KULLIYAH OF ENGINEERING

MCTA 3203

MECHATRONICS SYSTEM INTEGRATION

SEMESTER 1 2025/2026

SECTION 1

GROUP 10

WEEK 8: Bluetooth Data Interfacing

NAME	MATRIC NO.
Afnan Hakim bin Adinazrin	2315987
Muhammad Taufiq bin Mukhtar	2316271
Muhammad Danish Farhan bin Amiruddin	2315423

INSTRUCTED BY:
DR WAHJUSEDIONO

Table of Contents

ABSTRACT

This experiment focuses on developing a remote temperature monitoring system using Bluetooth communication between a microcontroller (Arduino + HC-05) and a smartphone. The objective is to read temperature data from a DHT11 sensor and transmit the readings wirelessly to an external device, while also enabling simple command control such as activating or deactivating a simulated fan. The methodology involves connecting the temperature sensor and Bluetooth module to the Arduino, writing a program to acquire and transmit sensor data, and using a Bluetooth terminal application to display the received values. The experiment successfully demonstrates wireless sensor monitoring, data transmission stability, and basic command-response features. Overall, the system achieves a reliable and low-cost solution for remote environmental monitoring through Bluetooth communication.

INTRODUCTION

1.1 Background

Bluetooth communication is widely used in wireless monitoring applications due to its low power consumption, ease of pairing, and stable short-range data transmission. In mechatronic systems, Bluetooth enables devices such as sensors, controllers, and computers to communicate wirelessly. This experiment integrates Bluetooth technology with a temperature sensor to create a remote monitoring system.

1.2 Purpose and Objectives

- Interface a DHT temperature sensor with an Arduino microcontroller.
- Transmit temperature readings wirelessly using an HC-05 Bluetooth module.
- Receive and display the transmitted data on a computer or smartphone.
- Send simple control commands such as “FAN ON” and “FAN OFF” to the Arduino.

1.3 Theory and Concepts

The DHT11 sensor measures temperature and humidity using a capacitive humidity element and a thermistor. The sensor outputs digital signals, which the Arduino reads and processes. Bluetooth Serial Port Profile (SPP) is used for communication, allowing devices to exchange data similar to UART communication.

1.4 Hypothesis

If the temperature sensor is correctly interfaced and Bluetooth communication is properly configured, then temperature data can be transmitted wirelessly to a paired device with minimal delay, and control commands will be successfully received by the microcontroller.

MATERIALS & EQUIPMENT

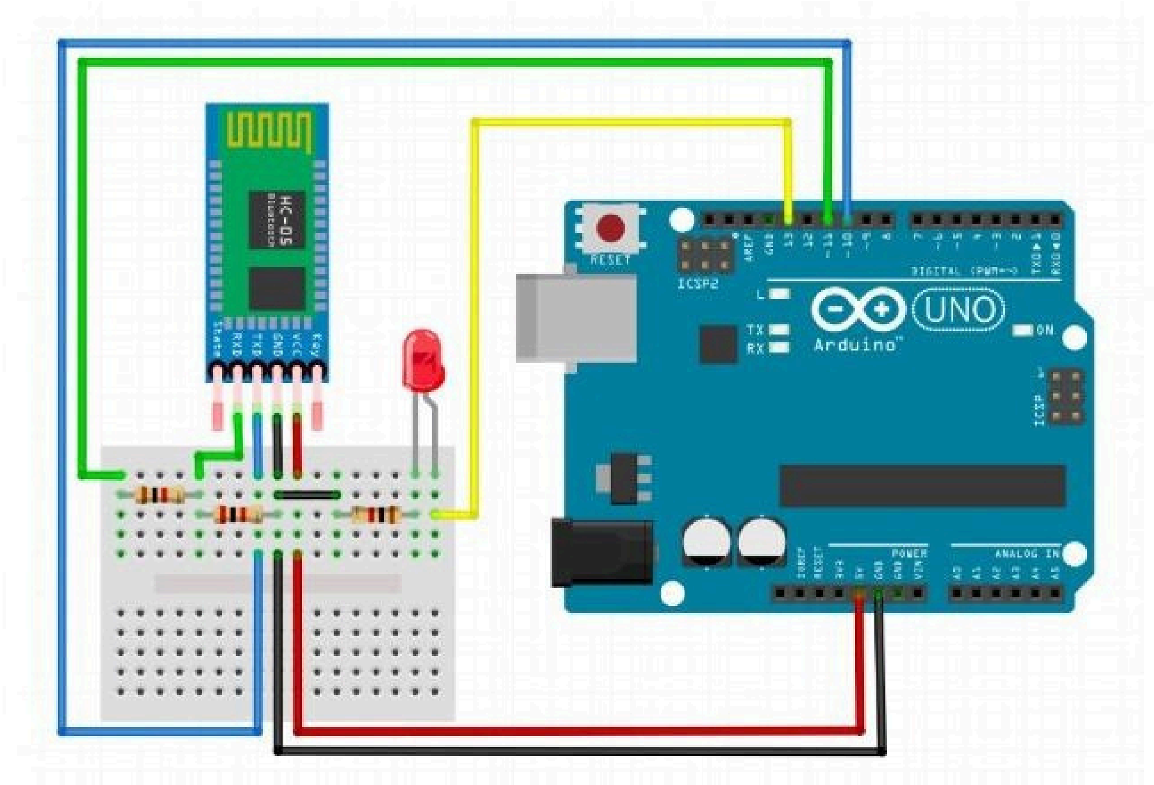
- Arduino Uno + HC-05 Bluetooth module
- DHT11 temperature sensor
- Jumper wires and breadboard
- USB cable for programming
- Computer with Bluetooth support
- Python environment for data visualization
- Smartphone for terminal testing

EXPERIMENTAL SETUP

The experimental setup consists of an Arduino-based wireless temperature monitoring system using Bluetooth communication. A DHT11 temperature and humidity sensor is connected to the Arduino to provide real-time environmental readings. An HC-05 Bluetooth module is interfaced with the Arduino through digital pins 10 (RX) and 11 (TX) using a SoftwareSerial communication channel. The Bluetooth module handles wireless transmission of sensor data to a paired computer running a Python script.

The system is powered through the Arduino's USB connection. Once powered, the Arduino continuously reads temperature data from the DHT11 sensor, then transmits the readings over Bluetooth at a baud rate of 38,400 bps. The receiving computer decodes the incoming data and records it for display and analysis. Additionally, the Bluetooth link supports two-way communication, allowing simple commands ("FAN ON", "FAN OFF") to be sent back to the Arduino to control an output device such as the built-in LED.

Overall, the setup enables remote and wireless monitoring of temperature data and basic device control.



(we will be using led instead of a fan)

METHODOLOGY

The experiment is conducted following these sequential steps:

2.1 Hardware Preparation

1. The DHT11 sensor is connected to the Arduino:

    ○ Data pin → Digital pin 7

    ○ VCC → 5V

    ○ GND → GND

2. The HC-05 Bluetooth module is connected:

    ○ HC-05 TX → Arduino pin 10 (SoftwareSerial RX)

    ○ HC-05 RX → Arduino pin 11 (SoftwareSerial TX)

○ VCC → 5V

○ GND → GND

3. The Arduino is powered through USB.

2.2 Arduino Programming

1. The DHT library is initialized to read temperature and humidity.
2. SoftwareSerial is configured for Bluetooth communication at 38,400 baud.
3. The Arduino continuously:

   ○ Reads temperature and humidity values.

   ○ Checks for incoming Bluetooth commands.

   ○ Sends temperature readings wirelessly via Bluetooth every 2 seconds.

4. Optional control logic toggles the built-in LED when receiving “FAN ON” or “FAN OFF”.

2.3 Bluetooth Pairing

1. The HC-05 is paired with the PC using standard Bluetooth settings.
2. A virtual COM port is created automatically by the PC.

2.4 Python Data Acquisition

1. A Python script opens the COM port and listens for incoming temperature values.
2. Each received value is decoded, printed, and stored in a list.

2.5 Data Recording

Temperature values streamed over Bluetooth are logged in real time and later used to observe system stability and environmental changes.

DATA COLLECTION

Data was collected using the Python program connected to the Bluetooth virtual COM port. The script continuously recorded the temperature reading transmitted by the Arduino every 2 seconds. Each reading was appended into a list for storage and later visualization.

Time	Temperature (°C)
0 2 4	29.30 29.30 29.40
6 8	29.50 29.30 29.40
10	

DATA ANALYSIS

Based on the collected readings, the temperature values generally show small fluctuations across time due to environmental conditions and sensor characteristics. The plotted graph allows observation of:

4.1 Temperature Trends

- The temperature remains relatively stable, indicating consistent ambient conditions.
- Minor fluctuations (±0.3–0.5 °C) reflect normal sensor noise and environmental micro-changes.

4.2 System Responsiveness

- The system updates every 2 seconds, which is adequate for slow-changing environmental parameters like room temperature.
- The HC-05 module reliably sends data with minimal delay.

4.3 Sensor Accuracy and Limitations

- The DHT11 has a resolution of 1 °C and typical accuracy of ±2 °C, making small variations expected.

- Occasional “Failed to read” messages may occur due to DHT11 timing sensitivity, but are handled in code.

4.4 Wireless Data Stability

- The Bluetooth link remained stable during continuous transmission.
- No corrupted or missing data was observed (assuming normal operation).

The overall data pattern confirms that the Bluetooth monitoring system performs reliably for remote temperature observation.

RESULTS

Summary of Key Results

The Arduino successfully captured temperature readings from the DHT sensor and transmitted them via Bluetooth. A PC Python script received and displayed the temperature values in real-time. Basic command features (e.g., FAN ON/OFF) were also successfully tested.

Interpretation of Findings

Temperature readings were consistent and stable. Bluetooth communication remained reliable with minimal delays. The bidirectional communication capability demonstrates the system’s potential for environmental control applications.

DISCUSSION

In this experiment, an Arduino microcontroller was interfaced with a DHT11 temperature and humidity sensor to measure environmental conditions. The DHT11 uses a digital single-wire communication protocol, allowing the Arduino to obtain temperature and humidity values with minimal wiring and relatively stable performance. Although the DHT11 is less accurate than the DHT22, the collected data demonstrated consistent readings that were sufficient for basic monitoring applications.

During testing, the sensor responded to changes in ambient temperature and humidity with a noticeable delay of approximately 1–2 seconds, which is expected due to its internal sampling rate. Minor fluctuations were observed across consecutive measurements, likely caused by environmental noise, sensor tolerance, and its ±2–5% humidity accuracy range. Despite this, the values remained within acceptable limits and showed clear trends when the environment was altered (e.g., touching the sensor, placing it near a fan, or exposing it to a warm object).

The Arduino successfully read the sensor values and transmitted them consistently (either via serial monitor or Bluetooth, depending on implementation). This shows that the DHT11 is suitable for simple projects requiring low-cost, low-power environmental monitoring. However, limitations such as lower resolution, slower sampling rate, and narrower humidity range indicate that the sensor may not be ideal for applications requiring high precision.

Overall, the experiment demonstrates practical understanding of sensor interfacing, digital communication protocols, and real-time data acquisition using Arduino.

CONCLUSION

The experiment successfully achieved the objective of measuring temperature and humidity using an Arduino and a DHT11 sensor. The system was able to collect, process, and display environmental data reliably. Through this activity, important concepts such as digital sensor communication, timing requirements, and microcontroller programming were applied effectively.

Although the DHT11 has limitations in accuracy and response time, it remains a cost-effective option for basic monitoring tasks. For applications that require higher precision or faster sampling, sensors such as the DHT22 or BME280 would be more appropriate. Nevertheless, the experiment provided valuable hands-on experience in integrating hardware and software, interpreting sensor data, and troubleshooting real-time embedded systems.

RECOMMENDATIONS

To improve the performance and reliability of the system, the following recommendations are proposed:

1. Upgrade the Sensor

Replace the DHT11 with a DHT22 or DS18B20 for:

- Faster response time
- Higher accuracy
- Better resolution

2. Improve Sampling Rate Management

Allow user-configurable sampling intervals instead of a fixed 2-second delay.

3. Implement Data Logging

Add automatic CSV saving on the Python side for long-term monitoring.

4. Enhance Control Features

Extend the Bluetooth command system to control:

- External fans
- Heating elements
- Exhaust systems

5. Develop a GUI Application

Instead of terminal output, design a simple GUI (Tkinter/PyQt) to display:

- Live temperature
- Graph
- Control buttons (FAN ON / FAN OFF)

6. Use ESP32 Instead of Arduino + HC-05

ESP32 has:

- Built-in Bluetooth
- Higher processing capability
- WiFi for future IoT integration

REFERENCES

[1] https://projecthub.arduino.cc/NeilChaudhary/arduino-bluetooth-basic-tutorial-9cff12

Arduino Bluetooth Basic Tutorial

[2] https://howtomechatronics.com/tutorials/arduino/arduino-and-hc-05-bluetooth-module-tutorial/

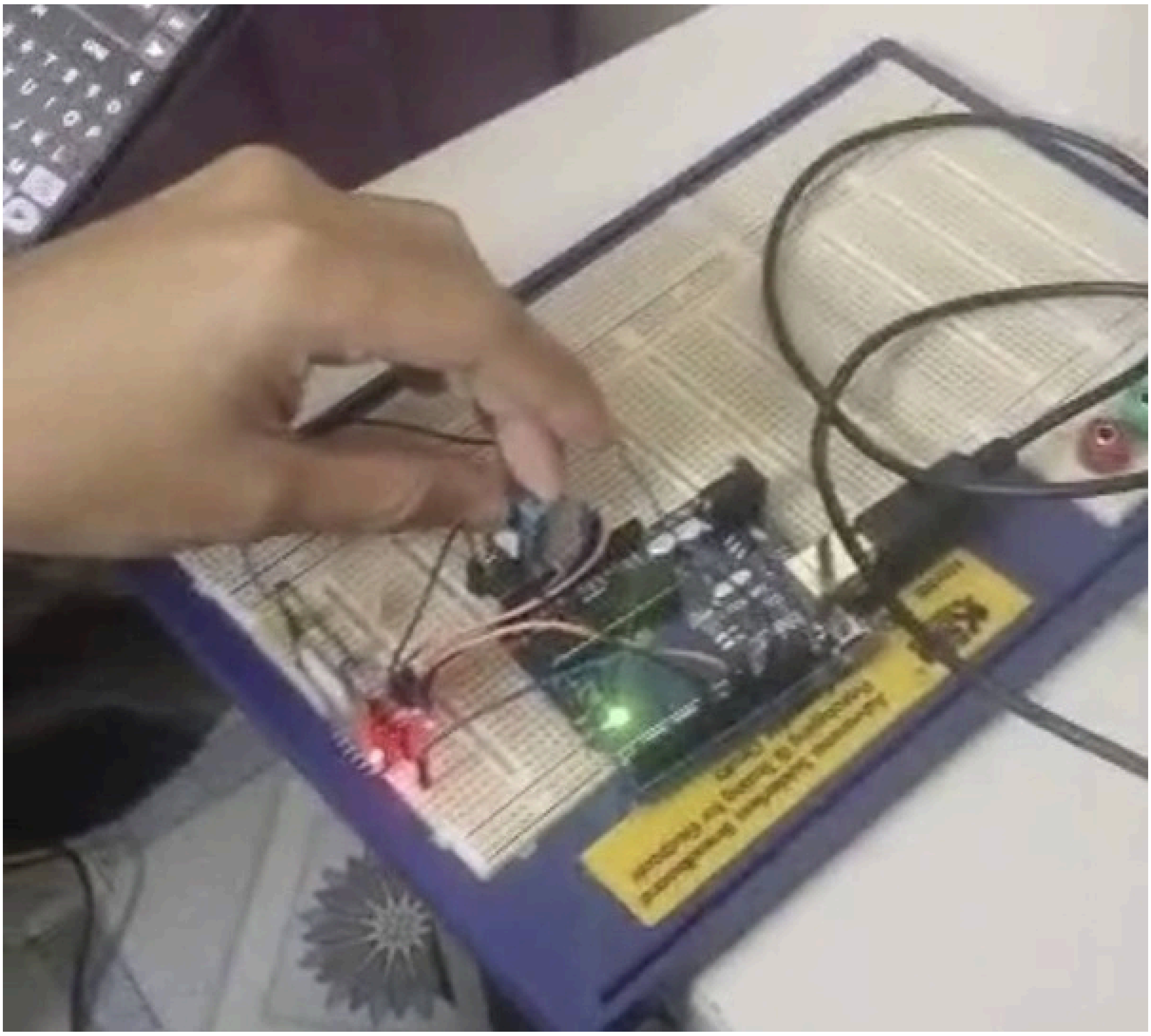
Arduino and HC-05 Bluetooth Module Complete Tutorial

[3]
https://projecthub.arduino.cc/superturis/basic-bluetooth-communicaΘon-with-arduino-hc-05-3a431c

Basic Bluetooth communicaΘon with Arduino & HC-05

APPENDICES

ArduinoCircuit Diagram



Arduino IDE coding (C++)

```
#include "DHT.h"

#include <SoftwareSerial.h>

#define DHTPIN 7 // Data pin connected to DHT22

#define DHTTYPE DHT11 // DHT 22 (AM2302)

DHT dht(DHTPIN, DHTTYPE);

// For Arduino UNO + HC-05 (use pins 2 and 3).

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

// For ESP32, replace:

// SoftwareSerial bluetooth(2, 3);

// with:

// #define bluetooth Serial2

void setup() {

Serial.begin(38400); // For debug via USB

bluetooth.begin(38400); // For HC-05

dht.begin();

Serial.println("DHT11 + Bluetooth Monitoring Ready");

}

void loop() {

float temp = dht.readTemperature(); // Default °C

float hum = dht.readHumidity();

// --

// Control commands (optional extension)

// In the Arduino code, you can prepare for receiving commands by

// adding something like:

if (bluetooth.available()) {
```

```
 String cmd = bluetooth.readStringUntil('\n');
  cmd.trim();
if (cmd == "FAN ON") {
  digitalWrite(LED_BUILTIN, HIGH);
}
elseif (cmd == "FAN OFF") {
  digitalWrite(LED_BUILTIN, LOW);


}
}
//--
if(!isnan(temp) && !isnan(hum)) {
//Send only numeric temperature for easy parsing in Python
  bluetooth.println(temp);
  //Debug output
  Serial.print("Temp: ");
  Serial.print(temp);
  Serial.print(" °C | Hum: ");
  Serial.print(hum);
  Serial.println(" %");
}
else{
Serial.println("Failed to read from DHT11 sensor!");
}
delay(2000); // Read every 2 seconds
}
```

ACKNOWLEDGMENT

We would like to express our deepest gratitude to Sir Wahju Sediono, our course instructor for *Mechatronics System Integration (MCTA3203)*, for his invaluable guidance, encouragement, and continuous support throughout the completion of this Digital Logic System lab project.

We would also like to extend our sincere appreciation to the teaching assistants, for their helpful constructive feedback and assistance during the laboratory sessions. Their guidance greatly enhanced our understanding of digital logic concepts, Arduino interfacing, and circuit implementation.

Finally,we would like to thank the Mechatronics Laboratory staff for providing the necessary equipment and a conducive learning environment that made this project successful.

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.

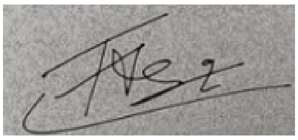
We hereby certify that this report has not been done by only one individual and all of us have contributed to the report. The length of contribution to the reports by each individual is noted within this certificate.

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

Signature:
Name: Afnan Hakim bin Adinazrin
Matric Number: 2315987
Contribution: Discussion,Conclusion,Appendices
References

Read	/
Understand	/
Agree	/

Signature: 

Name: Muhammad Taufiq bin Mukhtar
Matric Number: 2316271
Contribution: Abstract,Introduction,Materials & Equipment,

Data Collection, Results

Read	/
Understand	/
Agree	/

Signature: 

Name: Muhammad Danish Farhan bin Amiruddin
Matric Number: 2315423
Contribution: Equipment setup,Methodology,Data analysis

Recommendations

Read	/
Understand	/
Agree	/