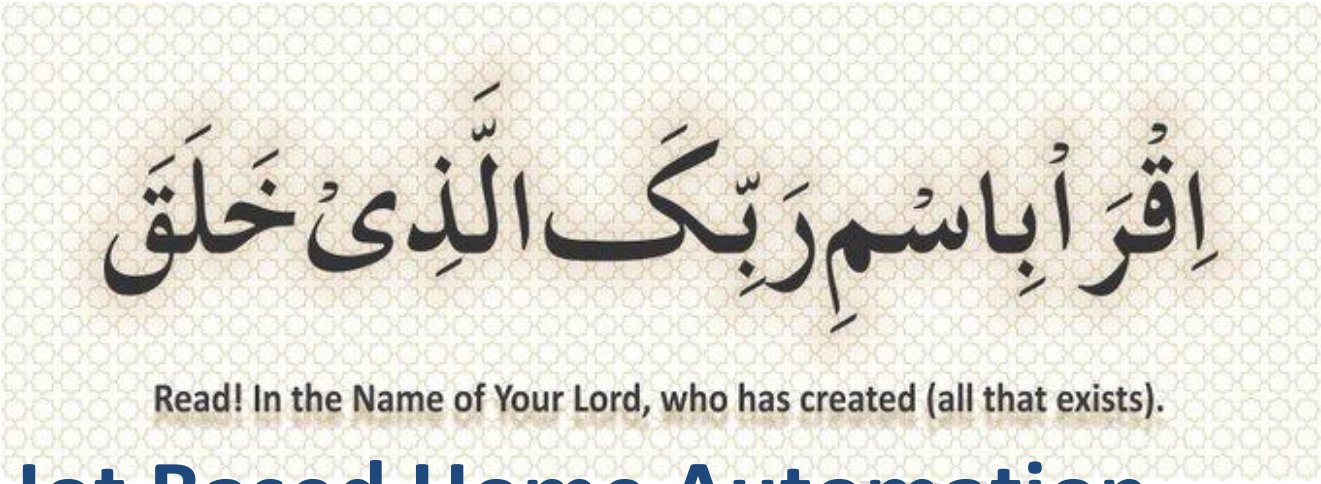

Iot Based Home Automation Using Bluetooth

Huma Khan Imran Ali Khan

Class : 04th

Academic Year : 2023-2024

Scholar's English School,

Beed By Pass, Aurangabad

Iot Based Home Automation Using Bluetooth

Project Submitted By

HUMA KHAN IMRAN ALI KHAN

Class : 04th

Academic Year : 2023-2024

Scholar's English School,
Beed By Pass, Aurangabad

Under the guidance of :

Ishrat Khatib (H.M)
Shaheen Pathan
Farooqui Tayyaba Miss
Tanzila Miss
Nubeera Technologies (Iot Division)

Figure 1

Introduction:

In the modern day's everyone uses smartphones and the internet. Therefore, Every smartphone has Bluetooth System. In this project, we will design a simple Arduino Bluetooth Control Home Automation using the HC-05 Bluetooth module, which is used to switch ON or OFF different electrical appliances remotely. Home Automation systems can make our life easy and secure. Here we will control 4 different home appliances using Smartphone App through Bluetooth communication.

Block Diagram:

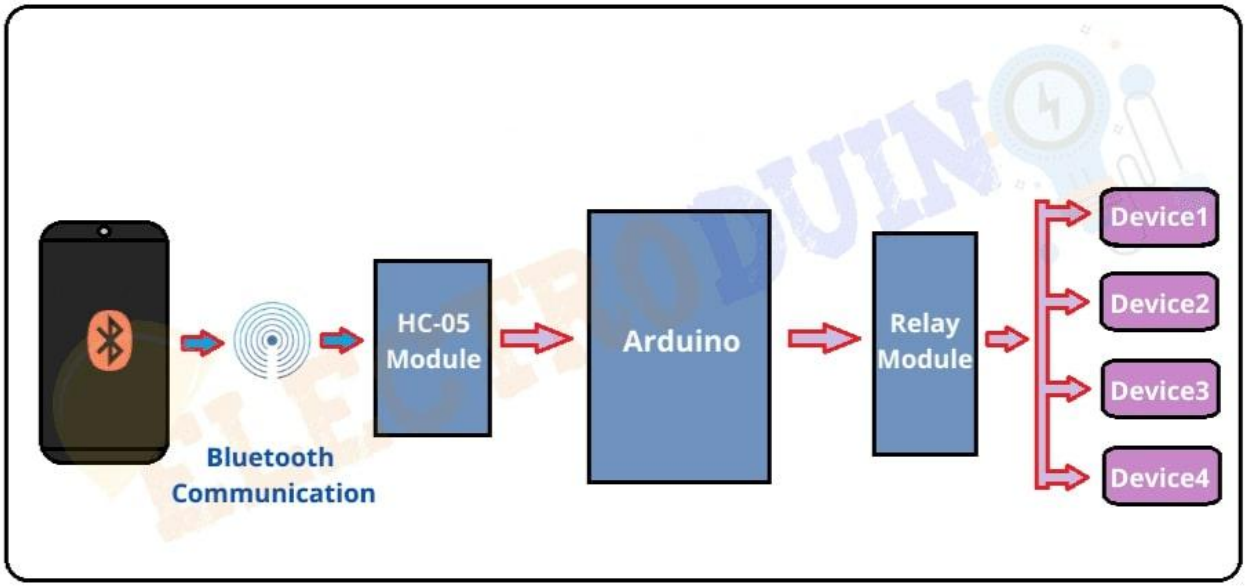

Figure 2

Components Required

Components Name	Quantity
Arduino UNO	1
HC-05 Bluetooth Module	1
4 Channel Relay Module	1
AC Bulb with Holder and Wire	4
PCB Prototyping board	1
9V 1 Amp Adapter	External **DC** Power supply for operating **Arduino**.
220V AC Power supply	
Connecting wires	As required in the circuit diagram
Smart Phone (Bluetooth supported)	1
Bluetooth Controller App	

HC-05 Bluetooth module:

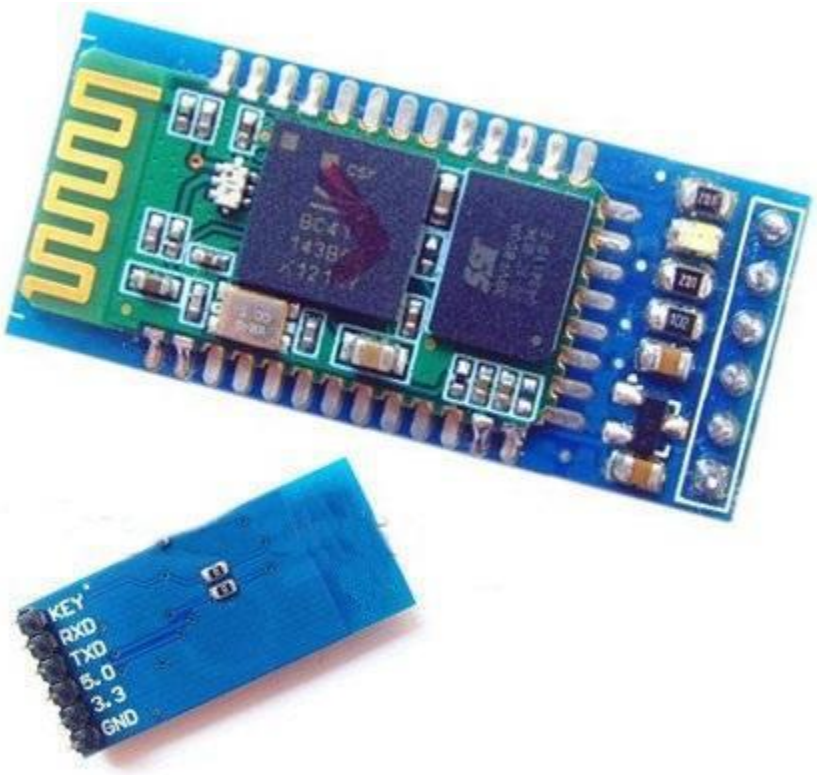

Figure 3

The Bluetooth module used is the HC-05 Bluetooth Module, shown in Figure 3. This module is cheap, readily available, and quick to acquire. It is also designed to ensure compatibility with Arduino microcontrollers. Furthermore, it is highly customizable, such as allowing its name to be changed, modifying its access password, and switching from master mode to slave mode. This last feature is important because it will enable future development of the Sensor Interface to add wireless Bluetooth sensor reading functionality. To communicate with Bluetooth sensors, the module should be in Master mode. However, when communicating with the Android device, the module should be in Slave mode. This Bluetooth module's ability to switch between these 2 modes will allow the Sensor Interface to communicate with Bluetooth sensors as wired sensors.

Figure 3 HC-05 Bluetooth Module

Figure 3 depicts the connections between the Bluetooth module and the microcontroller. The module has both a 5.0V input pin and a 3.3V input pin. Although the module may operate correctly with only one of these supply pins connected, both are connected to the microcontroller's output voltage pins to guarantee proper functionality.

The Bluetooth module's RX pin must connect with the microcontroller's TX pin, and its TX pin must connect to the microcontroller's RX pin. The Arduino Uno's default TX serial pin is P1 and the default RX pin is P0. However, these were not used for a couple of reasons. Most importantly, when P1 and P0 were used as TX and RX pins, the Bluetooth module would not communicate with the microcontroller. This meant that different pins had to be used. Fortunately, an Arduino library exists to change digital pins to RX/TX pins, called SoftwareSerial. Using this library, P3 was changed to a TX pin and P2 was changed to an RX pin. This decision has the added benefit of freeing up P1 and P0. This is beneficial because the Arduino has a useful debugging and monitoring feature called Serial Monitor. Using this, the Arduino can write messages to a computer screen when connected to a computer. However, this feature only works when P1 and P0 are not in use. Therefore, by using P3 and P2 for TX and RX instead of P1 and P0, testing and debugging was a much easier and faster process.

The Bluetooth module has one more input pin, labeled KEY. This pin is used for changing its mode between Master mode and Slave mode, as well as customizing its other properties. These options are not used currently, but they may be in the future. Therefore, this pin is connected to P4 on the Arduino Uno if property changes are ever desired.

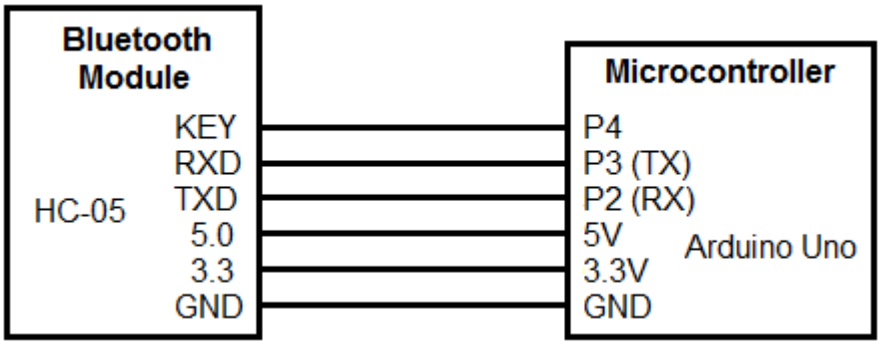

Figure 4 Microcontroller - Bluetooth Module Connectivity

The HC-05 Bluetooth module can communicate two-way wireless. so, it is used to communicate with any device like a Phone or Laptop with Bluetooth functionality. The module communicates with the help of UART at 9600 baud rate and it has a range up to <100m. It works on 3.3V and it has an onboard 5V to 3.3V regulator. The HC-05 has two operating modes, these are, Data mode and AT Command mode. The Data mode can send and receive data from other Bluetooth devices. The AT Command mode can be changed to the default device settings of HC-05.

This is a 6 pin module, but in this project, we will use only 4 pins, these are VCC, GND, TX, and RX. Where the VCC pin is connected to the 5V of the microcontroller, the GND pin is connected to the ground of the microcontroller, the RX pin is connected to the TX of the microcontroller and the TX pin is connected to the RX of the microcontroller.

Arduino:

Arduino comprises of both a physical programmable circuit board (commonly known as a microcontroller) and a programming software, or IDE (Integrated Development Environment) that can be run on a PC, used to compose and transfer PC code to the circuit board. It can be done by using the Arduino. Programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Unlike other programmable circuit boards, the Arduino does not require a different equipment (called a

software engineer) to upload code to the circuit board, one can essentially utilize a USB link.

Also, the Arduino IDE utilizes a rearranged rendition of C++, making it simpler to gure out how to program. In a word, Arduino make the functions of the microcontroller into a more accessible package. The Uno is one of the more prevalent boards in the Arduino family and an extraordinary option for the beginners.

Common Components of Arduino Boards:

There are different types of Arduino boards for different purposes. But all the boards have the majority of following components in common.

Figure 5

Starting clockwise from the top center:

- Analog Reference pin
- Digital Ground
- Digital Pins 2-13

- Digital Pins 0-1/Serial In/Out - TX/RX - These pins cannot be used for digital i/o (digitalRead and digitalWrite) if serial communication is also being used (e.g. Serial.begin).
- Reset Button - S1
- In-circuit Serial Programmer
- Analog In Pins 0-5
- Power and Ground Pins
- External Power Supply In (9-12VDC) - X1
- Toggles External Power and USB Power (place jumper on two pins closest to desired supply) - SV1
- USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board)

Breadboard:

A breadboard is a construction base for prototyping of electronics. In the 1970s the solderless bread-board (a.k.a. plugboard, a terminal array board) became available and nowadays the term "bread-board" is commonly used to refer to these. Because the solderless breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solderless breadboards are also popular with students and in technological education. A variety of electronic systems may be prototyped by using breadboards,from small analog and digital circuits to complete central processing units (CPUs)

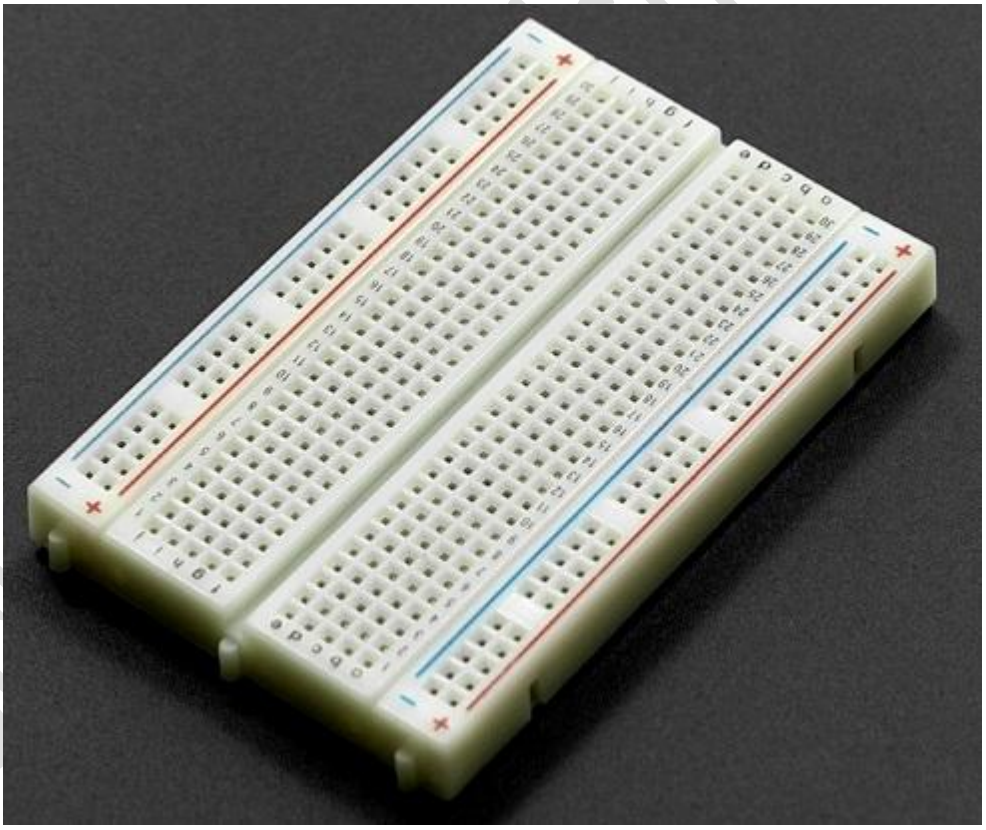

Figure 6

4 Channel 5v Relay Module:

The 4 Channel Relay Module is a convenient board which can be used to control high voltage, high current load such as motor, solenoid valves, lamps and AC load. It is designed to interface with microcontroller such as Arduino, PIC and etc. The relays terminal (COM, NO and NC) is being brought out with screw terminal. It also comes with a LED to indicate the status of relay.

Specification:

- Digital output controllable
- Compatible with any 5V microcontroller such as Arduino.
- Rated through-current: 10A (NO) 5A (NC)
- Control signal: TTL level
- Max. switching voltage 250VAC/30VDC
- Max. switching current 10A
- Size: 76mm x 56mm x 17mm

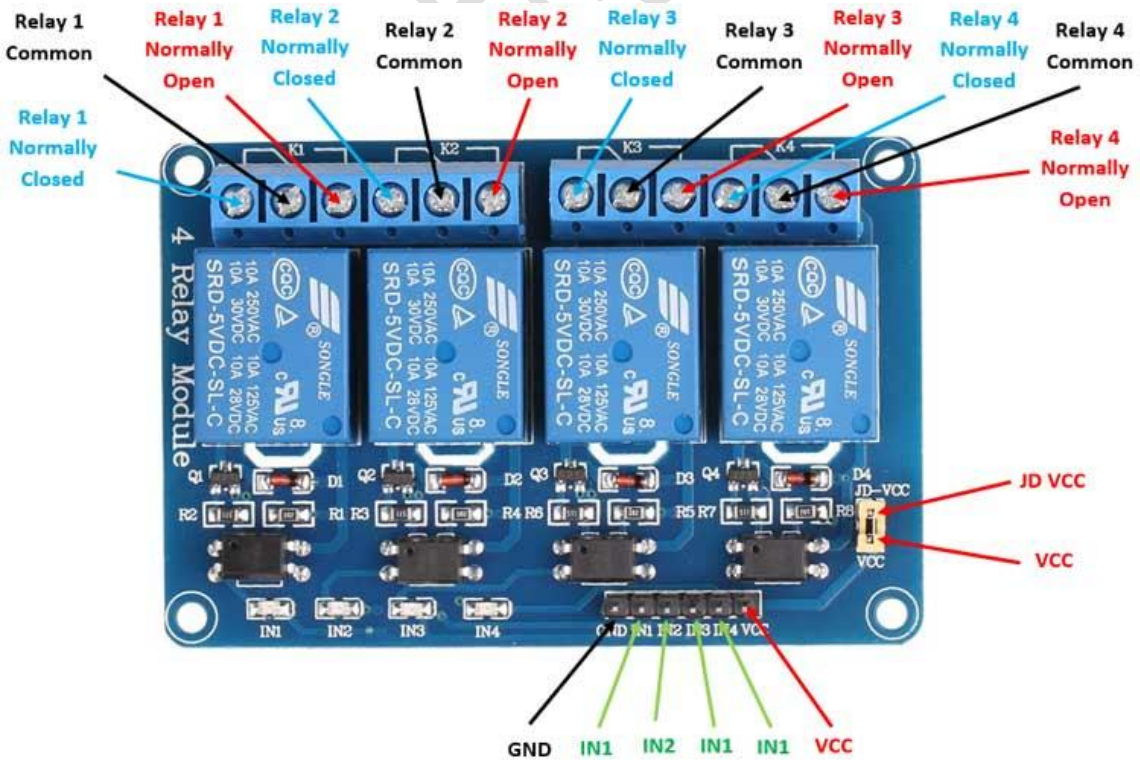

Figure 7

Circuit Diagram:

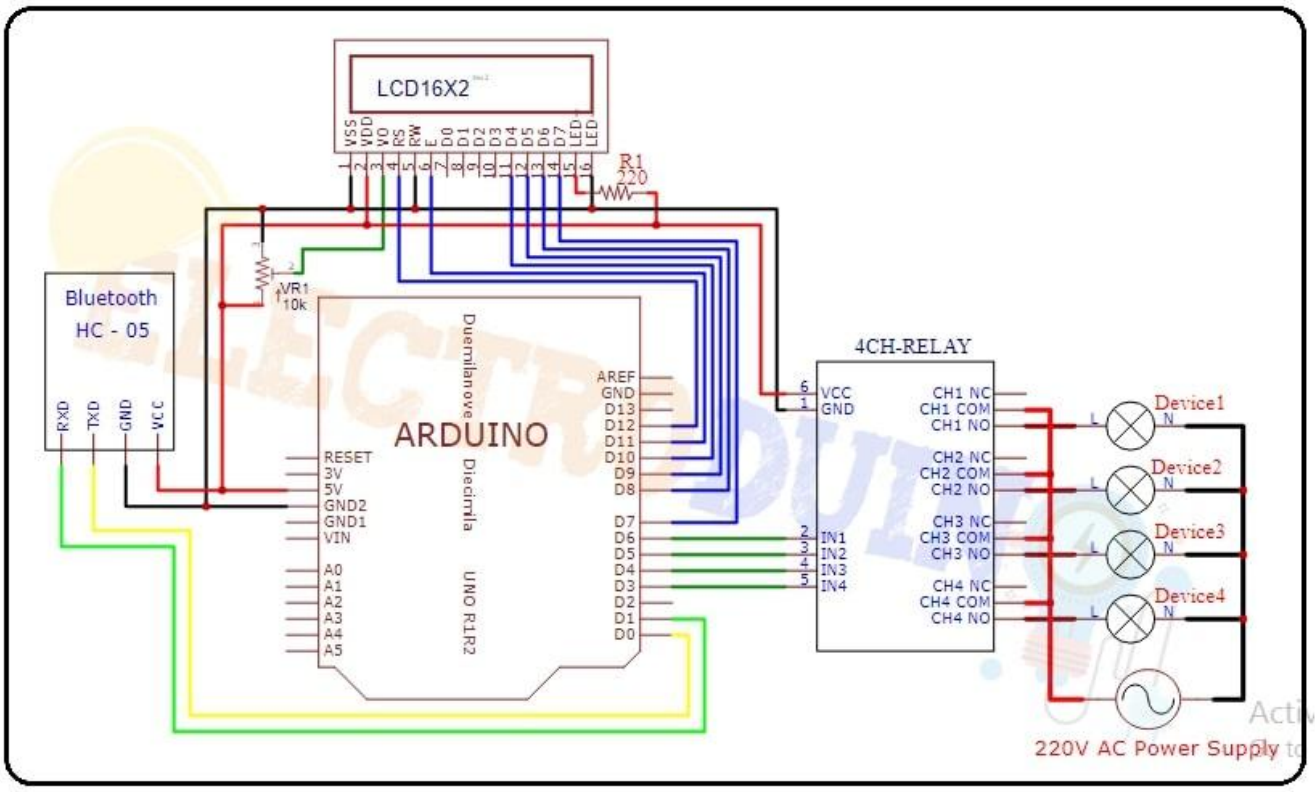

Figure 8

Circuit Wiring

Components Pin	**Arduino pin**
HC-05 Bluetooth module **Vcc**, Relay module **Vcc**, LCD Display (**Vdd, LED +)** Pins	Arduino **5V Pin**, but LCD Display **LED +** Pins connected to Arduino **5V Pin** through **220ohm resistor**.
HC-05 Bluetooth module **GND**, Relay module **GND,** LCD Display **(Vss, Rw,** and **LED -)** Pins	**GND (ground) Pin**
HC-05 Bluetooth module **TX**	Digital Pin **"D0 / RX"**
HC-05 Bluetooth module **RX**	Digital Pin **"D1 / TX"**
4 Channel Relay Modules **IN1, IN2, IN3, IN4**	Respectively Digital Pin **"D6", "D5", "D4", "D3",**
LCD Display **RS, E, D4, D5, D6, D7**	Respectively Digital Pin **"D12", "D11", "D10", "D9", "D8", "D7"**
LCD Display **V0** pin	10k Potentiometer **Signal** pin, Potentiometer one pin is connected to **5V** and another pin is connected to **GND**(ground)

Working Principle of Home Automation using Arduino and Bluetooth HC-05 Module

After complete the App setup/configuration, now we need to identify the load values of the App button. The app transmits Different load values when different buttons are pressed. When we will press a button on the app, the app sends a particular Load value to the Arduino through the Bluetooth module. These Lode Values are used in Arduino code to control a particular device by a particular app button. The Load values of the App buttons and their use is described in the below list.

App Buttons	Load Values	Device status	Control Relay (Relay Pin)
Button 1	**1**	Device1 On	Relay-1 (IN1)
	A	Device1 Off	
Button 2	**2**	Device2 On	Relay-2 (IN2)
	B	Device2 Off	
Button 3	**3**	Device3 On	Relay-3 (IN3)
	C	Device3 Off	
Button 4	**4**	Device4 On	Relay-4 (IN4)
	D	Device4 Off	
on All	**9**	All Devices (1, 2, 3 & 4) On	Relay-1, Relay-2 Relay-3, Relay-4 (IN1, IN2, IN3, IN4)
off All	**I**	All Devices (1, 2, 3 & 4) Off	

When we pressed any button of the App, the App sends a unique load value according to the button. The HC-05 Bluetooth Module received this unique load value and send it to the Arduino. Then, the Arduino compares the value with the predefined value of the button. If this value is matching then Arduino sends operating voltage to the relay module

For example, when we Press the App “Button 1“, then the app sends Load value “1” to the Bluetooth module. Then the Arduino gets this value through the Bluetooth module. Then the Arduino sends Low(0) input voltage to the Input-1 (IN1) pin of the relay module. Now the relay is in On mode. So the Device1 will also turn on, which is connected to the relay-1 of the relay module.

When we again Press the App “Button 1“, but this time the app sends Load value “A” to the Bluetooth module. Again the Arduino gets this value through the Bluetooth module. But this time the Arduino sends a High(5v) input voltage to the Input-1 (IN1) pin of the relay module. Now the relay is in Off mode. So the Device1 will also turn off, which is connected to the relay-1 of the relay module.

Arduino Bluetooth Control Home Automation Code

```
/*
Arduino Bluetooth Control Home Automation
*/
const int Device1 = 6; // Relay pin 1 (IN1)
const int Device2 = 5; // Relay pin 2 (IN2)
const int Device3 = 4; // Relay pin 3 (IN3)
const int Device4 = 3; // Relay pin 4 (IN4)
void setup()
 {
  Serial.begin(9600); //Sets the baud for serial data transmission
  // Set Relay pins as OUTPUT
  pinMode(Device1, OUTPUT);
  pinMode(Device2, OUTPUT);
  pinMode(Device3, OUTPUT);
  pinMode(Device4, OUTPUT);
  delay(2000);
}
void loop()
 {
// Read data from Bluetooth Module
  if (Serial.available() > 0)
```

```
{
 char Val = Serial.read();
 // Print Bluetooth Module data on serial monitor
 Serial.print("VAL: ");
 Serial.println(Val);
 //  When Bluetooth Module data "1" then Relay1 ON
if (Val == '1')
  {
   digitalWrite(Device1,LOW);
 delay(200);
  }
 //  When Bluetooth Module data "A" then Relay1 OFF
else if (Val=='A')
  {
   digitalWrite(Device1,HIGH);
   delay(200);
  }
 //  When Bluetooth Module data "2" then Relay2 ON
else if (Val=='2')
  {
  digitalWrite(Device2,LOW);
  delay(200);
  }
```

```
 //  When Bluetooth Module data "B" then Relay2 OFF
else if (Val=='B')
 {
 digitalWrite(Device2,HIGH);
 delay(200);
 }
 //  When Bluetooth Module data "3" then Relay3 ON
 else if (Val=='3')
  {
  digitalWrite(Device3,LOW);
  delay(200);
  }
 //  When Bluetooth Module data "C" then Relay3 OFF
 else if (Val=='C')
 {
 digitalWrite(Device3,HIGH);
 delay(200);
 }
//  When Bluetooth Module data "4" then Relay4 ON
else if (Val=='4')
 {
 digitalWrite(Device4,LOW);
 delay(200);
```

```
  }
//  When Bluetooth Module data "D" then Relay4 OFF
else if (Val=='D')
  {
  digitalWrite(Device4,HIGH);
  lcd.setCursor(13,1);
  lcd.print("OFF");
  delay(200);
  }
//  When Bluetooth Module data "9" then All Relay ON
else if (Val=='9')
  {
   digitalWrite(Device1, LOW);
   digitalWrite(Device2, LOW);
   digitalWrite(Device3, LOW);
   digitalWrite(Device4, LOW);
   digitalWrite(Device1, HIGH);
   digitalWrite(Device2, HIGH);
   digitalWrite(Device3, HIGH);
   digitalWrite(Device4, HIGH);
   delay(200);
  }
//  When Bluetooth Module data "9" then All Relay ON
```

```
  else if (Val=='I')
    {
     digitalWrite(Device1, HIGH);
     digitalWrite(Device2, HIGH);
     digitalWrite(Device3, HIGH);
     digitalWrite(Device4, HIGH);
     delay(200);
    }
  }
}
```

Bluetooth App configure

First of all, open the Play Store of your Android Smartphone and search for “Bluetooth Control for Arduino”. Then you will get this App. Now install this app.

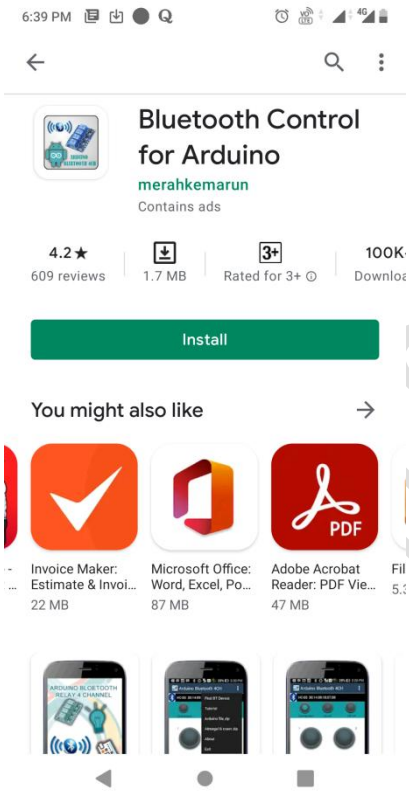

Figure 9

After completing the installation process, open the app and make sure the circuit is connected to the power supply and the Bluetooth module is on. Also, make sure the Bluetooth module pair with your smartphone. You can pair your phone with the HC-05 Bluetooth Module like we normally pair the phone with Bluetooth earphones or speakers. If your smartphone pair with the HC-05 Bluetooth module, then you can see the “HC-05 Bluetooth” name on the top side of the App and select it.

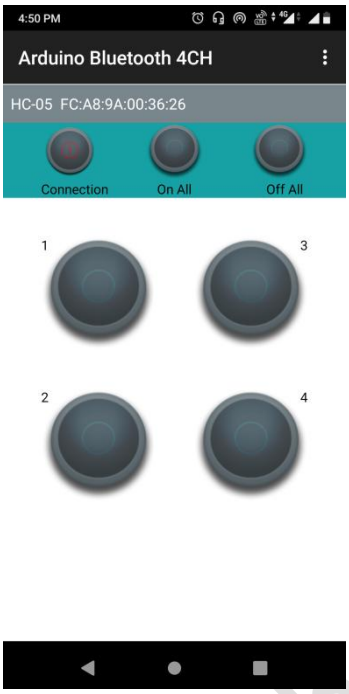

Figure 10

Now click on the **"Connection"** Button. When the App successfully connects to the Bluetooth Module, Then you can see the blue light on the "**Connection button**". At the same time, you can also see the LED on the Bluetooth Module is **blinking with some delay**. It means your Bluetooth is successfully connected.

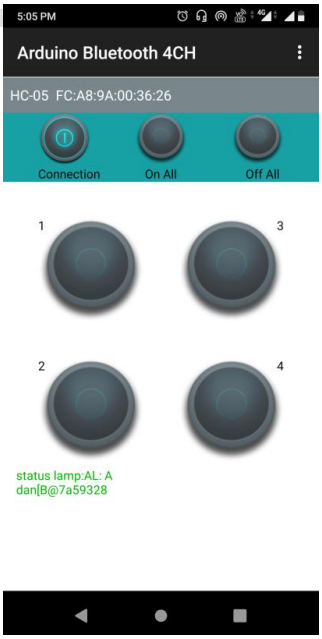

Figure 11

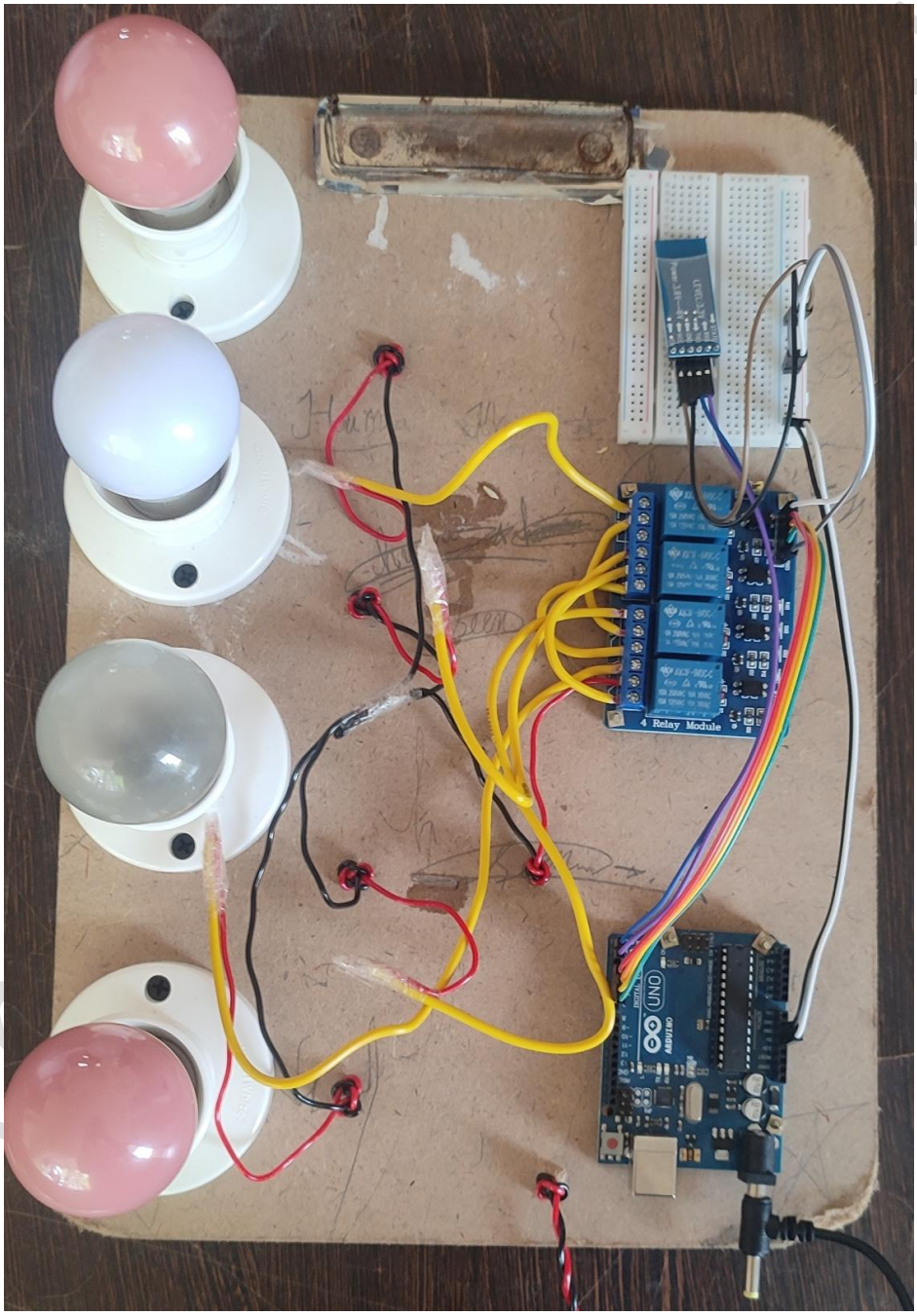

Figure 12