

## ***Bluetooth module Home Automation***

### **I. Summary**

In this project, we will explore how to control a lamp, fan, or any other electrical appliance in your space using an Arduino and a Bluetooth module. At the end of the project, you will be able to control the connected load from your smartphone.

### **II. Objectives**

- Control light via Bluetooth using Arduino.
- Schedule appliances to turn on/off at preset times Set the mood with the brightness controller.

### **.III. Industry-Based Applications**

Wireless communication has been used in industrial applications for more than 40 years. Among the first applications was used in wireless control of Automated Guided Vehicles. Using a wireless solution in industries Increases mobility eliminates expensive and maintenance-heavy transmission media such as flexible cables, swivels, etc. Overcoming large and problematic zones has to Achieve fast and efficient installation and commissioning. Ensure personnel safety in hazardous areas (for instance, when needing to climb in a crane) by offering a control possibility from a further distance than can be

the case with a cable.

#### **IV. Project Methodology**

This project is to lay the basic groundwork for controlling electrical loads using an Arduino controller and a smartphone. A communication medium between the user is provided by the Bluetooth module, through a smartphone such as an Android and the system by giving a voice command to the smartphone. A person can send commands to the Bluetooth voice control for Arduino's voice the software application installed on the phone i.e. connected via Bluetooth module.

A relay is used to control the home appliances with the Arduino. The relays used in this system are 4 pin relays. The relays are normally in the closed state. For our use-case, we want to turn on the bulb only when we send a signal from a smartphone. That's the reason we connect the load on the NO (Normally Open) terminal, so that when the relay is triggered from the Arduino, the contact brush flicks from NC to NO terminal, thereby completing the circuit to electromagnetic induction. We can do a lot more with home automation, that can be further explored and be used in projects using this as a smaller example of how it can be applied.

#### **Components:**

- 1x Arduino UNO



- 1x Solid-state Relay.



- 1x HC 05 Wireless Bluetooth Module



- 1x Lamp



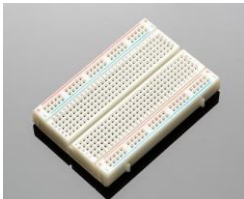
- 1x 2.2k ohm resistor



- 1x 1k ohm resistor



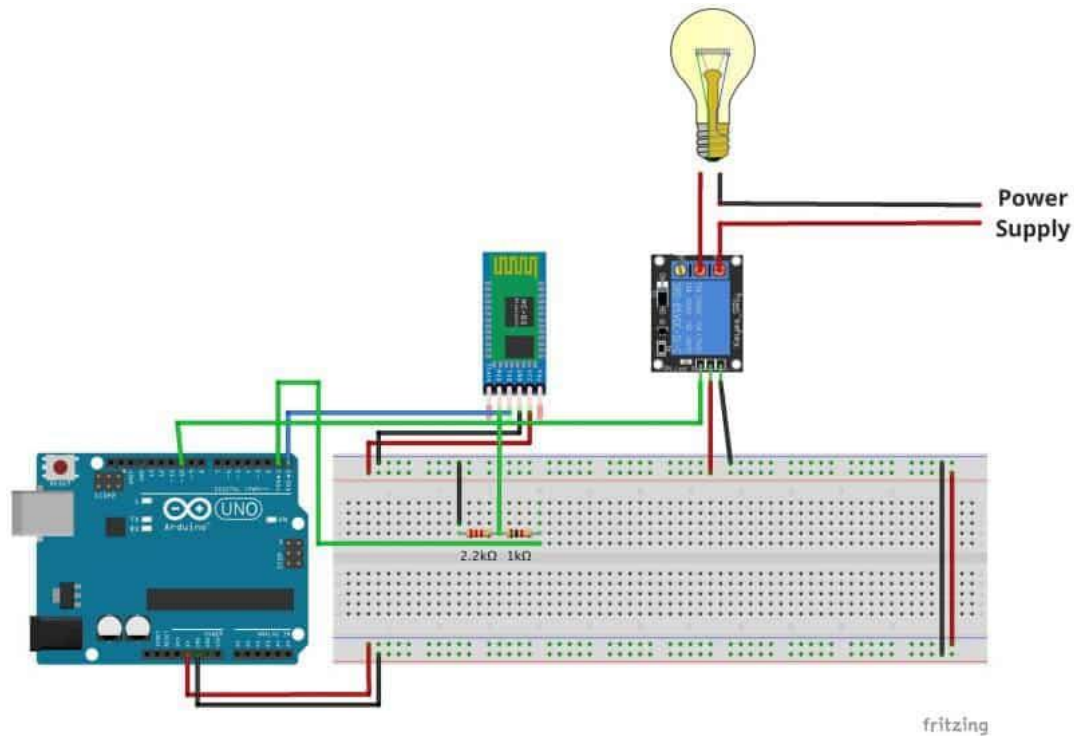
- 1x Breadboard



- Jumper wires



## Project Procedures:



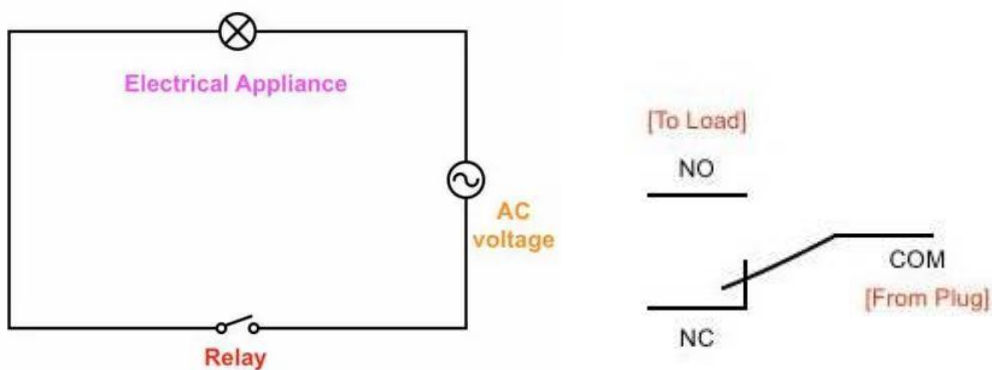
1. Connect 5V and GND pins of Arduino to the bus strips on the breadboard as shown in the circuit diagram.
2. To connect HC-05 module with Arduino, insert its 5V and GND pins to the bus strips on the breadboard.

Note: In case HC-05 module supports 3.3V, please power it using the 3.3 V supply from Arduino.

3. Connect the TXD pin on the HC-05 module with the RXD pin (Pin 0) of Arduino.
4. TXD on HC-05: Transmit data from the Bluetooth transceiver.

5. Pin 0 on Arduino (RXD): Receive data on Arduino two-way by connecting these pins, we are establishing two-way communication between Arduino and HC-05, so that we can turn the device on/off with the command properly.
6. Next, as the receiver data lines on HC-05 are 3.3V tolerant, we need to convert the 5V input signal from Arduino into a 3.3 V signal. While it can be achieved easily through a bi-directional logic level converter, we're using a voltage divider to convert the 5V input signal into a 3.3 V signal.
7. Voltage Divider: Connect 1k ohm and 2.2k ohm resistors across the GND and TXD on Arduino, and we obtain the 3.3 V tolerant input directly from the intersection of the two resistors.

### Setting up relay



Firstly, connect the 5V and GND pins to the bus terminals on the breadboard.

Connect the IN1 pin on the relay board with PIN 4 of Arduino.

In case you have a multi-channel module (2, 4, or 8 channels), you can connect IN2, IN3 ...

IN(n) with different digital pins of Arduino, and repeat the below steps for configuring other pins.

After connecting the signal pins to Arduino, we need to connect the AC load to the relay module.

Now, if you look carefully at the terminal block on the relay module, you would find three slots.

#### Video:

[https://drive.google.com/file/d/0B\\_mU1cI7wXgrRUd4S1RrT0dXbzA/view](https://drive.google.com/file/d/0B_mU1cI7wXgrRUd4S1RrT0dXbzA/view)

## Appendix

#### CODE:

```
#define RELAY_ON 0
#define RELAY_OFF 1
#define RELAY_1 4
char data = 0;

void setup() {
  // Set pin as output.
  pinMode(RELAY_1, OUTPUT);
  // Initialize RELAY1 = off. So that on reset it would be off by default
  digitalWrite(RELAY_1, RELAY_OFF);
  Serial.begin(9600);
  Serial.print("Type: 1 to turn on the bulb. 0 to turn it off!");
}

void loop() {
  if (Serial.available() > 0) {
```

```

data = Serial.read();           //Read the incoming data and store it into variable data
Serial.print(data);           //Print Value inside data in Serial monitor
Serial.print("\n");           //New line

if(data == '1'){
  digitalWrite(RELAY_1, RELAY_ON);
  Serial.println("Bulb is now turned ON.");
}
else if(data == '0'){
  digitalWrite(RELAY_1, RELAY_OFF);
  Serial.println("Bulb is now turned OFF.");
}
}
}

```

**Note:** While the HC-04 module is connected, you'll get the following error:

```
avrdude: stk500_getsync() attempt 1 of 10: not in sync: resp=0x00
```

An error occurred while uploading the sketch

simply unplug the jumper wire connected to Pin 0 of Arduino (Rx pin), and re-attempt to upload code on Arduino. You should now be able to update the code successfully. The reason these simple hack works is that Arduino is no longer receiving data from two sources, and therefore can receive the code from the computer.

## ***I. References***

1. Abhi.cool. (n.d.). How to Build a Bluetooth controlled Home Automation Setup Using Arduino. Retrieved from <https://smartify.in/knowledgebase/build-bluetooth-controlled-home-automation-setup-using-arduino/>