Experiment 4. Deploy IOT applications using Blue Bot Platform

Aim:

To design and deploy an IoT-based buzzer (LED) control system using the Blue Bot platform, a Bluetooth module, and an Arduino microcontroller, demonstrating real-time remote control of hardware components via a mobile application.

Apparatus Required

Arduino Board

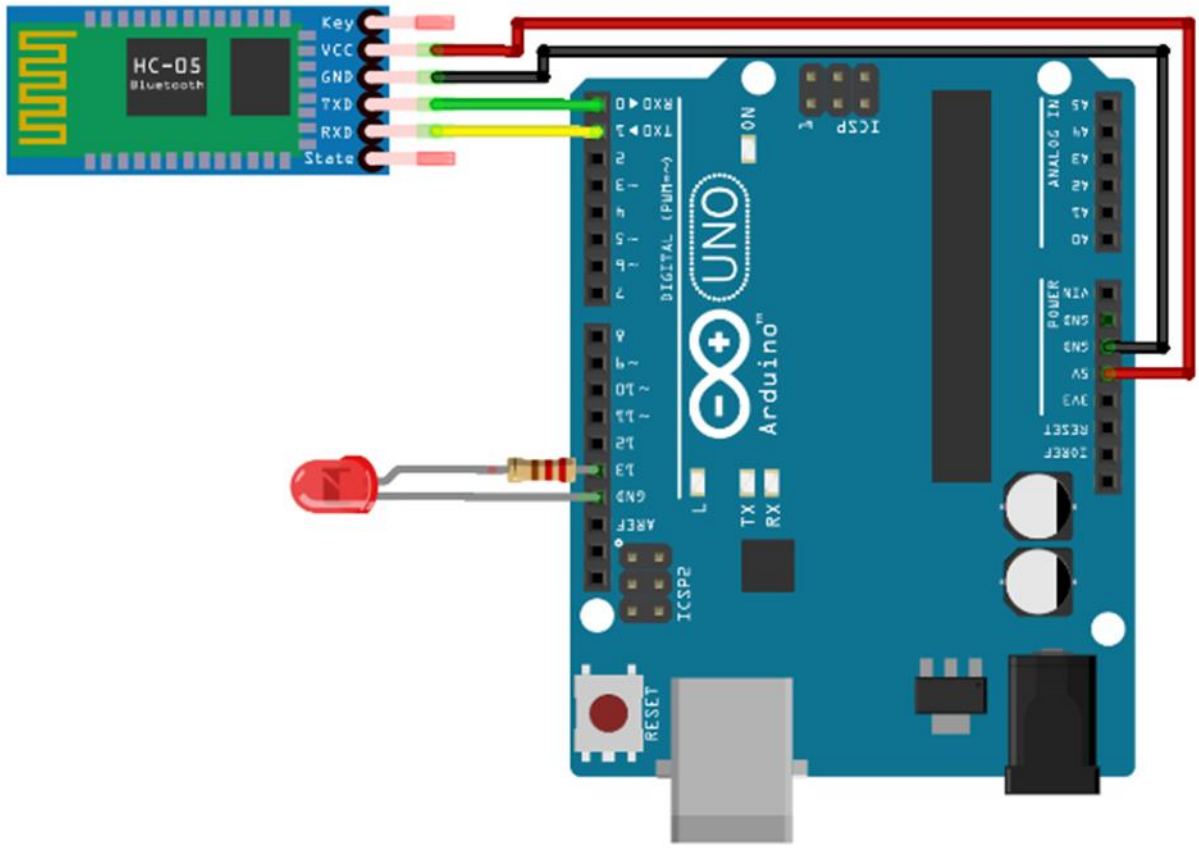
HC-05 Bluetooth Module

Buzzer or LED

Connecting Wires

Smartphone with Blue Bot App Installed

Laptop with Arduino IDE



Background Theory

The Internet of Things (IoT) is a technology that connects physical devices to the internet or local networks, enabling real-time monitoring and control. One of the key aspects of IoT is remote device automation, where sensors and actuators interact using wireless communication protocols. Bluetooth-based IoT applications, such as those deployed using the Blue Bot platform, facilitate short-range, low-power communication between a microcontroller (Arduino) and a smartphone app. The HC-05 Bluetooth module acts as a bridge, transmitting commands from the smartphone to the Arduino via serial communication (UART protocol). When the user presses a button in the Blue Bot app, the command is sent to the Arduino, which processes it and controls the connected buzzer (LED) accordingly. This setup demonstrates the integration of wireless communication and embedded systems, providing a fundamental approach to IoT-based device automation.

Procedure

Step 1: Hardware Setup

1. **Connect the Bluetooth Module (HC-05) to the Arduino:**
    - VCC → 5V
    - GND → GND
    - TX (HC-05) → RX (Arduino, Pin 10 for SoftwareSerial)
    - RX (HC-05) → TX (Arduino, Pin 11 for SoftwareSerial)
    -
2. **Connect the Buzzer (LED) to the Arduino:**
    - Buzzer (LED) Positive (+) → Digital Pin 13
    - Buzzer (LED) Negative (-) → GND

Step 2: Program the Arduino

Open Arduino IDE.

Upload the following code to control the buzzer (LED) via Bluetooth commands:

Program:

```
#include <SoftwareSerial.h>

SoftwareSerial BT(10, 11);  // RX, TX (for HC-05)

int buzzer = 13;

void setup() {

  pinMode(buzzer, OUTPUT);

  digitalWrite(buzzer, LOW);

  BT.begin(9600); // Start Bluetooth communication

}
```

```
void loop() {
  if (BT.available()) {
    char command = BT.read();  // Read command from Bluetooth
    if (command == '1') {
      digitalWrite(buzzer, HIGH);  // Turn buzzer ON
    }
    else if (command == '0') {
      digitalWrite(buzzer, LOW);  // Turn buzzer OFF
    }
  }
}
```

Algorithm

1. Initialize the Bluetooth module and set up the serial communication with Arduino.
2. Configure the buzzer (LED) pin as an output.
3. Establish a connection between the Bluetooth module and the Blue Bot mobile app.
4. Wait for incoming data from the mobile app via Bluetooth.
5. Read the received data and check if it matches predefined commands.
6. If the received command is "ON", activate the buzzer (LED).
7. If the received command is "OFF", deactivate the buzzer (LED).
8. Continue monitoring for incoming commands in a loop.

Step 3: Configure the Blue Bot App

1. Install Blue Bot from the Play Store/App Store.
2. Pair the HC-05 Bluetooth Module with your phone.
3. Open the App and Connect to HC-05.
4. Assign button controls:
    - Send ‘1’ for ON
    - Send ‘0’ for OFF

Step 4: Testing and Execution

1. Upload the code to the Arduino using the Arduino IDE.
2. Open the Blue Bot App and establish a Bluetooth connection.
3. Press the ON button → The buzzer (LED) should beep (glow).
4. Press the OFF button → The buzzer (LED) should stop (OFF).

Result

The experiment successfully demonstrated the deployment of an IoT application using the Blue Bot platform. The buzzer (LED) was controlled wirelessly via the Bluetooth module (HC-05) and Arduino, using commands sent from the Blue Bot app. The system responded efficiently, turning the buzzer (LED) ON and OFF as per the received signals.

Pre-Lab Questions:

1. What is the Blue Bot platform, and how is it used in IoT applications?
2. How does a Bluetooth module (HC-05) communicate with an Arduino microcontroller?
3. What is the difference between Bluetooth-based IoT communication and Wi-Fi-based IoT communication?
4. How does a buzzer work, and how can it be interfaced with an Arduino?
5. What are the basic AT commands used to configure the HC-05 Bluetooth module?

Post-Lab Questions:

6. What challenges did you face while interfacing the Bluetooth module with Arduino, and how did you resolve them?
7. How can the latency in Bluetooth communication impact real-time IoT applications?
8. Can the same setup be extended to control multiple devices simultaneously? If so, how?
9. How would you modify this experiment to integrate a sensor (e.g., temperature or motion sensor) along with the buzzer?
10. Compare the advantages and limitations of Bluetooth-based IoT with other wireless communication protocols like Zigbee and LoRa.