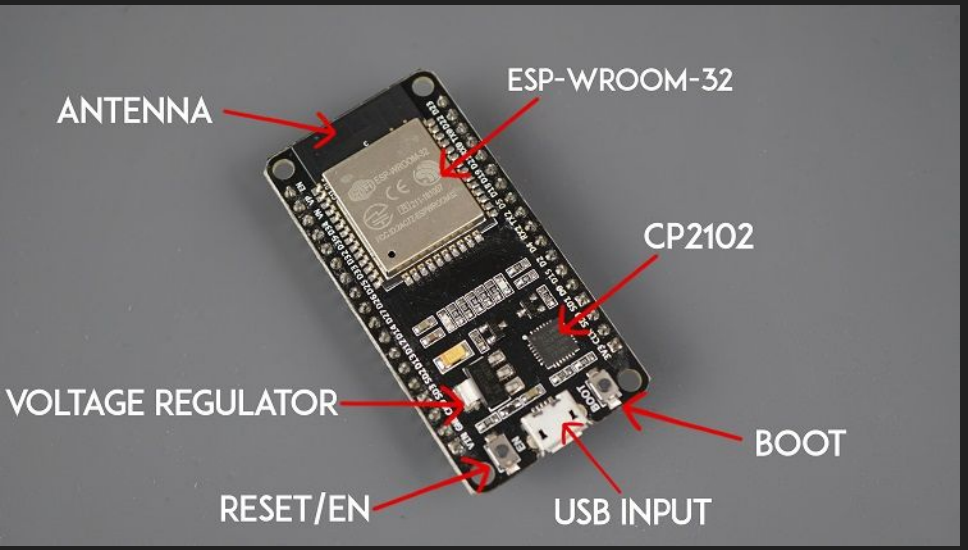
ESP32

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

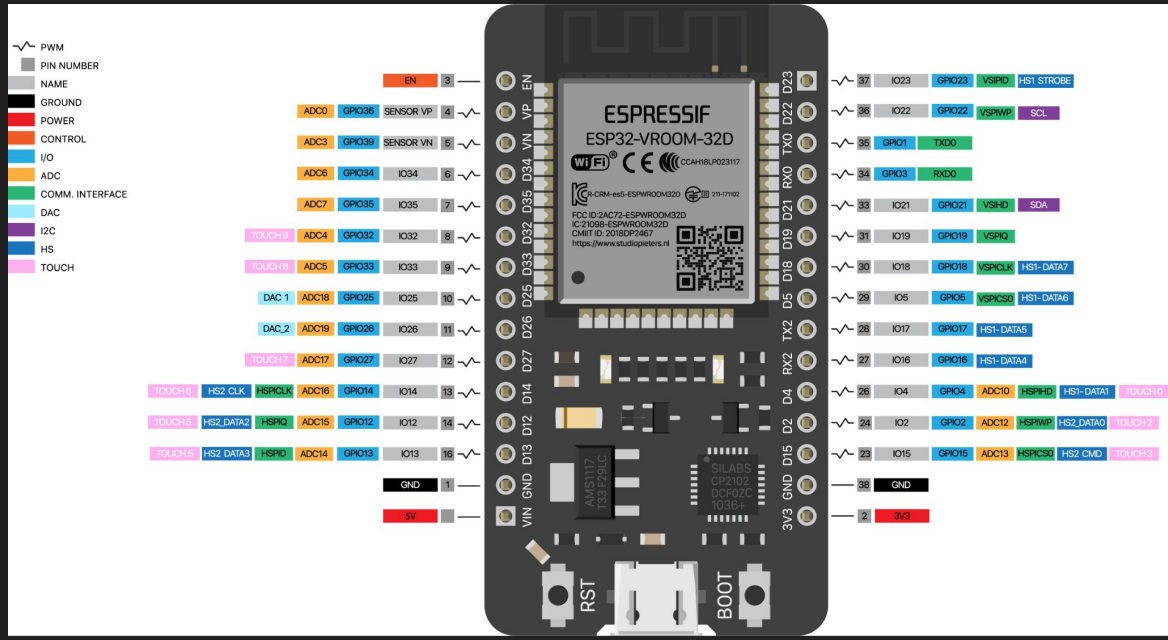
ESP-WROOM-32-chip(brains) or the microcontroller

GPIO

General Purpose Input Output Pins

GPIO 2-internally connected LED (blue)

Red LED-power indicator



Programming

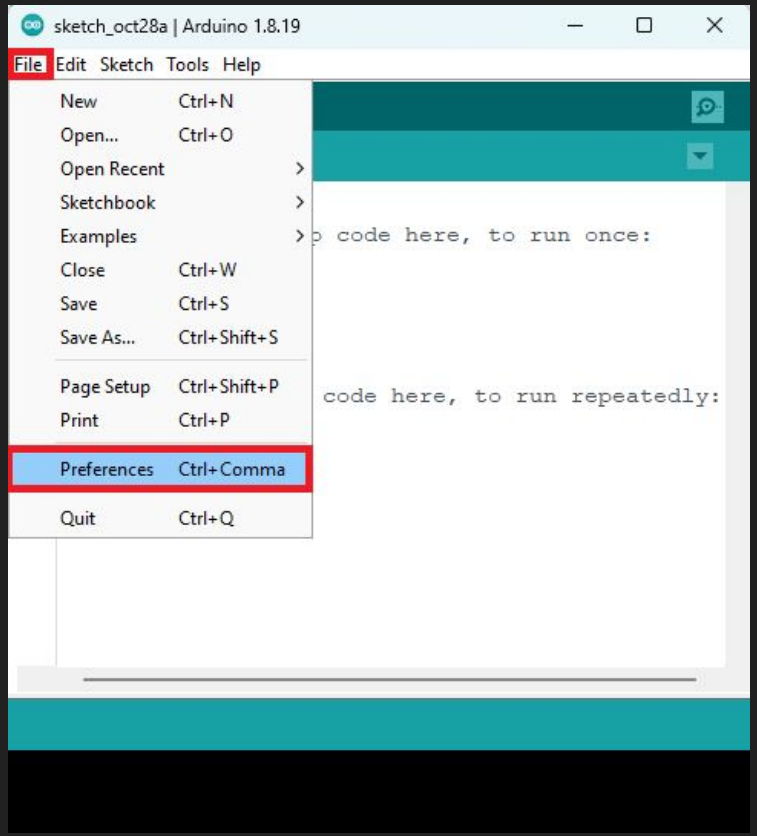
1. Download the Arduino IDE at https://www.arduino.cc/en/Main/Software

   School download link: https://drive.google.com/file/d/1VTVKKWvT9BWmKFO6p2zCBa12JeDmEdkT/view?thuser=0

2. Use the legacy version

Set up the Arduino IDE to use the ESP32

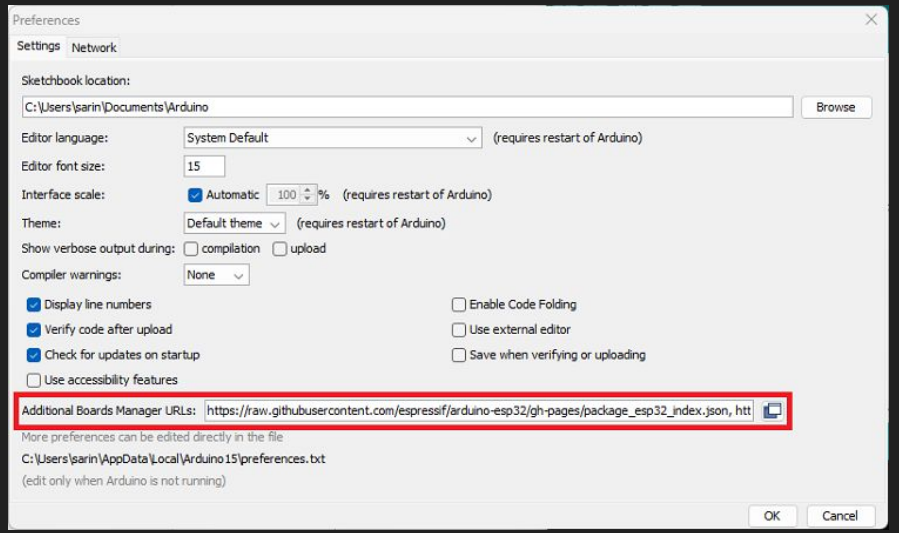
Go to **File>Preferences**



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
```

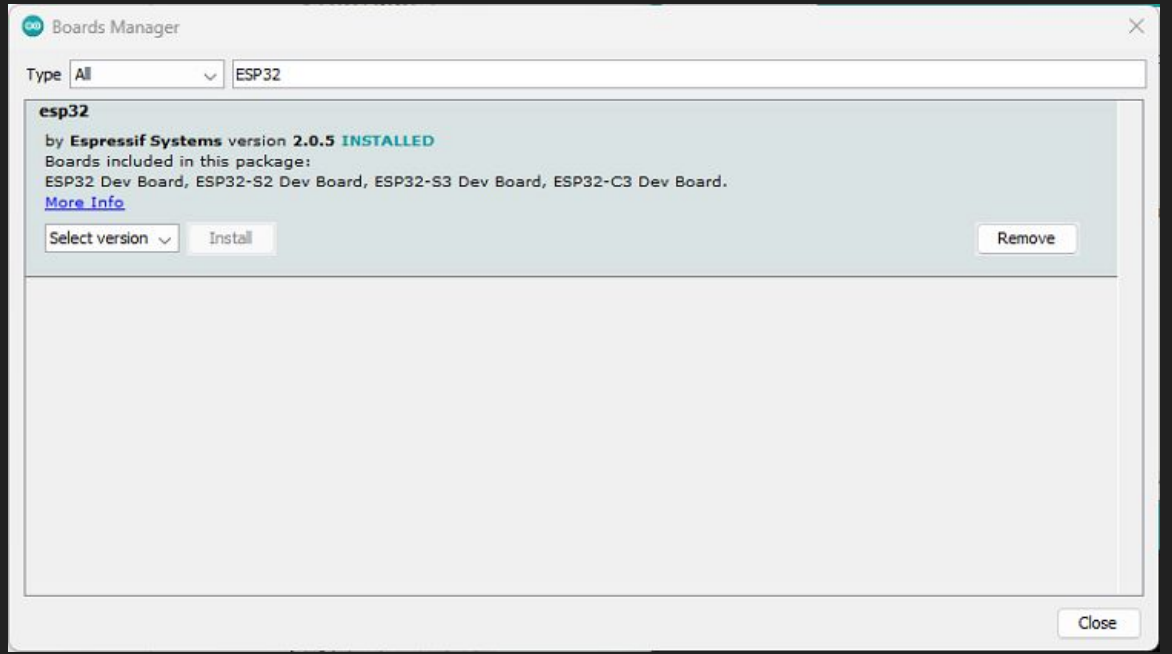


Install the board

Tools>Board>Boards Manager

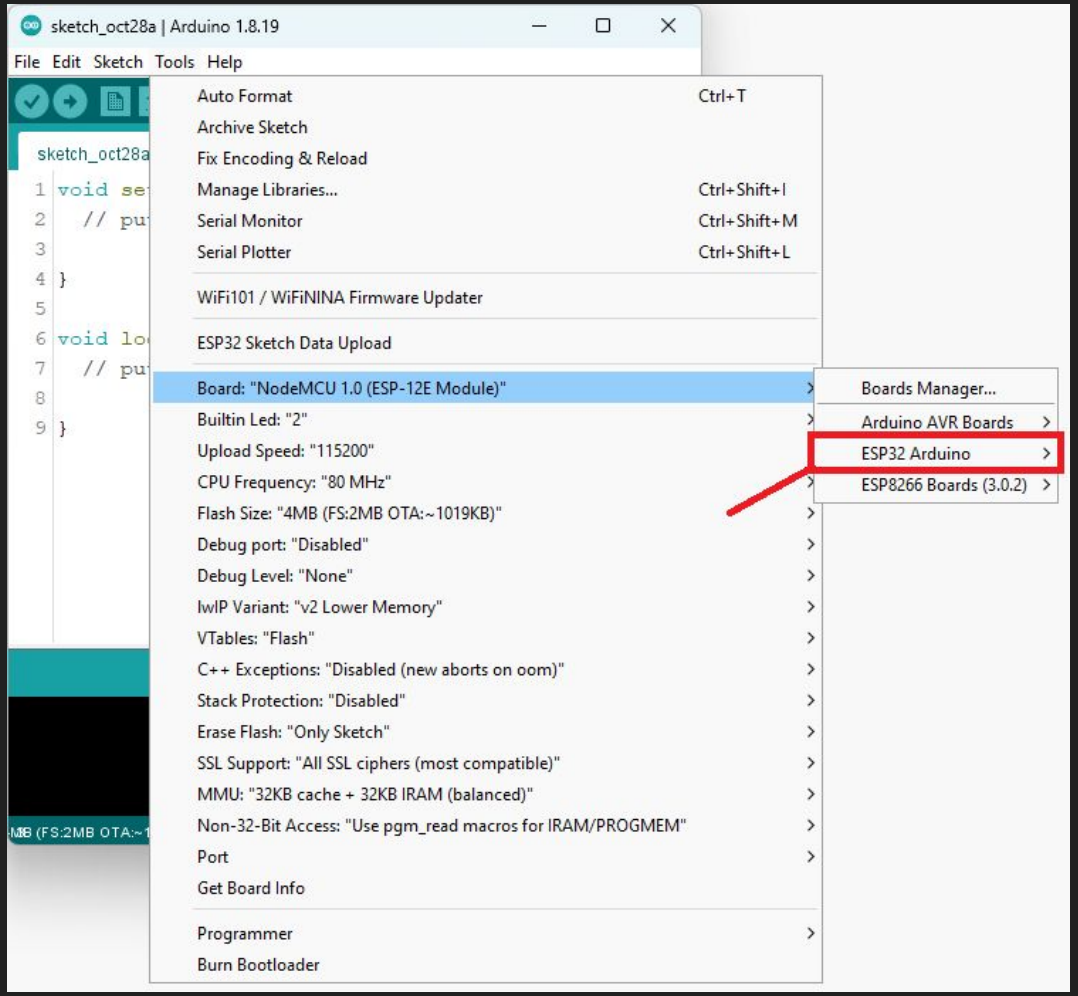
Search for esp32 and click Install

Restart Arduino



Select the board

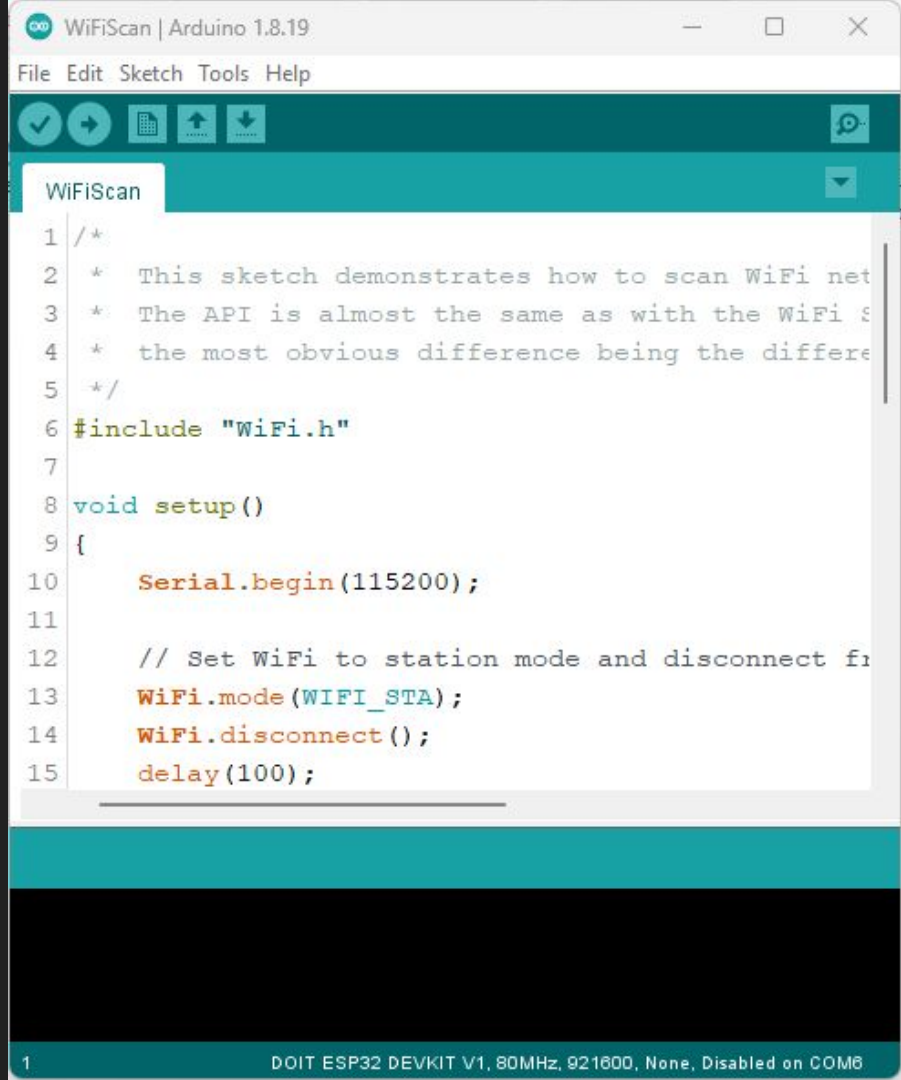
Tools>Board



Upload Code

Open sample code:

File>Examples>WiFi>WiFiScan



Next Steps

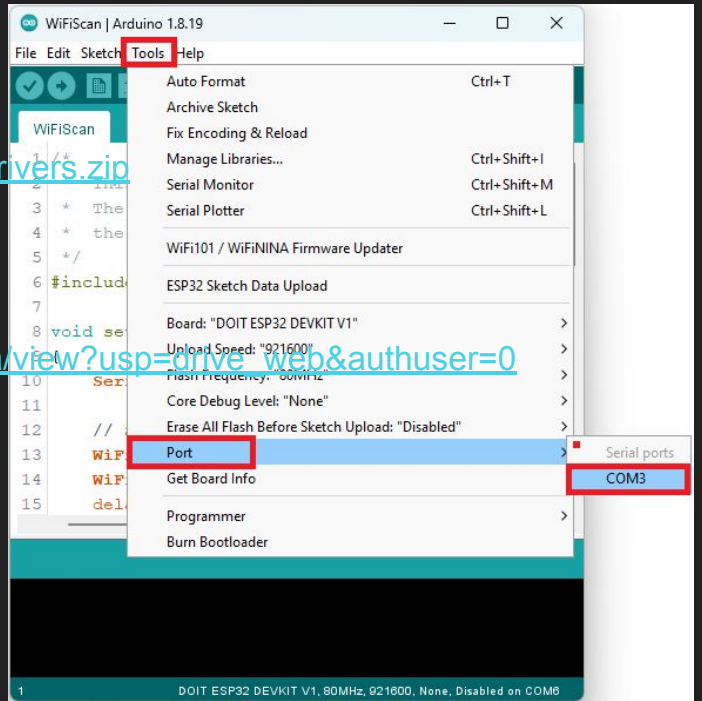
Tools>Board>select the proper board (ESP32 Dev Module)

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)]

Specific Download Link:

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



Upload the Code

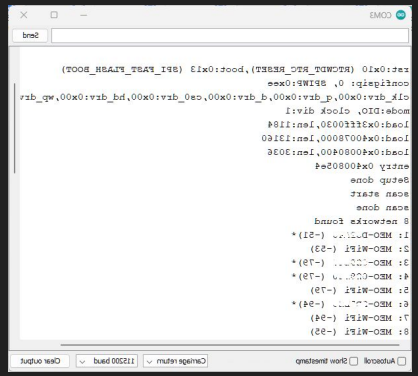
Press the Upload button

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:

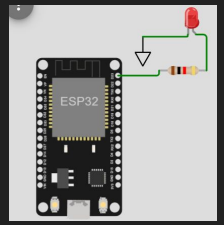


Questions and Exercises

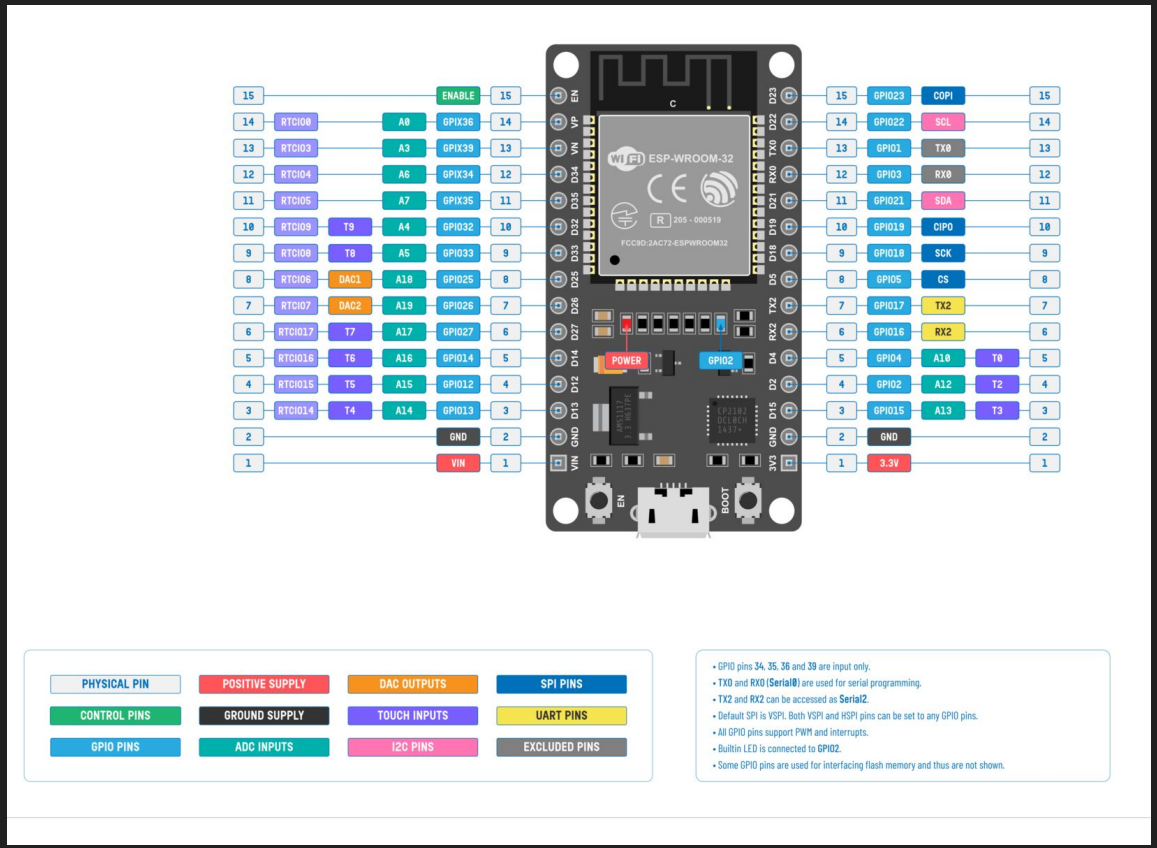
1. Find and list 3 sites where you can buy an esp32 board.
   a. https://www.canadarobotix.com/products/2594?variant=32091333328945&_gsid=tKA5uafthtF1&utm_source=chatgpt.com
   b. https://www.elektor.com/products/lilygo-ttgo-t-display-esp32-development-board-16-mb?variant=46907157905740&_gsid=tKA5uafthtF1&utm_source=chatgpt.com
   c. https://ezsbc.shop/products/esp32-breakout-and-development-board?variant=44055842652315&_gsid=tKA5uafthtF1&utm_source=chatgpt.com
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.



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
}
```