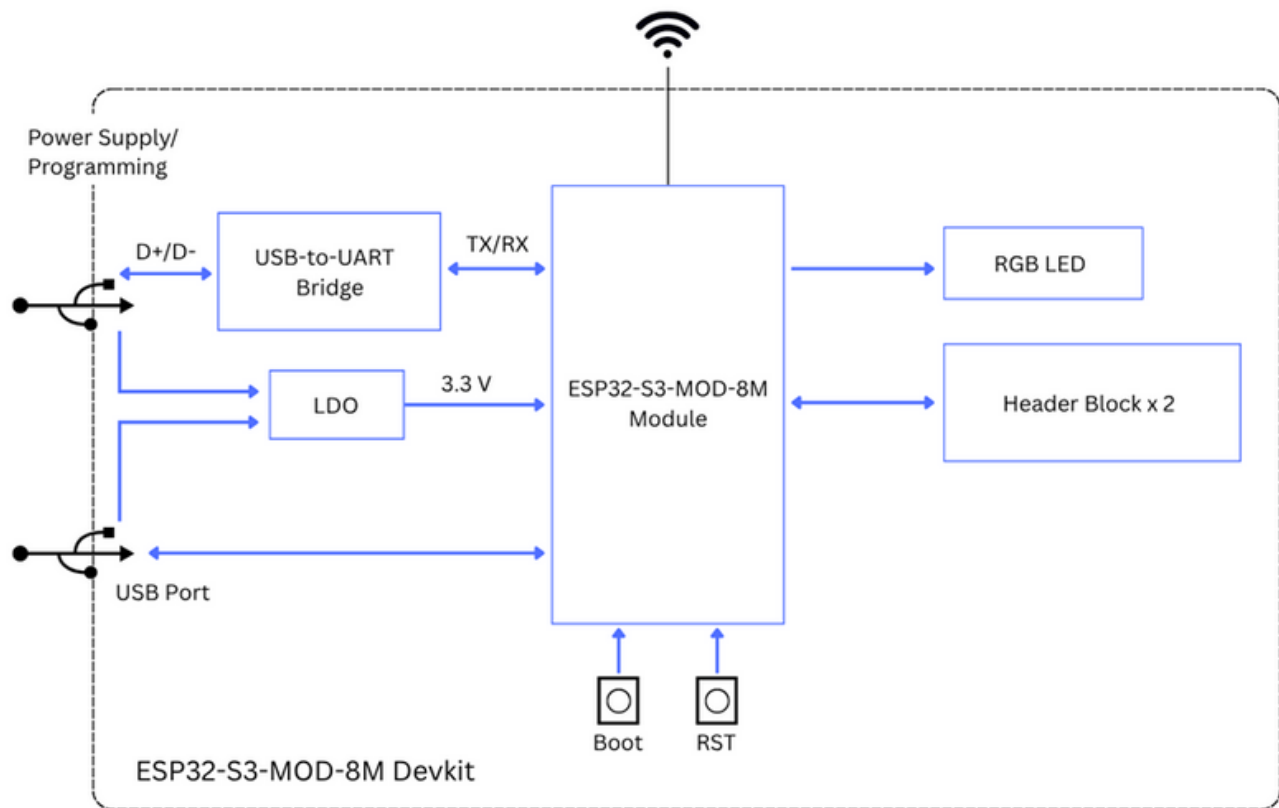
- Based on custom ESP32-S3-MOD-8M module (ESP32-S3 + 8MB Flash)
- Dual Micro USB connectors: one for UART, one for USB-OTG
- USB-to-UART bridge IC: CH343P
- Onboard WS2812B-2020 RGB LED
- Reset and Boot buttons
- Compatible with ESP-IDF and Arduino
- Breadboard-friendly design with castellated headers
- Operating voltage: 3.3V, USB-powered



3. Applications

- IoT and AIoT prototyping
- USB device/host development
- Bluetooth and Wi-Fi based control systems
- Wearables, smart gadgets, sensors

4. Block Diagram



5. Functional Description

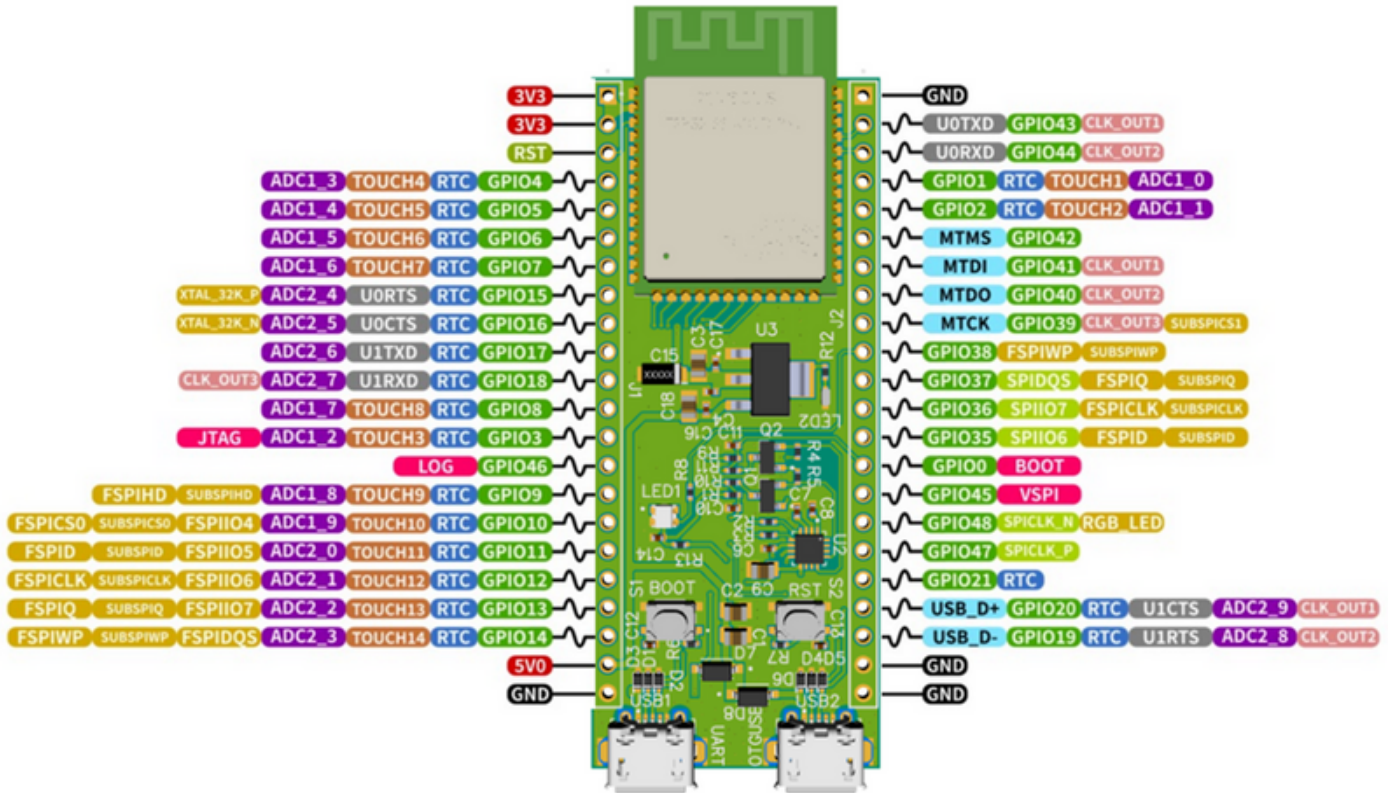
The **ESP32-S3-DEV-MODULE** integrates two USB ports for programming and OTG functions. The CH343P handles USB-to-UART conversion, providing easy flashing and debugging.

The boot and reset switches enable user-friendly firmware uploading.

An onboard RGB LED is connected to a user GPIO for status or effect indication.

6. Pin Description

The board exposes most ESP32-S3 GPIOs via 2.54 mm header pins, in a layout matching the ESP32-S3-DevKitC-1. Refer to the ESP32-S3-MOD-8M datasheet for exact pin functions.



 PWM Capable Pin

 JTAG/USB JTAG for Debugging and USB

 TOUCHX Touch Sensor Input Channel

 SERIAL Serial for Debug/Programming

 MISC Miscellaneous/SPI Functions

 RTC RTC Power Domain (VDD3P3_RTC)

 GPIOX GPIO Input and Output

 ADCX_CH Analog-to-Digital Converter

 OTHER Other Related Functions

 STRAP Strapping Pin Functions

 CLK_OUTx Clock Output

 PWR Power Rails (3V3 and 5V)

 GND Ground

7. Electrical Characteristics

Parameter	Min	Typ	Max	Unit
Supply Voltage (USB)	4.75	5.0	5.25	V
Operating Voltage (VDD)	3.0	3.3	3.6	V
Wi-Fi TX Current	-	120	-	mA
Wi-Fi RX Current	-	80	-	mA
Deep Sleep Current	-	10	20	μA

8. Usage Guide

8.1 Initial Setup

Connect the USB-UART port to your computer using a micro-USB cable. Ensure the appropriate CH343 driver is installed.

To enter flash mode:

1. Hold down the BOOT button (GPIO0).
2. Press the RST button.
3. Release the BOOT button.

You can now program the board using the Arduino IDE, Platform IO, or ESP-IDF.

8.2 Development Environment Configuration

Getting Started with Arduino IDE

Step 1 – Installing Arduino IDE

Open the [software download page](#) of the official website, and select the corresponding system and system bit to download.

Bring Your Projects to Life with Arduino Software



Arduino IDE 2.3.6
Release notes

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, check the [Arduino IDE 2.0 documentation](#).

Windows Win 10 or newer (64-bit) **DOWNLOAD**

Nightly Builds
Download a preview of the incoming release with the most updated features and bugfixes.

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

Step 2 - Run the installer and follow the default installation prompts.

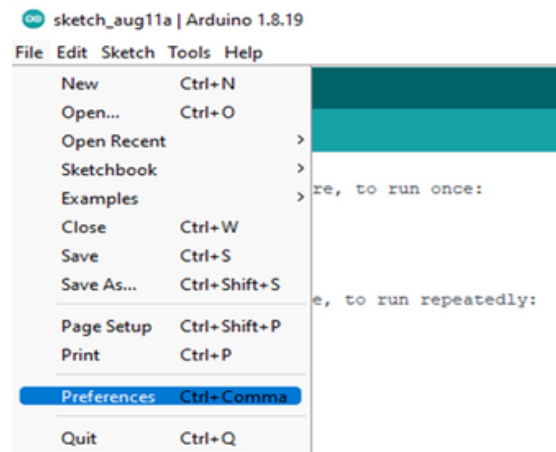
After installation, open the Arduino IDE.

Step 3 - Install Arduino-esp32-S3 Online

To install the ESP32 board in your Arduino IDE, follow these next instructions:

Open **Arduino IDE**.

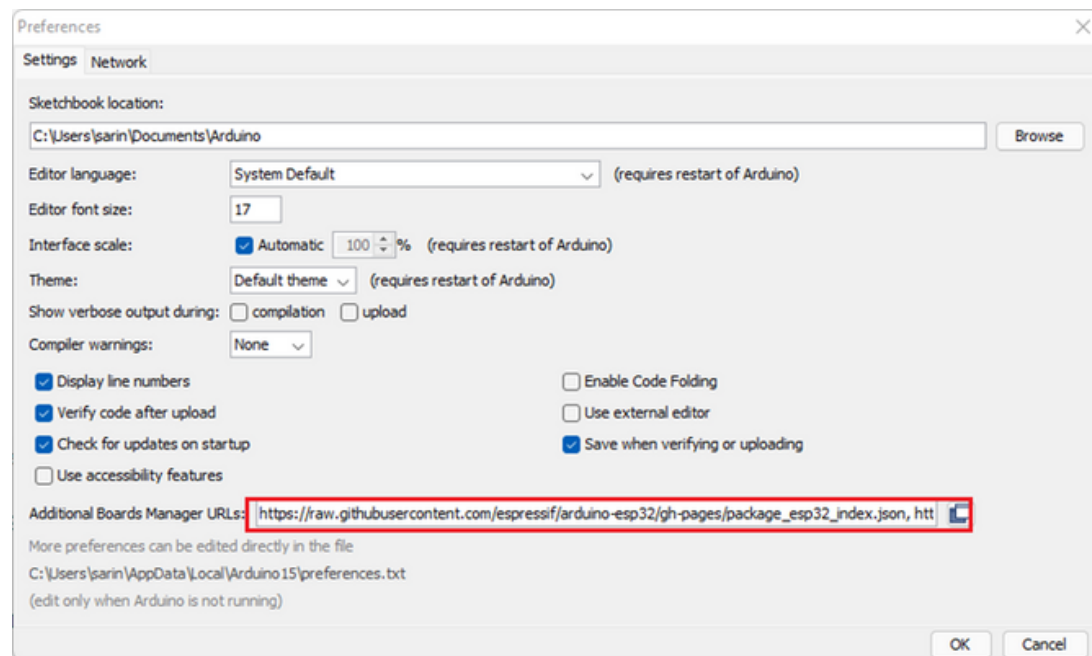
1. In your Arduino IDE, go to **File > Preferences**



2. Enter the following into the "Additional Board Manager URLs" field:

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

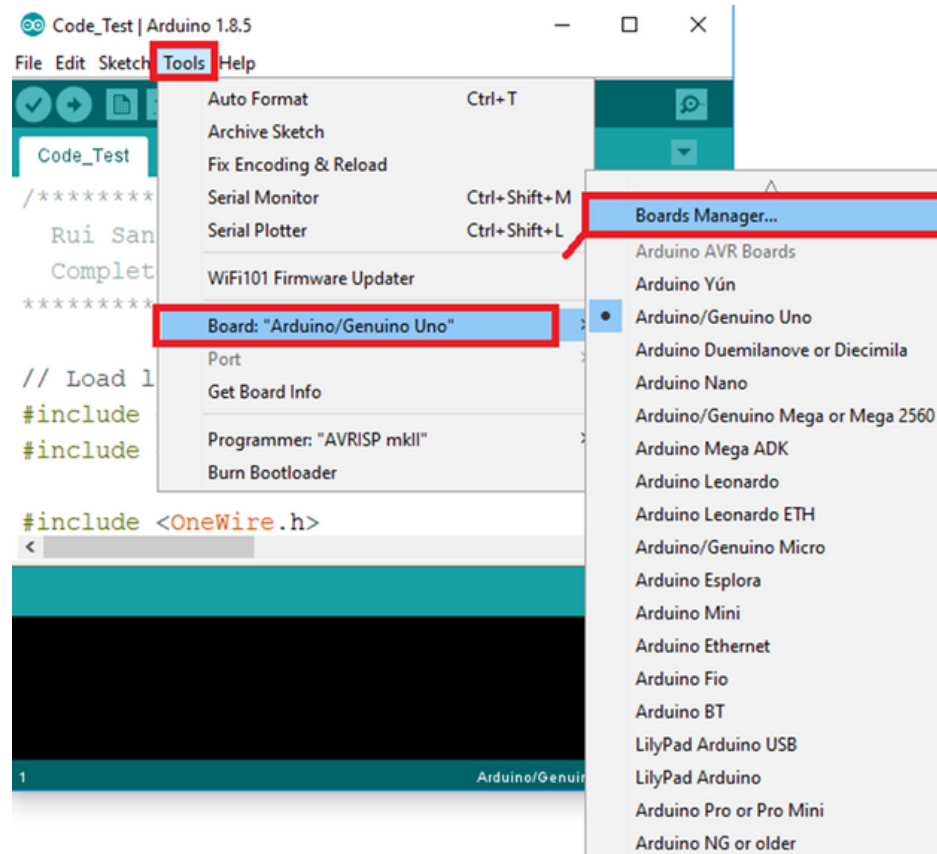
Then, click the "OK" button:



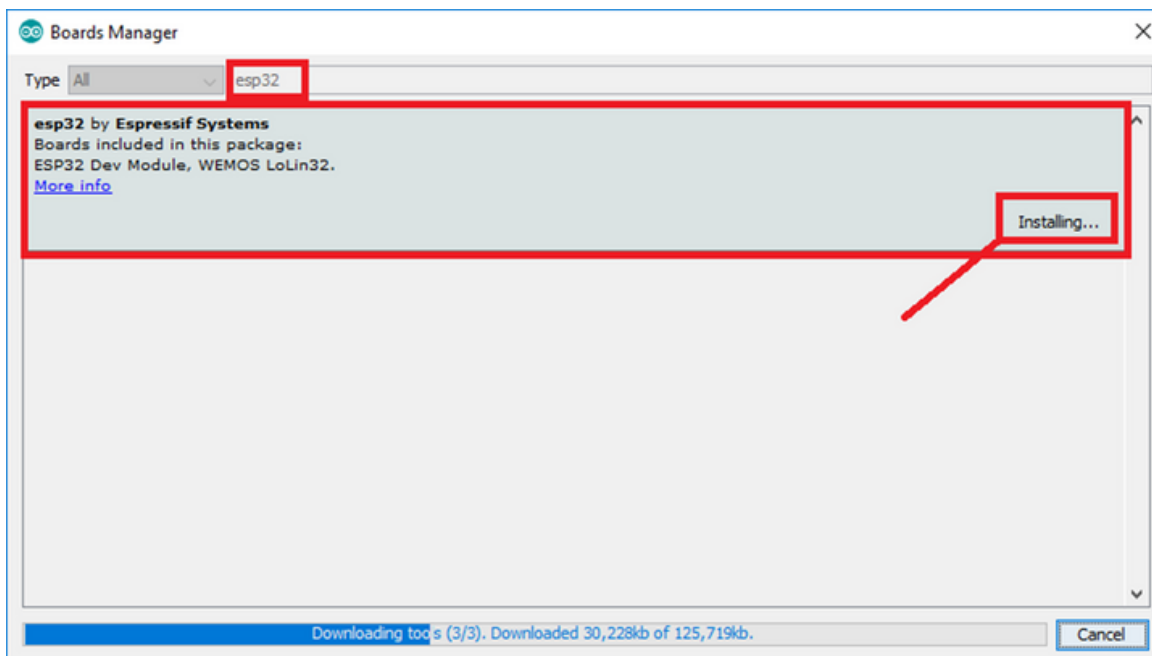
Note: if you already have the ESP8266 board's URL, you can separate the URLs with a comma as follows:

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json,
http://arduino.esp8266.com/stable/package_esp8266com_index.json

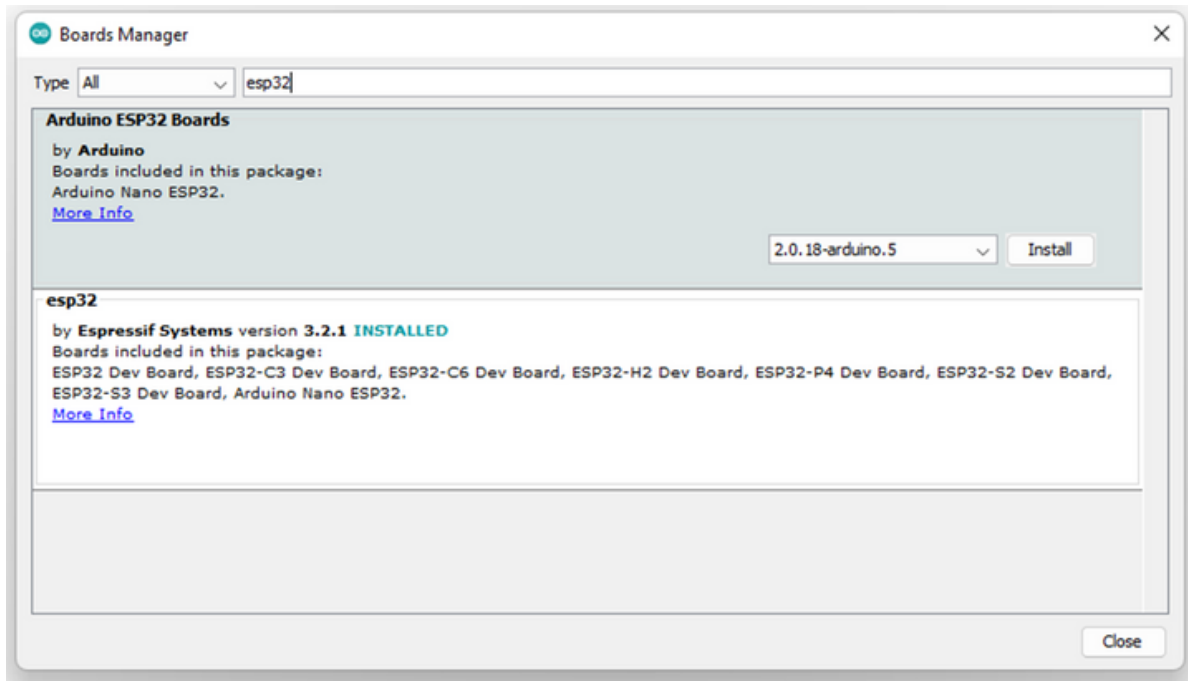
3. Open the Boards Manager. Go to **Tools > Board > Boards Manager...**



4. Search for **ESP32** and press the install button for the **"ESP32 by Espressif Systems"**:



5. That's it. It should be installed after a few seconds.



8.3 RGB LED Control

The onboard RGB LED is connected to GPIO38.

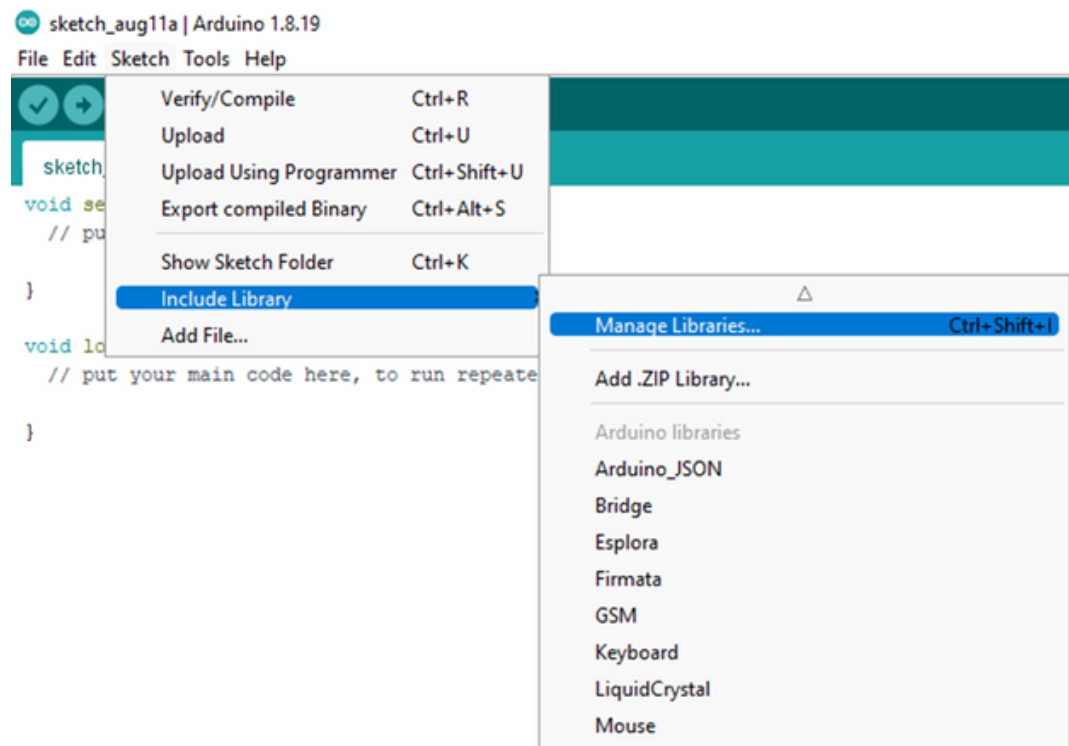
To install the **Adafruit NeoPixel** library for Arduino, follow these steps:

1. Open Arduino IDE

Launch the Arduino IDE on your computer.

2. Navigate to the Library Manager

- Go to the **Sketch** menu on the top menu bar.
- Select **Include Library > Manage Libraries...**



3. Search for the Adafruit NeoPixel Library

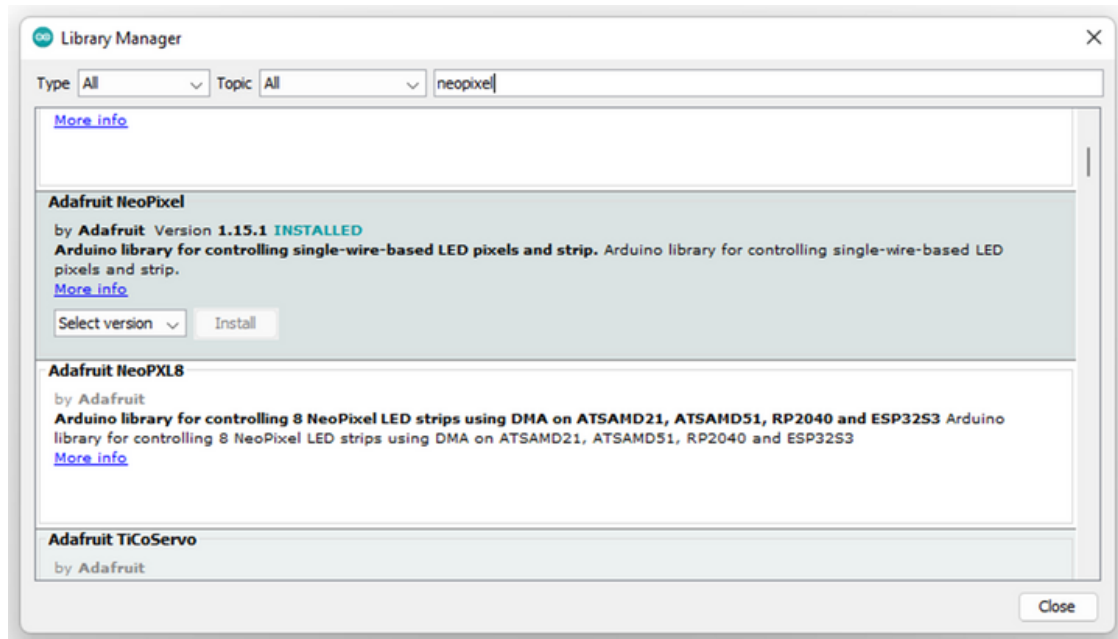
- In the Library Manager window, type "**Adafruit NeoPixel**" in the search bar at the top.
- Look for the **Adafruit NeoPixel** library by **Adafruit**.

4. Install the Library

- Click on the **Install** button next to the **Adafruit NeoPixel** library.
- Wait for the library to download and install (this may take a few moments).

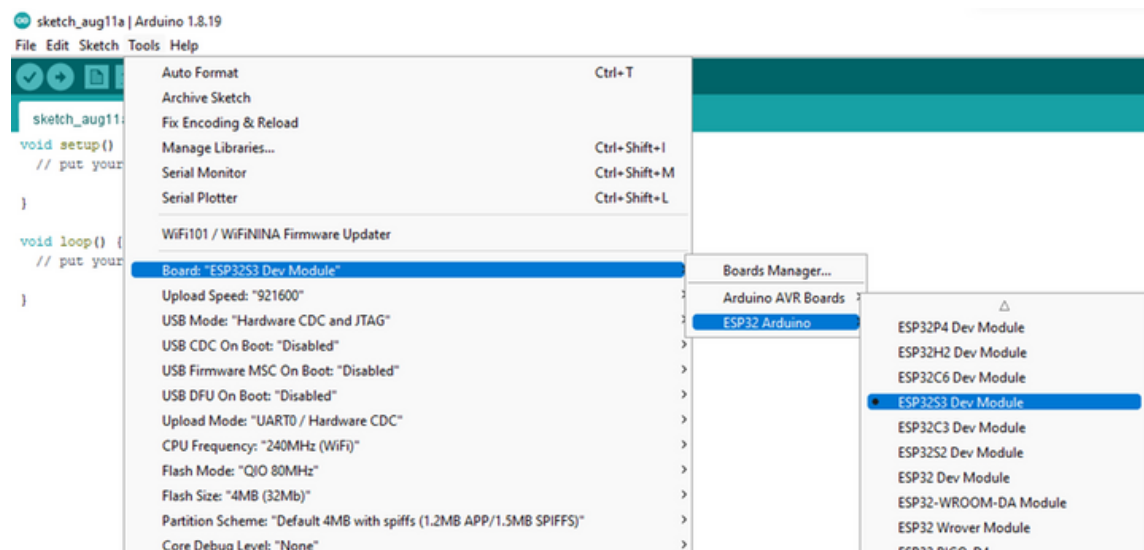
5. Close the Library Manager

- After the installation is complete, close the Library Manager.

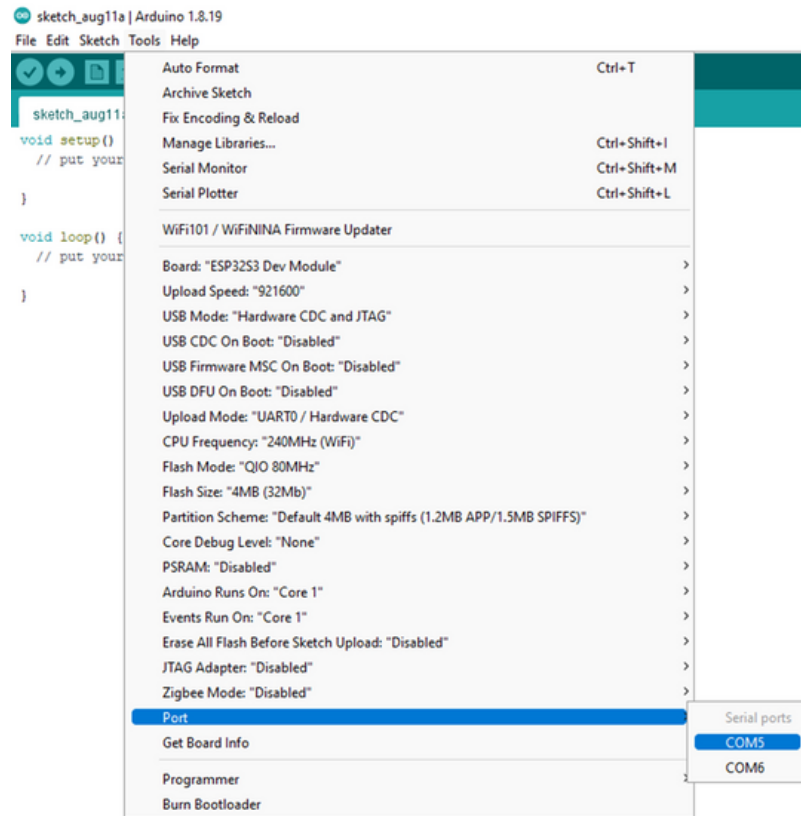


Testing the Installation, Uploading Your First Sketch to Control the RGB LED

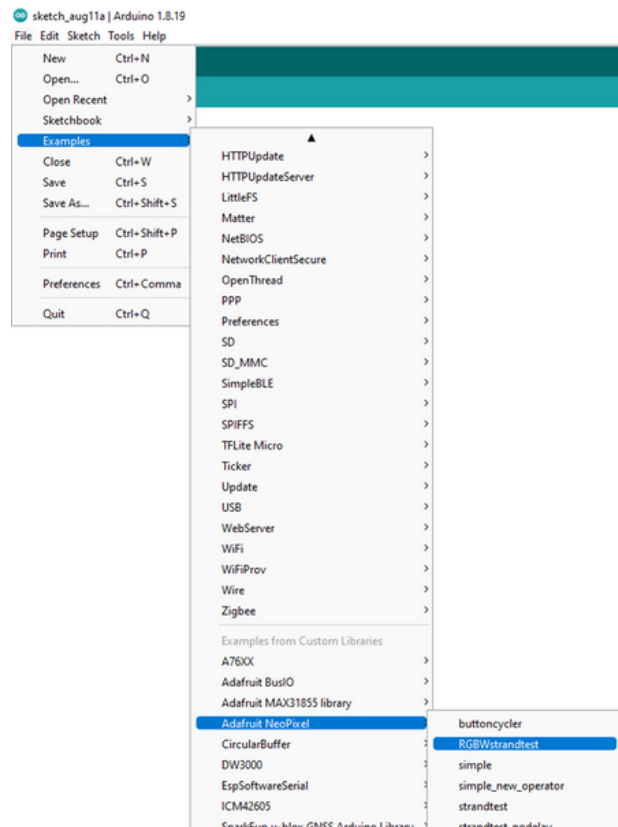
- Select your Board in **Tools > Board** menu (in my case it's the **ESP32S3 Dev Module**)



2. Go to **Tools** → **Port** and select the COM port corresponding to your board.



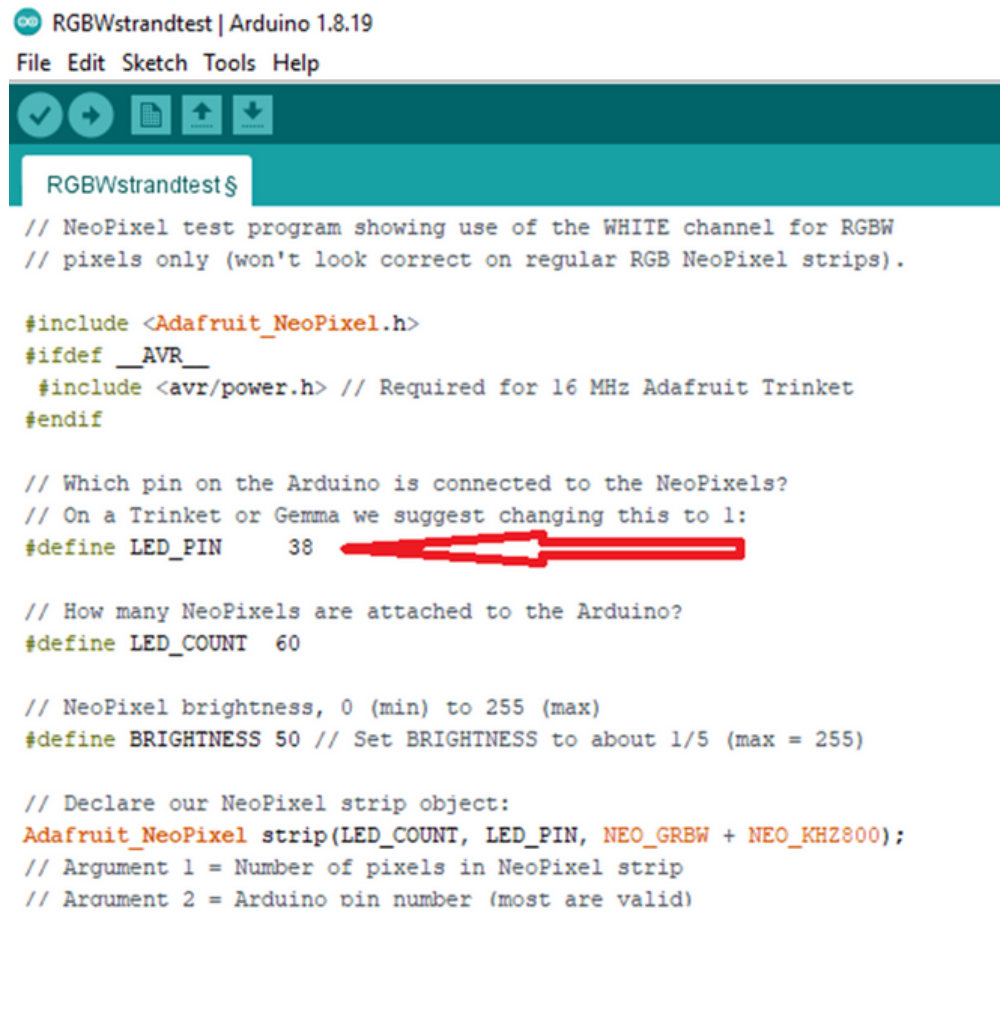
Go to **File** > **Examples** > **Adafruit NeoPixel**.
Select a simple example, such as **"strandtest"**.



Change the RGB LED PIN.

#define LED_PIN 38 as shown in pic.

Upload it to your Arduino board and see your NeoPixels light up!



```

RGBWstrandtest | Arduino 1.8.19
File Edit Sketch Tools Help

RGBWstrandtest$
// NeoPixel test program showing use of the WHITE channel for RGBW
// pixels only (won't look correct on regular RGB NeoPixel strips).

#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
#include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#endif

// Which pin on the Arduino is connected to the NeoPixels?
// On a Trinket or Gemma we suggest changing this to 1:
#define LED_PIN 38

// How many NeoPixels are attached to the Arduino?
#define LED_COUNT 60

// NeoPixel brightness, 0 (min) to 255 (max)
#define BRIGHTNESS 50 // Set BRIGHTNESS to about 1/5 (max = 255)

// Declare our NeoPixel strip object:
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRBW + NEO_KHZ800);
// Argument 1 = Number of pixels in NeoPixel strip
// Argument 2 = Arduino pin number (most are valid)

```

