A logo with black text

Description automatically generated

**Faculty of Science and Technology**

CST2590

# Internet of Thing

# (IoT)

**Student Name:** Syed Muhammad Haider Razvi **Student ID:** M00832169

**Lecturer:** Sir Nivede Issur

**Submission:** 23rd February 2024

**Module:** Internet of things cst2590

Groups: Random Groups from Week 7 till Week 9. With Feauvre, Byron, Raees, Keval.

Week 13 till Week 16: Haider Razvi, Raees, Keval, Madani, Gautam

# Week-1

**Research-Based Activity-1**

1. **What is the Internet of Things? Give 5 real-world examples of IoT.**

The concept of the Internet of Things (IoT) encompasses a network of tangible items, referred to as "things," which are integrated with sensors, software, and other technological components. These elements enable these objects to connect and exchange information with other devices and systems through the internet **(Oracle, 2023).**

Five real world examples for Internet of things include **Smart driving** in which the technology is on its top-notch level in electric cars. It can detect mechanical problems, there are driverless cars, etc. Secondly, **Smart wearables** such as **smart watches** would be another example for Internet of things. It can count the calories you have burnt, the footsteps a person has walked, and can receive text messages and calls from it **(Bevis by CBT Nuggets, 2023).**

Thirdly, there is **smart home security**, where the motion sensors work and detect anything which are out or inside the boundary. Moreover, **Smart living** for instance, heating and cooling sensors can be controlled through a smart phone. You can monitor the units you have used, turn it on or off from a smart phone. In last, **smart toll collection**, where if the sticker is on the front window of the car, it has a barcode and a registered car number, which the machine would automatically detect and let you through the toll **(Bevis by CBT Nuggets, 2023).**

1. **What are the different components of IoT?**

Devices,

Components,

Cloud,

Gateway,

Analytics,

User interface

**(Rajiv by RF Page, 2023).**

1. **What are the advantages of the Internet of Things in our daily life?**

Improved quality of life, cost reduction, higher productivity, minimal chances of error, real time access to information, control, and automation from home **(Zubovich by Sumatosoft, 2023).**

1. **What are the challenges associated with the use of IoT?**

High internet dependency, Cyber threats, Job displacement risks, lack of international standardization **(Zubovich by Sumatosoft, 2023).**

1. **How is IoT useful in manufacturing?**

Increased efficiency, less chances of error, cost reductions, increased output, digital twins **(Mendoza by Hitachi solutions, 2023).**

1. **How is IoT used in smart agriculture?**

IoT-enabled robots have the capacity to acquire and analyze real-time data concerning crop growth and soil conditions. This information can be utilized to improve the management of irrigation, fertilization, and pest control, resulting in higher crop yields and reduced expenditures **(Gerlee by Freeway, 2023).**

1. **How will IoT affect the healthcare industry?**

IoT has the potential to link various devices and systems in a healthcare setting, facilitating improved coordination of patient care. Moreover, companies in the healthcare sector employing IoT can harness this technology to enhance cost-effectiveness by boosting operational efficiency and enhancing patient results **(Gupta by appinventiv, 2023).**

1. **How can we best use IoT for smart cities?**

The application of IoT in urban settings offers a wide array of possibilities for improving efficiency, such as streamlining traffic management, handling waste management, constructing smart buildings, managing parking more efficiently, optimizing water usage, and enhancing public transportation, etc. For example, Traffic can be managed by using smart traffic lights. The traffic lights should automatically adapt based on traffic load.

Furthermore, installing cameras on traffic lights would allow for better monitoring of the traffic situation and would reduce violations of traffic rules. The cameras can also help the police department to catch the criminals. IoT technology in waste bins with sensors can be utilised to detect when the bins are full and dispose of them accordingly.

Smart buildings that use IoT to link and regulate all operations such as lights, air conditioning, heating, security, and so on. This will cut building operating expenses while also increasing efficiency **(Geeks for Geeks, 2020).**

1. **What is the Industrial Internet of Things?**

The industrial internet of things (IIoT) utilizes sophisticated sensors, actuators, and comparable tools, like radio frequency identification (RFID) tags, to enhance and improve manufacturing and industrial operations **(Gillis by TechTarget, 2023).**

These devices are linked to share and analyse data. This makes things run more smoothly and consistently. It's like a special internet for machines, and it's employed in a variety of industries such as manufacturing, energy conservation, and oil and gas management. IIoT utilizes data provided by conventional equipment in industrial settings over time by utilizing sophisticated machinery and real-time analysis **(Gillis by TechTarget, 2023).**

When machines are linked together, companies can spot issues faster, saving time and money. This also helps them understand their business better. In manufacturing, IIoT can help with checking quality, being more eco-friendly, tracing where things come from, and making the whole production process better. In industries, IIoT is important for things like knowing when machines need fixing, giving better customer service, saving energy, and keeping track of things **(Gillis by TechTarget, 2023).**

1. **What is the future of IoT in general?**

It is estimated that by 2025, there will be more than 21 billion IoT devices, Cybercriminals will continue to use IoT devices to facilitate DDoS attacks, More cities will become "smart", Artificial intelligence will become a bigger thing, Routers will continue to become more secure and smarter, 5G Networks will continue to fuel IoT growth, Cars will become even smarter, 5G's arrival will also open the door to new privacy and security concerns, etc (**Norton, 2019).**

1. **In general, explain how an IoT system works.**

These devices make use of the Internet Protocol (IP), which is also used to identify computers on the Internet and allow them to communicate with one another. The goal of the Internet of Things is to create devices that can send real-time data on their own, increasing efficacy and speeding up the delivery of important data without relying solely on human intervention **(Kenton by Investopedia, 2022).**

1. **Explain the differences between M2M and IoT.**

The M2M concept entails the communication and autonomous functioning of two or more machines without human involvement. This model exhibits a level of intelligence.

Several significant applications that utilize M2M technology for service provision include Warehouse Management Systems (WMS), Supply Chain Management (SCM), Energy Harvesting such as oil and gas, Customer billing like smart meters, and Traffic control.

Whereas IoT represents a network of interconnected devices that can gather and exchange data via the Internet without human involvement. It facilitates the interaction of objects with their surroundings, thereby enabling automated decision-making. Various applications and services utilizing IoT technology include Smart Home systems, Connected automobiles, applications in Agriculture and Retail, the development of Smart cities, and advancements in Healthcare **(Bhardwaj by IPwithease, 2020).**

1. **What is the Internet of Everything (IoE)?**

The Internet of Everything (IoE) is a network that interconnects people, devices, data, and operations, enabling overall intelligence and enhanced awareness within the interconnected environment. IoE functions as a unified system that boosts the abilities of the involved elements and incorporates network intelligence to streamline decision-making and simplify data sharing **(Kanade by Spiceworks, 2022).**

1. **What is an Embedded system, Microcontroller and Single board computer?**

Microcontroller is a unique form of computer designed primarily for accessing and operating with other hardware. It possesses limited resources in terms of CPU and RAM, ensuring low power consumption and the capacity for real-time, detailed processing of connected hardware. Unlike a full-fledged operating system, a microcontroller is highly specialized for specific use cases **(Sebastian via Geek Culture, 2021).**

On the other hand, Single-board Computers, often referred to as systems-on-a-chip, are compact complete computers. They come equipped with contemporary standard consumer interfaces such as HDMI, USB, WiFi, and Bluetooth. These computers incorporate SD cards or small SSDs for data storage, and they feature multiple CPU cores and RAM **(Sebastian via Geek Culture, 2021).**

# Week-2

**Please carry out research and complete the following requirements.**

1. Find 7 difference between Microcontroller and Microprocessor.

|  |  |
| --- | --- |
| **Microcontroller** | **Microprocessor** |
| **Microcontrollers are more affordable than microprocessors because they are designed for specific applications and tend to have fewer external components.** | **Due to their greater processing capability and the requirement for additional parts like memory and peripheral chips, microprocessors can be more expensive.** |
| **Microcontrollers often have limited on-chip memory (both RAM and ROM)** | **Programmes and data are often stored in external memory components by microprocessors, which also have larger memory addressing capabilities.** |
| **Microcontrollers are excellent for embedded systems with limited space since they are frequently offered in smaller packages and are more compact in size.** | **Microprocessors tend to have larger packages and may require additional components, which can result in a larger overall footprint.** |
| **Microcontrollers are commonly used in battery-powered devices and other energy-efficient applications since they are designed for low power consumption.** | **microprocessors use more power and might not be the best choice for battery-powered applications.** |
| **Microcontrollers are usually less powerful in terms of processing capabilities compared to microprocessors.** | **Microprocessors are designed for high-performance computing tasks and have more processing power, making them suitable for running complex operating systems and applications.** |
| **Microcontrollers are commonly used in embedded systems, such as appliances, cars, and Internet of Things (IoT) devices. They are intended for specific tasks or applications.** | **Microprocessors are frequently used in general-purpose computing devices like smartphones, servers, and personal computers.** |
| **Microcontrollers typically include integrated peripherals like timers, analog-to-digital converters (ADCs), UARTs, PWM controllers, and GPIO pins.** | **On the other hand, microprocessors often do not have integrated peripherals to the same extent. They rely on external components or additional chips to provide peripheral functionality.** |

**(Williams, 2023).**

**(Geeks, 2023).**

1. Carry out research and find 3 best Microcontrollers for IoT development.

|  |  |
| --- | --- |
| Name of IoT Controllers | Main Features |
| **Arduino MKR Series** | **The MKR line of Arduino products, including the MKR1000, MKR WiFi 1010, and MKR NB 1500, are renowned for their ease of use. These boards include connectivity choices like Wi-Fi, Bluetooth, and cellular (NB-IoT) and are based on several microcontroller chips.** |
| **Raspberry Pi Pico** | **A microcontroller board called the Raspberry Pi Pico is based on the RP2040 chip. It's an effective and affordable choice for IoT projects. The General-Purpose Input/Output (GPIO) pinout is flexible, and MicroPython or C/C++ can be utilised for programming.** |
| **ESP8266 and ESP32 (NodeMCU)** | **Due to their integrated Wi-Fi and Bluetooth capabilities, the ESP8266 and ESP32 microcontrollers are frequently utilised in the construction of IoT devices. They are popular for tasks requiring wireless connectivity.** |

**(Prakash, 2023).  
(Fahad, 2020).**

1. Carry out research and find 3 best Single Board Computer (SBC) for IoT development.

|  |  |
| --- | --- |
| Name of SBC | Main Features |
| **Raspberry Pi** | **known for its versatility and affordability. It's widely used in various IoT applications and projects due to its robust community support, a wide range of compatible accessories, and multiple models with varying capabilities.** |
| **Arduino** | **Arduino is another well-known SBC that's often used for IoT development. It's particularly popular for its ease of use and extensive library of add-on modules (shields) that allow for easy customization and expansion. Arduino boards are known for their compatibility with various sensors and actuators, making them suitable for a wide range of IoT applications.** |
| **BeagleBone** | **BeagleBone is a powerful SBC that offers high-performance computing capabilities. It's often preferred for IoT development due to its ample connectivity options, which include Ethernet, USB, and HDMI. BeagleBone boards are also known for their support for various programming languages, making them suitable for complex IoT projects.** |

**(Nvidia, n.d.)**

1. Find 5 differences between Single Board Computer and IoT microcontrollers.

|  |  |
| --- | --- |
| Single Board Computer | Microcontrollers |
| **higher processing power** | **limited processing power, making them ideal for basic control and monitoring functions in IoT devices.** |
| **built-in connectivity options such as Wi-Fi, Bluetooth, and multiple USB ports** | **connectivity options, usually have more limited I/O ports and are optimized for interfacing with sensors, actuators, and other basic input/output components common in IoT** |
| **consume more power due to their higher processing capabilities and the need to support multiple peripherals** | **minimal power consumption, making them suitable for battery-powered or energy-efficient** |
| **Expensive** | **Affordable** |
| **advanced processing capabilities** | **simple and specific IoT applications that require basic control and monitoring functions** |

1. Lists 10 common sensors used for IoT development.

|  |  |
| --- | --- |
| Sensor’s Name | Functions |
| 1. **Temperature Sensor** | **commonly used in climate control, weather stations** |
| 1. **Humidity Sensor** | **Measures and monitors the moisture levels in the air or other substances, commonly used in HVAC systems, greenhouses, and weather stations.** |
| 1. **Light Sensor** | **Measures the intensity of light in the surrounding environment. It's used in applications such as smart lighting, energy conservation, and display devices.** |
| 1. **Motion Sensor** | **Detects movement in its surrounding area, often utilized in security systems, lighting control, and smart appliances.** |
| 1. **Infrared sensor** | **Detects infrared radiation to measure heat and temperature, commonly used in security systems, temperature measurement, and motion detection.** |
| 1. **Proximity sensor** | **Detects the presence or absence of objects nearby without physical contact, commonly used in touchscreens, robotics, and occupancy detection systems.** |
| 1. **Gas sensor** 2. **Pressure sensor-** | **Detects the presence of various gases in the environment, commonly used in air quality monitoring, industrial safety, and environmental monitoring.**  **- Detects and measures pressure variations, often used in applications like weather monitoring, industrial automation, and automotive systems.** |
| 1. **Accelerometer** | **Measures acceleration and tilt forces, often used in fitness tracking devices, gaming controllers, and orientation detection in smartphones.** |
| 1. **Gyroscope sensor** | **Measures angular velocity and orientation, commonly found in navigation systems, drones, and motion-controlled devices.** |

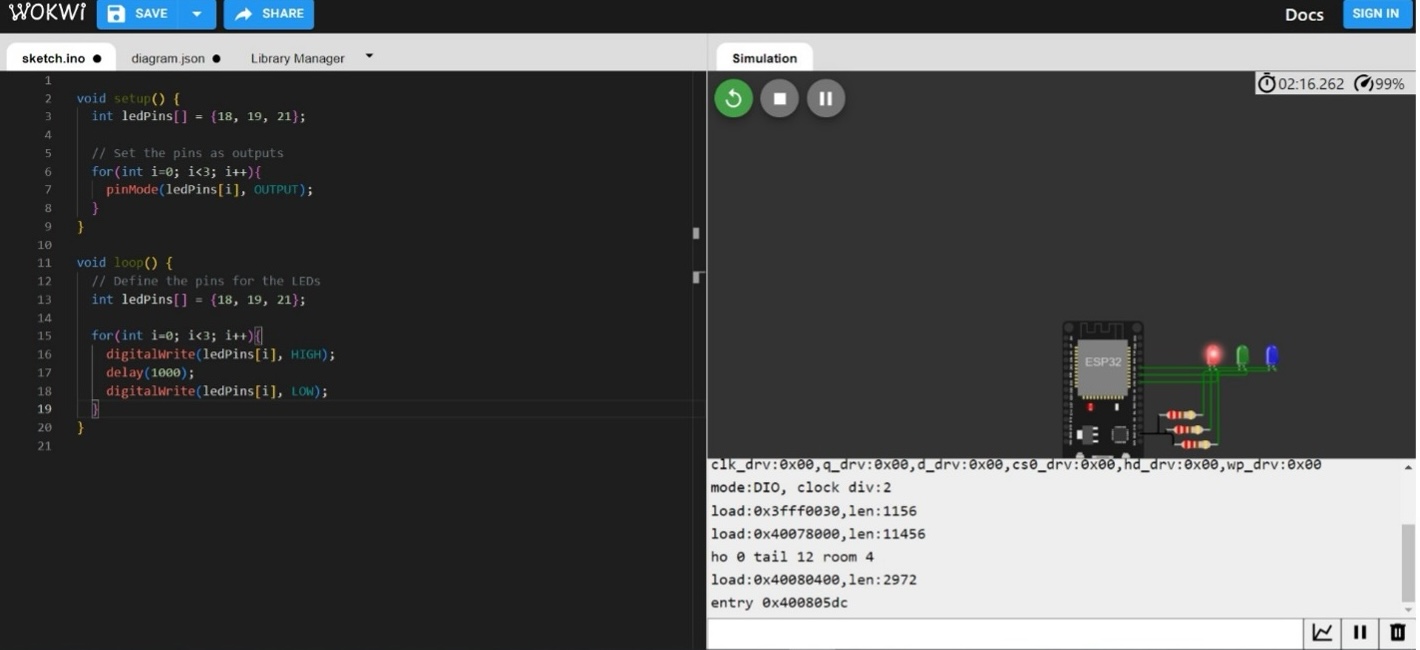
# 

# Week-3

**Task 1. Using 1 Arduino or Esp32/8266 board and 1 LED, program MCU to make the LED blink for every 1 sec.**

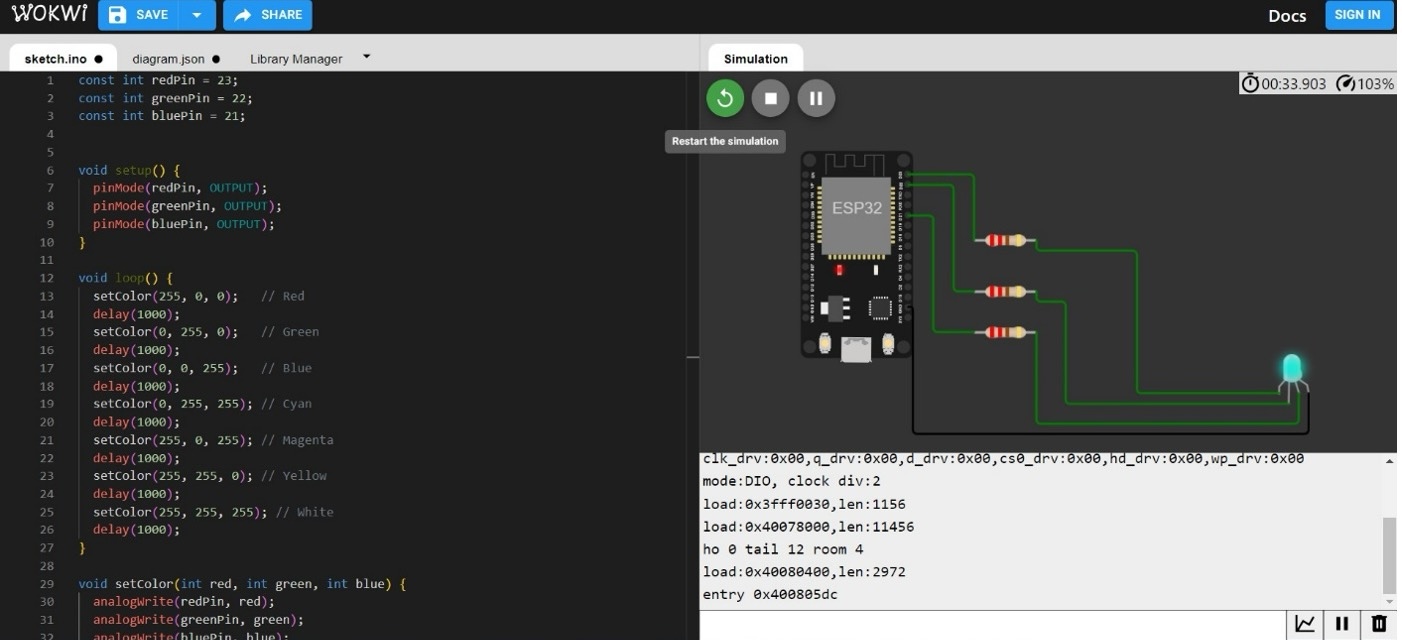
The program is called Wokwi and it is simulating an ESP32 microcontroller. The code on the left is written in Arduino and it is controlling an LED connected to pin 18 on the ESP32 board. The code sets the pin to output mode, then turns the LED on for 1 second, and then turns it off for 1 second. This creates a blinking effect.In essence, the program is simulating the blinking of an LED.

**Task 2. Using 1 Arduino or Esp32/8266 board and 3 LED of different colours, program MCU to make the 3 LED blink one after another or all together in a sequence for every 1 sec.**

****

This Wokwi simulation controls an LED. It repeatedly turns the LED on for 1 second, then off for 1 second, creating a blinking effect.

**Task 3. Using 1 Arduino or Esp32/8266 board and 1 RGB LED, program MCU to produce red, green, blue, cyan, magenta, yellow and white colours after every 1 sec.**

****

The first lines define three constant variables, redPin, greenPin, and bluePin, which are set to pins 23, 22, and 21, respectively. These pins are connected to the red, green, and blue LEDs.

The setup() function is called once when the program starts. It configures the three LED pins as outputs using the pinMode() function.

The loop() function is called repeatedly over and over again. It sets the color of the LED to red, green, blue, cyan, magenta, yellow, and white, using a delay of one second between each color change.

The code uses the setColor() function to change the color of the LED. This function takes three arguments: red, green, and blue, which correspond to the intensity of the red, green, and blue components of the LED's color. The analogWrite() function is used to set the intensity of each color component.

Week-4

# Week-5

**Task 1: Using 1 Arduino or Esp32/8266 board, 1 potenometer, 1LED program the MCU so that the brightness of the LED changes with the value of the Potenometer.**

A screenshot of a computer

Description automatically generated

#include "DHTesp.h"

const int DHT\_PIN = 15; // Input pin for DHT 22 sensor

DHTesp dhtSensor;

void setup() {

Serial.begin(9600);

dhtSensor.setup(DHT\_PIN, DHTesp::DHT22);

pinMode(33, OUTPUT); // Output pin for buzzer

}

void loop() {

TempAndHumidity data = dhtSensor.getTempAndHumidity();

Serial.println(String(data.temperature, 2) + "°C");

Serial.println(String(data.humidity, 1) + "%");

delay(500);

if (data.temperature > 50 || data.humidity > 75) {

digitalWrite(33, HIGH);

} else {

digitalWrite(33, LOW);

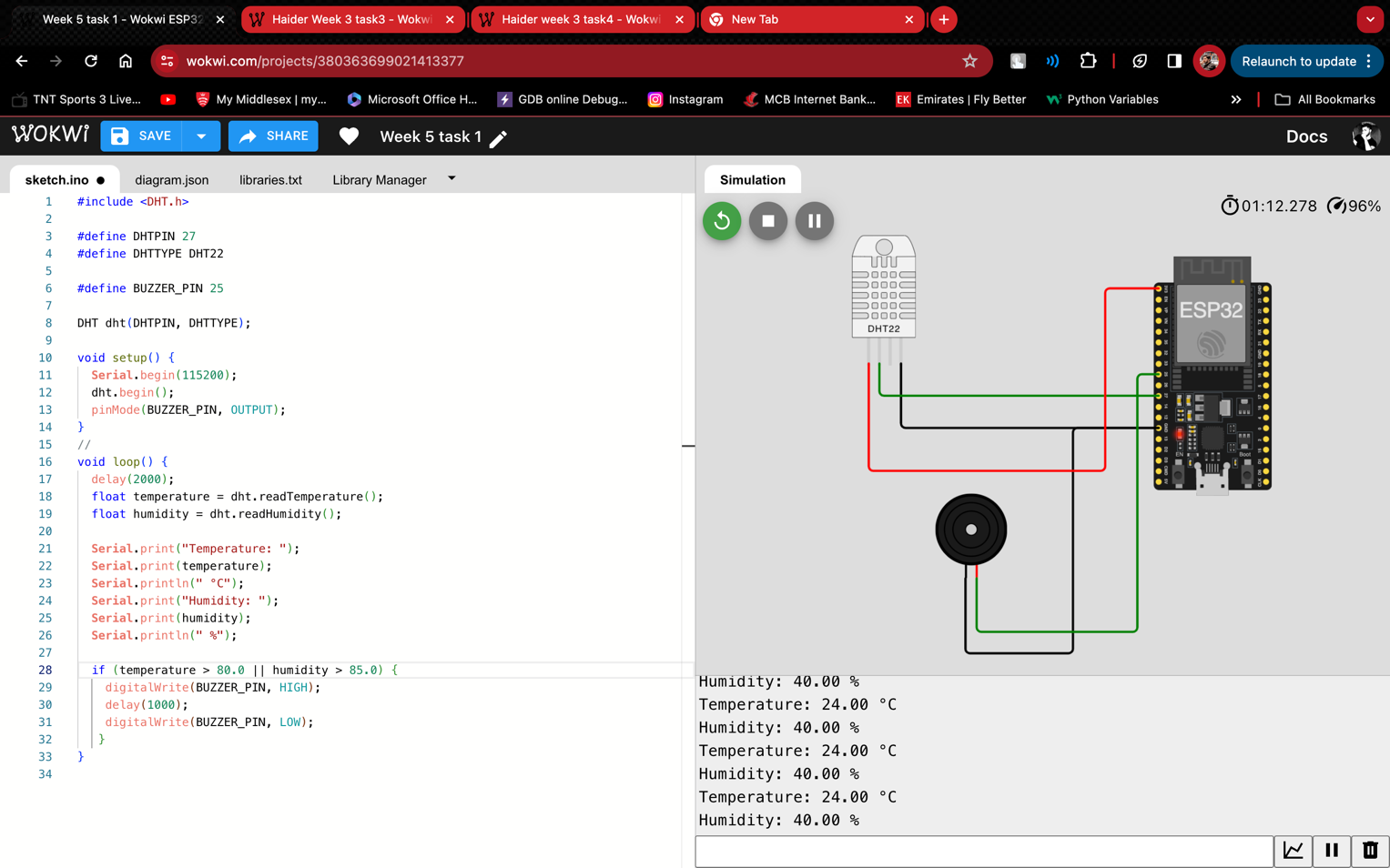
}

}

**The first part is for the potentiometer and LED. It sets up the serial communication, reads the value from the potentiometer, and writes the value to the LED.**

**The second part is for the DHT22 sensor. It includes the DHTesp library, sets up the sensor pin, and reads the temperature and humidity from the sensor. The code also prints the temperature and humidity values to the serial monitor and beeps the buzzer if the temperature is above 50 degrees Celsius or the humidity is above 75%.**

**Task 2: Using 1 Arduino or Esp32/8266 board, 1 DHT22 sensor, 1 buzzer, program the MCU to sound the buzzer as soon as either the temperature is above 50deg or humidity is above 75% (Hint: use the Serial Monitor to idenfy the best value to trigger the buzzer).**



The code includes libraries for DHT sensor and buzzer. The DHT sensor is used to measure temperature and humidity, and the buzzer is used to generate sound.

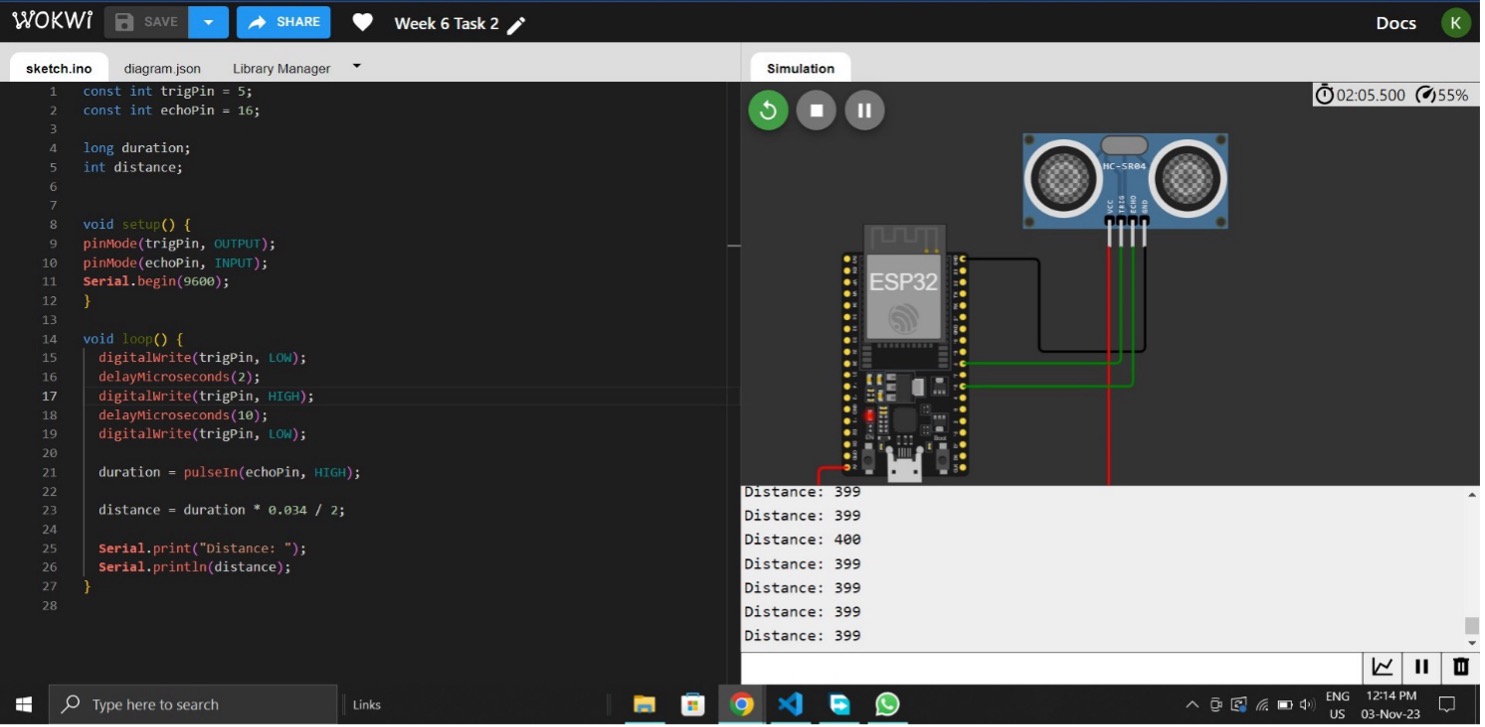
The code reads temperature and humidity values from the sensor and prints them to the serial monitor. If the temperature is above 80 degrees Celsius or the humidity is above 85%, the code will also turn on the buzzer for one second.

# Week-6

**Task 1. What is an Ultra Sonic Sensor?**

In simpler terms, an ultrasonic sensor acts like a super-powered bat. It sends out high-pitched sounds that humans can't hear, and listens for them to bounce back from objects. By measuring how long it takes for the sound to go out and come back, the sensor can figure out how far away the object is. It's like using sound as a ruler to measure distance (Jost, 2019).

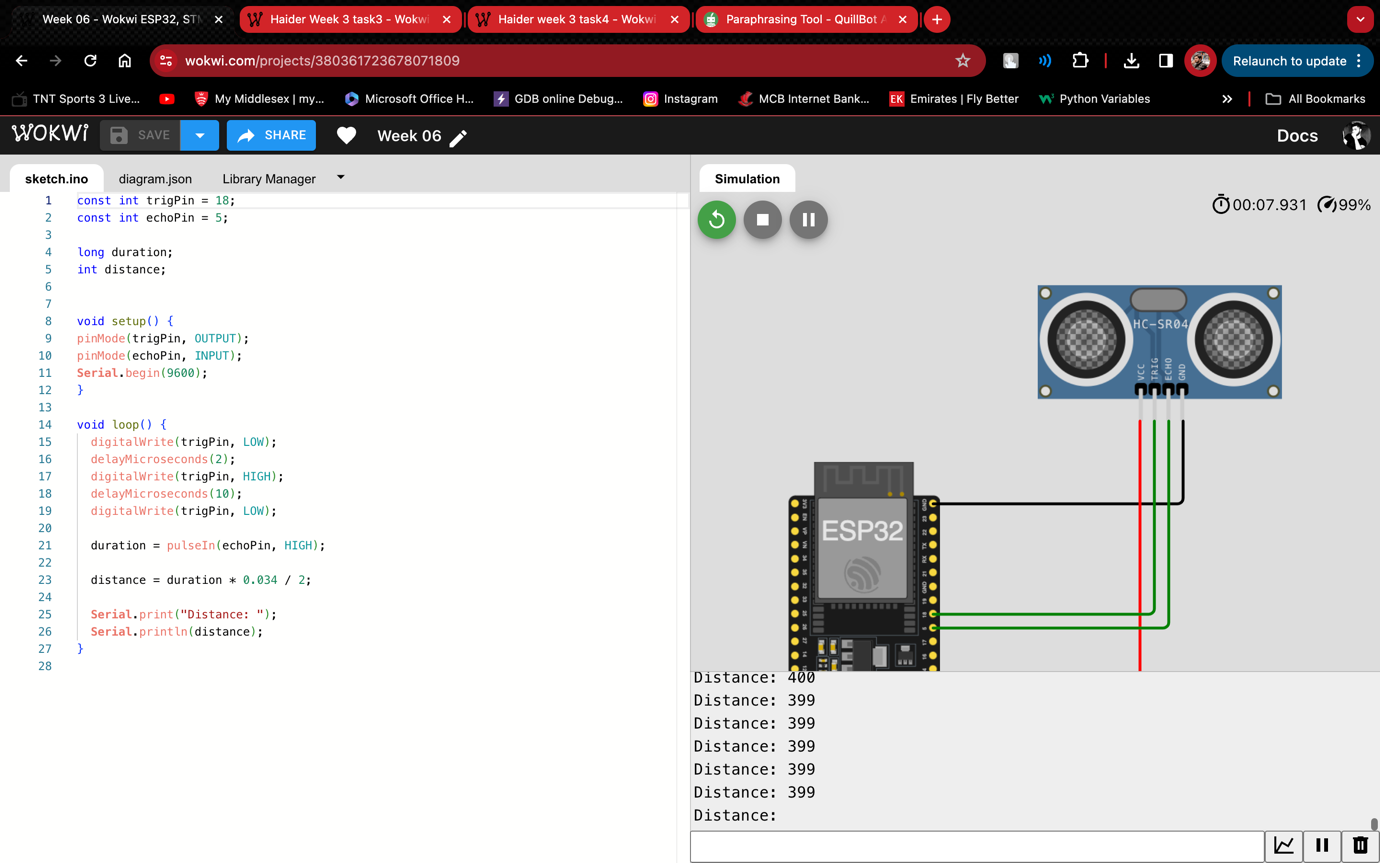
**Task 2. Using 4 Wires.**

****

It demonstrates how to connect an ultrasonic sensor to an Arduino board. The programme is being executed on Wokwi, a simulator for Arduino, Raspberry Pi Pico, and ESP32 devices.   
The screenshot's text displays the program's code, which consists of the following steps:  
1. Define pins for ultrasonic sensor trigger and echo signals.   
  
2. Configure the trigger pin as an output and the echo pin as an input.   
3. Send a pulse signal to the trigger pin to activate the ultrasonic sensor.   
4. Record the time it takes the pulse to travel to and from the object.   
5. Determine the distance to the item using the speed of sound.

6. Display the distance to the object on the serial display.  
The screenshot also shows the program's current reading of 399 cm.

**Task 3. Using 3 Wires.**

****

The ultrasonic sensor measures distance by sending out a sound wave and then measuring how long it takes for the wave to reflect off an object and return to the sensor. The ESP32 microcontroller is utilised to operate the ultrasonic sensor and interpret distance readings.  
  
The text in the image represents the code for the ESP32 microcontroller programme. The code has the following steps:  
  
Prepare the pins for the ultrasonic sensor.  
Send a sound wave using the ultrasonic sensor.  
Determine how long it takes for a sound wave to reflect off an item and return to the sensor.  
Calculate the distance to the item using the time it takes for the sound wave to travel.

Print out the distance to the serial monitor.  
The serial monitor's output, seen at the bottom of the screen, shows the distance to an object in centimetres.

# Week-7

**Task 2: Using 1 Arduino or Esp32/8266 board and 1 LED, program MCU to make the LED blink for every 1 sec.**

void setup() {

  // initialize digital pin LED\_BUILTIN as an output.

  pinMode(13, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

  digitalWrite(13, HIGH);  // turn the LED on (HIGH is the voltage level)

  delay(1000);                      // wait for a second

  digitalWrite(13, LOW);   // turn the LED off by making the voltage LOW

  delay(1000);                      // wait for a second

}

This code snippet controls an LED connected to pin 13 of an Arduino board. Here's the breakdown:  
  
  
1. Setup() function:  
  
  
pinMode(13, OUTPUT): Configures pin 13 as an output pin, allowing it to send voltage to external devices such as LEDs.  
2. The loop() method (which runs continuously):  
  
  
digitalWrite(13, HIGH): Enables the LED linked to pin 13 by setting the pin's voltage to HIGH (typically 5V).  
delay(1000): Pauses the programme for one second (1000 milliseconds).  
digitalWrite(13, LOW): Disables the LED by lowering the voltage on the pin (often 0V).  
delay(1000): Pauses the programme for an additional 1 second.



**Task 3: Using 1 Arduino or Esp32/8266 board and 3 LED of different colours, program MCU to make the 3 LED blink one after another or all together in a sequence for every 1 sec.**

void setup()

{

  pinMode(2, OUTPUT);

  pinMode(3, OUTPUT);

  pinMode(4, OUTPUT);

}

void loop()

{

  digitalWrite(2, HIGH);

  delay(1000); // Wait for 1000 millisecond(s)

  digitalWrite(2, LOW);

  delay(1000); // Wait for 1000 millisecond(s)

   digitalWrite(3, HIGH);

  delay(1000); // Wait for 1000 millisecond(s)

  digitalWrite(3, LOW);

  delay(1000); // Wait for 1000 millisecond(s)

   digitalWrite(4, HIGH);

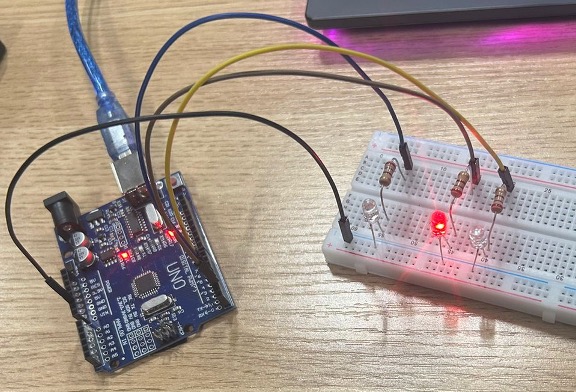
  delay(1000); // Wait for 1000 millisecond(s)

  digitalWrite(4, LOW);

  delay(1000); // Wait for 1000 millisecond(s)

}

This code generates a blinking LED pattern by turning each of the three LEDs attached to pins 2, 3, and 4 on for 1 second and then off for 1 second, resulting in a sequential blinking effect.

****

**Task 4: Using 1 Arduino or Esp32/8266 board and 1 RGB LED, program MCU to produce red, green, blue, cyan, magenta, yellow and white colours after every 1 sec.**

void setup()

{

  pinMode(9, OUTPUT);

  pinMode(10, OUTPUT);

  pinMode(11, OUTPUT);

}

void loop()

{

  analogWrite(9, HIGH);

  delay(1000); // Wait for 1000 millisecond(s)

  analogWrite(9, LOW);

  delay(1000); // Wait for 1000 millisecond(s)

   analogWrite(10, HIGH);

  delay(1000); // Wait for 1000 millisecond(s)

  analogWrite(10, LOW);

  delay(1000); // Wait for 1000 millisecond(s)

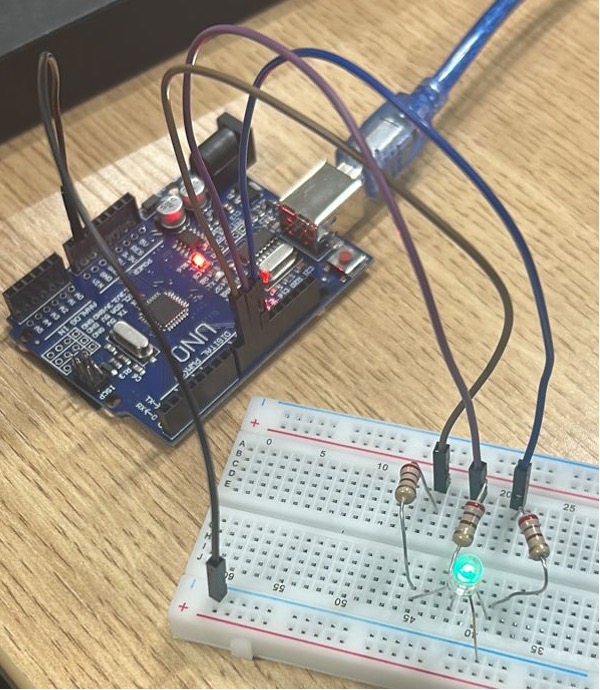
  analogWrite(11, HIGH);

  delay(1000); // Wait for 1000 millisecond(s)

  analogWrite(11, LOW);

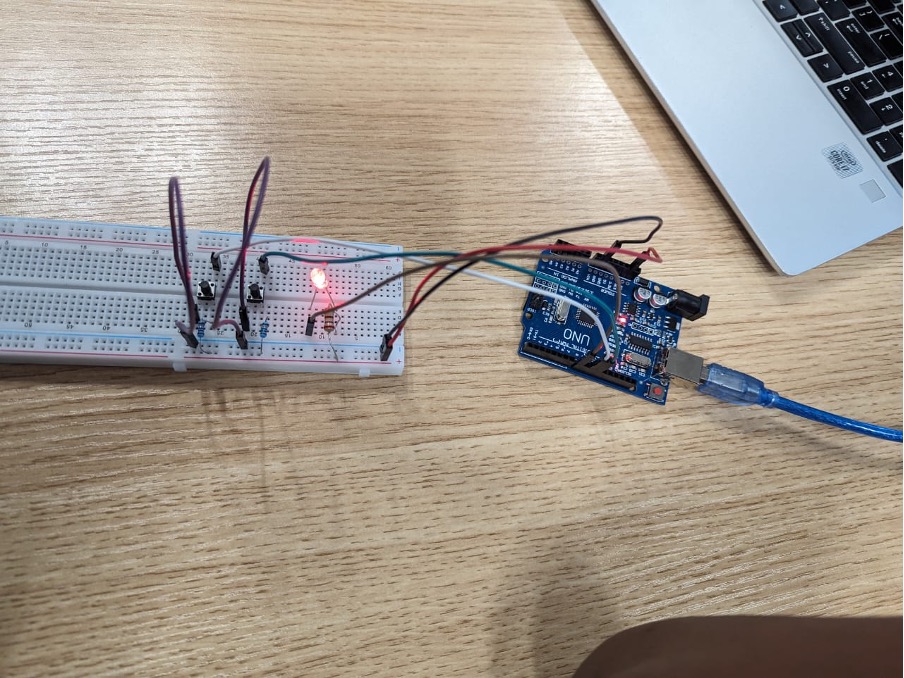
  delay(1000); // Wait for 1000 millisecond(s)

}

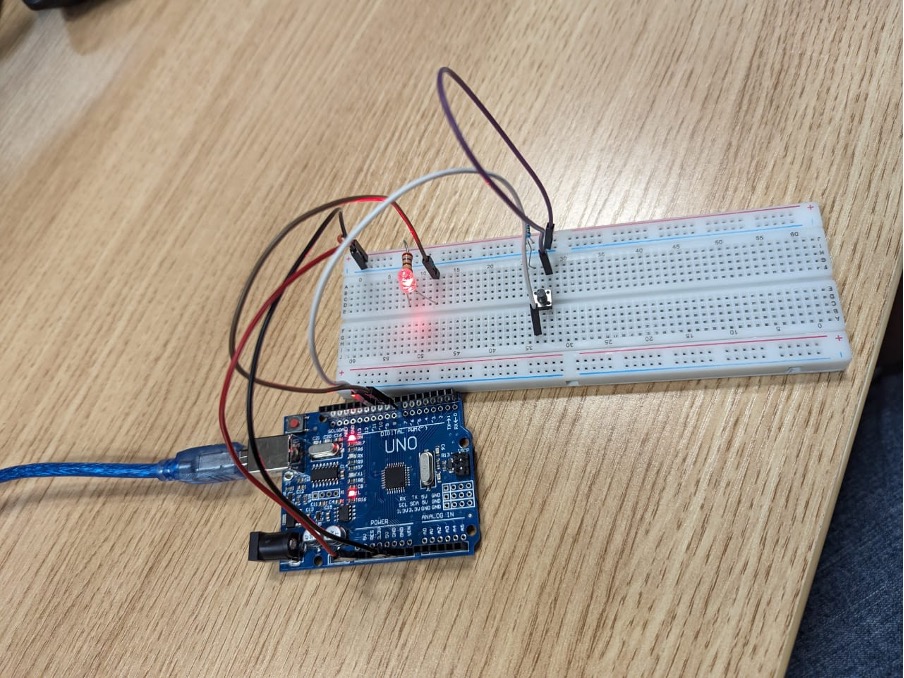
****

# Week-8

**Task 5: Using 1 Arduino or Esp32/8266 board, 1 LED and 2 push buton, program MCU to switch on the LED, if the user presses the first buton and switch off the LED, if the user presses the second buton.**

****

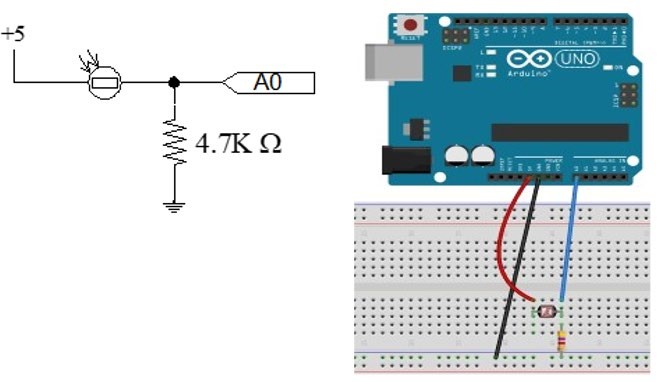
**Task 6: Using 1 Arduino or Esp32/8266 board, 1 LED and 1 push buton, program MCU to toggle the LED with on and off with the pressing of the buton.**

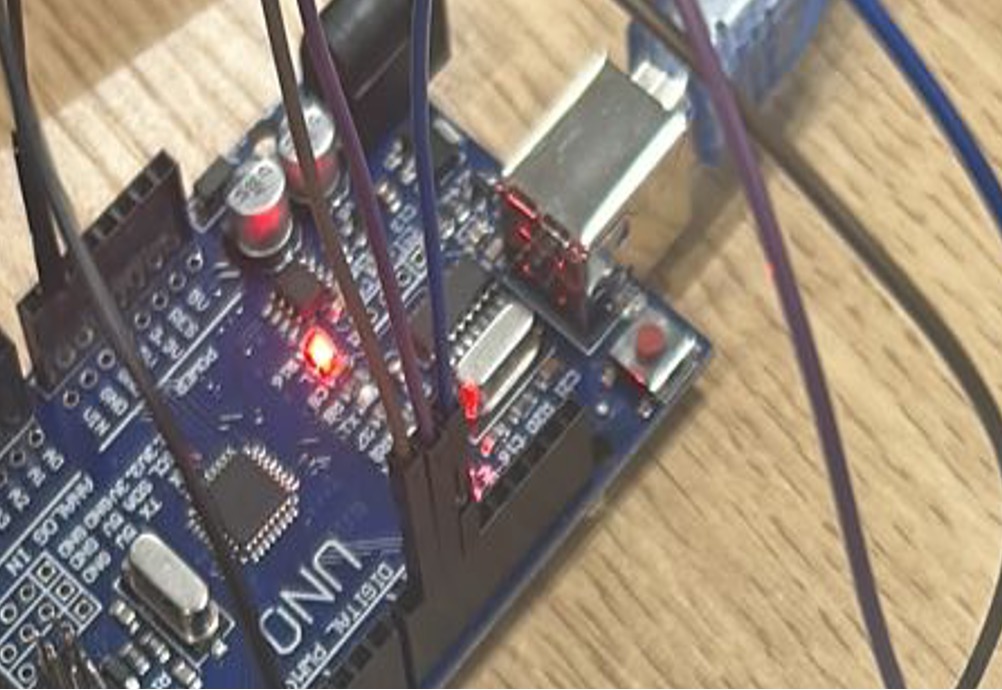
****

**Task 7: Using 1 Arduino or Esp32/8266 board, 1 temperature sensor (TMP36), program MCU to read its input value and convert it into appropriate temperature value and display the value in Serial Monitor. Hint:**

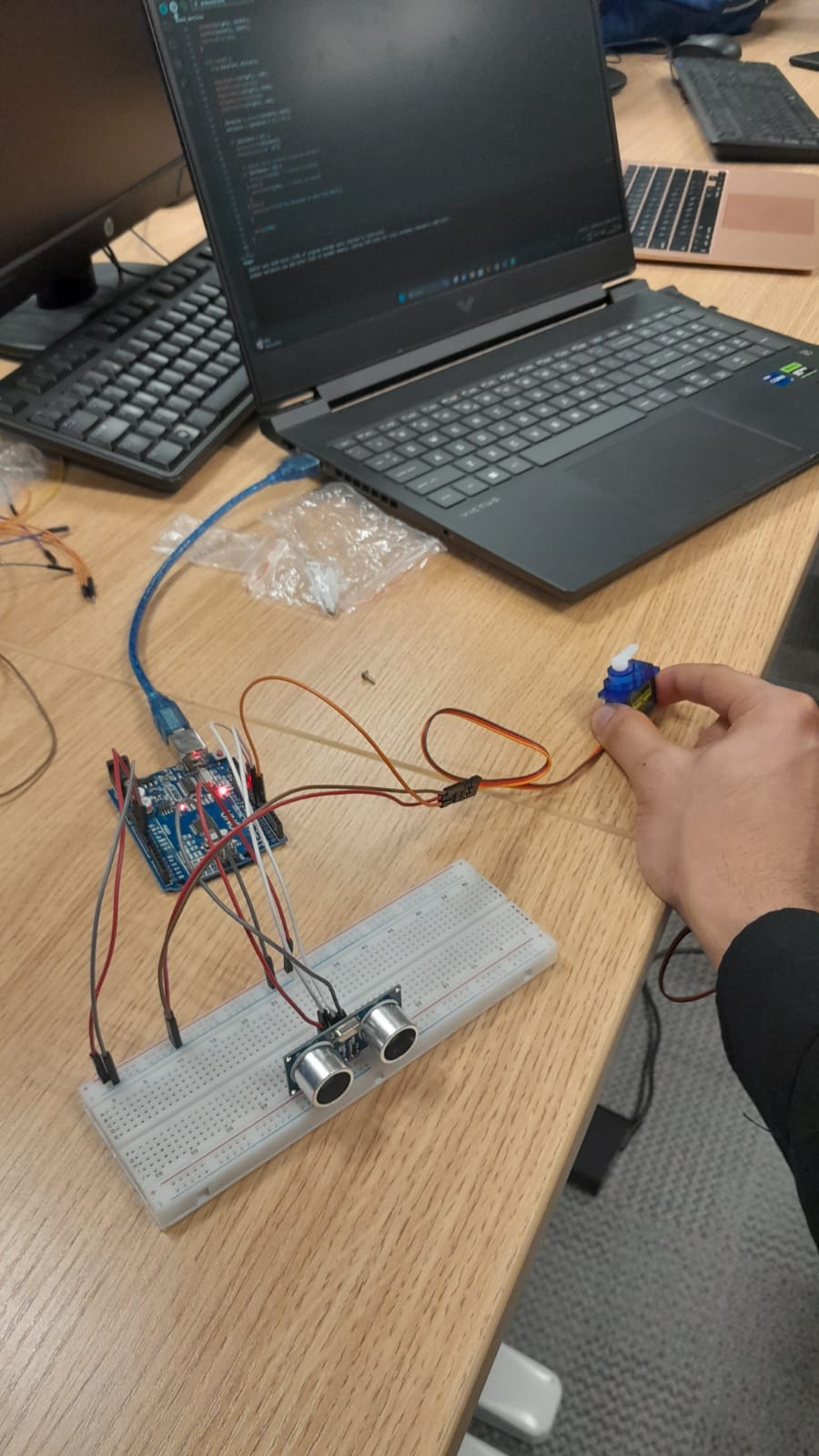
sensorInput = analogRead(A0); //read the analog sensor and store it temp = (double)sensorInput / 1024; //find percentage of input reading temp = temp \* 5; //multiply by 5V to get voltage temp = temp – 0.5; //Subtract the offset temp = temp \* 100; //Convert to degrees

**Task 8: Design the system as shown below and try to read the input analog value from light dependant resistor (LDR) and display in Serial Monitor.**





# Week-9



# Week-13

**Task 1. Accessing the Built-in RGB LED on the MKR WiFi 1010 - Program the MKR WiFi 1010’s built-in RGB LED to make it blink in three different colours.**

#include <WiFiNINA.h>

#include <utility/wifi\_drv.h>

void setup() {

  WiFiDrv::pinMode(25, OUTPUT); // Define green pin

  WiFiDrv::pinMode(26, OUTPUT); // Define red pin

  WiFiDrv::pinMode(27, OUTPUT); // Define blue pin

}

void loop() {

  WiFiDrv::analogWrite(25, 255);

  WiFiDrv::analogWrite(26, 0);

  WiFiDrv::analogWrite(27, 0);

  delay(1000);

  WiFiDrv::analogWrite(25, 0);

  WiFiDrv::analogWrite(26, 255);

  WiFiDrv::analogWrite(27, 0);

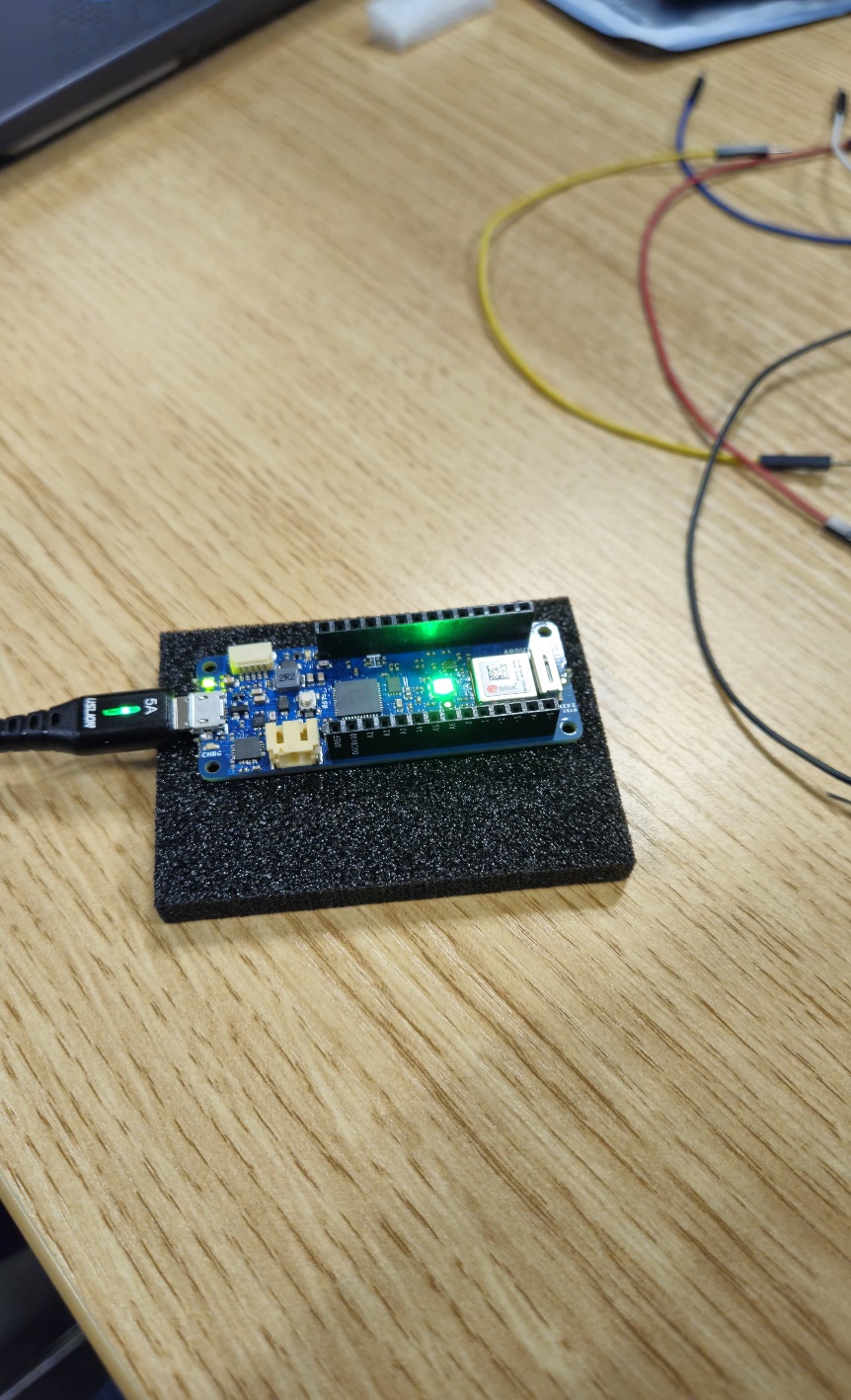
  delay(1000);

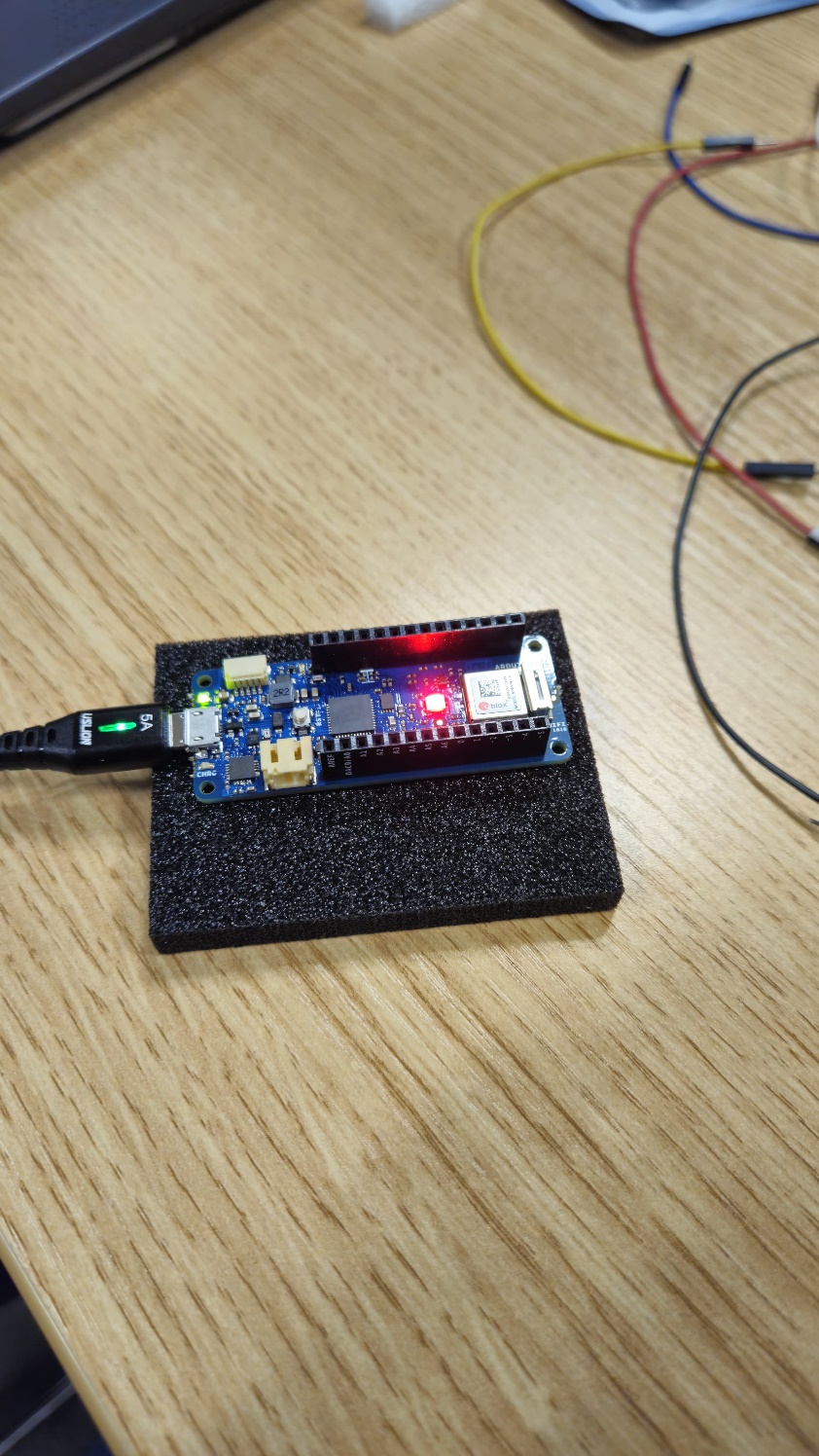
  WiFiDrv::analogWrite(25, 0);

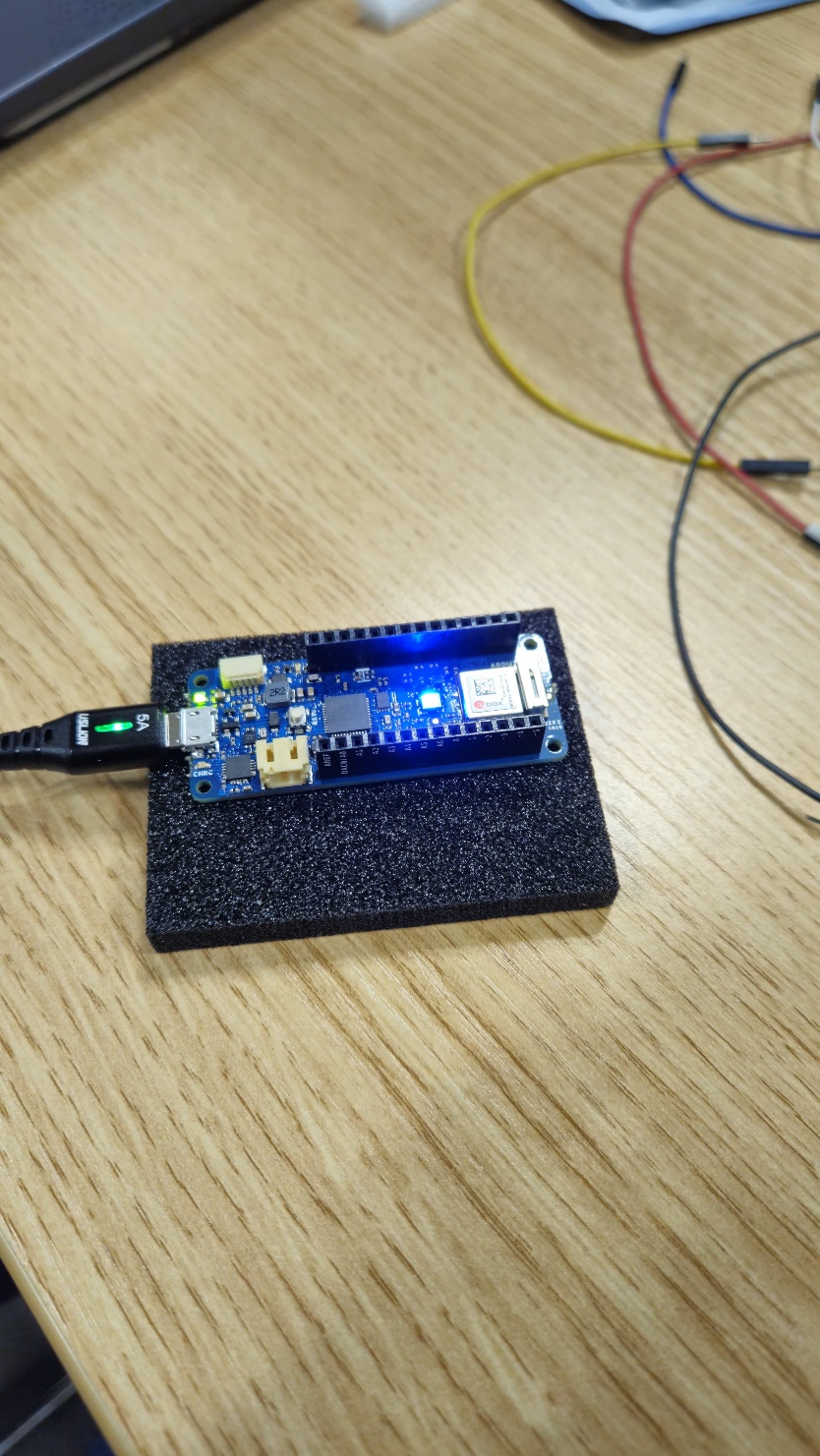
  WiFiDrv::analogWrite(26, 0);

  WiFiDrv::analogWrite(27, 255);

  delay(1000);







Code Breakdown:  
  
1. Header Inclusion:  
  
#include WiFiNINA.h>: This line includes the WiFiNINA library, which enables Wi-Fi connectivity on specific Arduino boards. However, it is important to remember that this library may not work with all Arduino boards or programming environments.  
#include utility/wifi\_drv.h>: This line likely contains a special header file named wifi\_drv.h, which may be unique to your configuration or board. Without the contents of this file, it is hard to discern its precise purpose or the functions it defines.  
2. Setup() Function:  
  
This function is called only once when the code begins to run.  
WiFiDrv::pinMode(25, OUTPUT): This line configures the pin attached to the green LED (pin 25) as an output pin, allowing the code to control its state (on/off).  
WiFiDrv::pinMode(26, OUTPUT);: Similarly, this line sets the pin attached to the red LED (pin 26) as an output.  
WiFiDrv::pinMode(27, OUTPUT);: Similarly, this line configures the pin associated with the blue LED (pin 27) as an output.  
3. Loop() Function:  
  
This function will continue to run as long as the code is active.  
The code inside the loop() function generates a basic color-changing pattern with the three LEDs:  
Green: WiFiDrv::analogWrite(25, 255);: Enables full brightness for the green LED.  
delay(1000): Pauses for one second.  
WiFiDrv::analogWrite(25, 0): Turns off the green LED (0 brightness).  
WiFiDrv::analogWrite(26, 255); sets the red LED to full brightness.  
delay(1000): Pauses for one second.  
WiFiDrv::analogWrite(26, 0): Turns off the red LED.  
WiFiDrv::analogWrite(27, 255);: Activates the blue LED at full brightness.  
delay(1000): Pauses for one second.  
WiFiDrv::analogWrite(27, 0): Turns off the blue LED.

**Task 2. Serial to OLED with MKR WiFi 1010 - Using 1 MKR WiFi 1010, 1 OLED screen, program the MCU such that a personalised message is displayed on the OLED.**

#include <SPI.h>

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#define SCREEN\_WIDTH 128 // OLED display width, in pixels

#define SCREEN\_HEIGHT 32 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)

#define OLED\_RESET 4 // Reset pin # (or -1 if sharing Arduino reset pin)

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &**Wire**, OLED\_RESET);

String message = "WEEK 13"; // Set the initial message

int text\_size; // Changes text size of the display

void setup() {

**Serial**.begin(9600);

  // SSD1306\_SWITCHCAPVCC = generate display voltage from 3.3V internally

  if (!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) { // Address 0x3C for 128x32

**Serial**.println(F("SSD1306 allocation failed"));

    for (;;); // loop forever if library fails to initialize

  }

  display.setTextColor(SSD1306\_WHITE); // Set text color to white

}

void loop() {

  // Clear the previous message on each iteration

  display.clearDisplay();

  // Set the text size based on the length of the message

  if (message.length() > 0 && message.length() <= 5) {

    display.setTextSize(3);

  } else if (message.length() > 5 && message.length() <= 20) {

    display.setTextSize(2);

  } else if (message.length() > 20) {

    display.setTextSize(1);

  }

  // Print the message on the OLED screen

  display.setCursor(0, 0);

  display.println(message);

  display.display();

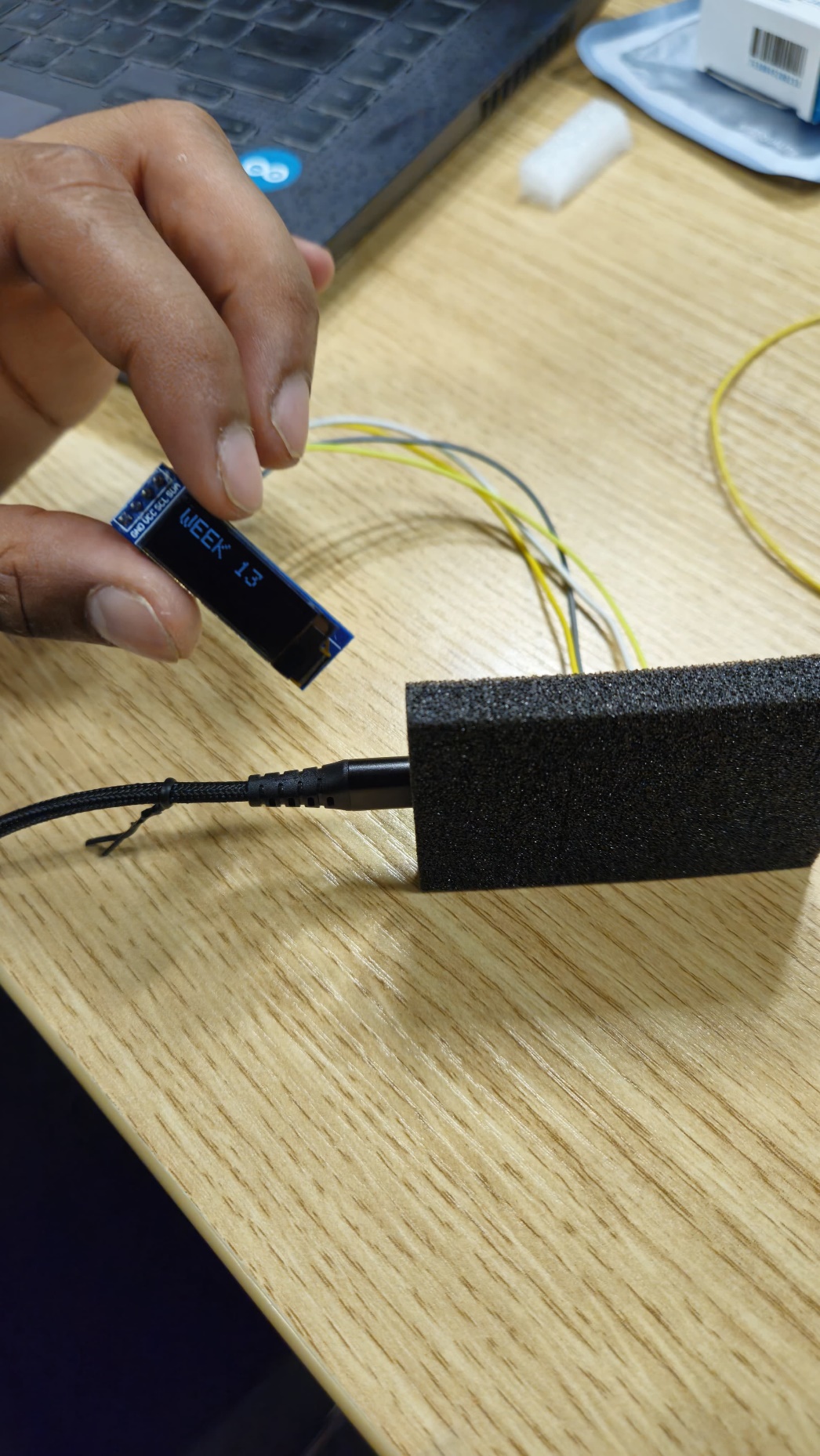
  // Print the message in the Serial monitor for feedback

**Serial**.println("Display has been updated with: " + message);

  // Delay for a short time before updating the display again

  delay(5000);

}



# Week-14

**Task 1.**

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#include <DHT.h>

#define SCREEN\_WIDTH 128 // OLED display width, in pixels

#define SCREEN\_HEIGHT 64 // OLED display height, in pixels

#define OLED\_RESET 4 // Reset pin # (or -1 if sharing Arduino reset pin)

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &**Wire**, OLED\_RESET);

#define DHTPIN 3     // Pin where the DHT sensor is connected

#define DHTTYPE DHT11 // DHT 11 sensor

DHT dht(DHTPIN, DHTTYPE);

void setup() {

**Serial**.begin(9600);

  if (!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) { // Address 0x3C for 128x64

**Serial**.println(F("SSD1306 allocation failed"));

    for (;;); // loop forever if library fails to initialize

  }

  display.setTextColor(SSD1306\_WHITE); // Set text color to white

}

void loop() {

  delay(2000);

  float humidity = dht.readHumidity();

  // Check if the humidity reading is valid

  if (!isnan(humidity)) {

    // Display humidity on OLED

    display.clearDisplay();

    display.setTextSize(2);

    display.setCursor(0, 0);

    display.print(F("Humidity: "));

    display.print(humidity);

    display.print(F(" %"));

    display.display();

    // Print the humidity percentage in the Serial monitor for feedback

**Serial**.print(F("Humidity: "));

**Serial**.print(humidity);

**Serial**.println(F(" %"));

  } else {

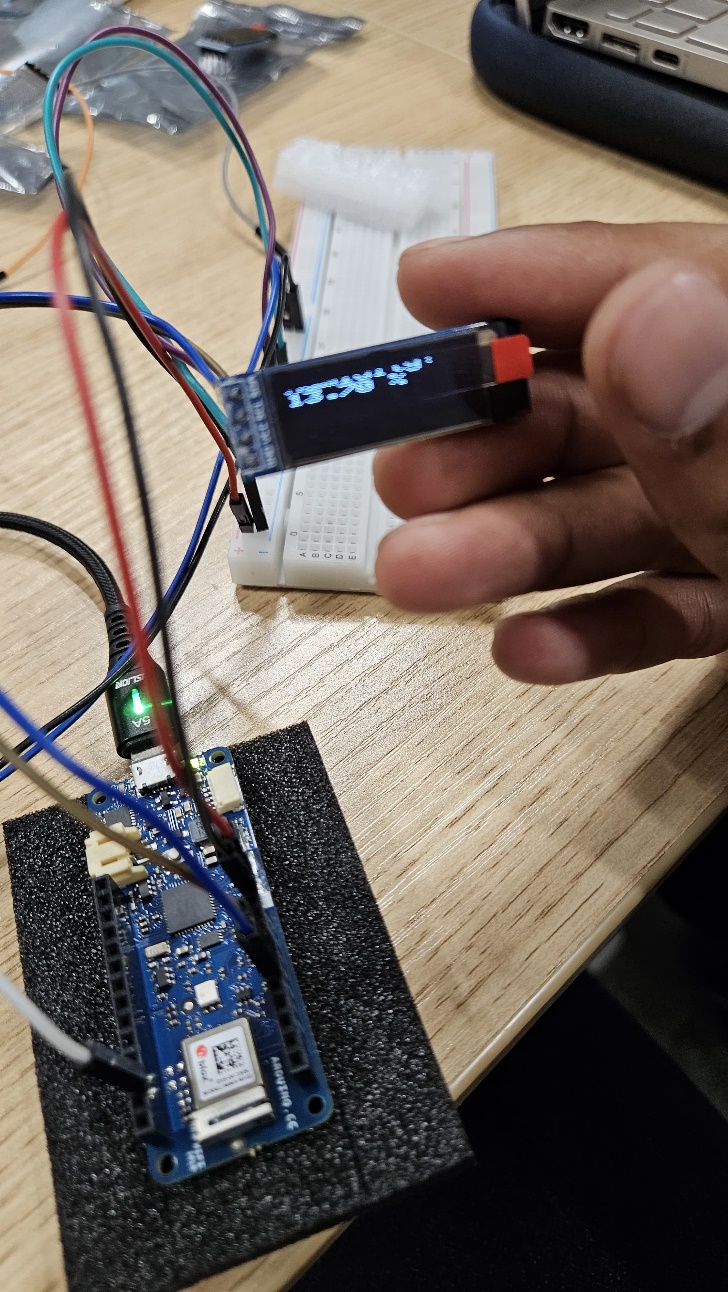
**Serial**.println(F("Failed to read humidity. Check the sensor and wiring."));

  }

  // Delay for a short time before updating the display again

  delay(5000);

}



# Week-15

**Task 1. Controlling LED from the cloud - Program the MKR WiFi 1010 such that an LED connected to PIN 9 is able to be controlled from the cloud. (use Blynk)**

\

#define BLYNK\_TEMPLATE\_ID "TMPL2LiapQ39n"

#define BLYNK\_TEMPLATE\_NAME "test"

#define BLYNK\_AUTH\_TOKEN "vHN5XoLUhvsPtd9evi3xc07PY0ZWN7yw"

#define BLYNK\_PRINT **Serial**

#include <WiFiNINA.h>

#include <BlynkSimpleWiFiNINA.h>

// You should get Auth Token in the Blynk App.

char auth[] = "vHN5XoLUhvsPtd9evi3xc07PY0ZWN7yw";

// Your WiFi credentials.

char ssid[] = "CST2590";            //  your network SSID (name) between the " "

char pass[] = "password";         // your network password between the " "

void setup()

{

**Serial**.begin(9600);

  Blynk.begin(auth, ssid, pass);

}

void loop()

{

  Blynk.run();

}

BLYNK\_WRITE(V0)  //read a value from cloud

{

  int value = param.asInt(); // Get value as integer

  digitalWrite(9, value); //blue led

**Serial**.println(value);

}

BLYNK\_WRITE(V1)  //read a value from cloud

{

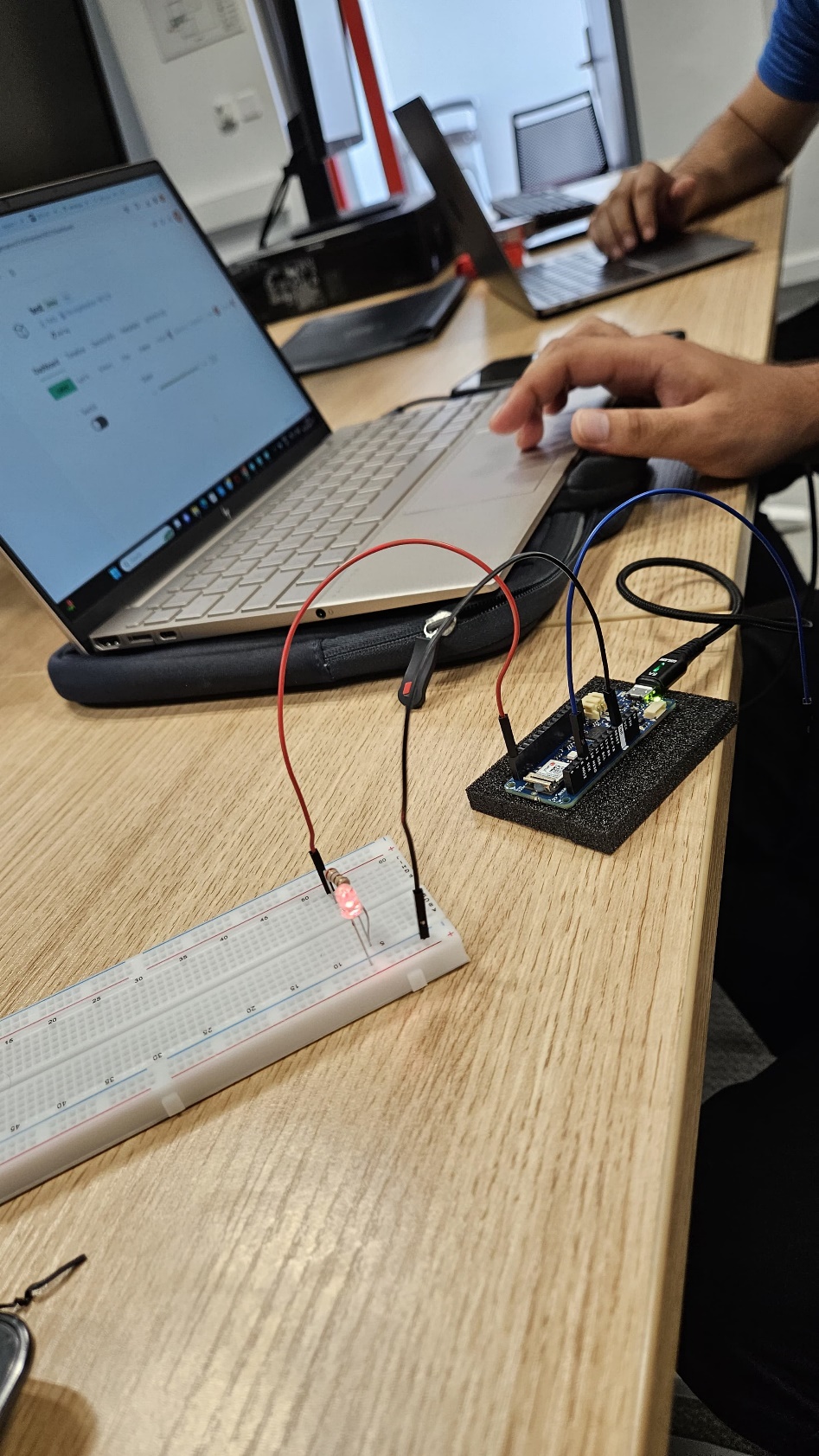
  int value2 = param.asInt(); // Get value as integer

  analogWrite(5, value2); //red led

**Serial**.println(value2);

}

//Blynk.virtualWrite(V7, 123);



# Week-16

**Reading the Humidity and displaying it on the OLED.**

Using 1 MKR WiFi 1010, 1 OLED screen, 1 humidity sensor, program the MKR WiFi 1010 to display the humidity in percentage on the OLED.

#define BLYNK\_TEMPLATE\_ID "TMPL2LiapQ39n"

#define BLYNK\_TEMPLATE\_NAME "test"

#define BLYNK\_AUTH\_TOKEN "vHN5XoLUhvsPtd9evi3xc07PY0ZWN7yw"

#define BLYNK\_PRINT Serial

#include <WiFiNINA.h>

#include <BlynkSimpleWiFiNINA.h>

// You should get Auth Token in the Blynk App.

char auth[] = "vHN5XoLUhvsPtd9evi3xc07PY0ZWN7yw";

// Your WiFi credentials.

char ssid[] = "CST2590";            //  your network SSID (name) between the " "

char pass[] = "password";         // your network password between the " "

void setup()

{

  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);

}

void loop()

{

  Blynk.run();

}

BLYNK\_WRITE(V0)  //read a value from cloud

{

  int value = param.asInt(); // Get value as integer

  digitalWrite(9, value); //blue led

  Serial.println(value);

}

BLYNK\_WRITE(V1)  //read a value from cloud

{

  int value2 = param.asInt(); // Get value as integer

  analogWrite(5, value2); //red led

  Serial.println(value2);

}

//Blynk.virtualWrite(V7, 123);

# Reference List:

# Richard Bevis. (2023). “7 Examples of IoT in Everyday Life”. CBTnuggets. Available at: <https://www.cbtnuggets.com/blog/technology/networking/seven-examples-of-iot-in-everyday-life> (Accessed: 10 October 2023).

1. Rajiv. (2023). “What are the major components of internet of Things”. RF Page. Available at: <https://www.rfpage.com/what-are-the-major-components-of-internet-of-things/> (Accessed: 10 October 2023).
2. Nikita Zubovich. (2023). “Advantages of Internet of Things: 10 Benefits you should know”. SUMATOSOFT. Available at: <https://sumatosoft.com/blog/advantages-of-internet-of-things-10-benefits-you-should-know> (Accessed: 10 October 2023).
3. Nikita Zubovich. (2023). “Advantages of Internet of Things: 10 Benefits you should know”. SUMATOSOFT. Available at: <https://sumatosoft.com/blog/advantages-of-internet-of-things-10-benefits-you-should-know> (Accessed: 10 October 2023).
4. Michael Mendoza. (2023). “IoT in Manufacturing: Applications Today and the Future”. Hitachi Solutions. Available at: <https://global.hitachi-solutions.com/blog/industrial-iot-benefits/> (Accessed: 10 October 2023).
5. Kasia Gerlee. (2023). “IoT in agriculture – 6 smart farming examples”. Freeway. Available at: <https://freeeway.com/iot-in-agriculture-6-smart-farming-examples/> )Accessed: 10 October 2023).
6. Dileep Gupta. (2023). “Understanding the Impact of IoT in Healthcare”. Appinventiv. Available at: <https://appinventiv.com/blog/iot-in-healthcare/#:~:text=End%2Dto%2DEnd%20Connectivity%20and%20Affordability&text=For%20example%2C%20IoT%20can%20connect,efficiencies%20and%20improving%20patient%20outcomes>. (Accessed: 11 October 2023).
7. Harkiran. (2020). “10 Applications of IoT in Creating Smart Cities”. Geeks for Geeks. Available at: <https://www.geeksforgeeks.org/10-applications-of-iot-in-creating-smart-cities/> (Accessed: 11 October 2023).
8. Alexander S. Gillis. (2023). “industrial internet of things (IIoT)”. TechTarget. Available at: <https://www.techtarget.com/iotagenda/definition/Industrial-Internet-of-Things-IIoT> (Accessed: 11 October 2023).
9. Norton. (2019). “The future of IoT: 10 predictions about the Internet of Things”. Norton. Available at: <https://us.norton.com/blog/iot/5-predictions-for-the-future-of-iot> (Accessed: 11 October 2023).
10. Will Kenton. (2022). What is the Internet of Things (IoT)? How It Works and Benefits”. Investopedia. Available at: <https://www.investopedia.com/terms/i/internet-things.asp> (Accessed: 11 October 2023).
11. Rashmi Bhardwaj. (2020). “IoT vs M2M: Difference between M2M and IoT Explained”. IPWITHEASE. Available at: <https://ipwithease.com/internet-of-things-vs-machine-to-machine-iot-vs-m2m/> (Accessed: 12 October 2023).
12. Vijay Kanade. (2022). “What is the Internet of ESv erything? Meaning, Examples, and Uses”. Spiceworks. Available at: <https://www.spiceworks.com/tech/iot/articles/what-is-internet-of-everthing/> (Accessed: 12 October 2023).
13. Sebastian. (2021). “Introduction to Microcontrollers and Single-Board Computers”. Medium. Available at: <https://medium.com/geekculture/introduction-to-microcontrollers-and-single-board-computers-5e2fa3011279#:~:text=Also%2C%20a%20microcontroller%20does%20not,%2C%20USB%2C%20Wifi%20and%20Bluetooth>. (Accessed: 12 October 2023).

# Richard Bevis. (2023). “7 Examples of IoT in Everyday Life”. CBTnuggets. Available at: <https://www.cbtnuggets.com/blog/technology/networking/seven-examples-of-iot-in-everyday-life> (Accessed: 10 October 2023).

1. Rajiv. (2023). “What are the major components of internet of Things”. RF Page. Available at: <https://www.rfpage.com/what-are-the-major-components-of-internet-of-things/> (Accessed: 10 October 2023).
2. Nikita Zubovich. (2023). “Advantages of Internet of Things: 10 Benefits you should know”. SUMATOSOFT. Available at: <https://sumatosoft.com/blog/advantages-of-internet-of-things-10-benefits-you-should-know> (Accessed: 10 October 2023).
3. Nikita Zubovich. (2023). “Advantages of Internet of Things: 10 Benefits you should know”. SUMATOSOFT. Available at: <https://sumatosoft.com/blog/advantages-of-internet-of-things-10-benefits-you-should-know> (Accessed: 10 October 2023).
4. Michael Mendoza. (2023). “IoT in Manufacturing: Applications Today and the Future”. Hitachi Solutions. Available at: <https://global.hitachi-solutions.com/blog/industrial-iot-benefits/> (Accessed: 10 October 2023).
5. Kasia Gerlee. (2023). “IoT in agriculture – 6 smart farming examples”. Freeway. Available at: <https://freeeway.com/iot-in-agriculture-6-smart-farming-examples/> )Accessed: 10 October 2023).
6. Dileep Gupta. (2023). “Understanding the Impact of IoT in Healthcare”. Appinventiv. Available at: <https://appinventiv.com/blog/iot-in-healthcare/#:~:text=End%2Dto%2DEnd%20Connectivity%20and%20Affordability&text=For%20example%2C%20IoT%20can%20connect,efficiencies%20and%20improving%20patient%20outcomes>. (Accessed: 11 October 2023).
7. Harkiran. (2020). “10 Applications of IoT in Creating Smart Cities”. Geeks for Geeks. Available at: <https://www.geeksforgeeks.org/10-applications-of-iot-in-creating-smart-cities/> (Accessed: 11 October 2023).
8. Alexander S. Gillis. (2023). “industrial internet of things (IIoT)”. TechTarget. Available at: <https://www.techtarget.com/iotagenda/definition/Industrial-Internet-of-Things-IIoT> (Accessed: 11 October 2023).
9. Norton. (2019). “The future of IoT: 10 predictions about the Internet of Things”. Norton. Available at: <https://us.norton.com/blog/iot/5-predictions-for-the-future-of-iot> (Accessed: 11 October 2023).
10. Will Kenton. (2022). What is the Internet of Things (IoT)? How It Works and Benefits”. Investopedia. Available at: <https://www.investopedia.com/terms/i/internet-things.asp> (Accessed: 11 October 2023).
11. Rashmi Bhardwaj. (2020). “IoT vs M2M: Difference between M2M and IoT Explained”. IPWITHEASE. Available at: <https://ipwithease.com/internet-of-things-vs-machine-to-machine-iot-vs-m2m/> (Accessed: 12 October 2023).
12. Vijay Kanade. (2022). “What is the Internet of ESv erything? Meaning, Examples, and Uses”. Spiceworks. Available at: <https://www.spiceworks.com/tech/iot/articles/what-is-internet-of-everthing/> (Accessed: 12 October 2023).
13. Sebastian. (2021). “Introduction to Microcontrollers and Single-Board Computers”. Medium. Available at: <https://medium.com/geekculture/introduction-to-microcontrollers-and-single-board-computers-5e2fa3011279#:~:text=Also%2C%20a%20microcontroller%20does%20not,%2C%20USB%2C%20Wifi%20and%20Bluetooth>. (Accessed: 12 October 2023).

# Jost, Danny. (2019) ‘What is an Ultrasonic sensor?’, FIERCE Electronics. Available at: <https://www.fierceelectronics.com/sensors/what-ultrasonic-sensor> (Accessed: 4 february 2024).