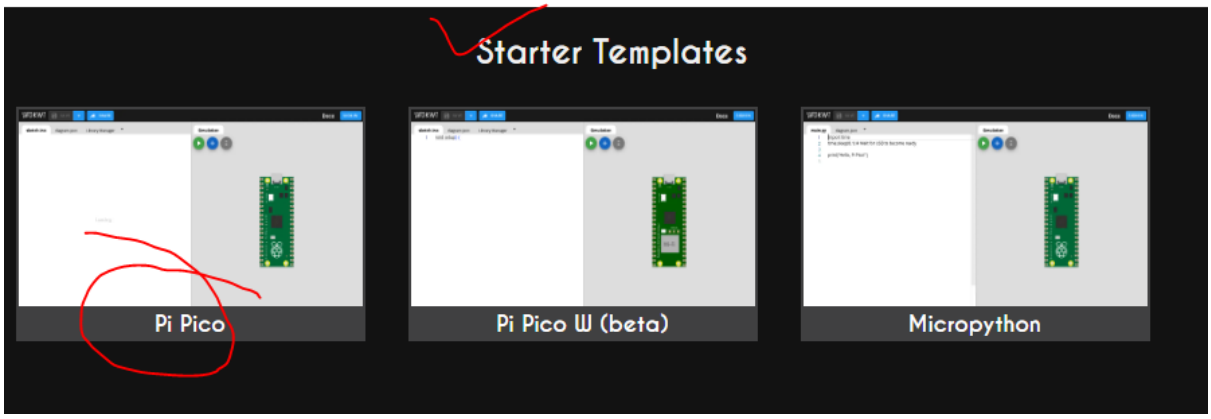
Exp-5: Implementing Raspberry Pi based IoT Edge Computing on Wokwi Simulator.

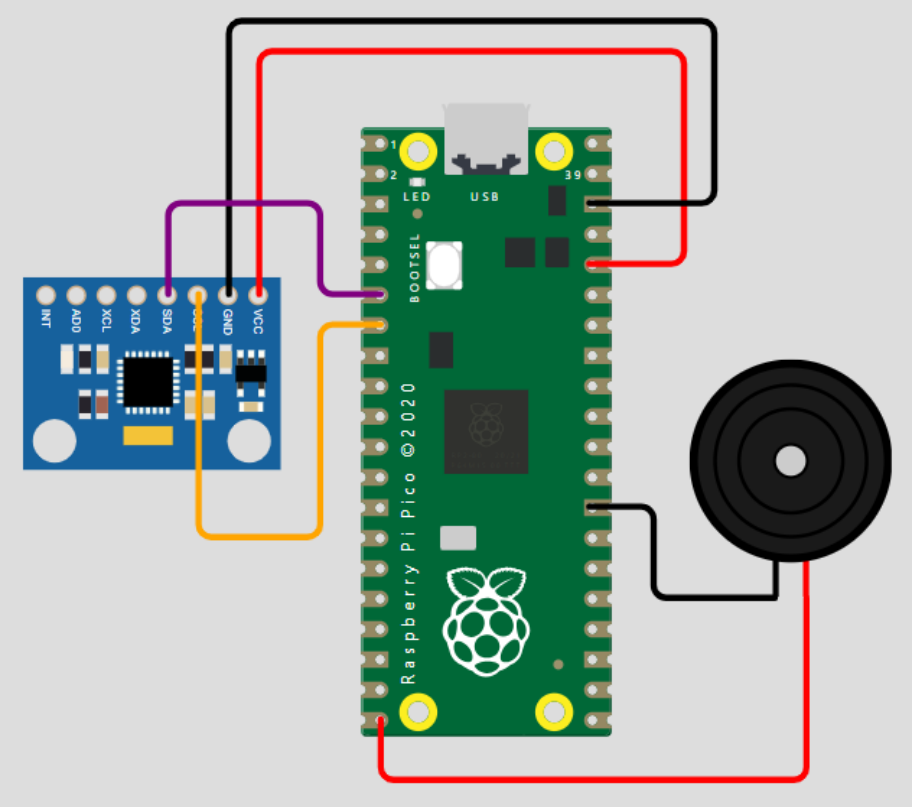
Step-1 Login to Wokwi Simulator (*https://wokwi.com/*) using the BMU email ID.

Step-2: Select the *'Pi Pico'* microcontroller and use *'Pi Pico'* starter template as shown in figure below:



Step-3: Practice Example-1 – Designing a fall detection system using Raspberry Pi, MPU6050 sensor (accelerometer) and a buzzer.

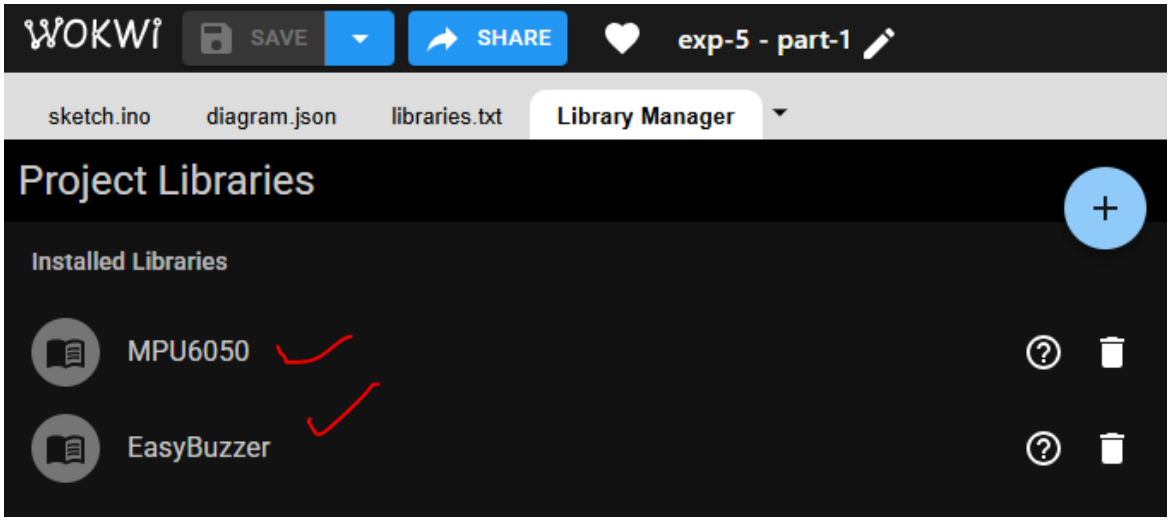
- **Design the following circuit using Raspberry Pi Pico, MPU6050 and a buzzer.**



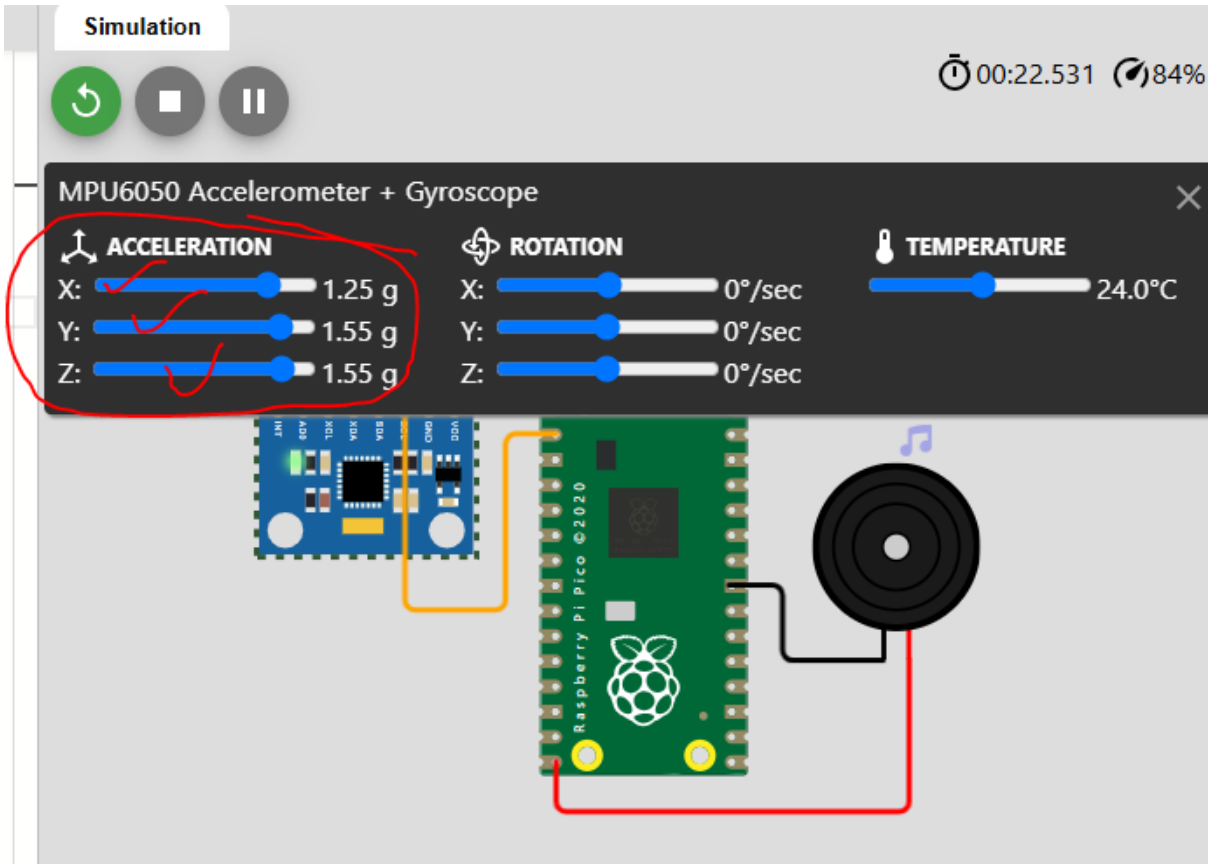
⇨ **3.1 - Write the following code:**

```
#include <Wire.h>
const int MPU = 0x68; // I2C address of the MPU6050
const int buzzer = 15; // Buzzer connected to GP15 on Pico

void setup()
{
  Wire.begin();
  pinMode(buzzer, OUTPUT);
  // Initialize MPU6050
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);

  Serial.begin(9600);
}

void loop()
{
  Wire.beginTransmission(MPU);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU, 6, true);
  int16_t AcX = Wire.read() << 8 | Wire.read();
  int16_t AcY = Wire.read() << 8 | Wire.read();
  int16_t AcZ = Wire.read() << 8 | Wire.read();
  float ax = AcX / 16384.0;
  float ay = AcY / 16384.0;
  float az = AcZ / 16384.0;
  float total_accel = sqrt(ax * ax + ay * ay + az * az);

  if (total_accel > 2.5)
  {
    Serial.println("Fall detected!");
    tone(buzzer, 1000); // 1000 Hz frequency
    delay(1000);
    noTone(buzzer); // Stop the tone
  }
  delay(500);
}
```

⇨ **3.2 -** Include the following **libraries** (see in the snap below)

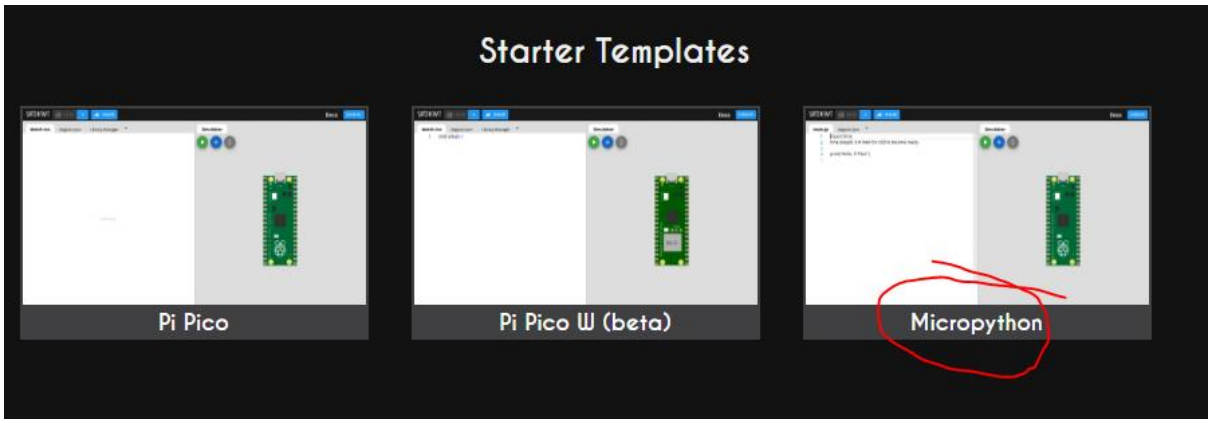


⇨ **3.3 - Save** your design and **Simulate** it. [It may take a few minutes. If fails repeatedly, refresh the browser.] Click on MPU6050 sensor and change the values of x, y, z accelerations. The buzzer must be sounded up as the values cross the threshold set in the program.

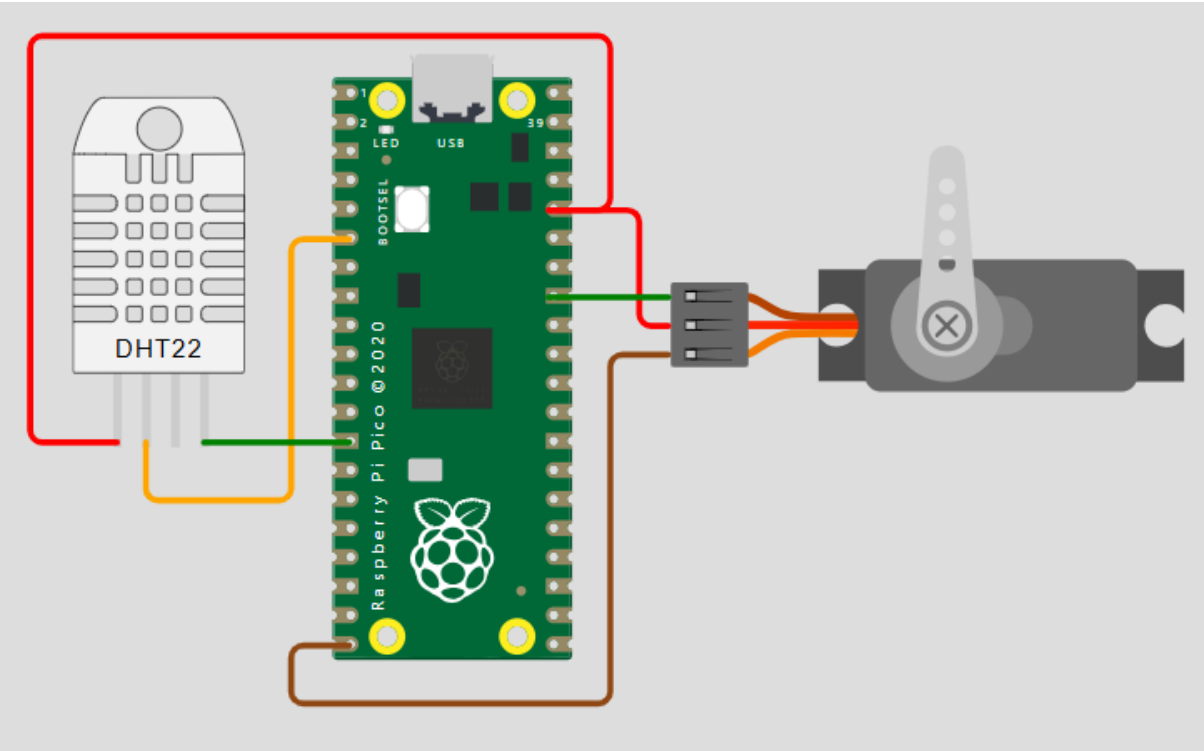


Step-4. Practice Example-2: Designing a fire detection system using Raspberry Pi, DHT22 sensor and a servo motor (to open emergency exit). ***[Coding in Python].***

4.1 – Select the *'Pi Pico'* microcontroller and use *'Micropython'* starter template as shown in figure below:



⇨ **4.2:** Design the following circuit using Raspberry Pi, DHT22 and a servo motor.



4.3: Write the following code.

```
from machine import Pin, PWM
from dht import DHT22
import time

# Pin configuration
dht22_pin = Pin(4)
servo_pin = PWM(Pin(15))

# DHT22 setup
dht22 = DHT22(dht22_pin)

# Servo setup
servo_pin.freq(50)

# Function to move servo
# angle: 0 (closed) to 90 (open)
def set_servo_angle(angle):
    duty = int(2000 + (angle / 180) * 6000)  # Adjusted duty cycle for
accurate servo control
    servo_pin.duty_u16(duty)

# Main loop
while True:
    try:
        dht22.measure()
        temp = dht22.temperature()
        print(f"Temperature: {temp}°C")

        if temp > 50:  # Fire detection threshold
            print("Fire detected! Opening emergency exit.")
            set_servo_angle(90)   # Open the servo (emergency exit)
        else:
            print("Temperature normal. Closing emergency exit.")
            set_servo_angle(0)    # Close the servo

    except Exception as e:
        print("Error reading sensor:", e)

    time.sleep(2)
```

4.4: Save the design and simulate it. [It may take a few minutes. If fails repeatedly, refresh the browser.] Click on DHT22 sensor and change the temperature value. The emergency door (servo motor) must be opened as the temperature rises above 50 C and closed as it drops below 50 C.