DISSERTATION

On

**HOME APPLIANCES ROBOTIZATION**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF DEGREE OF**

**MASTER OF COMPUTER APPLICATIONS**

Under the Guidance- Submitted By-

Dr. Deepali Kamthania Bhaswati Kalita

Professor 42717704418

VSIT, VIPS MCA-6B



VIVEKANANDA SCHOOL OF INFORMATION TECHNOLOGY

VIVEKANANDA INSTITUTE OF PROFESSIONAL STUDIES

**Affiliated to**

(GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY,DELHI)

**ACKNOWLEDGEMENT**

It gives me great pleasure & satisfaction to present my dissertation titled **“HOME APPLIANCES ROBOTIZATION”.** The completion of any task is not only the reward to the person actively involved in accomplishing it, but also the person involved in inspiring & guiding. I am highly indebted to my supervisor **Dr. Deepali Kamthania** for his/her invaluable support and guidance throughout the work.

I extended my heartfelt thanks to Prof. Supriya Madan, Dean of Vivekananda School of Information Technology for his support without which the work would have never been realized. Last but not the least; I would like to thank all friends who directly or indirectly helped me in completing my work.

Student’s Name: Bhaswati Kalita

Roll No: 42717704418



Vivekananda Institute of Professional Studies

AU Block, Outer Ring Road, Pitampura, Delhi- 110088

(NAAC Accredited ‘A’ Grade & ISO 9001:2008 Certified Institution)

***(Affiliated to GGSIP University & Bar Council of India)***

**Website:www.vips.edu**

Dated:

**DECLARATION**

I hereby certify that the dissertation titled **“HOME APPLIANCES ROBOTIZATION”** Submitted by **Ms Bhaswati Kalita (Class Roll No 42717704418),** in partial fulfillment of requirements for the award of degree of **Master of Computer Application** submitted in the School of Information Technology at **Vivekananda Institute of Professional Studies, Delhi** affiliated to **GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, Delhi** is an authentic record of my work carried out under the supervision of **Dr. Deepali Kamthania** The matter presented herein has not been submitted earlier by me in any other University / Institute for the award of any Degree.

**2021-7-26_10-15-7Name & Sign of the student**

**Bhaswati Kalita ( )**

(42717704418)

**…………………………….** **(Prof. Supriya Madan)**

**(Name of Project Supervisor) Dean (VSIT)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**HOME APPLIANCES ROBOTIZATION**

**Abstract:**

The home appliances robotization system based on the Android smartphone GUI is designed on Android Smartphones. A user logs into the smart Android phone interface, and clicks the buttons gently to send message commands from the GUI which will be transmitted to the relay board connected via arduino board. Then the arduino board processor recognizes the specified command, and controls the home appliance switches in the wireless radio frequency manner to achieve remote control of appliances ultimately. This system is designed in order to fulfil the needs of elderly and imprudent at home. Also, the smart home idea in the system developed the standard of living at home.

The main control system equipment’s wireless Bluetooth technology provides remote access from Android Phone. The design remains the subsist electrical switches and provides more safety control on the switches with low voltage activating method. The switches status is synchronized in all the control systems whereby every user interface indicates the real time existing switches status. The system intended to control electrical appliances and devices in the house with relatively low cost design, user-friendly interface and ease of installation. The users can manipulate appliances anytime, anywhere, letting our houses become more and more automated and intelligent. There are some problems in the PC monitor terminal, such as its great bulk, inconvenience to carry, high cost, limited monitoring range and so on. Therefore, it’s a good choice to design a terminal based on a phone.

**Keywords**: Android phone, Bluetooth, Wireless switch, Relay, Arduino.

**1. Introduction:**

Home appliances robotization is an automation of home, house work or household activity .In other words, it gives an opportunity to use a computer to control electrical devices.For example, It includes centralized control of ceiling fans. It can include centralized control of lighting application security and lack of gates and doors to provide flourish benefit case energy ability and security.

In the present, home automation is being popular due to its comfort, ability, flexible means of monitoring and controlling the electrical devices and other things according to user’s comfort and needs. The challenging part lies in the simplicity and cost of installing them in homes and various services with an increasing number of services to be monitored and controlled. The project named “**HOME ROBOTIZATION**” is an idea of home automation using android phones. The popularity of home automation has been rising comprehensively because of considerable capability and simplicity through android phone and tablet connectivity. A home automation system integrates electrical devices with each other. The techniques applied in home automation include those in building automation and the control of house old activities such as lighting control system and the use of other electrical appliance devices may be connected via a home network to offer the restraint by a pc and may permit remote access from the internet. Through the integration of information technologies with the home environment, method and appliance can be connected in a mobilized process which results in convenience, power efficiency and security benefits.

The android running OS in any phone connected to bluetooth can access the status of home appliances via an application. It displays the design and implementation of automation processes that can guide and control home appliances via android phone or tablet. Just see the figure like a simple smart home. Figure 1 shows the Home Automation by controlling Phone.



**Figure 1: Home Appliances Robotization System**

**2. Related Work:**

Day by day the work and life of human beings are increasingly busy and complicated with the rapid growth in communications and information technology. As the economic expansion is growing rapidly, the standard of living also keeps on rising up and the people are requiring more living functions. The concept of smart home has focused the attention of researchers. A lot of efforts have been made for the development of home automation to control that remotely. Smart Home is an advanced technology to make a house intelligent and automated. Usually, that technology has automation systems for lighting, temperature control, security and many other functions.

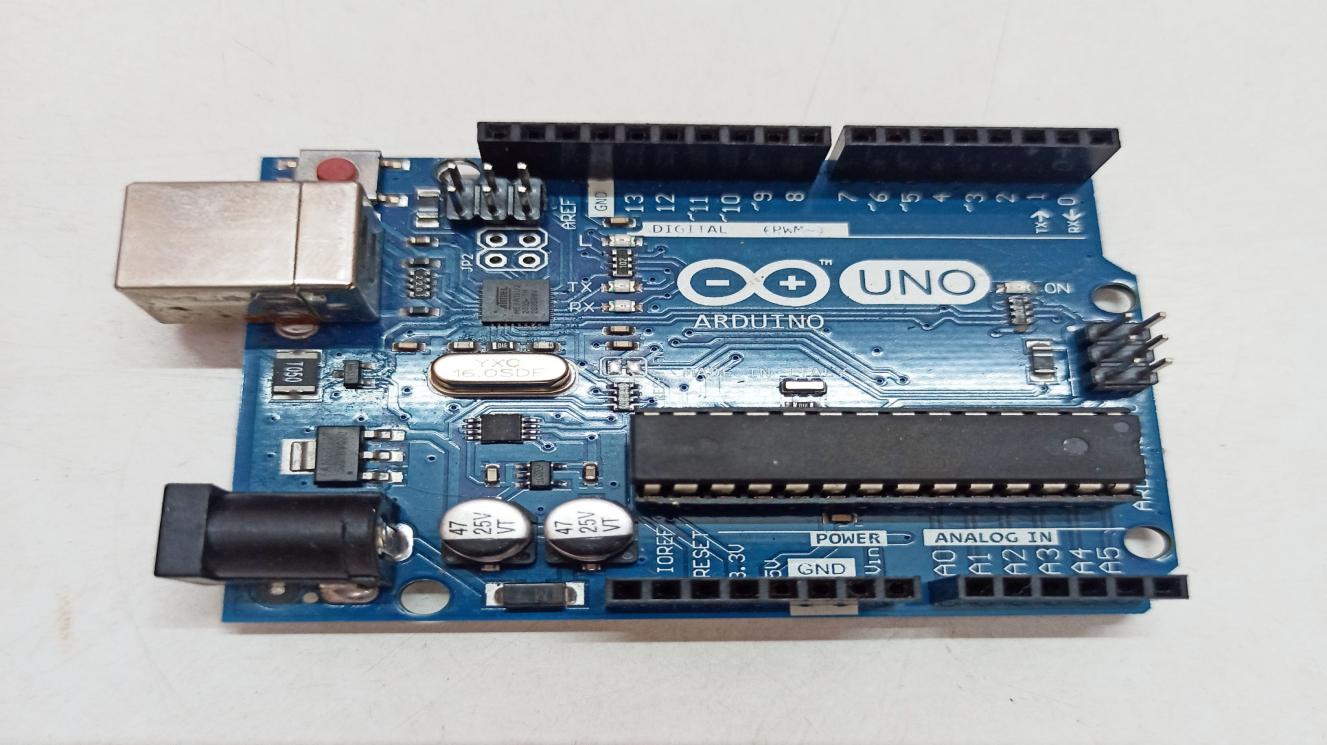
We have gone through different techniques for the implementation of smart home such as; phone-based remote controller for home and office automation, PC remote control of appliances by using telephone lines, [3] Blue-tooth wireless technology based home automation, internet based wireless home automation system, remote home automation monitoring using mobile through spoken commands, GSM-based remote sensing and control system using FPGA, GSM-Bluetooth based remote monitoring and control system with automatic light controller. [1]

Home automation gives you access to control devices in your home from a mobile device anywhere in the world. The term may be used for isolated programmable devices, like thermostats and sprinkler systems, but home automation more accurately describes homes in which nearly everything -- lights, appliances, electrical outlets, heating and cooling systems -- are hooked up to a remotely controllable network. From a home security perspective, this also includes your alarm system, and all of the doors, windows, locks, smoke detectors, surveillance cameras and any other sensors that are linked to it. [2].

**3. Devices used:**

**3.1 Arduino:**

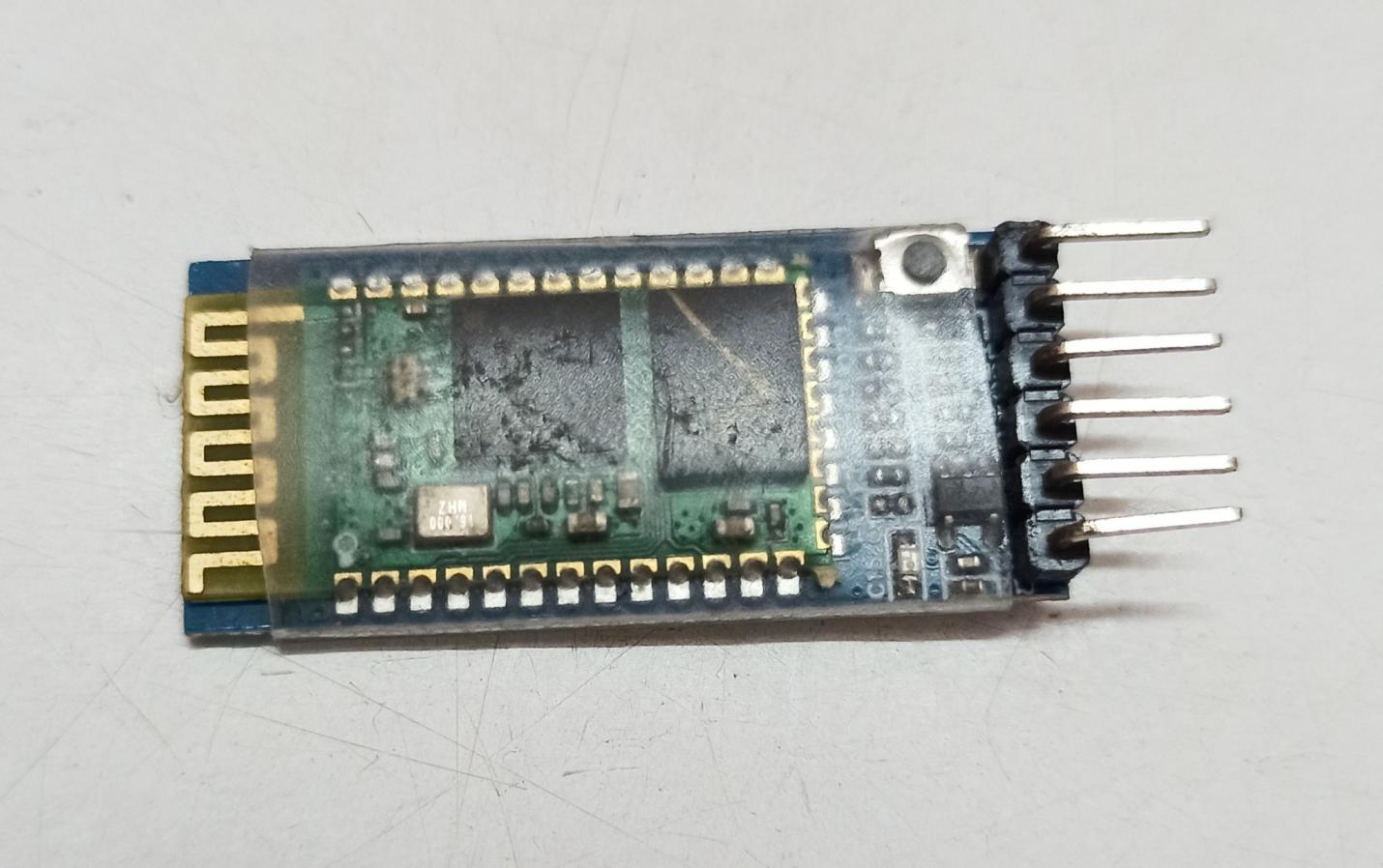
For making the project (Home appliances robotization System) a board must be needed: an arduino board. Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world.



**Figure 2: Arduino UNO board**

**3.2 Bluetooth module:**

HC-05 Bluetooth Module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. Its communication is via serial communication which makes an easy way to interface with a controller or PC. HC-05 Bluetooth module provides switching mode between master and slave mode which means it is able to use neither receiving nor transmitting data.



**Figure 3: Bluetooth module**

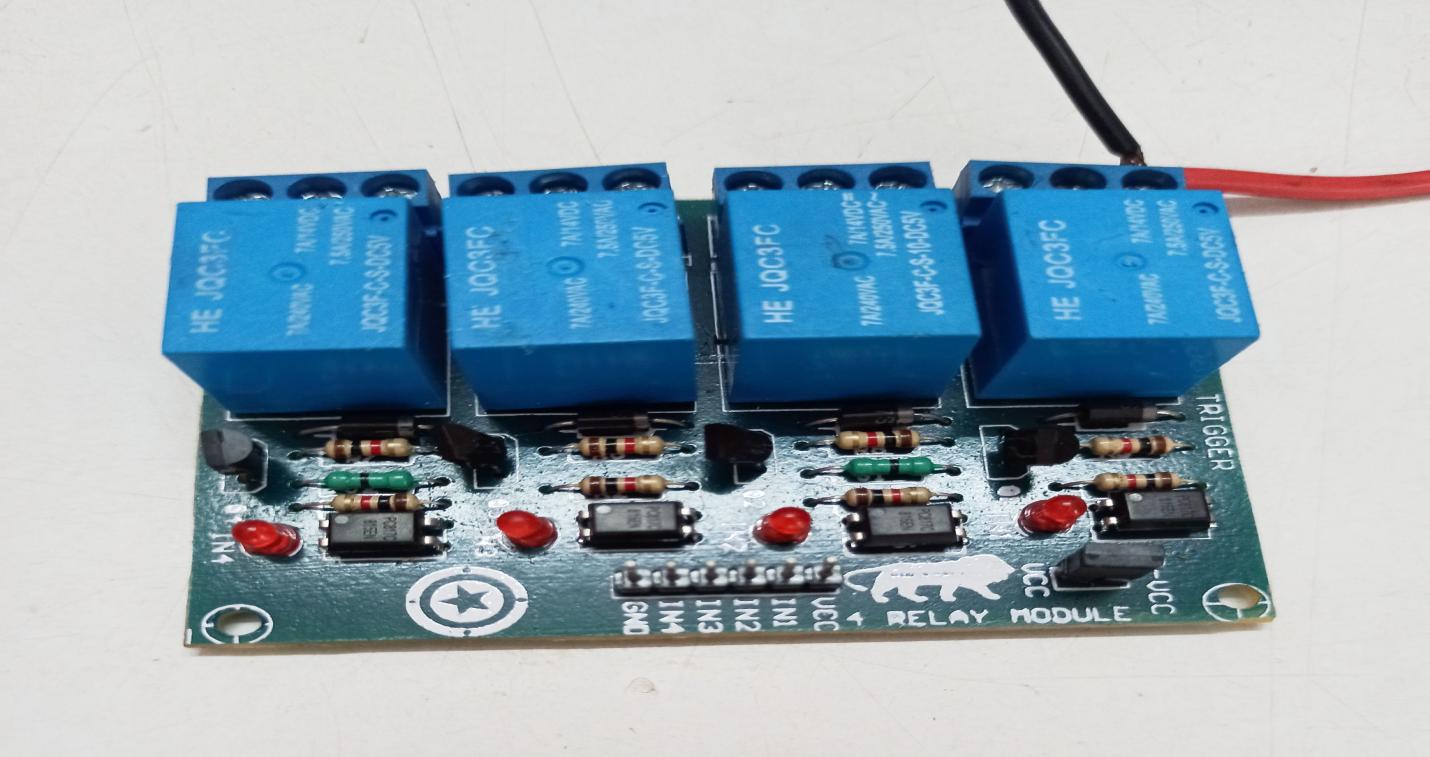
**Specification:**

* Model: HC-05
* Input Voltage: DC 5V
* Communication Method: Serial Communication
* Master and slave mode can be switched

**3.3 Relay:**

A **relay** is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. Relays are used where it is necessary to control a circuit by a low-power signal (with complete electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits as amplifiers: they repeated the signal coming in from one circuit and re-transmitted it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

For example, when you push the button on a TV remote to watch TV, it sends an electrical signal to the “relay” inside the TV, turning the main power ON. There are various types of relays used in many applications to control different amounts of currents and number of circuits.

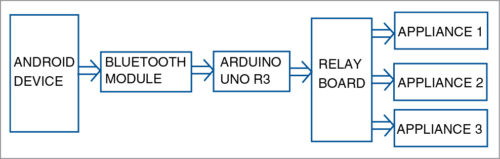


**Figure 4: Relay board**

**4. System Architecture:**

**4.1 System overview**

The system is composed of android mobile, Bluetooth module, home information center and wireless switch. And the architecture of the system is shown in Fig. 5 at any moment; the GUI in the phone allows the user to manually control any of the applications in the room. [4] First of all, the action listener should be set for each button which will be clicked to send a message command via the bluetooth network. After reading and parsing the commands, the PIC processor controls the wireless module to send the address and data codes to achieve remote control of appliances ultimately.

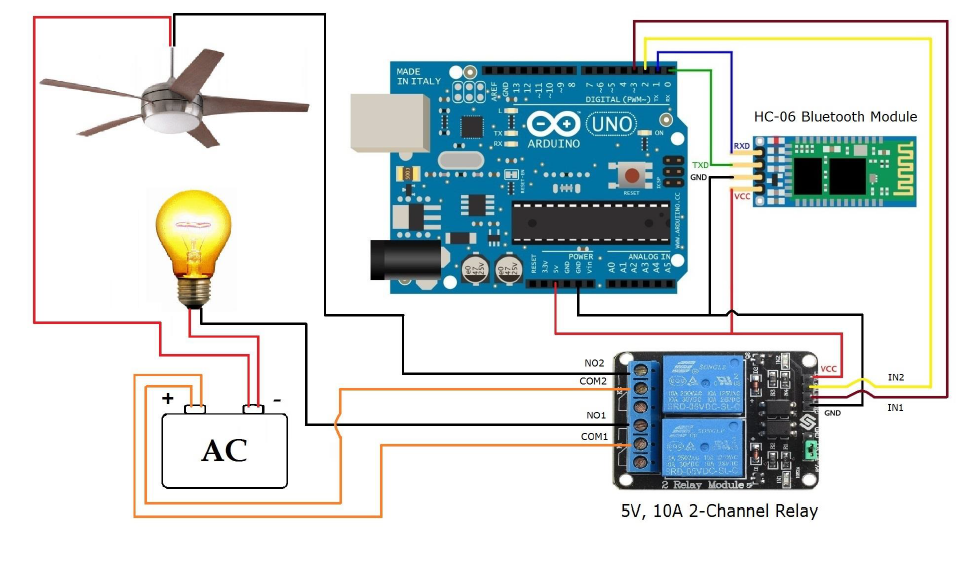


**Figure 5: Functional Block Diagram of the System.**

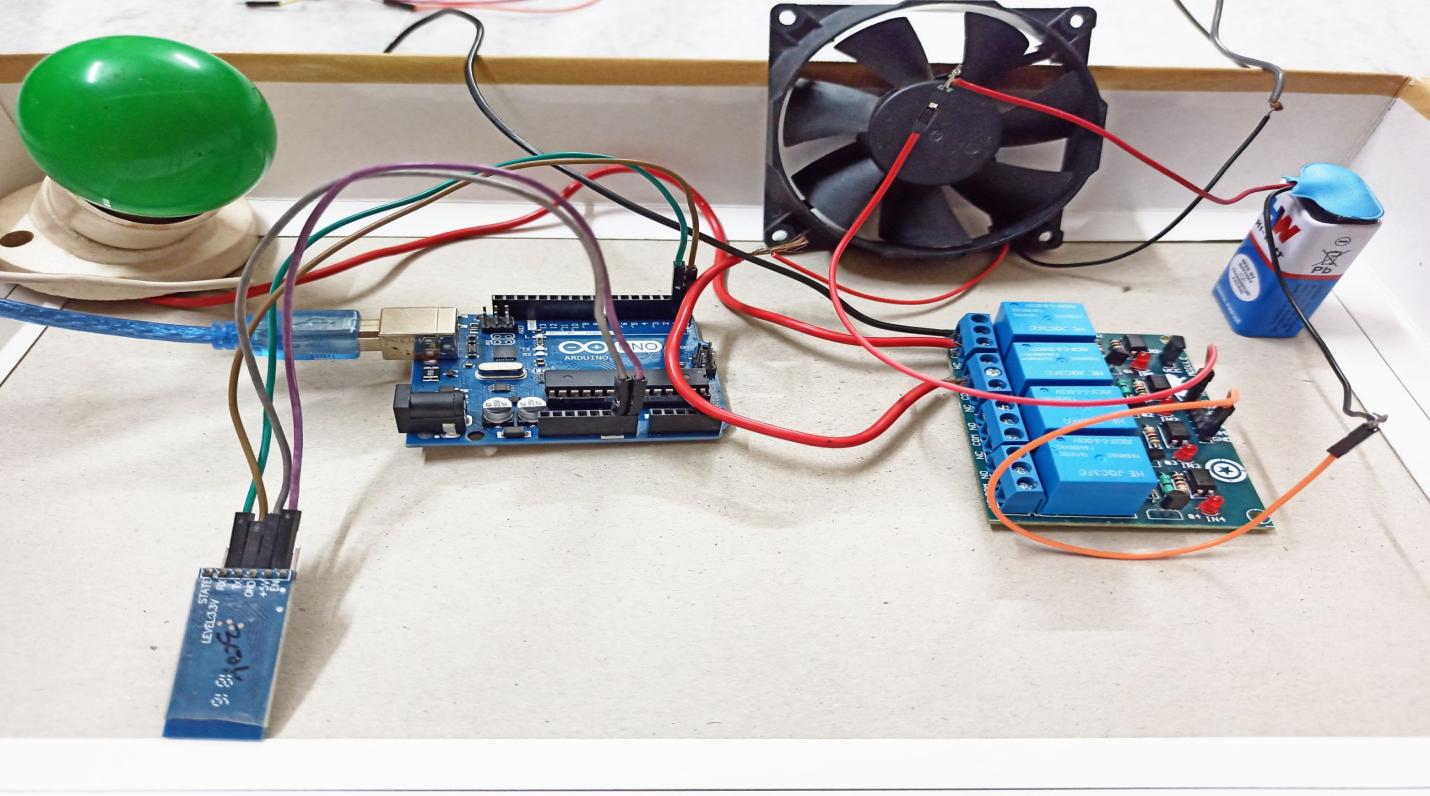
The terminal based on android phone is designed, which contains GUI design, user management and message command sending event. After entering the control page interface, users send predefined commands just by touching the appropriate button. The user management includes user login with a hardcoded name “**admin**” and password “**1234**”. The user login is to create a legitimate user, and give him the appropriate permissions.

**4.2 Hardware design:**

This section mainly discusses the hardware construction of the main control board. Fig 6 demonstrates the hardware block diagram in the main control board. For the Bluetooth module, a low cost HC-05 Bluetooth module is chosen to establish the Bluetooth connection between the main control board and the GUIs.



**Figure 6: Main Control Board Hardware Block Diagram**

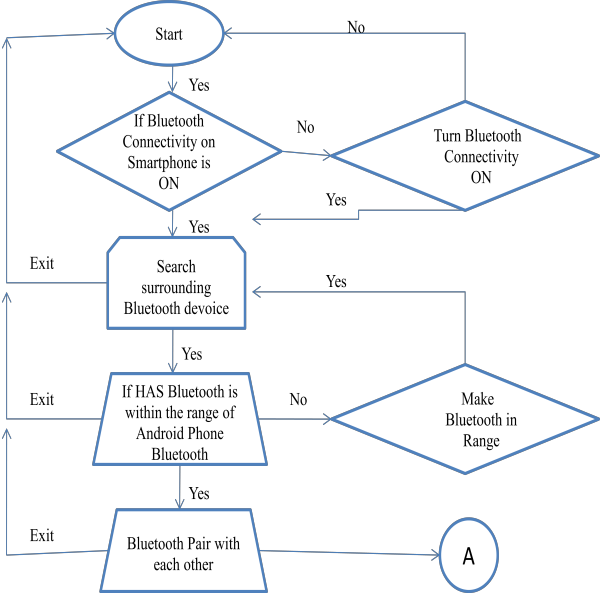
****

**Figure 7: Main Control Board Hardware design**

The electrical current is directly connected to the main control board whereby it separates the regulator and relay circuit. The voltage regulator is constructed by a common reliable regulator circuit which consists of transformer, rectifier and regulator. 5V and 3.3V DC output is regulated in order to fulfill the voltage needs of the specific components in the main control board. Moreover, the low voltage activating switches will replace the existing switches.

**4.3 Software design:**

Figure (8) shows the Bluetooth module and Android Smartphone connection flowchart, which show that transfer of data from Android application to Bluetooth Module after which is compare to programming code of AVR Microcontroller First check Bluetooth position of Android SmartPhone (Bluetooth is ON or OFF). If Bluetooth of an android SmartPhone is ON, then it checks the surrounding Bluetooth Module and if Bluetooth Connection is off then the user has to start the Bluetooth condition from OFF to ON of the Android Smartphone. Then the Smartphone searches for the surrounding Bluetooth Module. After searching the Bluetooth Module, it starts pairing between Bluetooth Module and Android Smartphone. And if pairing occurs then hardware initializes the serial port. And Then PCB Hardware gets ready to receive the data (Character) which is sent by the Android Application software.



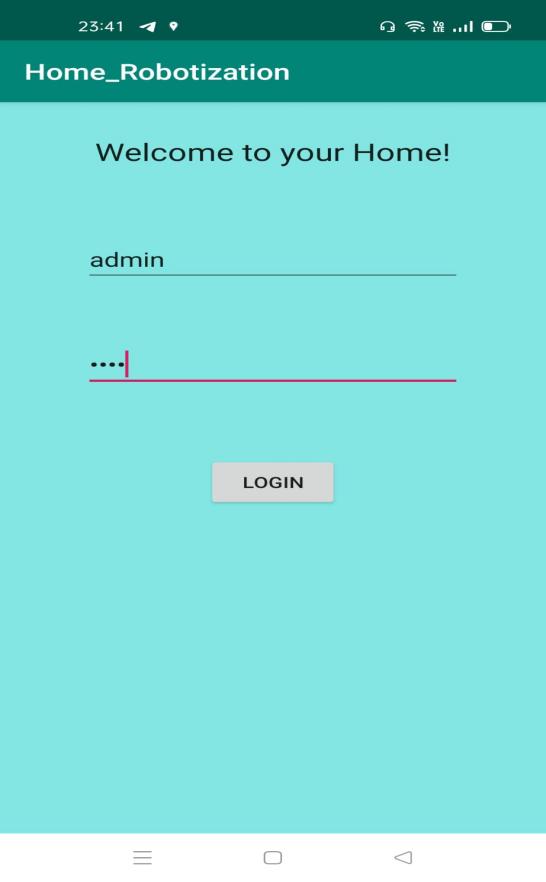
**Fig.7: A Flowchart for Bluetooth Module and Android Smartphone connection.**

When the application software sends the Text character to the microcontroller then this character gets compared with the internal Bourne code of the controller and accordingly the respective relay gets ON and OFF. The passion of relay depends on receiving character. if A or B is received by Bluetooth Receiver then Light is ON for turning On the UI button and Light OFF for Turning OFF the UI button. This is applicable for the entire button on the UI. Depending on the Button press, each character gets sent by Android Smartphone.

**4.4 Android Application:**

To control the relays with the help of a microcontroller relay is connected to the 4 th port of the Microcontroller. By pressing the button on the GUI of the Android Smartphone we can control the relay.

**App screenshots:**

** **

** **

****

Android's releases prior to 2.0 (1.0, 1.5, 16) were used exclusively on Android mobile phones. Most Android SmartPhone and some Android tablets now use a 2.x release and Android 3.0 was a tablet-oriented release but does not officially run on mobile phones. The current Android version is 11. Android's releases are nicknamed after sweets or dessert items like Frozen Yogurt ("Froyo") (2.2), Ginger Bread (2.3), Honeycomb (3.0), Ice Cream Sandwich (4.0), Jelly Bean (4.1), KitKat (4.4), Lollypop (5.0), Marshmallow (6.0), Nougat (7.0), Oreo (8.0), pie (9.0), Android 10 and Android 11 being the recent one.

Android applications are written in the Java language. The Android Software Development Kit (SDK)provides all necessary tools to develop Android applications (API). This includes a compiler, debugger and device emulator, as well as its own virtual machine (emulator) to run Android programs.

The Android SDK provides tools for code compilation and packaging data and resource files into an archive file with “.apk” extension called as an Android package. Android devices used the “.apk” file to install the application. Android's application framework allows for the creation of extremely feature rich and novel applications by using a set of reusable components. The amalgamation of the Android development environment with the Bluetooth wireless technology is known by Android’s support for the Bluetooth network stack, which permits a device to wirelessly exchange data with another Bluetooth device (SmartPhone Bluetooth with Bluetooth Module). The application framework enables access to the Bluetooth functionality using the Android Bluetooth APIs. These APIs allow wireless applications to connect to other Bluetooth devices for point-to-point and multipoint wireless features.

Using the Bluetooth APIs, an Android application can carry out the following functions:

· Scrutinize for other Bluetooth devices

· Enquire about the local Bluetooth adapter for pairing. Bluetooth devices

· Connect to other devices through service discovery.

· Exchange data to and from other devices

**5. Applications**

Following are the applications of home appliances robotization systems based on the android SmartPhone.

**5.1 Lighting appliance control subsystem**

Household appliances also don’t need to transmit large amounts of data, and its real-time requirement is not high, so take a combination of wired and wireless ways to make wiring easy. Alarm subsystem with landline phone alarm and GSM alarm has multiple protections. It can still alarm normally when the line of the landline phone is cut off. The entire system takes modular design thinking which contributes to the design clear and facilitates the user to select a different combination of modules to meet the needs of individual users.

Humanized operation interface allows users to use them more conveniently. Dimming control, a more comfortable environment and a certain scene was created. Appliance control function is not only our common household appliances such as microwave ovens, water dispensers, rice cookers, television sets, but also includes garden automatic irrigation systems, fountains and other equipment. Lighting appliance control subsystems can be achieved not only for a variety of romantic scenes, but also for the whole family safely. When you go out, just one lobby button, you can complete power outage, which significantly reduces the risk of fire and saves energy

**5.2 Curtain control subsystem**

Curtain control subsystems control all electric curtains including blinds, sunshades and skylights. It can automatically open and close in a particular time or at the intensity of light. You can control it by panel, remote control, and even the Internet and smartphones.

**6. Conclusions**

By designing the Android user interface and Home appliances robotization system, a home appliance control system based on the Android phone can be designed. It has combined android client, network transmission, and wireless switch, home appliances robotization to form a complete system, and the whole system works normally. Identifying message commands and wireless encoding are the two major tasks for home robotization systems. Android phones have advantages such as humane interface, customizable and extensible applications and android phone is easy to carry. By constantly improving the control function, the android phone allows us anytime, anywhere to control any device, and finally realizes the highly intelligent home.

**7. Future scope**

This project can be further developed by integrating it with the internet to monitor your home while sitting in a remote area. By doing this, one can keep an eye on his or her home through an internet connection to the user’s mobile phone or PC or laptop. This will not only improve the security of your home in this modern day world but will also assist in conservation of energy like if you left any home appliance switched on by mistake, then you can check the status of the appliance on the graphical interface made on your mobile and can switch it off using the internet connectivity.

As per our survey, there exist many systems that can control home appliances using android based phones/tablets. Each system has its unique features. Currently certain companies are officially registered and are working to provide better home automation system features. Following models describes the work being performed by others. N. Sriskanthan [5] explained the model for home automation using Bluetooth via PC. But unfortunately the system lacks to support mobile technology. Muhammad Izhar Ramli [6] designed a prototype electrical device control system using Web.

They also set the server with auto restart if the server condition is currently down. Hasan [5] has developed a telephone and PIC remote controlled device for controlling the devices pin check algorithm has been introduced where it was with cable network but not wireless communication. Amul Jadhav [7] developed an application in a universal XML format which can be easily ported to any other mobile devices rather than targeting a single platform. Pratik Gadtaula of Telemark University College, Faculty of Technology has done a Master’s thesis on “Home Automation” [8].

**8.References**

[1].Ki-Cheol Son, Jong-Yeol Lee. "The methods of android application speed up by using NDK". Awareness Science and Technology (iCAST), 2011 3rd International Conference on. Sept 2011, pp.382- 385.

[2]. Xiao Yuan, Yuliang Pan, Zaiying Ling. "The Application of Infrared Remote Controlled Code Lock in the Management of Industrial Machine Parameters” .Electrical and Control Engineering (ICECE), 2011 International Conference on.2011, pp. 418-421.

[3].Jianjun Lv, Zhishu Li, Mingyi Mao. "A new USB home appliances based on PC and infrared remote control protocol".2010 International Conferences on Computer and Communication Technologies in Agriculture Engineering.2010, pp.572 -575.

[4].Shengwen Chen, Chunghuang Yang, Chung-Huang Yang.“Design and Implementation of Live SD Acquisition Tool in Android Smart Phone". 2011 Fifth International Conference on Genetic and Evolutionary Computing. 2011, pp. 157-162.

[5]. Muhammad Izhar Ramli, Mohd Helmy Abd Wahab, Nabihah, “TOWARDS SMART

HOME: CONTROL ELECTRICAL DEVICES ONLINE”, Nur Nabihah Ahmad International

[6] E. Yavuz, B. Hasan, I. Serkan and K. Duygu. “Safe and Secure PIC Based Remote Control

Application for Intelligent Home”. International Journal of Computer Science and Network

Security, Vol. 7, No. 5,[april 1,2018 at10.pm]

[7] Amul Jadhav, S. Anand, Nilesh Dhangare, K.S. Wagh “Universal Mobile Application

Development (UMAD) On Home Automation” Marathwada Mitra Mandal’s Institute of

Technology, University of Pune, India Network and Complex Systems ISSN 2224-610X (Paper)

ISSN 2225-0603 (Online) Vol 2, No.2,[ last accesses april 2,2018 at 10 pm]

[8] Pratik Gadtaula, “Home Automation”, Telemark University College, Faculty of

Technology, Master’s Thesis, [last accesses April 4, 2018 at 10.pm]

**Android Code:**

**Page1: (MainActivity)**

**.java**

package com.example.home\_robotization;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import java.util.Timer;

import java.util.TimerTask;

public class MainActivity extends AppCompatActivity {

protected Timer timer;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

timer=new Timer();

timer.schedule(new TimerTask() {

@Override

public void run() {

Intent intent=new Intent(MainActivity.this,second\_page.class);

startActivity(intent);

finish();

}

},3000);

}

}

**.xml**

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#13A3AF"

tools:context=".MainActivity">

<TextView

android:layout\_width="316dp"

android:layout\_height="135dp"

android:layout\_marginTop="172dp"

android:text="@string/welcome\_to\_home\_appliances\_robotization"

android:textSize="30sp"

android:textStyle="bold"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

**Page2: (SecondPage)**

**.java**

package com.example.home\_robotization;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.EditText;

import android.widget.Toast;

import android.content.Intent;

public class second\_page extends AppCompatActivity {

EditText usr, pwd;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_second\_page);

}

public void userLogin(View view) {

//local log in

usr = (EditText) findViewById(R.id.username);

pwd = (EditText) findViewById(R.id.password);

if (usr.getText().toString().equals("admin") && pwd.getText().toString().equals("1234")) {

Toast.makeText(getApplicationContext(), "Correct!", Toast.LENGTH\_SHORT).show();

Intent myIntent = new Intent(second\_page.this, bluetooth.class);

second\_page.this.startActivity(myIntent);

} else {

Toast.makeText(getApplicationContext(), "Wrong!", Toast.LENGTH\_SHORT).show();

}

}

}

**.xml**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#7A02D1CA"

tools:context=".second\_page">

<EditText

android:layout\_width="250dp"

android:layout\_height="wrap\_content"

android:id="@+id/username"

android:layout\_marginTop="120dp"

android:layout\_gravity="center\_horizontal"

android:hint="Username"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true" />

<EditText

android:layout\_width="250dp"

android:layout\_height="wrap\_content"

android:id="@+id/password"

android:layout\_marginTop="50dp"

android:layout\_gravity="center\_horizontal"

android:hint="Password"

android:inputType="textPassword"

android:layout\_below="@+id/username"

android:layout\_alignStart="@+id/username" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textAppearance="?android:attr/textAppearanceLarge"

android:text="Welcome to your Home!"

android:layout\_marginTop="30dp"

android:id="@+id/textView"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Login"

android:id="@+id/login\_button"

android:layout\_below="@+id/password"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="60dp"

android:onClick="userLogin" />

</RelativeLayout>

**Page3: (Bluetooth)**

**.java**

package com.example.home\_robotization;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import com.suke.widget.SwitchButton;

import android.bluetooth.BluetoothAdapter;

import android.bluetooth.BluetoothDevice;

import android.bluetooth.BluetoothSocket;

import android.content.Intent;

import android.content.SharedPreferences;

//import android.support.v7.app.AppCompatActivity;

import android.util.Log;

import android.view.View;

import android.widget.TextView;

import android.widget.Toast;

import java.io.IOException;

import java.io.OutputStream;

import java.util.UUID;

public class bluetooth extends AppCompatActivity {

private static final String TAG = "BluetoothHomeAutomation";

private BluetoothSocket socket;

private BluetoothDevice device;

private OutputStream os;

private BluetoothAdapter mBluetoothAdapter;

public String deviceName;

public String deviceAddress;

UUID MY\_UUID = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");

TextView nameView;

SwitchButton device1, device2, device3, device4, device\_all;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_bluetooth);

nameView = (TextView)findViewById(R.id.controllerName);

device1 = (SwitchButton) findViewById(R.id.device1\_toggle);

device2 = (SwitchButton) findViewById(R.id.device2\_toggle);

device3 = (SwitchButton) findViewById(R.id.device3\_toggle);

device4 = (SwitchButton) findViewById(R.id.device4\_toggle);

device\_all = (SwitchButton) findViewById(R.id.device\_all\_toggle);

mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

if (mBluetoothAdapter == null) {

Toast.makeText(getApplicationContext(),R.string.Bluetooth\_NA, Toast.LENGTH\_LONG).show();

}

else if (!mBluetoothAdapter.isEnabled()) {

Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION\_REQUEST\_ENABLE);

startActivity(enableBtIntent);

}

device1.setOnCheckedChangeListener(new SwitchButton.OnCheckedChangeListener() {

@Override

public void onCheckedChanged(SwitchButton view, boolean isChecked) {

if (isChecked) {

sendSignal(1);

}

else {

sendSignal(5);

}

}

});

device2.setOnCheckedChangeListener(new SwitchButton.OnCheckedChangeListener() {

@Override

public void onCheckedChanged(SwitchButton view, boolean isChecked) {

if (isChecked)

sendSignal(2);

else

{

sendSignal(6);

}

}

});

device3.setOnCheckedChangeListener(new SwitchButton.OnCheckedChangeListener() {

@Override

public void onCheckedChanged(SwitchButton view, boolean isChecked) {

if (isChecked)

sendSignal(3);

else

{

sendSignal(7);

}

}

});

device4.setOnCheckedChangeListener(new SwitchButton.OnCheckedChangeListener() {

@Override

public void onCheckedChanged(SwitchButton view, boolean isChecked) {

if (isChecked)

sendSignal(4);

else

{

sendSignal(8);

}

}

});

device\_all.setOnCheckedChangeListener(new SwitchButton.OnCheckedChangeListener() {

@Override

public void onCheckedChanged(SwitchButton view, boolean isChecked) {

if (isChecked) {

device1.setChecked(true);

device2.setChecked(true);

device3.setChecked(true);

device4.setChecked(true);

}

else {

device1.setChecked(false);

device2.setChecked(false);

device3.setChecked(false);

device4.setChecked(false);

}

}

});

}

@Override

protected void onResume() {

super.onResume();

SharedPreferences preferences = getSharedPreferences("preferences",MODE\_PRIVATE);

deviceName = preferences.getString("controllerName", "HCHC05");

deviceAddress = preferences.getString("controllerAddress", "");

nameView.setText("Connected to " + "\"" + deviceName + "\"");

if(!deviceAddress.equals("")) {

device = mBluetoothAdapter.getRemoteDevice(deviceAddress);

try {

socket = device.createInsecureRfcommSocketToServiceRecord(MY\_UUID);

} catch (IOException e) {

//nameView.setText("Connected to " + "\"" + "NA" + "\"");

(getApplicationContext(), e.getMessage(), Toast.LENGTH\_LONG).show();

}

mBluetoothAdapter.cancelDiscovery();

if(socket != null){

try {

socket.connect();

} catch (IOException e) {

//nameView.setText("Connected to " + "\"" + "NA" + "\"");

(getApplicationContext(), e.getMessage(), Toast.LENGTH\_LONG).show();

Log.e(TAG, "socket connect failed: " + e.getMessage() + "\n");

try {

socket.close();

} catch (IOException e1) {

Log.e(TAG, "socket closing failed: " + e1.getMessage() + "\n");

(getApplicationContext(), e1.getMessage(), Toast.LENGTH\_LONG).show();

}

}

}

if (socket != null){

try {

os = socket.getOutputStream();

} catch (IOException e) {

Log.e(TAG, "getting output stream failed: " + e.getMessage() + "\n");

(getApplicationContext(), e.getMessage(), Toast.LENGTH\_LONG).show();

}

}

}

else {

Toast.makeText(getApplicationContext(),"Connect to a device",Toast.LENGTH\_LONG).show();

}

}

@Override

public void onPause() {

super.onPause();

if (os != null && socket.isConnected()) {

try {

os.flush();

} catch (IOException e) {

Log.e(TAG, "flushing output stream failed: " + e.getMessage() + "\n");

Toast.makeText(getApplicationContext(), e.getMessage(),Toast.LENGTH\_LONG).show();

}

}

try {

if(socket != null)

socket.close();

} catch (IOException e) {

Log.e(TAG, "closing socket failed: " + e.getMessage() + "\n");

Toast.makeText(getApplicationContext(), e.getMessage(),Toast.LENGTH\_LONG).show();

}

}

public void sendSignal(int message) {

if(socket != null){

if(socket.isConnected()) {

try {

if (os != null){

os.write(message);

Log.d("OS Message : ", "" + message);

}

} catch (IOException e) {

Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH\_LONG).show();

}

}

else {

(getBaseContext(), "Connect to the selected bluetooth device first", Toast.LENGTH\_LONG).show();

}

}

else {

(getBaseContext(), "Connect to the selected bluetooth device first", Toast.LENGTH\_LONG).show();

}

}

public void selectFromList(View view) {

Intent intent = new Intent(this,SelectController.class);

startActivity(intent);

}

}

**.xml**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:paddingBottom="16dp"

android:paddingLeft="16dp"

android:paddingRight="16dp"

android:paddingTop="16dp"

tools:context=".bluetooth"

android:background="@drawable/bg">

<TextView

android:id="@+id/controllerName"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentStart="true"

android:layout\_below="@+id/selectBluetoothButton"

android:layout\_marginTop="31dp"

android:text=""

android:textAppearance="@style/TextAppearance.AppCompat"

android:textColor="@color/colorPrimaryDark"

android:textSize="20sp" />

<Button

android:id="@+id/selectBluetoothButton"

style="@style/Widget.AppCompat.Button.Small"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/textView"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="45dp"

android:fontFamily="serif-monospace"

android:onClick="selectFromList"

android:text="Select Controller"

android:textAlignment="center"

android:textSize="20sp" />

<com.suke.widget.SwitchButton

android:id="@+id/device1\_toggle"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentEnd="true"

android:layout\_below="@+id/selectBluetoothButton"

android:layout\_marginEnd="48dp"

android:layout\_marginTop="91dp"

/>

<com.suke.widget.SwitchButton

android:id="@+id/device2\_toggle"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentEnd="true"

android:layout\_below="@+id/device1\_toggle"

android:layout\_marginEnd="48dp"

android:layout\_marginTop="30dp"/>

<com.suke.widget.SwitchButton

android:id="@+id/device3\_toggle"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentEnd="true"

android:layout\_below="@+id/device2\_toggle"

android:layout\_marginEnd="48dp"

android:layout\_marginTop="30dp" />

<com.suke.widget.SwitchButton

android:id="@+id/device4\_toggle"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentEnd="true"

android:layout\_below="@+id/device3\_toggle"

android:layout\_marginEnd="48dp"

android:layout\_marginTop="30dp" />

<com.suke.widget.SwitchButton

android:id="@+id/device\_all\_toggle"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentEnd="true"

android:layout\_below="@+id/device4\_toggle"

android:layout\_marginEnd="48dp"

android:layout\_marginTop="30dp" />

<TextView

android:id="@+id/textView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentStart="true"

android:layout\_alignParentTop="true"

android:fontFamily="serif-monospace"

android:text="Blutooth Controller"

android:textAlignment="center"

android:textAppearance="@style/TextAppearance.AppCompat"

android:textSize="26sp"

android:textStyle="bold" />

<TextView

android:id="@+id/device1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignTop="@+id/device1\_toggle"

android:layout\_marginEnd="63dp"

android:layout\_toStartOf="@+id/device1\_toggle"

android:text="Device 1"

android:textAlignment="center"

android:textAppearance="@style/TextAppearance.AppCompat"

android:textSize="24sp" />

<TextView

android:id="@+id/device2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignTop="@+id/device2\_toggle"

android:layout\_marginEnd="63dp"

android:layout\_toStartOf="@+id/device2\_toggle"

android:text="Device 2"

android:textAlignment="center"

android:textAppearance="@style/TextAppearance.AppCompat"

android:textSize="24sp" />

<TextView

android:id="@+id/device3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignTop="@+id/device3\_toggle"

android:layout\_marginEnd="63dp"

android:layout\_toStartOf="@+id/device3\_toggle"

android:text="Device 3"

android:textAlignment="center"

android:textAppearance="@style/TextAppearance.AppCompat"

android:textSize="24sp" />

<TextView

android:id="@+id/device4"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignTop="@+id/device4\_toggle"

android:layout\_marginEnd="63dp"

android:layout\_toStartOf="@+id/device4\_toggle"

android:text="Device 4"

android:textAlignment="center"

android:textAppearance="@style/TextAppearance.AppCompat"

android:textSize="24sp" />

<TextView

android:id="@+id/device\_all"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignStart="@+id/device4"

android:layout\_alignTop="@+id/device\_all\_toggle"

android:text="All Device"

android:textAlignment="center"

android:textAppearance="@style/TextAppearance.AppCompat"

android:textSize="24sp" />

</RelativeLayout>

**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

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

digitalWrite(relay1, HIGH); //Switch relay1 off

digitalWrite(relay2, HIGH); //Swtich relay2 off

digitalWrite(relay3, HIGH); //Switch relay1 off

digitalWrite(relay4, HIGH); //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

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, HIGH);

}

else if(inputs == "a")

{

digitalWrite(relay1, LOW);

}

else if(inputs == "B")

{

digitalWrite(relay2, HIGH);

}

else if(inputs == "b")

{

digitalWrite(relay2, LOW);

}

else if(inputs == "C")

{

digitalWrite(relay3, HIGH);

}

else if(inputs == "c")

{

digitalWrite(relay3, LOW);

}

else if(inputs == "D")

{

digitalWrite(relay4, HIGH);

}

else if(inputs == "d")

{

digitalWrite(relay4, LOW);

}

inputs="";

}

}