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

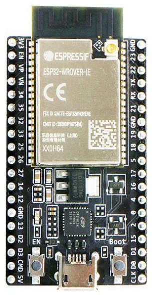
TABLE OF CONTENTS 1
INTRODUCTION 2
MATERIAL AND EQUIPMENT 3
EXPERIMENT SETUP 5
METHODOLOGY 6
RESULTS 7
DISCUSSION 8
CONCLUSION 9
RECOMMENDATIONS 10
STUDENT’S DECLARATION 11

INTRODUCTION

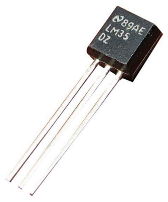
The objective of this experiment is to create a wireless temperature monitoring system using Wi-Fi, Arduino, and a temperature sensor or thermistor. The Arduino will read temperature data from the thermistor, send it to a Python script over Wi-Fi, and the Python script will display and log the temperature.

MATERIAL AND EQUIPMENT

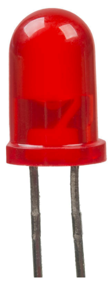
1. ESP32



2. Temperature sensor, LM35

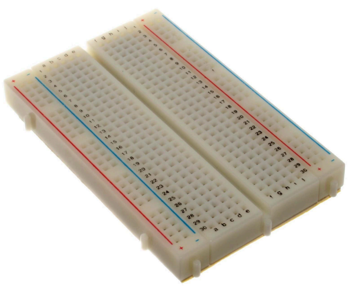


3. LED

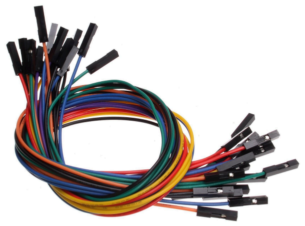


4. Smartphone with Bluetooth support

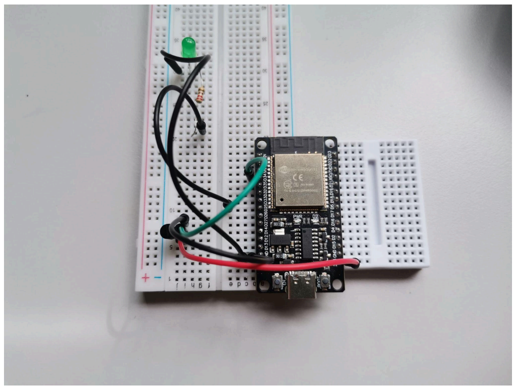
5. Breadboard



6. Jumper wires



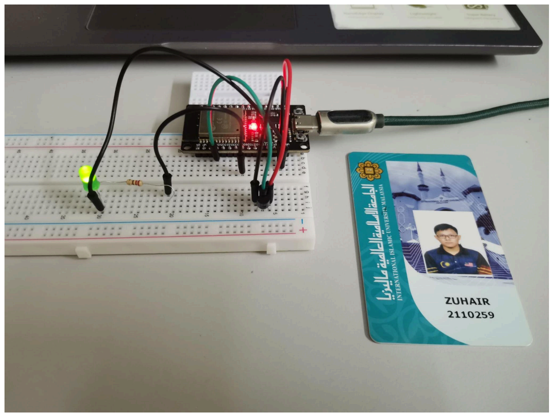
<u>EXPERIMENT SETUP</u>

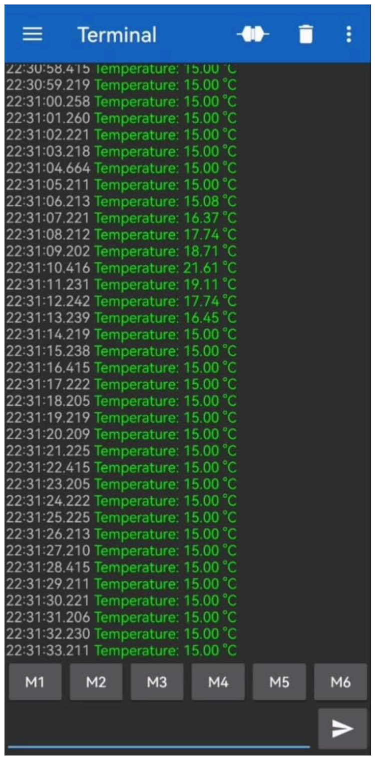


METHODOLOGY

1. Hardware Setup:
   - Connect the temperature sensor (thermistor) to the Arduino.
   - Connect the Bluetooth module to the Arduino.
   - Connect the Arduino to your Wi-Fi network using the built-in Wi-Fi capabilities.
2. Arduino Programming:
   - Write an Arduino sketch that reads temperature data from the sensor.
   - Set up Wi-Fi connectivity to send temperature data to a cloud service like ThingSpeak, where you can create a simple dashboard to visualize the data.
3. Bluetooth Programming:
   - Write an Arduino sketch to enable Bluetooth communication1
4. Remote Monitoring:
   - Access your ThingSpeak dashboard on your computer or smartphone to remotely monitor the temperature in real-time via the internet.

RESULTS





```
22:30:58.415 Temperature: 15.00 °C
22:30:59.219 Temperature: 15.00 °C
22:31:00.258 Temperature: 15.00 °C
22:31:01.260 Temperature: 15.00 °C
22:31:02.221 Temperature: 15.00 °C
22:31:03.218 Temperature: 15.00 °C
22:31:04.664 Temperature: 15.00 °C
22:31:05.211 Temperature: 15.00 °C
22:31:06.213 Temperature: 15.08 °C
22:31:07.221 Temperature: 16.37 °C
22:31:08.212 Temperature: 17.74 °C
22:31:09.202 Temperature: 18.71 °C
22:31:10.416 Temperature: 21.61 °C
22:31:11.231 Temperature: 19.11 °C
22:31:12.242 Temperature: 17.74 °C
22:31:13.239 Temperature: 16.45 °C
22:31:14.219 Temperature: 15.00 °C
22:31:15.238 Temperature: 15.00 °C
22:31:16.415 Temperature: 15.00 °C
22:31:17.222 Temperature: 15.00 °C
22:31:18.205 Temperature: 15.00 °C
22:31:19.219 Temperature: 15.00 °C
22:31:20.209 Temperature: 15.00 °C
22:31:21.225 Temperature: 15.00 °C
22:31:22.415 Temperature: 15.00 °C
22:31:23.205 Temperature: 15.00 °C
22:31:24.222 Temperature: 15.00 °C
22:31:25.225 Temperature: 15.00 °C
22:31:26.213 Temperature: 15.00 °C
22:31:27.210 Temperature: 15.00 °C
22:31:28.415 Temperature: 15.00 °C
22:31:29.211 Temperature: 15.00 °C
22:31:30.221 Temperature: 15.00 °C
22:31:31.206 Temperature: 15.00 °C
22:31:32.230 Temperature: 15.00 °C
22:31:33.211 Temperature: 15.00 °C
```

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 readibility.

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 Bluetooth communication. 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, using 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: [illegible]	AGREE	✔

NAME: Muhammad Nuruddeen Bin Mohammad Nasir	READ	✔
MATRIC NO: 2225943	UNDERSTAND	✔
SIGNATURE: [illegible]	AGREE	✔

NAME: MUHAMMAD AKMAL NIZHAM BIN AZLAN	READ	✔
MATRIC NO: 2213323	UNDERSTAND	✔
SIGNATURE : [illegible]	AGREE	✔