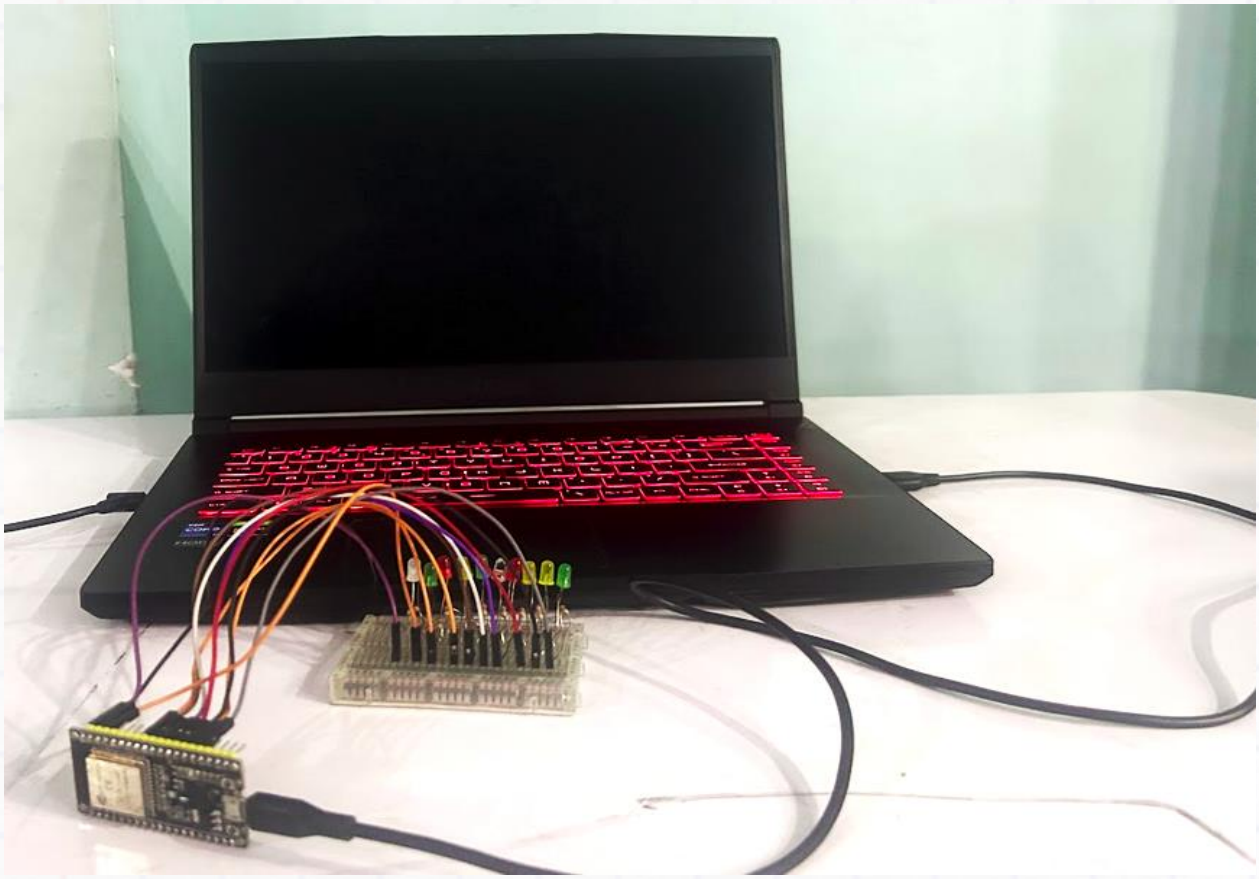
Build Dancing LED with ESP32 C-6



Contents

Prerequisites

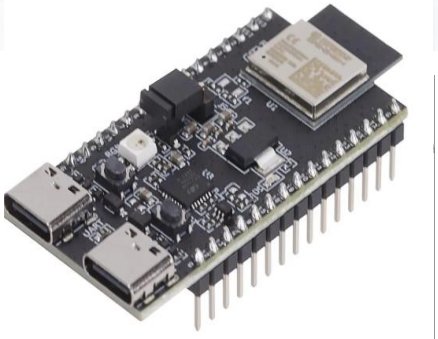
- What is a Microcontroller

Aim

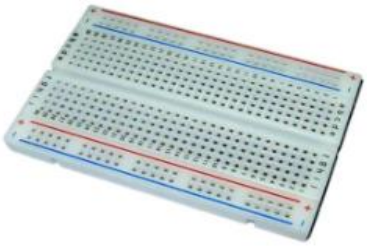
Learning the ESP32 module and Arduino IDE and using them to control LEDs.

Components

1. ESP32 C-6 Module
2. Breadboard (size = 400 point)
3. Jumper wires (male to female - 11nos.)
4. 1 USB cables
5. Code from laptop to ESP32.
6. 10 mini-LED bulbs
7. 10 220-ohm resistors



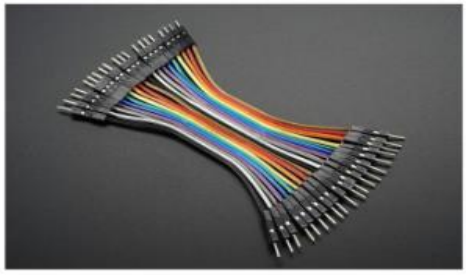
ESP32 C-6



BreadBoard

LED

Resistor



Jumper wires

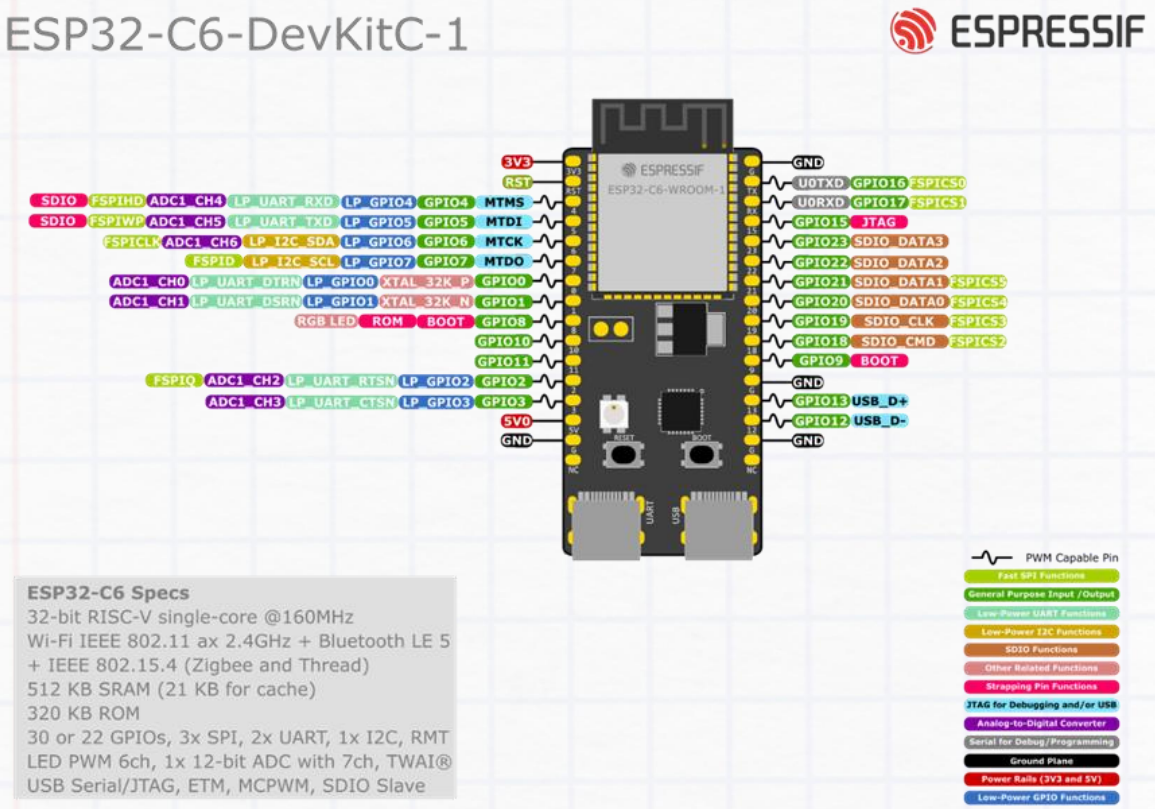
USB

Connections

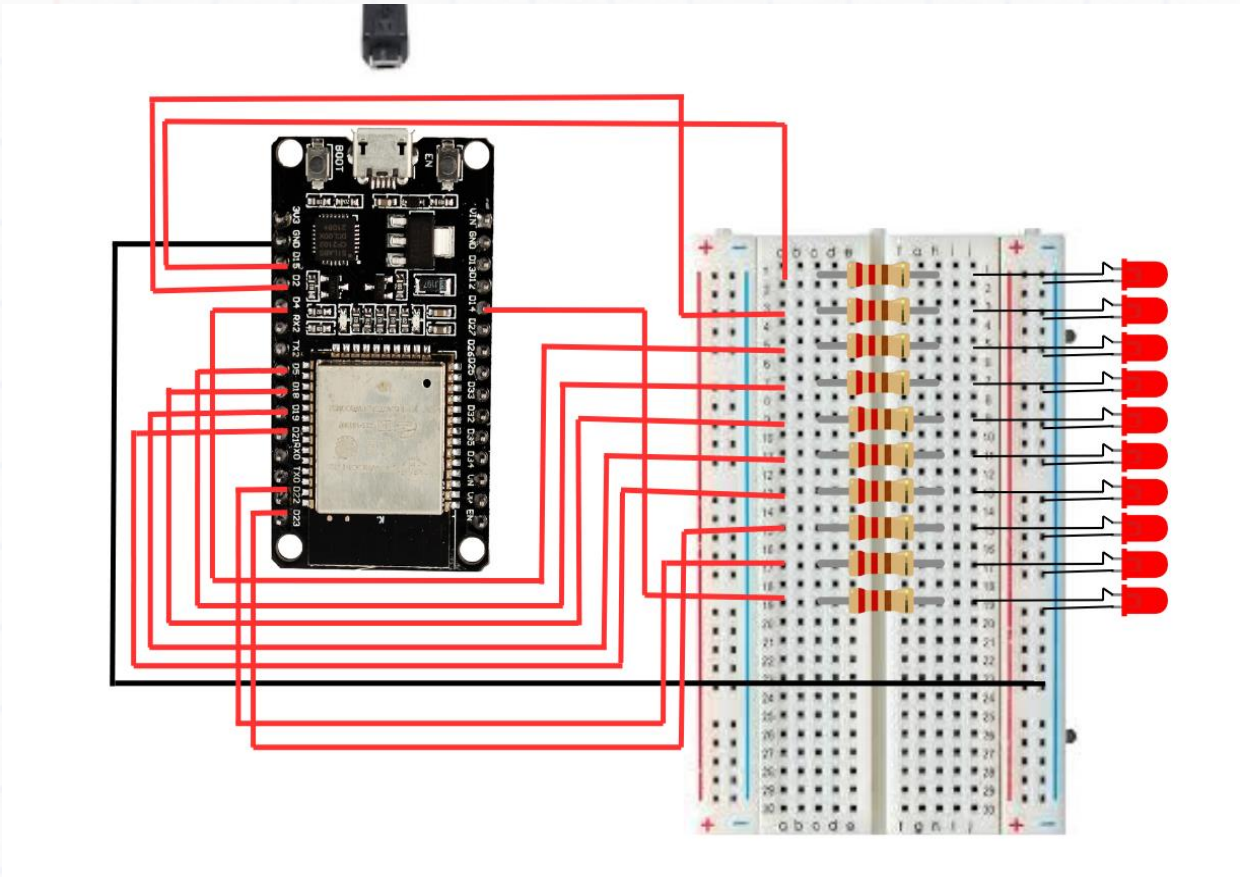
Safety Tip: Always ensure that the connections to the components are correct and completed before connecting the power supply to the ESP32. In LEDs, the positive electrode is the longer wire, and the negative electrode is the shorter wire. In a typical application, a resistor is connected in series with the LED to limit the amount of current and ensure the LED doesn't get damaged.

In LEDs, the positive electrode is the longer wire, and the negative electrode is the shorter wire. In a typical application, a resistor is connected in series with the LED to limit the amount of current and ensure the LED doesn't get damaged.

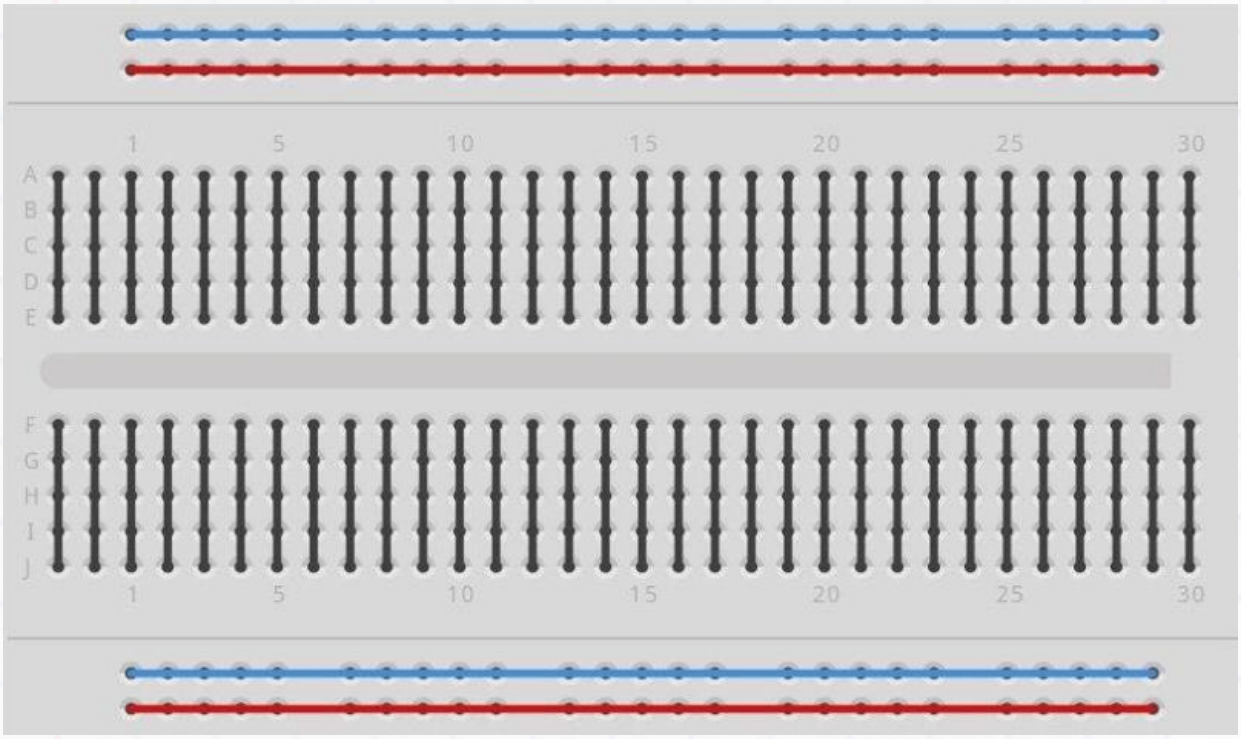
The ESP32 C-6 Module Pinout



Circuit Diagram



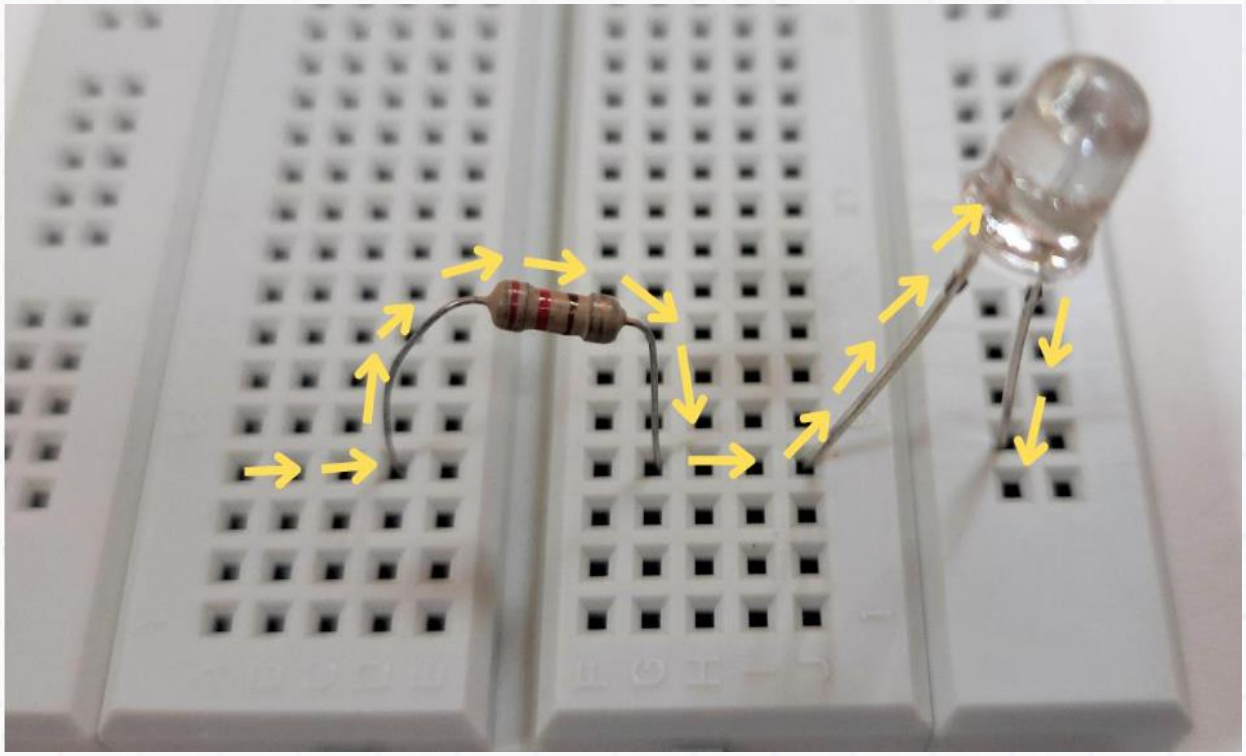
NOTE: Internal wiring of breadboard is as per the below diagram:



Follow the detailed steps in the following pages to complete the circuit.

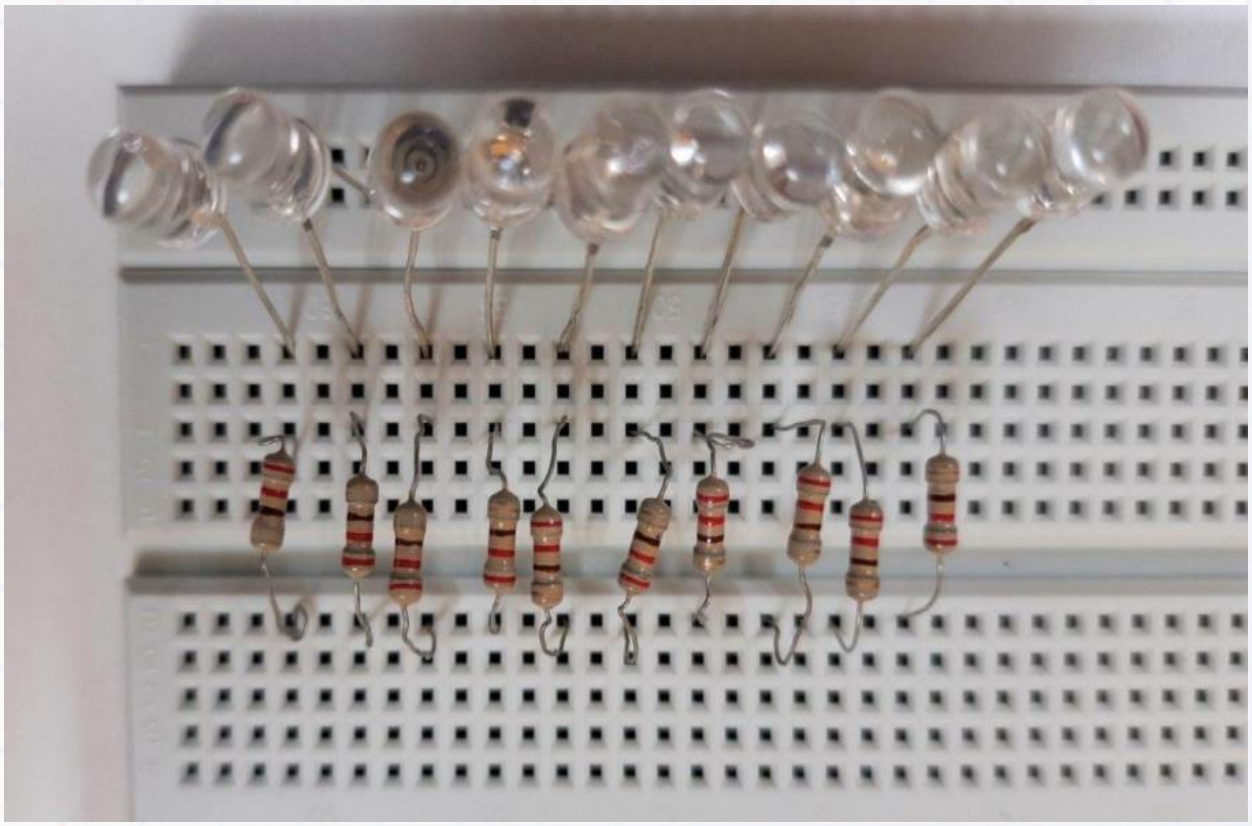
Detailed Connection Steps

Step 1



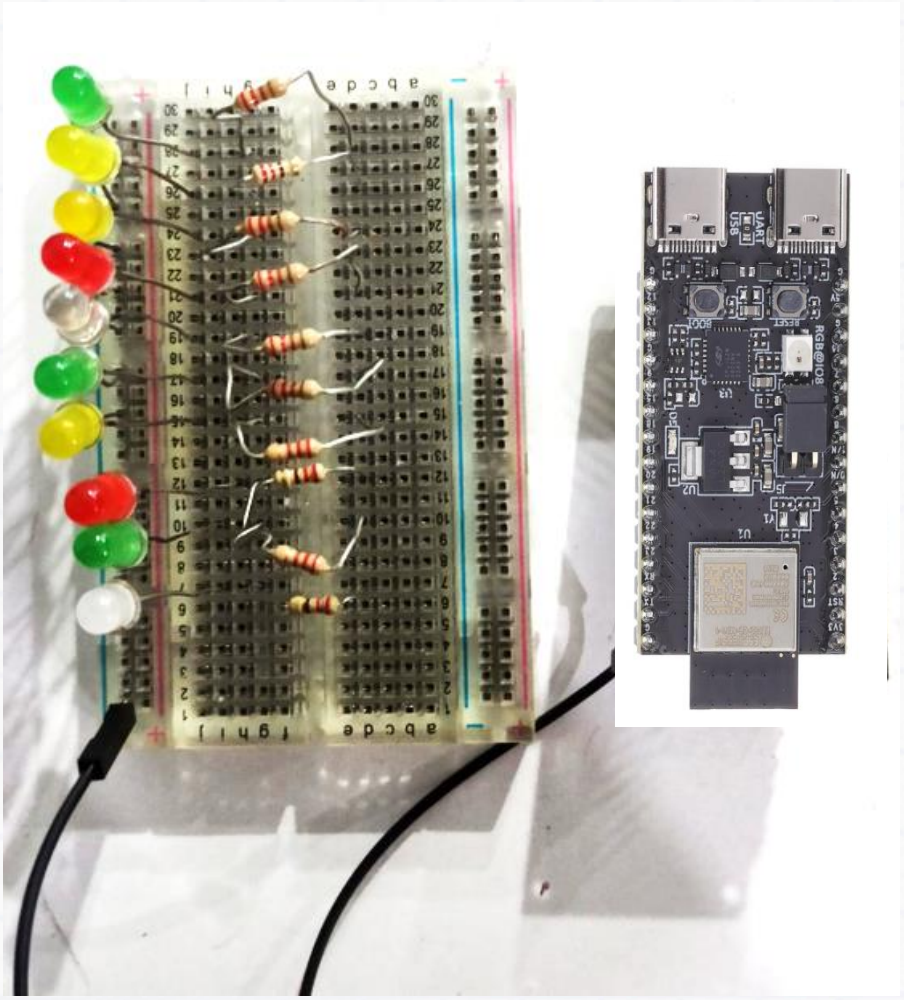
Connect the longer leg of the LED to the resistor and the shorter leg to the ground terminal. The circuit connection is as shown by the yellow arrows in the above image.

Step 2



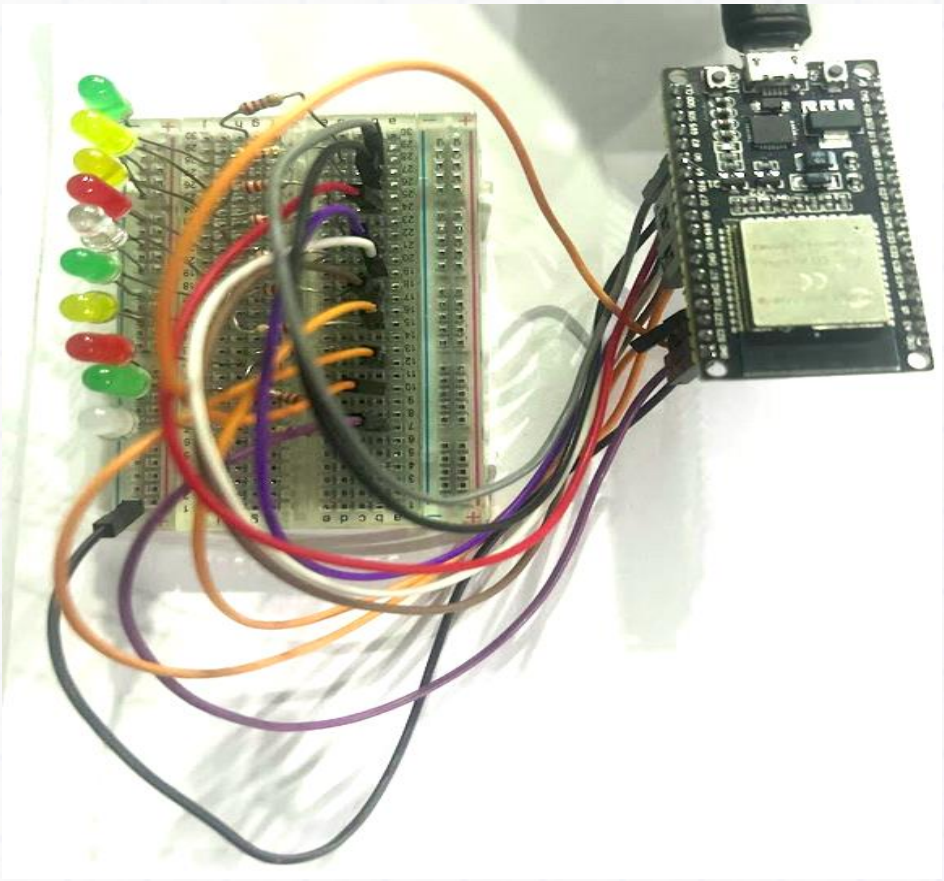
Repeat the connections similarly for the remaining LEDs (10 in total) as shown above. Ensure that the connections for each LED setup are in the same horizontal rows.

Step 3



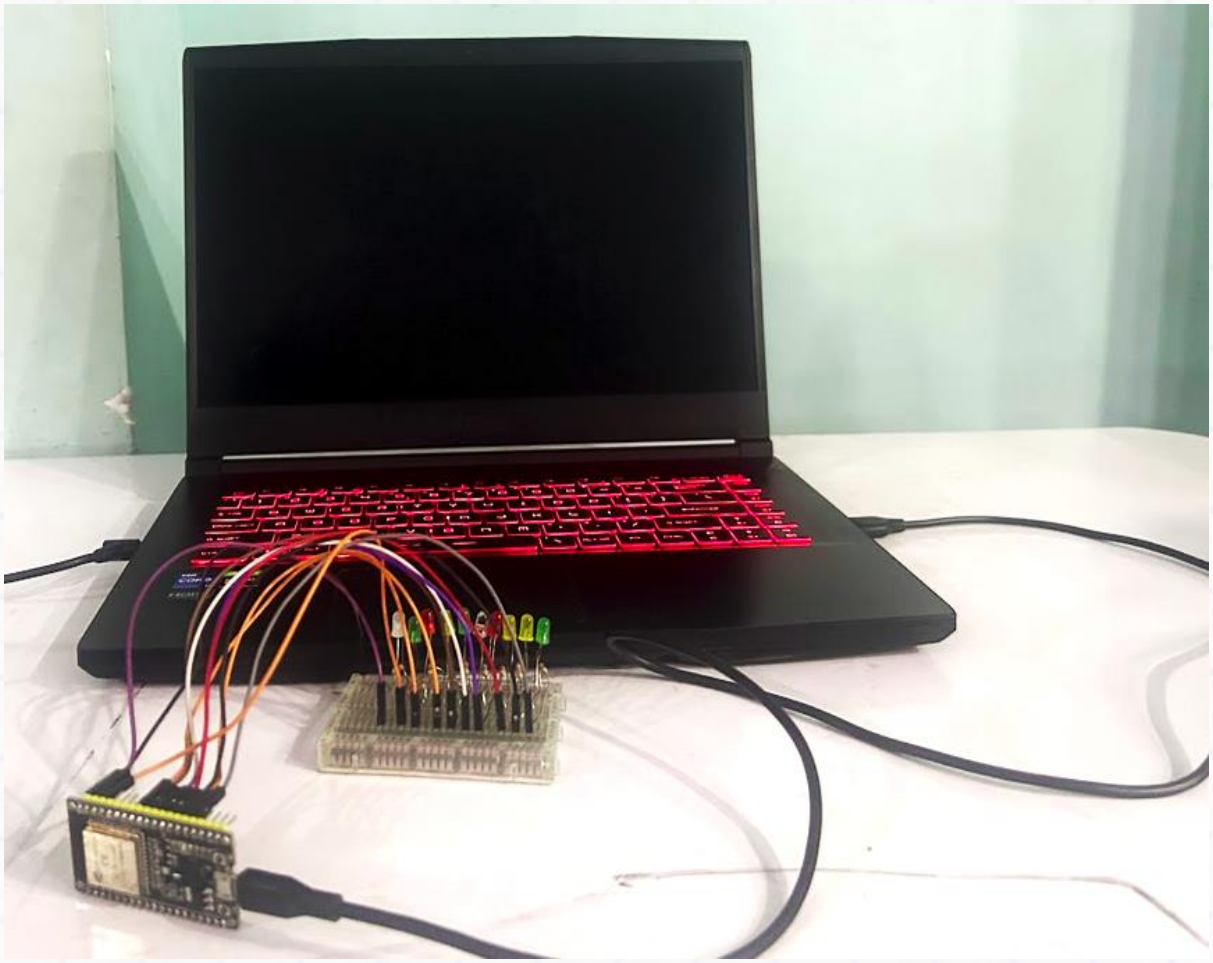
The shorter terminals of the LEDs in the breadboard are connected to the Ground terminal of the ESP32 using a jumper wire.

Step 4



The longer terminals of the LEDs with the resistors are connected to the GPIO pins of the ESP32 using more jumper wires.

Step 5



The complete circuit of LEDs and ESP32.

Step 6

The circuit is now complete, and you are ready to work on the software part

Software

Downloads & Installation

To download and install the Arduino IDE, follow the steps given in the ESP32 Base Document.

Launching the IDE

1. After installation, locate the Arduino IDE icon on your desktop (if you chose to create one during installation) or find it in your applications folder (on macOS) or start menu (on Windows).

2. Double-click the icon to launch the Arduino IDE.

Code

1. The code demonstrates how to control multiple LEDs connected to an Arduino board. The LEDs will blink in a sequence, each turning on and off for one second.

- These constants define the pins to which the LEDs are connected.

```
const int LED1 = 3;
const int LED2 = 4;
const int LED3 = 5;
const int LED4 = 6;
const int LED5 = 7;
const int LED6 = 8;
const int LED7 = 9;
const int LED8 = 10;
const int LED9 = 18;
const int LED10 = 19;
```

- The `setup()` function runs once when the microcontroller is powered on or reset. Here, it sets each LED pin as an output.

```
void setup() {
  // Initialize digital pins as outputs
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
```

```
  pinMode(LED5, OUTPUT);
  pinMode(LED6, OUTPUT);
  pinMode(LED7, OUTPUT);
  pinMode(LED8, OUTPUT);
  pinMode(LED9, OUTPUT);
  pinMode(LED10, OUTPUT);
}
```

- The `loop()` function runs repeatedly after the `setup()` function. Initially, it contains code to turn on LED1, wait for one second, turn it off, and wait for another second.

```
void loop() {
  digitalWrite(LED1, HIGH); // turn LED1 on
  delay(1000);              // wait for 1 second
  digitalWrite(LED1, LOW);  // turn LED1 off
  delay(1000);              // wait for 1 second
}
```

The sample code in the above loop() function shows how to blink LED1. To extend the functionality to control LEDs 2 through 10. For each LED, add similar code as shown for LED1, modifying the pin numbers accordingly.

Ensure each LED blinks in sequence, with each turning on for one second and off for one second before the next LED in the sequence blinks.

- Create a new sketch with name ***danceLED.ino***
- Copy and Paste the below code into your sketch and add the remaining LEDs code into the `loop()` function.

```
const int LED1 = 3;
const int LED2 = 4;
const int LED3 = 5;
const int LED4 = 6;
const int LED5 = 7;
const int LED6 = 8;
const int LED7 = 9;
const int LED8 = 10;
const int LED9 = 18;
const int LED10 = 19;


void setup() {
  // Initialize digital pins as outputs
```

```
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
  pinMode(LED5, OUTPUT);
  pinMode(LED6, OUTPUT);
  pinMode(LED7, OUTPUT);
  pinMode(LED8, OUTPUT);
  pinMode(LED9, OUTPUT);
  pinMode(LED10, OUTPUT);
}

void loop() {
  digitalWrite(LED1, HIGH); // turn LED1 on
  delay(1000);              // wait for 1 second
  digitalWrite(LED1, LOW);  // turn LED1 off
  delay(1000);              // wait for 1 second

//“YOUR REMAINING LEDs CODE HERE”

}
```

2. Save the Sketch:

    - Click on **File > Save As....**
    - Name the file "**dancing_led**" (without quotes).
    - Make sure the file extension is **.ino** (this is automatically handled by Arduino IDE).
    - Choose a location on your computer where you want to save the file.
    - Click Save.

Uploading the Code to ESP32

1. **Connect Arduino Board:**

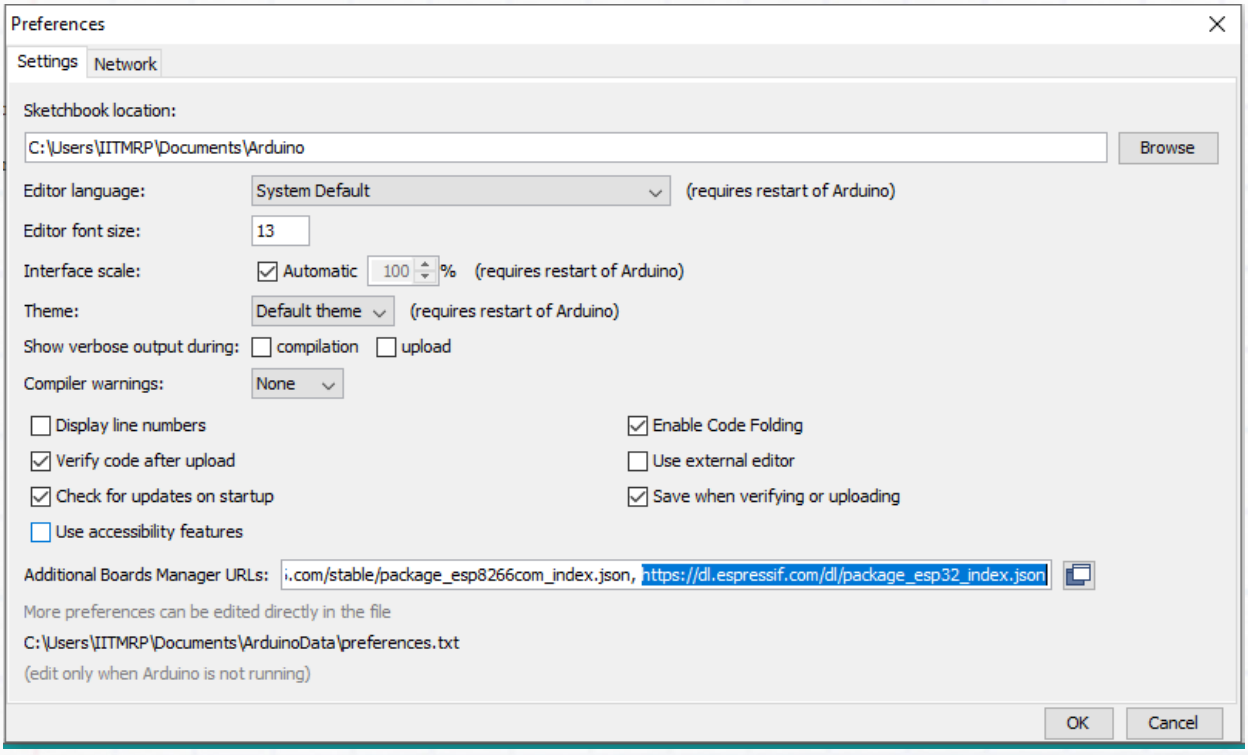
   Connect your Arduino board to your computer using a USB cable.

2. **Select Board and Port:**

   - Open the Arduino IDE if it's not already open.
   - Go to **File** > **Preferences**. In '**Additional Boards Manager URLs**', if the textbox is empty, paste the following link:

     https://dl.espressif.com/dl/package_esp32_index.json

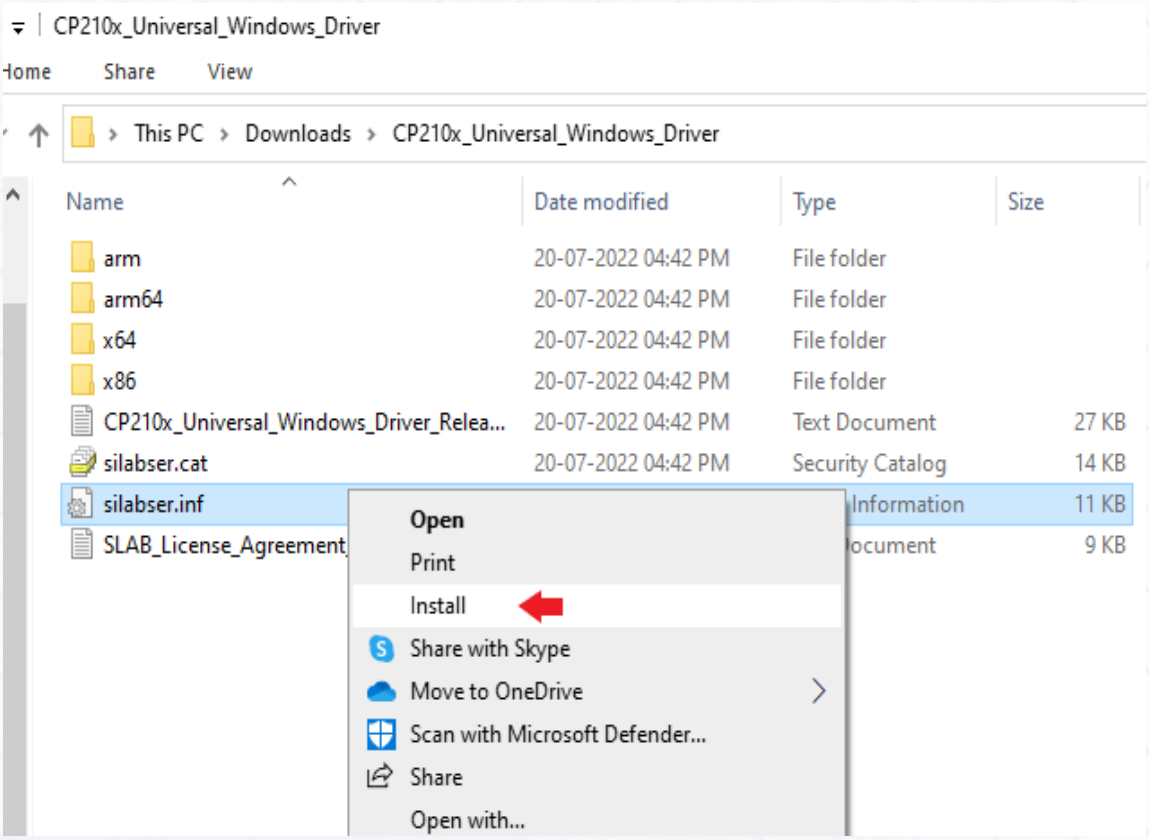
   If the textbox already has some other links, put a comma after them and paste the above link.



- Next, go to **Tools** > **Board** > **Boards Manager**. In the search bar on the top, search ‘esp32’ (version 2.0.11) and install the package. This will take some time to install.

3. **Select Port:**

- From this link, download the **CP210x Driver** for your OS (eg. CP210x Universal Windows Driver). Extract the ZIP folder. Right-click the **silabser.inf** file inside and select install. Follow the prompts until installation is successful. Restart your PC once installation completes.



- Go to **Tools > Port** and select the port to which your Arduino board is connected. The port will typically show as **COMx** on Windows, **/dev/cu.usbmodemxxxx** or **/dev/tty.usbmodemxxxx** on macOS, or **/dev/ttyUSBx** or **/dev/ttyACMx** on Linux.

4. **Verify and Compile Sketch:**

    - Click on the verify button (checkmark icon) or go to **Sketch > Verify/Compile** to compile your sketch. This step ensures that there are no syntax errors in your code.

5. **Upload Sketch:**

    - Once the sketch compiles successfully, click on the upload button (right arrow icon) or go to **Sketch > Upload**. This action will compile the sketch again (if necessary) and upload it to your Arduino board.

6. **Monitor Upload Progress:**

    - The status bar at the bottom of the Arduino IDE will show the progress of the upload process. During this time, you might see the onboard LED on your Arduino board blink rapidly, indicating that the sketch is being uploaded.

7. **Upload Completion:**

    - Once the upload is complete, the status bar will display "**Done uploading**".
    - If there are any errors during the upload process, carefully read the error messages in the output window at the bottom of the Arduino IDE. Common issues include incorrect board selection, missing drivers, or incorrect port selection.

8. **Verify Operation:**

    After uploading, your Arduino board should start executing the "dancing LED" sequence as per the code you wrote.

Tasks

1. Turn LED 1 ON
2. Turn LED 1 OFF
3. Make LED 1 blink with an interval of 500 ms.
4. Change the LED's frequency of blinking
5. Repeat the above for LEDs 1 and 2 simultaneously
6. Repeat the above for all 10 LEDs simultaneously
7. Make LEDs dance! - Program a new creative pattern with the LEDs. Take a short video clip of it and share it with the Build Club Discord community!

Booyah! You have now learnt how to write codes for the ESP32 C-6 with the Arduino IDE