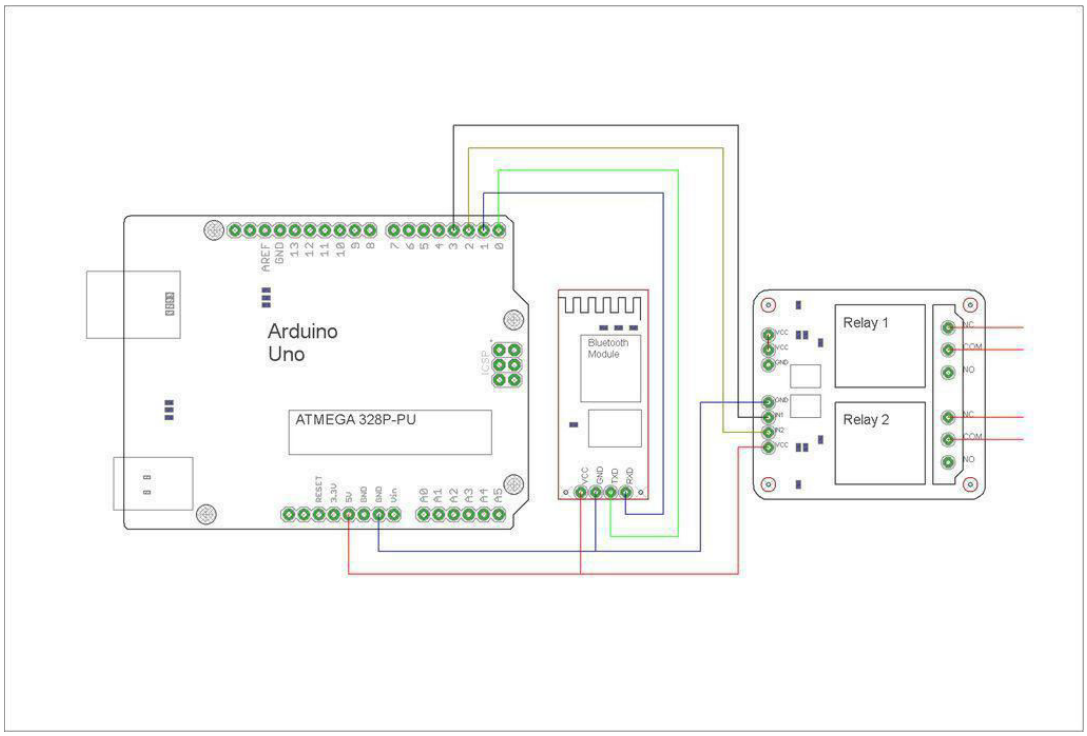
Controlling Home Appliances with an Android Application

Group Members
MD Ayat Ullah
AM Saadman Rafat

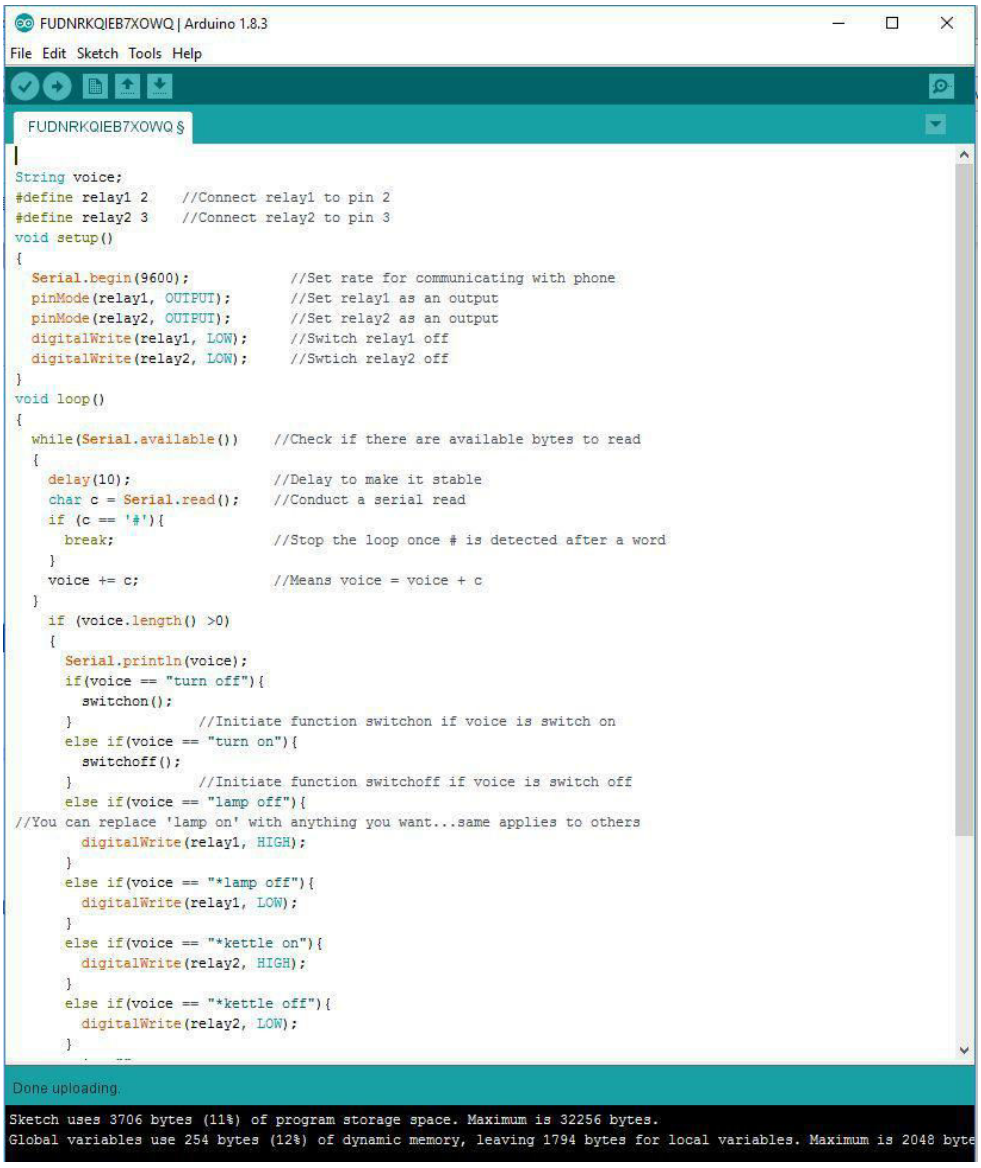
Introduction

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 and Android phone to make it happen. The android application uses voice input to control home appliances.

Circuit Diagram



Arduino Code



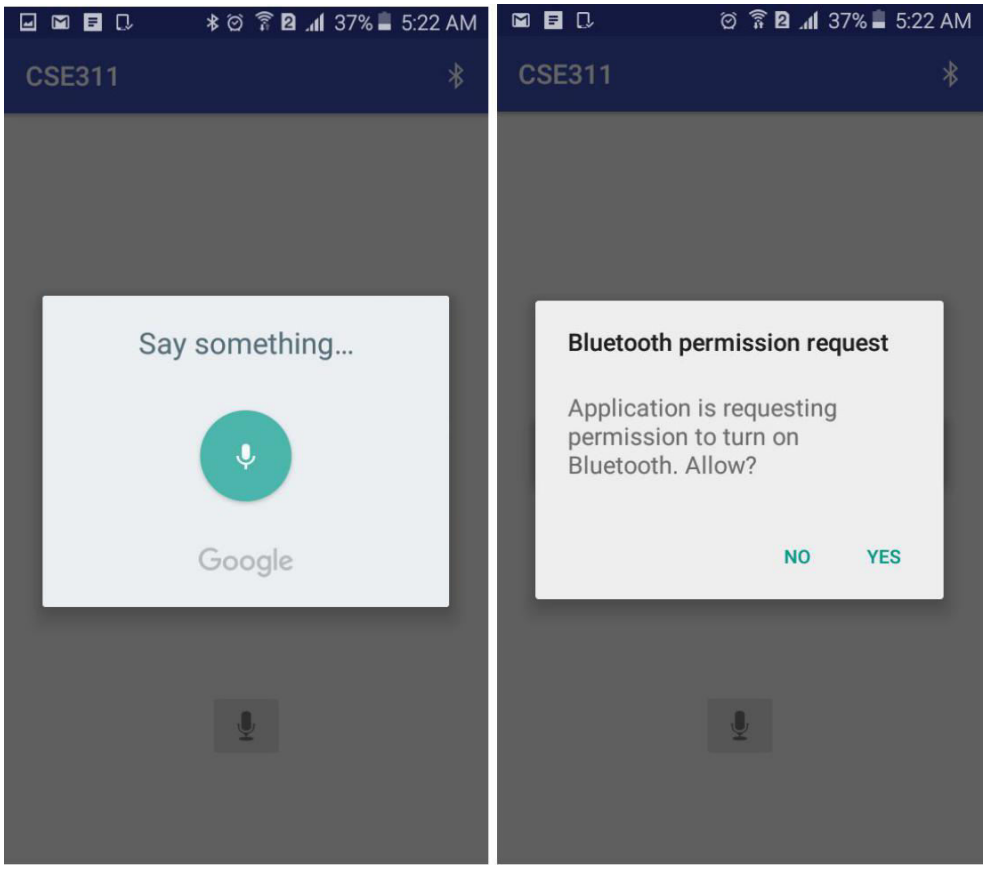
```
      digitalWrite(relay1, LOW);
    }
    else if(voice == "*kettle on"){
      digitalWrite(relay2, HIGH);
    }
    else if(voice == "*kettle off"){
      digitalWrite(relay2, LOW);
    }
    voice="";
   }
}
void switchon()               //Function for turning on relays
{
  digitalWrite(relay1, 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 byte
```

1 Arduino/Genuino 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.createRfcommSocketToServiceRecor

        }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 Devi

            });

        } 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 mmOutStream;
    PrintWriter out;
    CommunicationThread(BluetoothSocket bluetoothSocket) {
        connectedBluetoothSocket = bluetoothSocket;

        try {
            mmOutStream = connectedBluetoothSocket.getOutputStream();

        } catch (IOException e) {
            Toast.makeText(getApplicationContext(), "mmOutStream Failed", Toast.LENG
        }

    }

    public void write(byte[] bytes) {
        try {

            mmOutStream.write(bytes);
        }
         catch (IOException e) {
            Toast.makeText(getApplicationContext(), "Communication Failed", Toast.LE
        }
    }

}
```