A black background with white text

AI-generated content may be incorrect.

**LABORATORY REPORT**

**MECHATRONICS SYSTEM INTEGRATION MCTA 3203**

**SEMESTER 2 2023/2024**

**WEEK 8**

**SUBMISSION DATE:**

**GROUP: 6**

|  |  |  |
| --- | --- | --- |
|  | **NAME** | **MATRIC NUMBER** |
| 1. | MUHAMMAD NABIL IMAN BIN ABD RAHMAN | 2313551 |
| 2. | MUHAMMAD NAZHAN BIN MOHAMED NADZRI | 2313703 |
| 3. | MUHAMMAD HAZIQ AJMAL BIN MD KAMAL | 2312821 |

**Table of Contents**

[Abstract 3](#_Toc193058084)

[Introduction 3](#_Toc193058085)

[Materials and Equipment 3](#_Toc193058086)

[Methodology 5](#_Toc193058087)

[Question 6](#_Toc193058088)

[Discussion 6](#_Toc193058089)

[Conclusion 7](#_Toc193058090)

[Recommendations 7](#_Toc193058091)

[Appendix 7](#_Toc193058092)

[Acknowledgement 9](#_Toc193058093)

[Declaration 9](#_Toc193058094)

Abstract

This experiment focuses on wireless temperature monitoring and control using a microcontroller with integrated Wi-Fi and Bluetooth capabilities. By combining an Arduino-compatible board with a temperature sensor and a Bluetooth module, real-time data is transmitted and analyzed through the internet and smartphone applications. Data is logged to a cloud platform (ThingSpeak) via Wi-Fi and monitored remotely, while Bluetooth is used for user interaction to control actuators like fans or heaters. The study emphasizes integrating wireless communication into embedded systems for smart home or IoT applications.

Introduction

The advancement of wireless communication has significantly impacted embedded systems, especially in remote monitoring and control applications. This experiment introduces students to the fundamentals of wireless data interfacing using Bluetooth and Wi-Fi, enabling real-time data collection, logging, and control through smartphones or computers. The project involves an Arduino board equipped with Wi-Fi (e.g., ESP8266 or ESP32), a temperature sensor (like DHT11 or thermistor), and a Bluetooth module (e.g., HC-05). The goal is to develop a functional IoT-based system that reads environmental temperature data, logs it to an online dashboard (ThingSpeak), and allows actuator control via Bluetooth.

Materials and Equipments

1. Arduino board with Wi-Fi capability (e.g., Arduino ESP8266, Arduino MKR1000, or an ESP32)

2. Temperature sensor (e.g., DHT11 or DHT22)

3. Bluetooth module (e.g., HC-05 or HC-06)

4. Smartphone with Bluetooth support

5. Wi-Fi network and internet access

6. Power supply for the Arduino

7. Breadboard and jumper wires

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

.

• Complete the task below.

4. Remote Monitoring:

• Access your ThingSpeak dashboard on your computer or smartphone to remotely

monitor the temperature in real-time via the internet.

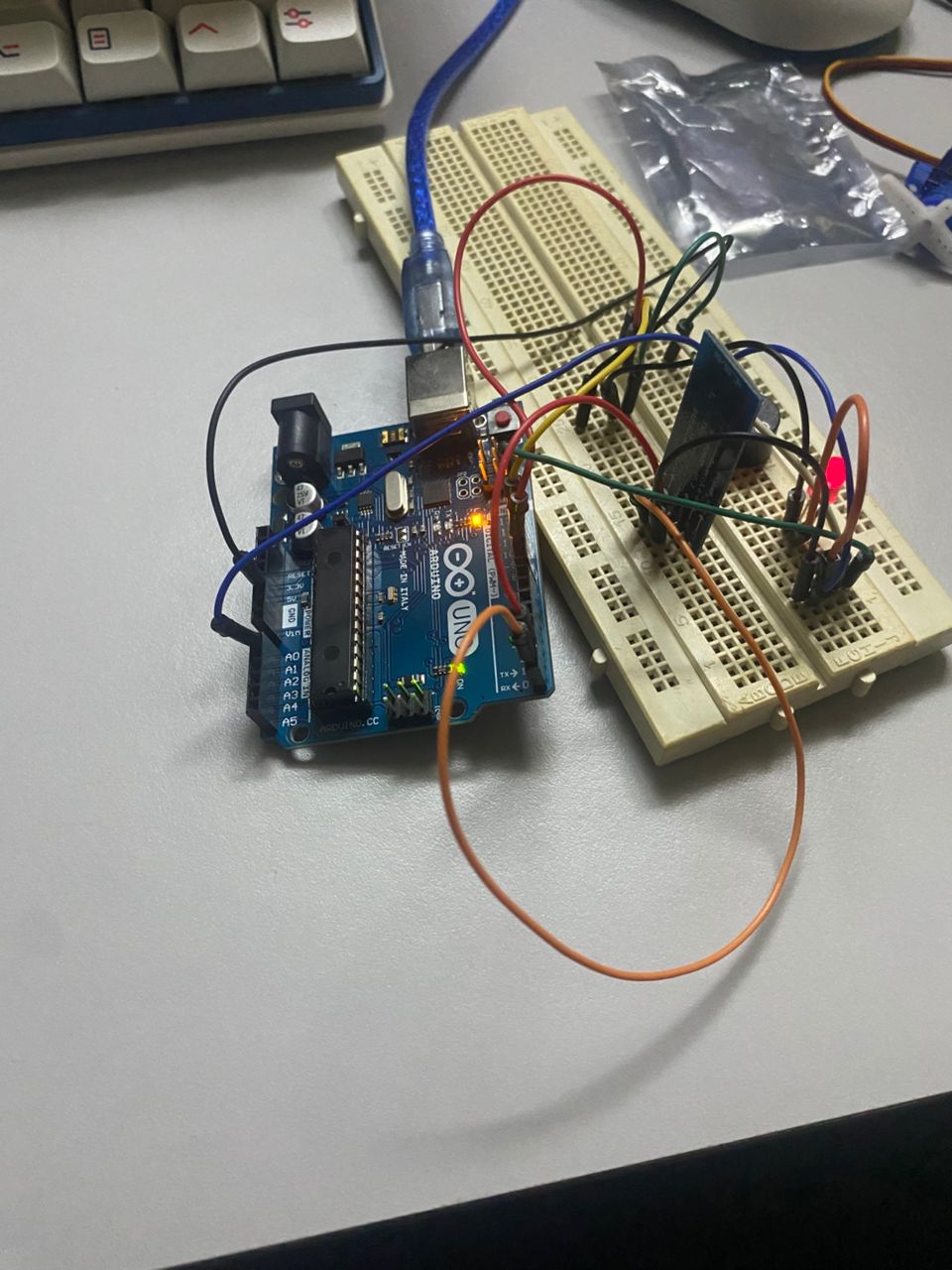
Task

Develop a simple smartphone application (or use an existing one) that communicates with the

Arduino via Bluetooth. This app should allow you to send commands to control a connected device,

like a fan or heater, based on the temperature data received from the Arduino.

Results



Discussion

The experiment is divided into two major parts: Wi-Fi-based remote monitoring and Bluetooth-based local control.

**Wi-Fi Data Transmission:**  
The Arduino is programmed to read temperature values from a thermistor. These readings are processed and sent to the ThingSpeak platform via the microcontroller's built-in Wi-Fi. ThingSpeak serves as a cloud-based dashboard that displays real-time temperature data and maintains a historical log, enabling users to observe environmental changes remotely from a computer or smartphone.

**Bluetooth Communication:**  
Simultaneously, the HC-05 Bluetooth module is connected to the Arduino using the SoftwareSerial library. A smartphone application (custom or third-party) communicates with the Arduino via Bluetooth, allowing users to send commands to actuators like fans or heaters based on the received temperature. This bidirectional communication loop ensures the system is not only monitoring but also capable of interactive control.

**Python Visualization:**  
An optional enhancement includes using a Python script (via pyserial) to receive data from Arduino over serial and visualize it using matplotlib. This highlights the integration of embedded systems with data science tools for a more analytical perspective.

**Observations:**

* Accurate temperature readings depend on the proper calibration of the thermistor and implementation of the Steinhart-Hart equation.
* Wi-Fi configuration and ThingSpeak integration require correct API key setup and stable connectivity.
* Bluetooth pairing requires precise baud rate configuration and mobile app interface handling.

Conclusion

This experiment effectively demonstrates a hybrid wireless monitoring and control system by utilizing both Wi-Fi and Bluetooth technologies. Students gain practical exposure to IoT concepts, sensor data acquisition, wireless data logging, and actuator control. The real-time capabilities and cloud integration make this setup suitable for modern applications such as smart home systems, environmental monitoring, and automation projects. The hands-on experience strengthens the understanding of integrating communication protocols with microcontroller-based systems.

Recommendations

1. **Implement Temperature Threshold Logic for Automation**  
   To enhance functionality, include logic in the Arduino code that automatically turns the fan or heater ON/OFF when temperature crosses user-defined thresholds. This would transform the system from passive monitoring to active environmental control.
2. **Integrate Mobile App Feedback Display**  
   Modify the mobile application to display current temperature readings alongside control buttons. This would provide real-time feedback to the user and make the interface more informative and user-friendly.

Acknowledgements

We would like to express my sincere gratitude to Dr Wahju Sediono, Dr Zulkifli Bin Zainal Abidin and the lab technician for their invaluable guidance, support, and encouragement throughout this project. Their expertise and insights have been instrumental in shaping the direction of this work. I would also like to extend my thanks to my fellow peers for their assistance and collaboration, which greatly contributed to the successful completion of this project.

Declaration

We hereby declare that the work presented in this report is entirely my own, except where otherwise acknowledged. I affirm that I have adhered to the principles of academic integrity and have not engaged in any form of plagiarism or unethical conduct in the completion of this project. All sources of information and assistance used in this work have been properly cited and acknowledged.