

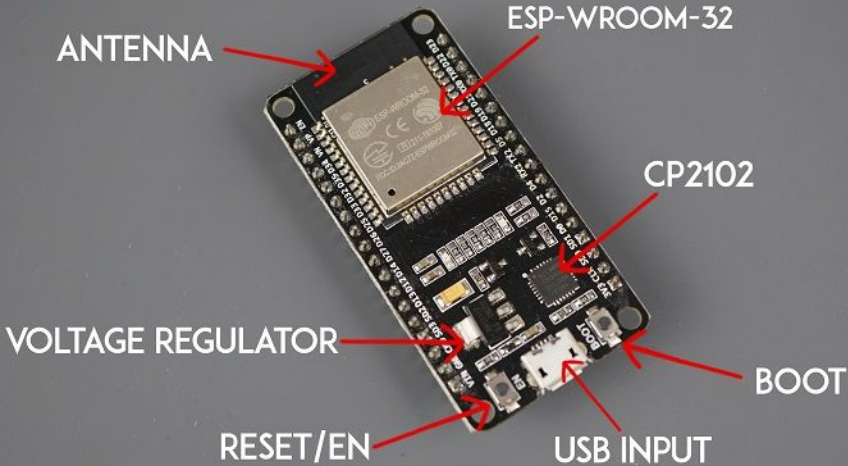
# ESP32

Microcontroller

# Characteristics

1. Low cost
2. Low power
3. Wi-Fi Capable
4. Bluetooth Capable
5. Dual Core
6. Rich IO interfaces-capacitive touch, ADC,DAC,UART,SPI,i2C,PWM
7. Arduino Compatible
8. Micropython Compatible
9. Successor to the ESP866
10. Uses 3.3V for its GPIO pins, not 5V like in the Arduino

# Hardware



Antennae-connect wirelessly

CP2102-USB to UART chip to communicate with PC via COM port

BOOT button-used to flash microcontroller

EN button-used to restart the microcontroller

Voltage Regulator-used to provide stable power to the microcontroller

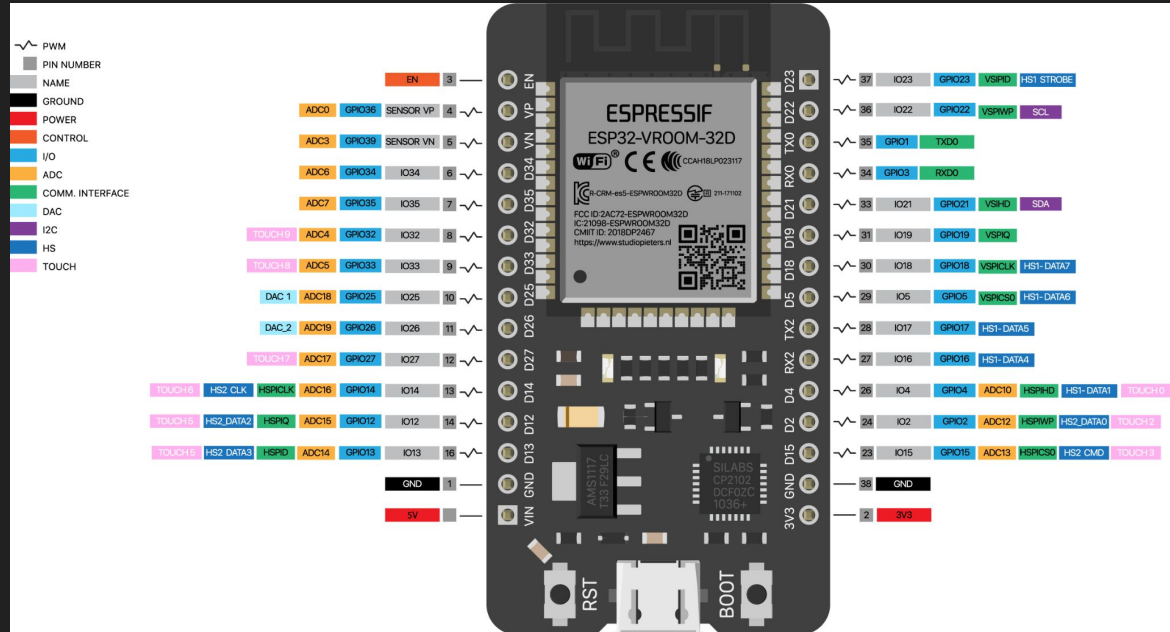
ESP-WROOM-32-chip(brains) or the microcontroller

# GPIO

General Purpose Input Output Pins

GPIO 2-internally connected LED (blue)

Red LED-power indicator



# Programming

1. Download the Arduino IDE at <https://www.arduino.cc/en/Main/Software>

School download link:

<https://drive.google.com/file/d/1VTVKKWvT9BWmKFO6p2zCBa12JeDmEdkT/view?thuser=0>



2. Use the legacy version

## Legacy IDE (1.8.X)



### Arduino IDE 1.8.19

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

#### SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

#### DOWNLOAD OPTIONS

**Windows** Win 7 and newer

**Windows** ZIP file

**Windows app** Win 8.1 or 10 [Get](#)

**Linux** 32 bits

**Linux** 64 bits

**Linux** ARM 32 bits

**Linux** ARM 64 bits

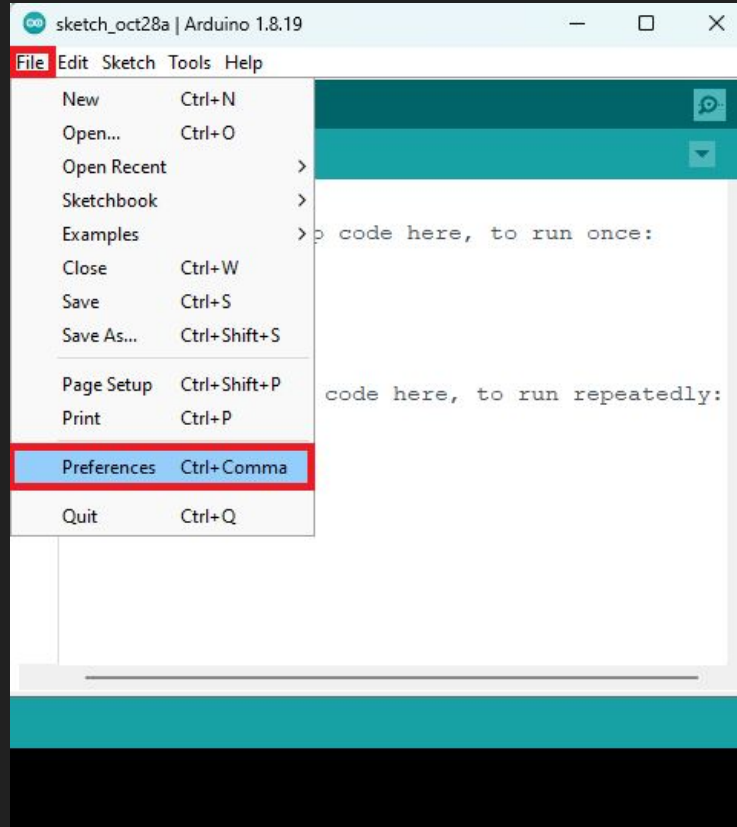
**Mac OS X** 10.10 or newer

[Release Notes](#)

[Checksums \(sha512\)](#)

# Set up the Arduino IDE to use the ESP32

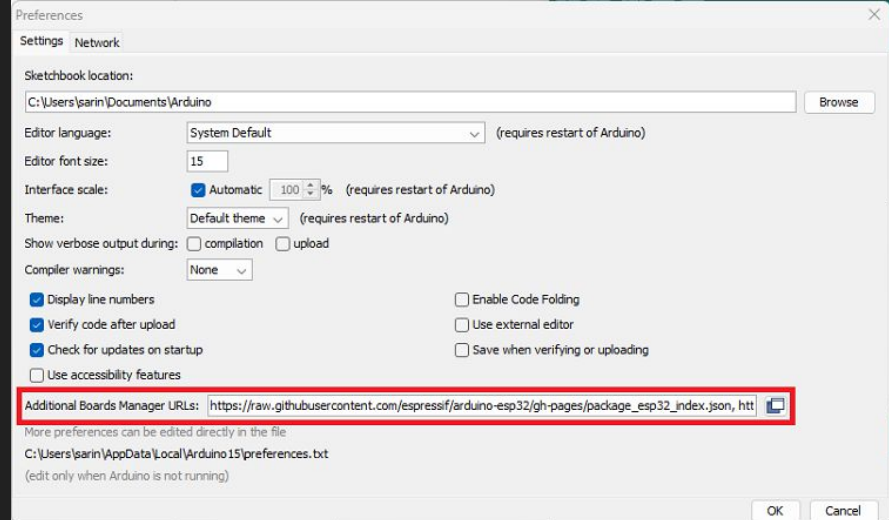
Go to **File>Preferences**



# Download and install additional boards

Enter the following into the Additional Boards Manager URLs:

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

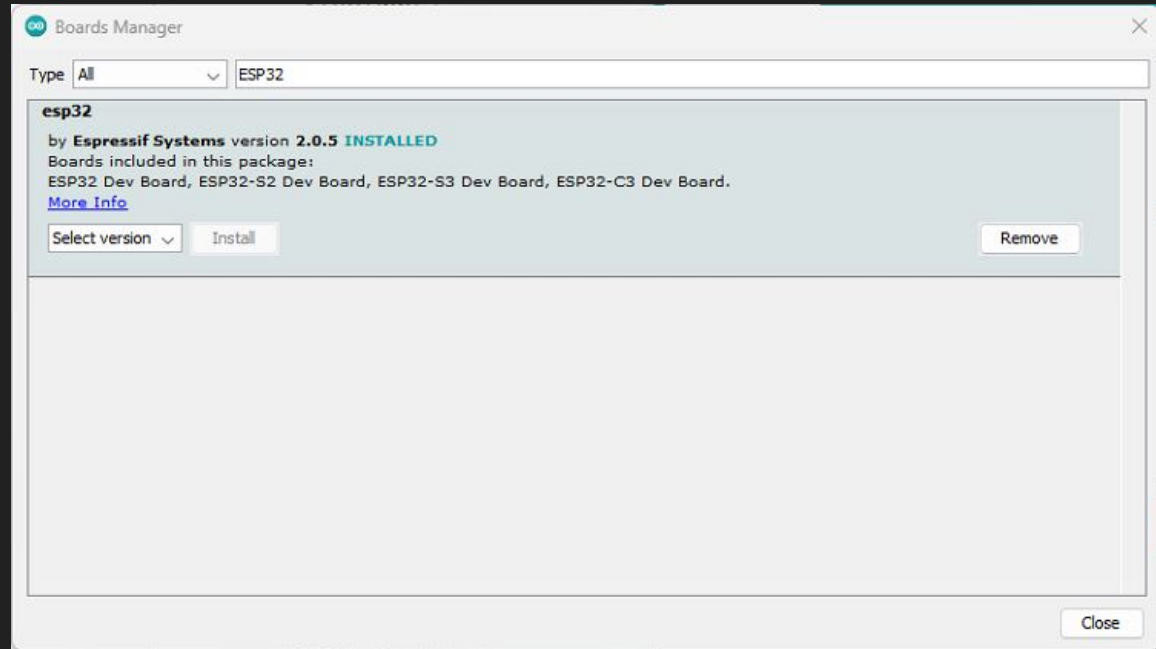


# Install the board

Tools>Board>Boards Manager

Search for esp32 and click Install

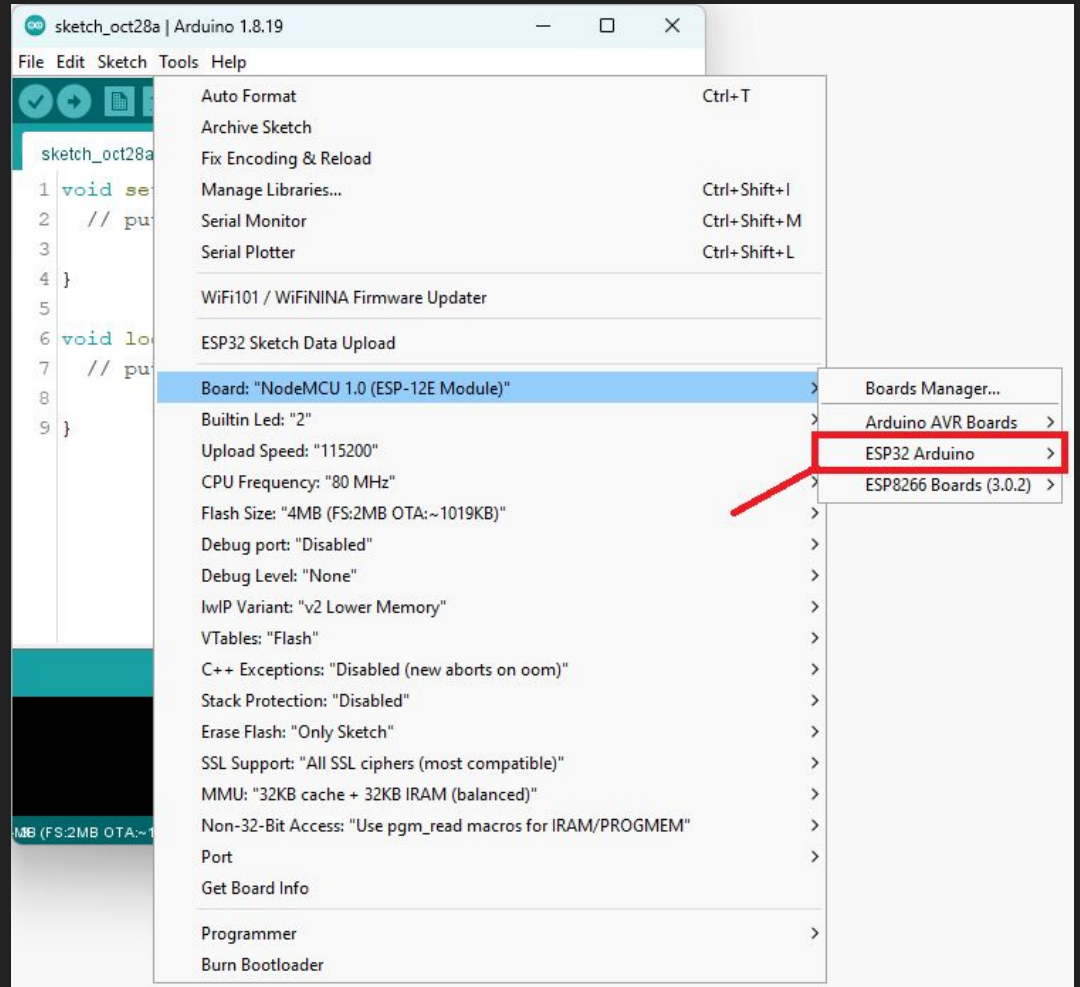
Restart Arduino





# Select the board

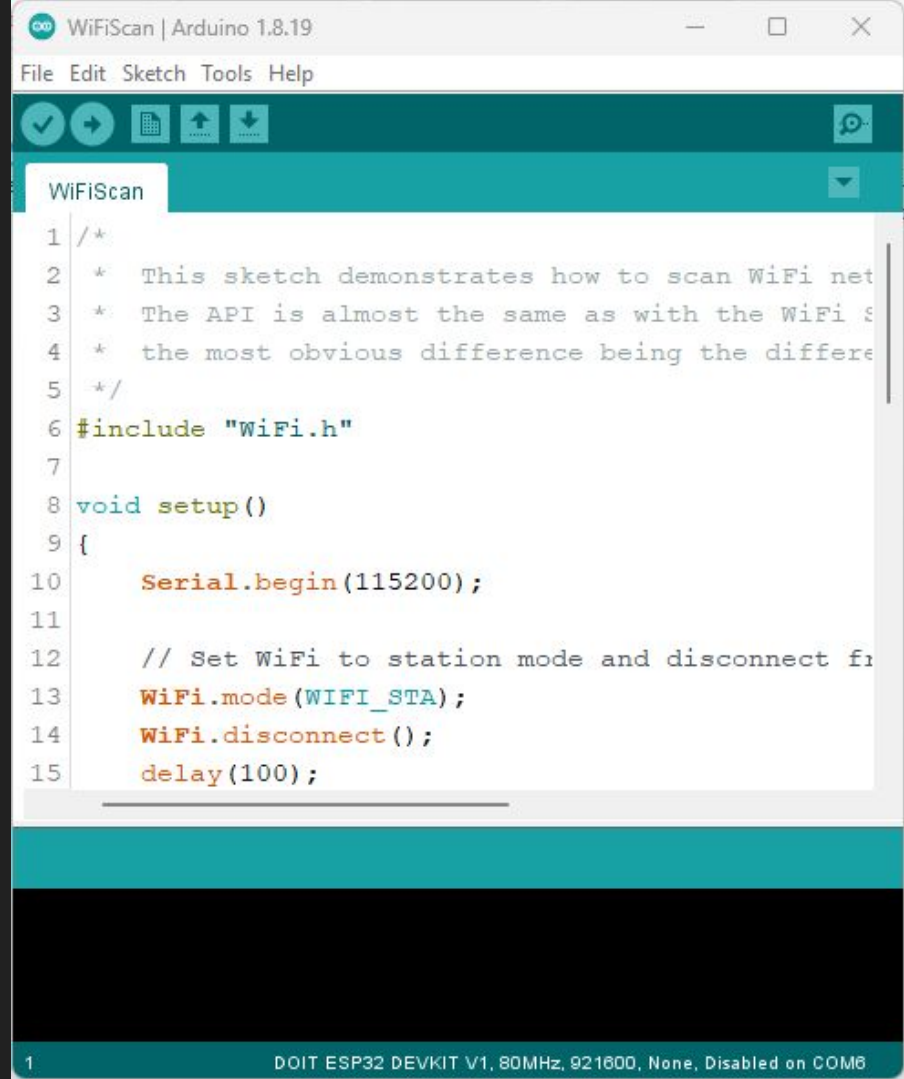
Tools>Board



# Upload Code

Open sample code:

File>Examples>WiFi>WiFiScan



The screenshot shows the Arduino IDE interface with the 'WiFiScan' sketch open. The title bar indicates 'WiFiScan | Arduino 1.8.19'. The menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. The toolbar contains icons for checking, running, and uploading code. The code editor displays the following code:

```
1 /*
2  * This sketch demonstrates how to scan WiFi networks.
3  * The API is almost the same as with the WiFi module,
4  * the most obvious difference being the difference in the
5  * pin numbers.
6  *
7  */
8 #include "WiFi.h"
9
10 void setup()
11 {
12     Serial.begin(115200);
13
14     // Set WiFi to station mode and disconnect from AP (if we have one)
15     WiFi.mode(WIFI_STA);
16     WiFi.disconnect();
17     delay(100);
```

The status bar at the bottom shows '1' on the left and 'DOIT ESP32 DEVKIT V1, 80MHz, 921600, None, Disabled on COM6' on the right.

# Next Steps

Tools>Board>select the proper board (ESP32 Dev Module)

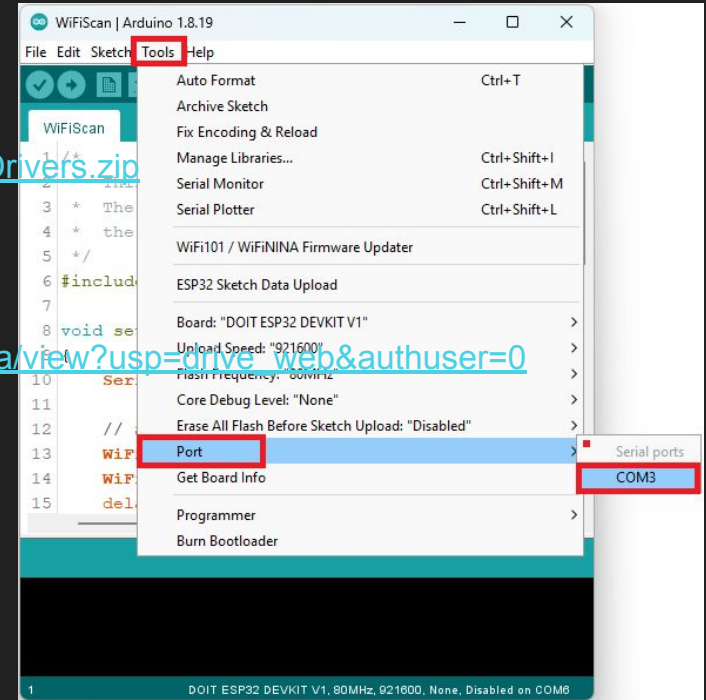
Tools>Port>select the available COM port (if this is greyed out you need to install the required USB drivers)-available at [CP210x USB to UART Bridge VCP Drivers - Silicon Labs \(silabs.com\)](https://www.silabs.com/documents/public/software/CP210x_Windows_Drivers.zip)

Specific Download Link:

[https://www.silabs.com/documents/public/software/CP210x\\_Windows\\_Drivers.zip](https://www.silabs.com/documents/public/software/CP210x_Windows_Drivers.zip)

School download link:

[https://drive.google.com/file/d/1px4EZwMkJB0xd6\\_dDcLf22oA7p5Vq\\_ea/view?usp=drive\\_web&authuser=0](https://drive.google.com/file/d/1px4EZwMkJB0xd6_dDcLf22oA7p5Vq_ea/view?usp=drive_web&authuser=0)



# Upload the Code

## Press the Upload button

If it fails then try again but hold the BOOT button (only need to do this while it connects and hold it until it says its done uploading)

Press the EN button to start the program (if it doesn't start automatically)

Open up the Serial Monitor: Tools>Serial Monitor and press the EN button

If all goes well you should see a list of nearby wifi networks:

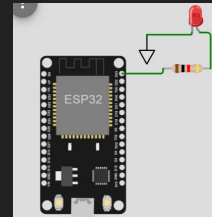
[illegible]

# Questions and Exercises

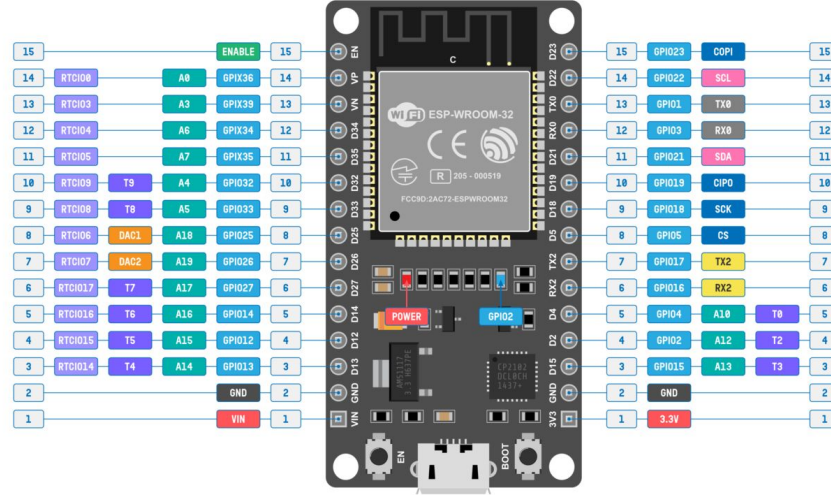
1. Find and list 3 sites where you can buy an esp32 board.
  - a. [https://www.canadarobotix.com/products/2594?variant=32091333328945&\\_gsid=tKA5uafthtF1&utm\\_source=chatgpt.com](https://www.canadarobotix.com/products/2594?variant=32091333328945&_gsid=tKA5uafthtF1&utm_source=chatgpt.com)
  - b. [https://www.elektor.com/products/lilygo-ttgo-t-display-esp32-development-board-16-mb?variant=46907157905740&\\_gsid=tKA5uafthtF1&utm\\_source=chatgpt.com](https://www.elektor.com/products/lilygo-ttgo-t-display-esp32-development-board-16-mb?variant=46907157905740&_gsid=tKA5uafthtF1&utm_source=chatgpt.com)
  - c. [https://ezsbc.shop/products/esp32-breakout-and-development-board?variant=44055842652315&\\_gsid=tKA5uafthtF1&utm\\_source=chatgpt.com](https://ezsbc.shop/products/esp32-breakout-and-development-board?variant=44055842652315&_gsid=tKA5uafthtF1&utm_source=chatgpt.com)
2. Perform some research and list some differences between the esp32 and its predecessor the esp866.

The ESP32 offers enhanced performance, more GPIOs, and additional features like Bluetooth support, making it suitable for complex projects. In contrast, the ESP8266 is cost-effective and ideal for simpler applications

3. Find a pinout of the esp32 that is different from the one included in this ppt and embed on the next slide.
4. Using the pinout determine how many GND pins are available. **3**
5. We will be using <https://wokwi.com/> to help us design and simulate some esp32 circuits. Sign up for an account there.
6. Using wokwi design the following circuit:



3. Find a pinout of the esp32 that is different from the one included in this ppt and embed on the next slide.



PHYSICAL PIN	POSITIVE SUPPLY	DAC OUTPUTS	SPI PINS
CONTROL PINS	GROUND SUPPLY	TOUCH INPUTS	UART PINS
GPIO PINS	ADC INPUTS	I2C PINS	EXCLUDED PINS

- GPIO pins 34, 35, 36 and 39 are input only.
- TX0 and RX0 (Serial0) are used for serial programming.
- TX2 and RX2 can be accessed as Serial2.
- Default SPI is VSP1. Both VSP1 and HSP1 pins can be set to any GPIO pins.
- All GPIO pins support PWM and interrupts.
- Built-in LED is connected to GPIO2.
- Some GPIO pins are used for interfacing flash memory and thus are not shown.

7. Create a simple LED flashing program with the built in LED on the ESP32. The pin attached to this LED should be #2. First perform a test with the wokwi simulator and then test your code on your ESP32.

// Define LED pin

const int ledPin = 2; // Built-in LED is usually on GPIO 2

void setup() {

    // Initialize the LED pin as an output

    pinMode(ledPin, OUTPUT);

}

void loop() {

    digitalWrite(ledPin, HIGH); // Turn the LED on

    delay(1000); // Wait for 1 second

    digitalWrite(ledPin, LOW); // Turn the LED off

    delay(1000); // Wait for 1 second

}