

# *Practical IoT (Internet of Things)*

## *BSCIS – DCIS, PIEAS*

### Lab 00: Getting Started with IoT Development

#### Objective

This lab introduces students to IoT development by setting up the Arduino IDE, installing ESP8266 and ESP32 boards, and flashing a basic LED blinking firmware onto both boards. Students will also create a circuit sketch using Fritzing.

#### Required Components

- ESP8266 (NodeMCU) and ESP32 boards
- USB cable (Micro-USB for NodeMCU, USB-C or Micro-USB for ESP32)
- LED (optional for external connection)
- Resistor (330Ω if using an external LED)
- Breadboard and jumper wires (if needed)
- Computer with Internet access

#### Step 1: Installing Arduino IDE

1. Download and install **Arduino IDE** from <https://www.arduino.cc/en/software>.

#### Downloads



 **Arduino IDE 2.3.4**

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.

**SOURCE CODE**

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

**DOWNLOAD OPTIONS**

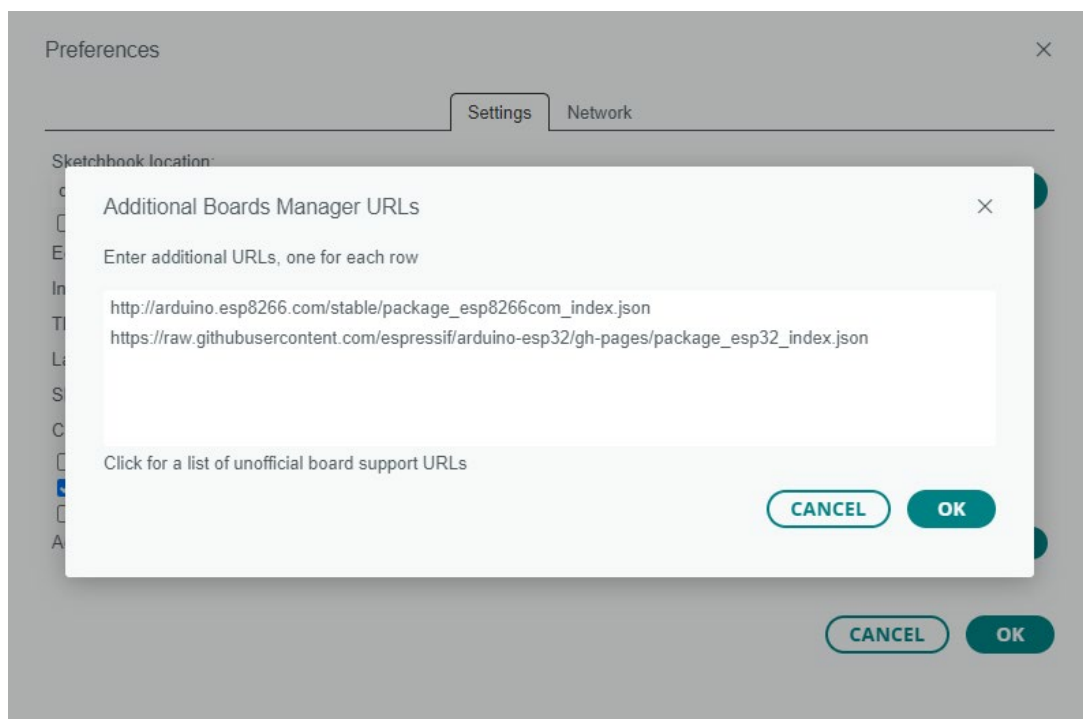
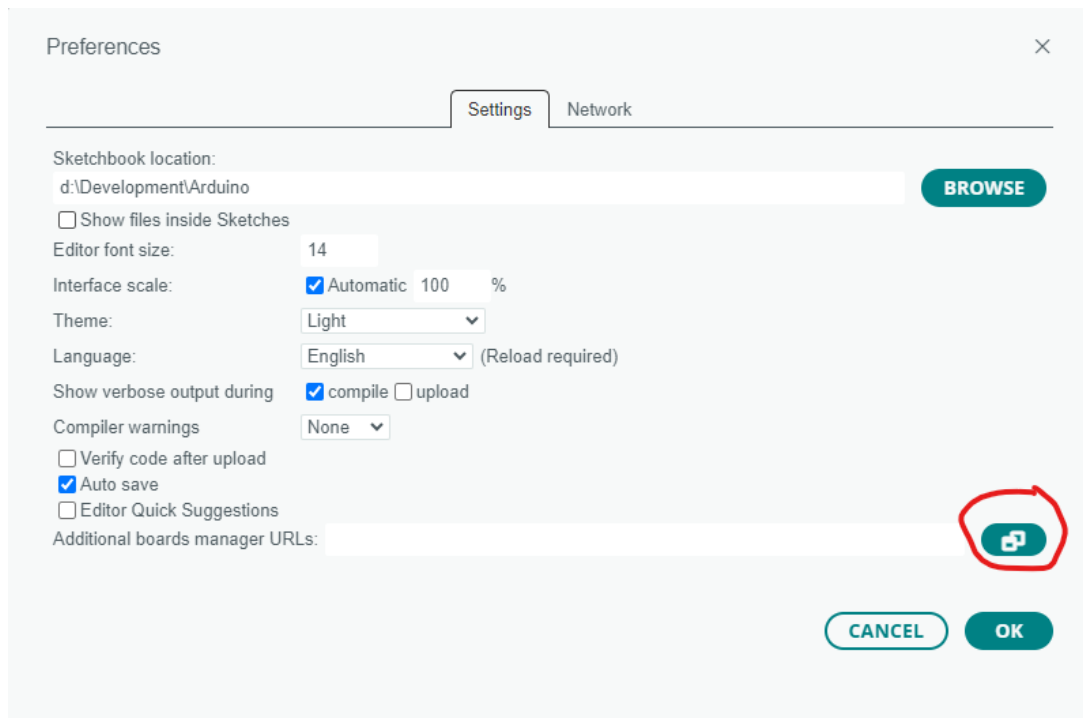
- Windows** Win 10 and newer, 64 bits
- Windows** MSI installer
- Windows** ZIP file
- Linux** ApptImage 64 bits (X86-64)
- Linux** ZIP file 64 bits (X86-64)
- macOS** Intel, 10.15: "Catalina" or newer, 64 bits
- macOS** Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

2. Open Arduino IDE and navigate to **File → Preferences**.

3. In the **Additional Board Manager URLs** field, enter:

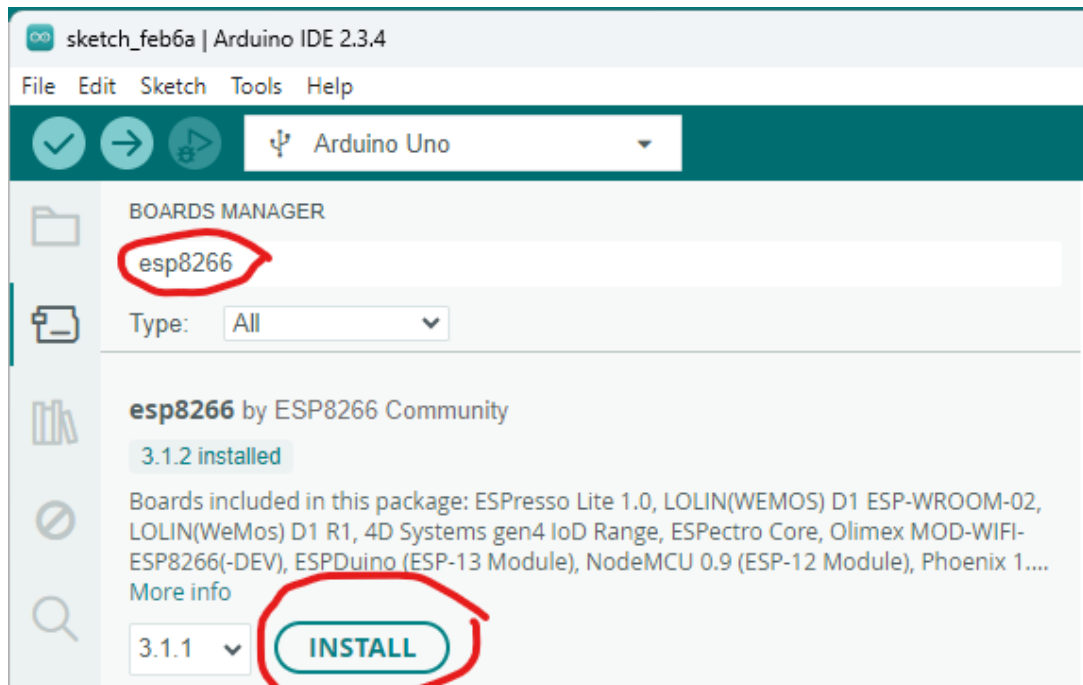
[https://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](https://arduino.esp8266.com/stable/package_esp8266com_index.json),  
[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)



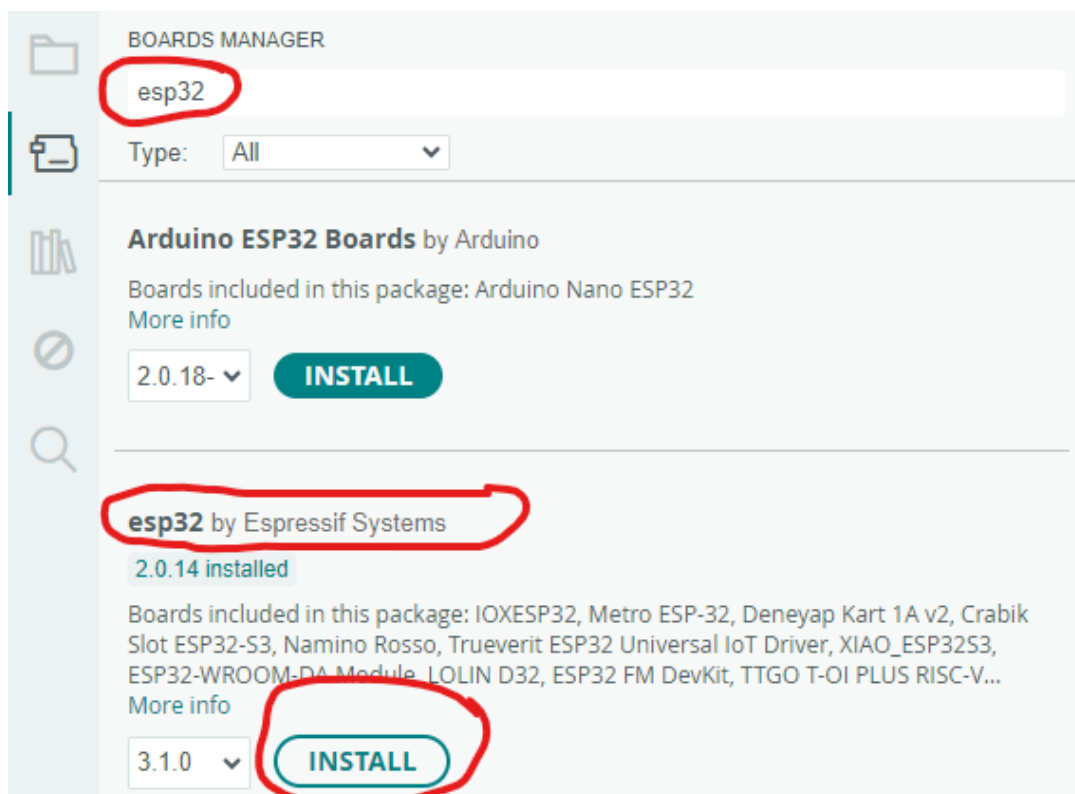
4. Click **OK** and close the Preferences window.

## Step 2: Installing ESP8266 and ESP32 Board Packages

1. Navigate to **Tools** → **Board** → **Board Manager**.
2. Search for **ESP8266** and install the latest version.



3. Search for **ESP32** and install the latest version.



4. Restart the Arduino IDE to apply changes.

### Step 3: Flashing LED Blinking Code

Once the setup is complete, we will upload a simple LED blinking code to both ESP8266 and ESP32.

## Code for ESP8266 (NodeMCU) and ESP32

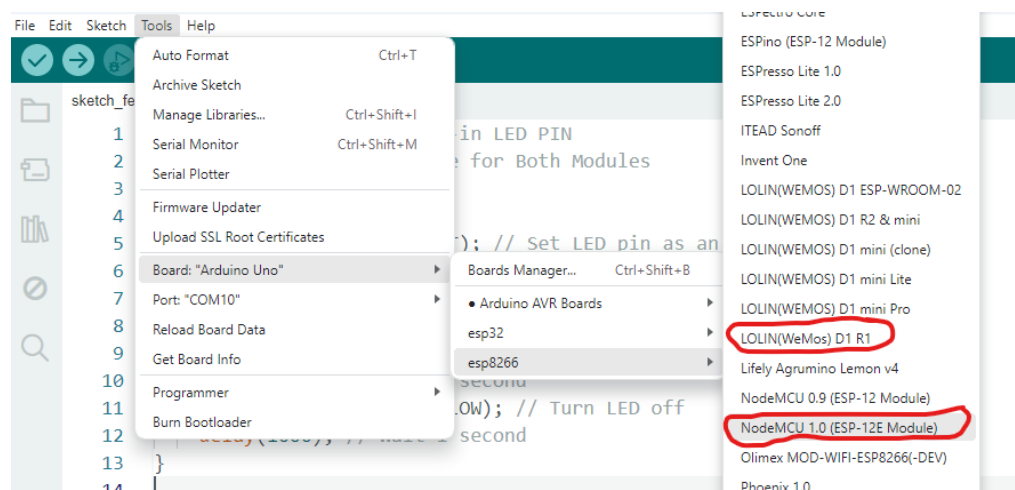
```
sketch_feb6a.ino
1  #define LED_PIN 2 // Built-in LED PIN
2  | | | | | | | | // Change for Both Modules
3
4  void setup() {
5  |   pinMode(LED_PIN, OUTPUT); // Set LED pin as an output
6  }
7
8  void loop() {
9  |   digitalWrite(LED_PIN, HIGH); // Turn LED on
10 |   delay(1000); // Wait 1 second
11 |   digitalWrite(LED_PIN, LOW); // Turn LED off
12 |   delay(1000); // Wait 1 second
13 }
14
```

## Explanation of the Code

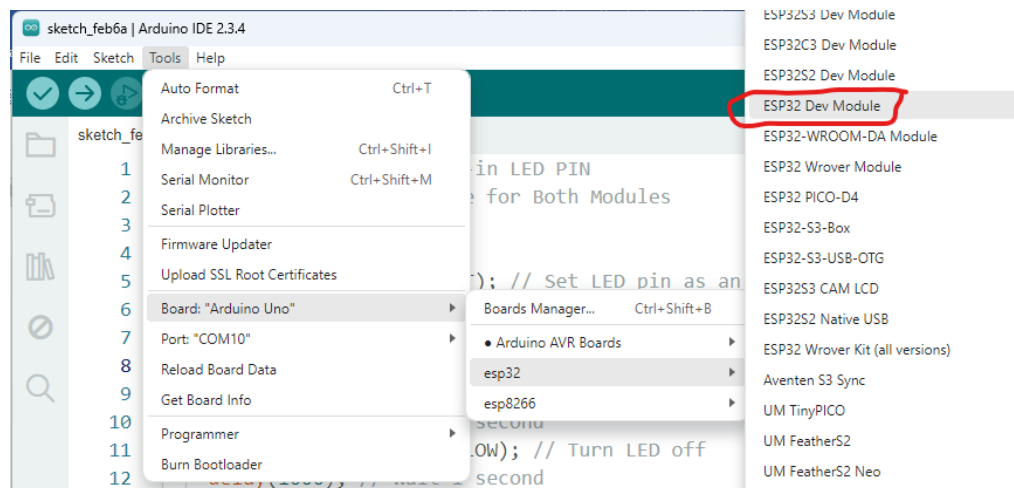
- **#define LED\_PIN 2:** Defines pin 2 as the LED pin (built-in LED on most ESP8266 and ESP32 boards).
- **void setup():** Runs once when the board starts, setting the LED pin as an output.
- **void loop():** Repeats indefinitely, turning the LED on for 1 second and then off for 1 second, creating a blinking effect.

## Step 4: Uploading the Code

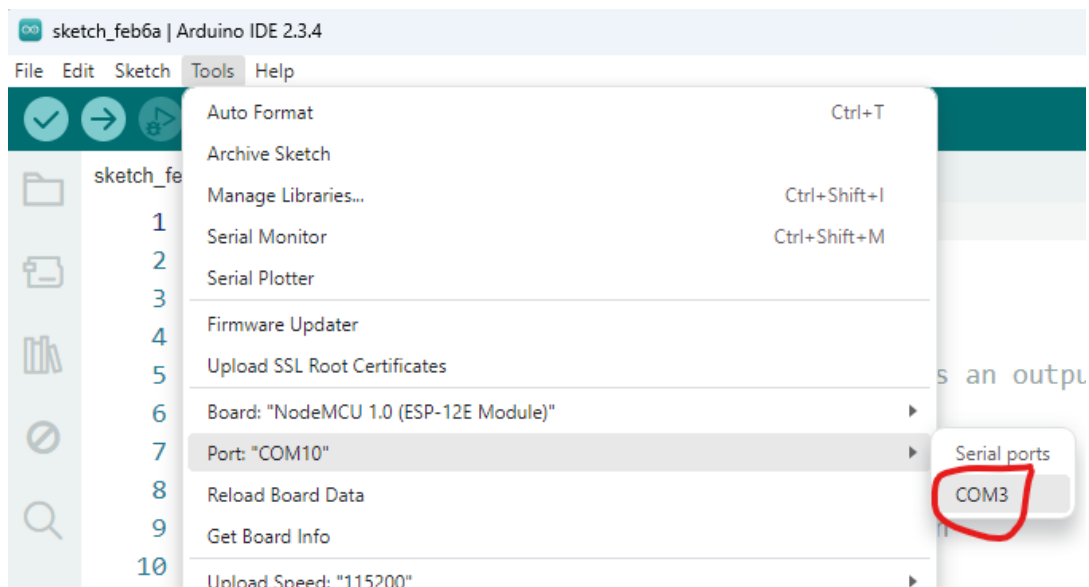
1. Connect the ESP8266/ESP32 board to your computer via USB.
2. In Arduino IDE, go to **Tools** → **Board** and select the appropriate board:
  - **NodeMCU 1.0 (ESP8266)** for ESP8266 (Depends on the board you using)



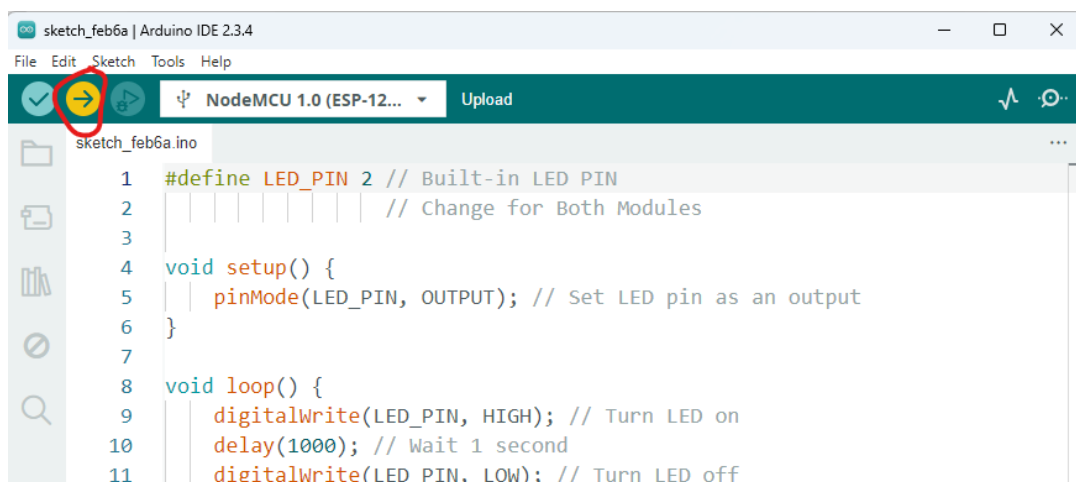
- **ESP32 Dev Module for ESP32**



3. Select the correct **COM port** under **Tools** → **Port**. (com port number varies based on system)



4. Click the **Upload** button (arrow icon) to flash the firmware.



```
Output
Using precompiled core: C:\Users\Naveed\AppData\Local\arduino\cores\5b058861dd466
Linking everything together...
"C:\Users\Naveed\AppData\Local\Arduino15\packages\esp8266\tools\python3\
"C:\Users\Naveed\AppData\Local\Arduino15\packages\esp8266\tools\python3\
"C:\Users\Naveed\AppData\Local\Arduino15\packages\esp8266\tools\python3\
"C:\Users\Naveed\AppData\Local\Arduino15\packages\esp8266\tools\python3\
Compiling sketch...
CANCEL
```

Ln 1, Col 1 NodeMCU 1.0 (ESP-12E Module) on COM3

```
Writing at 0x00000000... (25 %)
Writing at 0x0000c000... (33 %)
Writing at 0x00010000... (41 %)
Writing at 0x00014000... (50 %)
Writing at 0x00018000... (58 %)
Writing at 0x0001c000... (66 %)
Writing at 0x00020000... (75 %)
Writing at 0x00024000... (83 %)
Writing at 0x00028000... (91 %)
Writing at 0x0002c000... (100 %)
Wrote 265616 bytes (195725 compressed) at 0x00000000 in 17.3 seconds (effective 1
Hash of data verified.

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

Ln 13, Col 2 NodeMCU 1.0 (ESP-12E Module) on COM3

5. Open **Serial Monitor** (74880 baud rate) to check logs. You may need to press reset button.

sketch\_feb6a | Arduino IDE 2.3.4

File Edit Sketch Tools Help

Auto Format  
Archive Sketch  
Manage Libraries...  
**Serial Monitor**  
Serial Plotter  
Firmware Updater  
Upload SSL Root Certificates

Output Serial Monitor x

Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM3') New Line 74880 baud

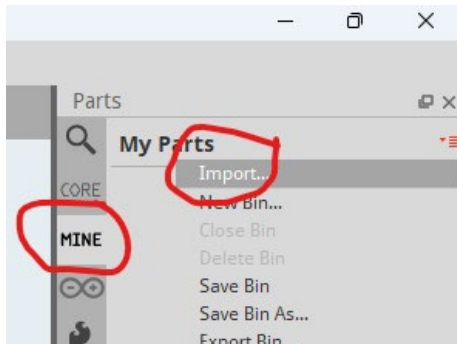
```
18:26:10.329 ->
18:26:10.329 -> ets Jan 8 2013,rst cause:2, boot mode:(3,7)
18:26:10.329 ->
18:26:10.329 -> load 0x4010f000, len 3424, room 16
18:26:10.329 -> tail 0
18:26:10.329 -> checksum 0x2e
18:26:10.329 -> load 0x3fff20b8, len 40, room 8
18:26:10.329 -> tail 0
18:26:10.329 -> checksum 0x2b
18:26:10.329 -> csum 0x2b
18:26:10.329 -> v00040d90
18:26:10.372 -> ~ld
18:26:10.418 -> rfc al sector: 1020
18:26:10.418 -> freq trace enable 0
18:26:10.418 -> rfc[112] : 0
```

Ln 13, Col 2 NodeMCU 1.0 (ESP-12E Module) on COM3

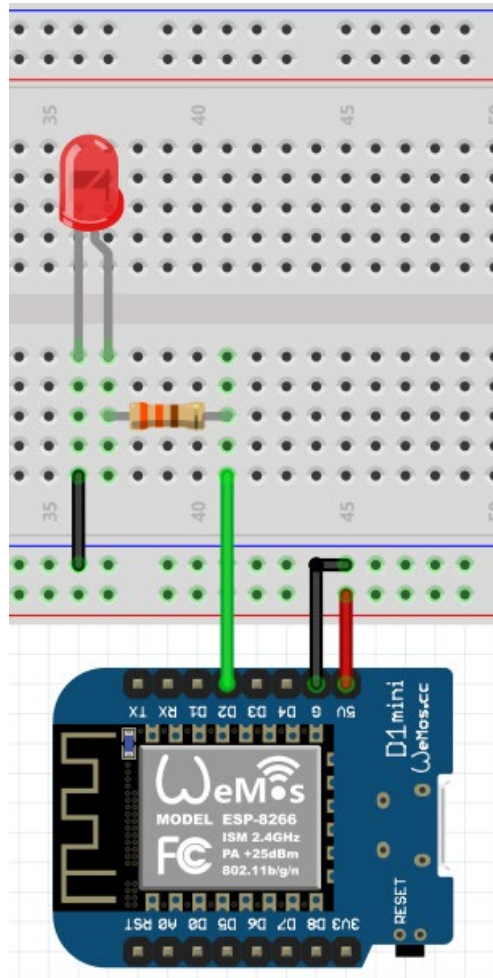
6. If everything is correct, the LED should start blinking.

## Step 5: Creating Circuit Sketch in Fritzing (Optional)

1. Download / unzip **Fritzing** from <https://idraak.pieas.edu.pk/index.php/s/Ms4RmitxaJCZyib>.
2. Import ESP32 into fritzing from <https://idraak.pieas.edu.pk/index.php/s/Q7TxYG3C92a8H7D>.



3. Open Fritzing and add components:
  - Select **ESP8266** or **ESP32** from the component list.
  - Add an LED and connect it to **GPIO2** through a **330Ω resistor**.
  - Connect **GND** to the other side of the LED.



## Conclusion

You have now successfully:

- Set up Arduino IDE.
- Installed ESP8266 and ESP32 board packages.
- Uploaded a basic LED blinking firmware.
- Created a circuit schematic using Fritzing.

## Submission

You must submit a single pdf file containing following items:

1. **Screenshot of Arduino IDE** showing successful compilation and upload (Students' Names should be in header comments of code file).
2. **visual proof** of the ESP8266 and ESP32 LEDs blinking.
3. **Screenshot of Fritzing circuit sketch** showing the LED connection.
4. **Brief report** (1-2 pages) explaining:
  - Steps taken in the lab.
  - Any challenges faced and solutions.
  - Explanation of the LED blinking code.