DELHI TECHNOLOGICAL

UNIVERSITY

SUBJECT- Communication Systems (EC206)

SUBMITTED To – Prof. Rohit Kumar

SUBMITTED BY- Abhay Lakhotra & Anshul

(2K19/EC/006) (2K19/EC/022)

COMMUNICATION SYSTEMS INNOVATIVE PROJECT REPORT

BLUETOOTH CONTROLLED ELECTRONIC HOME APPLIANCES

Index

ABSTRACT

In this project we are going to make an electronic hardware model which we can connect with our main electronic switches and control different electrical appliances and electronic devices with our android device using the help of Bluetooth technology. The project implementation is based on the Arduino microcontroller, with Bluetooth communications capability and this project will also be going to help elder or handicapped people those are facing difficulty in operating the conventional wall switches by allowing them to control all the electrical appliances by an android app on a device.

INTRODUCTION

As now a days we can see that technology is too advanced as now we can control our tv system with remote but have anyone think that what if we are able to control our electronic switches too with the help of our mobile. Yes, I know there are some which we can control with android device, but they are not pocket friendly so to solve. We have come up with a new method called Bluetooth controlled electronic home appliances using Arduino. This is super cost effective that a user doesn't even need to buy a remote control for this as he/she can control all appliances by using an android device. This will also save our time as person can control all devices using android app without even going towards switches and switch on/off them.

Why Bluetooth not WI-FI?

Bluetooth technology (Low power consumption, Low range and high bandwidth) uses low power signals. It requires a minimum amount of energy to function which results in less power consumption. Therefore, it is **Energy-Efficient.** Bluetooth can send voice and data simultaneously.

Disadvantages

- Slower rate (nearly 720 Kbps)
- Battery drain – It slowly drains the battery of our mobile.

Pros of our Home Automation project-

1. Security- with the help of this with a Tap of our finger we can turn on the lights when arrive at home so no worried about something hiding in the shadows. Or in your pathways. We can also turn it to turn on when not at home to look like someone is in home and ward off potential robbers.

2. Energy Efficiency- we can Increase home energy efficiency by remotley powering off systems and appliances when we are not using them.

COMPONENTS REQUIRED

- Arduino UNO R3 (atmega328p)
- Breadboard
- 8 channel 5v relay module
- (HC-05) Bluetooth module
- Power Supply
- Fan
- Lamp
- Connecting wires
- Arduino IDE
- Android device

Components Description

- Arduino UNO R3 (atmega328p)

Arduino Uno is basically a microcontroller board which works on the principle of Atmega 328p. It basically has

- A ceramic resonator that is 16MHz
- 14 digital input/output pins (6 out of which provide PWM output)
- A reset buttons.

- A USB connection
- A power jack
- Six analog input pins

It is basically an 8-bit microcontroller based on RISC architecture. The Arduino Uno(atmega328p) board is shown in the figure below

Technical specification of Arduino Uno is

1. Microcontroller - ATmega328P

2. Operating voltage - 5V

3. Input voltage(recommended) - 7-12V

4. Input voltage(limits) - 6-20V

5. Digital I/O pin - 14

6. Analog input pins - 6

7. Dc current per Input/Output pin - 40mA

8. Dc current used for 3.3V pin - 50mA

9. Flash memory – There is 32KB of flash memory out which 0.5 KB used by boot loader

10. SRAM - 2KB

11. EEPROM - 1KB

12. Clock Speed - 16 MHz

- <u>Breadboard</u>

A breadboard is basically a solderless device used for doing temporary prototypes with electronics and test circuits.

designs. In breadboard we can interconnect most of the circuits just by inserting their leads or terminal into the holes and then making connection through wires as per our need.

- <u>8 channel 5v relay module</u>

In relay module, transistors and other important components required to drive relay are embedded on the module itself.

- <u>(HC-05) Bluetooth module</u>

The Bluetooth module (HC-05) has six pins –

- Enable,

- VCC,
- Ground,
- Transmit Data (TxD),
- Receive Data (RxD)
- State.

The Enable and State pin are not used so we don't connect them in the circuit. The VCC and Ground pins are connected to the common VCC and Ground of the main Arduino module. The TxD and RxD pins of the module are connected to the pins 10 and 11 of the Arduino for transfer and receive of data

.

- POWER SUPPLY

- Fan

This is a type of load we are using in our project to verify that our hardware model is working properly or not

- Lamp

This is a type of load we are using in our project to verify that our hardware model is working properly or not

- Connecting wires

These are used to connect different components of our hardware model like Arduino, relay, Bluetooth module to each other.

- Arduino IDE

This is a software with the help of which we are going to code our Arduino board that what function it is going to perform at what input

- Android Device

With the help of this we are going to give Bluetooth command to our hardware model like when we have to turn on the circuit and when to turn It off. This device should contain Bluetooth connectivity as we have to connect the hardware model using Bluetooth

BLOCK DIAGRAM

Acc. To the block diagram as we can see that there are 6 components in our block

- Bluetooth Module
- Arduino
- Relay Module
- Main Power
- Load

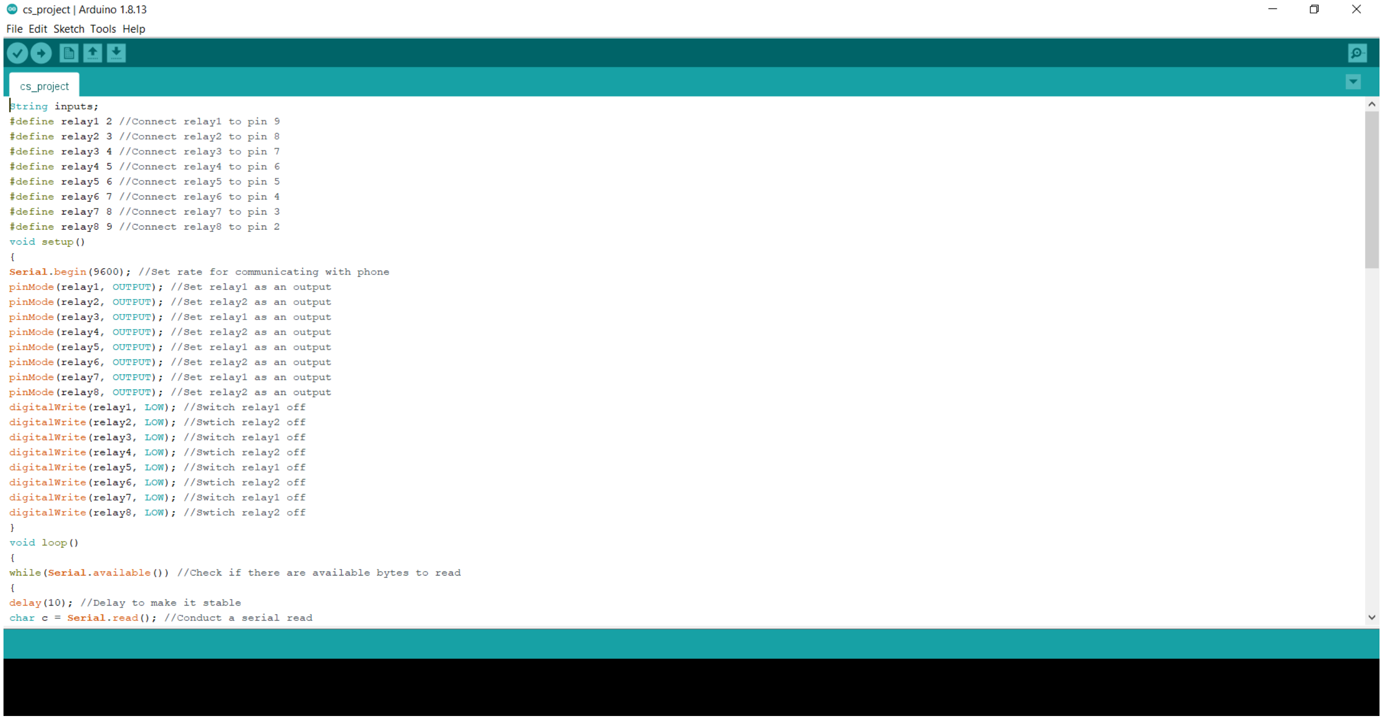
- 9V power source

Now from the diagram it can be seen that our Arduino module is connected to 9 v power supply and Bluetooth module which are basically the input of the Arduino here the Arduino will get the info of what to perform from the Bluetooth module. Then after that Arduino is connected to relay module this relay module take input from Arduino and work on the basic of that it is also connected to the main power supply and the output of the relay module is connected to our load which we are going to operate

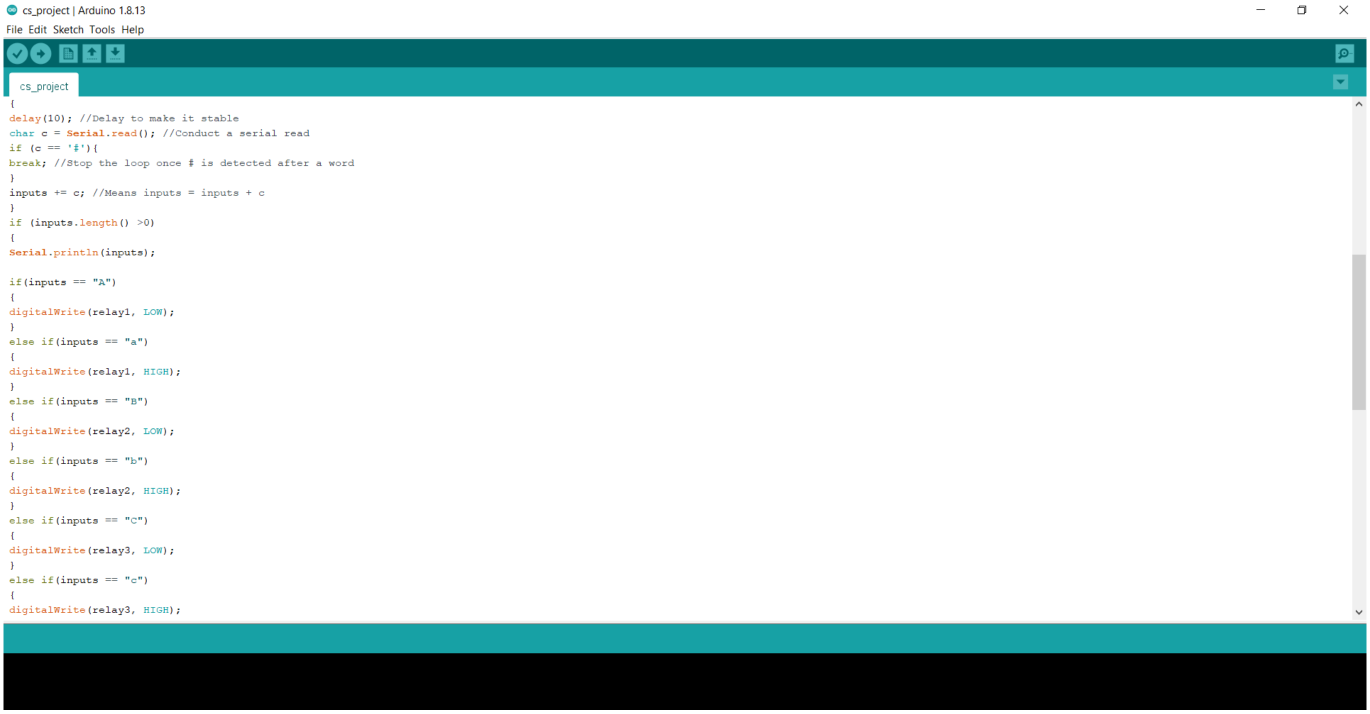
CIRCUIT DIAGRAM

The below circuit diagram has been performed (simulated) on the proteus software in order to ensure that the hardware model which we are going to implement is working properly. The same model is implemented on the hardware.

Arduino code

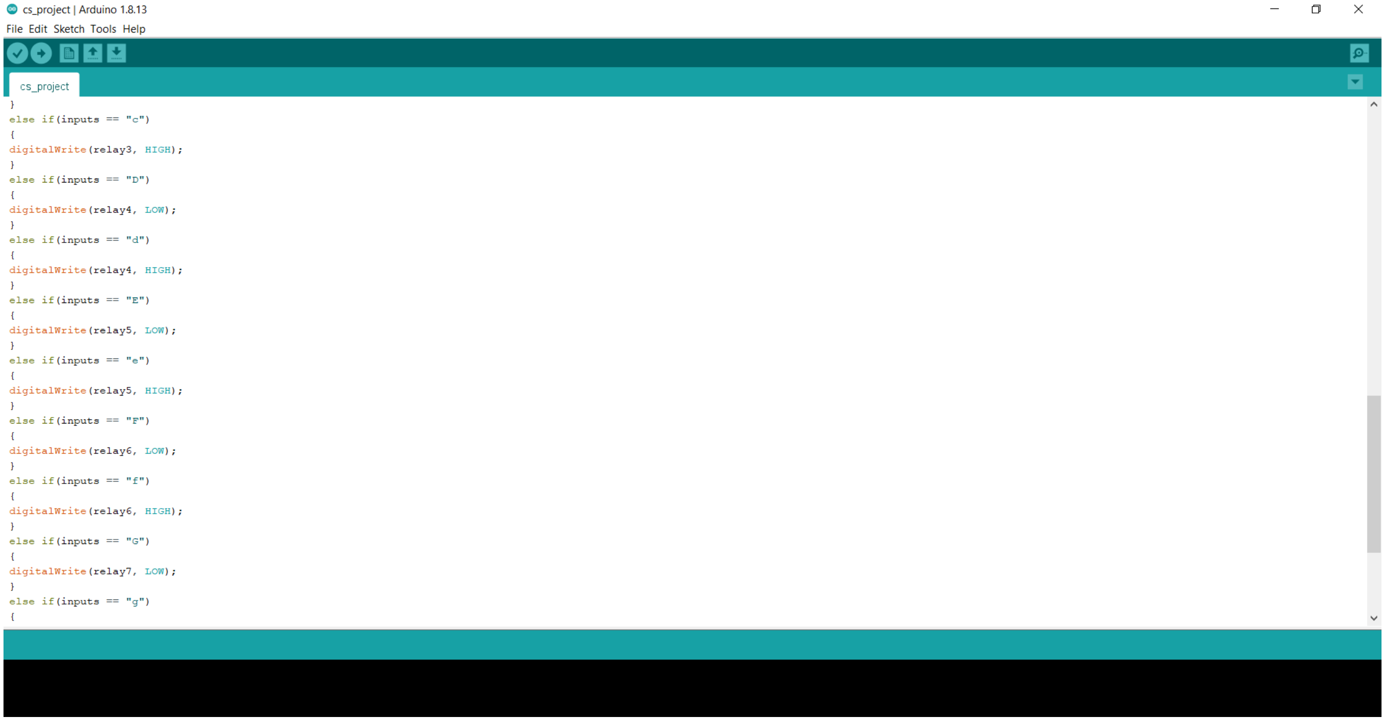


```
String inputs;
#define relay1 2 //Connect relay1 to pin 9
#define relay2 3 //Connect relay2 to pin 8
#define relay3 4 //Connect relay3 to pin 7
#define relay4 5 //Connect relay4 to pin 6
#define relay5 6 //Connect relay5 to pin 5
#define relay6 7 //Connect relay6 to pin 4
#define relay7 8 //Connect relay7 to pin 3
#define relay8 9 //Connect relay8 to pin 2
void setup()
{
Serial.begin(9600); //Set rate for communicating with phone
pinMode(relay1, OUTPUT); //Set relay1 as an output
pinMode(relay2, OUTPUT); //Set relay2 as an output
pinMode(relay3, OUTPUT); //Set relay1 as an output
pinMode(relay4, OUTPUT); //Set relay2 as an output
pinMode(relay5, OUTPUT); //Set relay1 as an output
pinMode(relay6, OUTPUT); //Set relay2 as an output
pinMode(relay7, OUTPUT); //Set relay1 as an output
pinMode(relay8, OUTPUT); //Set relay2 as an output
digitalWrite(relay1, LOW); //Switch relay1 off
digitalWrite(relay2, LOW); //Swtich relay2 off
digitalWrite(relay3, LOW); //Switch relay1 off
digitalWrite(relay4, LOW); //Swtich relay2 off
digitalWrite(relay5, LOW); //Switch relay1 off
digitalWrite(relay6, LOW); //Swtich relay2 off
digitalWrite(relay7, LOW); //Switch relay1 off
digitalWrite(relay8, LOW); //Swtich relay2 off
}
void loop()
{
while(Serial.available()) //Check if there are available bytes to read
{
delay(10); //Delay to make it stable
char c = Serial.read(); //Conduct a serial read
```

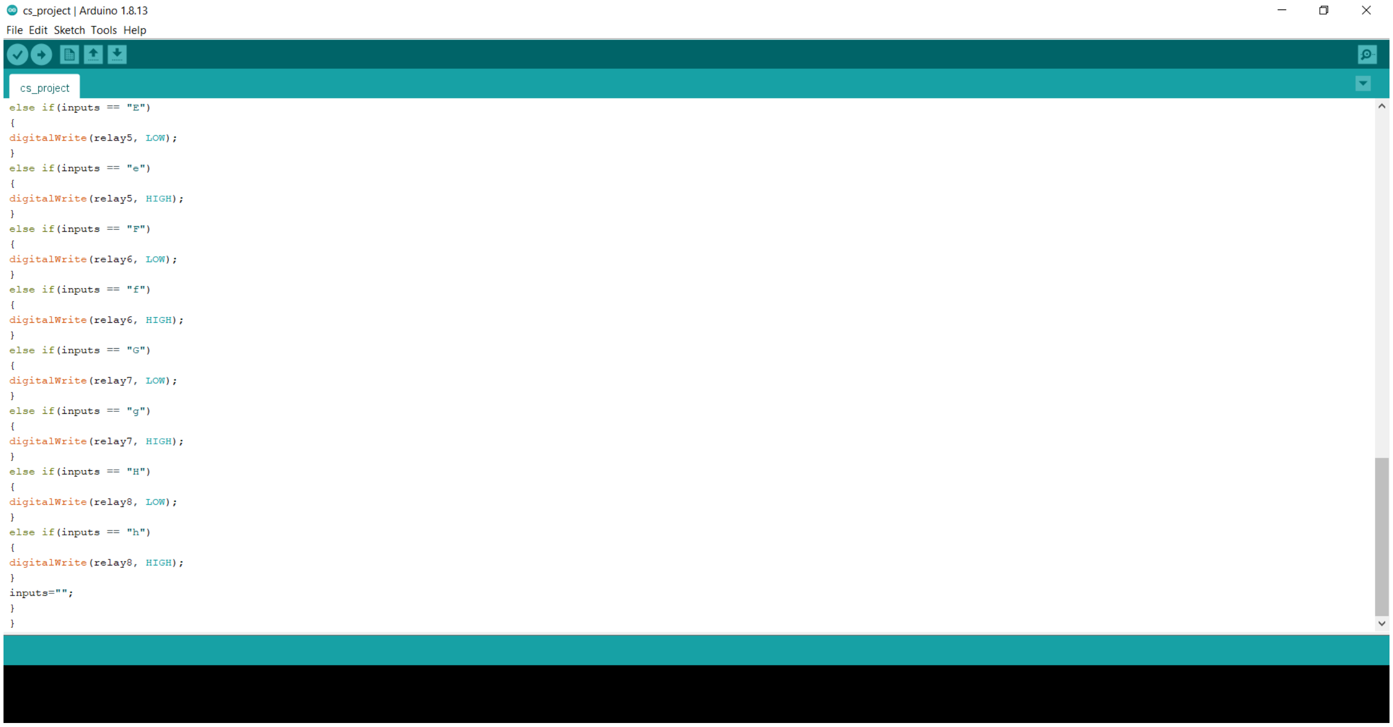


```
{
delay(10); //Delay to make it stable
char c = Serial.read(); //Conduct a serial read
if (c == '#'){
break; //Stop the loop once # is detected after a word
}
inputs += c; //Means inputs = inputs + c
}
if (inputs.length() >0)
{
Serial.println(inputs);

if(inputs == "A")
{
digitalWrite(relay1, LOW);
}
else if(inputs == "a")
{
digitalWrite(relay1, HIGH);
}
else if(inputs == "B")
{
digitalWrite(relay2, LOW);
}
else if(inputs == "b")
{
digitalWrite(relay2, HIGH);
}
else if(inputs == "C")
{
digitalWrite(relay3, LOW);
}
else if(inputs == "c")
{
digitalWrite(relay3, HIGH);
```



```
}
else if(inputs == "c")
{
digitalWrite(relay3, HIGH);
}
else if(inputs == "D")
{
digitalWrite(relay4, LOW);
}
else if(inputs == "d")
{
digitalWrite(relay4, HIGH);
}
else if(inputs == "E")
{
digitalWrite(relay5, LOW);
}
else if(inputs == "e")
{
digitalWrite(relay5, HIGH);
}
else if(inputs == "F")
{
digitalWrite(relay6, LOW);
}
else if(inputs == "f")
{
digitalWrite(relay6, HIGH);
}
else if(inputs == "G")
{
digitalWrite(relay7, LOW);
}
else if(inputs == "g")
{
```

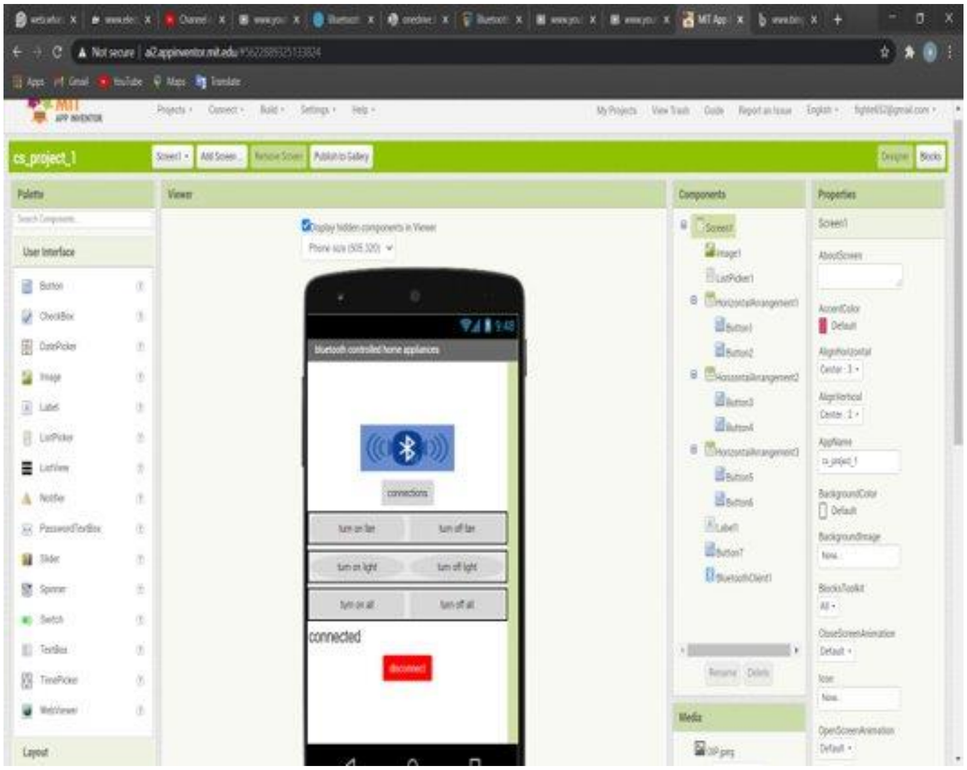


```
else if(inputs == "E")
{
digitalWrite(relay5, LOW);
}
else if(inputs == "e")
{
digitalWrite(relay5, HIGH);
}
else if(inputs == "F")
{
digitalWrite(relay6, LOW);
}
else if(inputs == "f")
{
digitalWrite(relay6, HIGH);
}
else if(inputs == "G")
{
digitalWrite(relay7, LOW);
}
else if(inputs == "g")
{
digitalWrite(relay7, HIGH);
}
else if(inputs == "H")
{
digitalWrite(relay8, LOW);
}
else if(inputs == "h")
{
digitalWrite(relay8, HIGH);
}
inputs="";
}
}
```

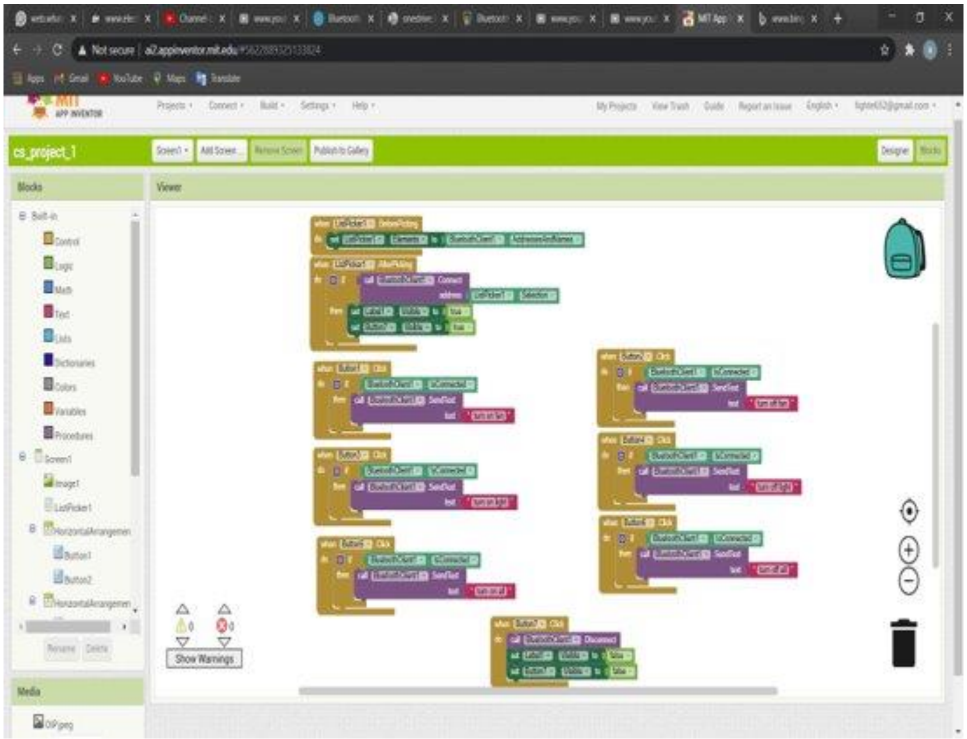
Mobile app (using MIT app inventor)

Basically, for this project we had made an android app using online software known to be MIT app inventor in this we can make the app according to what we need using block system like by joining the blocks we can verify that which steps is to be implement next. Then after we successfully make the interface and code part (using block) we can download that file in apk form and send it to our device where we can access it and run our model.

Interface



Code



Working Principle

Bluetooth module interfaced with microcontroller receives command from the android application installed on android device using Bluetooth technology (wireless communication). The program written to the microcontroller   communicates with the Bluetooth module serially to receive the commands. Microcontroller switches the loads automatically on the basis of commands received from the Bluetooth module.

Firstly, whenever the circuit is powered on, the Arduino module loads the required libraries and turn the relays to OFF position. At this time all appliances are in OFF position and none of our appliance is working. Now our Arduino module will wait for the numeric command to be received from the Bluetooth module (sent by our android device). All appliances are assigned numbers (example if there are four appliances connected 1 to 4 will be the no assigned to them). If either number (0 or 1) is obtained in the form of string from the Bluetooth module (HC-05), the status of the respective appliance will get toggled. As in default case, the pins connected to the relays they are having a LOW logic driving causing the relays module to switch the appliances OFF.

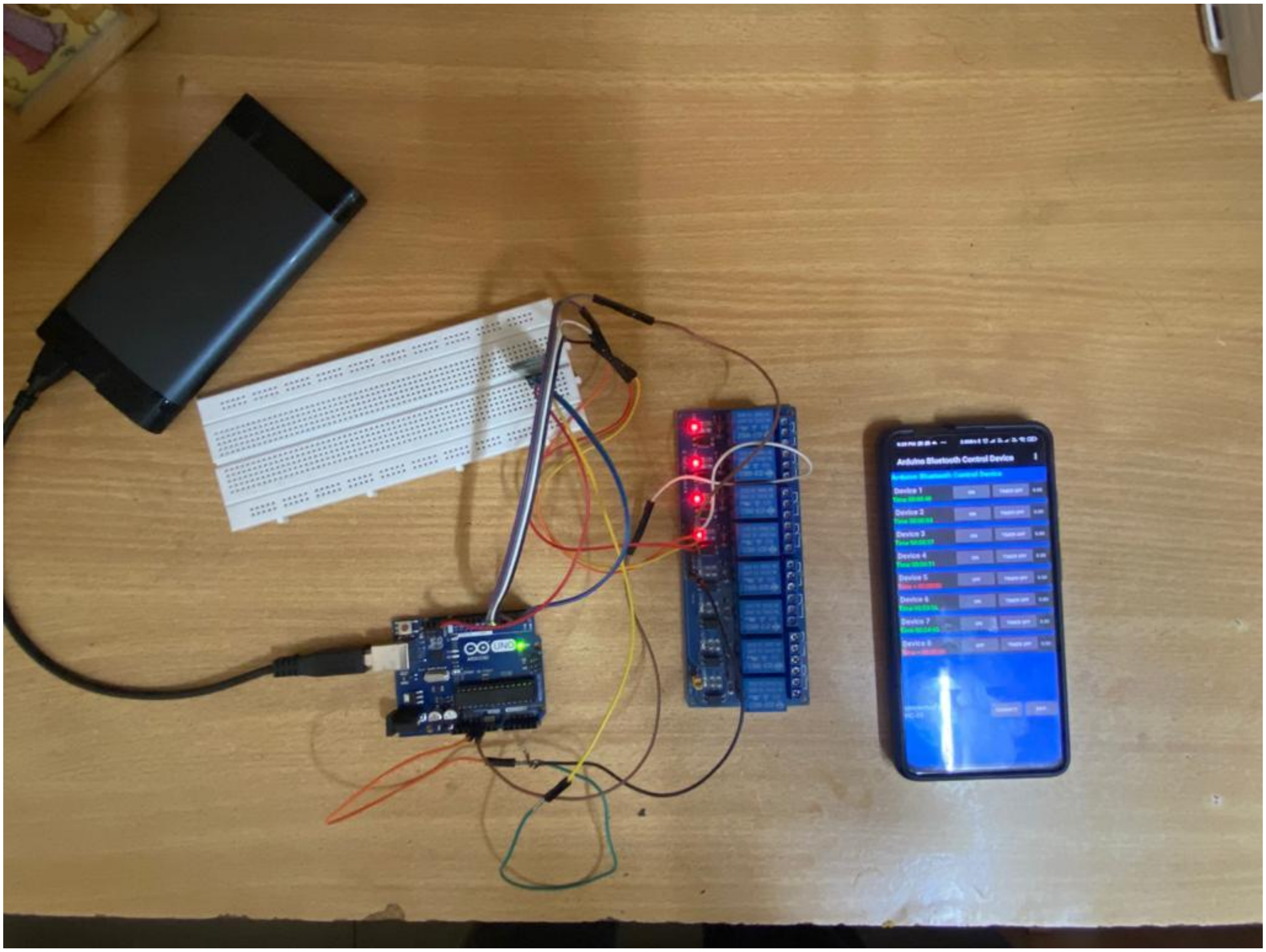
When an appliance is in OFF condition and we need to turn it ON a number representing it is passed through the Bluetooth app (command sent through android app), Now the Arduino module will switch the logic at the respective pin to HIGH triggering the relay module to switch ON the appliance. The change in the status of the appliance is updated on the LED on relay module indicating that relay is high and the appliance starts glowing due to forward biasing.

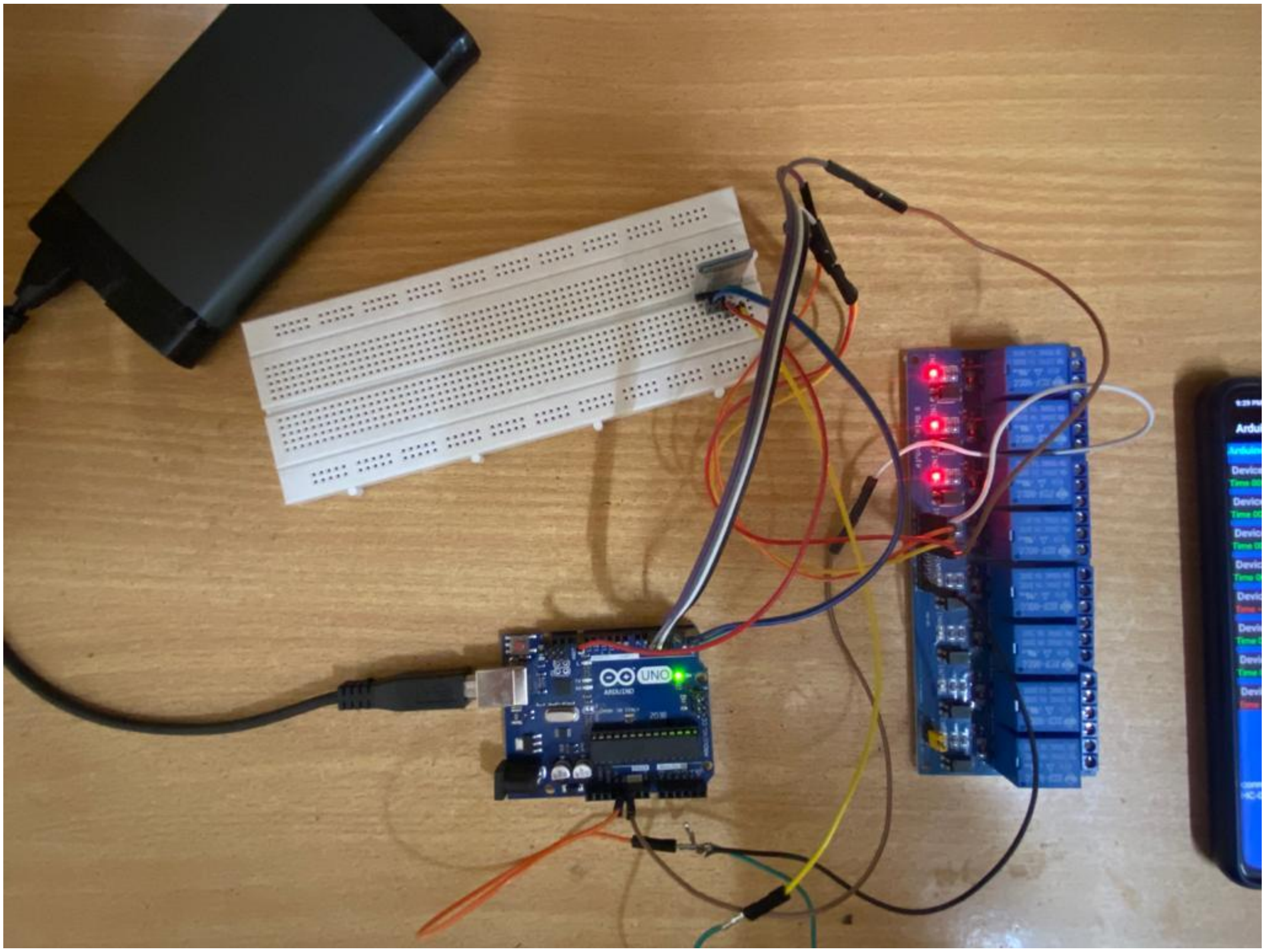
When an appliance is in ON condition and we need to turn it OFF a number representing it is passed through the Bluetooth app (command sent through android app), Now the Arduino module will switch the logic at the respective pin to LOW triggering the relay module to switch OFF the appliance. The change in the status of the appliance is updated on the LED on relay module indicating that relay is low and the appliance stops glowing due to lack of forward biasing.

The numbers are transferred to the Bluetooth module from the paired android phone. The android phone must be connected (paired) with the Bluetooth module.

Hardware Model

Here we are using our components (Arduino, relay, breadboard, HC-05, Power source, Android Device). Here for power source, we are using mobile power bank having output voltage of 9 volt . After that we have connected all the components according to our circuit diagram and then installed the Arduino code compiled using Arduino IDE to the main board and turned on our circuit giving power to it. Now we have to open the app which we had made using MIT app inventor in our Android device and connect it with the Bluetooth module of our circuit using Bluetooth technology in our device.Then we are ready to go now just press the button and switch the state of whatever appliances you want to turn ON or OFF.


Arduino Bluetooth Control Device
Device 1
Device 2
Device 3
Device 4
Device 5
Device 6
Device 7
Device 8
UNO
ARDUINO



Hardware model live working video

https://drive.google.com/drive/folders/1w-0Kcn9snfdYtTtEHFtPCe9QxDa3PvJ-?usp=sharing

Result

So, we successfully implemented the simulation and hardware model of the Bluetooth controlled electronic home appliances using Arduino and got the desired output as whenever we are sending the input of switching OFF or ON of electronic device which we want it is performing that action which you can see in the below attached video.

Future extension

In future we can add temperature sensors to monitor the surrounding temperature parameters. We can also use wireless camera to enhance the security features of the automated system.

References

1. https://smartify.in/knowledgebase/build-bluetooth-controlled-home-automation-setup-using-arduino/
2. https://www.electronicshub.org/bluetooth-controlled-electronic-home-appliances/
3. https://components101.com/wireless/hc-05-bluetooth-module
4. https://www.youtube.com/watch?v=gfK_C6ibGgY