

FORMAN CHRISTIAN COLLEGE

(A CHARTERED UNIVERSITY)



CSCS 306 A

FA24

Assignment 3 Report

Daim Bin Khalid – 251686775

Hafsah Shahbaz – 251684784

Syeda Manal Ammad - 251606966

25th November 2024

Introduction

The purpose of this assignment was to develop a Bluetooth-controlled system using an Arduino microcontroller, a relay module, and a mobile application built using MIT App Inventor. The objective was to wirelessly control an LED via a relay, demonstrating the principles of embedded systems and wireless communication. This project combines hardware and software elements to achieve seamless interaction between the physical and digital worlds.

Function

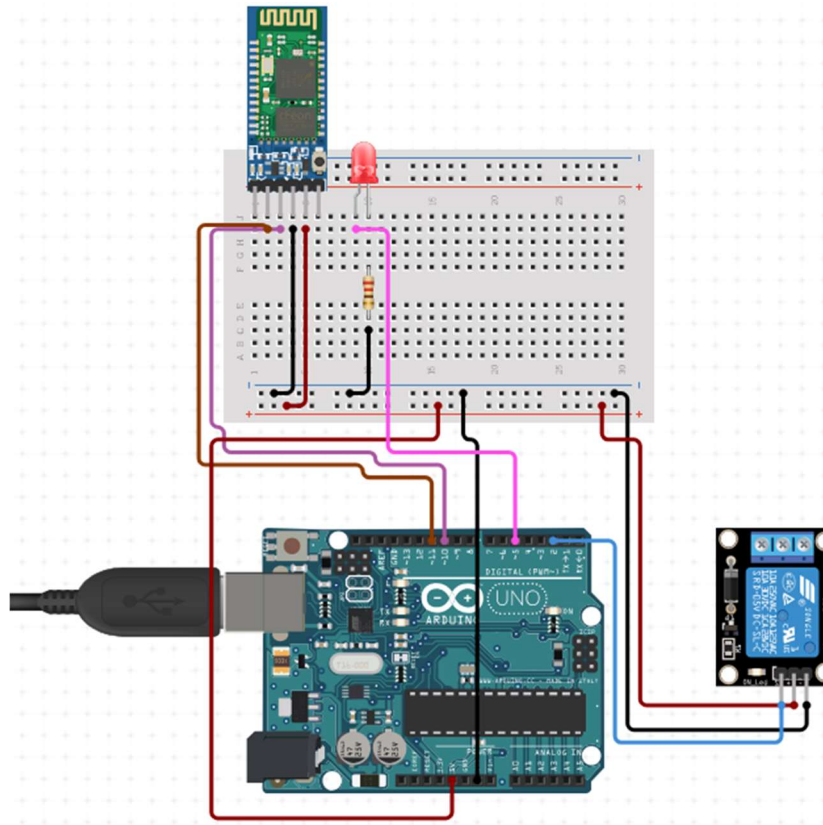
- Establish Bluetooth communication between Arduino and a mobile app.
- Control a relay to switch an LED ON and OFF.
- Create a mobile app with buttons to send commands (ON and OFF) using MIT App Inventor.
- Display Bluetooth connection status in the app.
- Implement Arduino code to process Bluetooth commands and operate the relay.

Pin Connections:

- **Bluetooth Module:**
 - VCC → Arduino 5V
 - GND → Arduino GND
 - TX → Arduino RX (Pin 0)
 - RX → Arduino TX (Pin 1)
- **Relay Module:**
 - VCC → Arduino 5V
 - GND → Arduino GND
 - IN → Arduino Pin 8
 - Relay COM → Arduino 5V.
- **LED Circuit:**
 - Positive leg (longer) → Relay NO (Normally Open) terminal.
 - Negative leg (shorter) → 220Ω resistor → Arduino GND

This setup allows the Arduino to control the relay, which in turn controls the LED.

Circuit Diagram



Algorithms and Logic

The Arduino sketch was designed to:

1. **Initialize the Serial Communication** for Bluetooth communication:

NOTE:

Ensuring RX/TX connections were disconnected during code uploads was critical to avoid conflicts.

2. **Control the Relay:**

Commands received via Bluetooth ('1' or '0') activate or deactivate the relay.

3. **Ensure Logic Compatibility:**

The relay operates with active-low logic, so the LOW signal turns it ON, and HIGH turns it OFF.

Code:

```
char receivedCommand; // To store the command received via
Bluetooth

int relayPin = 8;      // Pin connected to the relay IN terminal

void setup() {
    pinMode(relayPin, OUTPUT); // Set relay pin as output
    digitalWrite(relayPin, LOW); // Ensure relay starts in OFF
state
    Serial.begin(9600); // Initialize serial communication for
Bluetooth
    Serial.println("Ready to receive commands via Bluetooth!");
}

void loop() {
    // Check if data is available on the Bluetooth connection
    if (Serial.available() > 0) {
        receivedCommand = Serial.read(); // Read the incoming
command
        if (receivedCommand == '0') {
            digitalWrite(relayPin, HIGH); // Turn ON the relay
            Serial.println("Relay ON (LED ON)"); }
        else if (receivedCommand == '1') {
            digitalWrite(relayPin, LOW); // Turn OFF the relay
            Serial.println("Relay OFF (LED OFF)");
        } } }
```

Application Development

To create the mobile application, we used **MIT App Inventor**. The app provides an interface for selecting the Bluetooth module and sending commands ("1" and "0") to control the LED.

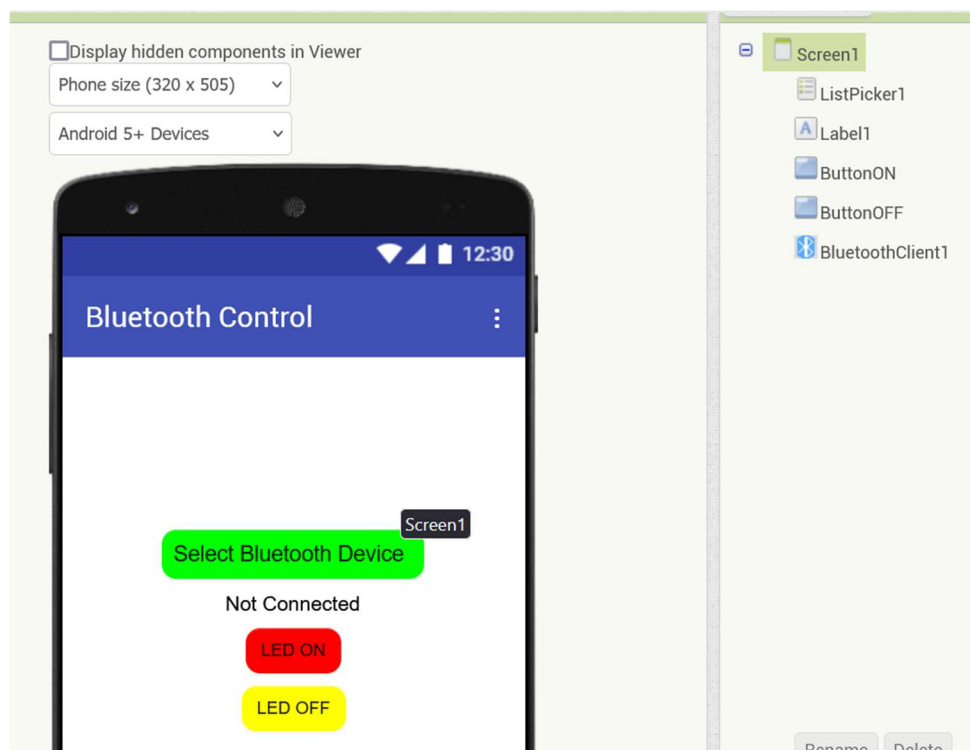
Steps to Develop the App:

1. Creating the App on a Mobile Device:

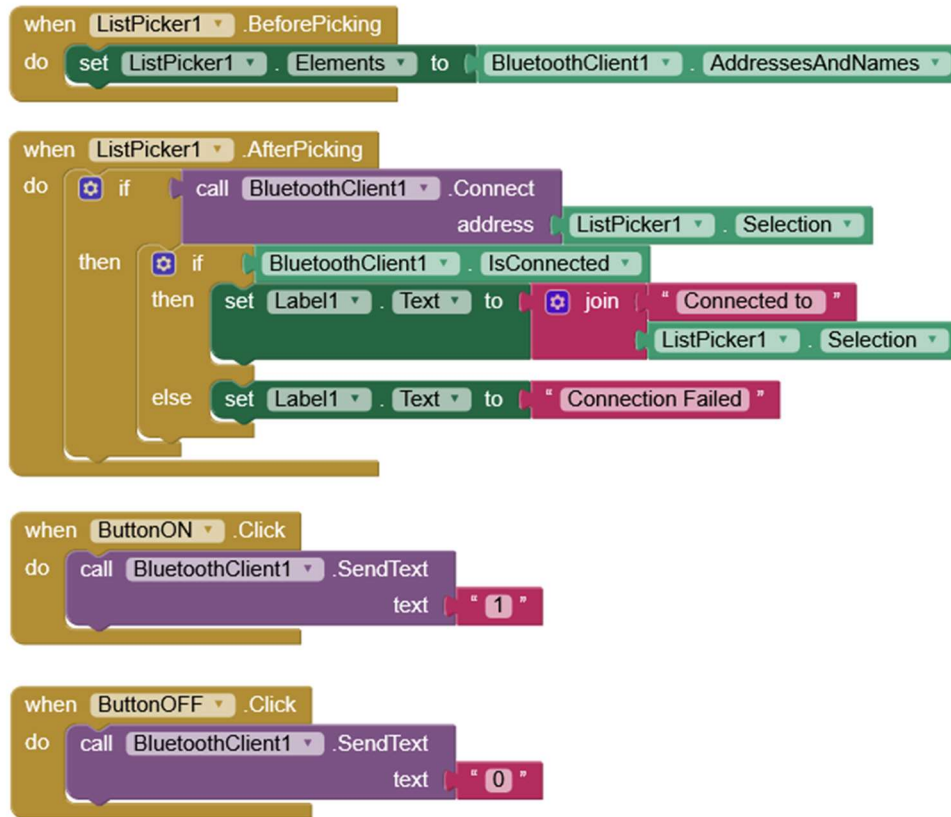
- We accessed MIT App Inventor from a laptop to design the application.
- The MIT AI2 Companion app was installed on the mobile device to test the app in real-time.
- The app was linked to the phone by scanning the QR code provided by MIT App Inventor.

2. User Interface Design:

- Added a **ListPicker** to display and select Bluetooth devices.
- Added two **Buttons** (LED ON and LED OFF) for control.
- Added a **Label** to display the connection status.
- Included a **BluetoothClient** component to handle communication.



3. Block Logic



References

Arduino FOSSEE. (n.d.). *Bluetooth Communication*. Retrieved November 25, 2024, from <https://arduino.fossee.in/node/47>

Output

