SECURITY AND VOICE CONTROL HOME AUTOMATION

VIT-AP
UNIVERSITY

UNDER THE GUIDANCE OF :

Asst. Prof. Sheela Jayachandran

SUBMITTED BY :

KOTRIKE RISHIKA – 20BCE7240

PRATHAPANI SATWIKA – 20BCD7160

MEKALA POORNIMAI – 20BCD7167

GONUGUNTLA ROHINI – 20BCD7174

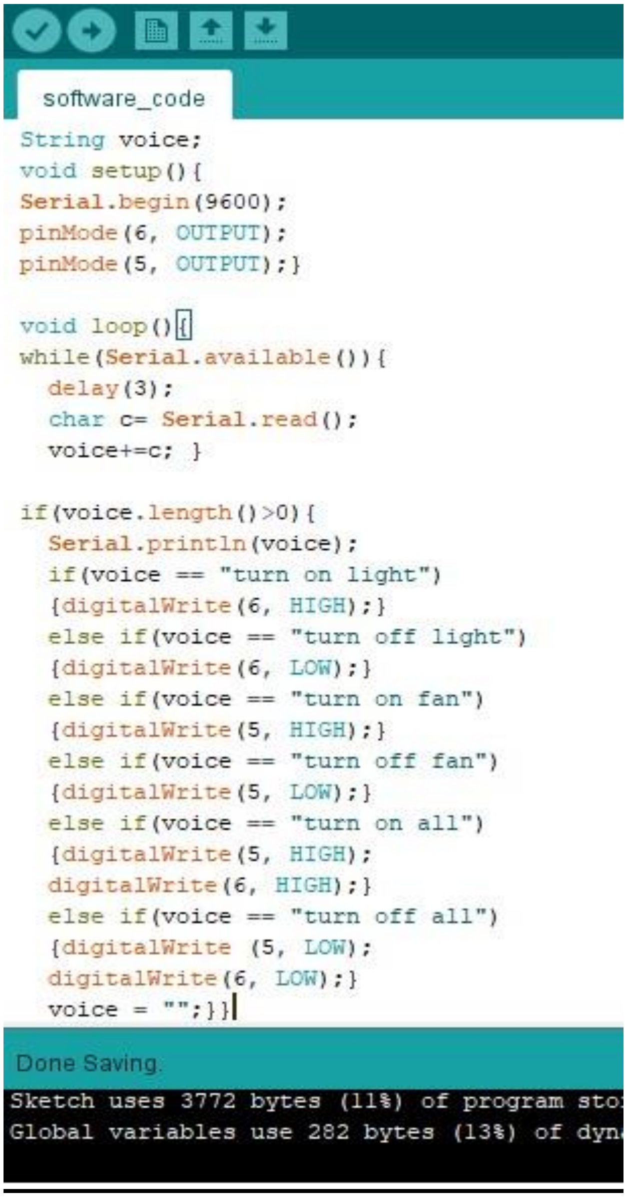
YENDLURI SRUTHI – 20BCE7396

HEMA VARSHINI DASARI – 20BCE7424

SOFTWARE MODULE :

In this project we used proteus, Arduino IDE and android application to control the components (Home appliances) present inside the circuit of proteus through Bluetooth module.

SOFTWARE CODE :

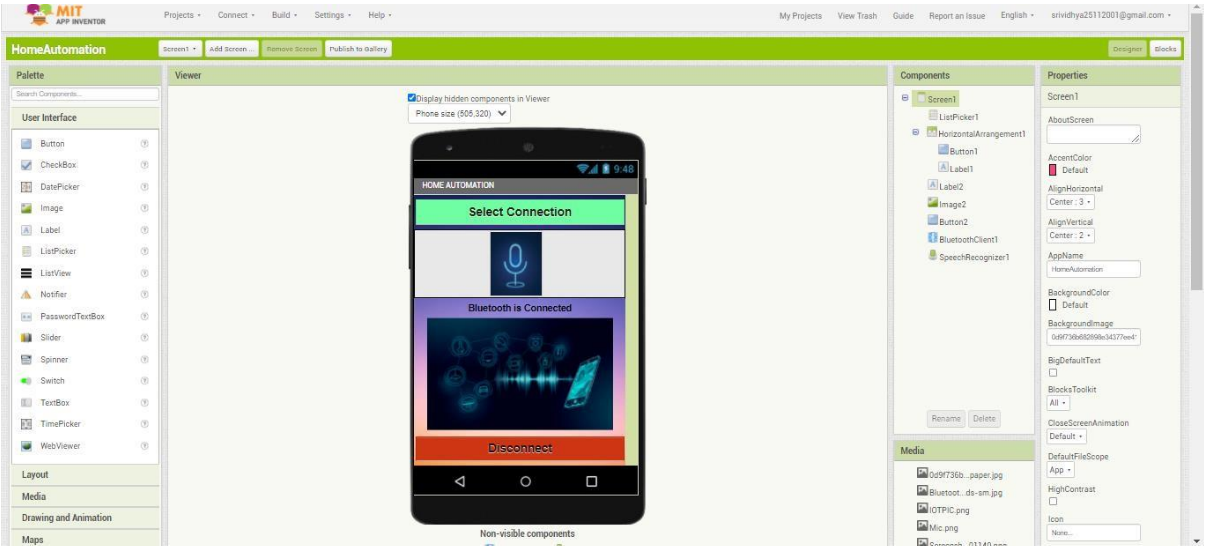


```
String voice;
void setup(){
Serial.begin(9600);
pinMode(6, OUTPUT);
pinMode(5, OUTPUT);}

void loop(){
while(Serial.available()){
  delay(3);
  char c= Serial.read();
  voice+=c; }

if(voice.length()>0){
  Serial.println(voice);
  if(voice == "turn on light")
  {digitalWrite(6, HIGH);}
  else if(voice == "turn off light")
  {digitalWrite(6, LOW);}
  else if(voice == "turn on fan")
  {digitalWrite(5, HIGH);}
  else if(voice == "turn off fan")
  {digitalWrite(5, LOW);}
  else if(voice == "turn on all")
  {digitalWrite(5, HIGH);
  digitalWrite(6, HIGH);}
  else if(voice == "turn off all")
  {digitalWrite (5, LOW);
  digitalWrite(6, LOW);}
  voice = "";}}
```

Done Saving.

```
Sketch uses 3772 bytes (11%) of program sto
Global variables use 282 bytes (13%) of dyn
```

ANDROID APP FOR SOFTWARE AND HARDWARE:



Components:

- Screen1
  - ListPicker1
  - HorizontalArrangement1
    - Button1
    - Label1
  - Label2
  - Image2
  - Button2
  - BluetoothClient1
  - SpeechRecognizer1

Media: 0d9f736b...paper.jpg, Bluetoot...ds-sm.jpg, IOTPIC.png, Mic.png, Screensh...01140.png

Properties — Screen1:

AboutScreen

AccentColor: Default

AlignHorizontal: Center : 3

AlignVertical: Center : 2

AppName: HomeAutomation

BackgroundColor: Default

BackgroundImage: 0d9f736b682898e34377ee4

BigDefaultText: □

BlocksToolkit: All

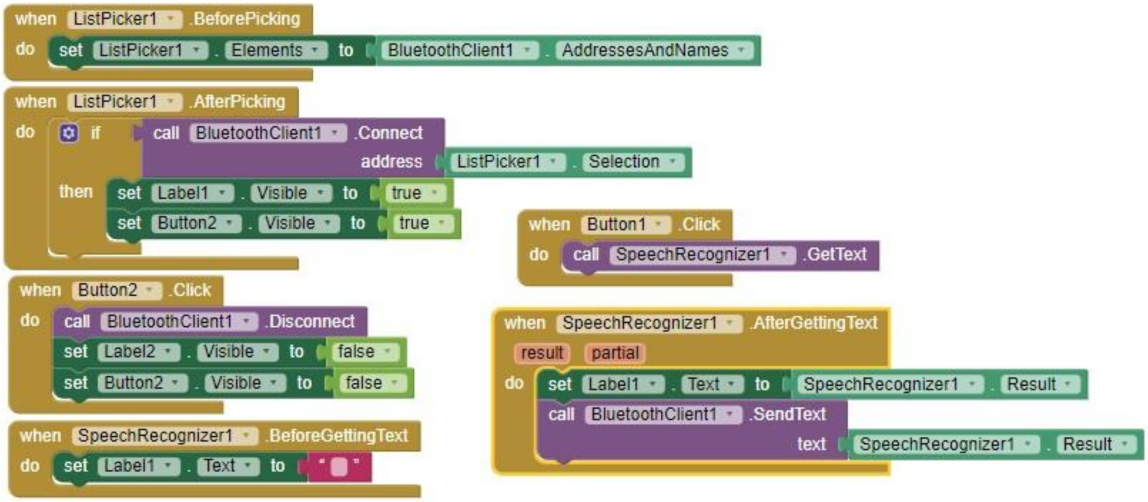
CloseScreenAnimation: Default

DefaultFileScope: App

HighContrast: □

Icon: None...

CODE BLOCKS FOR THE APP:

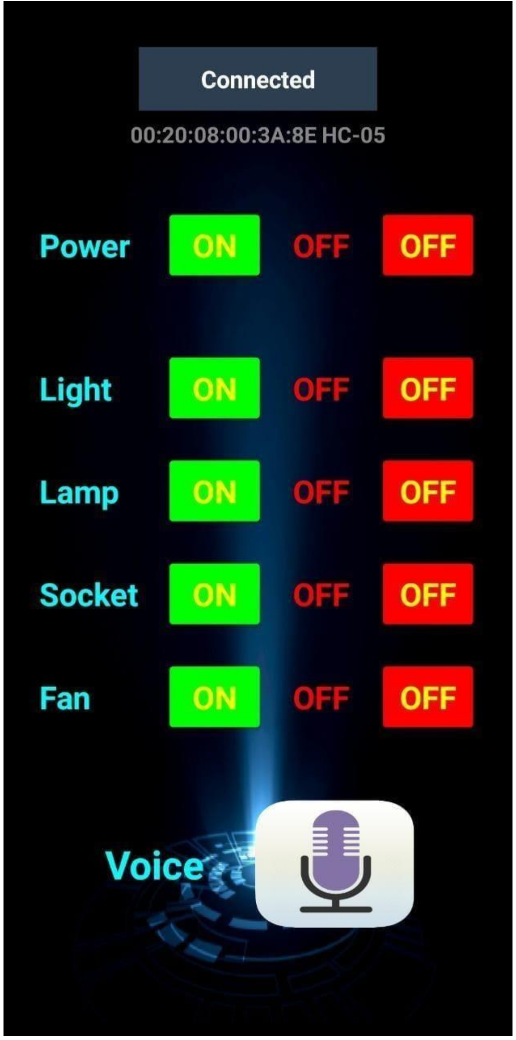


```
when ListPicker1 .BeforePicking
do  set ListPicker1 . Elements to  BluetoothClient1 . AddressesAndNames

when ListPicker1 .AfterPicking
do  if    call BluetoothClient1 .Connect
                              address  ListPicker1 . Selection
    then  set Label1 . Visible to  true
          set Button2 . Visible to  true

when Button1 .Click
do  call SpeechRecognizer1 .GetText

when Button2 .Click
do  call BluetoothClient1 .Disconnect
    set Label2 . Visible to  false
    set Button2 . Visible to  false

when SpeechRecognizer1 .AfterGettingText
 result  partial
do  set Label1 . Text to  SpeechRecognizer1 . Result
    call BluetoothClient1 .SendText
                     text  SpeechRecognizer1 . Result

when SpeechRecognizer1 .BeforeGettingText
do  set Label1 . Text to  " "
```

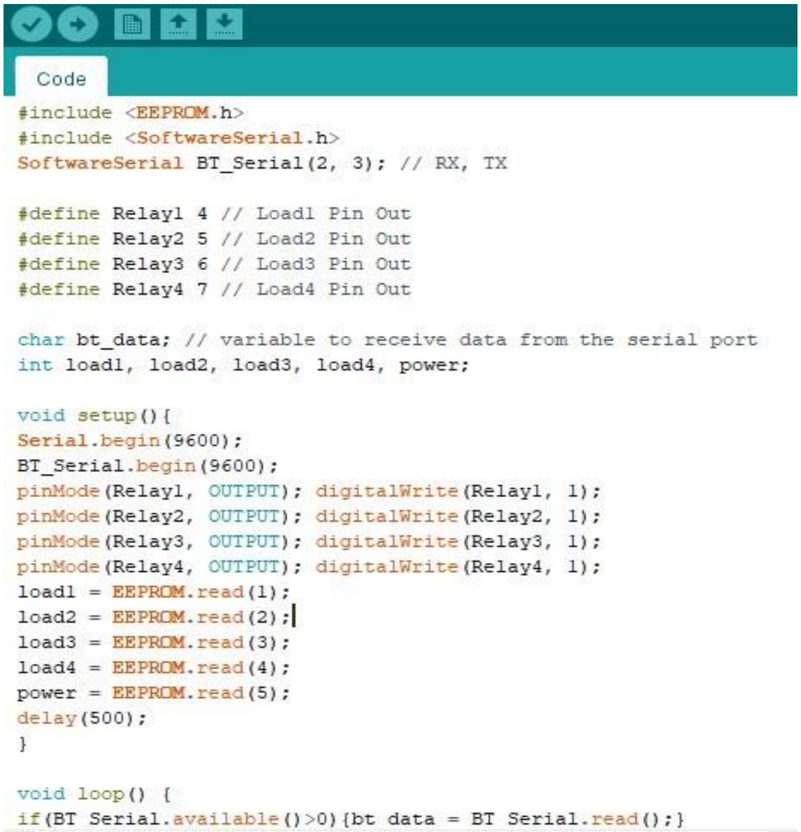
ANDROID APPLICATION FOR HARDWARE:

- The is created on MIT APP INVENTOR an online platform to develop android apps.
- For the hardware module according to the requirements the app is designed.

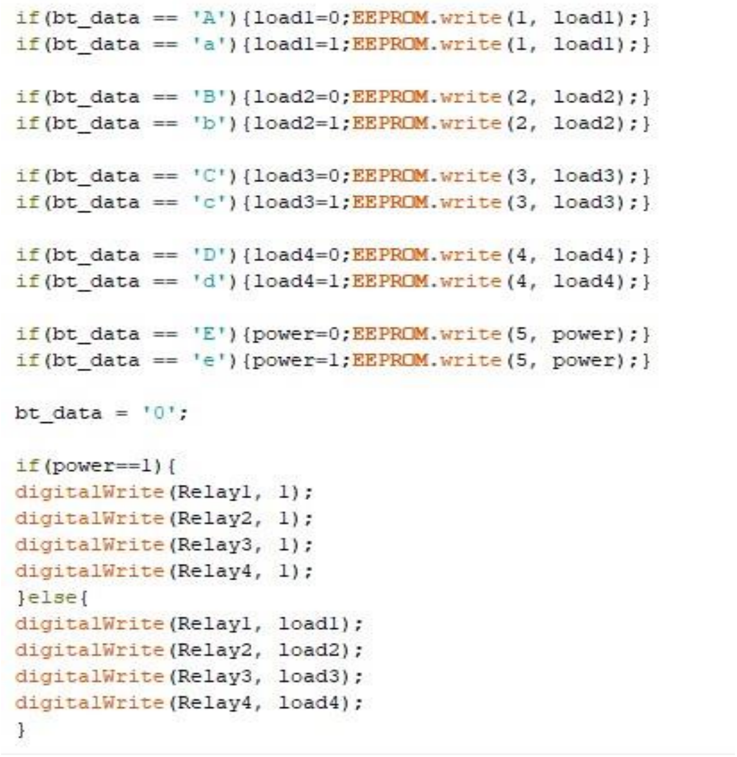
  • We have to connect the App to the HC – 05 Bluetooth Module that acts as the communication protocol.
- We have added two features to operate the appliances either manually through the buttons that are enabled or through the voice command.



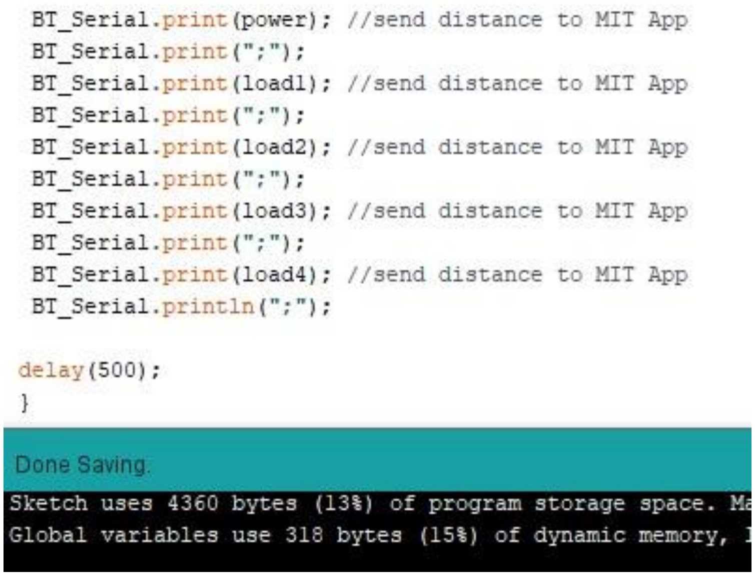
CODE:



```
#include <EEPROM.h>
#include <SoftwareSerial.h>
SoftwareSerial BT_Serial(2, 3); // RX, TX

#define Relay1 4 // Load1 Pin Out
#define Relay2 5 // Load2 Pin Out
#define Relay3 6 // Load3 Pin Out
#define Relay4 7 // Load4 Pin Out

char bt_data; // variable to receive data from the serial port
int load1, load2, load3, load4, power;

void setup(){
Serial.begin(9600);
BT_Serial.begin(9600);
pinMode(Relay1, OUTPUT); digitalWrite(Relay1, 1);
pinMode(Relay2, OUTPUT); digitalWrite(Relay2, 1);
pinMode(Relay3, OUTPUT); digitalWrite(Relay3, 1);
pinMode(Relay4, OUTPUT); digitalWrite(Relay4, 1);
load1 = EEPROM.read(1);
load2 = EEPROM.read(2);
load3 = EEPROM.read(3);
load4 = EEPROM.read(4);
power = EEPROM.read(5);
delay(500);
}

void loop() {
if(BT Serial.available()>0){bt data = BT Serial.read();}
```



```
  if(bt_data == 'A'){load1=0;EEPROM.write(1, load1);}
  if(bt_data == 'a'){load1=1;EEPROM.write(1, load1);}

  if(bt_data == 'B'){load2=0;EEPROM.write(2, load2);}
  if(bt_data == 'b'){load2=1;EEPROM.write(2, load2);}

  if(bt_data == 'C'){load3=0;EEPROM.write(3, load3);}
  if(bt_data == 'c'){load3=1;EEPROM.write(3, load3);}

  if(bt_data == 'D'){load4=0;EEPROM.write(4, load4);}
  if(bt_data == 'd'){load4=1;EEPROM.write(4, load4);}

  if(bt_data == 'E'){power=0;EEPROM.write(5, power);}
  if(bt_data == 'e'){power=1;EEPROM.write(5, power);}

  bt_data = '0';

  if(power==1){
  digitalWrite(Relay1, 1);
  digitalWrite(Relay2, 1);
  digitalWrite(Relay3, 1);
  digitalWrite(Relay4, 1);
  }else{
  digitalWrite(Relay1, load1);
  digitalWrite(Relay2, load2);
  digitalWrite(Relay3, load3);
  digitalWrite(Relay4, load4);
  }
```

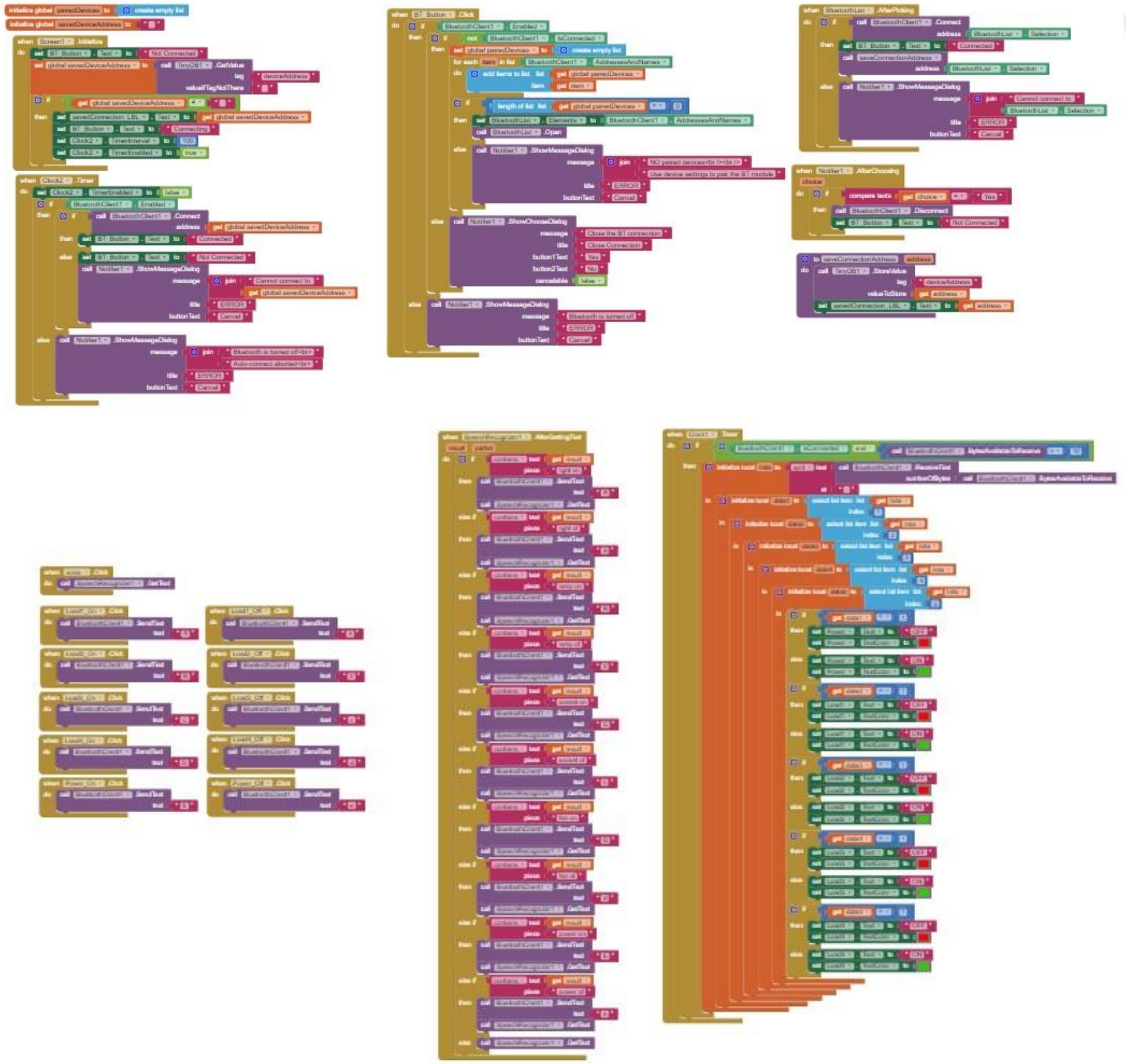


```
 BT_Serial.print(power); //send distance to MIT App
 BT_Serial.print(";");
 BT_Serial.print(load1); //send distance to MIT App
 BT_Serial.print(";");
 BT_Serial.print(load2); //send distance to MIT App
 BT_Serial.print(";");
 BT_Serial.print(load3); //send distance to MIT App
 BT_Serial.print(";");
 BT_Serial.print(load4); //send distance to MIT App
 BT_Serial.println(";");

delay(500);
}
```

Done Saving.

```
Sketch uses 4360 bytes (13%) of program storage space. Ma
Global variables use 318 bytes (15%) of dynamic memory, I
```

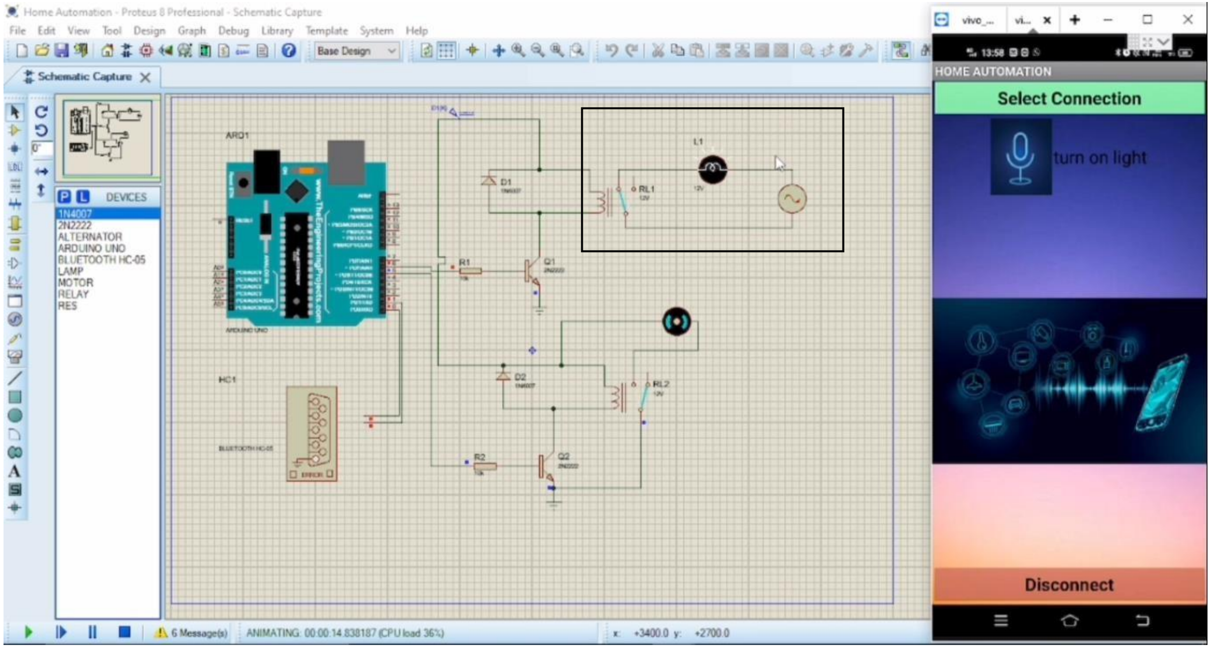
BLOCKS FOR ANDROID APP:



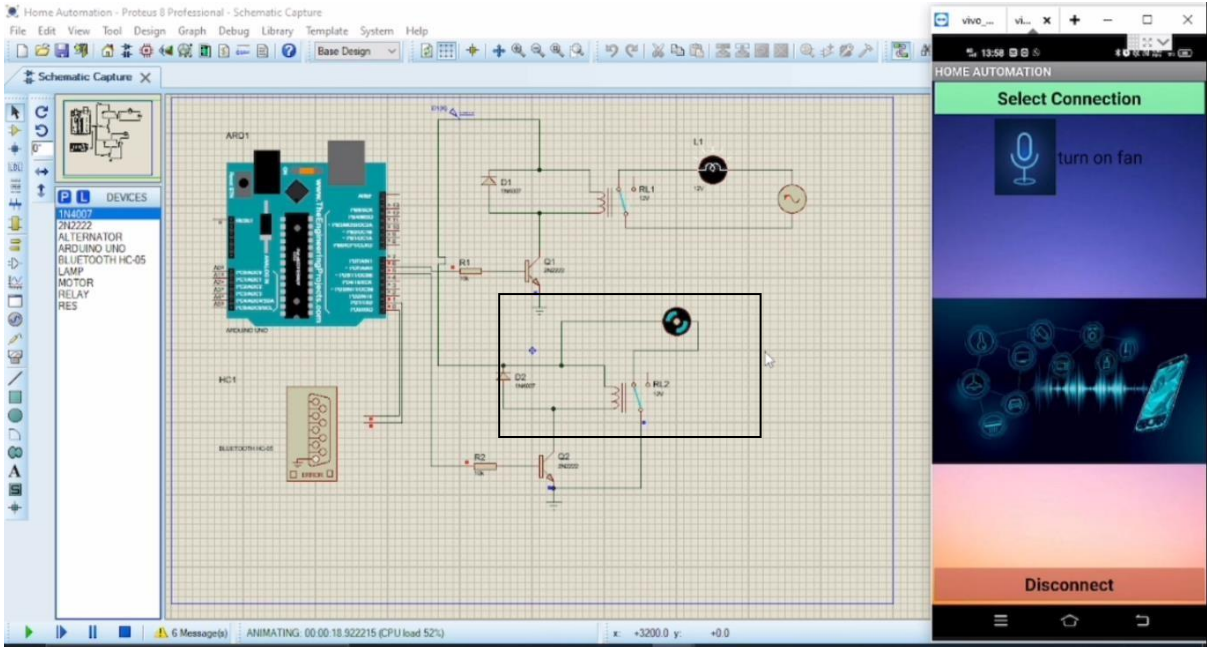
```
initialize global pairedDevices to create empty list
initialize global savedDeviceAddress to " "

when Screen1 .Initialize
do  set BT_Button . Text to "Not Connected"
    set global savedDeviceAddress to call TinyDB1 .GetValue
                                              tag "deviceAddress"
                                              valueIfTagNotThere " "
    if  get global savedDeviceAddress ≠ " "
    then set savedConnection_LBL . Text to get global savedDeviceAddress
         set BT_Button . Text to "Connecting"
         set Clock2 . TimerInterval to 100
         set Clock2 . TimerEnabled to true

when Clock2 .Timer
do  set Clock2 . TimerEnabled to false
    if  BluetoothClient1 . Enabled
    then if  call BluetoothClient1 .Connect
                                   address get global savedDeviceAddress
         then set BT_Button . Text to "Connected"
         else set BT_Button . Text to "Not Connected"
              call Notifier1 .ShowMessageDialog
                             message join "Cannot connect to "
                                          get global savedDeviceAddress
                             title "ERROR"
                             buttonText "Cancel"
    else call Notifier1 .ShowMessageDialog
                        message join "Bluetooth is turned off<br>"
                                     "Auto-connect aborted<br>"
                        title "ERROR"
                        buttonText "Cancel"

when BT_Button .Click
do  if  BluetoothClient1 . Enabled
    then if  not BluetoothClient1 . IsConnected
         then set global pairedDevices to create empty list
              for each item in list BluetoothClient1 . AddressesAndNames
              do  add items to list list get global pairedDevices
                                    item get item
              if  length of list list get global pairedDevices ≠ 0
              then set BluetoothList . Elements to BluetoothClient1 . AddressesAndNames
                   call BluetoothList .Open
              else call Notifier1 .ShowMessageDialog
                                  message join "NO paired devices<br /><br />"
                                               "Use device settings to pair the BT module"
                                  title "ERROR"
                                  buttonText "Cancel"
         else call Notifier1 .ShowChooseDialog
                             message "Close the BT connection"
                             title "Close Connection"
                             button1Text "Yes"
                             button2Text "No"
                             cancelable false
    else call Notifier1 .ShowMessageDialog
                        message "Bluetooth is turned off"
                        title "ERROR"
                        buttonText "Cancel"

when BluetoothList .AfterPicking
do  if  call BluetoothClient1 .Connect
                              address BluetoothList . Selection
    then set BT_Button . Text to "Connected"
         call saveConnectionAddress
                              address BluetoothList . Selection
    else call Notifier1 .ShowMessageDialog
                        message join "Cannot connect to "
                                     BluetoothList . Selection
                        title "ERROR"
                        buttonText "Cancel"

when Notifier1 .AfterChoosing
  choice
do  if  compare texts get choice = "Yes"
    then call BluetoothClient1 .Disconnect
         set BT_Button . Text to "Not Connected"

to saveConnectionAddress address
do  call TinyDB1 .StoreValue
                 tag "deviceAddress"
                 valueToStore get address
    set savedConnection_LBL . Text to get address
```

DEMONSTRATION:

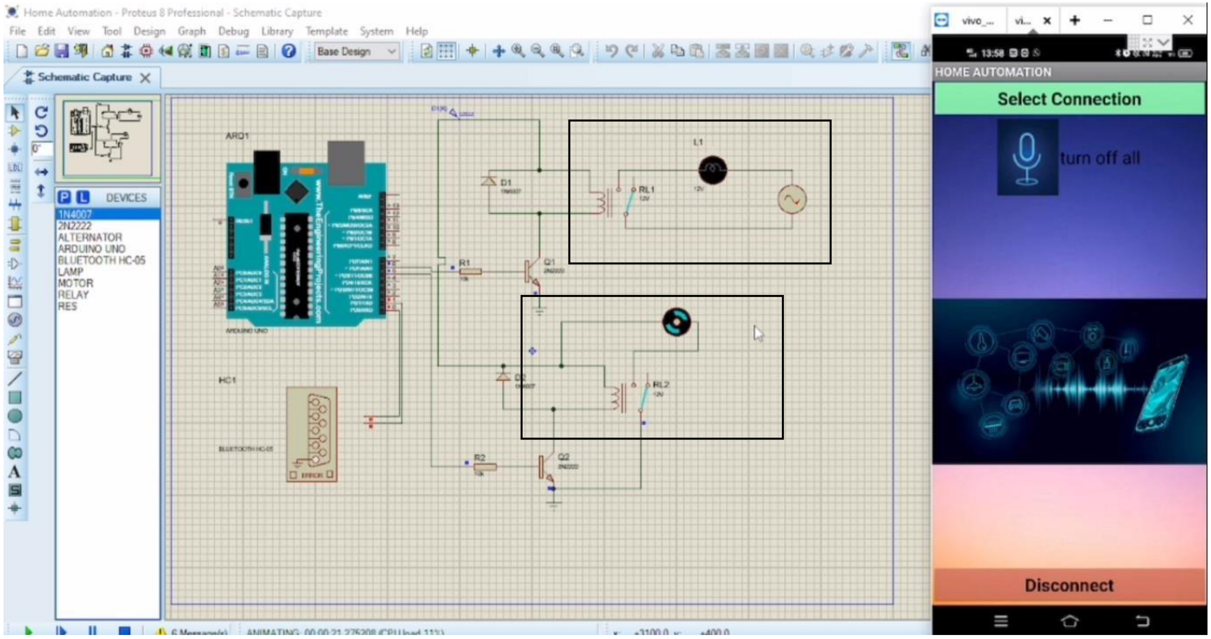
- **TURN ON LIGHT**(The relay connected to the light gets closed and the light bulb glows)



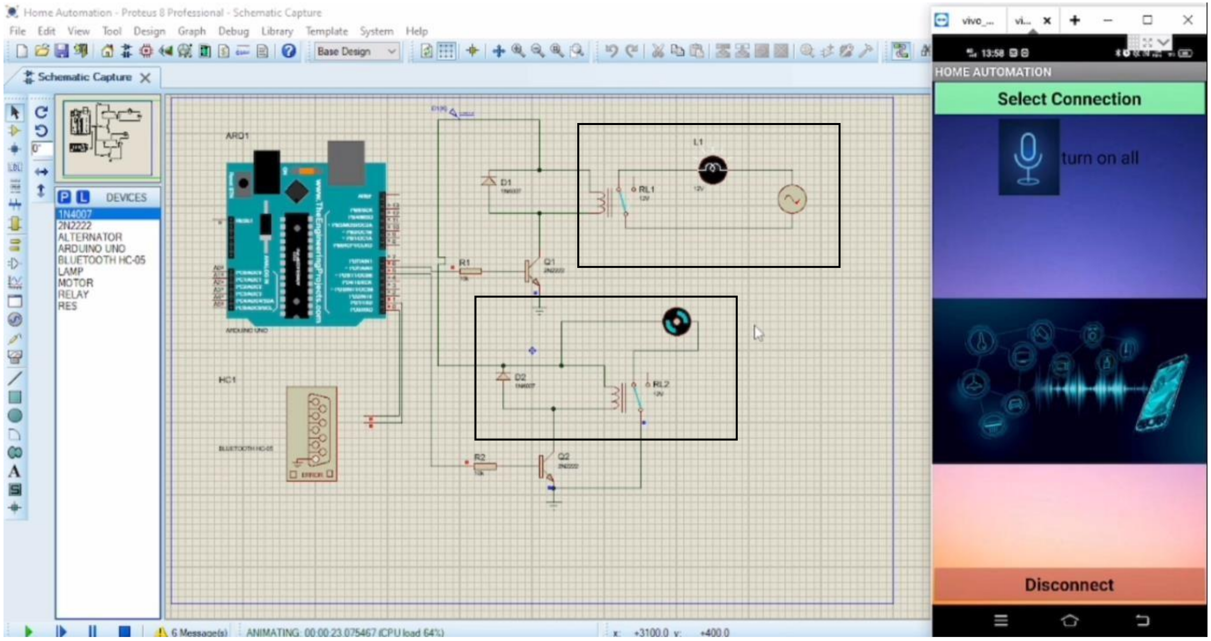
- **TURN ON FAN** (The relay connected to the fan gets closed and the fan rotates)



- **TURN OFF ALL**(The relay connected to the light and fan gets opened so the light bulb stops to glow and the fan stops to rotate)



- **TURN ON ALL**(The relay connected to the light and fan gets closed so the light bulb starts to glow and the fan starts to rotate)


Home Automation - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Base Design
Schematic Capture
P L DEVICES
1N4007
2N2222
ALTERNATOR
ARDUINO UNO
BLUETOOTH HC-05
LAMP
MOTOR
RELAY
RES
ARD1
ARDUINO UNO
www.TheEngineeringProjects.com
HC1
BLUETOOTH HC-05
D1
D2
R1
R2
Q1
Q2
RL1
RL2
L1
6 Message(s)
ANIMATING: 00:00:23.075467 (CPU load 64%)
x: +3100.0 y: +400.0
vivo_...
13:58
HOME AUTOMATION
Select Connection
turn on all
Disconnect