

Physical Computing using Tektork IoT Kit (Arduino ESP32 Based)

Installation of latest Arduino IDE:

Visit [Software | Arduino](https://www.arduino.cc/en/software) (<https://www.arduino.cc/en/software>) and download

Arduino IDE 2.2.1


The screenshot shows the Arduino website's software page. The top navigation bar includes links for PROFESSIONAL, EDUCATION, and STORE, along with a search bar. Below this, a secondary navigation bar highlights HARDWARE, SOFTWARE, CLOUD, DOCUMENTATION, COMMUNITY, BLOG, and ABOUT. The main content area features a section for the Arduino Web Editor, a preview of the IDE interface, and a 'Downloads' section. The 'Downloads' section includes the Arduino IDE 2.2.1 logo, a description of the new release, and a list of download options for Windows, Linux, and macOS. A 'Release Notes' link is also present.

Arduino Web Editor

Start coding online and save your sketches in the cloud. The most up-to-date version of the IDE includes all libraries and also supports new Arduino boards.

[CODE ONLINE](#) [GETTING STARTED](#)

Downloads



Arduino IDE 2.2.1

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.

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, 10.14: "Mojave" or newer, 64 bits
macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits

Windows MSI installer

Windows ZIP file

Linux Appliance 64 bits (X86-64)

Linux ZIP file 64 bits (X86-64)

macOS Intel, 10.14: "Mojave" or newer, 64 bits

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

[Release Notes](#)

Support the Arduino IDE

Since the release 1.x release in March 2015, the Arduino IDE has been downloaded **76,309,646** times — impressive! Help its development with a donation.

\$3

\$5

\$10

\$25

\$50

Other




















JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

Click Just DOWNLOAD. Arduino-ide_2.2.1_windows_64bit.zip will be downloaded.

Extract this zip.

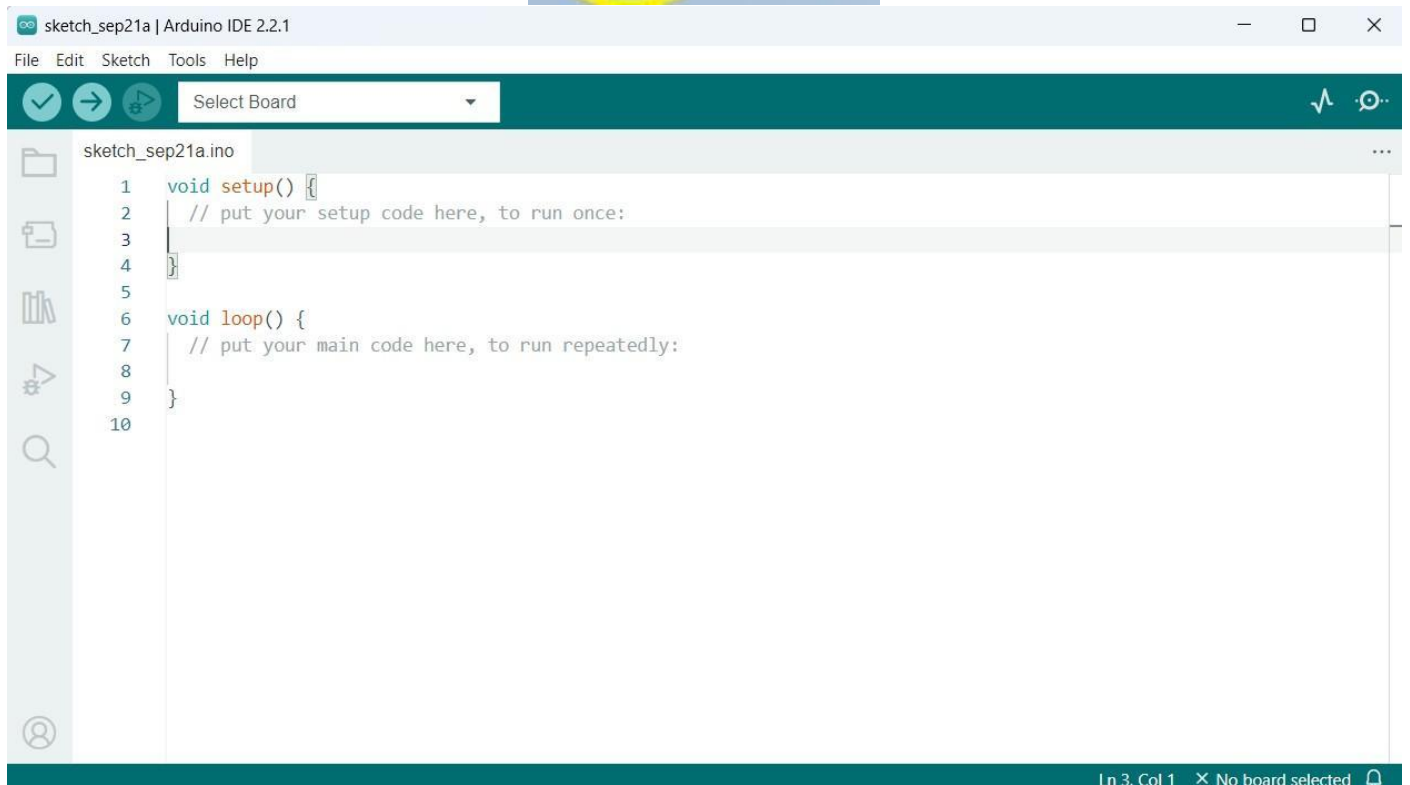
You will get a folder named Arduino-ide_2.2.1_windows_64bit

 locales	8/7/2023 6:29 AM	File folder	
 resources	8/7/2023 6:29 AM	File folder	
 swiftshader	8/7/2023 6:43 AM	File folder	
 Arduino IDE	8/7/2023 6:44 AM	Application	137,282 KB
 chrome_100_percent.pak	8/7/2023 6:29 AM	PAK File	139 KB
 chrome_200_percent.pak	8/7/2023 6:29 AM	PAK File	203 KB
 d3dcompiler_47.dll	8/7/2023 6:29 AM	Application extens...	4,419 KB
 ffmpeg.dll	8/7/2023 6:29 AM	Application extens...	2,640 KB
 icudtl.dat	8/7/2023 6:29 AM	DAT File	9,977 KB
 libEGL.dll	8/7/2023 6:29 AM	Application extens...	432 KB
 libGLESv2.dll	8/7/2023 6:29 AM	Application extens...	7,648 KB
 LICENSE.electron	8/7/2023 6:29 AM	Text Document	2 KB
 LICENSES.chromium	8/7/2023 6:29 AM	Firefox HTML Doc...	5,306 KB
 resources.pak	8/7/2023 6:43 AM	PAK File	5,733 KB
 snapshot_blob.bin	8/7/2023 6:43 AM	BIN File	49 KB
 v8_context_snapshot.bin	8/7/2023 6:43 AM	BIN File	161 KB
 vk_swiftshader.dll	8/7/2023 6:43 AM	Application extens...	4,363 KB
 vk_swiftshader_icd.json	8/7/2023 6:43 AM	JSON File	1 KB
 vulkan-1.dll	8/7/2023 6:43 AM	Application extens...	716 KB

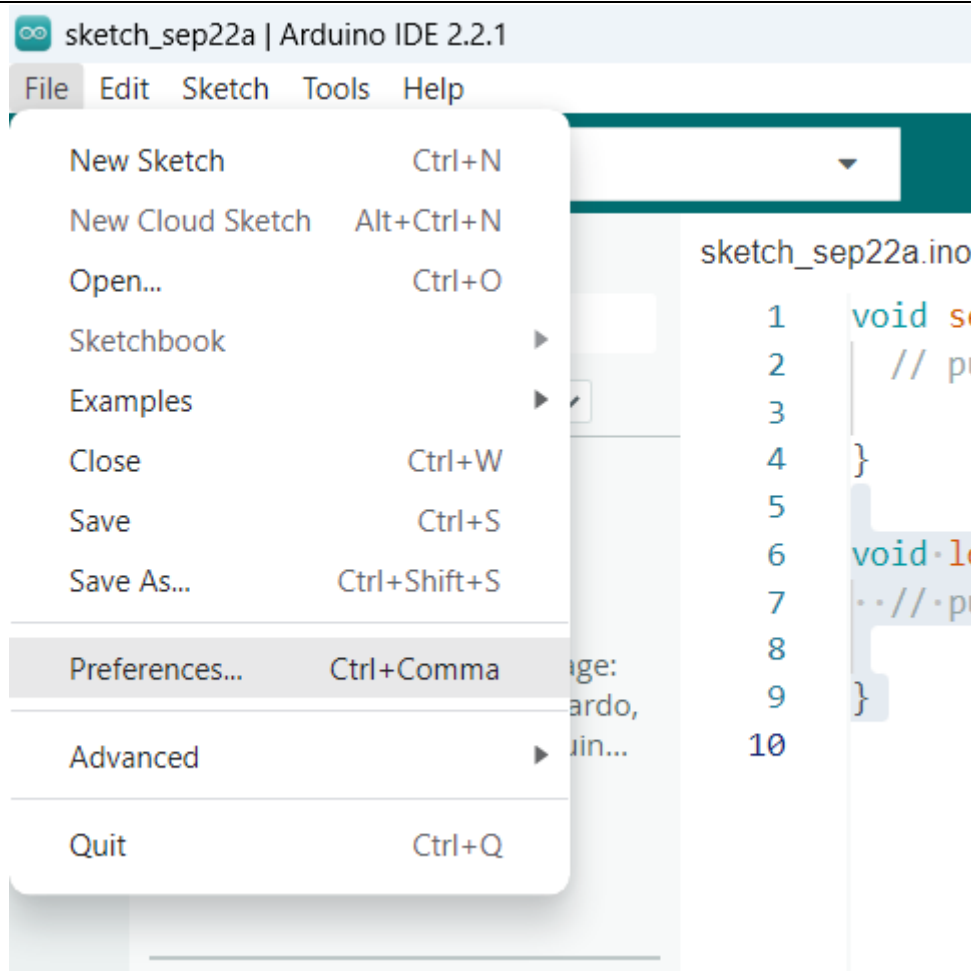
In this folder , right click Arduino IDE (Application file) to create a shortcut and place the Short Cut in the Desktop.


STM32Trust...


Arduino IDE -
Shortcut



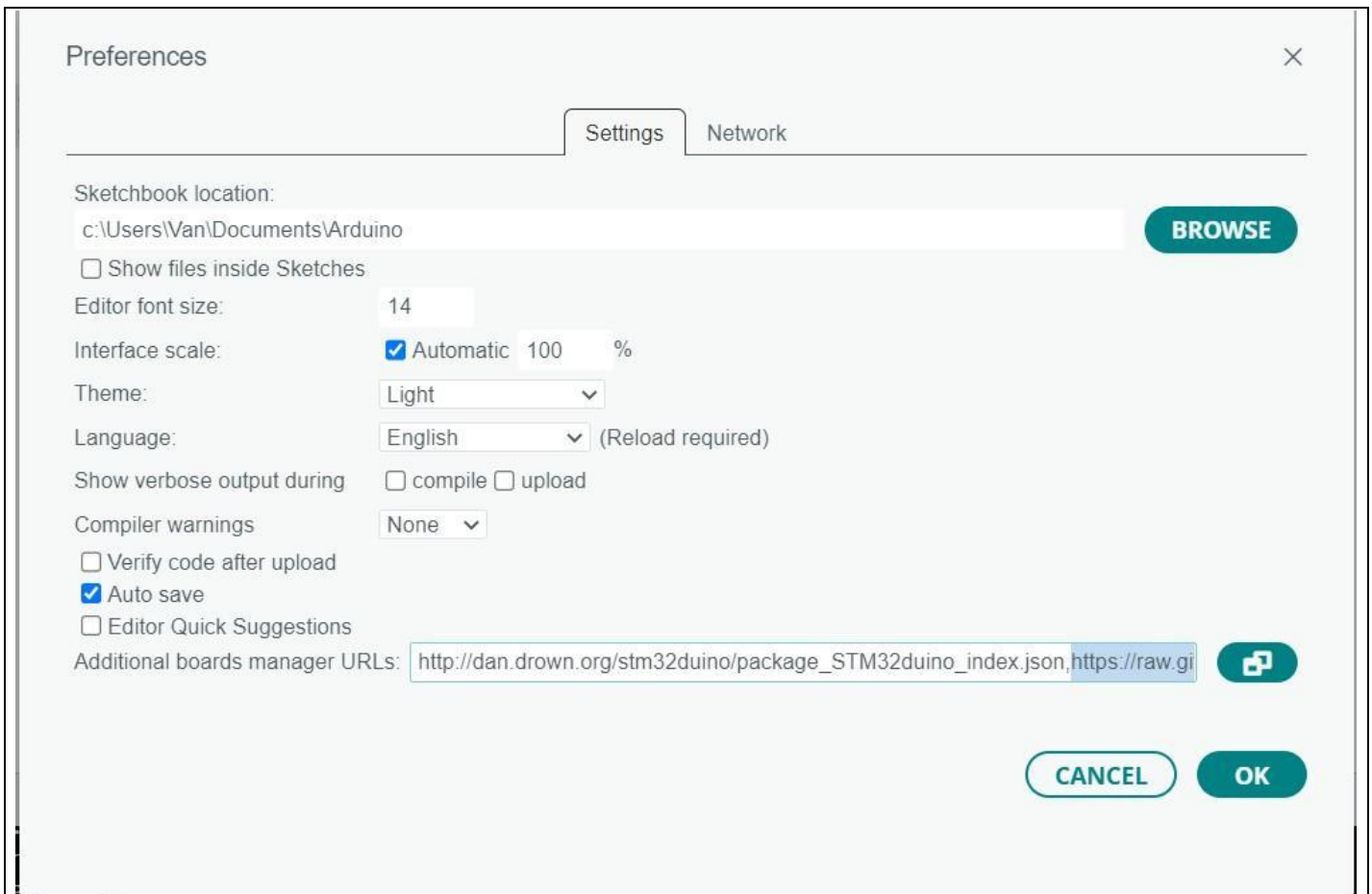
Arduino Installation was successfully completed.
Next we need to install ESP32 Board under the Boards Manager.
Click preferences under file menu.



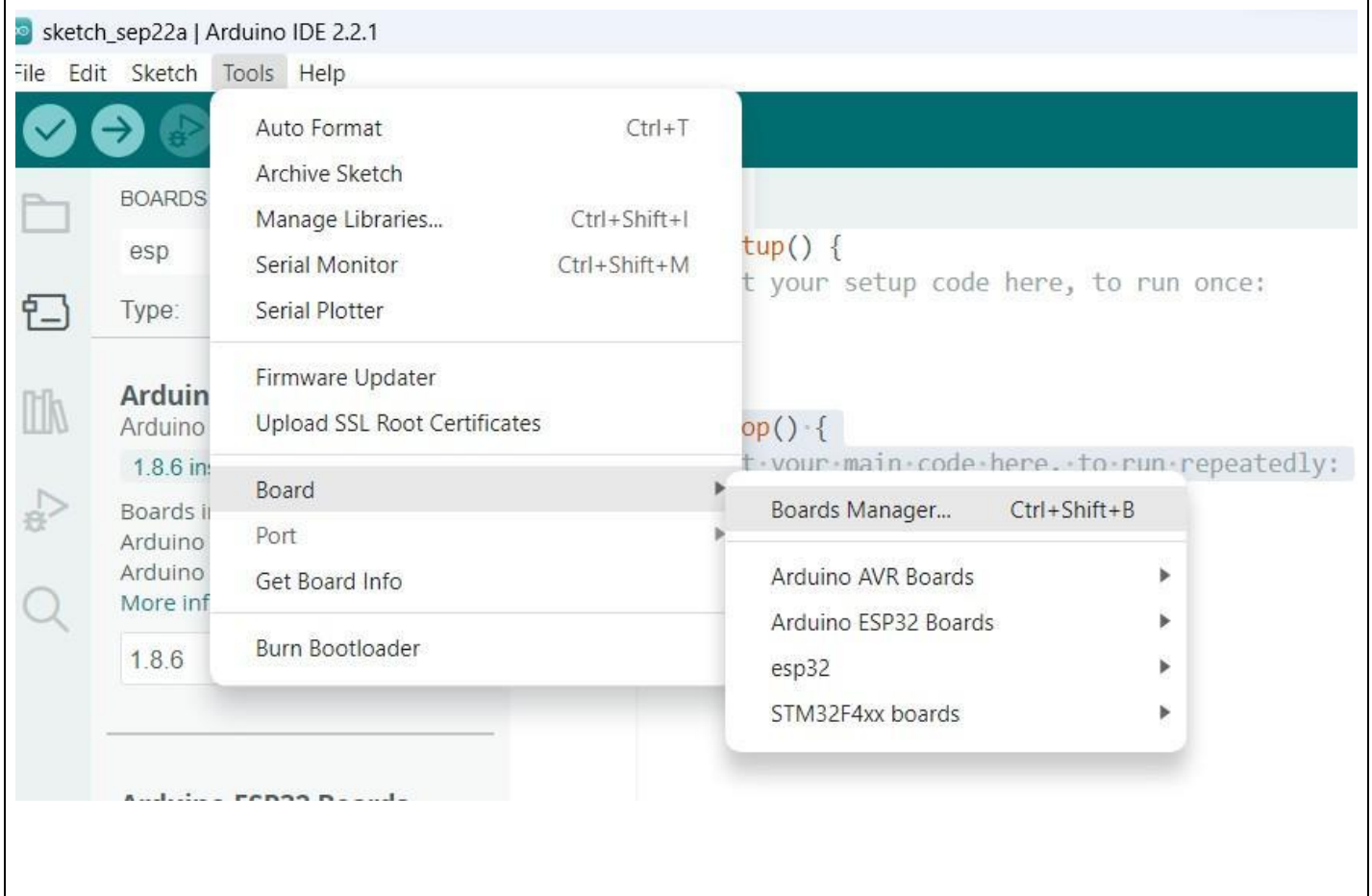
In the Additional boards manager, paste
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

If some URLs are already there, say
http://dan.drown.org/stm32duino/package_STM32duino_index.json

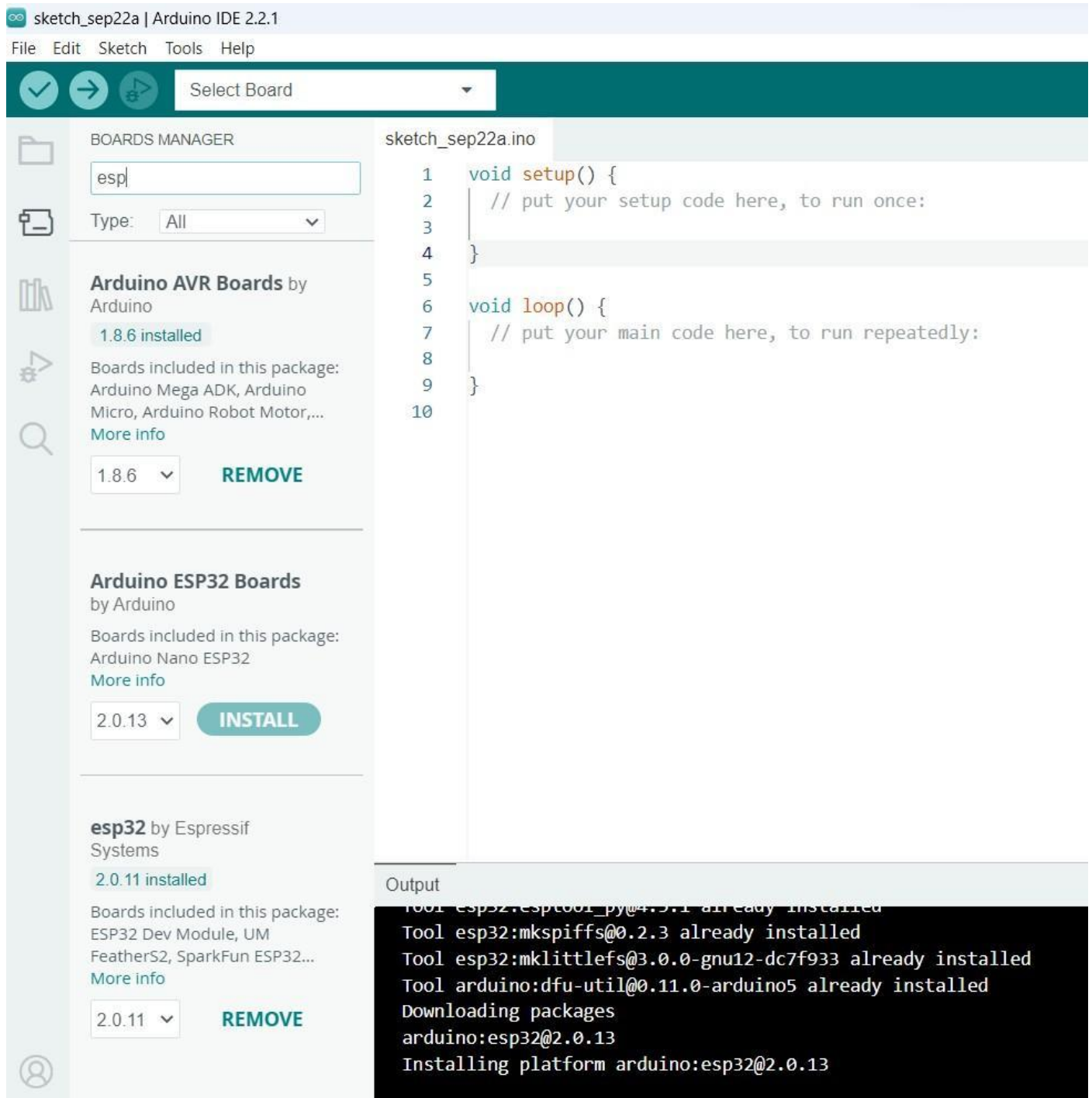
Just put a comma and then paste the URL for ESP32 Installation files.
http://dan.drown.org/stm32duino/package_STM32duino_index.json,https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json



Click OK to proceed Further.
Under Tools menu, Select Board and then Boards Manager.



In the following screen, type esp in the text box provided for Boards Manager.



The screenshot shows the Arduino IDE 2.2.1 interface. The top bar indicates the current sketch is 'sketch_sep22a'. The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu bar is a toolbar with icons for checking, running, and uploading, followed by a 'Select Board' dropdown menu. The left sidebar contains the 'BOARDS MANAGER' panel. In this panel, the search box contains the text 'esp'. Below the search box, the 'Type' is set to 'All'. The panel lists two categories of boards: 'Arduino AVR Boards by Arduino' (version 1.8.6 installed) and 'Arduino ESP32 Boards by Arduino' (version 2.0.13, with an 'INSTALL' button). The 'esp32 by Espressif Systems' package (version 2.0.11 installed) is also listed. The right pane shows the 'sketch_sep22a.ino' file with the following code:

```
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
10
```

The bottom pane shows the 'Output' window with the following text:

```
Tool esp32:esptool.py@4.31 already installed
Tool esp32:mkspiffs@0.2.3 already installed
Tool esp32:mklittlefs@3.0.0-gnu12-dc7f933 already installed
Tool arduino:dfu-util@0.11.0-arduino5 already installed
Downloading packages
arduino:esp32@2.0.13
Installing platform arduino:esp32@2.0.13
```

You will have a list of related board packages.

You have to install both esp32 by Espressif and Arduino ESP32 boards by Arduino

After clicking install button, it will take few minutes to install and after successful installation, Install Button will be changed to Remove Button.

Select Board

BOARDS MANAGER

esp

Type: All

Arduino AVR Boards by Arduino

1.8.6 installed

Boards included in this package: Arduino Nano, Arduino Leonardo, Arduino Fio, Arduino BT, Arduin...

More info

1.8.6 REMOVE

Arduino ESP32 Boards by Arduino

2.0.13 installed

Boards included in this package: Arduino Nano ESP32

More info

2.0.13 REMOVE

esp32 by Espressif Systems

2.0.11 installed

Boards included in this package: WiPy 3.0, Department of Alchemy MiniMain ESP32-S2, ESP32S2...

More info

2.0.11 REMOVE

sketch_sep22a.ino

```

1  void setup() {
2      // put your setup code
3
4  }
5
6  void loop() {
7      // put your main code h
8
9  }
10

```

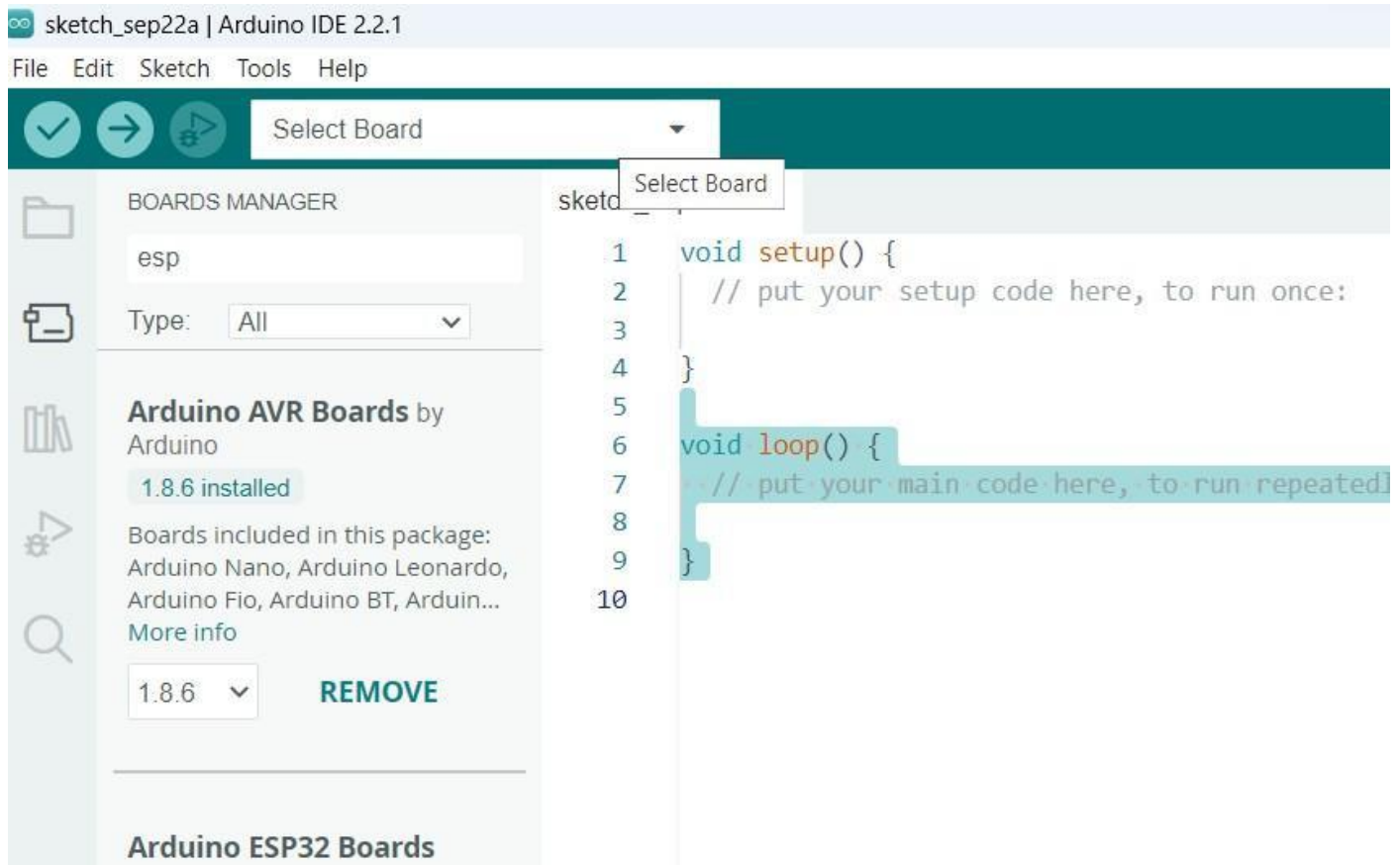
Output

```

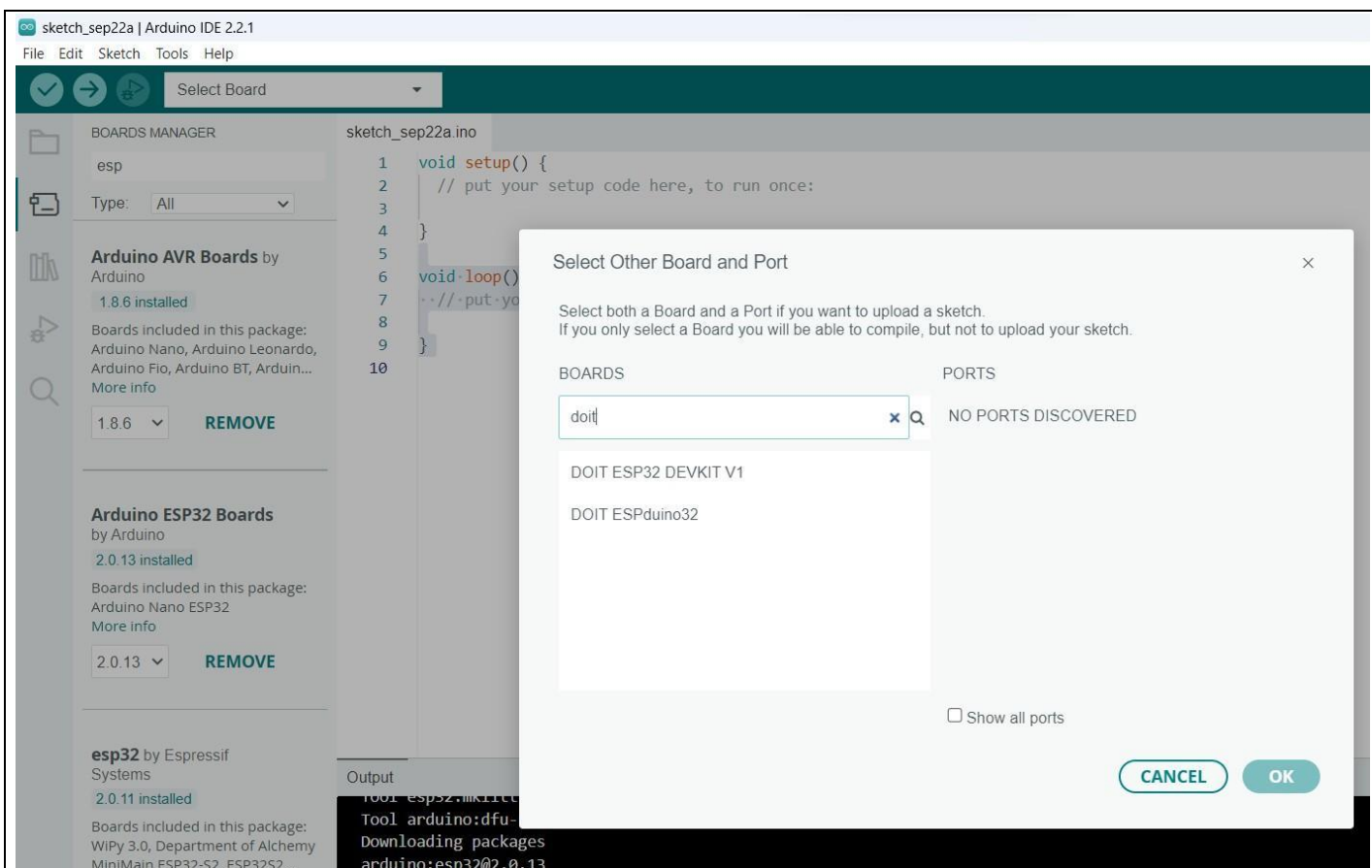
Tool esp32:mkidffiers@3.0.0-gnu
Tool arduino:dfu-util@0.11.0-arc
Downloading packages
arduino:esp32@2.0.13
Installing platform arduino:esp
Configuring platform.
Platform arduino:esp32@2.0.13 i

```


After installation, click select board



If you are able to see the below screen, your software installation was successfully completed.




Installation of CH340 USB-UART Bridge driver

CH340 is a TTL (serial) to USB converter and vice versa. This chip has been used in some boards such as Arduino boards (non-original Arduinos), ESP8266, etc. The boards using the CH340 chip, don't need a programmer in order to access the processor or to program them. But there is a downside. An extra driver must be installed before starting to work with boards having this IC. In this section, you will learn how to install the CH340 driver.

Installing CH340 Driver on Windows

If you connect your board to the computer before installing the driver, your computer will not recognize the board correctly and you will see following image in Device Manager.

To open Device Manager, search for it in the Windows Start menu.

Device Manager
Control panel

Apps

Settings

Settings

Printers & scanners

Device security

Device performance & health

View your PC name

Bluetooth and other devices settings

Search the web

devices & printers - See more search results

device manager

device

device and printer

Device Manager
Control panel

Open

device Manager

Device Manager

File Action View Help

DESKTOP-SJDL5PT

Audio inputs and outputs

Computer

Disk drives

Display adapters

DVD/CD-ROM drives

Human Interface Devices

IDE ATA/ATAPI controllers

Imaging devices

Keyboards

Mice and other pointing devices

Monitors

Network adapters

Other devices

USB Serial

Ports (COM & LPT)

Communications Port (COM1)

ECP Printer Port (LPT1)

Intel(R) Active Management Technology - SOL (COM3)

Print queues

Printers

Processors

Security devices

Software devices

Sound, video and game controllers

Storage controllers

Follow the steps below to install the CH340 driver:

Step 1: CH340 driver download

To install the CH340 driver for Windows, go to the website How to Install CH340 Drivers - SparkFun Learn (<https://learn.sparkfun.com/tutorials/how-to-install-ch340-drivers/all#drivers-if-you-need-them>) and download the EXE file for the driver. After the download is complete, execute the file, then select "Install" in the Driver Setup window. Once the installation is successful, you will see a dialog box confirming that the driver install was successful.

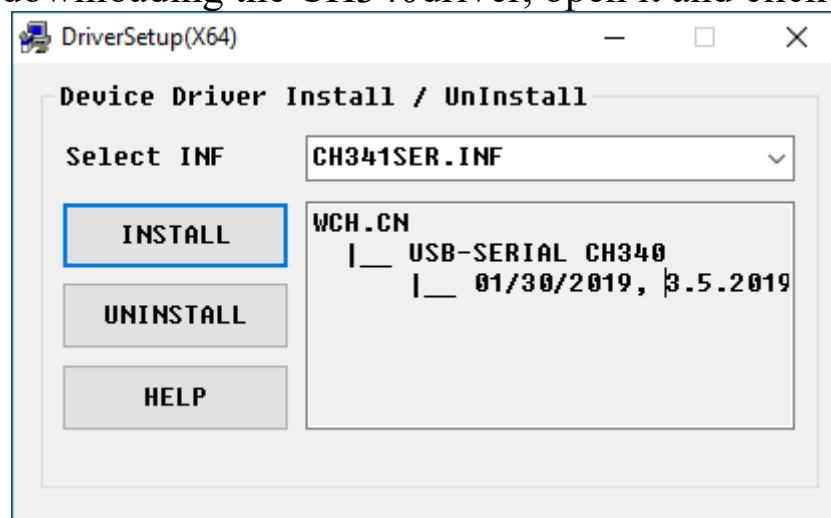
Alternate link for CH340 driver:

http://www.wch.cn/download/CH341SER_ZIP.html

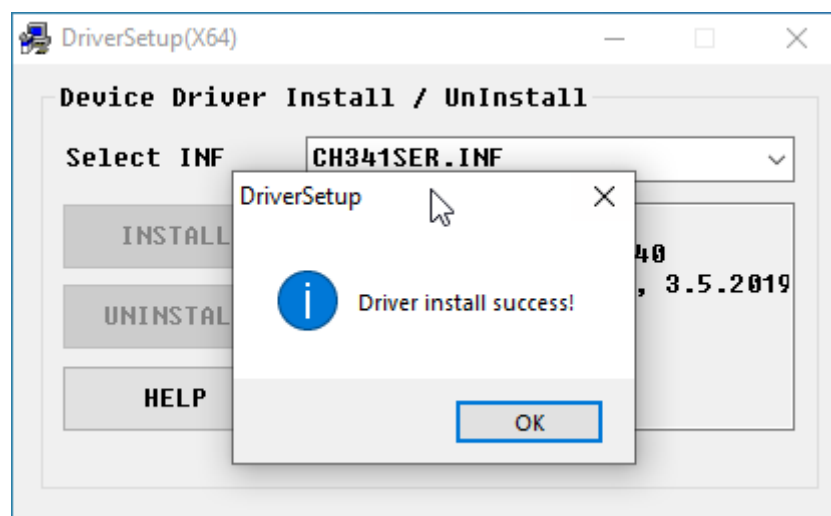
<https://electropeak.com/learn/download/ch341ser-zip/>

Step 2: Installing the driver

After downloading the CH340driver, open it and click Install.




























After successful installation you should see this message.



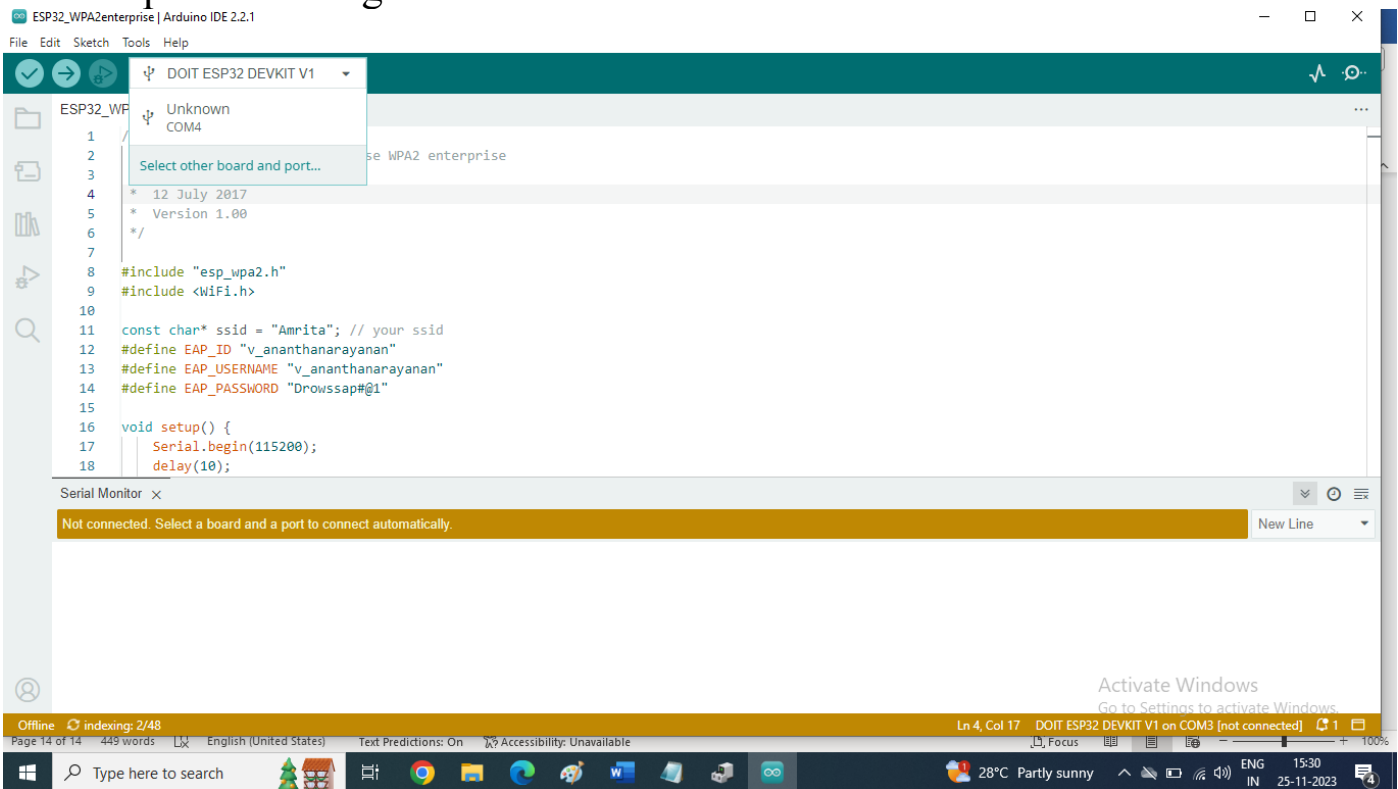
Device Manager

File Action View Help



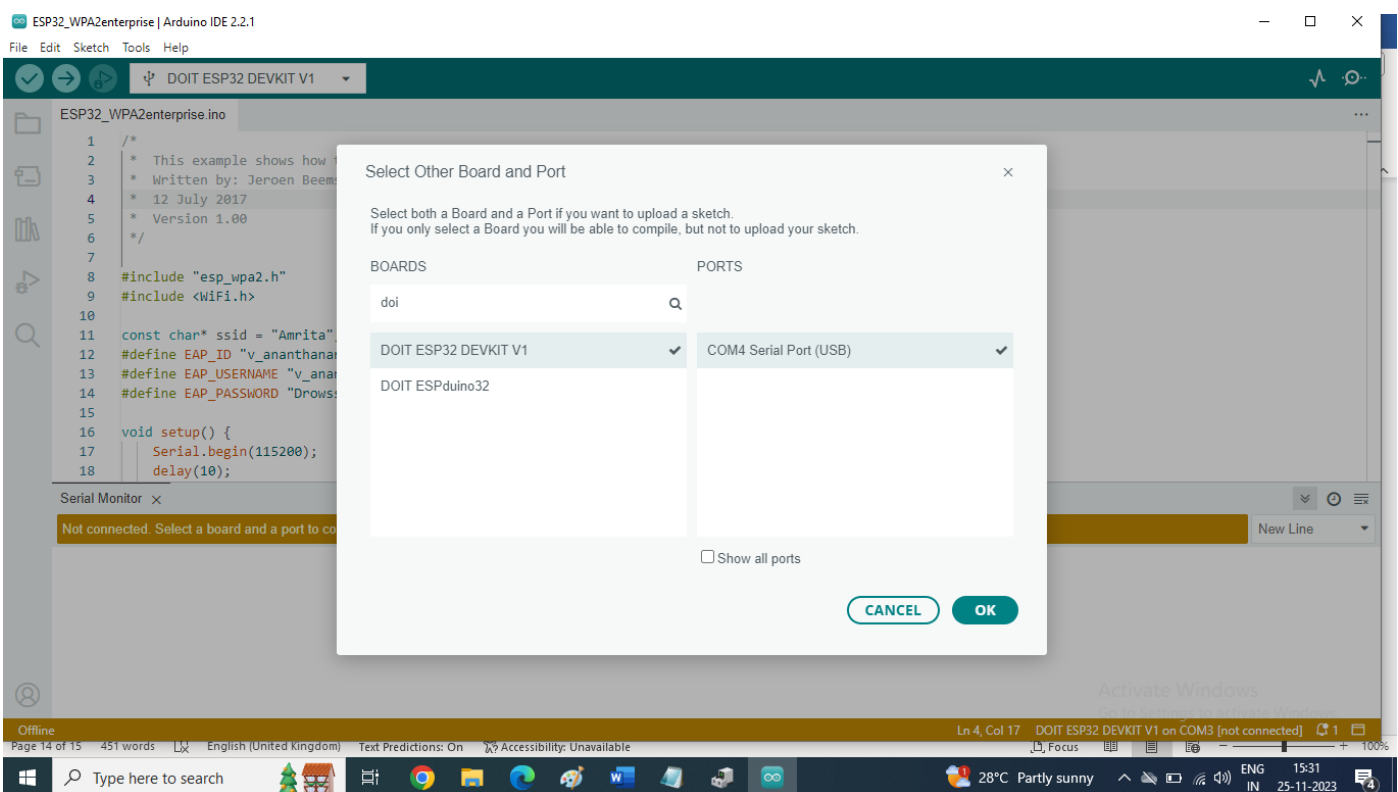
- ▼  DESKTOP-P4311J5
 - >  Audio inputs and outputs
 - >  Batteries
 - >  Bluetooth
 - >  Cameras
 - >  Computer
 - >  Disk drives
 - >  Display adapters
 - >  DVD/CD-ROM drives
 - >  Human Interface Devices
 - >  Keyboards
 - >  Mice and other pointing devices
 - >  Monitors
 - >  Network adapters
 - >  Other devices
 - ▼  Ports (COM & LPT)
 -  USB-SERIAL CH340 (COM4)
 - >  Print queues
 - >  Processors
 - >  SD host adapters
 - >  Software devices
 - >  Sound, video and game controllers
 - >  Storage controllers
 - >  System devices
 - >  Universal Serial Bus controllers

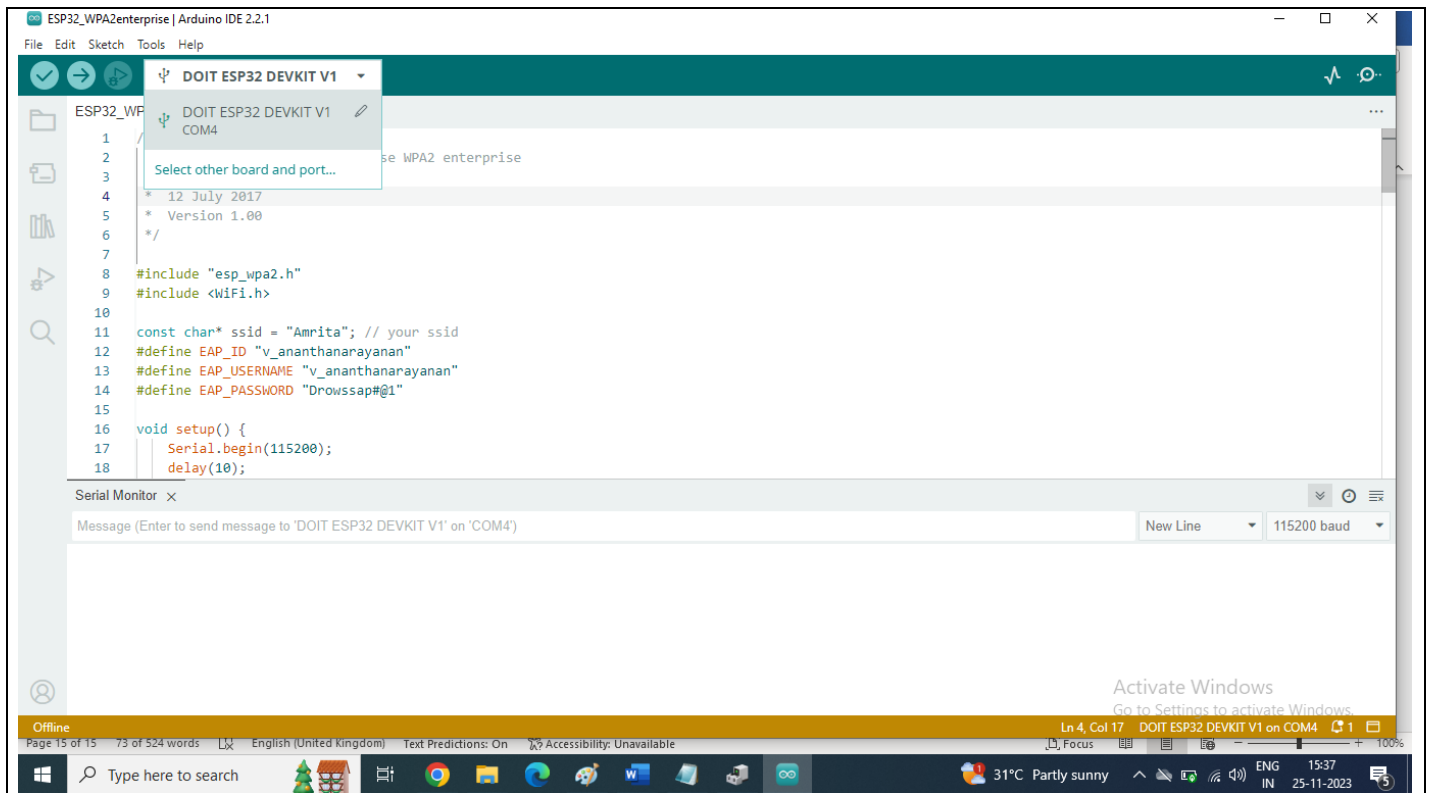
Step 4: Checking Correct Board and Driver Installation in Arduino IDE



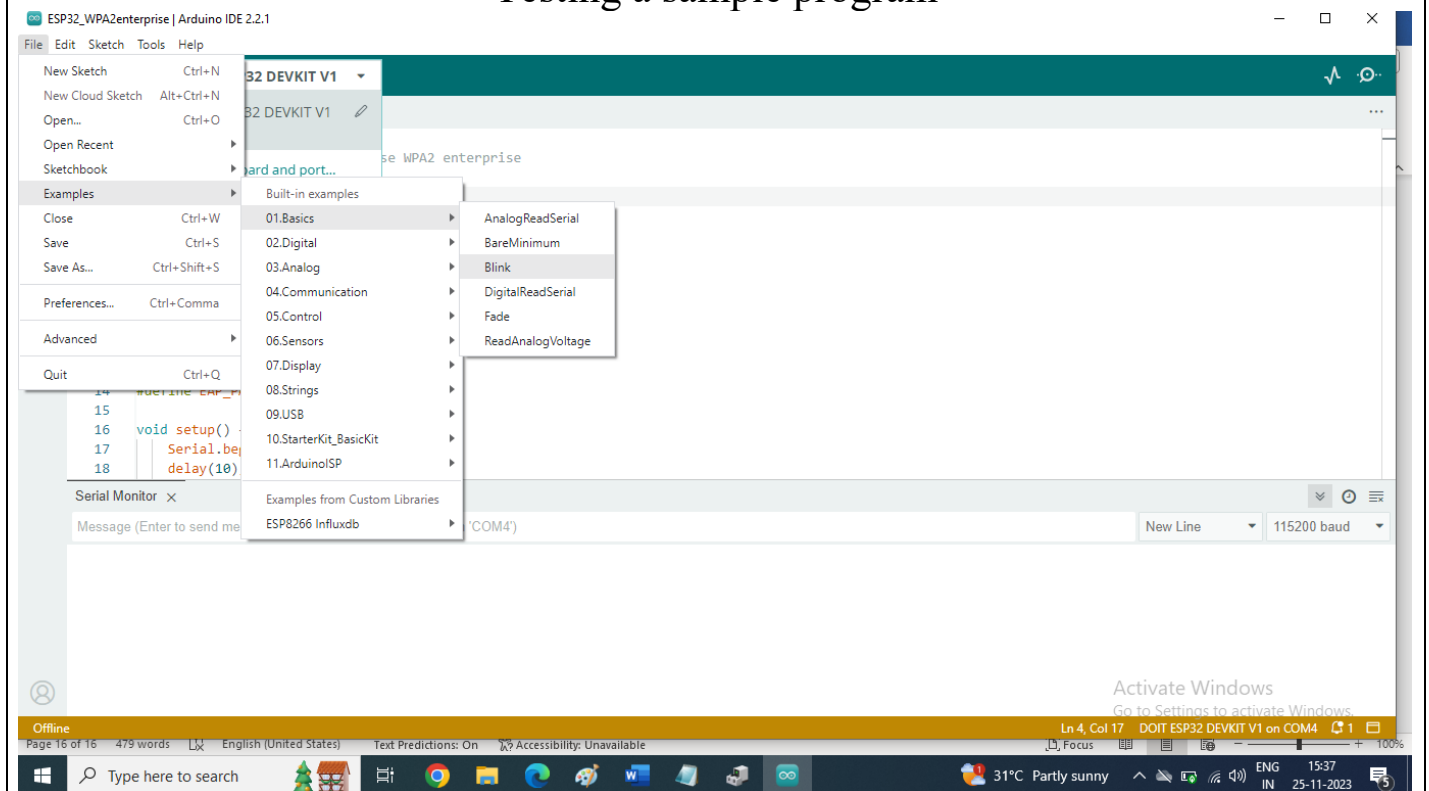
Type DOIT I the search text box for boards and select DOIT ESP32 DEVKIT V1 as well as select COM4 Serial Port (USB)

Note: It should match with the port number for USB-Serial CH340 listed in device Manager. Depends on the system, this COM Port may have various number from 3 on wards. Check your device manager for the COM port through which your IoT Kit is connected to your system / laptop.





Testing a sample program



```

void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

```

In our Tektork IoT Kit, User LED 1 is connected to GPIO Pin 33 and so, LED_BUILTIN is changed to 33.

```
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(33, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(33, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);            // wait for a second  
  digitalWrite(33, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);            // wait for a second  
}
```



Click Upload Button

Blink | Arduino IDE 2.2.1
File Edit Sketch Tools Help

DOIT ESP32 DEVKIT V1

Blink.ino

```
1  /*
2   Blink
3
4   Turns an LED on for one second, then off for one second, repeatedly.
5   ASC-IoT Kit had user LED1 at GPIO 33 and LED2 at GPIO 4
6   */
7
8   // the setup function runs once when you press reset or power the board
9   void setup() {
10    // initialize digital pin LED_BUILTIN as an output.
11    pinMode(33, OUTPUT);
12  }
13
14  // the loop function runs over and over again forever
15  void loop() {
16    digitalWrite(33, HIGH); // turn the LED on (HIGH is the voltage level)
17    delay(1000);           // wait for a second
18    digitalWrite(33, LOW);  // turn the LED off by making the voltage LOW
19    delay(1000);           // wait for a second
20  }
```

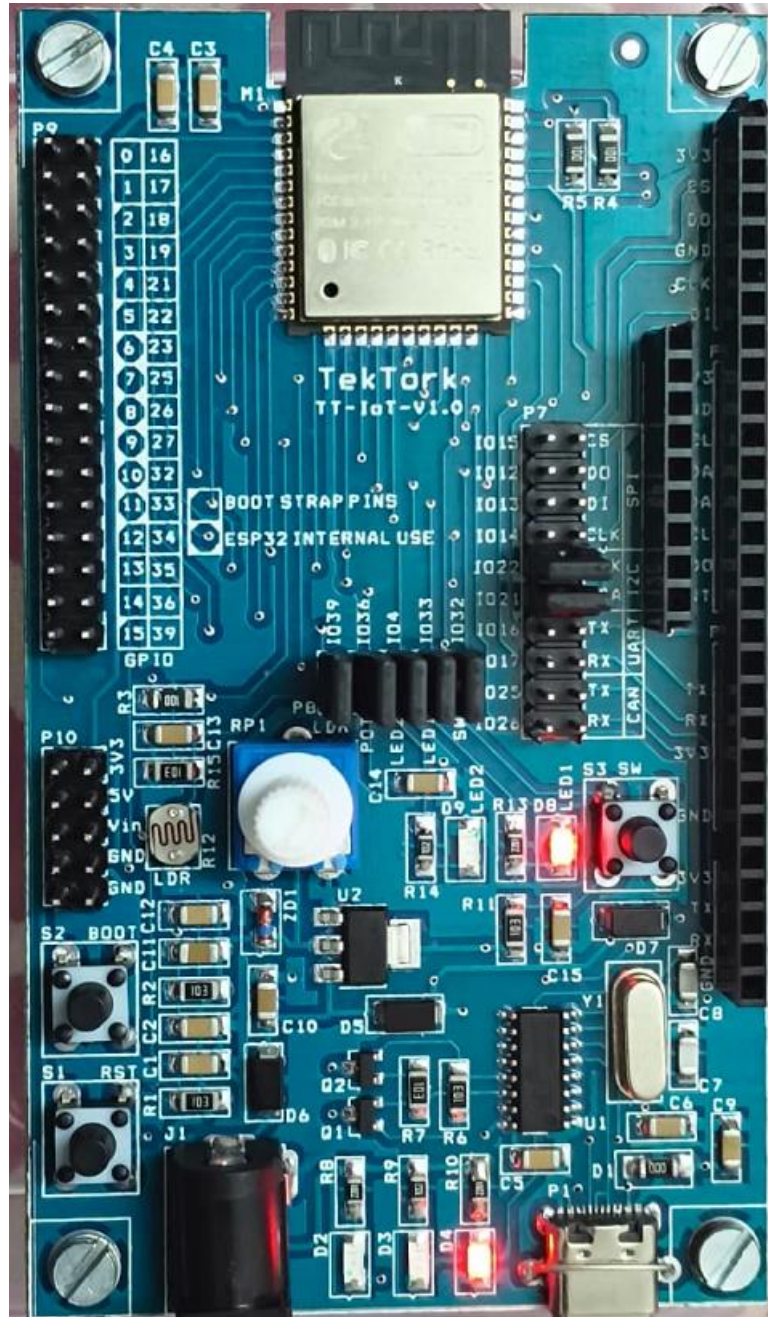
Output

```
Sketch uses 237117 bytes (18%) of program storage space. Maximum is 1310720 bytes.
Global variables use 21048 bytes (6%) of dynamic memory, leaving 306632 bytes for local variables. Maximum is 327680 bytes.
esptool.py v4.5.1
Serial port COM4
Connecting.....
Chip is ESP32-D0WD-V3 (revision v3.0)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: c4:de:e2:10:70:ac
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 921600
Changed.
Configuring flash size...
Flash will be erased from 0x00001000 to 0x00005fff...
Flash will be erased from 0x00008000 to 0x00008fff...
Flash will be erased from 0x0000e000 to 0x0000ffff...
Flash will be erased from 0x00010000 to 0x00049fff...
Compressed 17568 bytes to 12204...
Writing at 0x00001000... (100 %)
Wrote 17568 bytes (12204 compressed) at 0x00001000 in 0.5 seconds (effective 264.9
kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 146...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (146 compressed) at 0x00008000 in 0.1 seconds (effective 301.9 kbit/s)...
Hash of data verified.
Compressed 8192 bytes to 47...
Writing at 0x0000e000... (100 %)
Wrote 8192 bytes (47 compressed) at 0x0000e000 in 0.2 seconds (effective 428.1 kbit/s)...
Hash of data verified.
Compressed 237488 bytes to 130712...
Writing at 0x00010000... (12 %)
Writing at 0x0001d273... (25 %)
Writing at 0x00024369... (37 %)
Writing at 0x0002953b... (50 %)
Writing at 0x0002eb93... (62 %)
Writing at 0x00037102... (75 %)
```

Writing at 0x0003f146... (87 %)
Writing at 0x000447c5... (100 %)
Wrote 237488 bytes (130712 compressed) at 0x00010000 in 2.6 seconds (effective 720.4 kbit/s)..
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

After RESET, User LED User LED 1 at GPIO 33 blinks at the rate of 1 Second.



1. Basic LED Blink Program on ESP32

Full Code - Single LED Blink

```
const int ledPin = 33; // GPIO33 where LED is connected

void setup()
{
  pinMode(ledPin, OUTPUT); // Initialize GPIO33 as output
}

void loop()
{
  digitalWrite(ledPin, HIGH); // Turn LED ON
  delay(1000);                // Wait for 1 second
  digitalWrite(ledPin, LOW);  // Turn LED OFF
  delay(1000);                // Wait for 1 second
}
```

Step-by-Step Program Explanation

1.1. Pin Declaration

```
const int ledPin = 33; // GPIO33 where LED is connected
```

- Purpose: Defines which GPIO pin the LED is connected to
- Technical Detail: **const int** creates a constant integer variable that cannot be changed during program execution
- Hardware Connection: The LED is physically connected to GPIO pin 33 on the Tektork IoT board

1.2. Setup Function

```
void setup()
{
  pinMode(ledPin, OUTPUT); // Initialize GPIO33 as output
}
```

- Purpose: Configures the GPIO pin for output operation
- Technical Detail: `pinMode()` function sets the pin mode - INPUT or OUTPUT
- Execution: This function runs only once when the ESP32 powers up or resets
- Hardware Effect: Prepares GPIO33 to send voltage signals to control the LED

1.3. Main Loop Function

```
void loop()
{
  digitalWrite(ledPin, HIGH); // Turn LED ON
  delay(1000);                // Wait for 1 second
  digitalWrite(ledPin, LOW);  // Turn LED OFF
  delay(1000);                // Wait for 1 second
}
```

Line-by-Line Breakdown:

digitalWrite(ledPin, HIGH);

- Sets GPIO33 to HIGH voltage (3.3V)
- This turns the LED ON by providing 3.3V to the Anode terminal of LED.

delay(1000);

- Pauses program execution for 1000 milliseconds (1 second)
- LED remains ON during this time

digitalWrite(ledPin, LOW);

- Sets GPIO33 to LOW voltage (0V)
- This turns the LED OFF by making voltage at Anode terminal to 0V

delay(1000);

- Another 1-second pause with LED OFF
- LED blinks at the rate of 1 Second.

1.4. Program Flow

Line-by-Line Breakdown:

- The loop() function repeats continuously after setup() completes
- This creates an endless cycle: LED ON → Wait → LED OFF → Wait → Repeat
- The result is a visible blinking pattern at 1-second intervals

1.5. Expected Result

- The LED will blink ON and OFF repeatedly
- Each ON and OFF state lasts exactly 1 second
- This confirms that the ESP32 is programmed correctly and functioning properly

2. Troubleshooting Tips

Even with clear instructions, you may encounter issues during installation or when uploading your LED blink program. Below are common problems and straightforward solutions to help you get back on track.

2.1. Arduino IDE Won't Launch

- Issue: Double-clicking the IDE icon does nothing or shows an error message.
- Solution:
Ensure your operating system meets the minimum requirements (Windows 7/8/10, macOS 10.10+, Linux).
Reinstall the IDE: download the latest version from the Arduino website and install again.
Run the IDE as Administrator (Windows) or with elevated privileges (macOS/Linux).

ESP32 Board Not Listed in Boards Manager

- Issue: You don't see "ESP32 Dev Module" under Tools → Board.
- Solution:
In File → Preferences, confirm the Additional Boards Manager URL is exactly https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json.
After adding the URL, open Tools → Board → Boards Manager, search for "ESP32," and click Install.

2.2. COM Port Doesn't Appear or Is Grayed Out

- Issue: No COM port available under Tools → Port.
- Solution:
Verify the CH340 driver is installed correctly (check for "USB-SERIAL CH340" in Device Manager on Windows).
Try a different USB cable or USB port on your computer (some cables are power-only).
Restart your computer after driver installation.

2.3. Upload Fails with "Failed to Connect to ESP32: Timed Out"

- Issue: The IDE shows an error like A fatal error occurred: Timed out waiting for packet header.
- Solution:
Press and hold the BOOT button on the ESP32 board just before the upload starts, and release it when you see the "Connecting..." message in the IDE.
Lower the upload speed: in Tools → Upload Speed, choose a slower baud rate (e.g., 115200).

2.4. LED Doesn't Blink After Successful Upload

- Issue: IDE confirms upload, but the LED stays off or on continuously.
- Solution:
Confirm you are using the correct LED pin in your code (e.g., GPIO2 for the onboard LED on many ESP32 boards. Tektork IoT kit has LED at GPIO 33).
Check that the LED is properly connected (long leg to the GPIO pin, short leg to GND via a resistor).
Verify the resistor value (220 Ω –330 Ω) is appropriate.
- In Tektork IoT Kit, LEDs are available through GPIO 33 and GPIO 4 . Ensure the jumpers are placed appropriately in the Connector P8.

Code Formatting or Syntax Errors in IDE

- Issue: Red error messages in the IDE when compiling.
- Solution:
Ensure all code lines end with a semicolon (;) where required.
Verify matching braces {} and parentheses ().
Copy and paste code into the IDE's text editor to avoid hidden characters or incorrect quotation marks.

2.5. Driver Conflicts on macOS or Linux

- Issue: CH340 driver installation fails or another driver takes precedence.
- Solution:
On macOS, allow driver in System Preferences → Security & Privacy after installation.
On Linux, check dmesg logs for USB device errors and install libftdi or cp210x / ch340 modules if needed.
Uninstall conflicting drivers before installing CH340.

By following these troubleshooting steps, you can quickly resolve the most common hurdles and continue with your Arduino IDE and ESP32 based Tektork IoT to develop any IoT and CPS project development smoothly.

3. Conclusion

By following this guide, you have successfully:

- Installed the Arduino IDE and configured it for ESP32 development.
- Added the ESP32 board package and the CH340 USB-serial driver to enable communication with your board.
- Written, compiled, and uploaded a Basic LED Blink program to GPIO pin 33 on the ESP32.
- Observed the LED blinking on and off at 1-second intervals, confirming that your hardware and software are functioning correctly.

With this foundation, you can confidently explore more advanced projects. Try modifying the blink interval, control multiple LEDs, or integrate sensors and communication modules. Experimenting with different programs will deepen your understanding of embedded systems and prepare you for complex IoT applications.
