Analog Communication
Project Report

Designing an Arduino Application for Wireless Message Communication between Two Arduino boards.

Group Members:

ADITYA ABHILASH – 22EC01059

SADHANALA SIVA SURYA SAI – 22EC01013

TUSHAR JOSHI – 22EC01015

ADITYA UPADHYAY – 22EC01056

Component:

HC-05 Bluetooth, Module (*2), Arduino Board (*2), Breadboard, (*2), pushbutton (*2), LED (*2), Wires (as required)

Description:

In the contemporary landscape, communication stands as an essential component of our everyday routines. With our project, we aim to enhance this vital aspect by enabling seamless device-to-device communication through Bluetooth Modules. The provided circuit diagram offers a fundamental overview of the project, outlining the interconnectedness of devices and the flow of data within the system.

- **HC-05 Bluetooth Module**:

The HC-05 is a widely used Bluetooth module that enables wireless communication between devices. It operates in the 2.4GHz ISM band and can be configured as either a master or a slave device. It provides a simple and cost-effective solution for establishing serial communication between Arduino boards or between an Arduino and other Bluetooth-enabled devices.

- **Arduino Board**:

Arduino boards are open-source hardware platforms based on easy-to-use software and hardware. They consist of a microcontroller (e.g., ATmega328 in Arduino Uno) and support circuitry to program and control electronic devices. Arduino boards provide digital and analog input/output pins that can be interfaced with various sensors, actuators, and communication modules.

- **Breadboard**:

A breadboard is a solderless prototyping board used to build and test electronic circuits. It consists of a grid of interconnected metal strips beneath a plastic housing. Components can be inserted into the holes on the breadboard and connected using jumper wires. Breadboards are commonly used for temporary circuit prototyping and experimentation.

- **Pushbutton**:

A pushbutton is a simple electromechanical switch that completes or interrupts an electrical circuit when pressed. It consists of a button mechanism and electrical contacts. Pushbuttons are commonly used as user input devices in electronic projects to trigger actions or events, such as sending a signal to the Arduino when pressed.

- **LED**:

Light Emitting Diodes (LEDs) are semiconductor devices that emit light when an electric current passes through them. LEDs are available in various colors and sizes and are commonly used as indicators in electronic circuits. In this project, LEDs can be used to visually indicate the status of the system, such as indicating when a message is received or when the Bluetooth connection is established.

Configuring the HC-05 Bluetooth Module:

In configuring the HC-05 Bluetooth modules for our project, we first established a connection between the modules and our computer using a USB-to-serial converter. Ensuring the modules were in AT mode, we proceeded to configure them as both master and slave devices. For the slave module, we utilized AT commands to set its role as a slave (AT+ROLE=0) and assigned it a unique device name (AT+NAME=your_device_name). Additionally, we optionally set a passkey for pairing (AT+PSWD=your_passkey) and saved the changes made (AT+RESET). On the master module, we followed a similar process, but instead set its role as master (AT+ROLE=1) and initiated a scan for available devices (AT+INQ). Upon identifying the slave module, we paired them using its MAC address (AT+LINK=slave_mac_address) and saved the configurations (AT+RESET). Testing the setup, we confirmed the automatic connection between the master and slave modules and verified data transmission functionality.

Working:

- **Initialization**:

Initially, both Arduino boards are set up along with their respective HC-05 Bluetooth modules. Each HC-05 module is configured with specific settings such as baud rate, communication mode, and device name.

- **Pairing**:

The HC-05 modules are paired with each other to establish a Bluetooth connection. This pairing process ensures that the two modules can communicate exclusively with each other, preventing interference from other Bluetooth devices in the vicinity.

- **Message Sending**:

User input is facilitated through pushbuttons connected to one of the Arduino boards. When a user presses a pushbutton, the corresponding Arduino board sends a message signal to its HC-05 Bluetooth module.

- **Wireless Transmission**:

The HC-05 Bluetooth module receives the message signal from the Arduino board and transmits it wirelessly to the paired HC-05 module on the other Arduino board. The message is encoded and transmitted using Bluetooth communication protocols.

- **Message Reception**:

The receiving Arduino board's HC-05 module receives the transmitted message. The Arduino board then processes the received data, extracting the message content.

- **Feedback Display**:

Visual feedback is provided to the user through LEDs connected to both Arduino boards. Upon successful reception of a message, the LED on the receiving Arduino board illuminates, indicating message receipt. Additionally, the LED on the transmitting Arduino board may blink or change state to signify message transmission.

- **User Interaction**:

Users can engage in bidirectional communication by pressing pushbuttons on either Arduino board to send messages back and forth. The LEDs provide real-time feedback on the status of message transmission and reception, enhancing user interaction and system responsiveness.

- **Continuous Operation**:

The system can operate continuously, allowing users to send multiple messages between the two Arduino boards. The Bluetooth connection remains active, enabling seamless communication until either of the Arduino boards is powered off or the Bluetooth connection is manually terminated.

Conclusion:

The experiment involving the creation of an Arduino-based wireless messaging system utilizing HC-05 Bluetooth modules, pushbuttons, LEDs, and associated components yields significant findings. It underscores the feasibility and efficacy of employing HC-05 modules for short-range data transmission, demonstrating successful wireless communication between Arduino boards. The integration of hardware components such as pushbuttons and LEDs enhances user interaction and feedback within the system, while the utilization of breadboards enables rapid prototyping and iteration, crucial for refining and troubleshooting designs. Moving forward, this project opens up possibilities for various applications in remote control

systems, sensor networks, and Internet of Things (IoT) devices, highlighting the relevance and importance of analog communication in modern technology

CODE:

For MASTER Module:

```
#include <SoftwareSerial.h>
SoftwareSerial BTSerial(8, 9);
int state = 0;
const int ledPin = 4;
const int buttonPin = 2;
int buttonState = 1;
void setup()
{
  BTSerial.begin(9600);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  pinMode(buttonPin, INPUT);
  digitalWrite(buttonPin, HIGH);
}
void loop()
{
 if(BTSerial.available() > 0)
 {
    // Checks whether data is coming from the serial port
    state = BTSerial.read(); // Reads the data from the serial port
 }
 // Controlling the LED
 buttonState = digitalRead(buttonPin);
 if (buttonState == LOW)
 {
  BTSerial.write('1');
 }
 else
 {
  BTSerial.write('0');
 }

 if (state == '1')
 {
  digitalWrite(ledPin, HIGH); // LED ON
  state = 0;
 }
 else if (state == '0')
 {
  digitalWrite(ledPin, LOW); // LED OFF
```

```
  state = 0;
 }
}
```

For SLAVE Module:

```
#include <SoftwareSerial.h>
SoftwareSerial BTSerial(8, 9);
int state = 0;
const int led = 2;
const int button = 4;
int buttonstate = 1;
void setup()
{
  BTSerial.begin(9600);
  pinMode(led, OUTPUT);
  digitalWrite(led, LOW);
  pinMode(button, INPUT);
  digitalWrite(button, HIGH);
}
void loop()
{
 if(BTSerial.available() > 0)
 {
    // Checks whether data is coming from the serial port
    state = BTSerial.read(); // Reads the data from the serial port
 }
 // Reading the button
 buttonstate = digitalRead(button);
 if (buttonstate == LOW)
 {
   BTSerial.write('1'); // Sends '1' to the master to turn on LED
 }
 else
 {
   BTSerial.write('0');
 }

  if (state == '1')
 {
  digitalWrite(led, HIGH); // LED ON
  state = 0;
 }
 else if (state == '0')
 {
  digitalWrite(led, LOW); // LED OFF
  state = 0;
 }
}
```

CIRCUIT DIAGRAM