ESP32 WROOM DEVELOPMENT BOARD

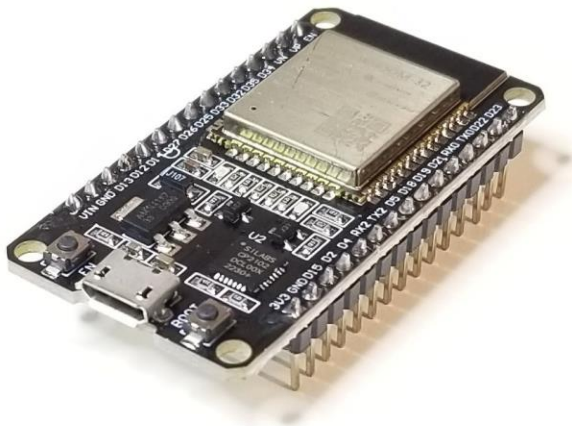
USER MANUAL

The ESP32 Development Board is a board created around ESP32 WROOM 32 chip, containing voltage regulator and USB programmer circuit for ESP32 chip, and a few many other features. For application development there is a choice between Arduino IDE or ESP-IDF (Native platform). Mostly users choose the Arduino IDE because of its simplicity and compatibility. ESP32 Development Board comes with a pre-installed firmware which allows to work with the interpreted language, sending commands through the serial port (CP2102 USB to Serial chip). The ESP32 Development board is specially designed to work on breadboard. It has a voltage regulator that allows it to feed directly from the USB port. The input/output pins work at 3.3V. The CP2102 chip is responsible for USB to serial communication.

ESP32 WROOM Development Board User Manual

Contents

Description



The ESP32 WROOM 32 series of Wi-Fi chips is produced by Espressif Systems. This module comes with ample GPIOs and support for a variety of protocols like SPI, I2C, I2S, UART, and more. The best part is that it comes with wireless networking included, which makes it different to other micro controllers like the Arduino. This means that it can easily control and monitor devices remotely via Wi-Fi and Bluetooth® at an affordable price. ESP32 WROOM 32 is a system-on-chip (SoC) integrating a 32-bit Tensilica microcontroller, standard digital peripheral interfaces, antenna switches, RF balun, power amplifier, low noise receive amplifier, filters and power management modules into a small package. It provides 2.4GHz Wi-Fi (802.11 b/g/n, supporting speeds up to 150MB/s), BLE and classic Bluetooth® wireless communication, 34 I/O pins, I2C and I2S interfaces, ADC (analog to digital conversion), DAC (digital to analog conversion), SPI interface, UART on dedicated pins, and PWM (Pulse Width Modulation). The processor core, called LX6 by Espressif, is based on Xtensa® dualcore 32-bit LX6 processor controller and runs at frequency range between 80-240MHz. It has a 448kB boot ROM, 520kB of on-chip SRAM, and 4MB of external flash memory which can be accessed through SPI interface.

Features:

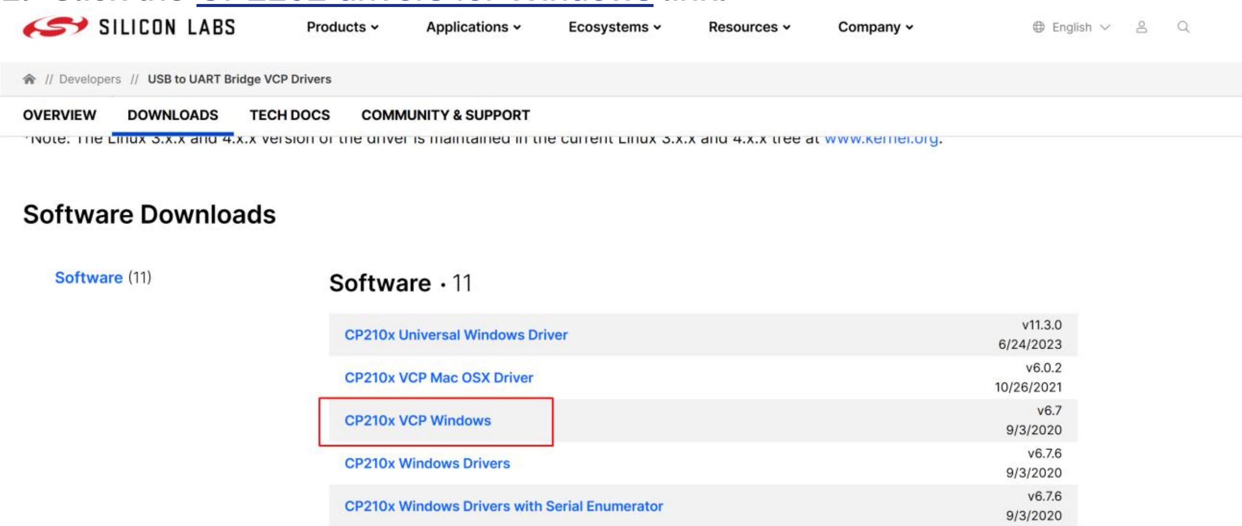
- Power supply voltage (USB) 5V DC
- Input/Output voltage 3.3V DC
- Operating current required min. 500mA
- SoC ESP32-WROOM 32
- CPU Xtensa® single-dual-core 32-bit LX6
- Clock frequency range 80MHz / 240MHz
- RAM 512kB
- External flash memory 4MB
- I/O pins 34
- ADC channels 18
- ADC Resolution 12-bit
- DAC channels 2
- DAC Resolution 8-bit
- Communication interfaces SPI, I2C, I2S, CAN, UART
- Wi-Fi protocols 802.11 b/g/n (802.11n up to 150 Mbps)
- Wi-Fi frequency 2.4 GHz - 2.5 GHz
- Bluetooth V4.2 - BLE and Classic Bluetooth
- Wireless antena PCB
- Dimensions 56x28x13mm(2.2x1.1x0.5in)

USB Driver

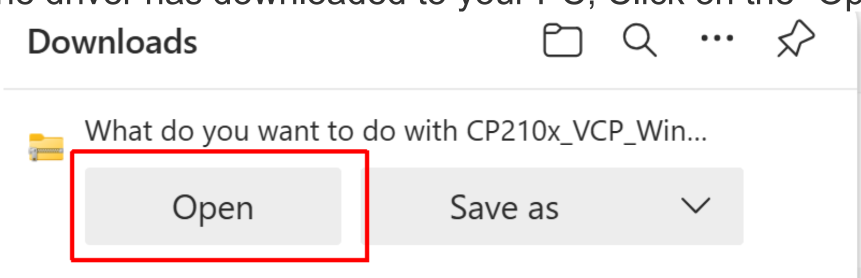
Windows Drivers and Installation Instructions

Before you connect your ESP32 WROOM Development Board CP2102 device to a computer running Microsoft Windows, you should install its drivers:

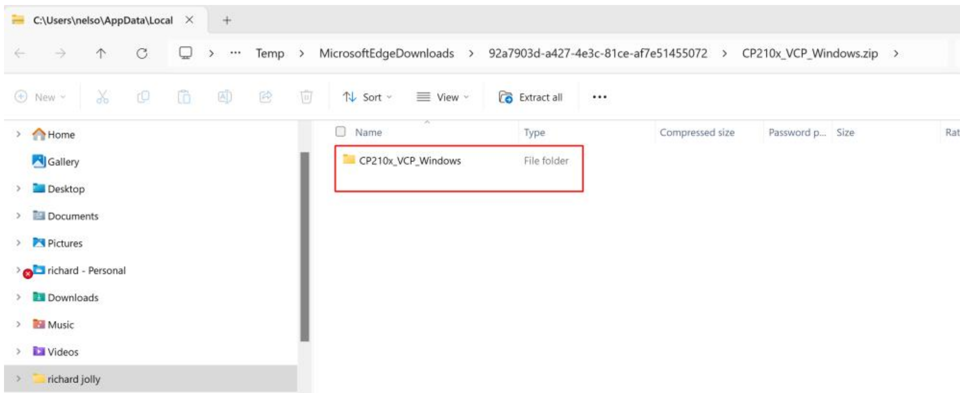
1. Click the CP2102 drivers for Windows link.



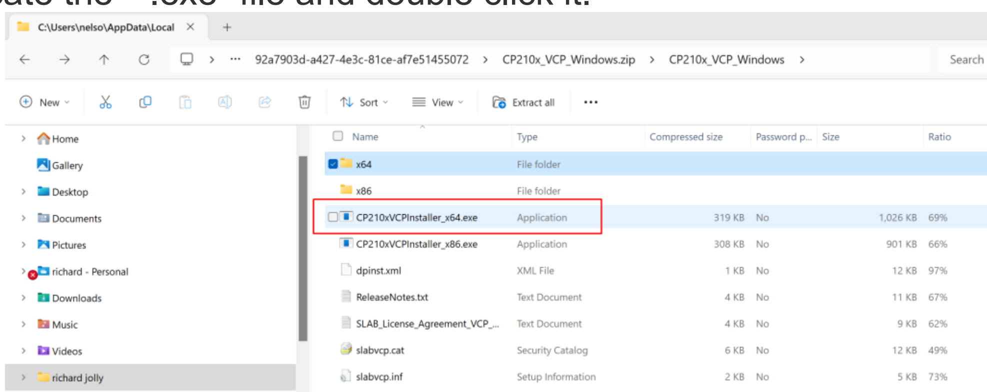
2. Click on the “CP210x VCP Windows” link
3. When the driver has downloaded to your PC, Click on the “Open” button



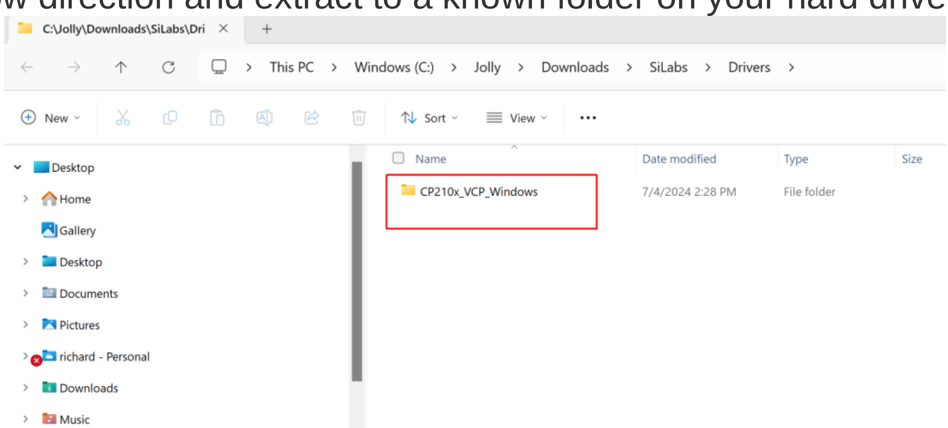
4. Once the file is opened, click on the folder



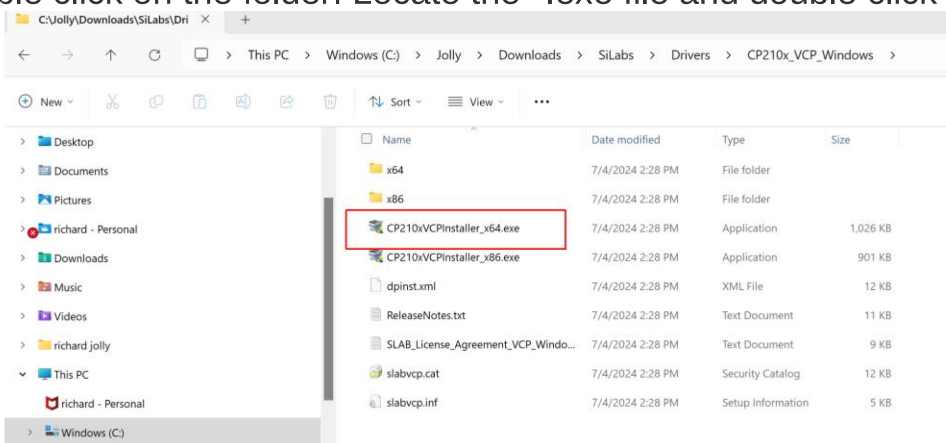
5. Locate the “*.exe” file and double click it.



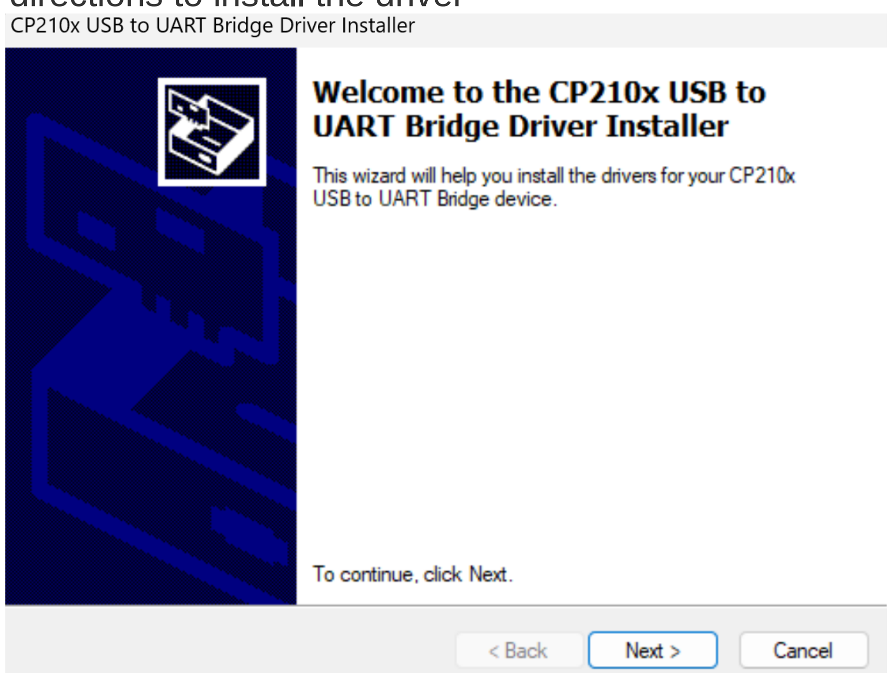
6. Follow direction and extract to a known folder on your hard drive



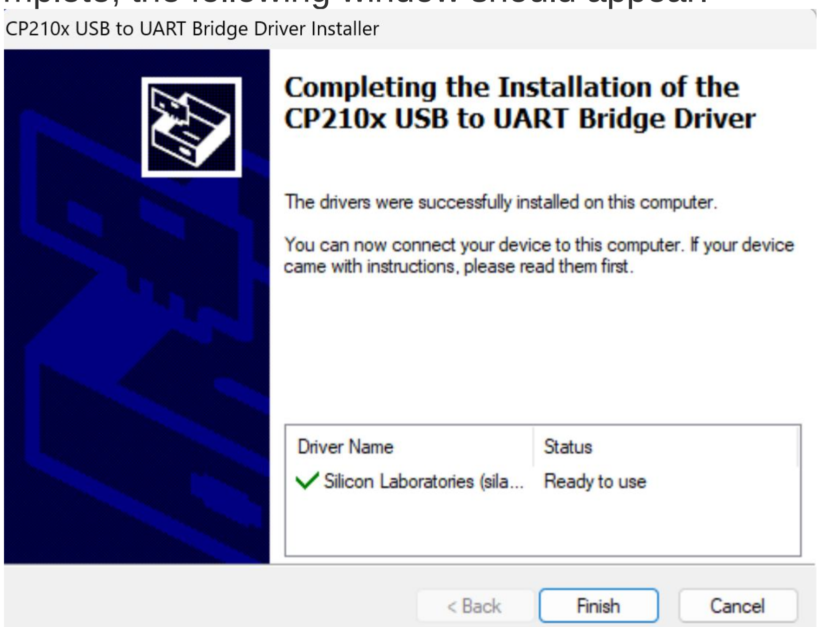
7. Double click on the folder. Locate the *.exe file and double click



8. Follow directions to install the driver



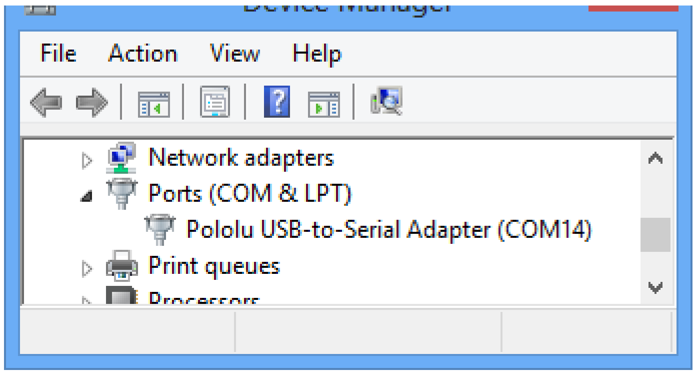
9. Once, complete, the following window should appear.



10. Driver is installed and ready to go.

Windows 11, Windows 10, users: Your computer should now automatically install the necessary drivers when you connect a ESP32 WROOM Development Board CP2102 device. No further action from you is required.

If you now go to your computer’s Device Manager and expand the “Ports (COM & LPT)” list, you should see “Pololu USB-to-Serial Adapter” as one of the COM ports.



Windows 10 device manager showing a ESP32 Development Board CP2102 device.

Some software will not allow connection to higher COM port numbers. If you need to change the COM port number assigned to your USB adapter, you can do so using the Windows device manager. Bring up the properties dialog for the "ESP32 WROOM Development Board USB-to-Serial Adapter" COM port and click the "Advanced…" button in the "Port Settings" tab. From this dialog you can change the COM port assigned to your device.

3. Mac OS X Drivers

To use the CP2102 on a computer running Mac OS X, you should download and install the **CP210x USB to UART Bridge VCP Drivers from Silicon Labs** for Mac OS X.

Additionally, we have older drivers that other customers have used successfully on both PowerPC- and Intel-based Macs running OS X version 10.3.9 or later, but these probably will not work on version 10.7 or later. You can download these drivers here: OSX_cp2102_drivers.dmg (723k dmg)

Mac OS X compatibility: We have confirmed that our CP2102-based devices work on Mac OS X 10.7 and other customers have used them successfully on older versions. We can assist with advanced technical issues, but most of our tech support staff does not use Macs, so basic support for Mac OS X is limited.

4. Linux Installation

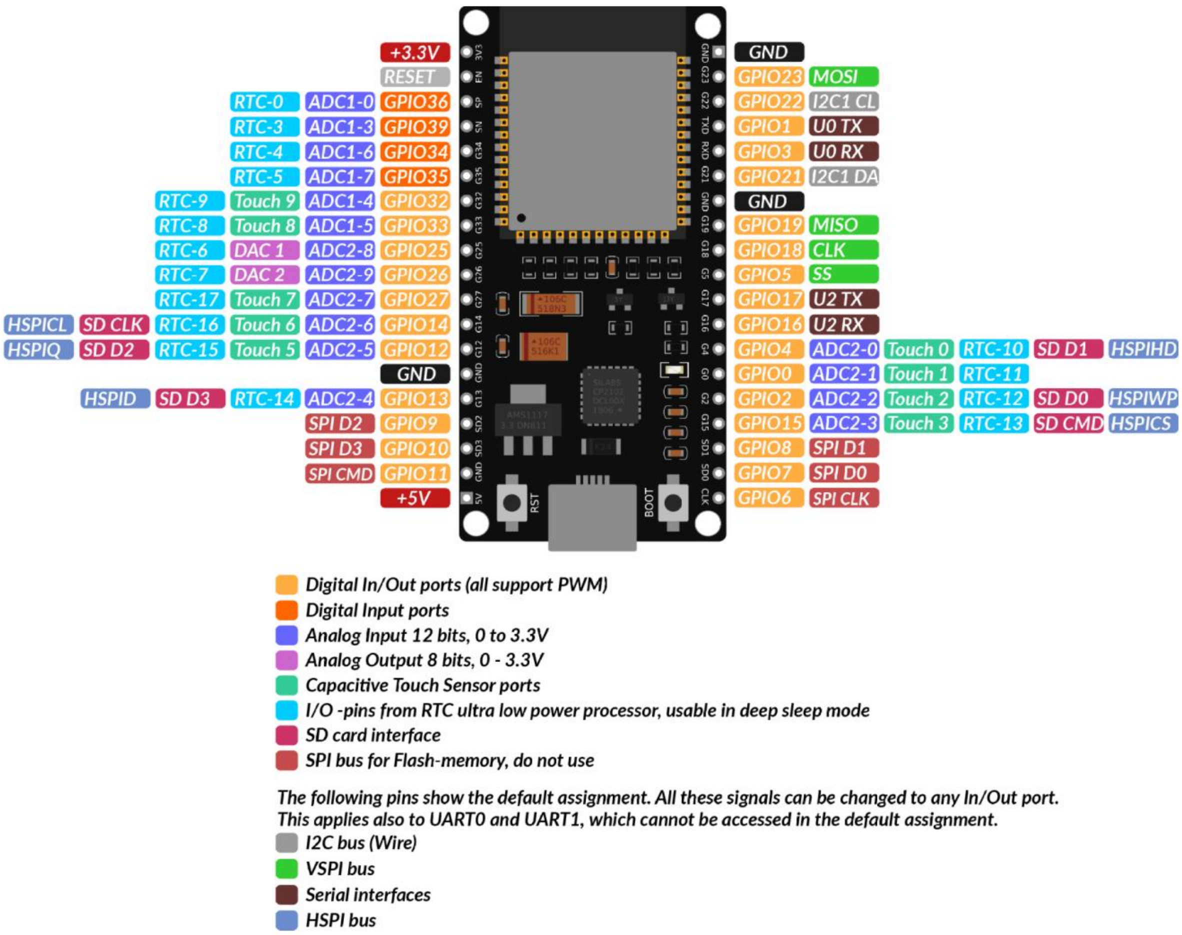
Recent versions of the linux kernel include support for the CP2102 USB-to-UART Bridge Controller as part of the usb-serial driver, so your ESP32 WROOM Development Board CP2102 should work right out of the box. We have verified that the CP2102 works with Ubuntu versions 7.04 and 8.04. If you experience any problems, we recommend you upgrade to the most recent version of your distribution.

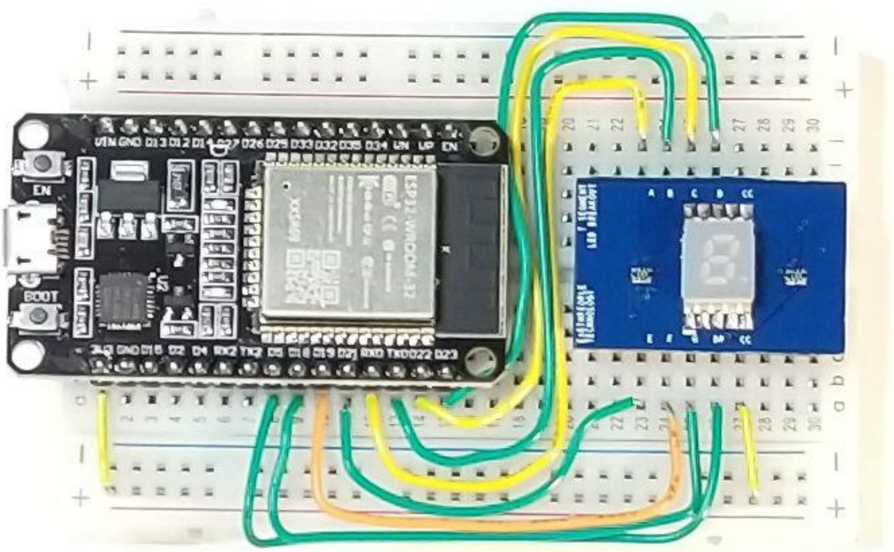
Hardware Details

ESP32 WROOM 32 is a system-on-chip (SoC) integrating a 32-bit Tensilica microcontroller, standard digital peripheral interfaces, antenna switches, RF balun, power amplifier, low noise receive amplifier, filters and power management modules into a small package. It provides 2.4GHz Wi-Fi (802.11 b/g/n, supporting speeds up to 150MB/s), BLE and classic Bluetooth® wireless communication, 34 I/O pins, I2C and I2S interfaces, ADC (analog to digital conversion), DAC (digital to analog conversion), SPI interface, UART on dedicated pins, and PWM (Pulse Width Modulation). The processor core, called LX6 by Espressif, is based on Xtensa® dualcore 32-bit LX6 processor controller and runs at frequency range between 80-240MHz. It has a 448kB boot ROM, 520kB of on-chip SRAM, and 4MB of external flash memory which can be accessed through SPI interface.

Pinout

The ESP32 Dev Kit C V2 has 38 pins. The pinout is shown on the following image:


+3.3V
RESET
RTC-0 ADC1-0 GPIO36
RTC-3 ADC1-3 GPIO39
RTC-4 ADC1-6 GPIO34
RTC-5 ADC1-7 GPIO35
RTC-9 Touch 9 ADC1-4 GPIO32
RTC-8 Touch 8 ADC1-5 GPIO33
RTC-6 DAC 1 ADC2-8 GPIO25
RTC-7 DAC 2 ADC2-9 GPIO26
RTC-17 Touch 7 ADC2-7 GPIO27
HSPICL SD CLK RTC-16 Touch 6 ADC2-6 GPIO14
HSPIQ SD D2 RTC-15 Touch 5 ADC2-5 GPIO12
GND
HSPID SD D3 RTC-14 ADC2-4 GPIO13
SPI D2 GPIO9
SPI D3 GPIO10
SPI CMD GPIO11
+5V
GND
GPIO23 MOSI
GPIO22 I2C1 CL
GPIO1 U0 TX
GPIO3 U0 RX
GPIO21 I2C1 DA
GND
GPIO19 MISO
GPIO18 CLK
GPIO5 SS
GPIO17 U2 TX
GPIO16 U2 RX
GPIO4 ADC2-0 Touch 0 RTC-10 SD D1 HSPIHD
GPIO0 ADC2-1 Touch 1 RTC-11
GPIO2 ADC2-2 Touch 2 RTC-12 SD D0 HSPIWP
GPIO15 ADC2-3 Touch 3 RTC-13 SD CMD HSPICS
GPIO8 SPI D1
GPIO7 SPI D0
GPIO6 SPI CLK
RST
BOOT
Digital In/Out ports (all support PWM)
Digital Input ports
Analog Input 12 bits, 0 to 3.3V
Analog Output 8 bits, 0 - 3.3V
Capacitive Touch Sensor ports
I/O -pins from RTC ultra low power processor, usable in deep sleep mode
SD card interface
SPI bus for Flash-memory, do not use
The following pins show the default assignment. All these signals can be changed to any In/Out port.
This applies also to UART0 and UART1, which cannot be accessed in the default assignment.
I2C bus (Wire)
VSPI bus
Serial interfaces
HSPI bus



NOTE: The absolute maximum current drawn per one GPIO is 40mA according to the "Recommended Operating Conditions" section in theESP32 datasheet.

Pins description

Just like a normal Arduino board, the ESP32 Dev Kit C V2 has digital input/output pins (GPIO pins - General Purpose Input/Output pins). These digital input/outputs operate at 3.3V.

5V voltage must not be connected to any ESP32 chip pins! The pins are not 5V tolerant, applying more than 3.3V on any pin will destroy the chip. he GPIO pins 34 to 39 are GPIs – input only pins. These pins do not have internal pull-ups or pull-down resistors. They cannot be used as outputs, so use these pins only as inputs: GPIO 34, GPIO 35, GPIO 36, GPIO 39 There is an integrated SPI flash on the ESP-WROOM-32 chip. The pins GPIO6 to GPIO 11 are exposed in certain ESP32 development boards. These pins are connected to the integrated SPI flash on the chip and are not recommended for other uses.

GPIO 6 (SCK/CLK), GPIO 7 (SDO/SD0), GPIO 8 (SDI/SD1), GPIO 9 (SHD/SD2), GPIO 10 (SWP/SD3), GPIO 11 (CSC/CMD).

Capacitive Touch sensor pins

The ESP32 has 10 internal capacitive touch sensors. The capacitive touch pins can also be used to wake up the ESP32 from deep sleep. These internal touch sensors are connected to these GPIOs: T0 (GPIO 4), T1 (GPIO 0), T2 (GPIO 2), T3 (GPIO 15), T4 (GPIO 13), T5(GPIO 12), T6

(GPIO 14), T7 (GPIO 27), T8 (GPIO 33), T9 (GPIO 32).

Analog to Digial converter pins

The ESP32 has 18x12 bits ADC (Analog to Digital converter) input channels

(while the ESP8266 only has 1x 10 bits ADC). These are the GPIOs that

can be used as ADC and respective channels:

ADC1_CH0 (GPIO 36), ADC1_CH1 (GPIO 37), ADC1_CH2 (GPIO 38),

ADC1_CH3 (GPIO 39), ADC1_CH4 (GPIO 32), ADC1_CH5 (GPIO 33),

ADC1_CH6 (GPIO 34), ADC1_CH7 (GPIO 35), ADC2_CH0 (GPIO 4),

ADC2_CH1 (GPIO 0), ADC2_CH2 (GPIO 2), ADC2_CH3 (GPIO 15),

ADC2_CH4 (GPIO 13), ADC2_CH5 (GPIO 12), ADC2_CH6 (GPIO 14),

ADC2_CH7 (GPIO 27), ADC2_CH8 (GPIO 25), ADC2_CH9 (GPIO 26).

Digital to Analog converter pins

There are 2 x 8 bits DAC (Digital to Analog converter) channels on the ESP32 to convert digital signals into analog voltage signal outputs. These are the DAC channels:

DAC1 (GPIO25), DAC2 (GPIO26).

Real Time Clock GPIO pins

There is RTC (Real time clock) GPIO support on the ESP32. The GPIOs routed to the RTC low-power subsystem can be used when the ESP32 is in deep sleep. These RTC GPIOs can be used to wake up the ESP32 from deep sleep when the Ultra Low Power (ULP) co-processor is running. The following GPIOs can be used as an external wake up source: RTC_GPIO0 (GPIO36), RTC_GPIO3 (GPIO39), RTC_GPIO4 (GPIO34), RTC_GPIO5 (GPIO35), RTC_GPIO6 (GPIO25), RTC_GPIO7 (GPIO26), RTC_GPIO8 (GPIO33), RTC_GPIO9 (GPIO32), RTC_GPIO10 (GPIO4), RTC_GPIO11 (GPIO0), RTC_GPIO12 (GPIO2), RTC_GPIO13 (GPIO15), RTC_GPIO14 (GPIO13), RTC_GPIO15 (GPIO12), RTC_GPIO16 (GPIO14), RTC_GPIO17 (GPIO27).

PWM (Pulse Width Modulation) pins

The ESP32 LED PWM (Pulse width modulation) controller has 16 independent channels that can be configured to generate PWM signals with different properties. All pins that can act as outputs can be used as PWM pins (GPIOs 34 to 39 cannot generate PWM). To set a PWM signal, you need to define these parameters in the code: Signal’s frequency, Duty cycle, PWM channel, GPIO where you want to output the signal.

The I2C interface pins

The ESP32 has two I2C channels and any pin can be set as SDA or SCL. When using the ESP32 with the Arduino IDE, the default I2C pins are:

GPIO 21 (SDA), GPIO 22 (SCL).

SPI inteface pins

By default, the pin mapping for SPI pins is:

SPI MOSI MISO CLK CS VSPI GPIO 23 GPIO 19 GPIO 18 GPIO 5 HSPI GPIO 13 GPIO 12 GPIO 14 GPIO 15

Strapping Pins

Following pins are used to put the ESP32 into bootloader or flashing mode:

GPIO 0, GPIO 2, GPIO 4, GPIO 5 (must be HIGH during boot), GPIO 12 (must be LOW during boot), GPIO 15 (must be HIGH during boot).

Most development boards put the pins in the right state for flashing or boot mode. If some peripherals are connected to the strapping pins and the IDE is unable to upload the code or flash the ESP32, it may be because those peripherals are preventing the ESP32 to enter the right mode. After resetting, flashing, or booting, those pins work as expected. There is Boot Mode Selection documentation guide on the following link. Further and more extensive explanations are not in the scope of this eBook so please, refer to the datasheet.

Pins HIGH at Boot

Some GPIOs change their state to HIGH or output PWM signals at boot or reset. This means that if outputs are connected to these GPIOs this may get unexpected results when the ESP32 resets or boots.

GPIO 1, GPIO 3, GPIO 5, GPIO 6 to GPIO 11 (connected to the ESP32 integrated SPI flash memory - not recommended for use),

GPIO 14, GPIO 15.

Enable (EN) Enable (EN) is the 3.3V regulator's enable pin. It has a pulled up state and it

needs to be connected to ground to disable the 3.3V regulator. This means that this pin can be connected to a push button to restart your ESP32, for example.

USB to Serial communication

The ESP32 Dev Kit C V2 has a microUSB connection port. It is made around CP21202 chip made by Silicon Laboratories which allows USB to UART serial comunication. The chip has the virtual COM port (VCP) feature that appears as COM port in PC applications. The CP2102 UART interface implements all RS-232 signals, including control and handshaking signals, so existing system firmware does not need to be modified. To be able to use the ESP32 the driver has to be installed.

WiFi Communication

ESP32 Dev Kit C V2 has integrated Wi-Fi communication interface and can operate in three different modes: Wi-Fi station, Wi-Fi access point, and both at the same time. It supports the following features:

- 802.11b and 802.11g data-rates
- 802.11n MCS0-7 in both 20MHz and 40MHz bandwidth
- 802.11n MCS32
- 802.11n 0.4µS guard-interval
- Data-rate up to 150 Mbps
- Receiving STBC 2x1

- Up to 21 dBm transmitting power
- Adjustable transmitting power
- Antenna diversity and selection (software-managed hardware)

Bluetooth Communication

The ESP32 Dev Kit C V2 has an integrated Bluetooth Radio and supports following features:

- Class-1, class-2 and class-3 transmit output powers and over 30 dB dynamic control range
- π/4 DQPSK and 8 DPSK modulation
- High performance in NZIF receiver sensitivity with over 98 dB dynamic range
- Class-1 operation without external PA
- Internal SRAM allows full speed data transfer, mixed voice and data, and full piconet operation
- Logic for forward error correction, header error control, access code correlation, CRC, demodulation, encryption bit stream generation, whitening and transmit pulse shaping
- ACL, SCO, eSCO and AFH
- A-law, µ-law and CVSD digital audio CODEC in PCM interface
- SBC audio CODEC
- Power management for low power applications
- SMP with 128-bit AES

Also, the Bluetooh Radio has support for the following communication

interface protocols:

- UART HCI interface, up to 4 Mbps
- SDIO / SPI HCI interface
- I2C interface
- PCM / I2S audio interface.

Other features

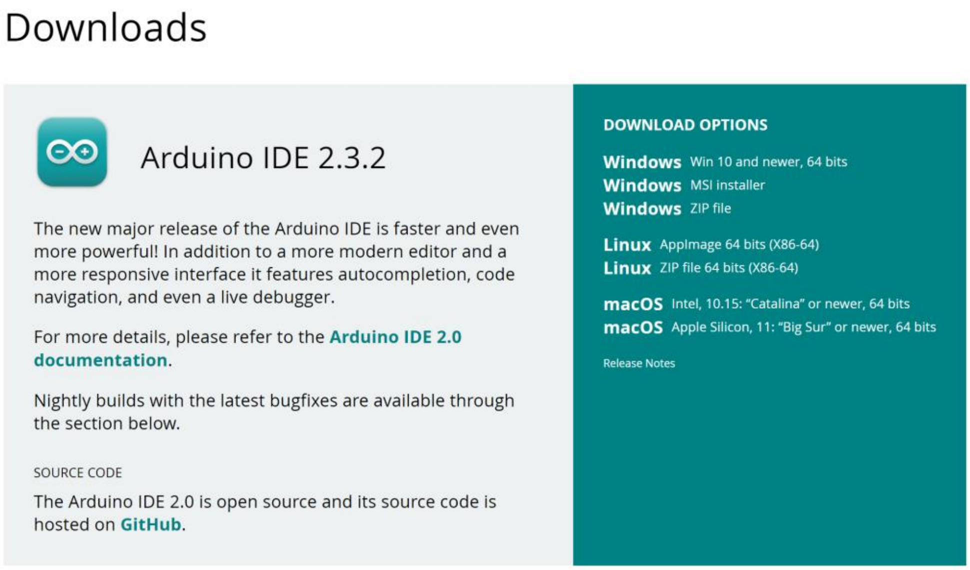
ESP32-WROOM 32D chip has an integrated Hall Effect Sensor that detects changes in the magnetic field in its surroundings.

The Hall sensor is based on an N-carrier resistor. When the chip is in the magnetic field, the Hall sensor develops a small voltage on the resistor, which can be directly measured by the analog-digital converter (ADC), or amplified by the ultra low noise analog pre-amplifier and then measured by the ADC.

The temperature sensor generates a voltage that varies with temperature. The voltage is internally converted via an analog-to-digital converter into a digital code. The temperature sensor has a range of -40°C to 125°C. As the offset of the temperature sensor varies from chip to chip due to process variation, together with the heat generated by the Wi-Fi circuitry itself (which affects measurements), the internal temperature sensor is only suitable for applications that detect temperature changes instead of absolute temperatures and for calibration purposes as well. However, if the user calibrates the temperature sensor and uses the device in a minimally powered-on application, the results could be accurate enough.

How to set-up Arduino IDE

If the Arduino IDE is not installed, follow the link and download the installation file for the operating system of choice. The Arduino IDE version used for this User Manual is 2.3.2



For Windows users, double click on the downloaded .exe file and follow the instructions in the installation window.

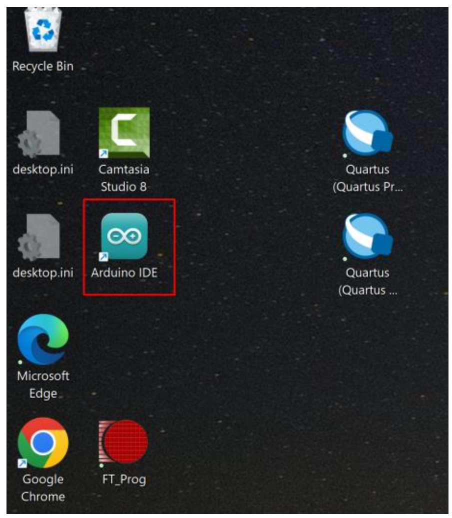
For Linux users, download a file with the extension .tar.xz, which has to be extracted. When it is extracted, go to the extracted directory and open the terminal in that directory. Two .sh scripts have to be executed, the first called arduino-linux-setup.sh and the second called install.sh. To run the first script in the terminal, open the terminal in the extracted directory and run the following command:

sh arduino-linux-setup.sh user_name

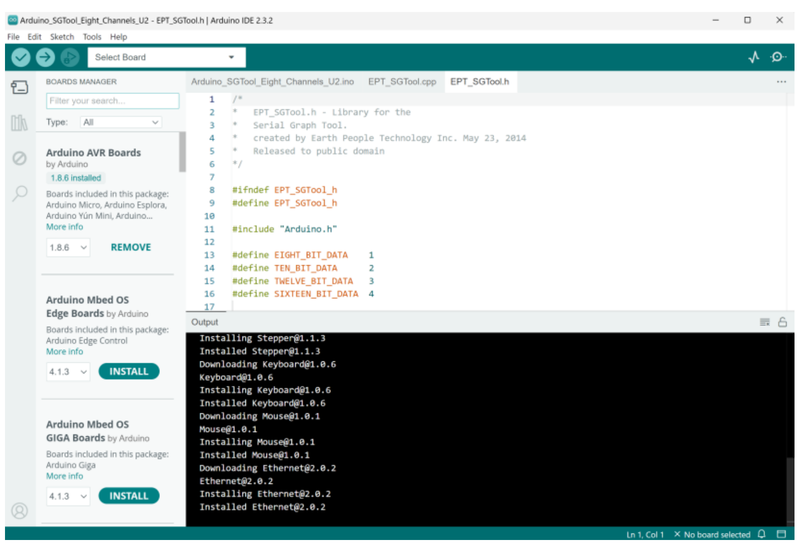
user_name - is the name of a superuser in the Linux operating system. A password for the superuser has to be entered when the command is started. Wait for a few minutes for the script to complete everything.

The second script, called install.sh, has to be used after the installation of the first script. Run the following command in the terminal (extracted directory): sh install.sh

After the installation of these scripts, go to the All Apps, where the Arduino IDE is installed. Click on the Arduino IDE icon.

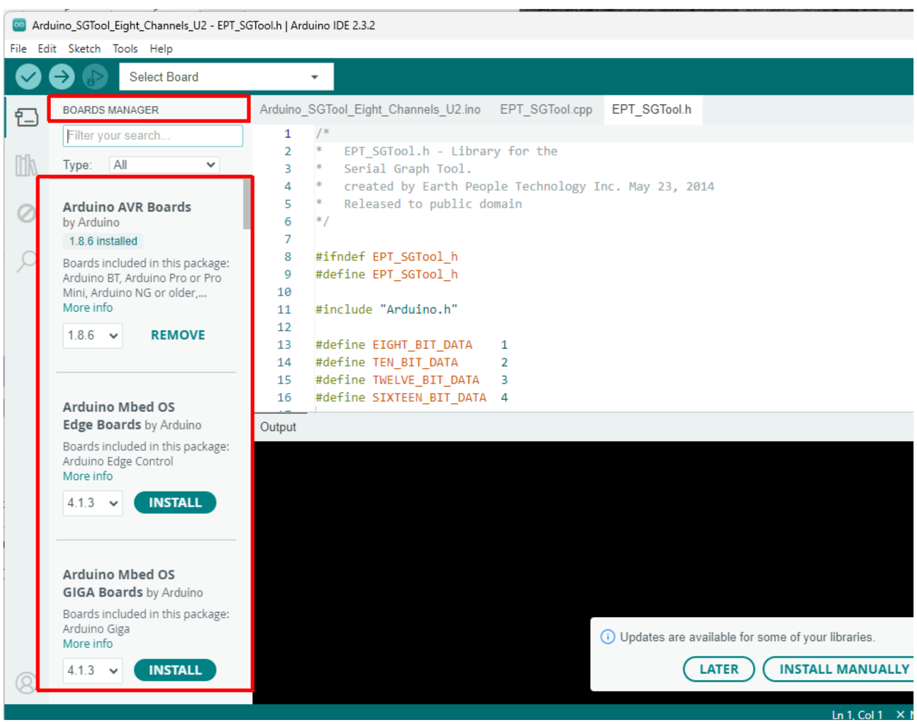


The Arduino IDE will open.

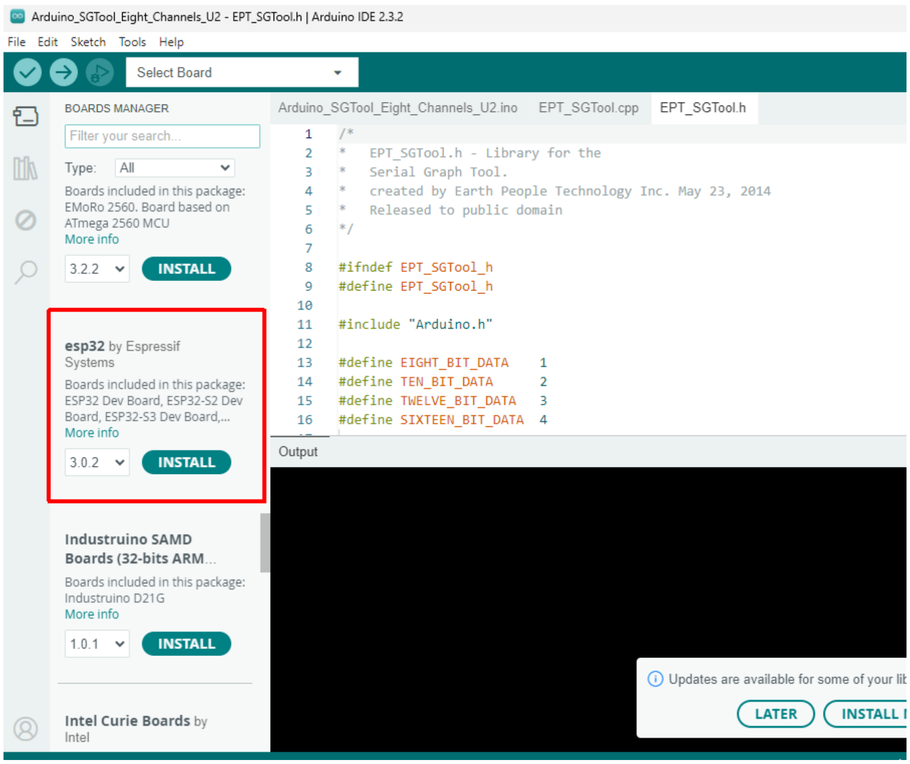


Almost all operating systems come with a text editor preinstalled (for example, Windows comes with Notepad, Linux Ubuntu comes with Gedit, Linux Raspbian comes with Leafpad, etc.). All of these text editors are perfectly fine for the purpose of the User Manual.

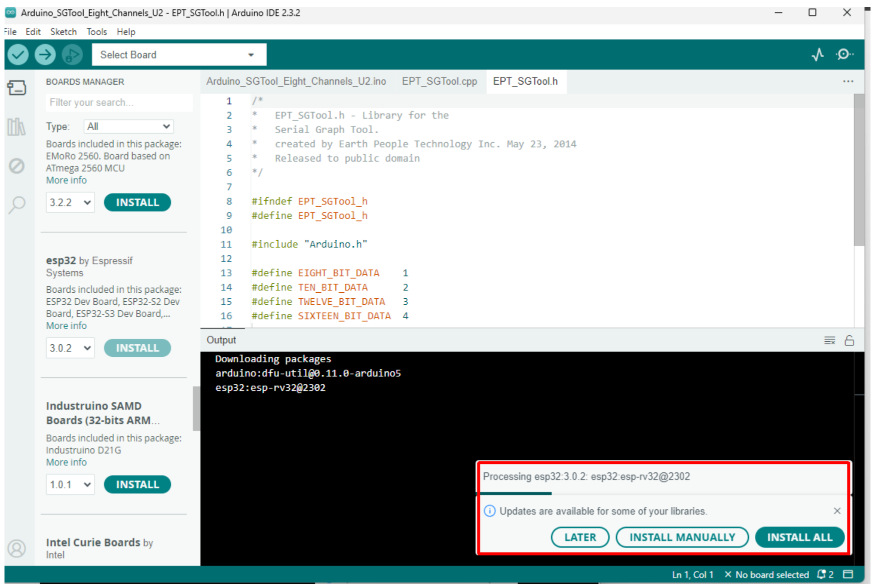
Next, to install support for the ESP32 platform, open Arduino IDE and go to the left side of the the window.



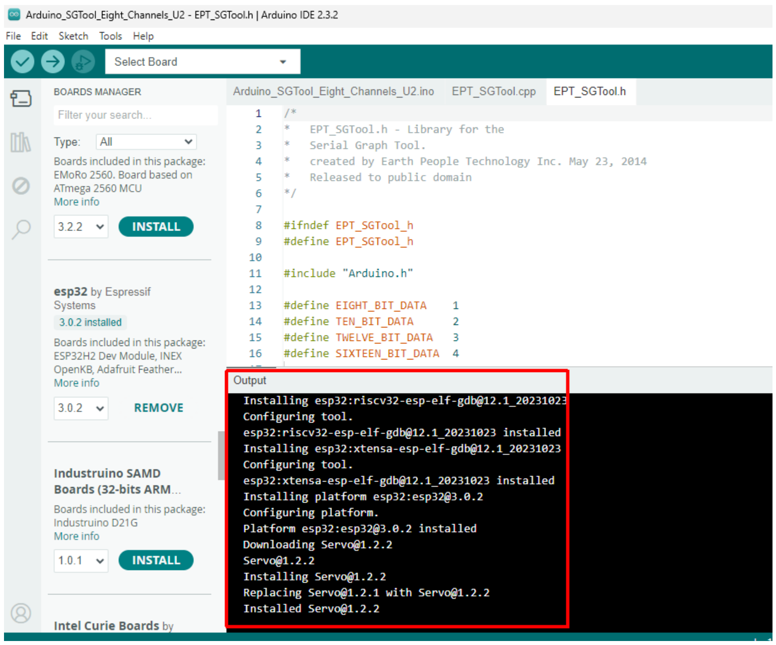
Under “Board Manager”, scroll down to find the “esp32 by Espressiff” tab on the left side.


Arduino_SGTool_Eight_Channels_U2 - EPT_SGTool.h | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Select Board
BOARDS MANAGER
Filter your search...
Type: All
Boards included in this package: EMoRo 2560. Board based on ATmega 2560 MCU
More info
3.2.2
INSTALL
esp32 by Espressif Systems
Boards included in this package: ESP32 Dev Board, ESP32-S2 Dev Board, ESP32-S3 Dev Board,...
More info
3.0.2
INSTALL
Industruino SAMD Boards (32-bits ARM...
Boards included in this package: Industruino D21G
More info
1.0.1
INSTALL
Intel Curie Boards by Intel
Arduino_SGTool_Eight_Channels_U2.ino
EPT_SGTool.cpp
EPT_SGTool.h
/*
* EPT_SGTool.h - Library for the
* Serial Graph Tool.
* created by Earth People Technology Inc. May 23, 2014
* Released to public domain
*/
#ifndef EPT_SGTool_h
#define EPT_SGTool_h
#include "Arduino.h"
#define EIGHT_BIT_DATA 1
#define TEN_BIT_DATA 2
#define TWELVE_BIT_DATA 3
#define SIXTEEN_BIT_DATA 4
Output
Updates are available for some of your lib
LATER
INSTALL

Click “Install”. The library packages will automatically install.

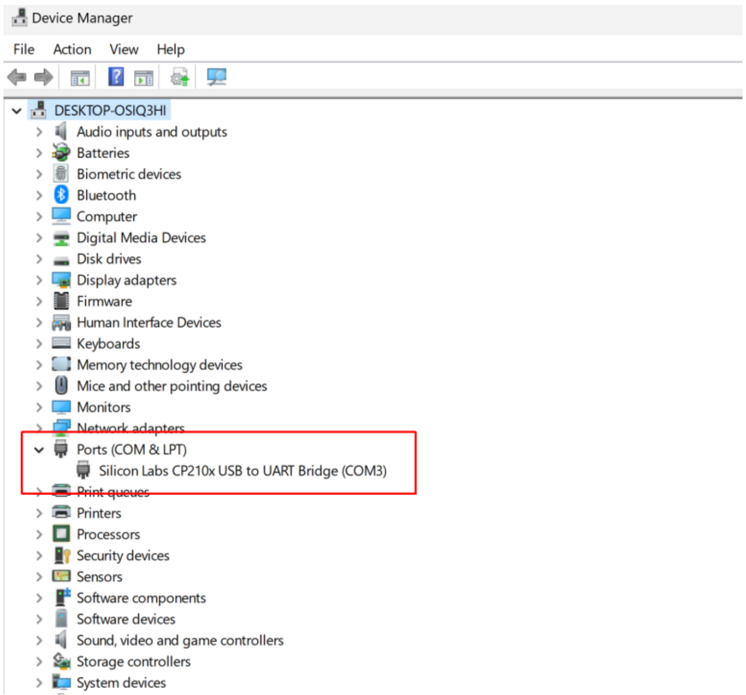


Once all the library packages are installed, you will see the following message:

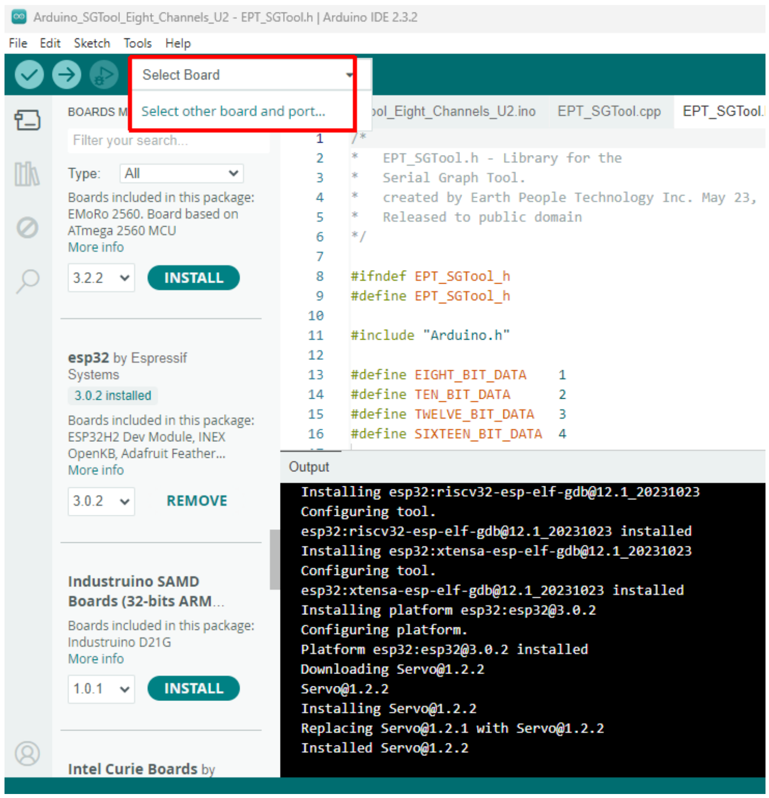


Next, plug in the ESP32 WROOM Development Board into an available USB port:

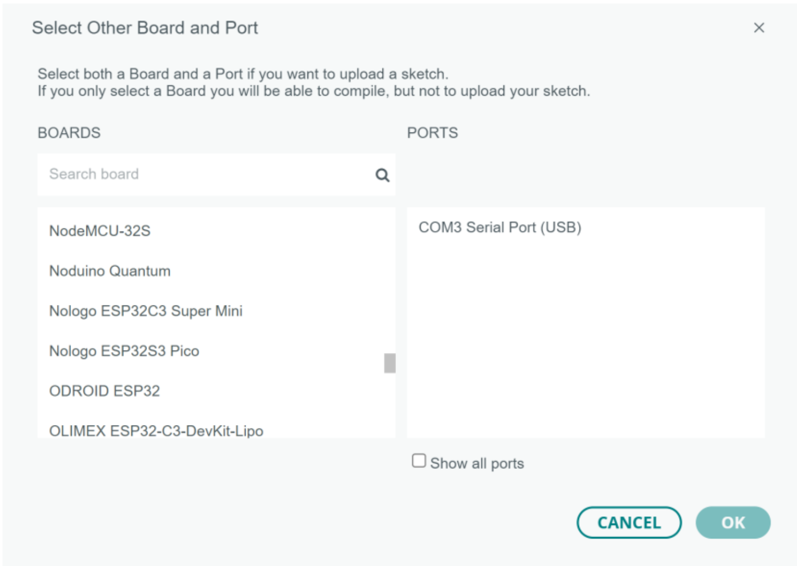
If the CP2102 driver was properly loaded in the previous steps, you should the COM Port properly enumerated in the Device Manager.

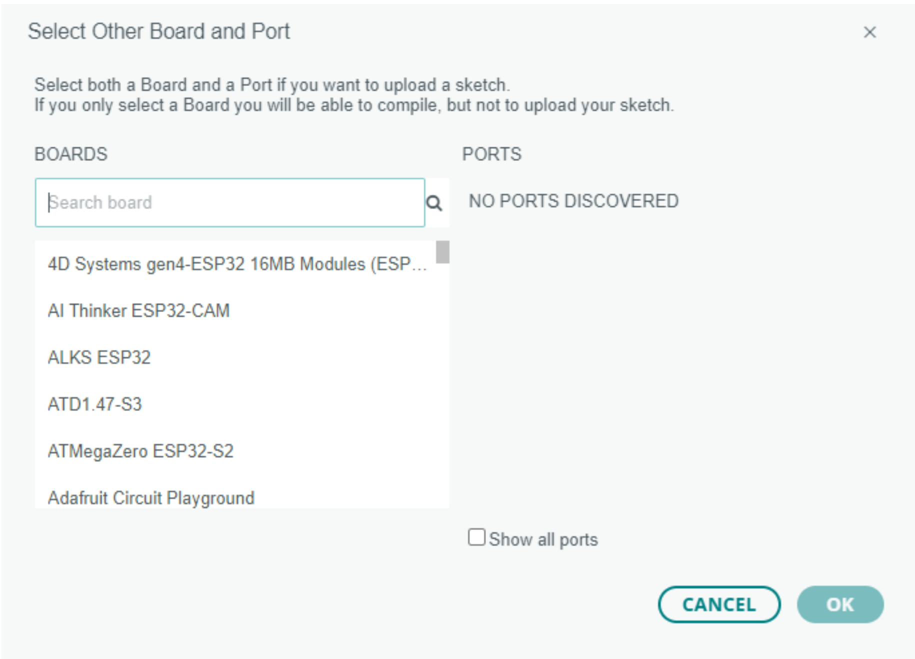


Then, go to the Arduino IDE and click on the drop down menu at the “Select Board”

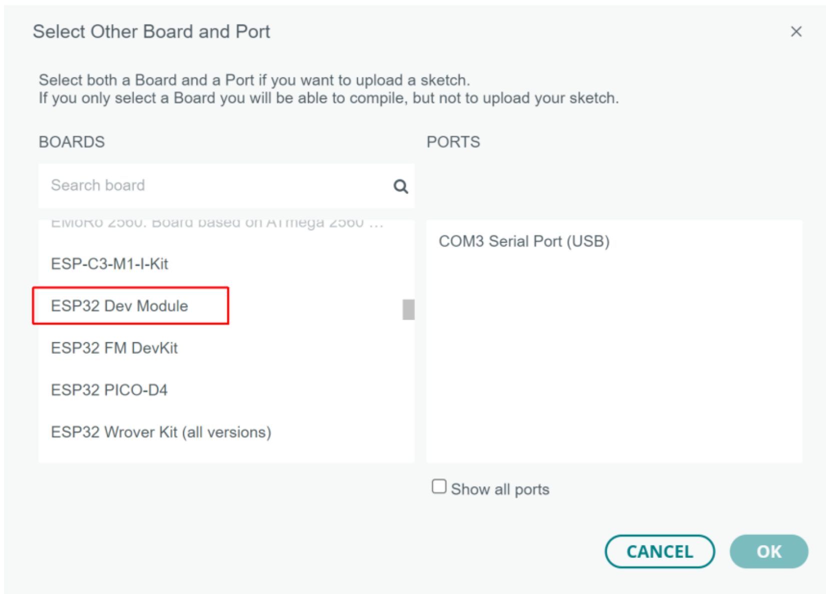


Click on the “Select other board and port”. At the “Select Other Board and Port” window,

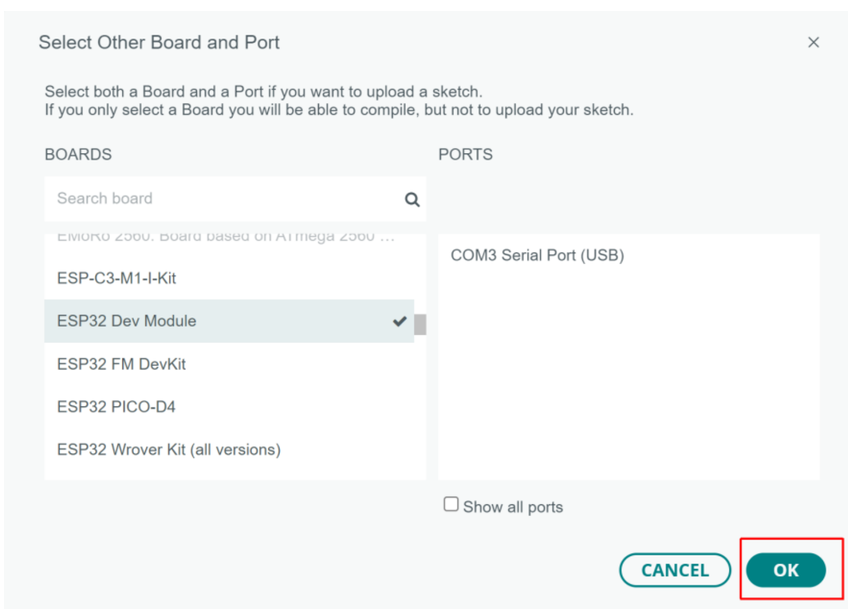




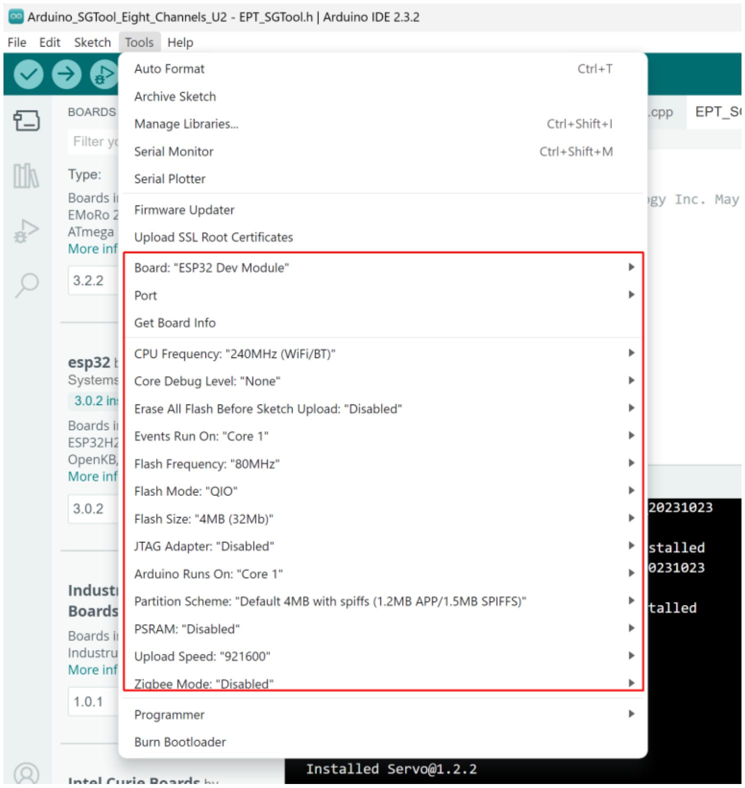
Scroll down to the ESP32 Dev Module



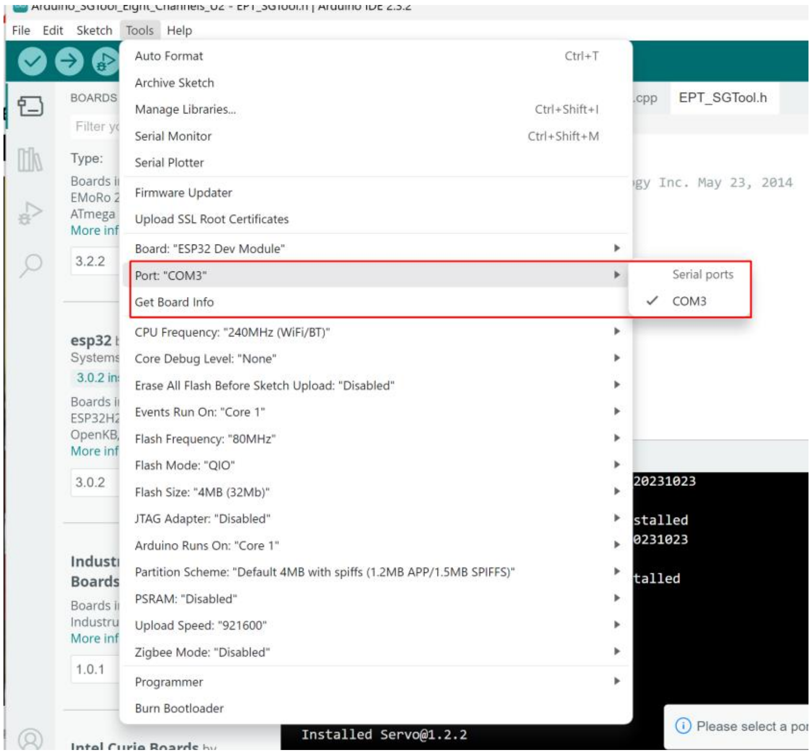
Select the Board and click on "Ok"



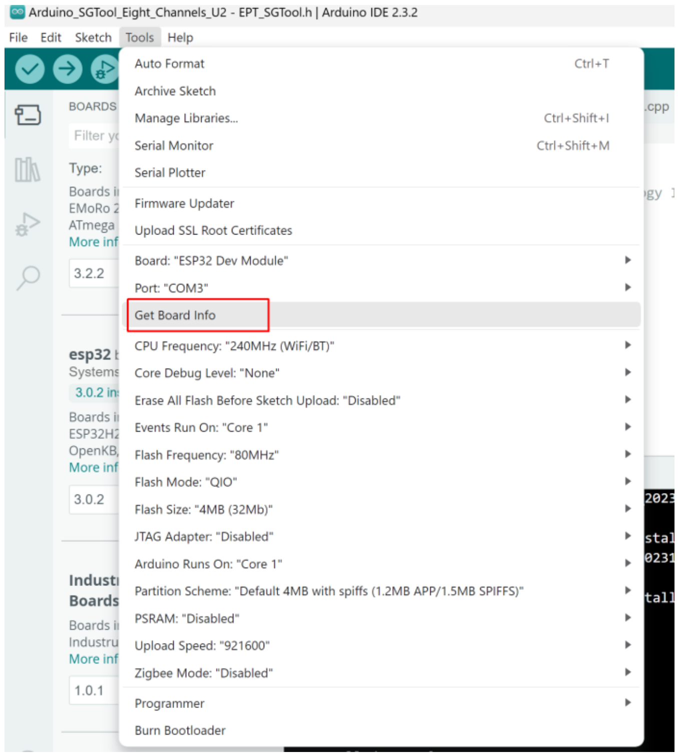
Next, under the Arduino IDE, click on "Tools" and ensure the following is read.



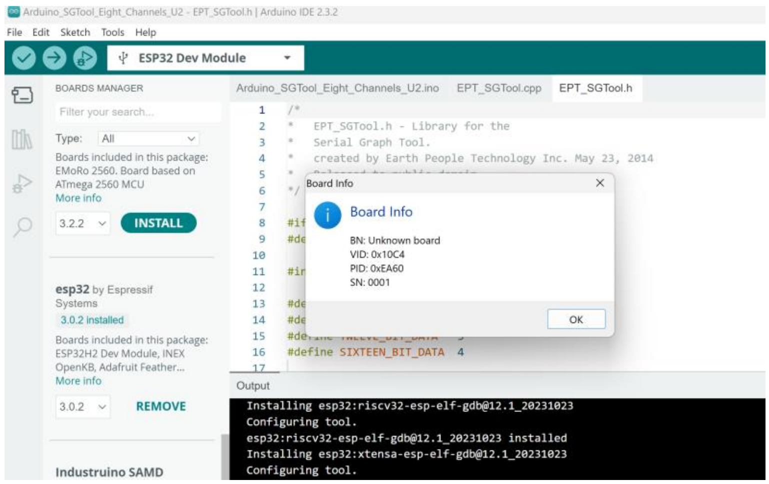
Under "Tools", select "Port" and select the CP2102 COM Port number found under Device Manager.



Next, under “Tools” select the “Get Board Info”.

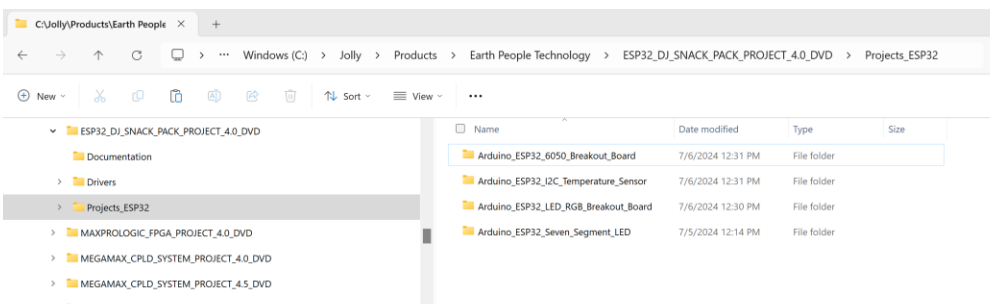


You should see a similar message with the Board VID and PID numbers.

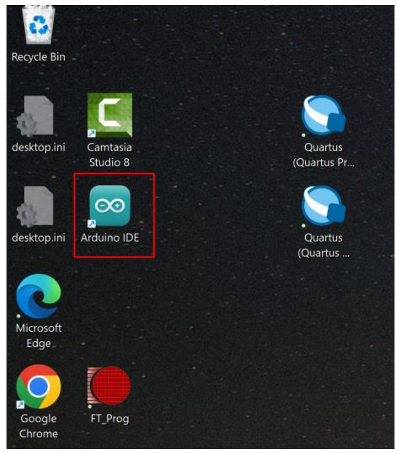


Building Your First ESP32 Project

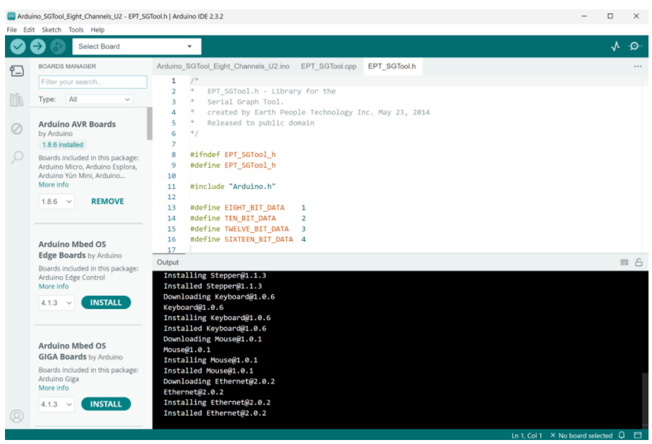
Go to the Project DVD and locate the Arduino ESP32 project folder.



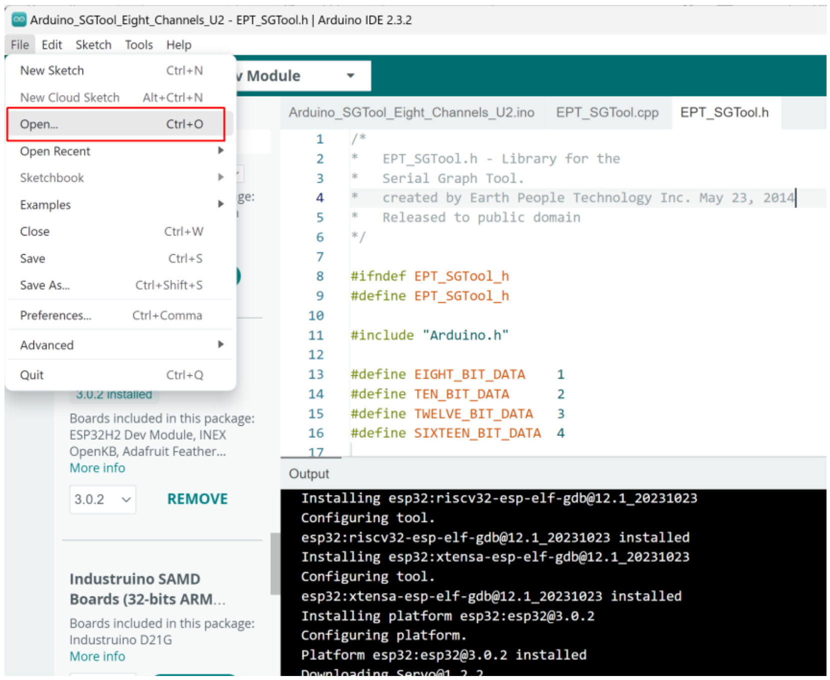
Open the Arduino IDE by clicking in the Icon.



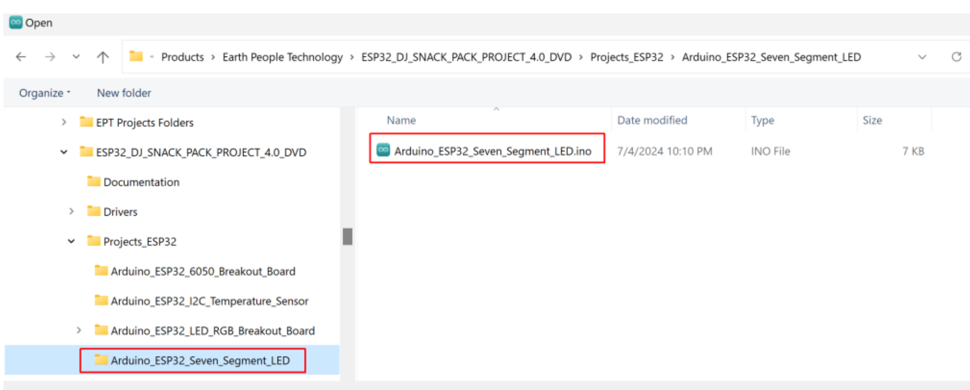
The Arduino IDE will open.



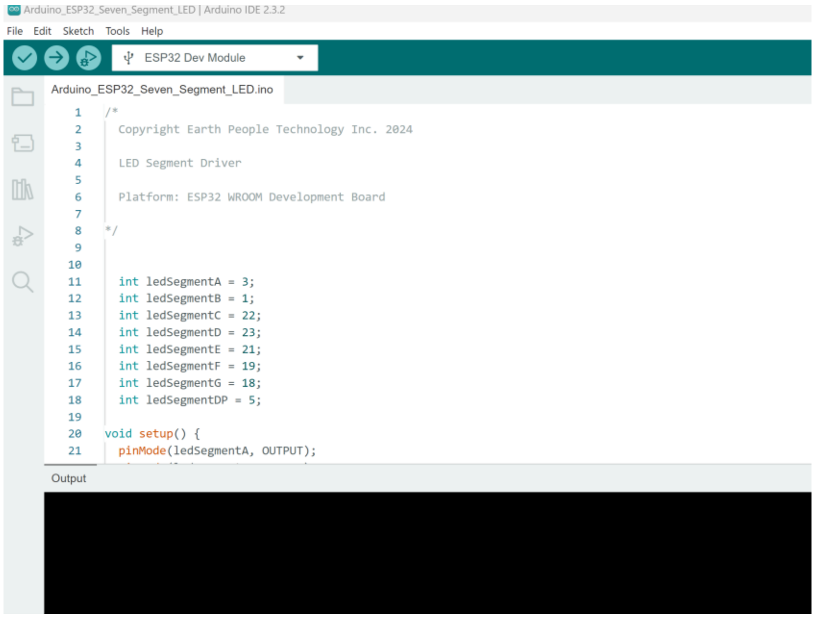
Click on “File” ->”Open”



In the dialog box, locate the "Arduino_ESP32_Seven_Segment_LED" project in the Project DVD.

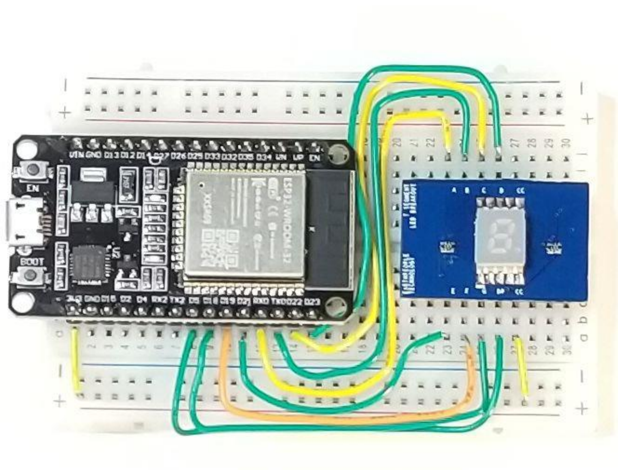


The Arduino_ESP32_Seven_Segment_LED project will load.

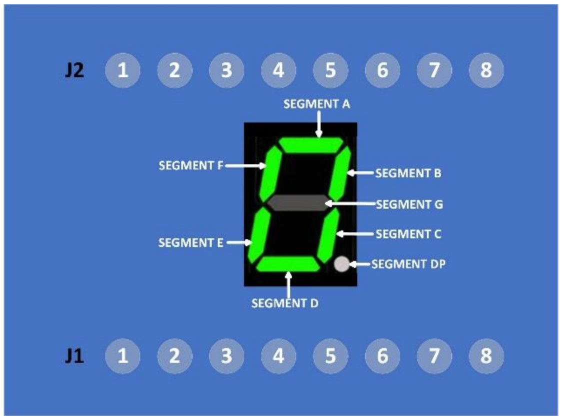


Use the following items to build the hardware:

- ESP32 WROOM Development Board
- EPT Seven Segment LED Driver Board
- 180 Breakout Board
- 140 Piece Wire Kit
-



Connect the following:

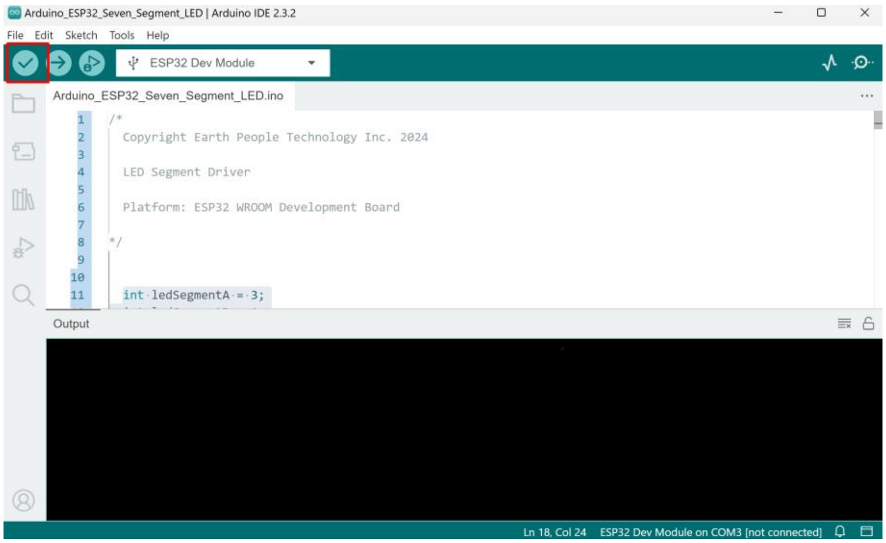


Seven Segment Connector	Pin #	Signal	Pin #	ESP32 WROOM
J1	1	SEGMENT_A	3	
	2	SEGMENT_B	1	
	3	SEGMENT_C	22	
	4	SEGMENT_D	23	
	5	+3.3V	+3.3V	
	6	NC	NC	
	7	NC	NC	
	8	Ground	NC	

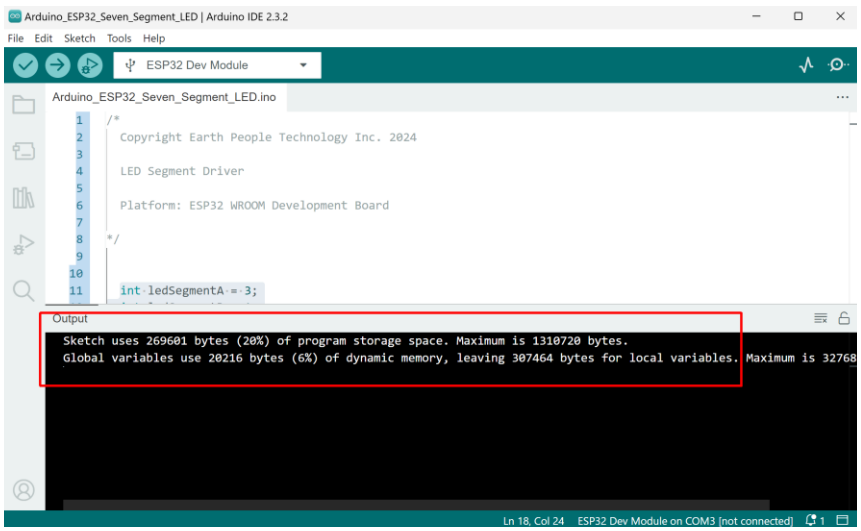
Connector	Pin #	Signal	Pin #	ESP32 WROOM
J2	1	SEGMENT_E	21	
	2	SEGMENT_F	19	
	3	SEGMENT_G	18	
	4	SEGMENT_DP	5	

	5	+3.3V	NC	
	6	NC	NC	
	7	NC	NC	
	8	Ground	NC	

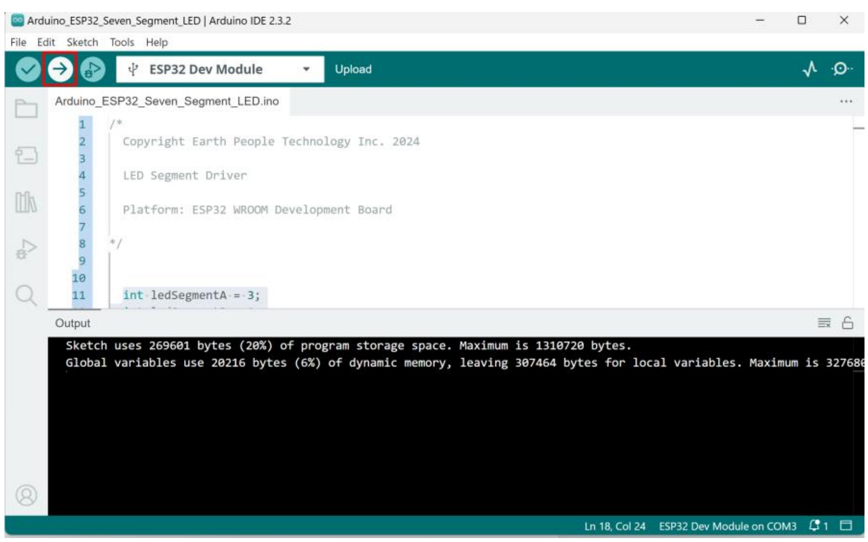
Next, Compile the project



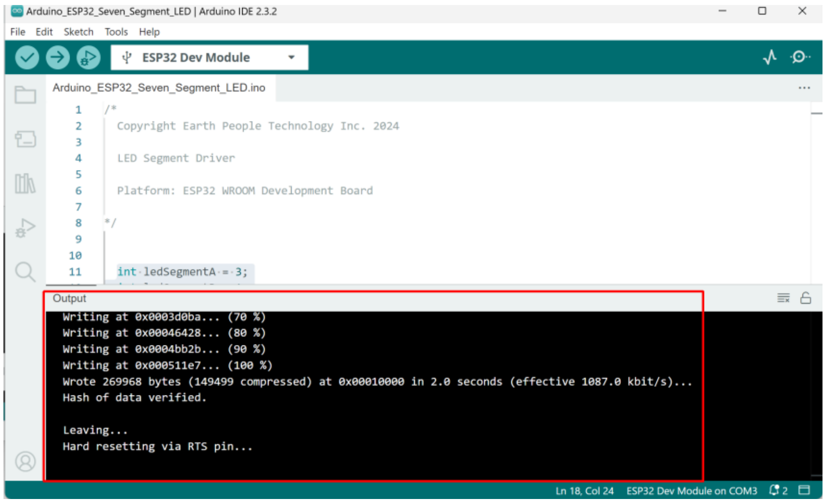
Ensure the code has compiled without errors, Output should be similar:



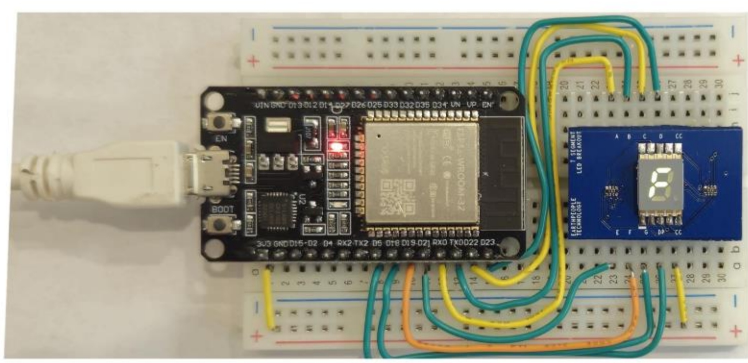
Upload the code



IF the Upload is successful, the output should be similar:



Review the output of the project



Try the next project