LAU

الجامعة اللبنانية الأميركية

Lebanese American University

Zaki Home Automation

Hrag Terzian

Advisor: Azzam Mourad

CSC599

Table of Content:

1. Preface

This document is written to provide an overview about the Zaki Home Automation and specifics about the hardware used and the details about the general features, the application structure, and screenshot.
The report is submitted to Dr. Azzam Mourad as part of the requirements for the course: CSC599: Capstone Project.

2. Glossary

- **Node:** A device controlled by an esp8266 chip, that has wifi capabilities and arduino os. Used to switch on or off using relays any electronic device
- **Controller:** a central device, usually a raspberry pi, that connects to all the nodes in the house using http get, and hosts a java socket server to get inputs from the users
- **Capacitor:** A capacitor is a passive two-terminal electrical component that stores potential energy in an electric field.
- **Relay:** A relay is an electrically operated switch.
- **Arduino:** 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 and digital world.

3. Introduction

Home automation is redefining what it means to build an intelligent smart home. From switching off the lights when no-one's in the room, to automatically setting the right temperature before you get home from work, the latest smart devices are making hands-free home automation a reality. Turning your home into a fully automated smart home is now more accessible and affordable than ever before, thanks to the advances in smart technology over the recent years. Whether you want to automate single household items like your light switches or door locks, or wanting to fully automate a connected system of intelligent devices, the combinations are limitless.

A fully automated home is a complex machine made up of a number of independent but communicating parts, smart devices. Whether lighting, heating, securing, or providing entertainment, every smart device in the automated home environment does a different job, but all of them carry out at least one of the three basic role: sensing or detecting things, process information, or respond to an input, situation or command. The most common feature of smart devices is the ability to sense changes to the environment in which they are located, such as motion detection light or temperature sensing. These sensors are like the eyes and ears of the automated home, they receive inputs from the surrounding environment and either react accordingly themselves or pass the information on to other devices in the home. Sometimes they do both. For example, lights can be triggered to turn off as a response to your thermostat knowing you're away from home. Basically, the smart 'sensing' devices provide the information for the other elements of your automated home to decide what they need to do next. Once a sensor is activated, the device that then actually carries out a specific task in response to this input (known as an actuator) gets to work. For example, while Amazon Echo may be the device receiving your voice command to make coffee, it's the smart coffee maker that actually gets the job done. With a home automation system in place, you can quite literally have parts of your daily home routine happen automatically.

4. Background

The idea behind Zaki, is the ability to automate your home using simple hardware and software, the can be accessible by anyone with a simple background of electricity and components, coding, and basic input output methods. Zaki system uses a combination of wifi connected arduino, “nodes”, and a raspberry pi, “Controller”, to automate anything that is controlled with a switch or infrared remote. The Zaki source code and nodes schematic will be available as open source for anyone with the curiosity of smart home to be able to automate his house cheaply. Cost of the project will vary according to the size of the house and the number of nodes needed: Cost of raspberry pi: 70$, Cost per node: 10$. This cost is relatively cheap in contrast of the other systems on the market

5. Design and Features

5. a. Nodes

Zaki Home Automation incorporates 3 main programs- the nodes, the controller, and the mobile application. The nodes are made up of an esp8266 wifi arduino controller, relays, transistors, resistors, leds and many other electrical components(Figure 1. & 2.). The arduino code uses two methods, the setup, and the loop. In the setup, we setup up every pins to its function, input or output, and the wifi settings. The loop keeps on looping to check if there is a connection or an input from the physical buttons, if the nodes include buttons. The command given to the nodes come from the url of the http get request sent from the controller.

5. a. i. Hardware

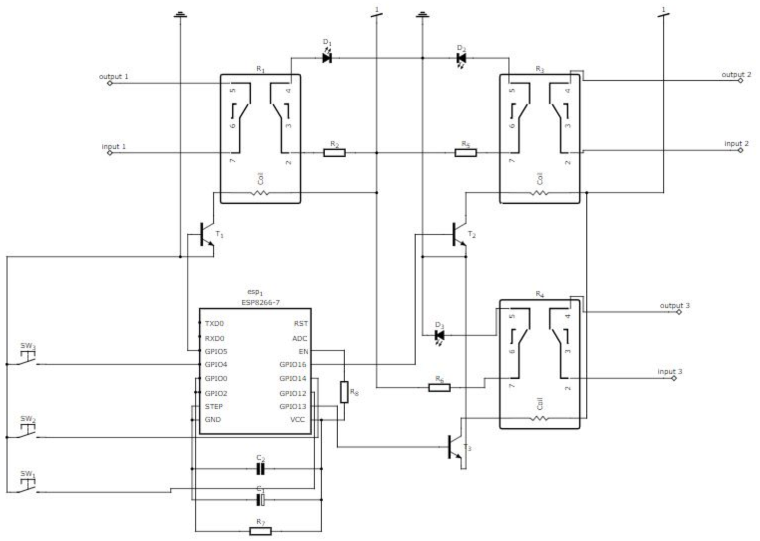

Figure 1. Node's circuit design includes 3 buttons and 3 relays

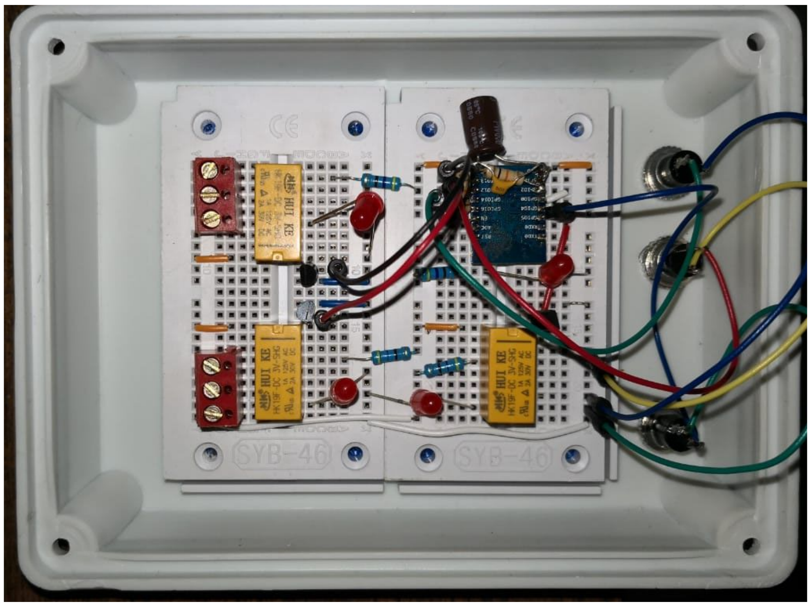

Figure 2. Example of node

The esp8266 is connected to two capacitors, one to reduce electrical nodes, and the second is for voltage stabilization, three buttons are connected to the input pins on the esp8266 that are used as physical switches. Three transistors are connected to the output pins. The transistors, when activated, opens the relays with then opens the circuit for the lights or a/c. The leds are also placed for visual status of the switches.

5. a. i. Code

5. a. i. 1. Setup

```
void setup() {
        ESP.wdtEnable(100000);
        Serial.begin(115200);
        delay(10);
       //setup pin modes
        pinMode(doorPin, OUTPUT);
        digitalWrite(doorPin, LOW);

        pinMode(light1out, OUTPUT);
        digitalWrite(light1out, LOW);

        pinMode(light2out, OUTPUT);
        digitalWrite(light2out, LOW);

        pinMode(doorIn,  INPUT_PULLUP);
        pinMode(light1in,  INPUT_PULLUP);
        pinMode(light2in,  INPUT_PULLUP);


        // Connect to WiFi network
        Serial.println();
        Serial.println();
        Serial.print("Connecting to ");
        Serial.println(ssid);
        WiFi.mode(WIFI_STA);
        WiFi.begin(ssid, password);

        while (WiFi.status() != WL_CONNECTED) {
                delay(500);
                Serial.print(".");
        }
        Serial.println("");
        Serial.println("WiFi connected");

        // Start the server
        server.begin();
        Serial.println("Server started"); this URL to connect: ");
        Serial.print("http://");
        Serial.print(WiFi.localIP());
        Serial.println("/");
}
```

Figure 3. Setup Function

The esp is flashed with arduino controller code with two functions, setup(Figure 3.) and loop(Figure 4.). The setup function's main purpose is setting up the pin modes, as input pin or output using the pinMode() function, and the wifi connection to the main router, using the WiFi.begin().

5. a. i. 2. Loop

```
void loop() {

 //switch controller
 if(digitalRead(light1in)==LOW){

  light1=!light1;
  digitalWrite(light1out,light1);
  delay(1000);
 }

  if(digitalRead(light2in)==LOW){

  light2=!light2;
  digitalWrite(light2out,light2);
  delay(1000);

 }


  if(digitalRead(doorIn)==LOW){

  door=!door;
  digitalWrite(doorPin,door);
  ESP.wdtFeed();
  delay(3000);
  door=!door;
  digitalWrite(doorPin,door);

 }



 ESP.wdtFeed();

 WiFiClient client = server.available();
 if (!client) {
  return;
 }

 // Wait until the client sends some data

 while(!client.available()){
  ESP.wdtFeed();
  delay(1);
 }

 // Read the first line of the request
 String request = client.readStringUntil('\r');

 client.flush();

 // Match the request


 if (request.indexOf("/12UNLOCK") != -1) {
  digitalWrite(doorPin, HIGH);

 }
 if (request.indexOf("/13LOCK") != -1){
  digitalWrite(doorPin, LOW);

 }
 if (request.indexOf("/14PULSE") != -1){
  digitalWrite(doorPin, HIGH);
  ESP.wdtFeed();
  delay(3000);
  digitalWrite(doorPin, LOW);

 }
if (request.indexOf("/15TOGGLE1") != -1){
  light1=!light1;
  digitalWrite(light1out,light1);

 }
 if (request.indexOf("/16TOGGLE2") != -1){
  light2=!light2;
  digitalWrite(light2out,light2);

 }
 //shake: hand shake to know what are the relays and what are their commands
 String shake="";
  if (request.indexOf("/0SHAKE") != -1){
   shake="DOOR/12UNLOCK/13LOCK/14PULSE//LIGHT1/15TOGGLE1//LIGHT2/16TOGGLE2//\n";
  }

 // Return the response
 client.println("HTTP/1.1 200 OK");
 client.println("Content-Type: text/html");
 client.println(""); //  do not forget this one
 client.println("<!DOCTYPE HTML>");
 client.println("<html>");
 client.print(shake);

 String Tdoor="//DOOR/";
 if(door==HIGH)Tdoor=Tdoor+"OPEN";
 else Tdoor=Tdoor+"CLOSED";
 String Tlight1="//LIGHT1/";
 if(light1==HIGH)Tlight1=Tlight1+"ON";
 else Tlight1=Tlight1+"OFF";
 String Tlight2="//LIGHT2/";
 if(light2==HIGH)Tlight2=Tlight2+"ON";
 else Tlight2=Tlight2+"OFF";
 String hrag = Tdoor+Tlight1+Tlight2;
 client.print(hrag);
 delay(1);
 Serial.println("Client disconnected");
 Serial.println("");
}
```

Figure 4. The loop function

The loop function runs indefinitely, it checks if a switch has been changed and processes it accordingly. The loop also checks if the controller, the client, has connected to the node, if a client has connected it checks the request and using simple if-else-if changes the output of the pins according to the command sent by the controller.

5. b. Controller

In addition to the nodes, Zaki consists of one controller, usually a Raspberry Pi, that manages the inputs from the user or the sensors and sends the outputs to the specific node. The controllers program is made up of several classes and functions, the main classes are Server Class, the main Zaki Class, and the Logger Class. The server class starts a java socket server and waits for and input by the user. It then parses the input and sends the specific command to the Zaki Class, where the command is processed and then sent to the specific node. The loggers job is to log every time the user arrives home and the time the user leaves home, and the uses the log to predict a schedule for the user by day and time. This schedule can then be used to turn off the lights when the user is out of the house, or turn on the air conditioner 15 minutes before the user arrives.

5. b. i. Code Flow

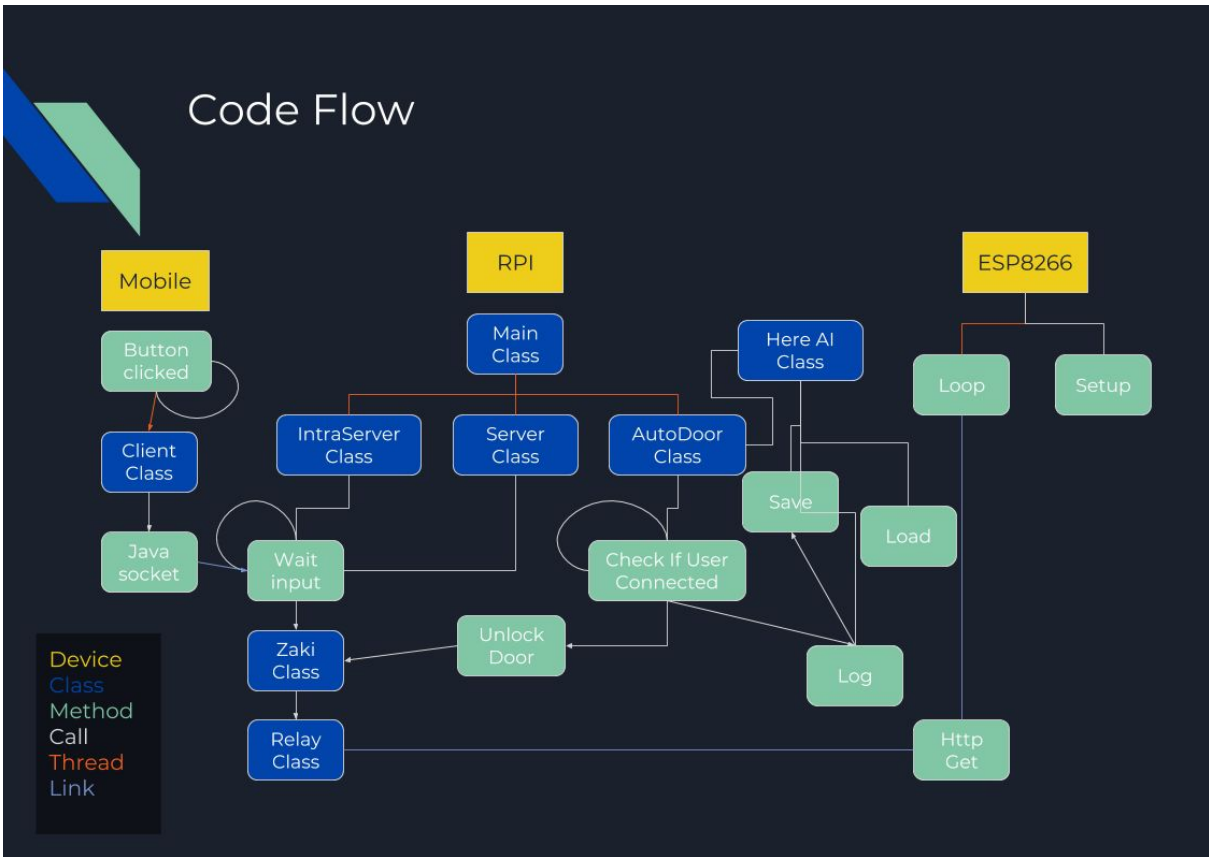

Figure 5. Code Flow chart specifying the classes used and their relations

5. b. i. 1. Server Class

```
public void runServer() {

                    String command;
                    String result;
                    try {
                              ServerSocket welcomSocket = new ServerSocket(6788);

                              while (true) {
                                        Socket connectionSocket = welcomSocket.accept();
                                        BufferedReader inFromClient = new BufferedReader(
                                                            new
InputStreamReader(connectionSocket.getInputStream()));
                                        DataOutputStream outToClient = new
DataOutputStream(connectionSocket.getOutputStream());
                                        command = inFromClient.readLine();

                                        result = zaki.execute(command);

                                        outToClient.writeBytes(result);
                              }
                    } catch (Exception E) {
                              System.out.println(E.toString());
                    }
          }
```

Figure 6. Controller Server Function

The server runs indefinitely and wait for a connection from a client, when a connection is received the command is then parsed and sent to the Zaki Class

5. b. i. 2. Zaki Class

```
private Relays main = new Relays("192.168.1.10");
        public Zaki() {

        }
        public String execute(String com) {
                switch (com) {
                case "OPEN" :
                        return main.command("12UNLOCK");
                case "CLOSE" :
                        return main.command("13LOCK");
                case "PULSE" :
                        return main.command("14PULSE");
                case "LIGHT1" :
                        return main.command("15TOGGLE1");
                case "LIGHT2" :
                        return main.command("16TOGGLE2");

                }
                return "error";
        }
```

Figure 7. Zaki Class that processes the input

Using a switch, the zaki class checks readable commands that were given by the client, and sends out the specific command to the respectful relay accordingly.

5. b. i. 3. Relay Class

```
public String command(String com) {
			StringBuilder result = new StringBuilder();
			try {
				System.out.println("connecting");
				URL url = new URL("http://" + ip + "/" + com);
				HttpURLConnection conn = (HttpURLConnection) url.openConnection();
				conn.setRequestMethod("GET");
				BufferedReader rd = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
				String line;
				while ((line = rd.readLine()) != null) {
					result.append(line);
				}
				rd.close();
			} catch (Exception E) {
				System.out.println("relays error");
				System.out.println(E.toString());
			}
			return result.toString();
		}
```

Figure 8. Relay command function

The relay class then parses the command and creates an http connection, which sends the request as a GET function, specifying the command in the url.

5. b. i. 4. Log Class

```
public void fix(int day) {

                    float hscore = 0;

                    for (int i = 0; i < hrag[day].size() - 1; i++) {

                              float i1 = hrag[day].get(i)[1];
                              float i2 = hrag[day].get(i + 1)[1];
                              float time1 = hrag[day].get(i)[0];
                              float time2 = hrag[day].get(i + 1)[0];
                              float score1 = hrag[day].get(i)[2];
                              float score2 = hrag[day].get(i + 1)[2];
                              if (hscore < score1)
                                        hscore = score1;
                              if (hscore < score2)
                                        hscore = score2;

                              if (i1 == i2 && (time2 - time1) <= 1) {

                                        float temp = hrag[day].get(i)[0] + hrag[day].get(i + 1)[0];
                                        hrag[day].get(i)[0] = temp / 2;
                                        if (score2 > score1)
                                                  hrag[day].get(i)[2] = score2;
                                        hrag[day].remove(i + 1);
                                        hrag[day].get(i)[2] = score1 + 10;
                                        i--;

                              } else
                                        hrag[day].get(i)[2] = score1--;
                    }

          }
```

Figure 9. Logger Class, predicting schedule function

The logger class logs the time of arrivals and departure of each user by pinging each ip address configured before hand. Then the logger class check the whole log each time a new it gets a new entry and adds to the "score" of the entry if it is repeated, or subtracts from the "score" if the entry does not happen again. Each entry with a score of zero will be considered as a one time thing and not a regular part of a user's schedule and thus will be deleted. This way Zaki would have a simple but effect way of learning the users behavior and schedule. With this log Zaki can then be programmed to turn on the a/c before the user arrives or turn off all lights and power when the user is not at home. The possibilities are endless!

With this code design, the controller would work efficiently and simply. The benefits of this style of code flow are mainly to help people without advanced programming knowledge to implement this system in his/her house.

5. c. Mobile

The Zaki project also includes a mobile app that send commands to the controller. For now the mobile app is for android only, and it uses simple java socket client to connect to the java socket server on the controller.

```
public String request(String r) {

        String host = "hragterzian.hopto.org";
        Socket clientSocket = null;
        String reply="nopthing";

        try {
            clientSocket = new Socket(host, 6788);

            DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());
            BufferedReader inFromServer = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));


            outToServer.writeBytes(r + '\n');
            reply = inFromServer.readLine();


            clientSocket.close();

        } catch (IOException e) {
            e.printStackTrace();
        }
        return reply;
    }
```

Figure 10. Mobile app java socket client

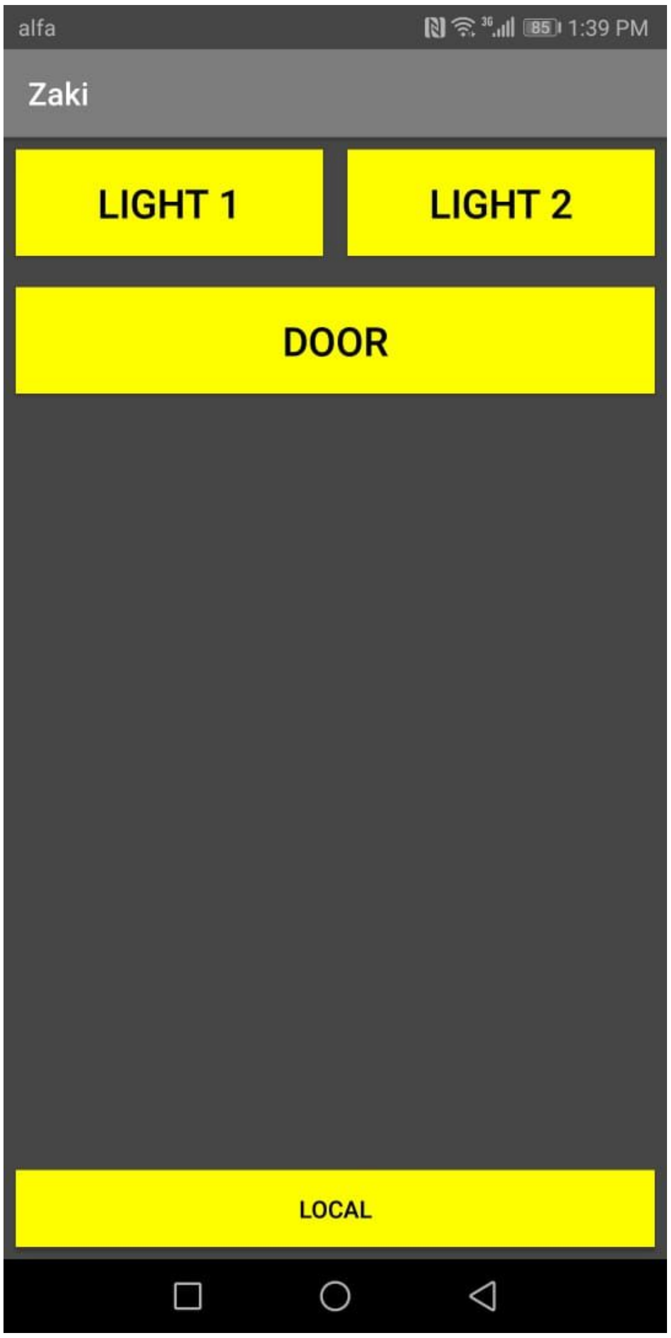

Figure 11. Mobile App Layout

The mobile app is fairly simple, buttons to send commands to the controller, but future upgrades are necessary. First, configuration page will be added to be able to change settings on the controller, i.e. timer for the coffee machine or time before the a/c automatically turns on. Second, buttons will be dynamically added, the application will talk to the controller to check how many commands can be sent to it, from the zaki.java class. Third, passwords and encryptions will be used so that each user or phone has a specific code, where an admin can grant or revoke specific permissions of each user.

6. Future Upgrades

In computer science, there is always room for improvement and upgrades. Zaki Home Automation will have more features added to it periodically. First, the most important upgrade will be accessing or sending commands to the controller using the internet when not at home, but this feature needs to be studied more and implemented delicately to avoid security issues and hacks. Second, Zaki nodes will be upgraded to support over the air upgrades, this is especially useful for nodes that are hard to reach or permanently installed. Finally, Zaki will also incorporate voice commands and amazon echo to control the nodes using voice.

7. Conclusion

In conclusion, the zaki home automation project is a simple, affordable and upgradable solution to all home automation need. Furthermore, with the help of the development community and open source, after a while, the Zaki project will house many other new features that will help the user live in a more smart world. IOT is the future of this world, every electrical appliance will be connected to the internet, it’s time to invest time and research into this field.