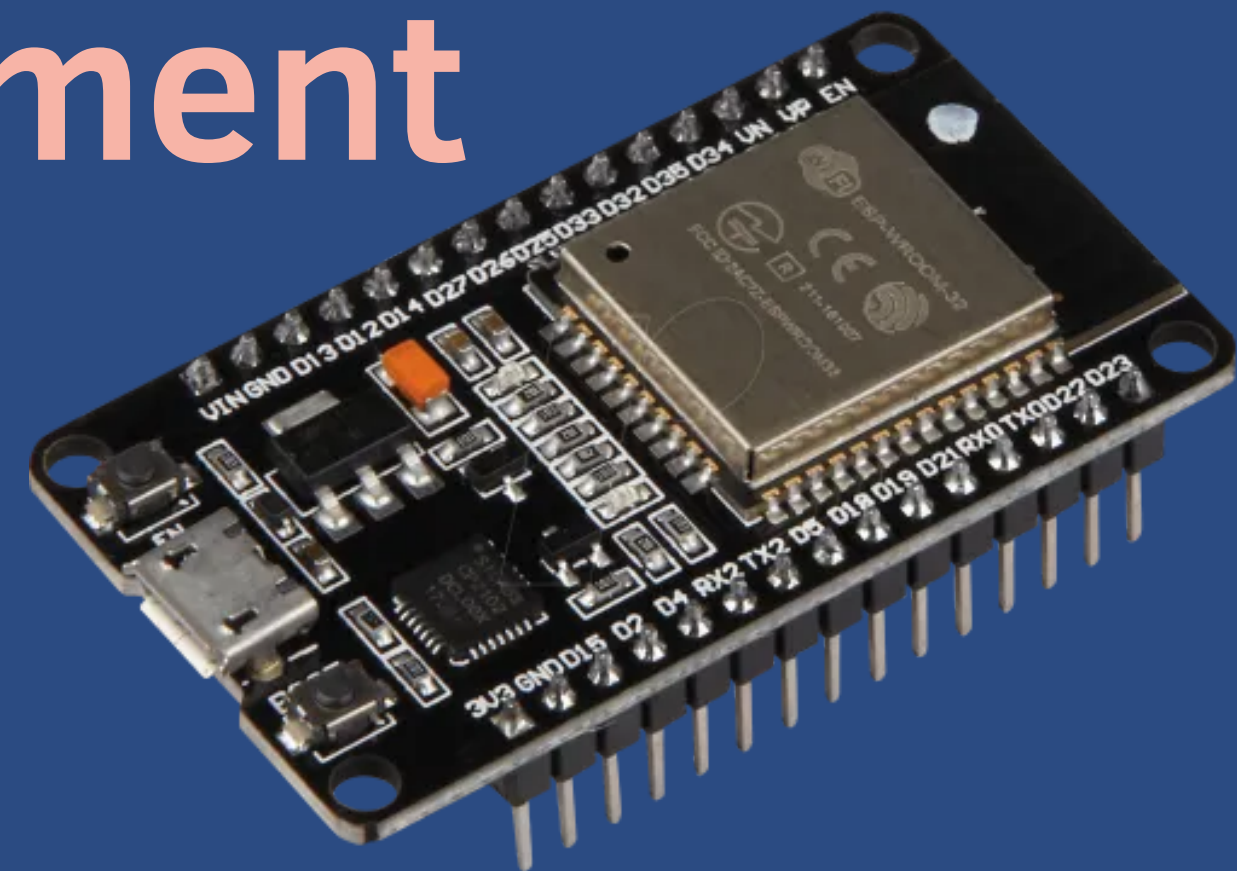
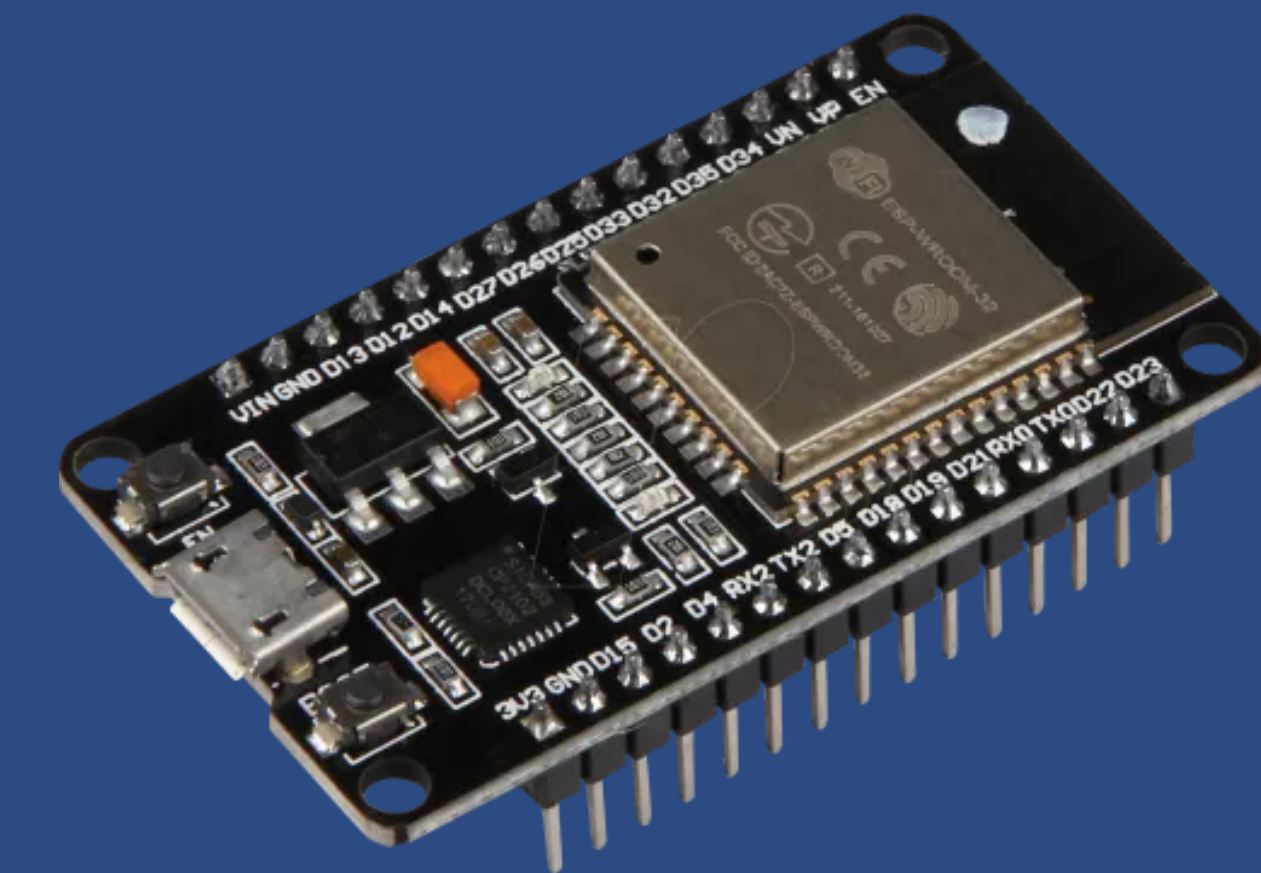




Getting Started with the ESP32 Development Board





ESP32 IS A SERIES OF LOW-COST, LOW-POWER SYSTEM-ON-A-CHIP (SOC) MICROCONTROLLERS WITH INTEGRATED WI-FI AND DUAL-MODE BLUETOOTH CAPABILITIES.

- The ESP32 is a popular choice for implementing Internet of Things (IoT) projects due to its low cost, small size, and low power consumption. It has a dual-core microprocessor, built-in Wi-Fi and Bluetooth connectivity, and a variety of peripherals such as ADC, DAC, and touch sensors. It can be programmed using the Arduino Integrated Development Environment (IDE) or using the Espressif IoT Development Framework (ESP-IDF).
- Some common applications of the ESP32 include home automation, wearable electronics, industrial control, and smart appliances. It is also commonly used as a development board for prototyping and testing new IoT applications.



Where is ESP32 used?

1

2

3

4

5

HOME AUTOMATION:

You can use the ESP32 to control and monitor devices around your homes, such as lights, thermostats, and appliances.

SENSOR NETWORKS

You can use the ESP32 to gather data from sensors and transmit it to a central server or cloud platform for further analysis and processing.

INDUSTRIAL CONTROL

You can use the ESP32 to control and monitor industrial processes and equipment, such as pumps, valves, and conveyor belts.

WIRELESS COMMUNICATIONS

You can use the ESP32's built-in Wi-Fi and Bluetooth capabilities to connect to other devices and send and receive data wirelessly.

INTERNET OF THINGS (IOT)

You can use the ESP32 to build a variety of IoT applications, such as smart home systems, wearable devices, and smart city infrastructure.

How to Program ESP32?

THERE ARE SEVERAL WAYS TO PROGRAM THE ESP32 MICROCONTROLLER. HERE ARE THE TWO MOST COMMON METHODS:

- Using the Arduino Integrated Development Environment (IDE):
The ESP32 can be programmed using the Arduino IDE, which is a popular open-source platform for developing and testing microcontroller-based projects.
- Using the Espressif IoT Development Framework (ESP-IDF):
The ESP-IDF is a set of development tools and libraries specifically designed for the ESP32 microcontroller. It provides a more powerful and flexible development environment than the Arduino IDE, but it also requires a more in-depth understanding of programming and microcontroller hardware.



Programming ESP32 Board with Arduino IDE



1. **INSTALL THE ARDUINO IDE:** DOWNLOAD AND INSTALL THE LATEST VERSION OF THE ARDUINO IDE FROM THE OFFICIAL ARDUINO WEBSITE ([HTTPS://WWW.ARDUINO.CC/EN/MAIN/SOFTWARE](https://www.arduino.cc/en/main/software)).
2. **INSTALL THE ESP32 BOARD DEFINITION FILES:** OPEN THE ARDUINO IDE AND GO TO "FILE" > "PREFERENCES". IN THE "ADDITIONAL BOARDS MANAGER URLS" FIELD, ENTER THE FOLLOWING URL: "[HTTPS://DL.ESPRESSIF.COM/DL/PACKAGE_ESP32_INDEX.JSON](https://dl.espressif.com/dl/package_esp32_index.json)" THEN, GO TO "TOOLS" > "BOARD" > "BOARDS MANAGER" AND SEARCH FOR "ESP32". SELECT THE "ESP32 BY ESPRESSIF SYSTEMS" PACKAGE AND CLICK "INSTALL".
3. **CONNECT THE ESP32 TO YOUR COMPUTER:** CONNECT THE ESP32 TO YOUR COMPUTER USING A USB-TO-SERIAL CONVERTER AND A COMPATIBLE USB CABLE. YOU'LL ALSO NEED TO INSTALL THE NECESSARY DRIVERS FOR YOUR USB-TO-SERIAL CONVERTER IF YOU'RE USING WINDOWS.
4. **SELECT THE ESP32 BOARD FROM THE ARDUINO IDE:** GO TO "TOOLS" > "BOARD" AND SELECT THE "ESP32 DEV MODULE" FROM THE LIST OF AVAILABLE BOARDS.
5. **WRITE AND UPLOAD CODE TO THE ESP32:** WITH THE ESP32 BOARD SELECTED, YOU CAN NOW WRITE AND UPLOAD CODE TO THE MICROCONTROLLER USING THE ARDUINO IDE. SIMPLY WRITE YOUR CODE IN THE ARDUINO IDE EDITOR, THEN CLICK THE "UPLOAD" BUTTON TO COMPILE AND UPLOAD IT TO THE ESP32.

ESP32 Led Blink



1.	Connect the LED to the ESP32: Connect the positive leg of the LED (the longer leg) to a digital output pin on the ESP32, and the negative leg (the shorter leg) to a ground pin on the ESP32.
2.	Write the code: Open the Arduino IDE and enter the following code:
3.	Upload the code to the ESP32: Connect the ESP32 to your computer using a USB-to-serial converter and a compatible USB cable. Select the correct COM port and board type in the Arduino IDE (see previous answer for more information). Then, click the "Upload" button to compile and upload the code to the ESP32.
4.	Watch the LED blink: The LED should start blinking on and off at a rate of 1 second per blink. If the LED does not blink, check your connections and make sure that the code is correct.

```
void setup() {  
  // Set the LED pin as an output  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  // Turn on the LED  
  digitalWrite(LED_BUILTIN, HIGH);  
  // Wait for 1000 milliseconds (1 second)  
  delay(1000);  
  // Turn off the LED  
  digitalWrite(LED_BUILTIN, LOW);  
  // Wait for 1000 milliseconds (1 second)  
  delay(1000);  
}
```