



Controlling Home Appliances with an Android Application

Group Members

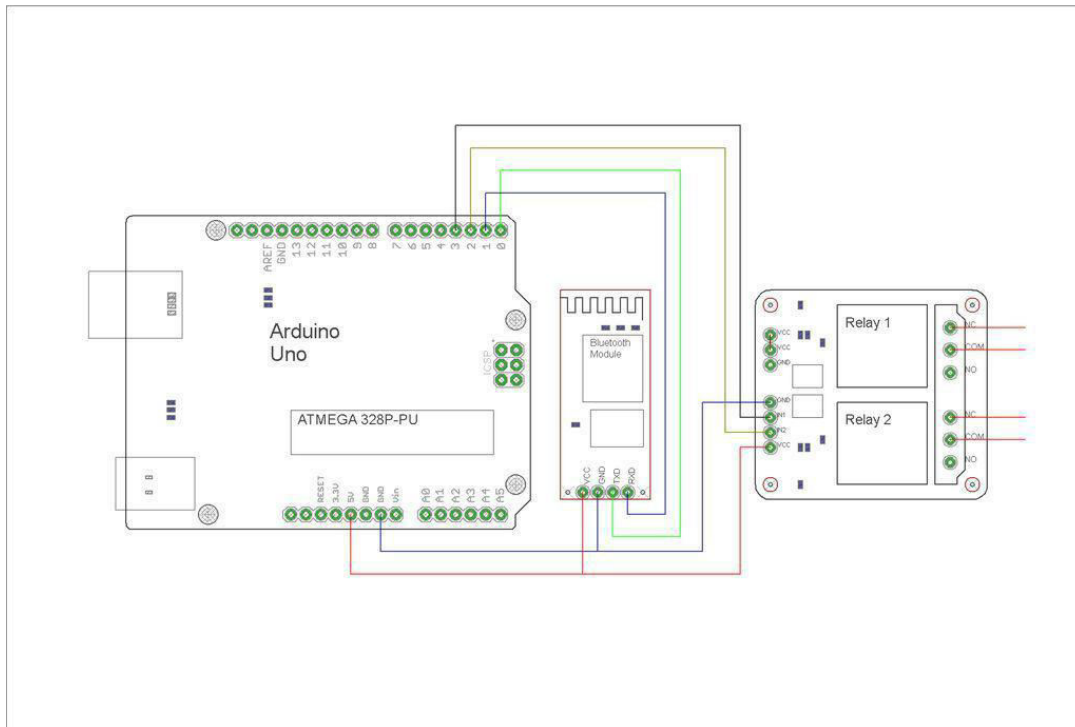
MD Ayat Ullah
AM Saadman Rafat



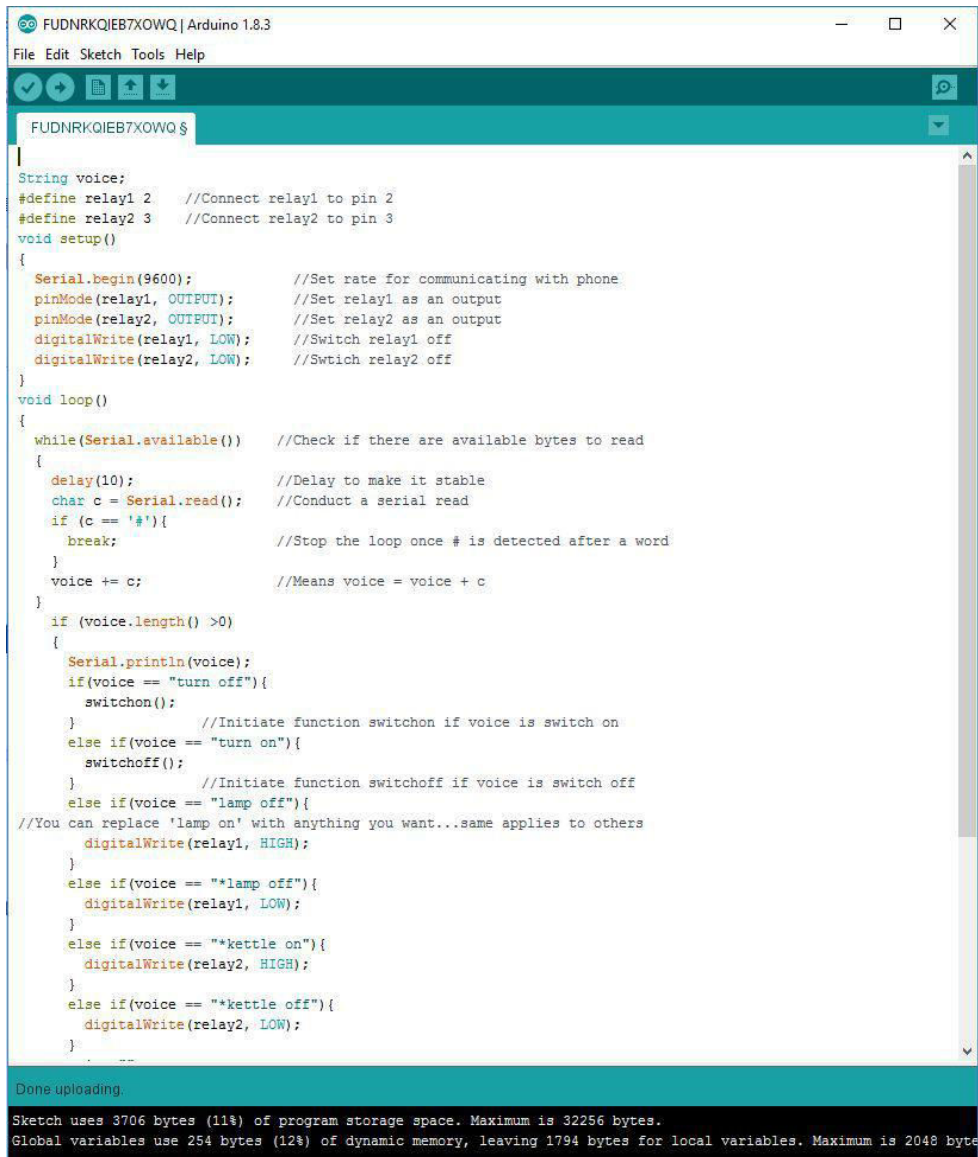
Introduction

We made a stand-alone device. It is connected to the mains and controls the power outlet of the electrical device that is plugged into it. We worked on to give the users an Android App, which they can use to control their home appliances. This project uses the capability of an Arduino, Bluetooth module, several relays and of course an Android phone to make it happen. The android application uses voice input to control home appliances.

Circuit Diagram



Arduino Code



```
FUDNRKQIEB7XOWQ | Arduino 1.8.3
File Edit Sketch Tools Help

FUDNRKQIEB7XOWQ $

String voice;
#define relay1 2 //Connect relay1 to pin 2
#define relay2 3 //Connect relay2 to pin 3
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
  digitalWrite(relay1, LOW); //Switch relay1 off
  digitalWrite(relay2, LOW); //Switch 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
    }
    voice += c; //Means voice = voice + c
  }
  if (voice.length() > 0)
  {
    Serial.println(voice);
    if(voice == "turn off"){
      switchon();
    } //Initiate function switchon if voice is switch on
    else if(voice == "turn on"){
      switchoff();
    } //Initiate function switchoff if voice is switch off
    else if(voice == "lamp off"){
      //You can replace 'lamp on' with anything you want...same applies to others
      digitalWrite(relay1, HIGH);
    }
    else if(voice == "*lamp off"){
      digitalWrite(relay1, LOW);
    }
    else if(voice == "*kettle on"){
      digitalWrite(relay2, HIGH);
    }
    else if(voice == "*kettle off"){
      digitalWrite(relay2, LOW);
    }
  }
}

Done uploading.

Sketch uses 3706 bytes (11% of program storage space. Maximum is 32256 bytes.
Global variables use 254 bytes (12% of dynamic memory, leaving 1794 bytes for local variables. Maximum is 2048 byte
```

```
        digitalWrite(relay1, LOW);
    }
    else if(voice == "*kettle on"){
        digitalWrite(relay2, HIGH);
    }
    else if(voice == "*kettle off"){
        digitalWrite(relay2, LOW);
    }
    voice="";
}

void switchon()           //Function for turning on relays
{
    digitalWrite(relay1, HIGH);
    digitalWrite(relay2, HIGH);
}

void switchoff()          //Function for turning on relays
{
    digitalWrite(relay1, LOW);
    digitalWrite(relay2, LOW);
}
```

Done uploading.

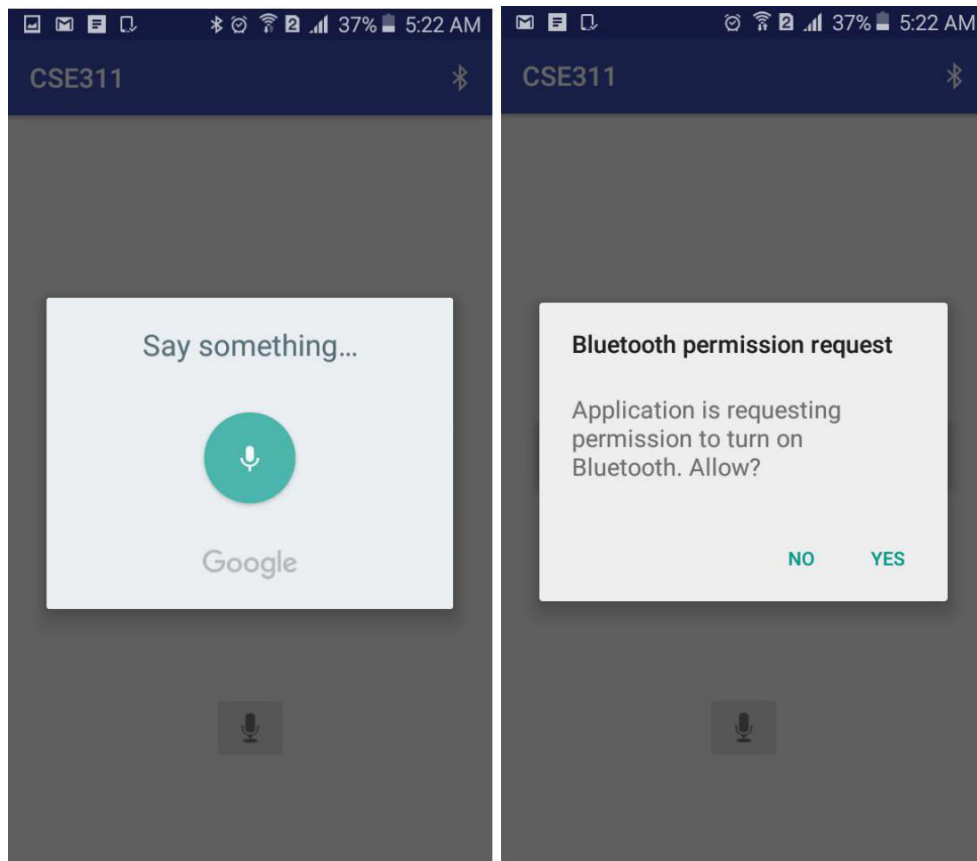
Sketch uses 3706 bytes (11%) of program storage space. Maximum is 32256 bytes.

Global variables use 254 bytes (12%) of dynamic memory, leaving 1794 bytes for local variables. Maximum is 2048 bytes.

< 1

Arduino/Genuine Uno on COM7

Android Application



Android Application Code

In this code we are trying to connect to the HC-05 Bluetooth Module.

```
private class BluetoothConnectThread extends Thread {
    private BluetoothSocket bluetoothSocket = null;
    private final BluetoothDevice bluetoothDevice;
    public BluetoothConnectThread(BluetoothDevice bluetoothDevice) {
        this.bluetoothDevice = bluetoothDevice;
        try {
            bluetoothSocket = bluetoothDevice.createRfcommSocketToServiceRecord
        } catch (IOException e) {
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    Toast.makeText(getApplicationContext(), "Device Not Found.",
                }
            });
        }
    }
    public void run() {
        try {
            bluetoothSocket.connect();
            runOnUiThread(() -> {
                Toast.makeText(getApplicationContext(), "Connected To Device",
            });
        } catch (IOException e) {
            runOnUiThread(() -> {
                Toast.makeText(getApplicationContext(), "Device Not Found.",
            });
        }
    }
    communicationThread = new CommunicationThread(bluetoothSocket);
}
```

Here we are trying to send the voice command to the HC-05 module.

```
private class CommunicationThread{
    BluetoothSocket connectedBluetoothSocket;
    InputStream connectedInputStream;
    OutputStream mmOutputStream;
    PrintWriter out;
    CommunicationThread(BluetoothSocket bluetoothSocket) {
        connectedBluetoothSocket = bluetoothSocket;

        try {
            mmOutputStream = connectedBluetoothSocket.getOutputStream();
        } catch (IOException e) {
            Toast.makeText(getApplicationContext(), "mmOutputStream Failed", Toast.LENGTH_SHORT).show();
        }
    }

    public void write(byte[] bytes) {
        try {
            mmOutputStream.write(bytes);
        } catch (IOException e) {
            Toast.makeText(getApplicationContext(), "Communication Failed", Toast.LENGTH_SHORT).show();
        }
    }
}
```