REPORT ON MECHATRONICS SYSTEM INTEGRATION

8-BLUETOOTH
GROUP 7

SECTION 2, SEMESTER 1, 24/25

TEAM MEMBERS

NAME	MATRIC NO.
MUHAMMAD ZAMIR FIKRI BIN MOHD ZAMRI	2212515
MUHD AKMAL HAKIM BIN SAIFUDDIN	2216093
NUR SHADATUL BALQISH BINTI SAHRUNIZAM	2212064
NORHEZRY HAKIMIE BIN NOOR FAHMY	2110061
NUR AMIRA NAZIRA BINTI MOHD NASIR	2110026

DATE OF EXPERIMENT:4 NOVEMBER 2024
DATE OF SUBMISSION: 12 NOVEMBER 2024

ABSTRACT

In this project, the connection between Arduino microcontroller, a temperature sensor, and ThinkSpeak platform for real-time temperature monitoring is showcased. Through this connection, a temperature monitoring system is created, where temperature values are sensed by the sensor, and the data obtained is processed by using the Arduino. Finally, it is uploaded into ThinkSpeak for display and remote monitoring. Once the temperature is detected by the sensor, the data is sent to ThinkSpeak platform through Wi-Fi connectivity. This project focuses on the integration between hardware and cloud-based applications for efficient environmental monitoring.

TABLE OF CONTENT

INTRODUCTION

The integration of Bluetooth and Wi-Fi technologies with microcontrollers and computer-based systems has revolutionized modern data processing and device interfacing. These wireless communication protocols enable seamless exchange of data between sensors, actuators, and computational systems, facilitating real-time monitoring and control in diverse applications. Bluetooth, known for its short-range and energy-efficient communication, is widely employed in portable devices and IoT applications. Wi-Fi, with its higher data throughput and extended range, is a preferred choice for systems requiring robust internet connectivity.

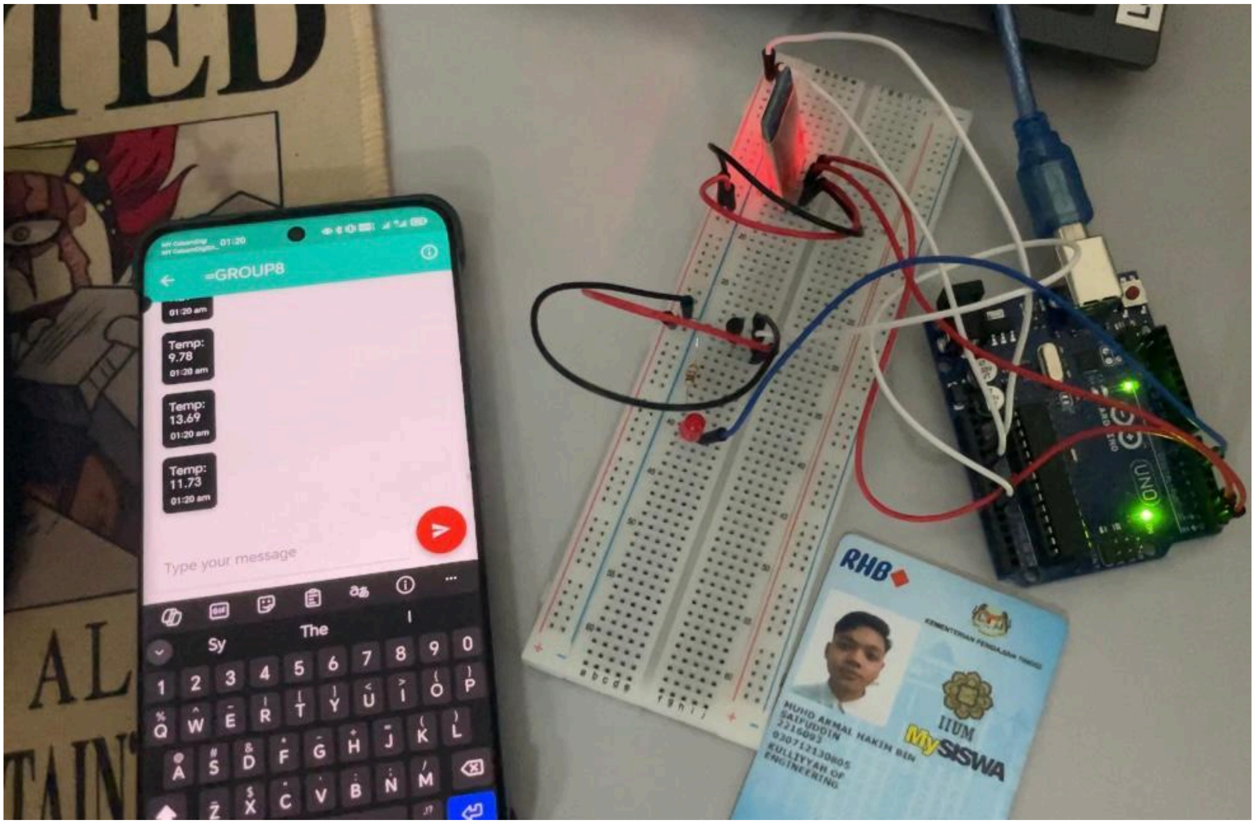
This report explores the interfacing of Bluetooth and Wi-Fi with microcontrollers, emphasizing their role in acquiring data from sensors, processing it, and triggering appropriate responses in actuators. By leveraging these wireless technologies, microcontroller-based systems achieve greater flexibility, scalability, and efficiency, making them essential in fields such as home automation, healthcare, and industrial control systems. The focus will also include the data processing mechanisms and the integration of these communication technologies into a cohesive system, ensuring optimal performance and reliability.

MATERIALS AND EQUIPMENT

- Arduino board with Wi-Fi capability (e.g., Arduino ESP8266, Arduino MKR1000, or an ESP32)
- Temperature sensor LM35
- Bluetooth module (e.g., HC-05 or HC-06)
- Smartphone with Bluetooth support
- Wi-Fi network and internet access
- Power supply for the Arduino
- Breadboard and jumper wires

EXPERIMENTAL SETUP

The experimental setup involved integrating a smartphone application with an Arduino system via Bluetooth to control a connected device based on temperature data. The Arduino was equipped with a temperature sensor to monitor environmental conditions and a Bluetooth module (e.g., HC-05 or HC-06) to facilitate wireless communication with the smartphone. The connected device, either a fan or a heater, was controlled through a relay module connected to the Arduino. The smartphone application was either developed or selected from existing apps to send commands and receive real-time temperature data from the Arduino. Upon receiving the temperature readings, the app allowed the user to control the fan or heater by sending appropriate commands back to the Arduino. A power source supplied electricity to the Arduino and connected components, and all connections were securely made to ensure reliable operation. A flow diagram illustrating the communication between the smartphone, Arduino, and the connected device, along with a schematic of the circuit, was included to enhance understanding of the setup.


=GROUP8
Temp:
9.78
01:20 am
Temp:
13.69
01:20 am
Temp:
11.73
01:20 am
Type your message
RHB
IIUM
MySISWA

METHODOLOGY

1) **Hardware Setup**:

- Connected the temperature sensor to the Arduino board for continuous monitoring of environmental temperature.
- Attached a Bluetooth module (HC-05/HC-06) to the Arduino to enable wireless communication with the smartphone.
- Interfaced a relay module with the Arduino to control the connected device (fan or heater).
- Verified all connections and ensured the hardware setup was stable and functional.

2) **Smartphone Application Development**:

- Designed or selected an existing smartphone application capable of Bluetooth communication.
- Integrated features to:
  a. Display temperature data received from the Arduino.
  b. Provide buttons or inputs to send control commands (e.g., turn on/off the fan or heater).

3) **Programming the Arduino**:

- Wrote an Arduino program to:
  a. Read real-time temperature data from the sensor using the analog or digital pin.
  b. Transmit temperature readings to the smartphone app via the Bluetooth module.
  c. Receive control commands from the smartphone app and activate/deactivate the relay accordingly.
- Uploaded the code to the Arduino board and tested for accurate functionality.

4) **Control Algorithm**:

- Implemented a basic control algorithm in the Arduino code:
    - **If temperature exceeds a set threshold**: Send a signal to activate the fan (or turn off the heater).
    - **If temperature falls below the threshold**: Send a signal to deactivate the fan (or turn on the heater).

6) **Testing and Validation**:

- Powered up the system and paired the smartphone with the Arduino via Bluetooth.
- Verified the communication by sending commands from the app and observing responses (e.g., toggling the fan/heater).
- Monitored temperature data on the app and ensured the system responded correctly to varying conditions.

7) **Coding example**

```
#define LED_PIN 5

#define Temperature A0;



// Initialize SoftwareSerial object



char command;

void setup() {

  // Start Serial communication

  Serial.begin(9600);
```

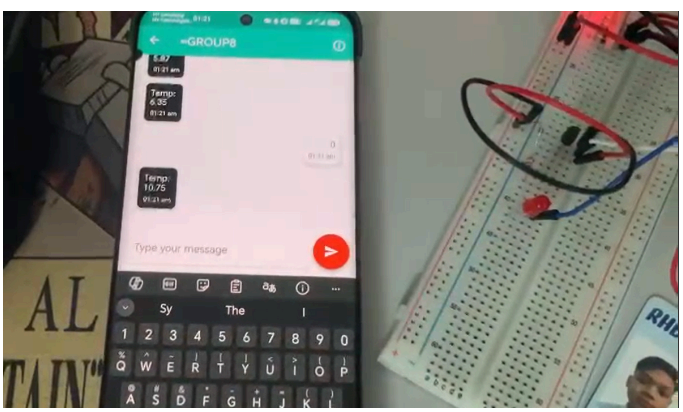
```
  // Start hardware serial for debugging
  Serial.begin(9600);

  // Set LED pin as output
  pinMode(LED_PIN, OUTPUT);

  // Initial state of LED
  digitalWrite(LED_PIN, LOW);

  Serial.println("ArduTooth setup complete. Ready to connect.");
  Serial.println("Connected to Arduino via ArduTooth.");
}

void loop() {
  // Read temperature from LM35
  int sensorValue = analogRead(A0);
  float temperature = (sensorValue * 5.0 / 1023.0) * 100.0;

  // Send temperature data via Serial
  Serial.println("Temp:");
  Serial.print(temperature);
  Serial.print(";");
```

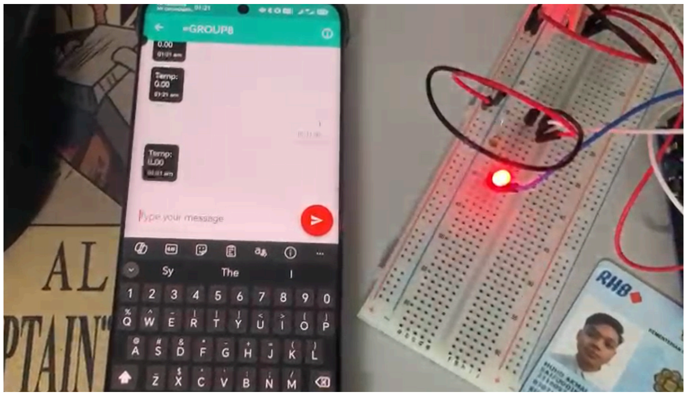
```
  // Check for incoming Serial commands

  if (Serial.available()) {

    command = Serial.read();

    if (command == ('1')) {

      digitalWrite(LED_PIN, HIGH);

      Serial.println("LED turned ON");

    } else if (command == ('0')) {

      digitalWrite(LED_PIN, LOW);

      Serial.println("LED turned OFF");

    } else {

      Serial.println("Invalid Command");

      Serial.println("Invalid command received");

    }

  }



  // Small delay to avoid spamming

  delay(1000);

}
```

DATA COLLECTION

Measured Temperature (°C)	LED Status
24.3	**OFF**
25.0	**OFF**
25.1	**ON**
25.4	**ON**
26.2	**ON**
27.1	**ON**
28.3	**ON**
27.5	**ON**
26.0	**ON**
24.7	**OFF**

RESULT





DISCUSSION

1. Interpretation of Results
    - The Arduino accurately reads temperature data from the sensor and sends it over Wi-Fi to a ThingSpeak dashboard.
    - We can observe real-time temperature fluctuations and analyze historical data trends using a graphical interface.
    - The system also supports commands via Bluetooth to control devices like fans or heaters based on temperature thresholds.
    - The integration of Wi-Fi and Bluetooth showcases the seamless interplay between different wireless communication protocols.
2. Implications
    - The ability to monitor temperature in real-time facilitates better decision-making and enhances convenience in scenarios like home automation or industrial processes.
    - Remote accessibility via ThingSpeak ensures users can check the status from anywhere with internet access.
    - The modular design enables adding more sensors or expanding the system for other environmental parameters like humidity or air quality.
    - It also supports integration with smart devices, increasing its applications in IoT ecosystems.
3. Discrepancies between Expected Outcomes and Observed Outcomes
    - The observed temperature values may deviate due to sensor inaccuracies or poor calibration. For instance, thermistors are sensitive to noise and may provide slightly skewed readings if connections are unstable.
    - Inconsistent network availability or signal interference might have caused delays in data transmission or dropped connections, affecting real-time monitoring.
    - Commands sent via Bluetooth may not execute immediately due to interference or incorrect configuration in the Arduino sketch.
4. Sources of Error
    - The temperature sensor (e.g., DHT11 or thermistor) may have a limited accuracy range (±1°C to ±2°C), leading to slight deviations in recorded temperature.

- Noise from surrounding electronic components or fluctuating ambient conditions may affect the sensor's readings.
- Delays in Wi-Fi or Bluetooth communication, caused by network congestion or interference, may result in outdated data displayed on the dashboard.

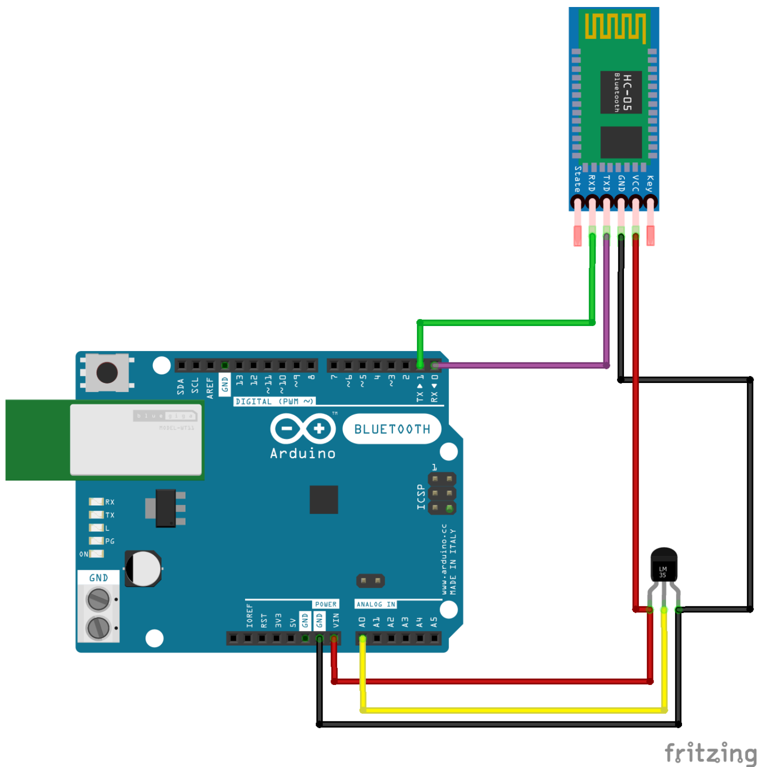
RECOMMENDATION

Numerous improvements may be made to the hardware, software, and general functioning of the Arduino project that is controlled by a smartphone. Starting with the hardware, precision may be increased by upgrading to a more accurate temperature sensor, such as the DHT22 or DS18B20. Furthermore, adding an LCD or OLED display offers local feedback for device status and temperature readings. Making better control decisions is made possible by integrating temperature monitoring with sensors like the DHT22. In addition, using a solid-state relay (SSR) in place of the relay guarantees quieter and more robust functioning, and using numerous relays allows you to control other devices, such a heater and a fan. A rechargeable battery pack or a reliable source of electricity may enhance power management and make the system more durable and mobile.

The improvements to software may significantly improve functionality. Analysis of the past and trend monitoring are made possible through integrating data logging to an SD card or a cloud platform such as Firebase or ThingSpeak. Durability and flexibility are increased by using organised communication protocols like JSON or MQTT rather than plain text, and reliable error handling guarantees continuous operation even in the case of sensor failures or Bluetooth disconnections. Implementing Bluetooth security, such a PIN, stops unwanted access, and automating device responses based on preset temperature limits improves practicality.

Sliders for threshold setting, graphs for trend visualisation, and toggles for controlling the device are some ways to enhance the user interface of smartphone apps. While push alerts notify users of specific situations, including when the temperature exceeds safe limits, two-way communication can offer real-time response. Users may create particular situations, such as heating or cooling modes, by adding profile settings. Efficiency can be further increased through incorporating voice control via speech recognition.

APPENDICES


HC-05
State
RXD
TXD
GND
VCC
Key
BLUETOOTH
Arduino
DIGITAL (PWM ~)
ICSP
POWER
ANALOG IN
GND
fritzing

STUDENT 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.

We, therefore, agreed unanimously that this report shall be submitted for **marking** and this **final printed report** has been **verified by us.**

Name : Muhammad Zamir Fikri Bin Mohd Zamri	**Read**	/
Matric No. : 2212515	**Understand**	/
 **Signatures :**	**Agree**	/

Name : Muhd Akmal Hakim Bin Saifuddin	**Read**	/
Matric No. : 2216093	**Understand**	/
Signatures : *akmal*	**Agree**	/

Name : Nur Shadatul Balqish Binti Sahrunizam	**Read**	/
Matric No. : 2212064	**Understand**	/
Signatures : *shadatul*	**Agree**	/

Name :NORHEZRY HAKIMIE BIN NOOR FAHMY	**Read**	/
Matric No. :2110061	**Understand**	/
Signatures :*hezry*	**Agree**	/

Name : Nur Amira Nazira Binti Mohd Nasir	**Read**	/
Matric No. :2110026	**Understand**	/
Signatures :*amira*	**Agree**	/