الجامعة الاسلامية العالمية ماليزيا

INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

يونيۏرسيتي اسلام انتارابڠسا مليسيا

Garden of Knowledge and Virtue

SEM 1 25/26

SECTION 1

COURSE CODE: MCTA 3202

LECTURER NAME:
ZULKIFLI BIN ZAINAL ABIDIN

WEEK 8: BLUETOOTH DATA INTERFERING

PREPARED BY:
GROUP 1

NAME	MATRIC NUMBER
AMIR MUIZZUDDIN BIN NORAIDIL RAZMAN	2129579
MUHAMMAD HAIKAL HANIF BIN ABDUL RAZAK	2213297
AHMAD ILHAM BIN ABDUL AZIZ	2112109

INTRODUCTION

Our objective for today's task is to create a wireless temperature monitoring system using Bluetooth communication between an ESP32 and a smartphone. The Arduino will read the temperature from a DHT22 sensor, send it over Bluetooth, and it is able to receive simple control commands from the paired devices. The commands in this situation are "FAN ON" and "FAN OFF" . By the end of this task, we will able to how to make a program that can receive input and send output via Bluetooth and able to use DHT22 sensor to measure the surrounding temperature

MATERIALS

- ESP32
- Temperature sensor (DHT22)
- Smartphone with Bluetooth support
- Power supply
- Breadboard and jumper wires

EXPERIMENT SETUP

1. Hardware Setup:
    - Connect DHT 22 sensor to the ESP32
    - Power up the system

2. Arduino Programming

    Write an Arduino sketch to:

    - Read temperature data from DHT22 sensor
    - Transmit the data over Bluetooth serial connection
    - Able to receive input commands such as "FAN ON" and "FAN OFF" via Bluetooth

3. Bluetooth Programming
    - Pair the ESP 32 Bluetooth with the desired devices (smartphone)
    - Use a serial terminal app (Serial Bluetooth Terminal) to view the temperature data and send commands through it

4. Remote Monitoring
    - Observe real-time temperature readings on your desired device

- Send control commands

METHODOLOGY

1. Hardware Setup

1. The ESP32/Arduino board was connected to the DHT22 temperature sensor following the pin configuration in the manual.

2. For Arduino, the HC-05 Bluetooth module was connected through RX/TX pins.

3. The system was powered on, and the sensor was placed in the test environment.

2. Programming the Microcontroller

1. An Arduino sketch was uploaded to the board to:

   ○ Read temperature (°C) and humidity data from the DHT22.

   ○ Send the temperature data over Bluetooth at 9600 baud.

2. A command-reading function was added to allow the system to receive text commands such as “FAN ON” or “FAN OFF” to toggle an LED.

3. Bluetooth Pairing

1. The HC-05/ESP32 Bluetooth was paired with a smartphone or laptop.

2. A Bluetooth serial terminal app was opened to view temperature readings sent from the microcontroller.

3. Commands were typed into the terminal and transmitted back to the microcontroller to test the control function.

4. Data Collection

1. Temperature readings received over Bluetooth were recorded over time.

2. The effect of sending control commands (e.g., turning the LED/fan simulation on or off) was observed.

3. Optional: a Python script was used to log and plot the temperature trend.

CONCLUSION

This experiment successfully demonstrated a functional wireless temperature monitoring system using Bluetooth communication between an ESP32/Arduino and a paired device. The DHT22 sensor provided real-time temperature data, which was reliably transmitted over a serial Bluetooth link for display on a smartphone or computer. The system also supported simple bidirectional control, where user commands such as “FAN ON” and “FAN OFF” were recognized by the microcontroller and used to toggle an output device, simulating environmental control functionality.

The experiment met its objectives by integrating sensor acquisition, serial data transmission, Bluetooth pairing, and remote command execution into a cohesive system. This practical setup reflects core principles of IoT communication, demonstrating how microcontroller-based wireless monitoring can be applied in real-world scenarios such as HVAC control, smart home systems, and remote industrial monitoring.

Overall, the experiment provided hands-on exposure to Bluetooth interfacing, serial communication protocols, and sensor integration—skills that are essential in modern mechatronics and embedded systems development.