Project Report

CSE 3200:
Software Development Project - II

The Smart Room

Developed by:-
Md.Asaf-Uddowla Golap
Roll No:1207005
Shaikh Akib Shahriyar
Roll No:1207011

Under Supervision of:
Prof. Dr. Md. Aminul Haque Akhand
Department of Computer Science & Engineering
Khulna University of Engineering and Technology

Signature

Acknowledgment

With the blessings and limitless mercy of Almighty, we are able to do this. We express our heartiest gratitude to Almighty Allah for this.

A word of special thanks must go to our highly esteemed teacher and this project's supervisor, **Prof. Dr. Md. Aminul Haque Akhand**, Department of Computer Science & Engineering, KUET, for his excellent advices and right directions without which our project may not have reached a state it is in now.

We would like to thank specially **Sheikh Imran Hossain,** Lecturer, Department of Computer Science & Engineering, KUET, who inspired us to implement different Ideas throughout this project. We would also like to thank our friends for their association also.

Any constructive comments, suggestions, criticism from teachers as well as seniors will be highly appreciated and gratefully acknowledged.

Abstract

The sole objective of this project is to provide improved convenience, comfort, energy efficiency and security. Those criteria are fulfilled by the help of microcontroller(Arduino), Wi-fi modules and Smartphone/PC. It's a complete embedded room automation system.

This project is focused to control the room appliances with a single touch. It also emphasizes on reducing power wastage. It is a reliable and powerful system with full time surveillance.

From home and away, "Smart Room" makes control simple, with everything you need in a single, elegant app. The app lets you switch easily between Lights, Fan, Doorlock, full time surveillance and more from the comfort of your favourite seat.

Our target is to offer the ultimate experience in whole room automation, ideal for discerning homeowners and elderly persons.

INDEX

1. Introduction:

You can have the perfect morning every morning when you wake up in a smartly controlled room. One touch of the "light" button and the lights will gently illuminate your side of the bed, without blinding you or disrupting your family members. It’s possible, all without fumbling for switches or a remote. And that's just the beginning in a room where all the technology is seamlessly integrated.

You’re on your way to work or the airport, and you can’t remember whether you turned off the lights and turned on the security system or locked the door. Did someone intrude in your home? Do you go back? Or Continue on your way and hope for the best? Or do you simply pull out your smart phone, open an app, and make sure everything is OK? Convenience, control, and peace of mind are the powerful combination where our project ***Smart Room*** comes in!

This project is a demonstration of how to design and build a multi-purpose automated system that can switch ON and OFF any electrical household appliance (including the security door, light, fan etc.), by sending a signal to a microcontroller (Arduino). When It controls the appliance, the phone will receive a feedback message from the appliance, whether it’s switched ON or OFF. Modern microcontrollers are very powerful for building smart electronic devices that can remotely control electrical appliances through Wi-Fi connectivity.

2. Objective:

Our target is to help users to control the electrical appliances of their rooms in a very optimized and efficient manner, where wireless connectivity is the major advantage.

Providing security to the user is another top priority. This automated system will provide facilities not usually provided by the traditional switchboards or wired control system. This project will try to improve user convenience, comfort, energy efficiency and security.

System Overview

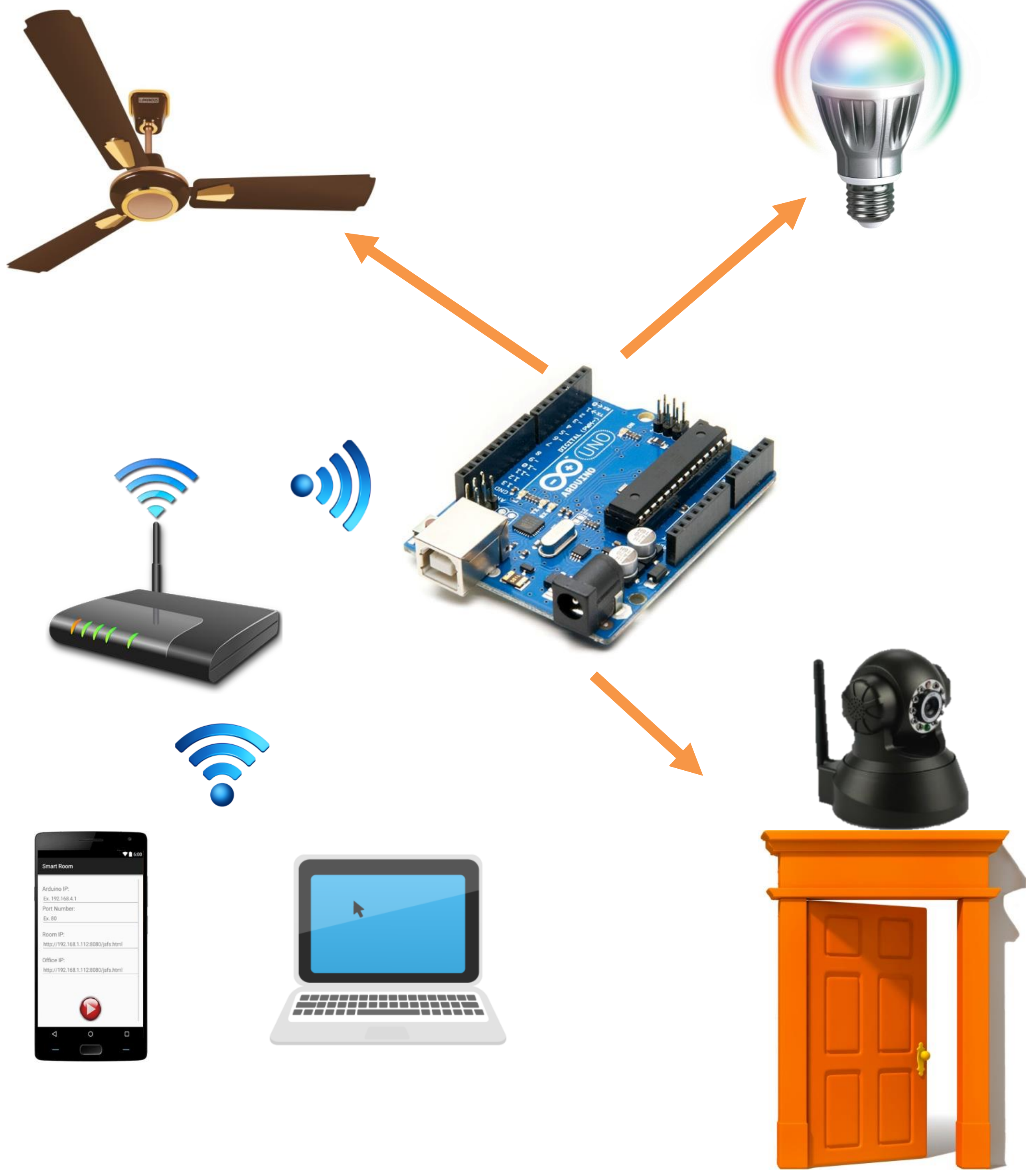


Fig 1: Functional Block Diagram for SHS.

4. Project Structure:

The project is divided into two sections, hardware & Software. They are shown below:

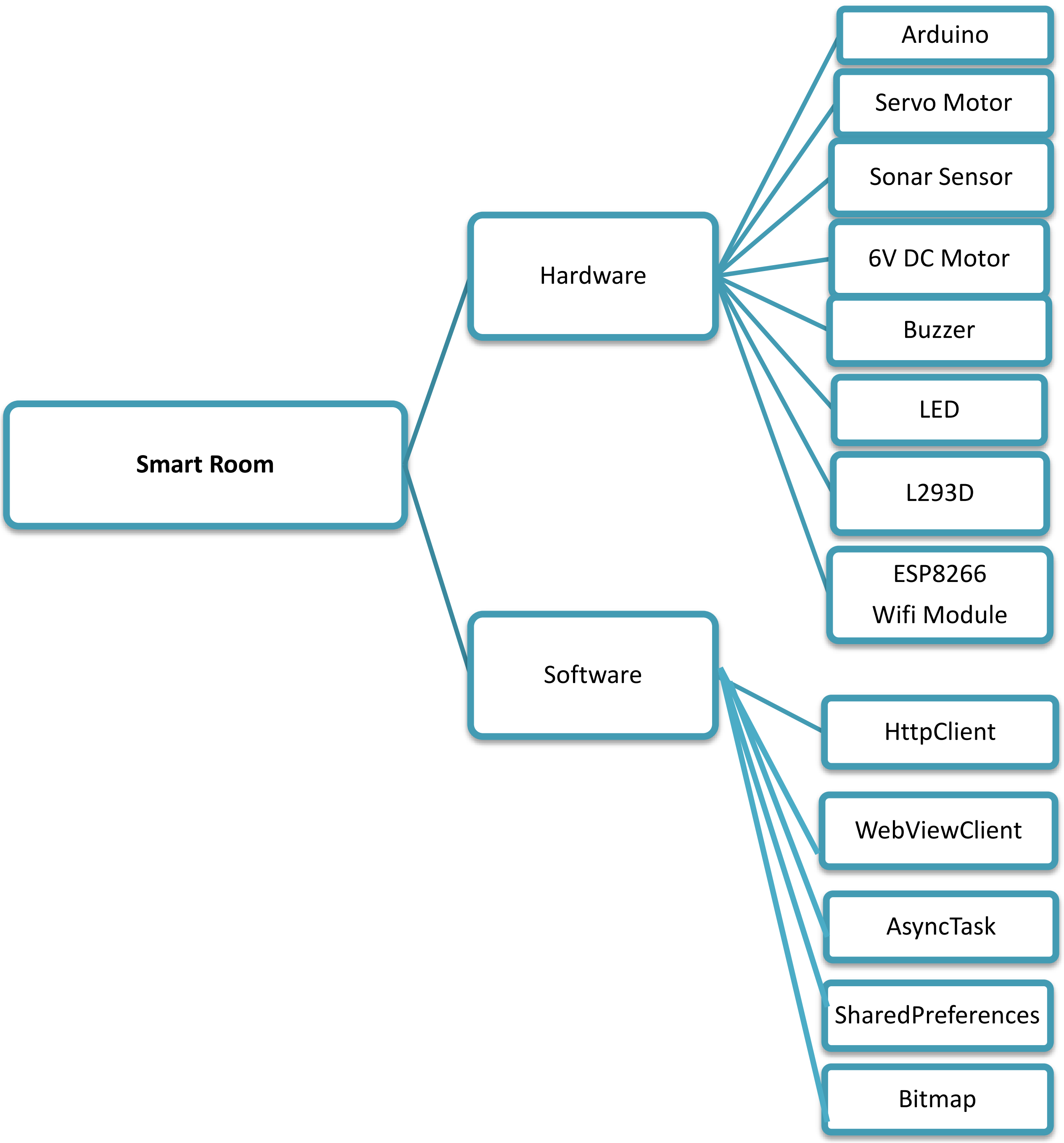


Fig 2: Project Structure

5. Control Flow of the Project: The Smart Room

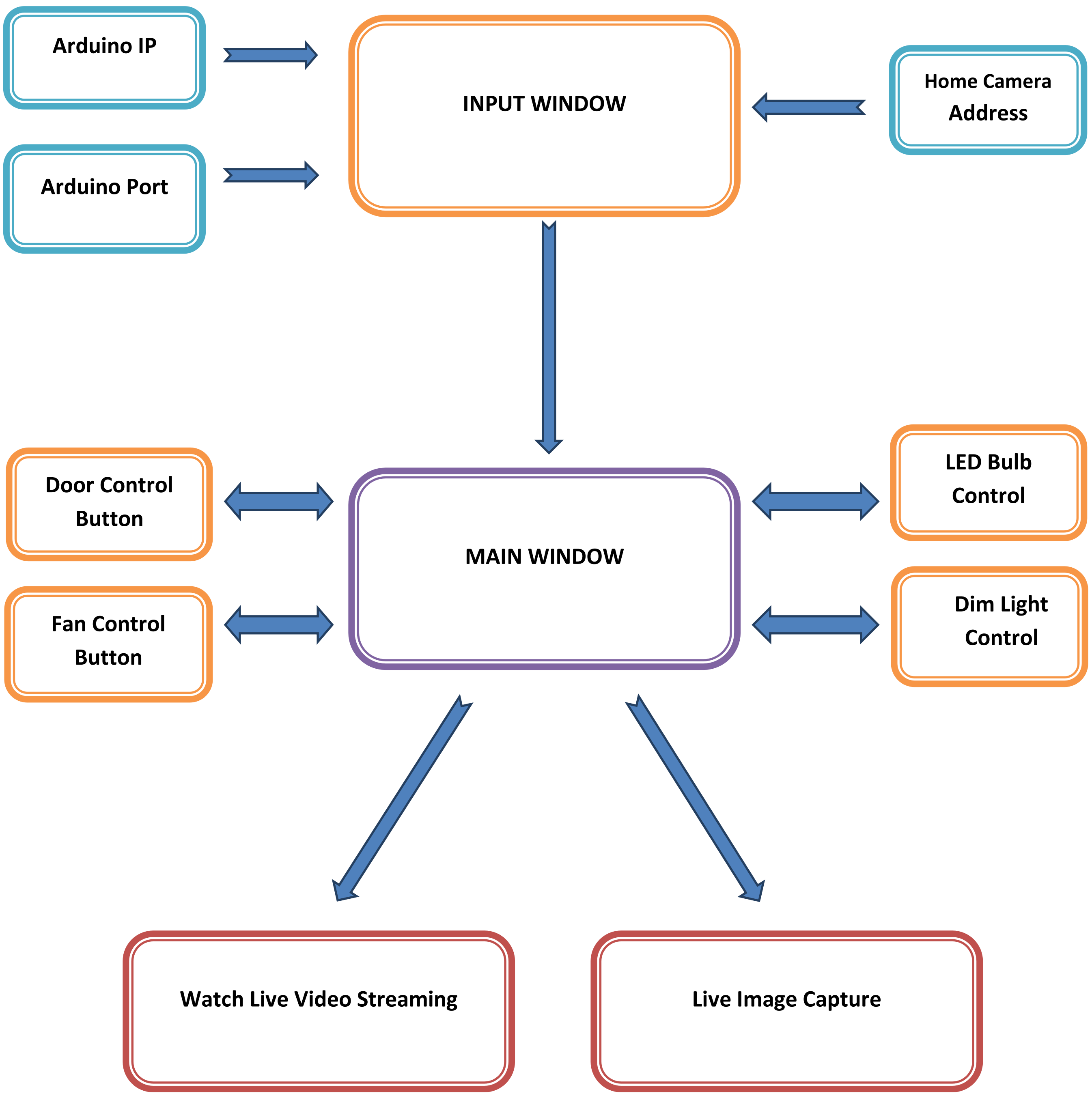


Fig 3: Control Flow Diagram

6. Circuit Diagram:

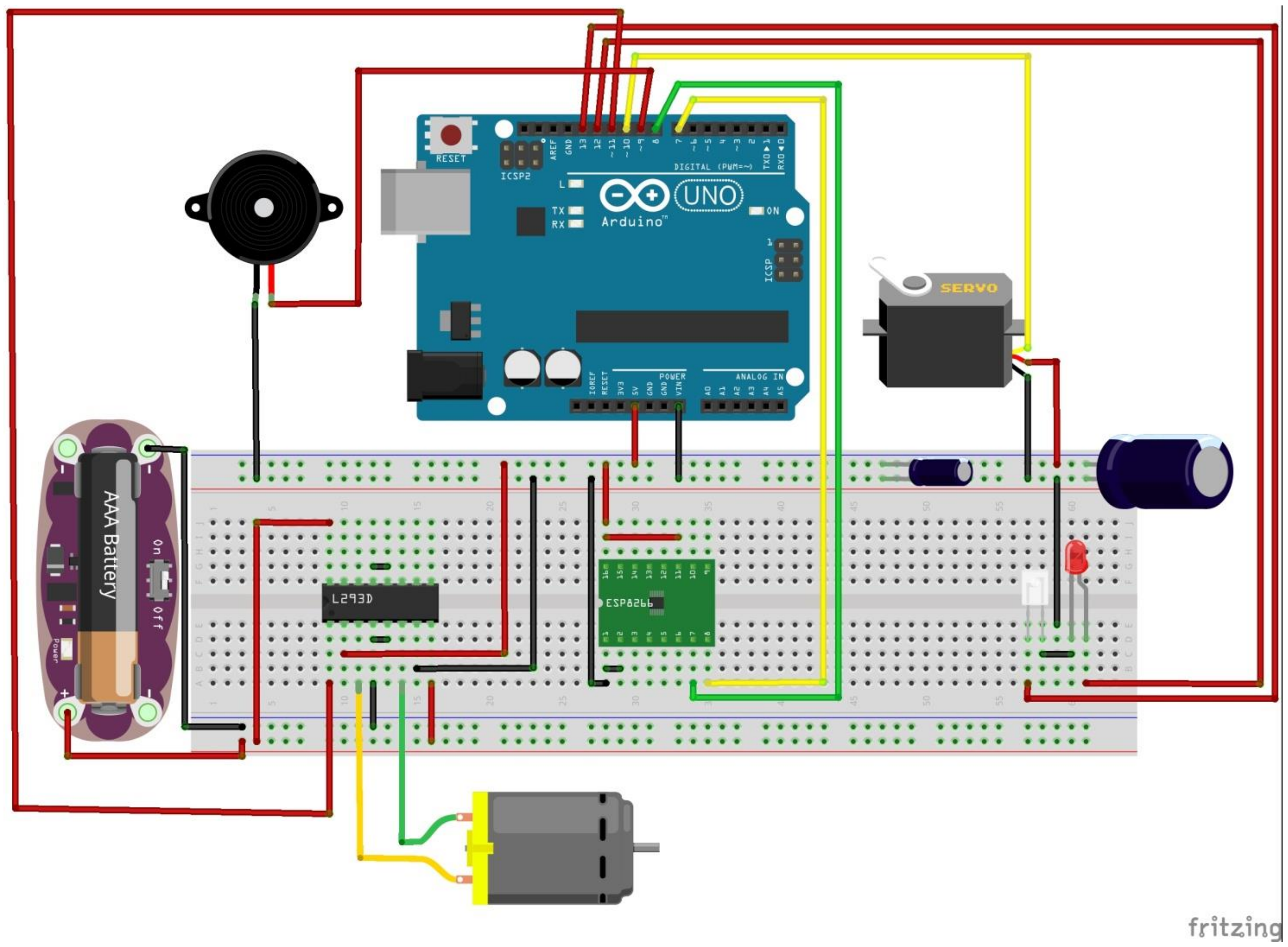


Fig 4: Circuit Connection Diagram

7. Current Prototype:

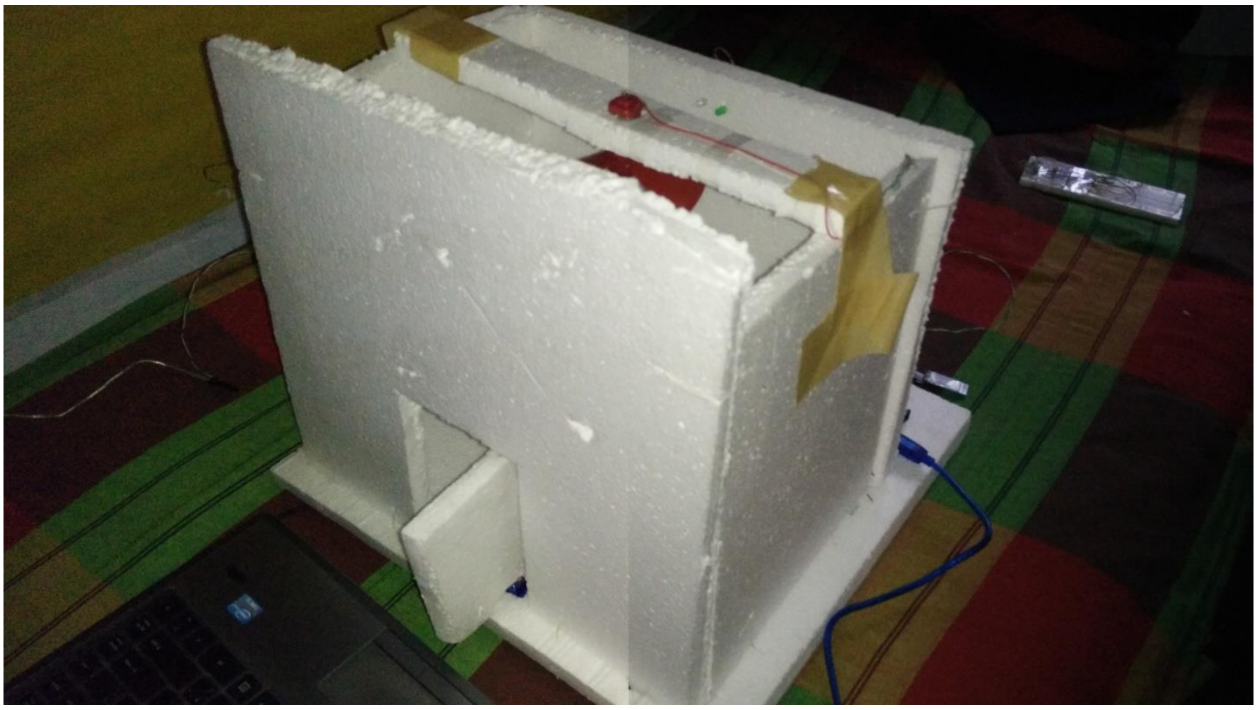


Fig 5: Basic prototype of the SRS

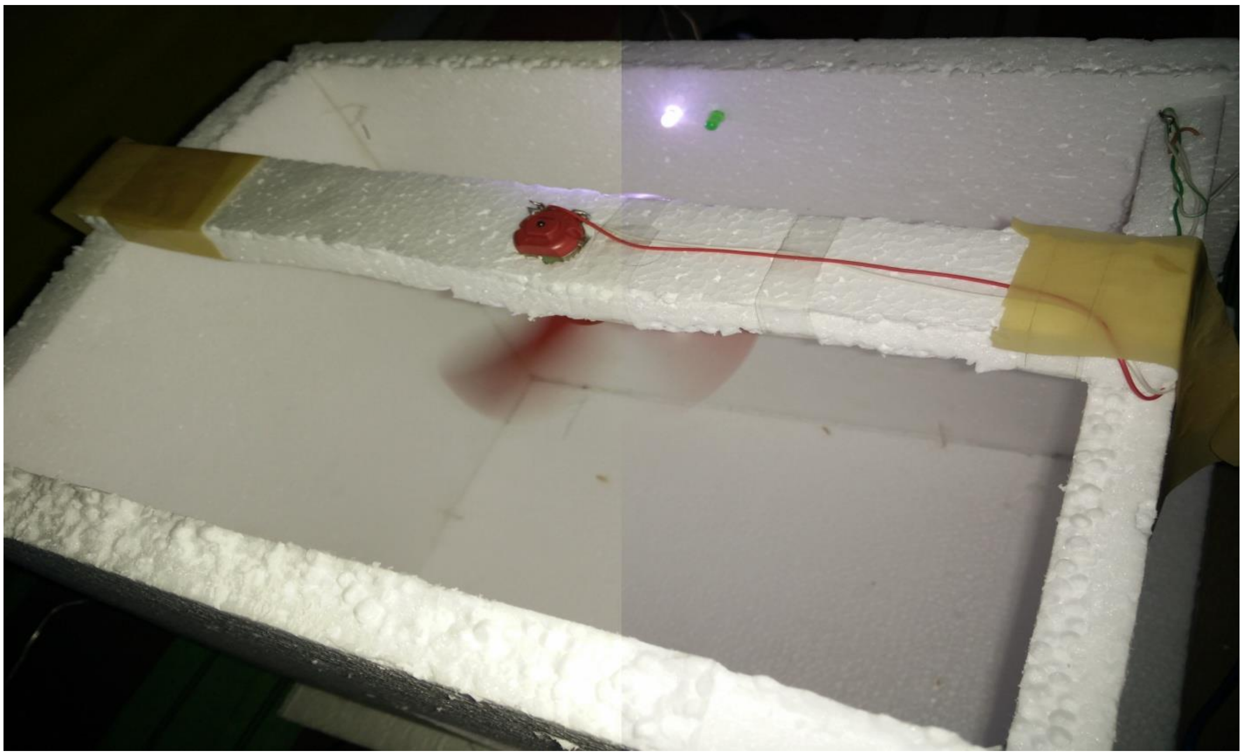


Fig 6: Fan Control

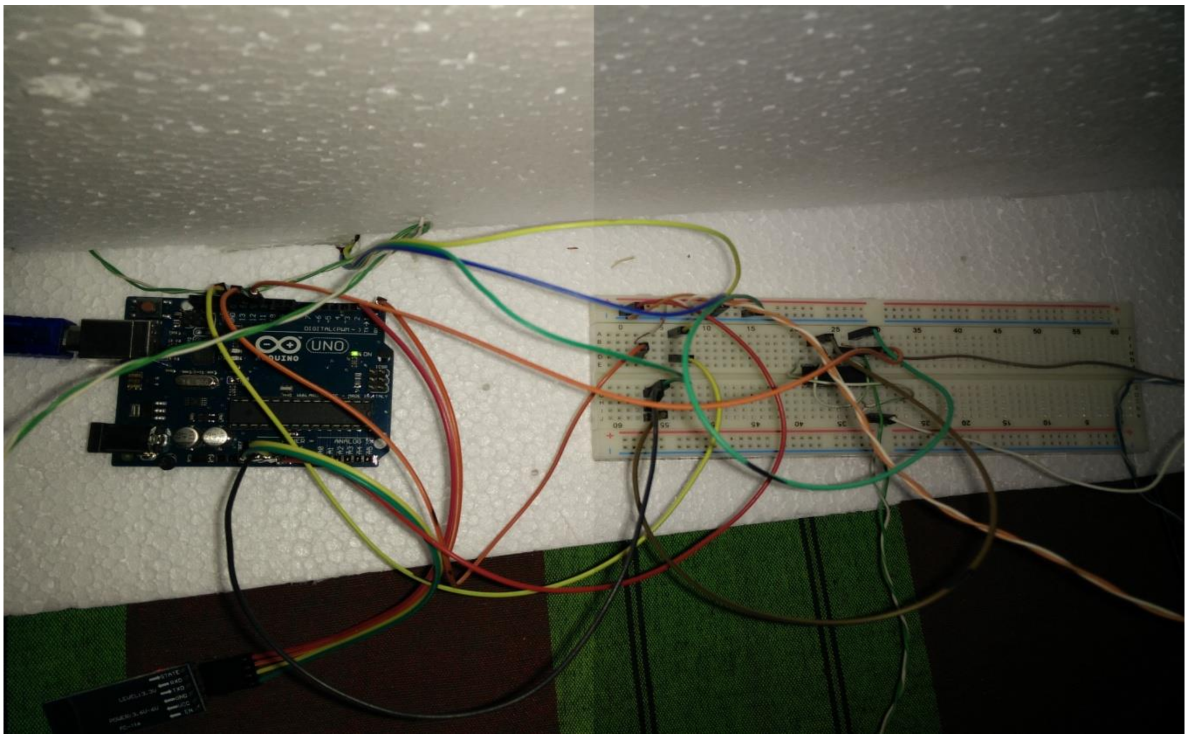
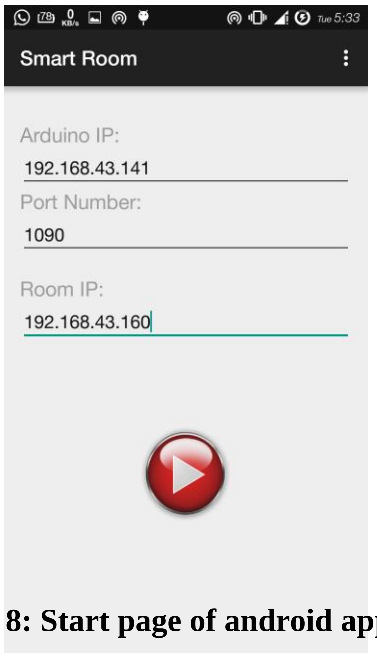


Fig 7: Circuitry in the background

8. A Brief Discussion on how 'Smart Room' Works:

The application starts with an instance of input activity class, which consists of three input text fields. They are:



Smart Room

Arduino IP:
192.168.43.141

Port Number:
1090

Room IP:
192.168.43.160

Fig 8: Start page of android app.

- Arduino IP
- Arduino Port
- Stream Address of IP camera (Home)

The input texts are stored into the 'Shared Preferences' (cache) of android operating system. This saved string data is transferred through the **Intent** call of main activity class.

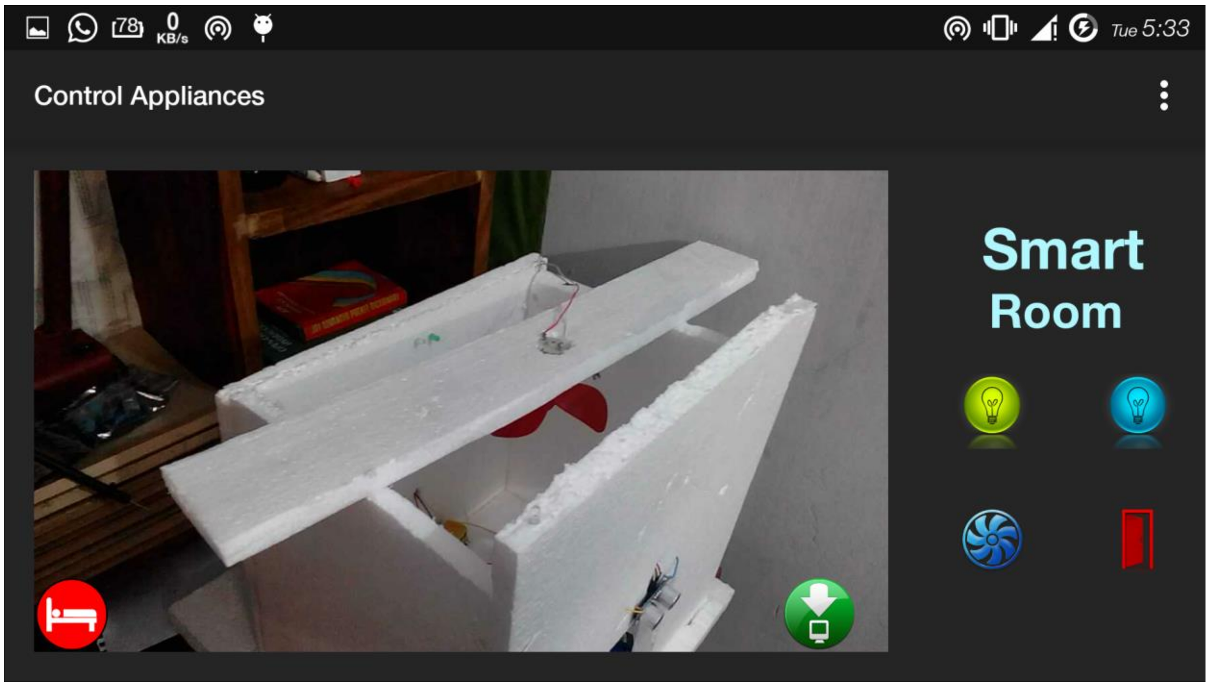
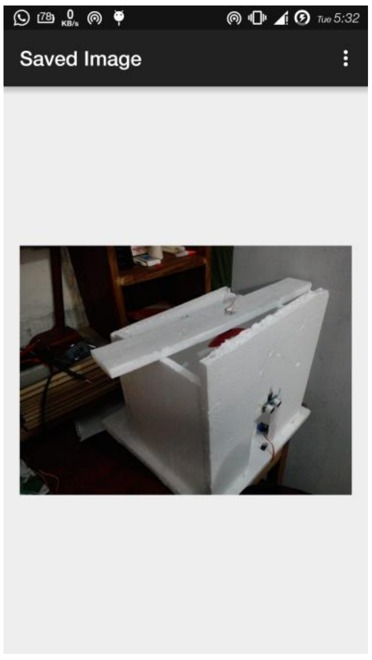


Fig 9: Control Interface of android app.

The Main Activity class consists of different layouts which hold Web View, capture Image Button, Led Bulb Control Buttons, Fan Control Button, and Door Control Button.

1. **Web View**: The Web view displays the live video stream from IP cameras.
2. **Led Bulb Button**: It controls the led bulb inside the room. If the bulb is already ON, it toggles it to OFF.
3. **Fan Control Button**: This button control the operation of fan. Toggles the state of the fan when the button is clicked.
4. **Door Control button**: It controls the movement of the door which can be opened & closed through a simple click on this button.
5. **Save Image Button**: This button can instantly save the live streamed picture into device storage.



The “**Save Image**” button instantly captures image from IP camera.

Fig 10: Captured Image from Live Stream.

9. Control from Web Interface:

‘**Smart Room’** can also be controlled from any web browser. The UI of the web are similar to the android companion application.

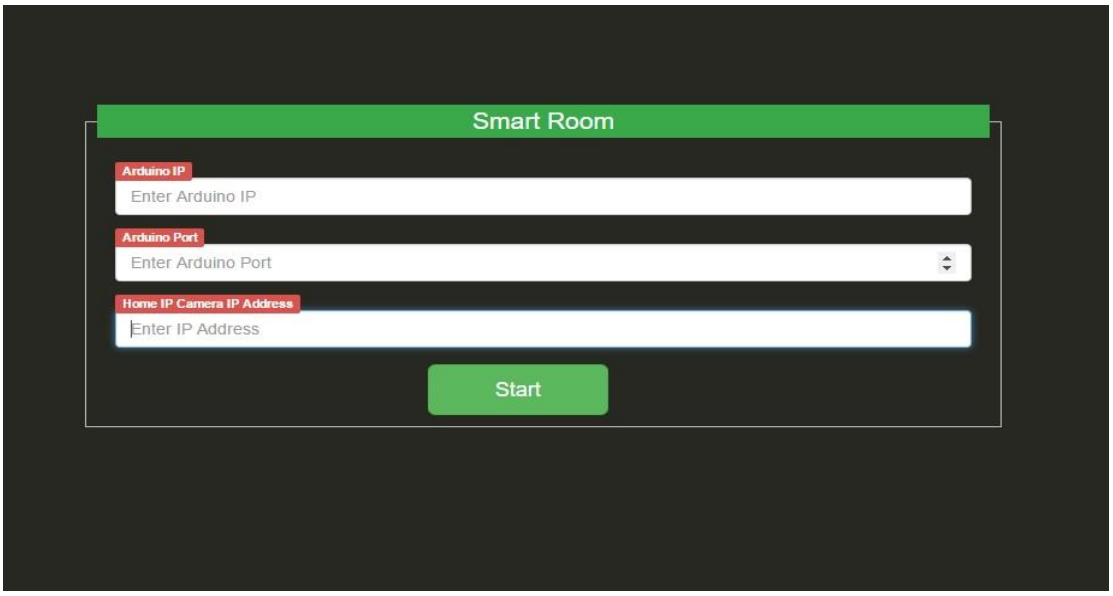


Fig 11: Start Page of Web UI.

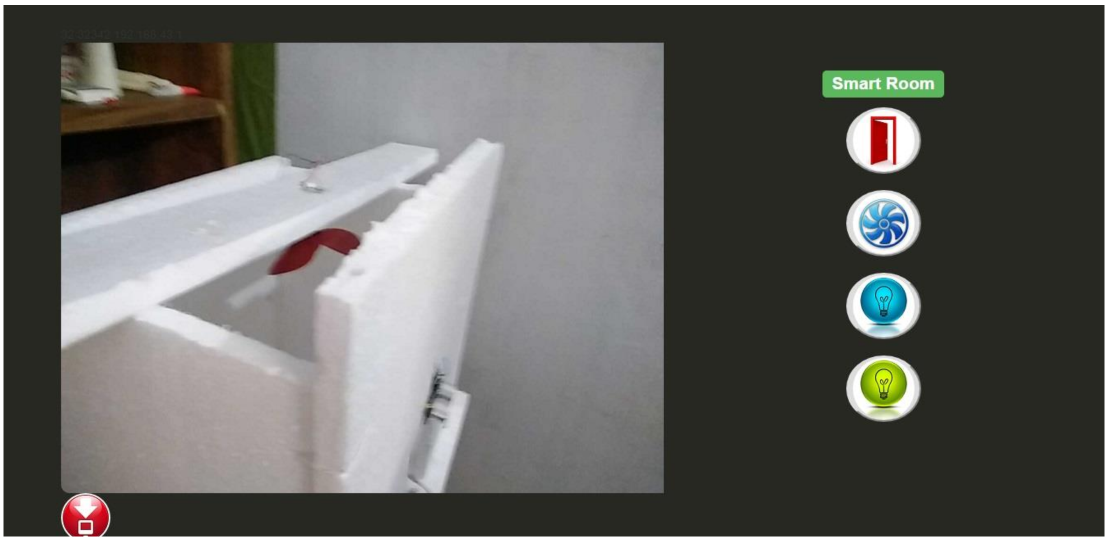


Fig 12: Control panel of Web UI.

10. Working process In Arduino:

When a button is clicked from the android apps, a corresponding pin is sent wrapped in a HTTP GetRequest, sent through the Wi-fi network to the Wi-fi module(Esp 8266) connected to the arduino.

After receiving the HTTP GetRequest, Arduino calculates the received data to determine the pin clicked from the android app. When the pin number is retrieved it toggles the corresponding pin.

Depending on the pin number & current pin state, the fan, led, door is turned ON or OFF. After executing this operation, Arduino sends an acknowledgement signal to the android app through the connected Wi-Fi module, telling the user which pin is turned ON or OFF.

10. Limitations

- The Wi-Fi module requires stable power supply to operate smoothly.
- Only a single component can be controlled at a time.
- We have to wait 3 seconds to properly control the corresponding device & ensure its change of state.

11. Future Improvement

- ✓ Support full time surveillance.
- ✓ Wide area coverage through Internet.
- ✓ Allow Remote Access through cloud based solution.
- ✓ More compact design.
- ✓ Modularity.
- ✓ Voice control over natural language processing.
- ✓ Control on the go.
- ✓ Thief alert.
- ✓ Baby Monitoring.
- ✓ Old age monitoring.
- ✓ Home atmosphere control.

12. Conclusion

The “Smart room” as a standalone system tries to complete the services it promises to provide to the user. The target user group of this system is increasing day by day as more and more people are becoming interested in automated system approach.

13. Reference

I. **‘**Programming Your Home’
  ---by **Mike Riley**

II. ‘Professional Android Sensor Programming’
  ---by **Greg Milette, Adam Stroud**

III. ‘Professional Android Open Accessory-Programming With Arduino’
  ---by **Andreas Goransson, David Ruiz.**

IV. http://developer.android.com/

V. http://youtube.com/

VI. http://instructables.com/

VII. http://stackoverflow.com/

VIII. https://www.**arduino**.cc/

IX. www.**esp8266**.com/

X. http://espressif.com/

XI. http://w3school.com/

XII. https://cdn.sparkfun.com/assets/learn_tutorials/4/0/3/4A-ESP8266__AT_Instruction_Set__EN_v0.30.pdf

Arduino Sketch

```
#include <SoftwareSerial.h>
#include <Servo.h>
#define trigPin 2
#define echoPin 4
#define led 5
#define led2 6

Servo myservo;

int pos = 0;


#define DEBUG true

SoftwareSerial esp8266(7,8);
void setup()
{
  Serial.begin(9600);
  esp8266.begin(9600);

  pinMode(10,OUTPUT);
  digitalWrite(10,HIGH);

  pinMode(11,OUTPUT);
  digitalWrite(11,LOW);

  pinMode(12,OUTPUT);
  digitalWrite(12,LOW);

  pinMode(13,OUTPUT);
  digitalWrite(13,LOW);

   pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  pinMode(led, OUTPUT);
  digitalWrite(led,LOW);

  pinMode(led2, OUTPUT);
  digitalWrite(led2,HIGH);

  myservo.attach(3);
  myservo.write(pos);




  sendCommand("AT+RST\r\n",2000,DEBUG);
  sendCommand("AT+CWMODE=3\r\n",1000,DEBUG);
  sendCommand("AT+CWJAP=\"room\",\"12345678\"\r\n",3000,DEBUG);
  delay(5000);
  sendCommand("AT+CIFSR\r\n",1000,DEBUG);
  sendCommand("AT+CIPMUX=1\r\n",1000,DEBUG);
  sendCommand("AT+CIPSERVER=1,1090\r\n",1000,DEBUG);
  sendCommand("AT+CIFSR\r\n",1000,DEBUG);

  Serial.println("Server Ready");
}

void loop()
{

  if(esp8266.available())
  {

    char pid[] = "+IPD,";
    if(esp8266.find(pid))
    {
     delay(500);
     int connectionId = esp8266.read()-48;
     char pin[] = "pin=";
     esp8266.find(pin);

     int pinNumber = (esp8266.read()-48);
     int secondNumber = (esp8266.read()-48);
     if(secondNumber>=0 && secondNumber<=9)
     {
      pinNumber*=10;
      pinNumber +=secondNumber;
       digitalWrite(led,LOW);
      if(pinNumber==10)
      {
         servomotor();
      }

      else { digitalWrite(pinNumber, !digitalRead(pinNumber));  }
     }

     String content;
     content = "Pin ";
     content += pinNumber;
     content += " is ";

     if(digitalRead(pinNumber))
     {
       content += "ON";
     }
     else
     {
       content += "OFF";
     }

     sendHTTPResponse(connectionId,content);
     String closeCommand = "AT+CIPCLOSE=";
     closeCommand+=connectionId;
     closeCommand+="\r\n";

     sendCommand(closeCommand,1000,DEBUG);
    }
  }
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
  if (distance < 50) {
    digitalWrite(led,HIGH);
  digitalWrite(led2,LOW);
}
  else {
    digitalWrite(led,LOW);
    digitalWrite(led2,HIGH);
  }
  if (distance >= 200 || distance <= 0){

  }
  else {

  }
}


void servomotor()
{
         if(pos == 0)
          {
           int i=0;
             for(;i<=90;i++)
             {
             myservo.write(i);
             pos=i;
             delay(10);
             }
           }
          else if(pos == 90)
           {
              int i=90;
               for(;i>=0;i--)
               {
               myservo.write(i);
               pos=i;
               delay(10);
               }
            }

            delay(500);
 }

String sendData(String command, const int timeout, boolean debug)
{
    String response = "";

    int dataSize = command.length();
    char data[dataSize];
    command.toCharArray(data,dataSize);
    esp8266.write(data,dataSize);
    if(debug)
    {
      Serial.println("\r\n====== HTTP Response From Arduino ======");
      Serial.write(data,dataSize);
      Serial.println("\r\n========================================");
    }

    long int time = millis();

    while( (time+timeout) > millis())
    {
      while(esp8266.available())
      {


        char c = esp8266.read();
        response+=c;
      }
    }

    if(debug)
    {
      Serial.print(response);
    }

    return response;
}

void sendHTTPResponse(int connectionId, String content)
{
     String httpResponse;
     String httpHeader;
     httpHeader = "HTTP/1.1 200 OK\r\nContent-Type: text/html; charset=UTF-8\r\n";
     httpHeader += "Content-Length: ";
     httpHeader += content.length();
     httpHeader += "\r\n";
     httpHeader +="Connection: close\r\n\r\n";
     httpResponse = httpHeader + content + " ";
     sendCIPData(connectionId,httpResponse);
}


void sendCIPData(int connectionId, String data)
{
   String cipSend = "AT+CIPSEND=";
   cipSend += connectionId;
   cipSend += ",";
   cipSend +=data.length();
   cipSend +="\r\n";
   sendCommand(cipSend,1000,DEBUG);
   sendData(data,1000,DEBUG);
}


String sendCommand(String command, const int timeout, boolean debug)
{
    String response = "";
    esp8266.print(command);

    long int time = millis();

    while( (time+timeout) > millis())
    {
      while(esp8266.available())
      {


        char c = esp8266.read();
        response+=c;
      }
    }

    if(debug)
    {
      Serial.print(response);
    }

    return response;
}
```