KULLIYYAH OF ENGINEERING (KOE)

MECHATRONICS SYSTEM INTEGRATION (MCTA3203)

SEMESTER 1, 24/25

SECTION 1

PROJECT REPORT WEEK 8

TITLE : BLUETOOTH

PREPARED BY :

NO	NAME	MATRIC NO
1	ARIF EMRULLAH BIN TAJUL ARIFFIN	2215359
2	AZLIYANA SYAHIRAH BINTI AZAHARI	2210620
3	DAMIA MAISARAH BINTI ZAWAWI	2217830
4	HUDA BINTI AB RAHMAN AL-QARI	2226676
5	IKMAL HAKIM BIN ZAKI	2125625

DATE OF SUBMISSION : 12TH DECEMBER 2024

ABSTRACT

This experiment explores wireless data interfacing between microcontrollers and computer-based systems using Bluetooth and Wi-Fi technologies. The focus is on designing a remote temperature monitoring and control system. By leveraging an Arduino microcontroller and a temperature sensor or thermistor, the system collects temperature data and transmits it via Wi-Fi to a Python-based script. This script processes, displays, and logs the data for real-time analysis, showcasing the effective integration of sensors, actuators, and wireless communication for environmental monitoring.

TABLE OF CONTENTS

INTRODUCTION

The advancement of wireless communication technologies such as Bluetooth and Wi-Fi has revolutionized data interfacing in microcontroller-based systems. This experiment aims to develop a remote temperature monitoring and control system, highlighting the seamless integration of sensors, actuators, and wireless data transmission. The system employs an Arduino microcontroller interfaced with a temperature sensor or thermistor to measure temperature data. The collected data is sent to a computer via Wi-Fi, where a Python script processes, displays, and logs the information. By demonstrating the practical application of wireless interfacing for real-time monitoring, this experiment emphasizes the efficiency and versatility of combining microcontroller systems with modern communication protocols.

MATERIALS AND EQUIPMENTS

1. Arduino Uno
2. Temperature sensor: LM35
3. Bluetooth module: HC-05
4. Smartphone with Bluetooth support
5. Power supply for the Arduino
6. Breadboard and jumper wires

EXPERIMENTS SETUP

1. Hardware Setup:
    - The temperature sensor (LM35) was connected to the Arduino.
    - The Bluetooth module was connected to the Arduino.
    - The Arduino was connected to the Wi-Fi network using its built-in Wi-Fi capabilities.
2. Arduino Programming:
    - An Arduino sketch was written to read temperature data from the sensor.
3. Bluetooth Programming:
    - An Arduino sketch was written to enable Bluetooth communication.
    - The corresponding tasks were completed.
4. Remote Monitoring:
    - The Arduino serial terminal was accessed on a smartphone to monitor the temperature in real-time via the bluetooth connection.

METHODOLOGY

CODING

```
#include <SoftwareSerial.h>

const int analogPin = A0;        // Analog pin connected to LM35
const int fanPin = 7;            // Pin connected to fan

SoftwareSerial BTSerial(10, 11); // RX, TX

void setup() {
  pinMode(fanPin, OUTPUT);

  Serial.begin(9600);            // Debugging via Serial Monitor
  BTSerial.begin(9600);          // Bluetooth module initialization

  BTSerial.println("Bluetooth Connected!"); // Send a welcome message
}

void loop() {
  // Read temperature data
  int sensorValue = analogRead(analogPin);
  double voltage = sensorValue * (5.0 / 1023.0);
  double temperature = voltage * 100.0;

  // Send temperature data over Bluetooth
  BTSerial.print("Temperature: ");
  BTSerial.print(temperature);
  BTSerial.println(" °C");

  // Check if a command is received from the app
```

```
  if (BTSerial.available()) {
    String command = BTSerial.readStringUntil('\n');
    command.trim(); // Remove any extra spaces or newlines

    if (command == "FAN ON") {
      digitalWrite(fanPin, HIGH);
      BTSerial.println("Fan is ON");
    } else if (command == "FAN OFF") {
      digitalWrite(fanPin, LOW);
      BTSerial.println("Fan is OFF");
    }  else if(temperature >= 30) {
      digitalWrite(fanPin, HIGH);
    }  else if (temperature <= 29) {
      digitalWrite(fanPin, LOW);
    } else {
      BTSerial.println("Unknown Command");
    }
  }

  delay(1000); // Delay for 1 second
}
```

PROCEDURES

1. The temperature sensor was placed in a room or area designated for temperature monitoring and control.
2. The Arduino was connected to a power source and ensured to be connected to the Wi-Fi network.
3. The room's temperature was monitored in real-time using the Arduino serial monitor in android.

RESULTS

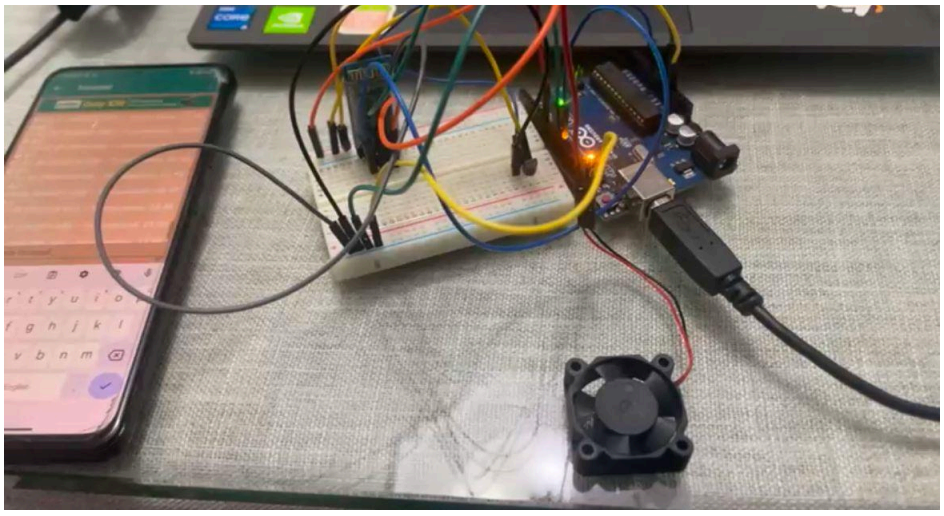


Figure 1.0 connection of bluetooth module (HC-05) connected to smartphone

The outcome of this experiment was the LM35 sensor continuously measured temperature and transmitted it over bluetooth and the data was displayed on the serial monitor on the android smartphone.

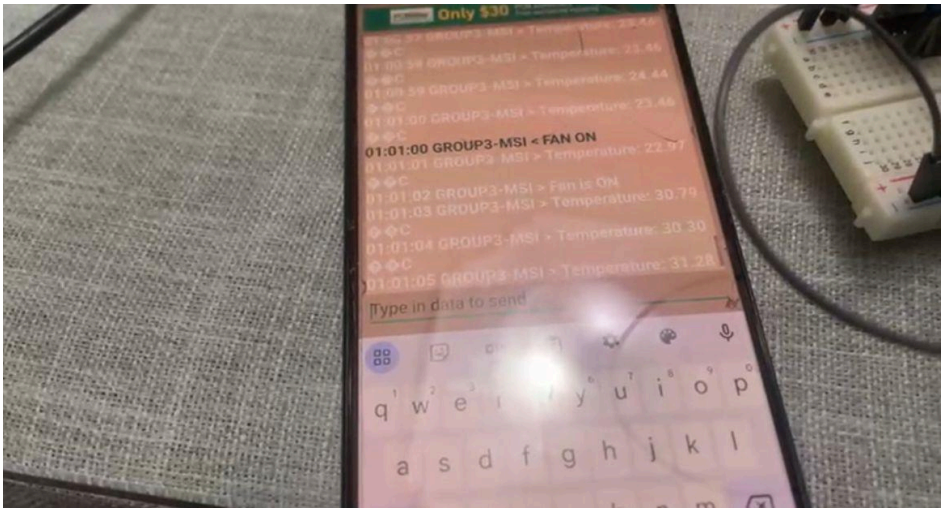


Figure 1.1 smartphone displayed data and received command

The bluetooth device on the mobile app ‘Adruino terminal app’ can send the following commands which are “FAN ON” to turn the fan on and “FAN OFF” to turn the fan off. The other feature was automatic temperature based fan control. When the temperature exceed 30 C the fan turns on automatically and when the temperature is 29 C or lower, the fan turns off automatically.

DISCUSSIONS

· **Software**

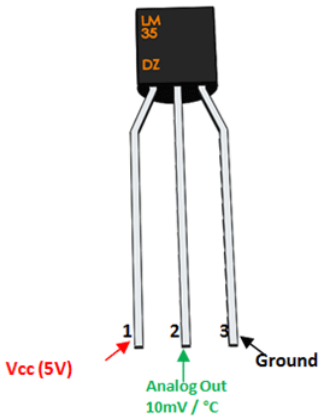
The Arduino serves as the primary DAQ hardware, programmed to interface with both Wi-Fi and Bluetooth for data transmission and control. The Wi-Fi module transmits temperature data from a thermistor to the ThingSpeak platform, where the data is visualized and analyzed through a cloud-based dashboard. Simultaneously, the HC-05 Bluetooth module enables local

communication, allowing commands to be sent from a smartphone application for controlling devices like fans or heaters. A Python script on the computer side reads serial data from the Arduino using the pyserial library, displaying real-time temperature updates and logging data for further analysis. This integration simplifies the process of remote temperature monitoring and control by combining wireless communication, data acquisition, and user-friendly visualization.

· **Electrical**

The circuit setup involves connecting the temperature sensor (DHT11 or DHT22) to the Arduino's analog input pins for data acquisition. The power (VCC) and ground pins of the sensor are connected to the Arduino's 5V and GND pins, respectively, to supply the required power. The HC-05 Bluetooth module is also connected to the Arduino via digital pins, enabling wireless communication. The Arduino’s built-in Wi-Fi module is configured to connect to a local Wi-Fi network, facilitating remote data transmission. Pull-up resistors are used where necessary to stabilize signal readings and ensure accurate temperature measurements. This configuration enables seamless data collection, processing, and transmission between the hardware components.

· **Hardware**

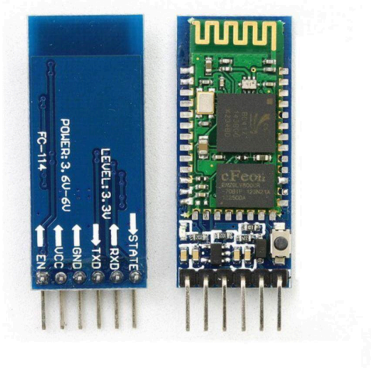


LM 35

- Temperature sensor that provides an analog output voltage directly proportional to the temperature in Celsius.

Fan

- 5V fan is a device that creates airflow using rotating blades.



HC-05 bluetooth module

- HC-05 bluetooth module is used to enable wireless communication between microcontrollers and devices like smartphones.

CONCLUSION

The remote temperature monitoring system effectively illustrate how wireless connection, real time data processing and arduino hardware based can all be combines. Although the system does a good job of carrying out its intended tasks, there is room for considerable improvement in terms of scalability and user experience.

RECOMMENDATIONS

Optimise data transfer to reduce latency and increase dependability which enhance the data transfer protocol between ESP32 and monitoring platform. Also, can expand functionality which

increases the system's monitoring capabilities, incorporating more sensors such for light and humidity.

ACKNOWLEDGEMENT

We would like to express our gratitude to Dr. Wahju Sediono, Dr. Ali Sophian, Dr. Zulkifli Bin Zainal Abidin, for providing the necessary resources and facilities to conduct this and support throughout the duration of the project. Additionally, we extend our appreciation to all individuals who contributed to the success of this lab report through their valuable insights and feedback.

STUDENT'S DECLARATION

Certificate of Originality and Authenticity

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 have not been untaken or done by unspecified sources or persons.

We hereby certify that this report has not been done by only one individual and all of us have contributed to the report. The length of contribution to the reports by each individual is noted within this certificate.

We also hereby certify that we have read and understand the content of the total report and that no further improvement on the reports is needed from any of the individual contributors to the report.

Signature: 	**Read**	**/**
Name: Arif Emrullah Bin Tajul Arifin	**Understand**	**/**
Matric No: 2215359	**Agree**	**/**

Signature: 	**Read**	/
Name: Azliyana Syahirah Binti Azahari	**Understand**	/
Matric No: 2210620	**Agree**	/

Signature: 	**Read**	/
Name: Damia Maisarah Binti Zawawi	**Understand**	/
Matric No: 2217830	**Agree**	/

Signature: 	**Read**	/
Name: Ikmal Hakim Bin Zaki	**Understand**	/
Matric No: 2125625	**Agree**	/

Signature: 	**Read**	/
Name: Huda binti Ab Rahman Al-Qari	**Understand**	/
Matric No: 2226676	**Agree**	/