ABSTRACT

With the advent of technology, life has become faster in pace and shorter in interactions, with others, as well as with the surroundings. In such a scenario, there is a need to have an endeavor to have everything at the push of a button away, and more importantly, automated. Home Automation is such an endeavor, in which, all the electrical appliances present at home are connected to each other, having interactions with sensors placed at strategic positions in a closed loop manner in order to perform meager tasks automatically, leaving less burden on the humans. With this project we are promoting the fact that Home Automation can greatly contribute to energy conservation too.

Keywords— Home Automation, ESP8266, sensors

TABLE OF CONTENTS

CHAPTER NO.01

INTRODUCTION

INTRODUTION

A brief definition of Home Automation [1] may be given in this way as, "Home Automation is the technology, in which every electrical appliance present inside a particular house are connected to one another and to a set of *"sensors"* placed at particular strategic positions, reading specific data in a closed loop fashion to serve the purpose to automate all the connected home appliances."

The project aims at designing an advanced home automation system using normal web server and Wi-Fi technology. The devices can be switched ON/OFF and sensors can be read using a Personal Computer (PC) through Wi-Fi.

The most important part in a fully automated system are the sensors, be it light sensors, heat sensors, smoke detectors etc. The data they acquire are then sent to the microcontroller unit, which then processes the data and performs specific switching of the Home Appliances in a real time fashion.

Wi-Fi (Short for Wireless Fidelity) is a wireless technology that uses radio frequency to transmit data through the air. Wi-Fi has initial speeds of 1mbps to 2mbps. Wi-Fi transmits data in the frequency band of 2.4 GHz. It implements the concept of frequency division multiplexing technology. Range of Wi-Fi technology is 40-300feet.

The controlling device for the automation in the project is a Arduino UNO [2] based ESP8266MOD . The data sent from PC over Wi-Fi will be received by Wi-Fi module connected to ESP8266 Module. ESP8266 reads the data and decides the switching action of electrical devices connected to it through Relays.

CHAPTER NO.02

OBJECTIVE OF THE PROJECT

OBJECTIVE OF THE PROJECT

In this project i have set these specific objectives for our version of the automated system.

- Turn ON or Turn OFF the appliances when it is needed
- He/They can then turn them ON or OFF using the default IR remote, or use a web application or server for the purpose as convenient.
- A person can remotely access the home appliances if he feels the need be.
- A person can remotely access the home appliances to check the status.
- A person can check the temperature on real time by using web server remotely.

In addition to these there are many more possibilities that we can explore with Home Automation systems [3] .

CHAPTER NO.03

LITERATURE SURVEY

LITERATURE SURVEY

REVIEW OF RELATED LITERATURE

When people think about home automation, most of them may imagine living in a smart home: One remote controller for every household appliance, cooking the rice automatically, starting air conditioner automatically, heating water for bath automatically and shading the window automatically when night coming. To some extent home automation equals to smart home. They both bring out smart living condition and make our life more convenient and fast.

REVIEW OF FOREIGN STUDIES

IN 2002,Tan, Lee and Soh proposed in their paper, the development of an Internet-based system to allow monitoring of important process variables from a distributed control system (DCS). This paper proposes hardware and software design considerations which enable the user to access the process variables on the DCS, remotely and effectively.

IN 2003, Potamitis, Georgila, Fakotakis, and Kokkinakis, G. suggested the use of speech to interact remotely with the home appliances to perform a particular action on behalf of the user. The approach is inclined for people with disability to perform real-life operations at home by directing appliances through speech. Voice separation strategy is selected to take appropriate decision by speech recognition

In the year 2006 , S. M. Anamul Haque,S. M. Kamruzzaman and Md. Ashraful

Islam proposed a system entitled " A System for Smart -Home Control of Appliances Based on Time and Speech Interaction" that controls the home appliances using the personal computer. This system is developed by using the Visual Basic 6.0 as programming language and Microsoft voice engine tools for speech recognition purpose. Appliances can be either controlled by timer or by voice command.

Ciubotaru-Petrescu, Chiciudean, Cioarga, and Stanescu (2006) present a design and implementation of SMS based control for monitoring systems. The paper has three modules involving sensing unit for monitoring the complex applications. A processing unit, that is microcontroller and a communication module that uses GPRS modem or cell phone via serial port RS-232. The SMS is used for status reporting such as power failure.

Jawarkar, Ahmed, Ladhake, and Thakare (2008) propose remote monitoring through mobile phone involving the use of spoken commands. The spoken commands are generated and sent in the form of text SMS to the control system and then the microcontroller on the basis of SMS takes a decision of a particular task.

Prof. Era Johri Dept. Of Information And Technology K.J.Somaiya College Of Engineering VIDYAVIHAR, MUMBAI " Remote Controlled Home Automation Using Android Application via WiFi Connectivity".

CHAPTER NO.04

SCOPE OF THE PROJECT

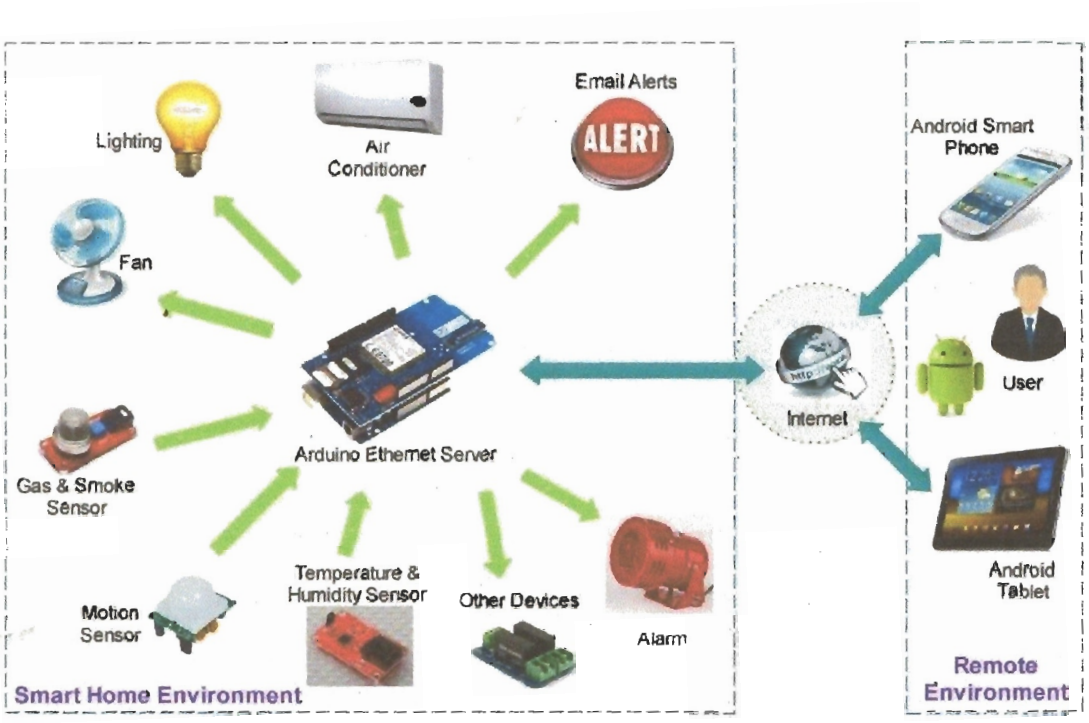


Fig: Samrt home environment

Day by day, the field of automation is blooming and these systems are having great impact on human beings. The project which is to be implemented is a home automation using Easy IOT [4] Web server and WIFI and has very good future development.

In the current system web server is installed on a windows PC so the home appliances can be controlled using only by using the device on which webserver is installed.This can be further developed installing webserver on cloud .

Advantage of installing webserver on the cloud is that home can be controlled by using any device which has WIFI 802.1 and a web browser. By visiting the IP address of the cloud the control actions can be taken.

In short

- It can be used commercially in industries
- It can be use for reduce power consumption
- It can be used in traffic road lighting system
- It can be used in lab, storage, industries, shopping mall or etc for automatic control of light, temperature or other perimeters control.
- This can be used for reduce human effort and time

CHAPTER NO.05

METHODOLOGY

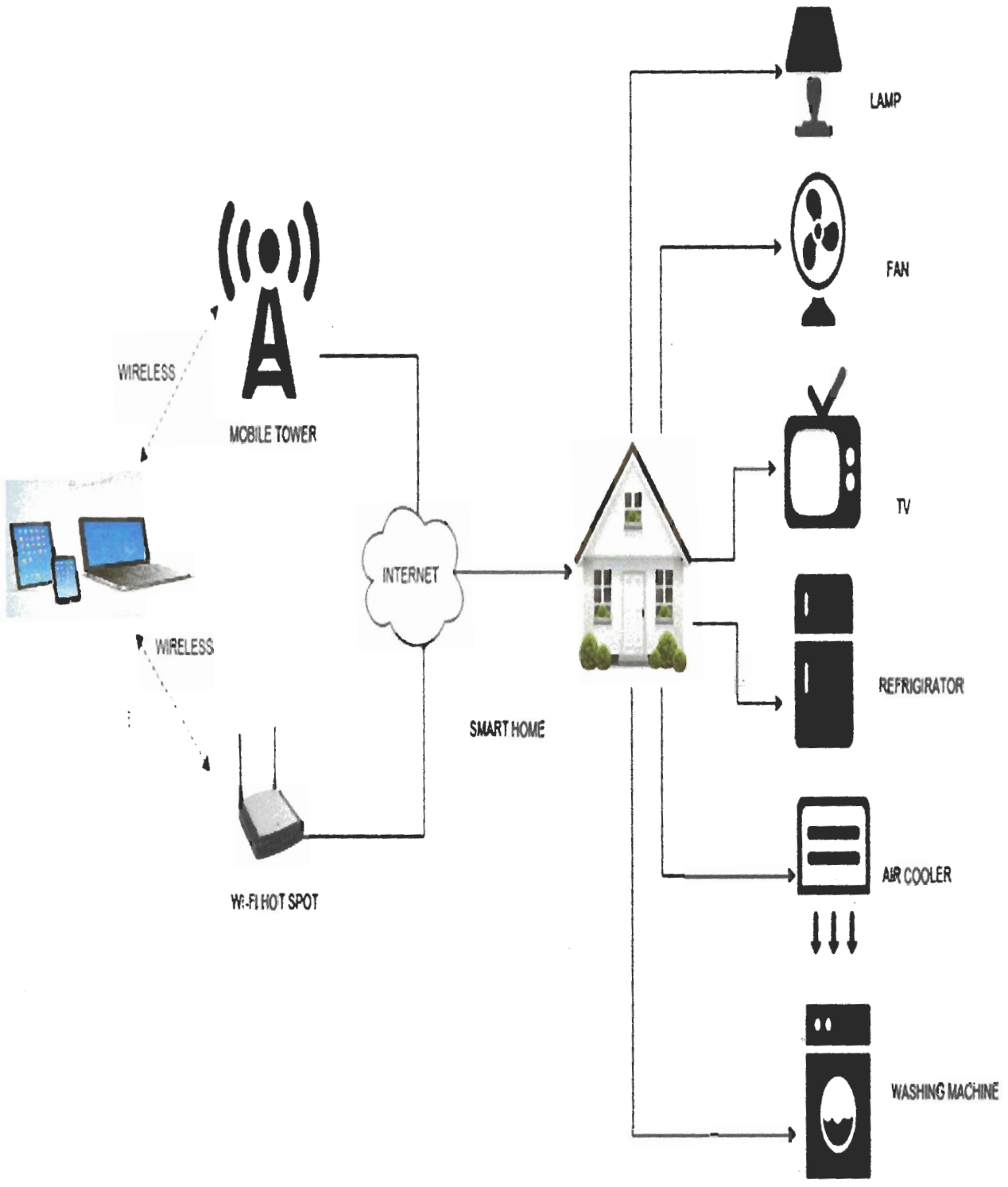
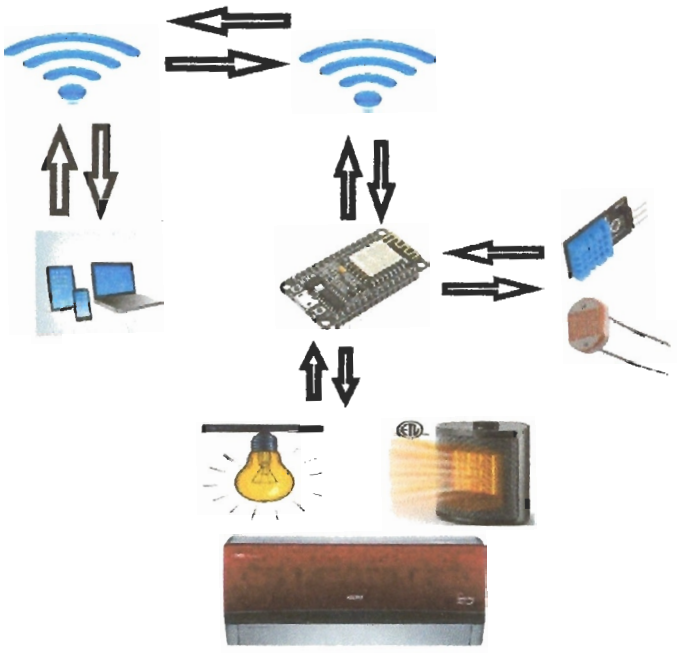


Fig: Method of Home Automation

In this project the heart is the Esp8266 module. This module will control the whole automation operation.



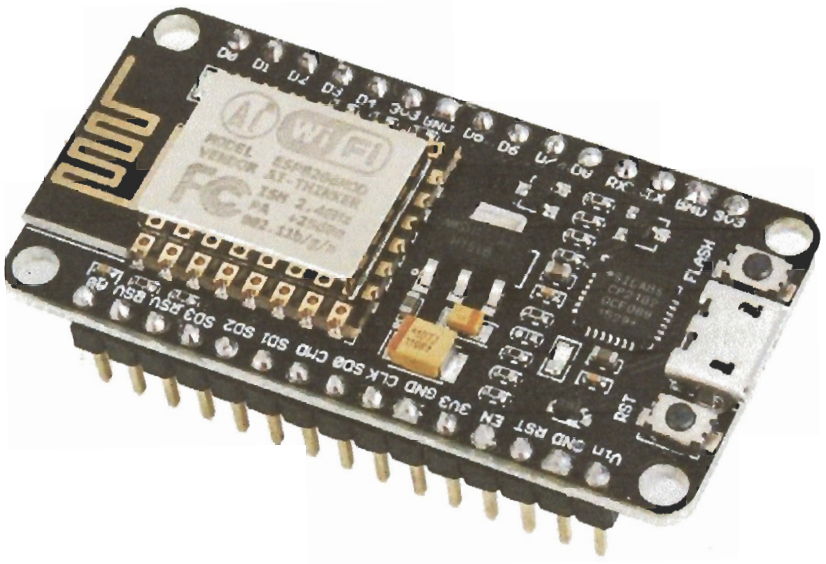
This module is connected to the internet via WIFI. User can check and control this by locally or over web by Smartphone or other smart devices.

- When the mode is local, it will automatically control the home appliances depending on the reading of the sensors installed in the house. To have full control a master switch is used to over-ride the control from the board.
- When the mode is web, all appliance will shut down and these can be turned on only by web switches.
- The web and local mode can be control from both web and the circuit easily

CHAPTER NO.06 (A)

HARDWARE

ESP 8266MOD:-



ESP 8266MOD or NodeMCU v3 [5], [6] is a development board which runs on the ESP8266 with the Espressif Non-OS SDK, and hardware based on the ESP-12 module. The device features 4MB of flash memory, 80MHz of system clock, around 50k of usable RAM and an on chip Wifi Transceiver.

Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers (and that' s just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existance interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support. In the Documents section below you will find many resources to aid you in using the ESP8266, even instructions on how to transforming this module into an IoT (Internet of Things) solution!

Features:

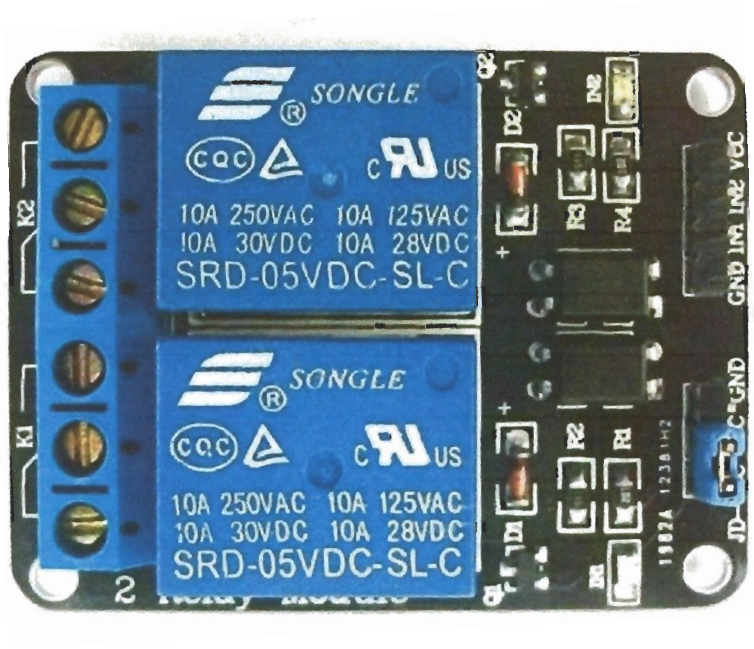
- 802.11 b/g/n
- Wi-Fi Direct (P2P),soft-AP
- Integrated TCP/IP protocolstack
- IntegratedTRswitch,balun,LNA,poweramplifierandmatchingnetwork
- IntegratedPLLs,regulators,DCXOandpowermanagementunits
- +19.5dBm output power in 802.11b mode
- Power down leakage current of<10uA
- 1MB Flash Memory
- Integratedlowpower32-bitCPUcouldbeusedasapplicationprocessor

- SDIO1.1/2.0,SPI,UART
- STBC,1×1MIMO,2×1MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)

Specification of ESP8266:

- Wi-Fi Direct (P2P),soft-AP
- Integrated TCP/IP protocolstack
- IntegratedTRswitch,balun,LNA,poweramplifierandmatchingnetwork
- IntegratedPLLs,regulators,DCXOandpowermanagement units
- 19.5dBm output power in 802.11b mode
- Power down leakage current of<10uA
- 1MB Flash Memory
- Integratedlowpower32-bitCPUcouldbeusedasapplicationprocessor
- Standby power consumption of< 1.0mW (DTIM3)

Relay Board:-



A relay [7] is an electrical device which is generally used to control high voltages using very low voltage as an Input. This consists of a coil wrapped around a pole and a two small metal flaps(nodes) that are used to close the circuit. One of the node is fixed and other is movable. Whenever an electricity is passed through the coil, it creates a magnetic field and attracts the moving node towards the static node and the circuit gets completed. So, just by applying small voltage to power up the coil we can actually complete the circuit for the high voltage [8] to travel. Also, as the static node is not

physically connected to the coil there is very less chance that the Microcontroller powering the coil gets damaged if something goes wrong.

This is Two Channel relay board controlled by computer USB port. The usb relay board is with 2 SPDT relays rated up to 10A each. You may control devices 220V / 120V (up to 2) [9] directly with one such relay unit. It is fully powered by the computer USB port. Suitable for home automation applications, hobby projects, industrial automation. The free software allows to control relays manually, create timers (weekly and calendar) and multivibrators, use date and time for alarms or control from command line.

Features:

- Power led: Yes
- Relay leds: Yes Highquality
- 4 SPDT Relay channels - selectable by user:
- JQC-3FC/T73DC5V(7A/250VAC,10A/125VAC,12A/120VAC,10A/ 28VDC)
- RAS-05-15 (10A/250VAC, 15A/120VAC, 15A/24VDC)
- PCB parameters: FR4 / 1.5mm / two layers / metalized holes/HAL/ white stamp / solder mask / extra PCB openings for better voltage isolation / doubled high voltagetracks
- Powersupply: fromUSBport
- Currentconsumption:400mA
- Chip: FT245RL
- Size: 77mm x 56mm x 17mm
- Supported by DRM software (Windows and Linux): Yes

- Supported by Denkovi Command line tool (Windows, Linux): Yes
- Android software available (low cost but very useful): Yes

Advantages:

- High quality
- Low cost
- No extra power supply
- Software with many functions
- Control electrical devices according weekday/date/time
- Create timers or pulses with our software

Applications:

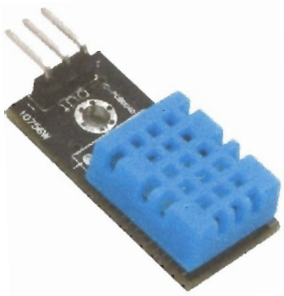
- Home automation
- Robotics
- Alarms
- Timers
- Open doors and windows via PC
- Aquariums applications

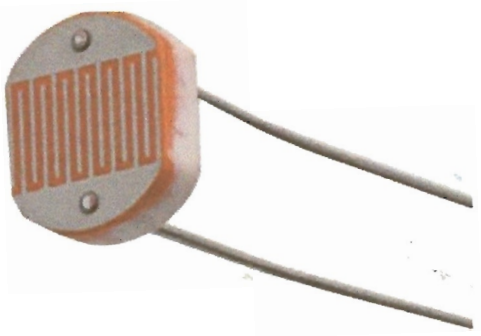
Additional Information:

This is relay board with 2 SPDT Relays controlled from USB port of your computer. The main purpose of this USB relay module is to help you building projects regarding robotics and home automation .You may control different electrical devices like home lights, DC motors, pneumatic cylinders, lasers and so

on. Each such board requires one USB port. The more USB ports you have the more such relay units you may connect and control. . The relay module outputs are controlled by FT245RL. It has 8 bit data output register (this device use only 2 of them). The USB relay card can not be controlled directly via COM port - you need to download our DRM Software to control the device. The usb relay unit can not work without PC. Only one such device can be supplyed from single USB port. If you want to supply many such devices you need USB HUB with extra power supply.

Sensors:

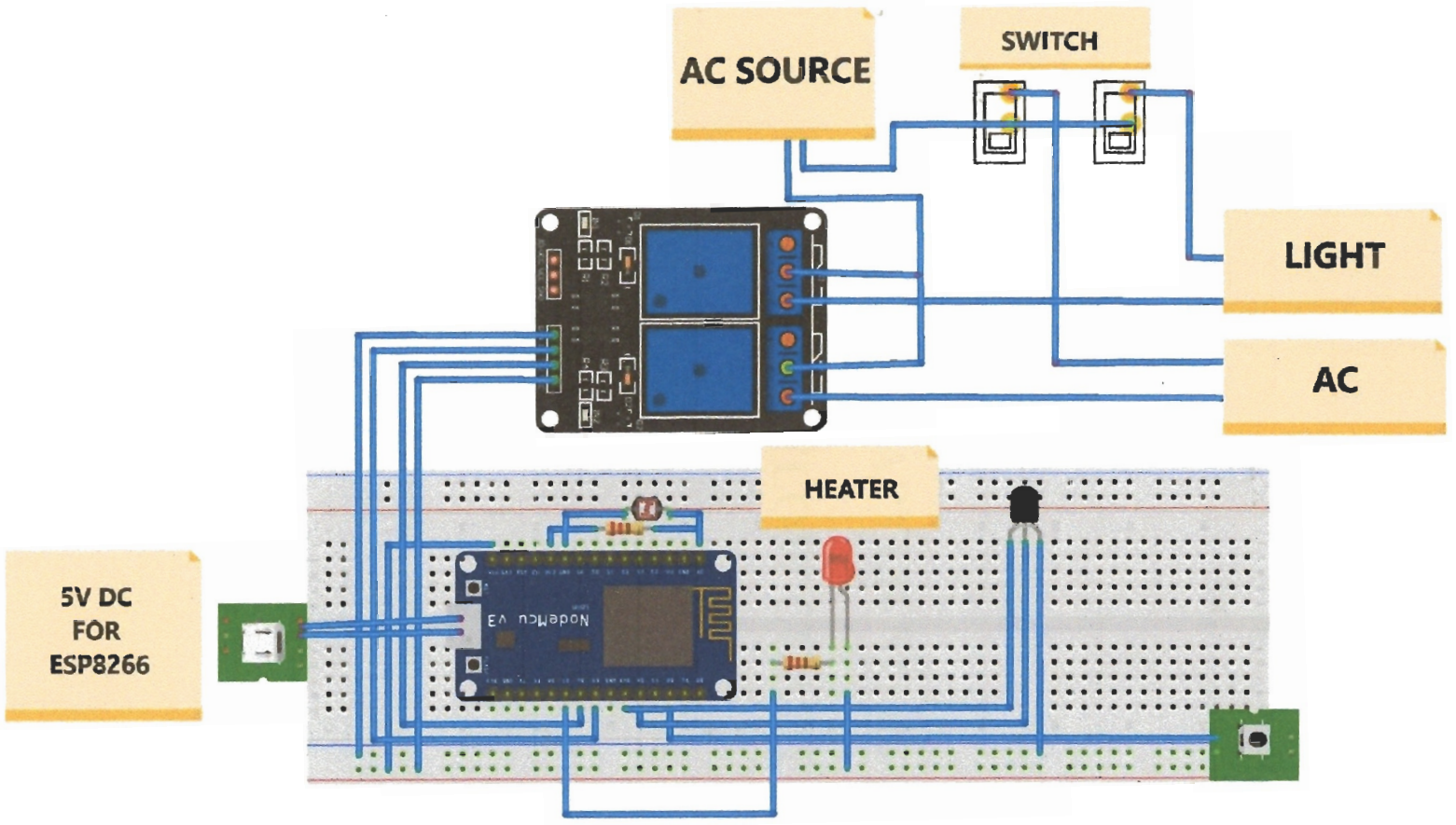




For this project two sensors are used. A light sensor [10] and a temperature sensor [11] is used. This sensors are directly connected to the esp8266 module. They continuously sync data to the module and the module control the states depending on the data from the sensors.

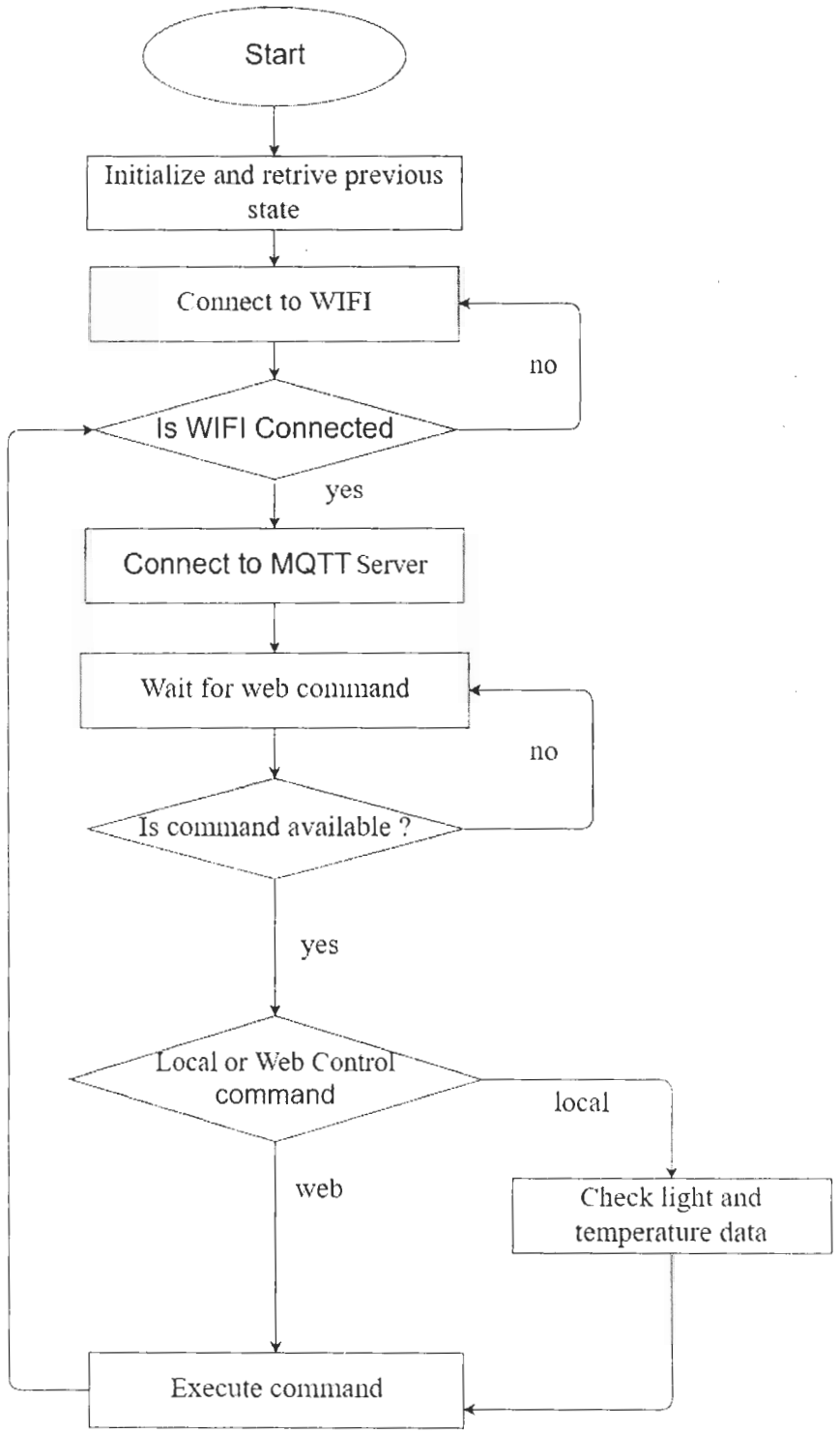
CHAPTER NO.06 (B)

CIRCUIT DIAGRAM


AC SOURCE
SWITCH
LIGHT
AC
HEATER
5V DC
FOR
ESP8266
NodeMcu V3

CHAPTER NO.06 (C)

FLOW CHART


Start
Initialize and retrive previous state
Connect to WIFI
no
Is WIFI Connected
yes
Connect to MQTT Server
Wait for web command
no
Is command available ?
yes
Local or Web Control command
local
web
Check light and temperature data
Execute command

CHAPTER NO.06 (D)

HARDWARE CODE

Arduino Code

```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
#include <DNSServer.h>
#include <ESP8266WebServer.h>
#include <WiFiManager.h>
#include "DHT.h"
#include <EEPROM.h>
#define light_sw D5
#define AC_sw D6
#define heater_sw D7
#define DHTPIN D4
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
/************************* Adafruit.io Setup
*********************************/
#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883                   // use 8883 for SSL
#define AIO_USERNAME    "Shehanrahmanrueee"
#define AIO_KEY         "d4346bf603064f6e82074063384ff55f"
WiFiClient client;
```

```
boolean stat_1, stat_2, stat_3, control, turn;

char* Message;

char* Message_1;

char* Message_2;

char* Message_3;

char data[50];

//const char* a = "hello" ;

float temp;

unsigned long lastTime;

unsigned long last_time;

#define interuptPin  D2

void ICACHE_RAM_ATTR sys_ovrd();

Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT,
AIO_USERNAME, AIO_KEY);

Adafruit_MQTT_Publish switchstatus = Adafruit_MQTT_Publish(&mqtt,
AIO_USERNAME "/feeds/switchstate");

Adafruit_MQTT_Subscribe autoManual = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME "/feeds/auto_manual");

Adafruit_MQTT_Subscribe onoffbutton_1 = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME "/feeds/onoff_1");

Adafruit_MQTT_Subscribe onoffbutton_2 = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME "/feeds/onoff_2");

Adafruit_MQTT_Subscribe onoffbutton_3 = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME "/feeds/onoff_3");


void MQTT_connect();
```

```
void setup() {

 pinMode(light_sw, OUTPUT);
 pinMode(AC_sw, OUTPUT);
 pinMode(heater_sw, OUTPUT);

 Serial.begin(115200);
 pinMode(interuptPin, INPUT_PULLUP);
 attachInterrupt( digitalPinToInterrupt(interuptPin), sys_ovrd, FALLING);

 delay(10);
 dht.begin();
 Serial.println(F("WIFI Switch"));

 WiFiManager wifiManager;
 wifiManager.autoConnect("WIFI_switch");
 Serial.println("WiFi connected");
 Serial.println("IP address: "); Serial.println(WiFi.localIP());
 mqtt.subscribe(&autoManual);
 mqtt.subscribe(&onoffbutton_1);
 mqtt.subscribe(&onoffbutton_2);
 mqtt.subscribe(&onoffbutton_3);
 EEPROM.begin(512);
```

```
 retrive_state();
}
void loop() {
 MQTT_connect();
 Adafruit_MQTT_Subscribe *subscription;
 while ((subscription = mqtt.readSubscription(5000))) {

  if (subscription == &autoManual) {
   Serial.print(F("control "));
   Serial.println((char *)autoManual.lastread);

   if (strcmp((char *)autoManual.lastread, "LOCAL") == 0) {
    control = true;

    EEPROM.put(3, 1);
    EEPROM.commit();
   }
   if (strcmp((char *)autoManual.lastread, "WEB") == 0) {
    control = false;
    EEPROM.put(3, 0);
    EEPROM.commit();
   }
  }
```

```
if (control == false) {

  if (subscription == &onoffbutton_1) {
   Serial.print(F("Got_1: "));
   Serial.println((char *)onoffbutton_1.lastread);

   if (strcmp((char *)onoffbutton_1.lastread, "ON") == 0) {
    digitalWrite(light_sw, LOW );
    stat_1 = true;

    EEPROM.put(0, 1);
    EEPROM.commit();
   }
   if (strcmp((char *)onoffbutton_1.lastread, "OFF") == 0) {
    digitalWrite(light_sw, HIGH );
    stat_1 = false;
    EEPROM.put(0, 0);
    EEPROM.commit();
   }
  }

  if (subscription == &onoffbutton_2) {
   Serial.print(F("Got_2: "));
   Serial.println((char *)onoffbutton_2.lastread);
```

```
  if (strcmp((char *)onoffbutton_2.lastread, "ON") == 0) {
   digitalWrite(AC_sw, LOW );
   stat_2 = true;
   EEPROM.put(1, 1);
   EEPROM.commit();
  }
  if (strcmp((char *)onoffbutton_2.lastread, "OFF") == 0) {
   digitalWrite(AC_sw, HIGH );
   stat_2 = false;
   EEPROM.put(1, 0);
   EEPROM.commit();
  }
 }
 if (subscription == &onoffbutton_3) {
  Serial.print(F("Got_3: "));
  Serial.println((char *)onoffbutton_3.lastread);

  if (strcmp((char *)onoffbutton_3.lastread, "ON") == 0) {
   digitalWrite(heater_sw, LOW );
   stat_3 = true;
   EEPROM.put(2, 1);
   EEPROM.commit();
  }
```

```
    if (strcmp((char *)onoffbutton_3.lastread, "OFF") == 0) {
      digitalWrite(heater_sw, HIGH );
      stat_3 = false;
      EEPROM.put(2, 0);
      EEPROM.commit();
    }
   }
  }
}

if (control == true)
{
 if (analogRead(A0) > 850)
 {
  digitalWrite(light_sw, LOW);
  stat_1 = true;
 }
 if (analogRead(A0) <= 850)
 {
  digitalWrite(light_sw, HIGH);
  stat_1 = false;
 }
 if (lastTime - millis() > 2000 )
 {
```

```
temp = dht.readTemperature();
lastTime = millis();
Serial.println(temp);

if (temp >= 26)
{
 digitalWrite(AC_sw, LOW);
 stat_2 = true;
 digitalWrite(heater_sw, HIGH);
 stat_3 = false;
}

else if (temp < 26 && temp > 20 )
{
 digitalWrite(AC_sw, HIGH);
 stat_2 = false;
 digitalWrite(heater_sw, HIGH);
 stat_3 = false;
}

else if (temp <= 20 )
{
 digitalWrite(AC_sw, HIGH);
 stat_2 = false;
```

```
      digitalWrite(heater_sw, LOW);
      stat_3 = true;
     }

   }
   Serial.println("Running in auto mode");
 }
 sendStatus();
 /*
   if(! mqtt.ping()) {
   mqtt.disconnect();
   }
 */
}

void sys_ovrd()
{

 if (millis() - last_time > 500)
 {
   control = !control;
   Serial.println("CHANGING CONTROL");
   if (control == false)
   {
```

```
    digitalWrite(light_sw, HIGH);
    digitalWrite(AC_sw, HIGH);
    digitalWrite(heater_sw, HIGH);
    stat_1 = false;
    stat_2 = false;
    stat_3 = false;
   }
  }
  last_time = millis();

}
void MQTT_connect() {
  int8_t ret;

  // Stop if already connected.
  if (mqtt.connected()) {
   return;
  }
  Serial.print("Connecting to MQTT... ");

  uint8_t retries = 3;
  while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
   Serial.println(mqtt.connectErrorString(ret));
   Serial.println("Retrying MQTT connection in 5 seconds...");
```

```
    mqtt.disconnect();
    delay(5000);  // wait 5 seconds
    retries--;
    if (retries == 0) {
     // basically die and wait for WDT to reset me
     while (1);
    }
   }
   Serial.println("MQTT Connected!");
}

void sendStatus()
{

  if (turn == true) {
    if (stat_1 == false && stat_2 == false && stat_3 == false)  //000
    {
      if (!switchstatus.publish("Light:OFF AC:OFF Heater:OFF")) {
        Serial.println(F("Failed"));
      } else {
        Serial.println(F("OK!"));
      }
    }
```

```
else if (stat_1 == false && stat_2 == false && stat_3 == true) //001
{
 if (!switchstatus.publish("Light:OFF AC:OFF Heater:ON")) {
  Serial.println(F("Failed"));
 } else {
  Serial.println(F("OK!"));
 }
}

else if (stat_1 == false && stat_2 == true && stat_3 == false)  //010
{
 if (!switchstatus.publish("Light:OFF AC:ON Heater:OFF")) {
  Serial.println(F("Failed"));
 } else {
  Serial.println(F("OK!"));
 }
}

else if (stat_1 == false && stat_2 == true && stat_3 == true)  //011
{
 if (!switchstatus.publish("Light:OFF AC:ON Heater:ON")) {
  Serial.println(F("Failed"));
 } else {
  Serial.println(F("OK!"));
```

```
  }
 }

 else if (stat_1 == true && stat_2 == false && stat_3 == false)  //100
 {
  if (!switchstatus.publish("Light:ON AC:OFF Heater:OFF")) {
   Serial.println(F("Failed"));
  } else {
   Serial.println(F("OK!"));
  }
 }

 else if (stat_1 == true && stat_2 == false && stat_3 == true)  //101
 {
  if (!switchstatus.publish("Light:ON AC:OFF Heater:ON")) {
   Serial.println(F("Failed"));
  } else {
   Serial.println(F("OK!"));
  }
 }

 else if (stat_1 == true && stat_2 == true && stat_3 == false)  //110
 {
  if (!switchstatus.publish("Light:ON AC:ON Heater:OFF")) {
```

```
    Serial.println(F("Failed"));
   } else {
    Serial.println(F("OK!"));
   }
  }

  else if (stat_1 == true && stat_2 == true && stat_3 == true)  //111
  {
   if (!switchstatus.publish("Light:ON AC:ON Heater:ON")) {
    Serial.println(F("Failed"));
   } else {
    Serial.println(F("OK!"));
   }
  }

 }
 if (turn == false) {
  temp = dht.readTemperature();
  if (control == false )
  {
   sprintf (data, "Control: Web, Temperature: %f ", temp);
  }
  if (control == true )
  {
```

```
    Serial.println(F("Failed"));
   } else {
    Serial.println(F("OK!"));
   }
  }

  else if (stat_1 == true && stat_2 == true && stat_3 == true)  //111
  {
   if (!switchstatus.publish("Light:ON AC:ON Heater:ON")) {
    Serial.println(F("Failed"));
   } else {
    Serial.println(F("OK!"));
   }
  }

 }
 if (turn == false) {
  temp = dht.readTemperature();
  if (control == false )
  {
   sprintf (data, "Control: Web, Temperature: %f ", temp);
  }
  if (control == true )
  {
```

```
    sprintf (data, "Control: Local, Temperature: %f ", temp);
   }
   switchstatus.publish(data);
  }

  turn = !turn;

}

void retrive_state()

{
  int tem;
  tem = EEPROM.read(0);
  if (tem == 1)
  {
   digitalWrite(light_sw, LOW );
   stat_1 = true;
  }
  else
  {
   digitalWrite(light_sw, HIGH );
   stat_1 = false;
  }
```

```
delay(100);


tem = EEPROM.read(1);
if (tem == 1)
{
 digitalWrite(AC_sw, LOW );
 stat_2 = true;
}
else
{
 digitalWrite(AC_sw, HIGH );
 stat_2 = false;
}
delay(100);
 tem = EEPROM.read(2);
if (tem == 1)
{
 digitalWrite(heater_sw, LOW );
 stat_3 = true;
}
else
{
 digitalWrite(heater_sw, HIGH );
 stat_3 = false;
```

```
  }
  delay(100);
  tem = EEPROM.read(3);
  if (tem == 1)
  {
    control = true;
  }
  else
  {
    control = false;
  }
}
```

CHAPTER NO.07

SOFTWARE

AND

WEB SERVER

ESP8266 FIRMWARE

Required hardware and software:

- You will need a Windows PC for this update
- You will need some form of USB to Serial converter that allows operation at 5V. I used a breadboard. It allows easy plugging, which then allows me to hookup the pins of the ESP8266 module via jumper wires to the corresponding pins on the USB<->Serial board.
- The firmware updating software only works on COM ports 1-6. If your USB<->Serial device enumerates to a higher port number than that, you will have to change it via Device Manager in Windows

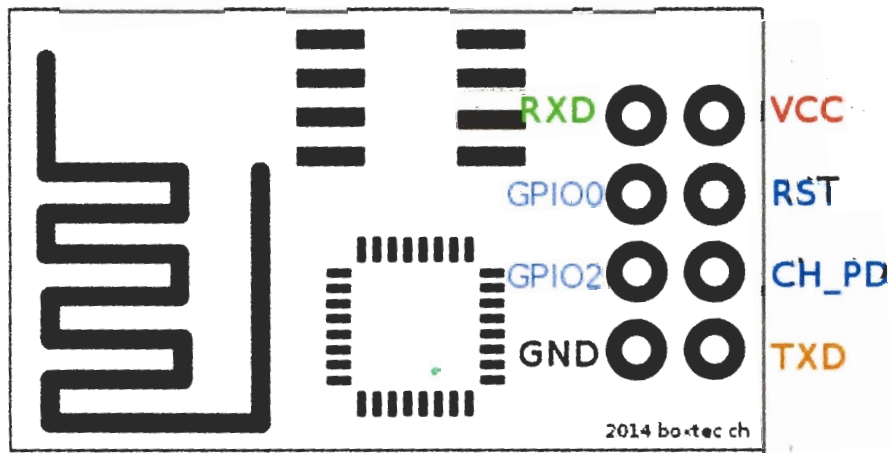


Fig: Pin out diagram for the ESP8266 Module.

You need to hook up these pins from the ESP8266 to your USB<->Serialboard:

- VCC to 5V
- GND to ground
- CH_PD to 5V
- TXD to RX,RXD to TX(this may depend on the USB<->Serialboard you are using. If it doesn't work, try swapping them around)
- GPIO0 to ground (for the duration of firmware upgrading.After all the upgrades have been loaded, it needs to be disconnected)

You will need to unplug and re-plug the USB cable 4 times during the process, so make sure you can reach it easily.

WEB SERVER

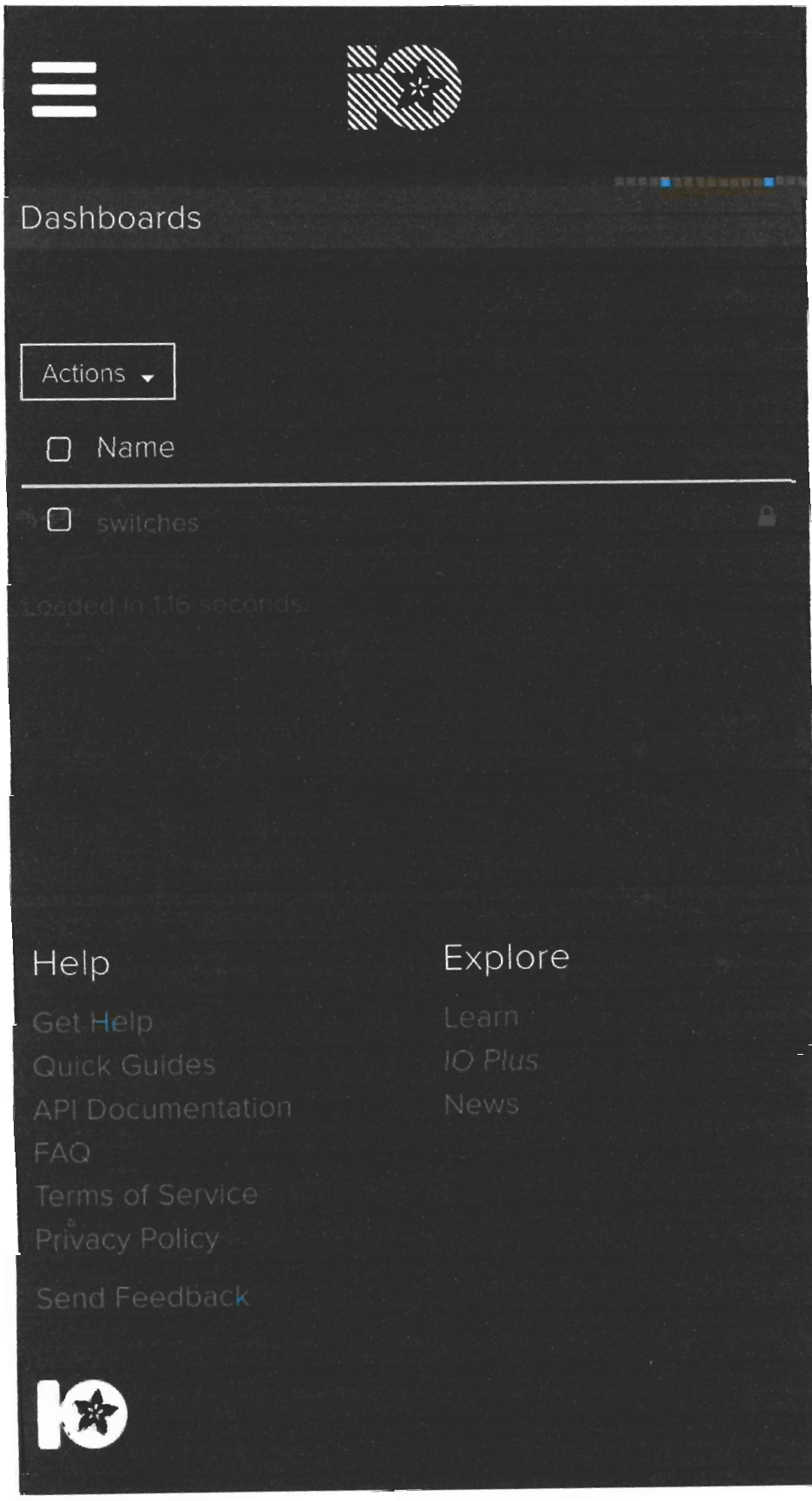
For this project I used a third party web server to control and monitor all the status of the automation.

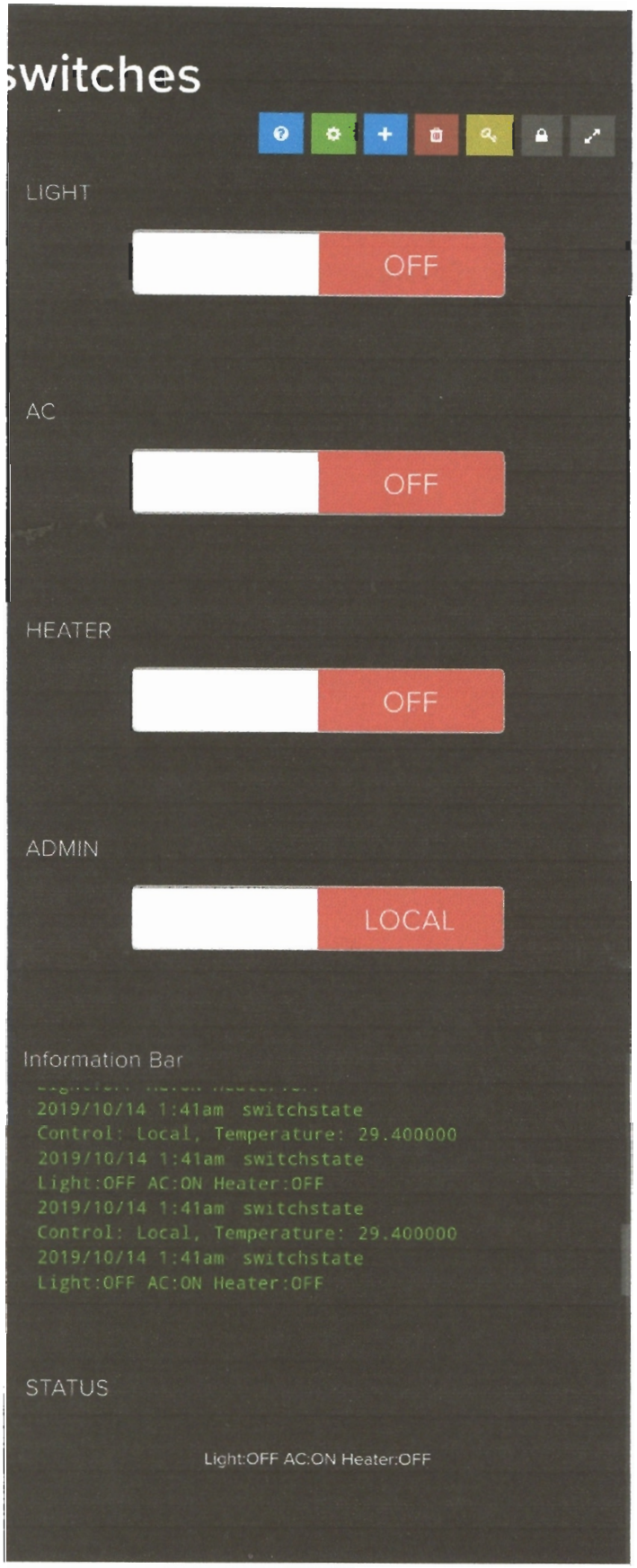
Server Address: io.adafruit.com

Here in the dashboard a folder named "SWITCHES" is formed. Insiede this the swichthes and the status bar and information bars are developed. There are mainly 4 switches.

- The web and local switch is for administration control of the automation system.
- The light switch is for the control of light on web mode
- The Ac switch is for the control of light on web mode
- The heater switch is for the control of light on web mode

The status bar is showing the status of the switches, and the information bar is displaying the data found from the ESP8266 Module.


Dashboards
Actions
Name
switches
Loaded in 1.16 seconds
Help
Get Help
Quick Guides
API Documentation
FAQ
Terms of Service
Privacy Policy
Send Feedback
Explore
Learn
IO Plus
News


switches
LIGHT
OFF
AC
OFF
HEATER
OFF
ADMIN
LOCAL
Information Bar
2019/10/14 1:41am switchstate
Control: Local, Temperature: 29.400000
2019/10/14 1:41am switchstate
Light:OFF AC:ON Heater:OFF
2019/10/14 1:41am switchstate
Control: Local, Temperature: 29.400000
2019/10/14 1:41am switchstate
Light:OFF AC:ON Heater:OFF
STATUS
Light:OFF AC:ON Heater:OFF

Description of Switches and Status bars

LIGHT switch:

The LIGHT switch controls the light load at web control mode and at local mode this switch is opened and the MCU control the light depending on sensors used.

AC switch:

The AC switch controls the AC load at web control mode and at local mode this switch is opened and the MCU control the switch depending on temprature sensors used.

HEATER switch:

The HEATER switch controls the HEATER load at web control mode and at local mode this switch is opened and the MCU control the switch depending on temprature sensors used.

ADMIN switch:

The ADMIN switch controls the administration control of the automation system. This operates in web mode or local mode. Where the web mode is the web control stage and the local mode is the automation stage.

Information Bar:

The information shows

- Admin mode
- The switch status
- Temperature
- Previous switch status
- Previous logs of temperature
- Real time changes of temperature and switches

STATUS Bar:

The status bar indicates the real time status of the load switches whether the switches are on ON or OFF state.

CHAPTER NO.08

CONCLUSION

CONCLUSION

In conclusion, my device works as expected. I am expecting to add more and diverse features to take "automated" to "smart" homes. Going by the trend, this is going to be the future of clean and smart living. These devices can run 24x7, 365 days and with a little tweak, I can also monitor the real time status of home anytime from anywhere and control home appliance . Indirectly, this can also be used as anti-power consumption devices to monitor the home remotely.

Limitations of the Work

Web server is a complicated part for this project. So I used a third party server and for this desired switches are not found in web control but the web control feature was not compromised.

Web security for the system is compromised a little for this project where as this project is easy to control and fully safe and environment friendly.

Future work

In my project the monitoring system can be further enhanced by using GSM alart system, which supports another real-time application and safety feature. For industrial purposes this can be developed for automation of Lab, Sensitive Storage or etc . In future all the devices that we used everyday that will connected to a microcontroller and we can use and control them wirelessly not only remotely but also using upcoming technologies. In future use of advanced alarm process and more upgraded automation can be more helpful for using.

CHAPTER NO.09

REFRENCES

&

BIBLIOGRAPHY

REFRENCES & BIBLIOGRAPHY :

REFRENCES

[1] C. Gomez and J. Paradells, "Wireless home automation networks: A survey of architectures and technologies," *IEEE Commun. Mag.*, 2010.

[2] Y. A. Badamasi, "The working principle of an Arduino," in *Proceedings of the 11th International Conference on Electronics, Computer and Computation, ICECCO 2014*, 2014.

[3] M. Khan, B. N. Silva, and K. Han, "Internet of Things Based Energy Aware Smart Home Control System," *IEEE Access*, 2016.

[4] B. L. Risteska Stojkoska and K. V. Trivodaliev, "A review of Internet of Things for smart home: Challenges and solutions," *Journal of Cleaner Production*. 2017.

[5] EINSTRONIC, "Nodemcu," *Einstronic*, 2017.

[6] S. Tonage, S. Yemul, R. Jare, and V. Patki, "IoT based home automation system using NodeMCU ESP8266 module," *Int. J. Adv. Res. Dev.*, 2018.

[7] H. C. Roberts, B. G. Lipták, and W. P. Durden, "Relays," in *Instrument Engineers Handbook, Fourth Edition: Process Control and Optimization*, 2005.

[8] S. G. A. Perez, T. S. Sidhu, and M. S. Sachdev, "Modeling relays for power system protection studies," *Electr. Eng.*, 2006.

[9] C. R. Bayliss and B. J. Hardy, "Relay Protection," in *Transmission and Distribution Electrical Engineering*, 2012.

[10] E. Efficiency, E. Circuit, L. Dependent, and R. Circuits, "Light dependent resistor (ldr)," *Energy*, 2010.

[11] W. S. Jaroszynski, "TEMPERATURE MEASUREMENT.," *Test Meas. World*, 1983.

Websites:

- https://en.wikipedia.org/wiki/Relay
- https://www.goodsky.com.tw/en/relays/general-purpose-relay
- https://en.wikipedia.org/wiki/NodeMCU
- https://nodemcu.readthedocs.io/en/master/
- https://docs.zerynth.com/latest/official/board.zerynth.nodemcu3/docs/index.html
- https://www.academia.edu/
- https://en.wikipedia.org/wiki/Main_Page
- https://iot-playground.com/
- https://www.watelectrical.com
- https://www.kitronik.co.uk/blog
- https://www.arduino.cc/
- https://ocw.mit.edu/index.htm

Journals & other books:

- Kusuma S M,Assistant Professor,Department of telecommunication, MSRIT, Bangalore, India."Home Automation Using Internet of Things."
- Niharika Shrotriya,Anjali Kulkarni, Priti Gadhave, International Journal of Science,Engineering and Technology Research (IJSETR), "SMART HOME USING WI-FI"

- Anushri Aware, SonaliVaidya,PriyankaAshture, VarshaGaiwal PES' s Modern College of Engineering, Pune-04, International Journal of Engineering Research and General Science Volume 3, " Home Automation using Cloud Network"