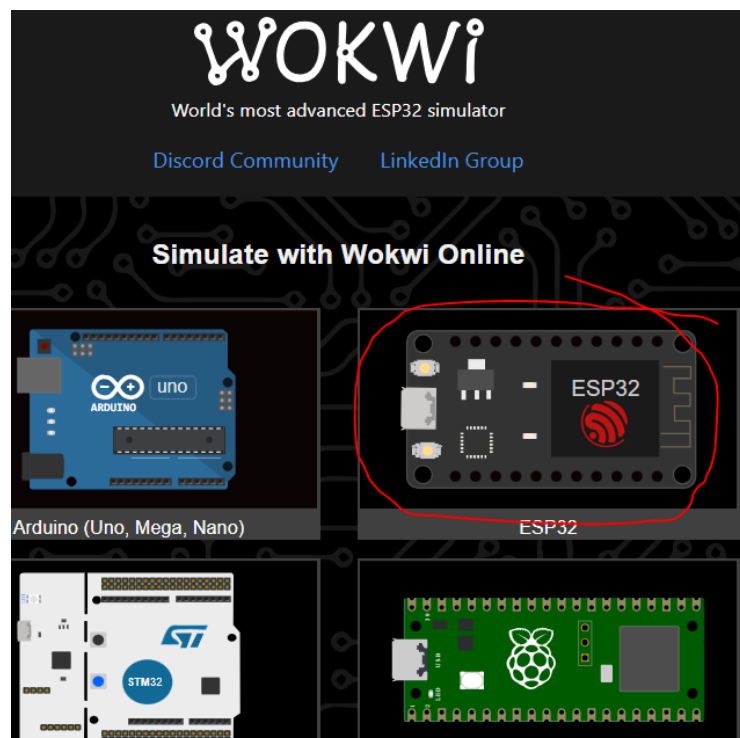


## Exp-2: Hand-on Practice of IoT Circuits and Systems on the Wokwi Simulator.

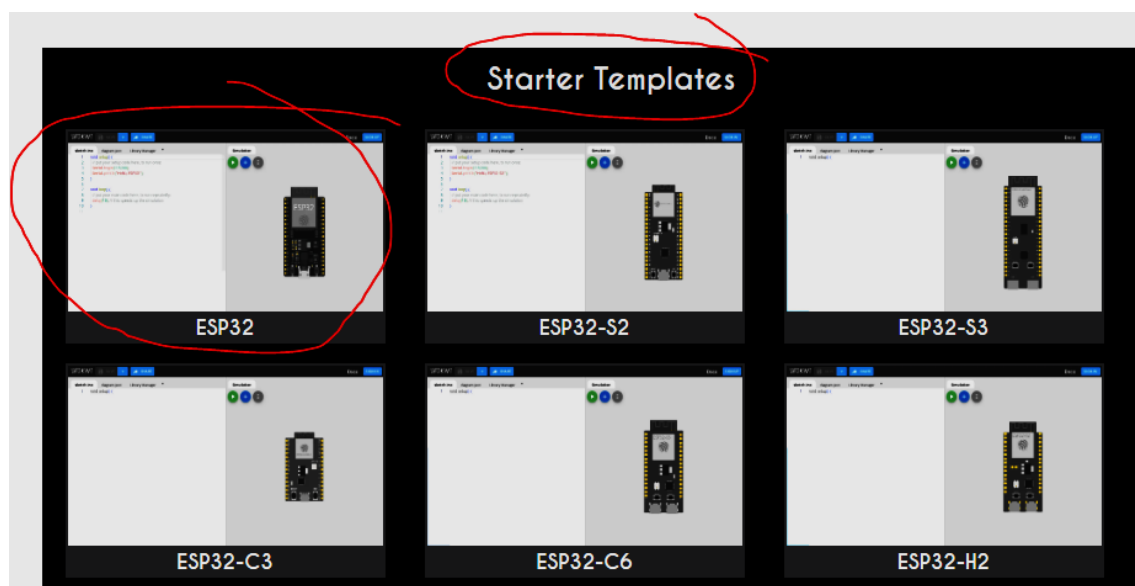
**Step-1:** Open the online simulation platform by using the link: <https://wokwi.com/>

On the top right corner, click on [sign-in/up](#) and [sign-in with Google](#) using your BMU email ID.

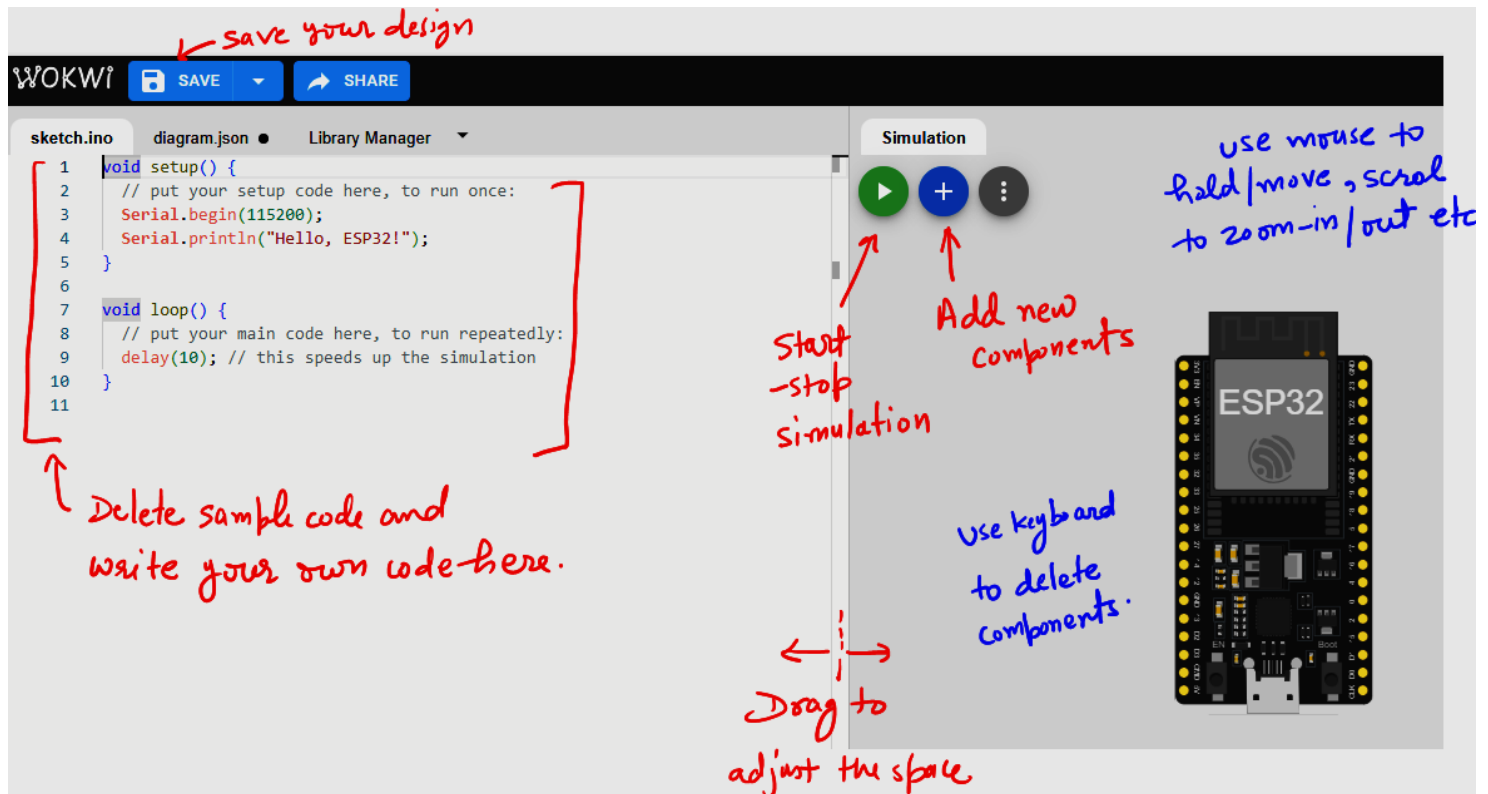
You will be directed to main page as shown in figure below. [On this page, select \(click\) the ESP32.](#)



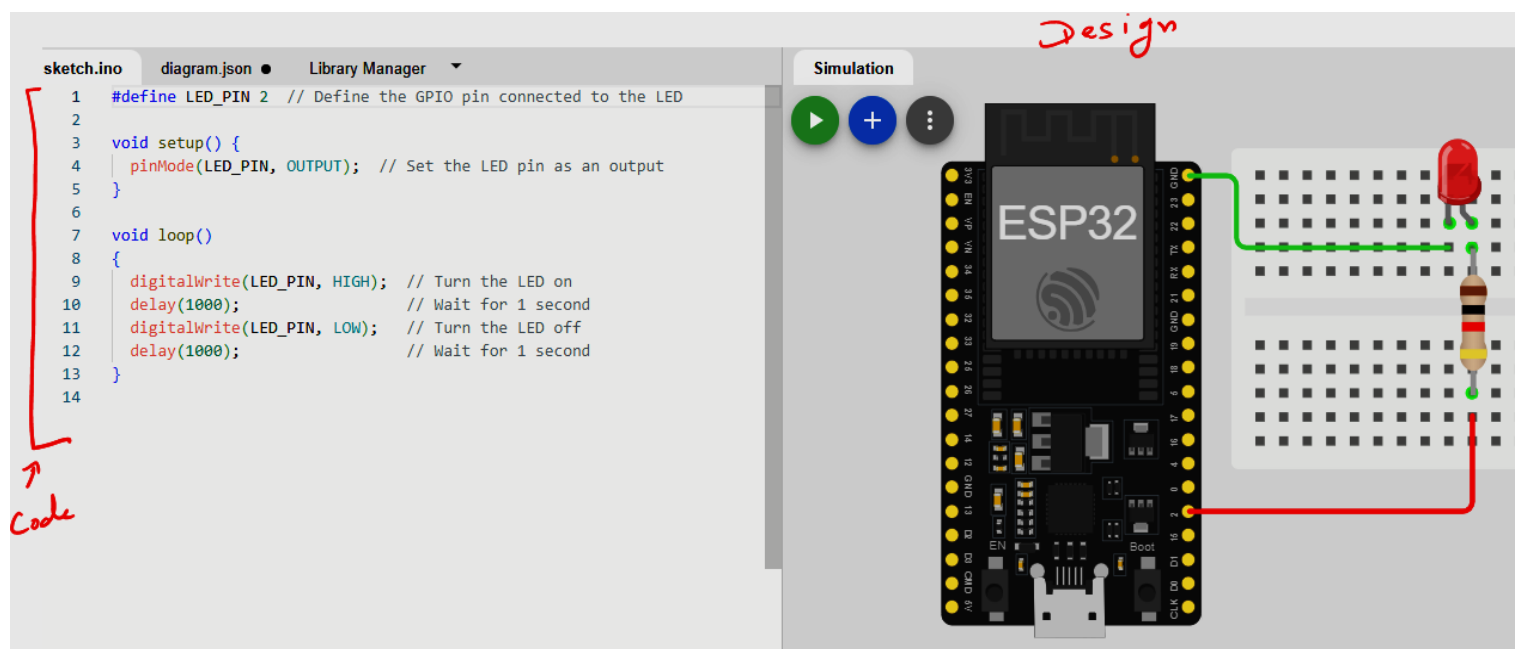
**Step-2** Subsequently, you will see several templates of using ESP32 including featured-projects and starter-templates etc. Under the [Starter Templates](#), [click on ESP32 template and open it](#) as shown in figure below:



**Step-3:** You will get a simulation work space similar to Tinkercad. Please get familiar with requisite functions as shown in figure below (READ THE COMMENTS/ANNOTATIONS CAREFULLY AND TRY THE SAME ON YOUR WINDOW):



**Step-4:** On the work space window, click on '+' button to add a breadboard, LED and resistor. Put LED (straight leg -ve, twisted leg +) and resistor on the breadboard and connect with the wires to the **pin-2 and ground of ESP-32**. The wires can be obtained by clicking on end-points of components or on the pins of ESP32. Click/select a component (or wire) to change its colour. **SEE THE FIGURE BELOW:**



Now click on simulation button and notice the LED; it must blink with the given delay of 1 sec. [delay(1000) = 1 sec].

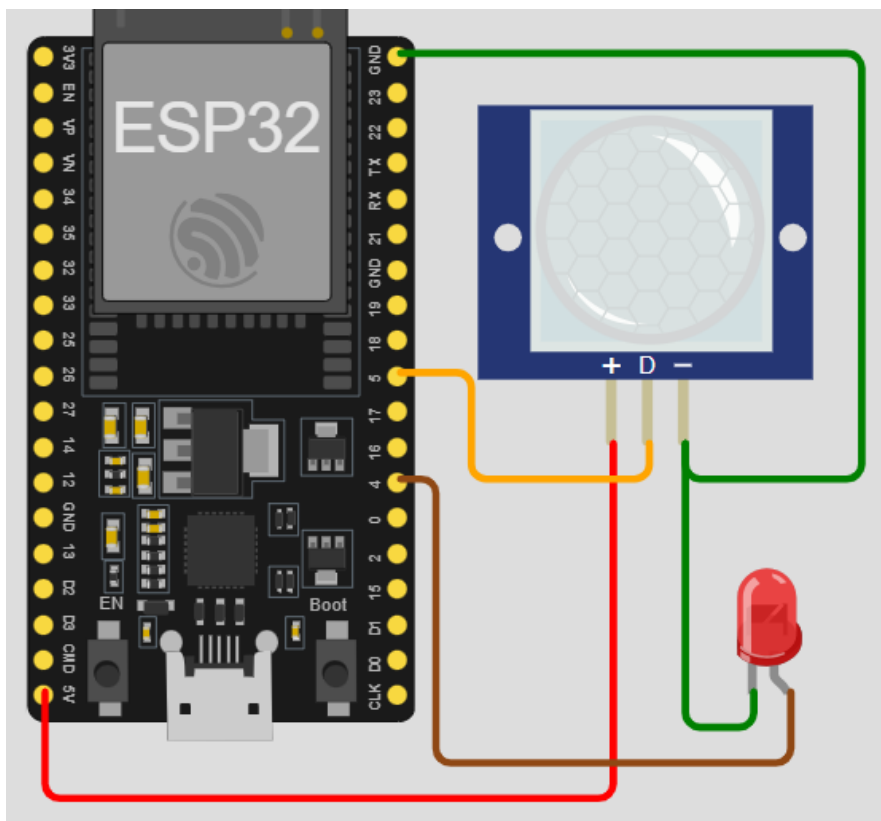


Subsequently SAVE your design by using the save button (tab) on the top left corner. UPTO THIS POINTS, YOU WOULD GET FAMILIAR WITH THE WORKING ON Wokwi SIMALATION PLATFORM.

**Step-5:** After finishing practice till above step, click on **WOKWI** icon on the top-left corner of the window and return to the home screen. To simulate another design, click on ESP32 Icon, and select the ESP32 starter template again as did in step-1 and 2.

**PRACTICE EXAMPLE-1:** Using ESP32, PIR sensor and an LED, design an automatic lighting system such that the light (LED) turns ON if motion is detected and vice-versa.

**Circuit diagram:**

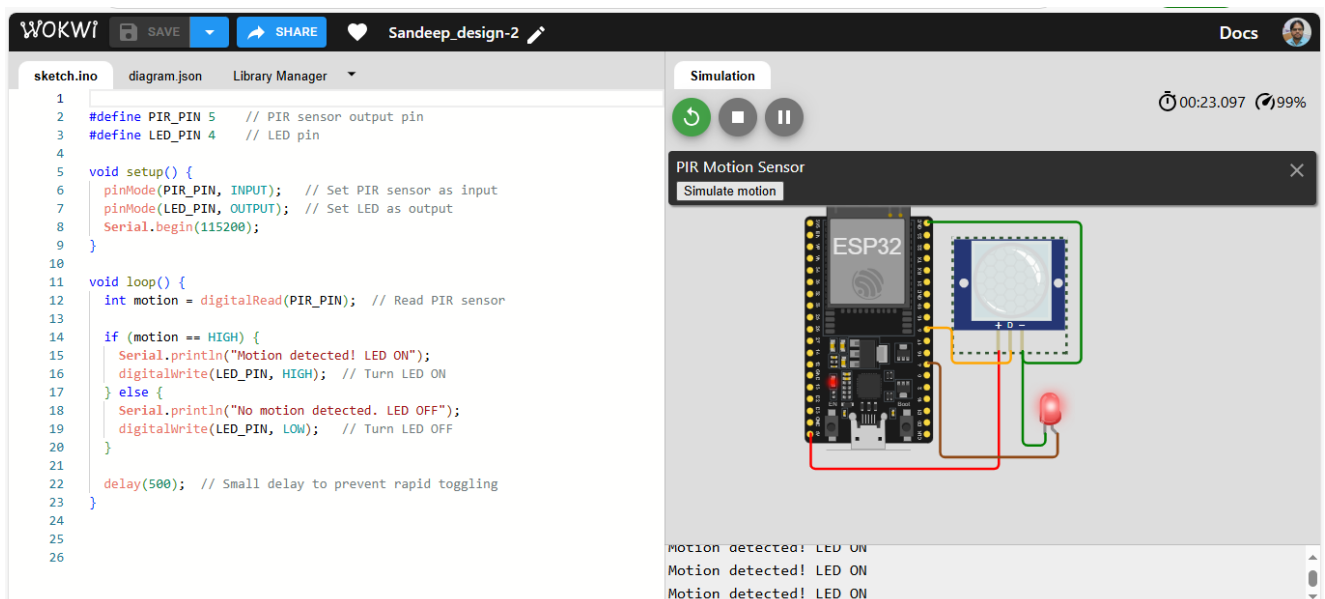


Code:

```
sketch.ino  diagram.json  Library Manager  ▼

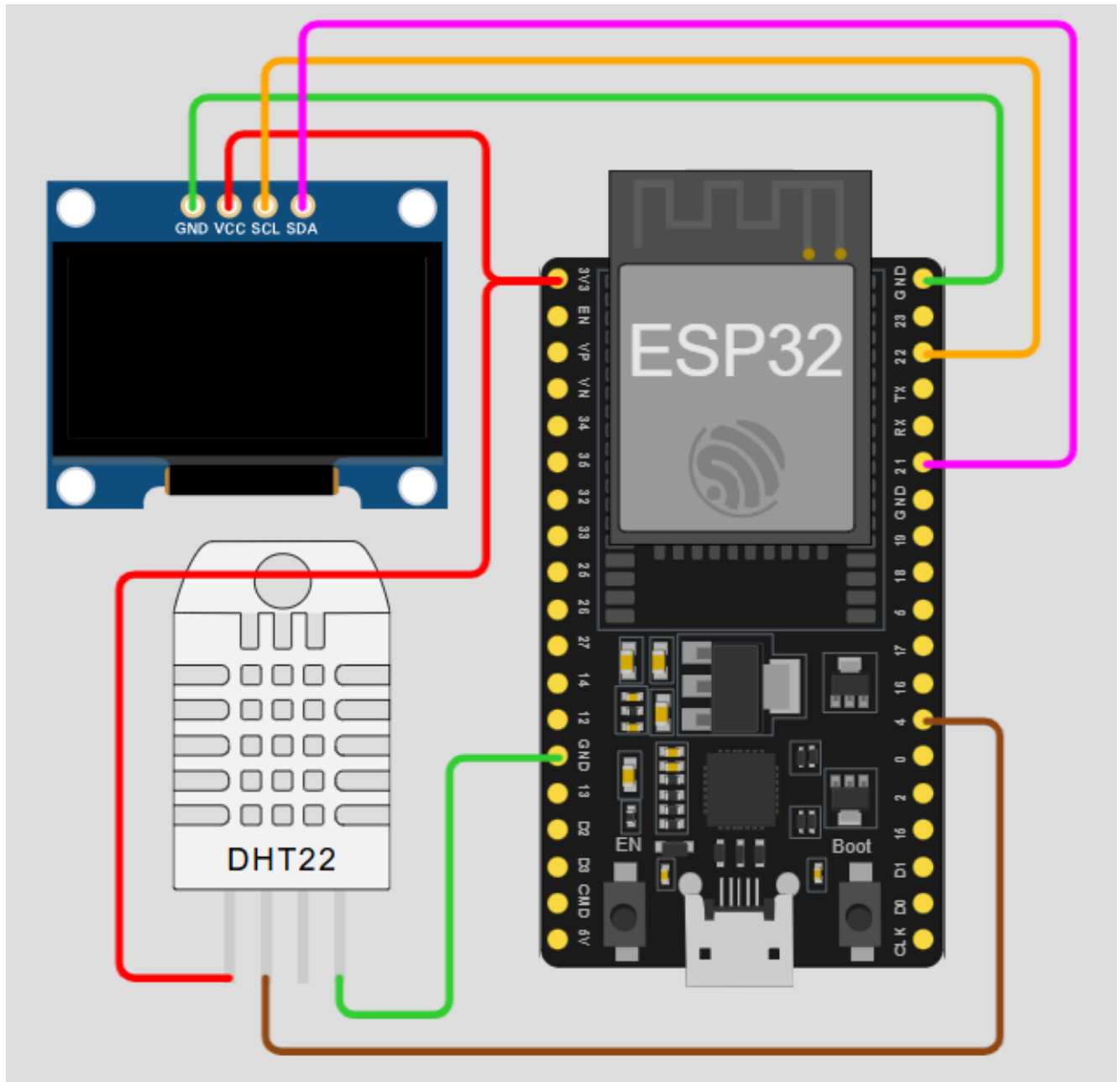
1
2  #define PIR_PIN 5    // PIR sensor output pin
3  #define LED_PIN 4    // LED pin
4
5  void setup() {
6      pinMode(PIR_PIN, INPUT); // Set PIR sensor as input
7      pinMode(LED_PIN, OUTPUT); // Set LED as output
8      Serial.begin(115200);
9  }
10
11 void loop() {
12     int motion = digitalRead(PIR_PIN); // Read PIR sensor
13
14     if (motion == HIGH) {
15         Serial.println("Motion detected! LED ON");
16         digitalWrite(LED_PIN, HIGH); // Turn LED ON
17     } else {
18         Serial.println("No motion detected. LED OFF");
19         digitalWrite(LED_PIN, LOW); // Turn LED OFF
20     }
21
22     delay(500); // Small delay to prevent rapid toggling
23 }
24
```

*Simulate the design and click on the PIR sensor to create the motion instance (simulate motion) as shown in figure below.*



**Practice Example-2:** Using ESP32, DHT22 sensor and an OLED (SSD1306), design a system to display the sensed temperature and humidity data on the OLED screen.

**Circuit Diagram:**



**Code** [Copy-paste line by line]:

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <DHT.h>

#define SCREEN_WIDTH 128 // OLED display width
#define SCREEN_HEIGHT 64 // OLED display height
#define OLED_RESET -1 // Reset pin not needed for SSD1306
#define OLED_ADDRESS 0x3C // I2C address for OLED display

#define DHTPIN 4 // GPIO pin connected to DHT22
#define DHTTYPE DHT22 // Define the sensor type

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(115200);
  dht.begin();

  if (!display.begin(SSD1306_SWITCHCAPVCC, OLED_ADDRESS)) {
    Serial.println("SSD1306 allocation failed");
    for (;;);
  }

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(10, 10);
  display.println("Initializing...");
  display.display();
  delay(2000);
}

void loop() {
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

  if (isnan(temperature) || isnan(humidity)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.print(" °C, Humidity: ");
  Serial.print(humidity);
```

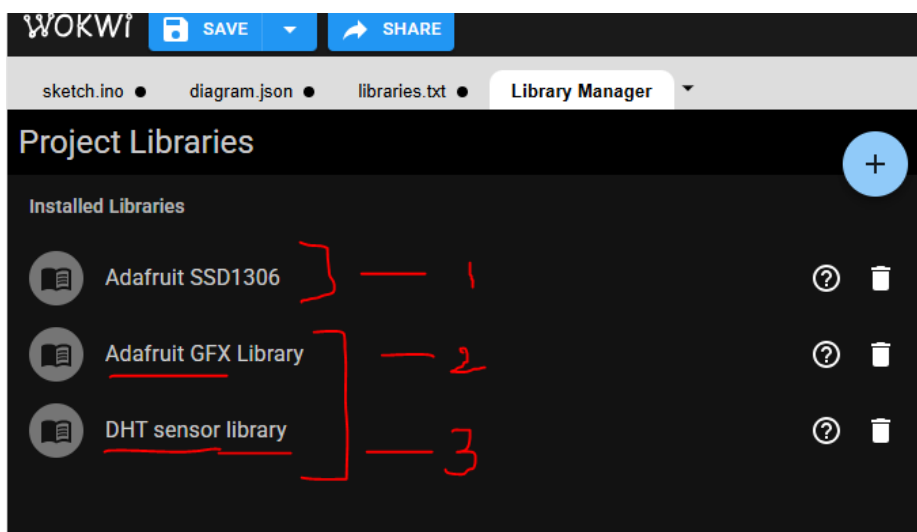
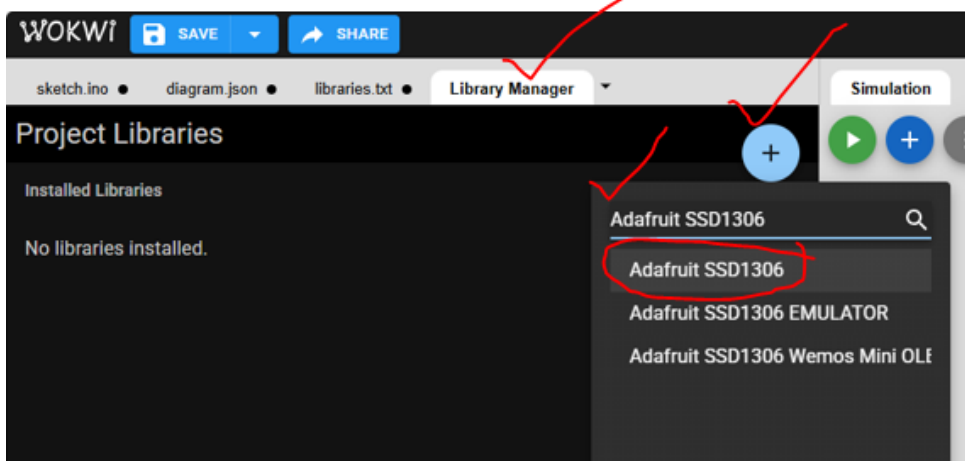
```
Serial.println(" %");

display.clearDisplay();
display.setTextSize(2);
display.setCursor(10, 10);
display.print("Temp: ");
display.print(temperature);
display.print(" C");

display.setCursor(10, 35);
display.print("Hum: ");
display.print(humidity);
display.print(" %");

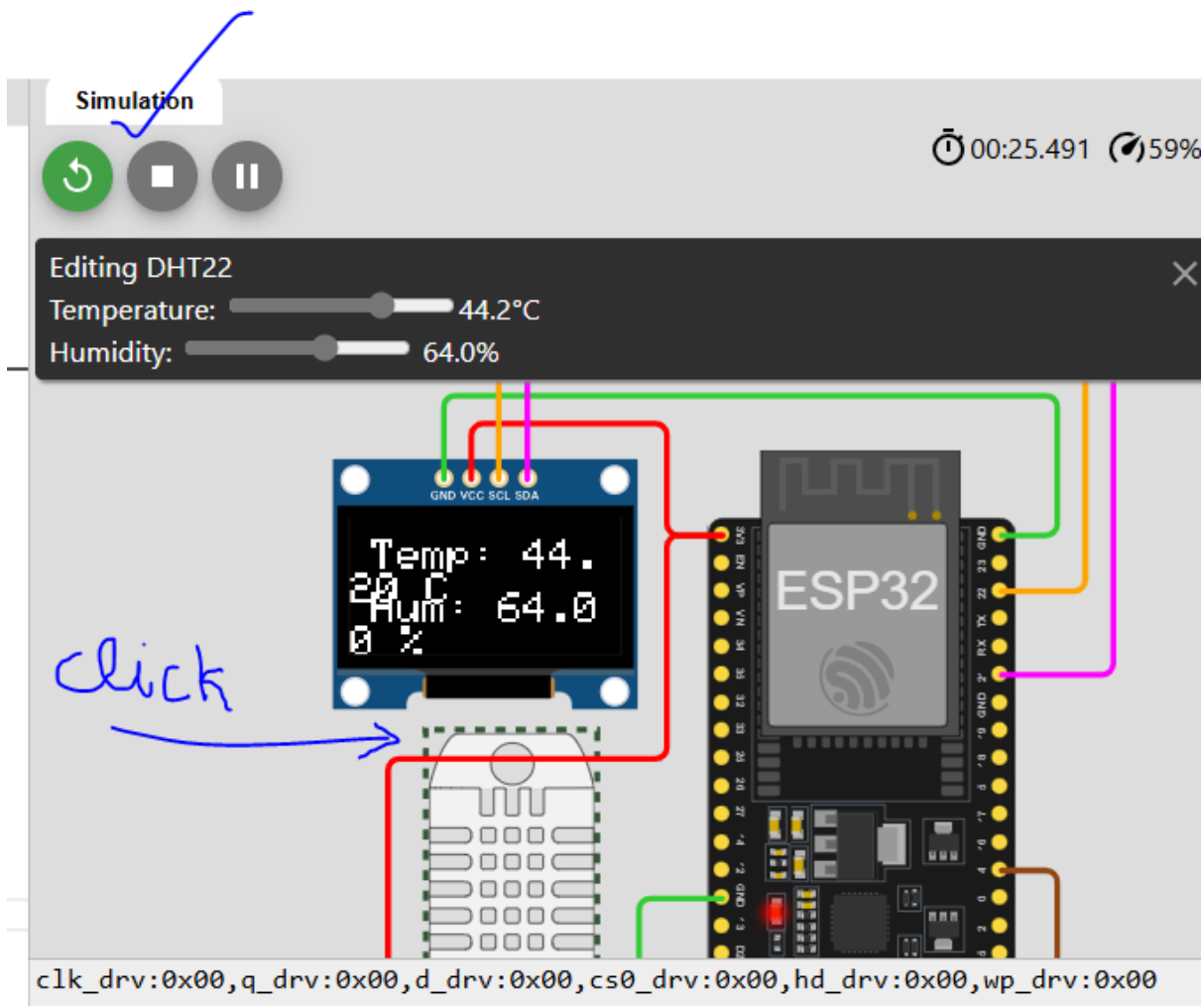
display.display();
delay(2000); // Update every 2 seconds
}
```

**Libraries to be installed:** Library manager → Click on + sign → add libraries as shown in two snaps below.



**After adding the above THREE libraries,** click on 'sketch.ino' to return on coding/design screen.

**Finally simulate the design** to see the readings of DHT22 sensor on OLED screen. You **can click on DHT sensor to change the sensed data** and the same will be displayed on OLED. [See snap below].



### Exercise Problem (Do it yourself):

Design an automatic door opening/closing system by using ESP32, ultrasonic sensor and a servo motor. [As a person comes in sensing range → Open the door; as the person moves away from sensor → close the door].