

الجامعة الإسلامية العالمية ماليزيا
INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA
يُونِيسَيْتِي إِسْلَامُ، إِنْتَارَايْغُسَا مَلِيسِيَا

Garden of Knowledge and Virtue

MECHATRONICS SYSTEM INTEGRATION (MCTA 3203)

SEMESTER 2, 2024/2025

ACTIVITY REPORT

WEEK 8

BLUETOOTH

DATE OF SUBMISSION:

5.5.2025

| NAME(GROUP 7) | MATRIC NO. |
|--|------------|
| MUHAMMAD AMMAR ZUHAIR BIN NOR AZMAN SHAH | 2110259 |
| MUHAMMAD NURUDDEEN BIN MOHAMMAD NASIR | 225943 |
| MUHAMMAD AKMAL NIZHAM BIN AZLAN | 2213323 |

TABLE OF CONTENTS

| | |
|------------------------------------|-----------|
| TABLE OF CONTENTS..... | 1 |
| INTRODUCTION..... | 2 |
| MATERIAL AND EQUIPMENT..... | 3 |
| METHODOLOGY..... | 5 |
| RESULTS..... | 6 |
| DISCUSSION..... | 8 |
| CONCLUSION..... | 10 |
| RECOMMENDATIONS..... | 11 |
| STUDENT'S DECLARATION..... | 12 |

INTRODUCTION

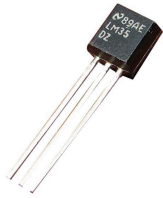
The objective of this experiment is to create a wireless temperature monitoring system using Wi-Fi, Arduino, and a temperature sensor or thermistor. The Arduino will read temperature data from the thermistor, send it to a Python script over Wi-Fi, and the Python script will display and log the temperature.

MATERIAL AND EQUIPMENT

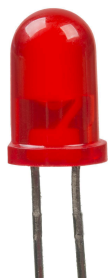
1. ESP32



2. Temperature sensor, LM35



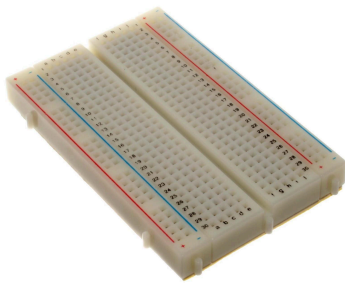
3. LED



4. Smartphone with Bluetooth support



5. Breadboard



6. Jumper wires

METHODOLOGY

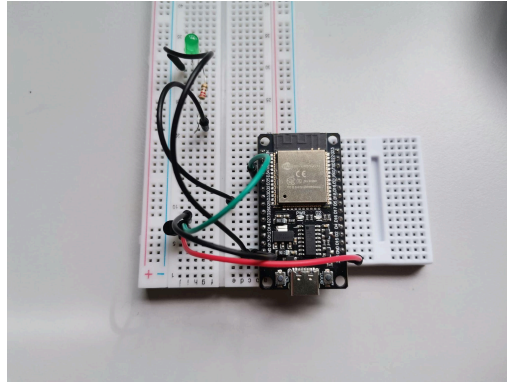


Figure 1

1. The LM35 sensor legs are connected from the left to 3.3v, pin 34 and GND, respectively, on the ESP32.
2. The LED's cathode leg is connected to the GND while the anode leg is connected to pin 25, in series with a 220Ω resistor.
3. Libraries that are required for the ESP32 are installed.
4. Codes are uploaded into the ESP32 to initialise Bluetooth communication.
5. An app named Serial Bluetooth Terminal is installed on a Bluetooth-capable phone.
6. Bluetooth is switched on, and the phone is paired with the ESP32.
7. Once the devices are paired, the app is launched to begin connecting the app with ESP32.
8. Once connected, temperature data will be automatically printed.
9. Commands can be sent through the app to control the LED.

RESULTS

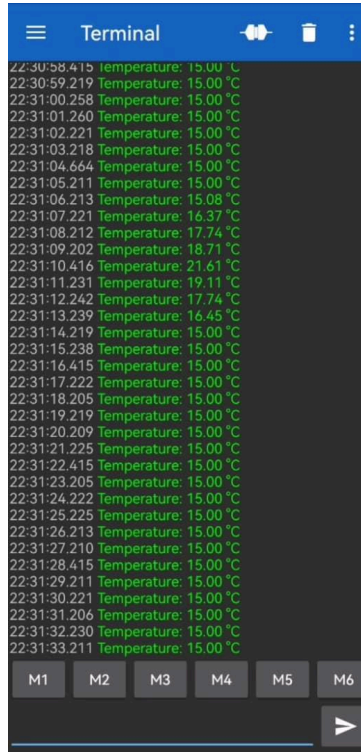


Figure 2

As shown in Figure 2 above, the app Bluetooth Serial Terminal printed the temperature data that was collected by the LM35 sensor through ESP32 via Bluetooth. However, the data that we got constantly prints out 0. Due to that, we had to add a certain value to offset the error.

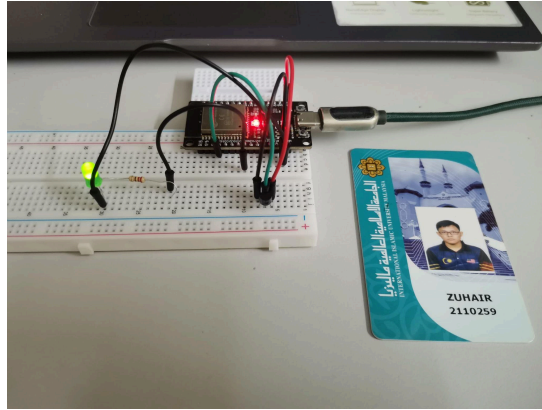


Figure 3

Other than that, as shown in Figure 3, we successfully managed to control the LED by sending a command through the Bluetooth Serial Terminal app. To switch on the LED, a command '1' is sent. To switch off the LED, a command of '0' is sent. The action that was done was also printed, which can be used as a record for the user for reference.

DISCUSSION

Arduino IDE code:

```
#include "BluetoothSerial.h"

BluetoothSerial SerialBT;

const int lm35Pin = 34;
const int ledPin = 25;

void setup() {
    Serial.begin(115200);
    SerialBT.begin("ESP32_temp");
    Serial.println("Bluetooth started. Waiting for connection...");

    pinMode(ledPin, OUTPUT);
    digitalWrite(ledPin, LOW);
}

void loop() {
    //Temperature reading
    int adcValue = analogRead(lm35Pin);
    float voltage = adcValue * (3.3/4095.0);
    float temperature = (voltage * 100.0) + 15.0;
    SerialBT.print("Temperature: ");
    SerialBT.print(temperature);
    SerialBT.println(" °C");
    delay(1000);

    //Lamp
    if(SerialBT.available()){
        char command = SerialBT.read();
        if(command == '1'){
            digitalWrite(ledPin, HIGH);
            SerialBT.println("Lamp is switched ON");
        }
        else if(command == '0'){
            digitalWrite(ledPin, LOW);
            SerialBT.println("Lamp is OFF");
        }
    }
}
```

```
}  
}  
}
```

Global variable:

In the global variable, we include a library *BluetoothSerial.h* that enables Bluetooth Classic that allow an ESP32 to communicate via Bluetooth. We create a variable named *SerialBT* to be used on other parts of the codes. We also declare pins 34 and 25 to an LM35 and LED respectively to help the code readability.

void setup() function:

Under this function, we open up serial communication to initialise the USB serial monitor for debugging and name the Bluetooth device as ESP32_temp. We also set the LED pin as an output and ensure that it is turned off in the beginning.

void loop() function:

Under this function is where we start calculating the temperature. The *lm35Pin* will receive an analogue voltage from the LM35. The value will then be converted into a digital value as voltage. After that, convert the voltage value into degrees (°C) by multiplying by 100. In the formula, we also added 15. This is because the LM35 is constantly printing a value of 0. Adding 15 is an offset added to simulate a minimum base temperature for testing and calibration. Then, we finally print the value in the app. We give a delay of 1s to avoid the data being jammed at once.

Under this function, we also set up communication that allows us to send commands to the ESP32 to turn on or off the LED using our phone. The logic starts with the ESP32 checking whether a Bluetooth command has been received. To turn on the LED, a command of '1' is sent. And to turn it off, we can send the command '0'

CONCLUSION

In this experiment, we successfully developed a basic wireless temperature monitoring system using the ESP32 microcontroller, LM35 temperature sensor, and LED. The system was capable of reading temperature data and transmitting it via Bluetooth for monitoring.

Additionally, a basic command system was implemented to control an LED remotely through Bluetooth. Despite some challenges with inaccurate temperature readings, we were able to mitigate this with a software offset. Overall, the experiment demonstrated the fundamental integration of sensors, microcontrollers, and wireless communication in a mechatronic system


RECOMMENDATIONS


An error occurred in our experiment. During our observations, we found out that the Bluetooth app that we are using printed a lot of 0°C. Hence, the reason why we added 15.0 in the formula to calculate the temperature as it serves as an offset to simulate a minimum base temperature for testing and calibration. The error might occur from a faulty temperature sensor that we are using. To avoid this error, components that are in good condition must be ensured to avoid bad observations that can affect any project that we are doing. Other than that, we can improve our temperature sensor. The temperature sensor that we are using in this experiment is LM35. Changing the sensor with other temperature sensors, such as DHT11 or DHT22, can ensure better temperature sensing accuracy. Using filters can also improve our readings, as filters can be used to attenuate any unwanted noise that can disturb our observations.


STUDENT'S DECLARATION

This is to certify that we are responsible for the work submitted in this report, that the original work is our own except as specified in the references and acknowledgement, and that the original work contained herein has not been taken or done by unspecified sources or persons. We hereby certify that only one individual has not done this report and that all of us have contributed to the report. The length of each individual's contribution to the reports is noted within this certificate.

We also hereby certify that we have read and understand the content of the total report, and no further improvement on the report is needed from any of the individual contributors to the report. We, therefore, agreed unanimously that this report should be submitted for marking, and this final printed report has been verified by us.

| | | |
|--|------------|---|
| NAME: Muhammad Ammar Zuhair Bin Nor Azman Shah | READ | ✓ |
| MATRIC NO: 2110259 | UNDERSTAND | ✓ |
| SIGNATURE:  | AGREE | ✓ |

| | | |
|--|------------|--|
| NAME: Muhammad Nuruddeen Bin Mohammad Nasir | READ | |
| MATRIC NO: 2225943 | UNDERSTAND | |
| SIGNATURE:  | AGREE | |

| | | |
|---|------------|--|
| NAME: MUHAMMAD AKMAL NIZHAM BIN AZLAN | READ | |
| MATRIC NO: 2213323 | UNDERSTAND | |
| SIGNATURE :  | AGREE | |