IOT BASED VOICE AND REMOTE-CONTROLLED SMART HOME AUTOMATION

A PROJECT REPORT

Submitted by

K. Karpaka Priya

K. Mari

&

P.K. Muthumari

BACHELOR OF ENGINEERING

IN ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MOTHER THERESA ENGINEERING COLLEGE

VAGAIKULAM, THOOTHUKUDI-628102

CONTENT

TITLE	PAGE NO
ABSTRACT	3
INTRODUCTION	4
OBJECTIVE	5
PROBLEM STATEMENT	5
PROPOSED SOLUTION	6
REASON FOR SCALABILITY INNOVATION	6
IMPLEMENTATION	7
CIRCUIT DIAGRAM	7
MODEL IMAGE	8
BLOCK DIAGRAM	8
WORKING	9
COMPONENTS	10
SOFTWARE DESCRIPTION	13
SOFTWARE OVERVIEW	14
PROGRAMMING OVERVIEW	16
SORCE CODE	22
SCOPE FOR SOLUTION	29
APPLICATION	29
MERITS	30
CONCLUSION	30
REFERENCE	31

ABSTRACT:

With an advancement of Automation technology, life is obtaining easier and easier altogether aspects day by day. In today's world, Automatic systems are being most popular over manual system. With the fast increase with in the variety of users of web over the past decade has created web vicinity and parcel of life, and IOT is that the late stand rising web technology. Here in the model of our project, we have designed a wireless control of home appliances using mobile phones. It can be more useful to the handicap who finds difficulty in controlling home appliances by manual mode. And in addition, we have inserted voice command control system. With a constant development in technology, voice command systems such as Amazon Alexa and Apple's Siri is becoming a more natural part of standard living. This project aims to implement a Voice controlled home automation based on IOT. Which is being remotely controlled and Monitored by any Android OS smart phones. It can be accessed through the Android's google assistant feature or Alexa application, which can be more useful, helpful and easier to those who struggle to do their day-to-day activities. This method is intended to be low priced and expandable permitting a range of devices to be controlled. We have designed this project in order to save electrical energy upto a certain limit instead of wasting it.

INTRODUCTION:

Home automation is constructing automation for a domestic, mentioned as a sensible home or smart house. In the IoT home automation ecosystem, you can control your devices like light, fan, TV, etc. A domestic automation system can monitor and/or manage home attributes adore lighting system, and appliances. It is very helpful to control your home devices. Once it coupled with the internet a very important constituent of the Internet of Things. The program for control of the system makes use of both wall-mounted terminals, tablet or desktop computers, a smartphone application, or an online interface that may even be approachable off-site through the Internet. One of the key benefits of IoT-enabled home automation is the ability to control and monitor a wide range of devices and systems from a single, centralized location. This can include everything from lighting and temperature control to security cameras and alarm systems. This can be useful for controlling energy consumption and ensuring the safety and security of the home. In addition, IoT-enabled home automation systems can integrate with other smart home technologies, such as voice assistants like Alexa and Google Home, to provide additional functionality and convenience. Overall, IoT-enabled home automation can provide many benefits to homeowners, including increased convenience, energy efficiency, and security.

OBJECTIVE:

This project is designed to use the voice recognition technology to control home appliances. Eg: light. Its implementation mainly focused on the disabled people. Google Assistant and Alexa will be used as a voice input. Sinric Pro is a website where we can create a distinct keys for different electrical appliances. Software program is responsible for the key which is created in sinric pro to activate. Application software will be responsible for converting the voice command into the text format to the NODE MCU. This node mcu is intelligent enough to convert the text command executable according to the programming that is involved as per the requirements. Commands which are used are light ON/OFF. The objective is to use this IoT technology for the peoples of all ages , genders and even with disabled people to enjoy their life in a ease and pleasant way.

PROBLEM STATEMENT:

Today's generation, people are looking for ways and to make better life-style by using the technologies that are available. Conventional wall switches are located in different parts of a house and thus necessitates manual operations like to switch on or off to control the various home appliances. Switch controlling system in home appliances is not useful for current modern life style and also for disabled people. It gets virtually impossible to keep track of appliances that they are running and also to monitor their performances.

PROPOSED SYSTEM:

Internet has become a common interface to connect many people. IoT plays a vital role in the smart city applications. Currently IoT is an emerging technology spread globally. IoT connects all the hardware components to Internet which can be controlled from everywhere in the universe. A smart home allows homeowners to control appliances, thermostats, lights, and other devices remotely using a smart phone or tablet through an internet connection. Smart homes can be set up through wireless or hardwired systems. Smart home technology provides homeowners with convenience and cost savings.

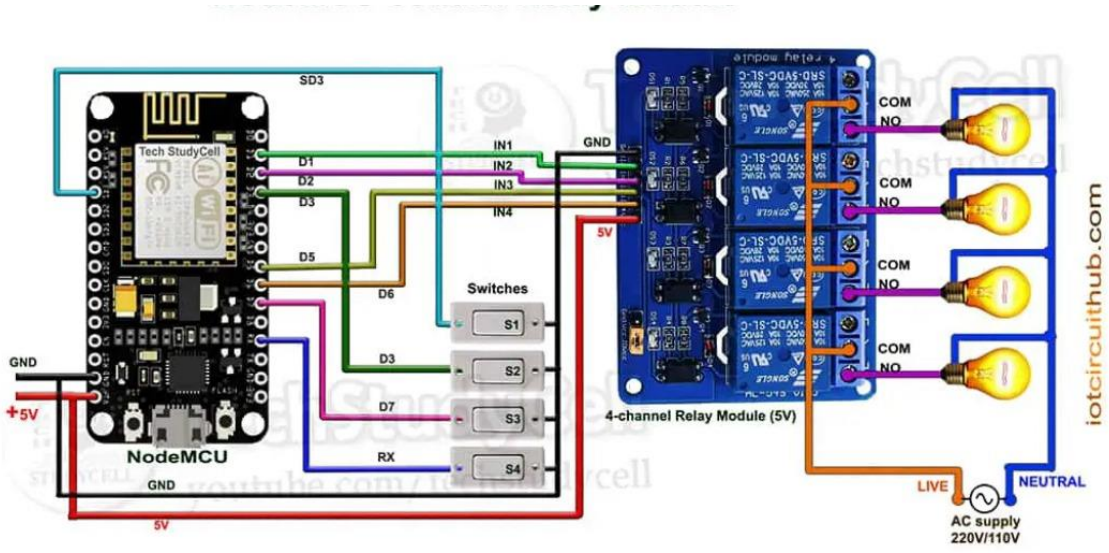
REASON FOR SCALABILITY INNOVATION:

Home Automation Systems have been continuously evolving in the past few years. Some of the systems are focusing their attention on providing remote access and control of the system. Some others are more focused on the interconnectivity of the devices and on integrating them into greater systems, like smart cities. In this Innovative Automation, we see change as an opportunity to grow and improve. This philosophy can be seen in our high-quality automation systems. Using this type of automation innovation will ensure that the system can grow and change with our life and that it will be compatible with tomorrow’s technology

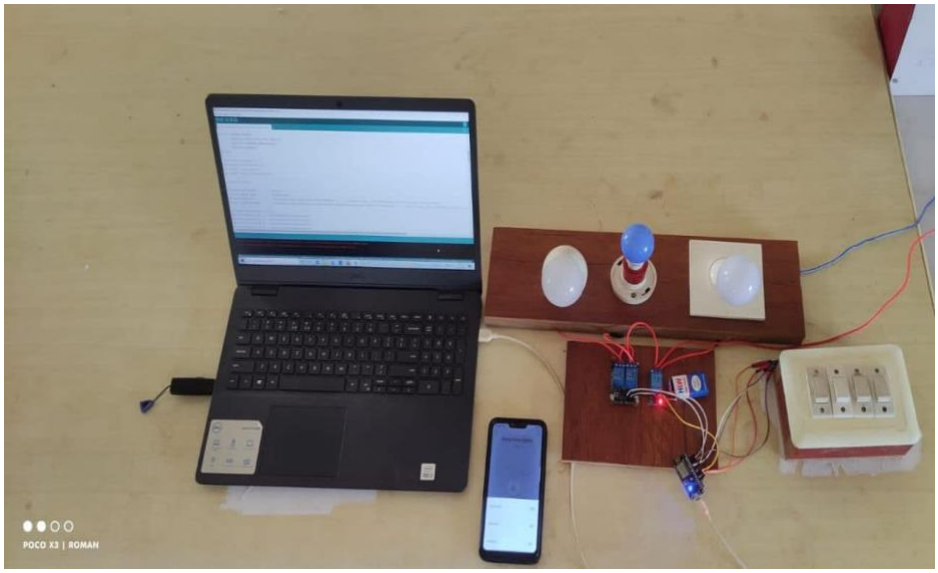
IMPLEMENTATION:

The problem we came across is to monitor and control the home appliances remotely. So we have designed a circuit which is more easier to monitor and control the appliances remotely using the concept of IoT. Our project is easier to use, we can control the appliances using our mobile phones. Nowadays every android mobiles can access google assistant, so we decided to make the control of the appliances using that feature. Using google assistant by simply saying "Hey google turn ON or OFF kitchen light" we can turn it ON and OFF. Suppose we are out of station and we are in doubt that did we turn off the appliances, we can turn it OFF using google home appliances.

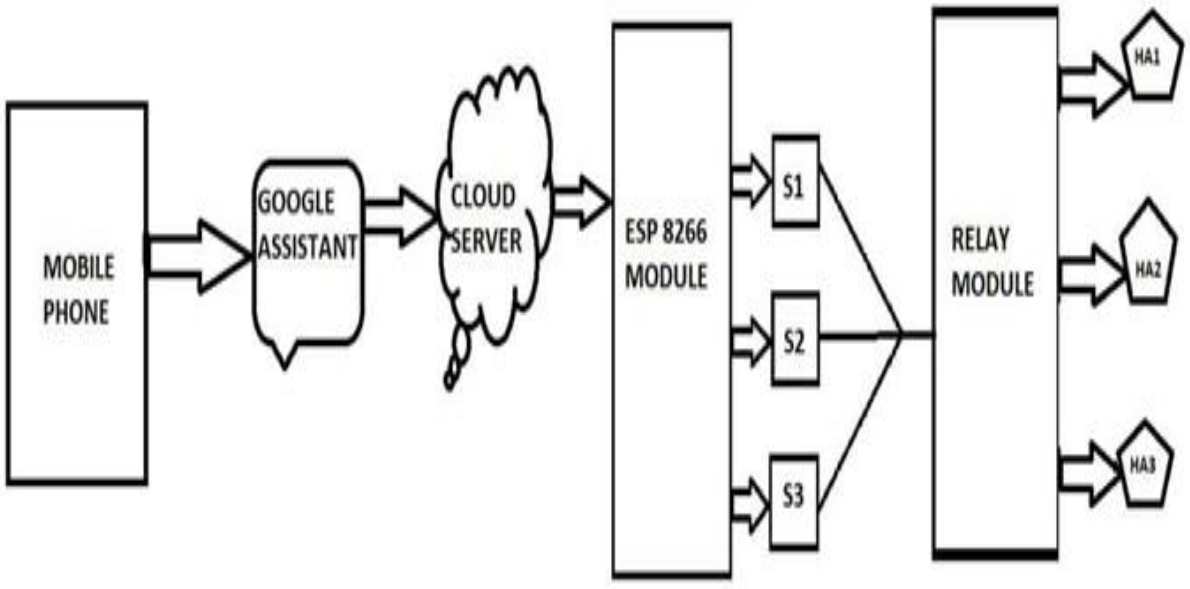
CIRCUIT DIAGRAM:



MODEL IMAGE:



BLOCK DIAGRAM:



WORKING :

Four main components used:

- ESP 8266 MODULE
- RELAY MODULE
- SINRIC PRO WEBSITE
- GOOGLE HOME APPLICATION

In sinric pro website , we create a distinct key for the home appliances. The keys are controlled by Google home through software program which is dumped inside the ESP 8266 module. The output is generated for distinct key and it is transferred to the relay module. When the Google home's button turns ON , it turns on the home appliances, and when the Google home's button turns OFF, it turns OFF the home appliances.

COMPONENTS:

1.ESP 8266:

ESP 8266 is a chip which is a highly integrated Wi-Fi SOC solution where in the IoT industry. It can act either as the slave to a host MCU or as a standalone application. It has an ESP8266, plus a separate processor for Espruino with loads of RAM and CPU power. The ESP8285 is an ESP8266 with 1 MiB of built-in flash , allowing for single-chip devices capable of connecting to Wi-Fi. It is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to wi-Fi network.

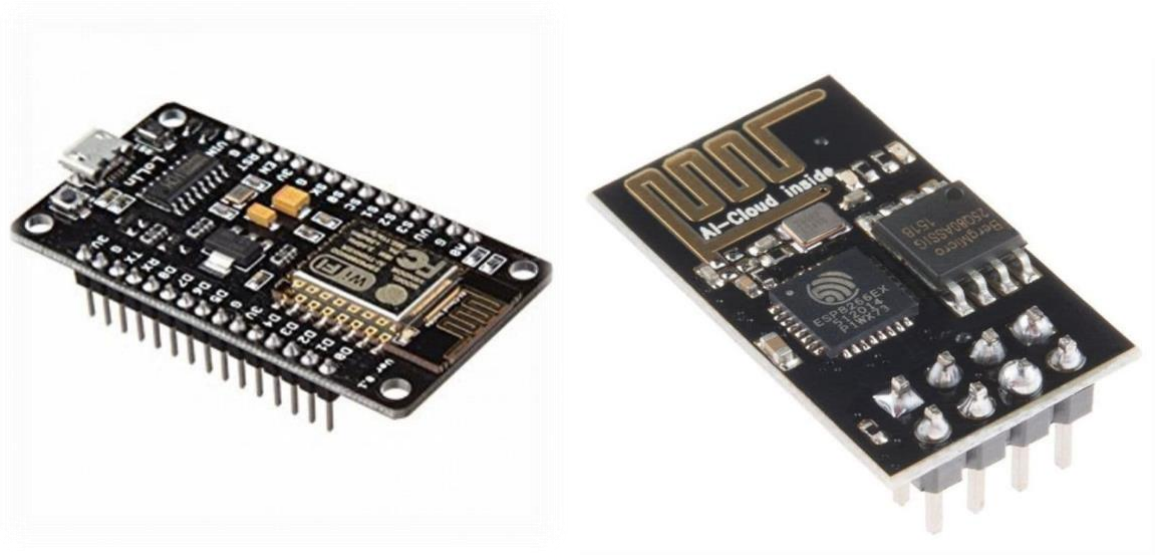


Fig:ESP8266 MODULE

2.RELAY MODULE:

Relays are used where it is necessary to control a circuit by a separate low-power signal , or where several circuits must be controlled by one signal. It is an electrically operated switch that can be turned on or off deciding to let current flow through or not. They are designed to be controlled with low voltages like 3.3 v or 5v. They commonly use an electromagnet to operate their internal mechanical switching mechanism.When a relay contact is open, this will switch power ON for a circuit when the coil is activated.

Fig :Relay module

Fig :Relay module

SMART PHONE:

An android phone is a powerful , high-tech smart phone that turns on the android operating system (OS) developed by Google. Any type of Smart Android Phone provide the Wi-Fi to the respected ESP 8266 MODULE. GOOGLE ASSISTANT in the android phone application is responsible for the input voice command . SYNRIC PRO enables developers to integrate IoT development boards with third-party applications or with ALEXA and GOOGLE HOME. GOOGLE HOME and ALEXA is an app which can be downloaded easily in any smart phones.

SOFTWARE DESCRIPTION:

ARDUINO IDE:

The ARDUINO Integrated Development Environment or arduino Software (IDE) - contains a text editor for writing code, a message area, A text console, a toolbar with buttons for common functions and a series of menus. Fig 4.13 shows the ARDUINO IDE. It connects to the ARDUINO and Genuino hardware to upload programs and communicate with them.

SOFTWARE OVERVIEW:

Programs written using ARDUINO Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension ARDUINO. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays. The console displays text output by the ARDUINO Software (IDE), including complete error messages and other information.

The bottom right hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor. Versions of the Software (IDE) prior to 1.0 saved sketches with the extension. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the into extension on save.

The following are the basic menu elements present in the ARDUINO IDE Software for coding, compiling and uploading a sketch on to an ARDUINO. Each menu has a different icon for easy identification. This icons are shown below along with there operations.

 **Verify**

Check your code for errors compiling it.

 Upload

Compiles your code and uploads it to the configured board. See uploading below for details.

 New

Create a new sketch

Open

Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

 Save

Save your Sketch

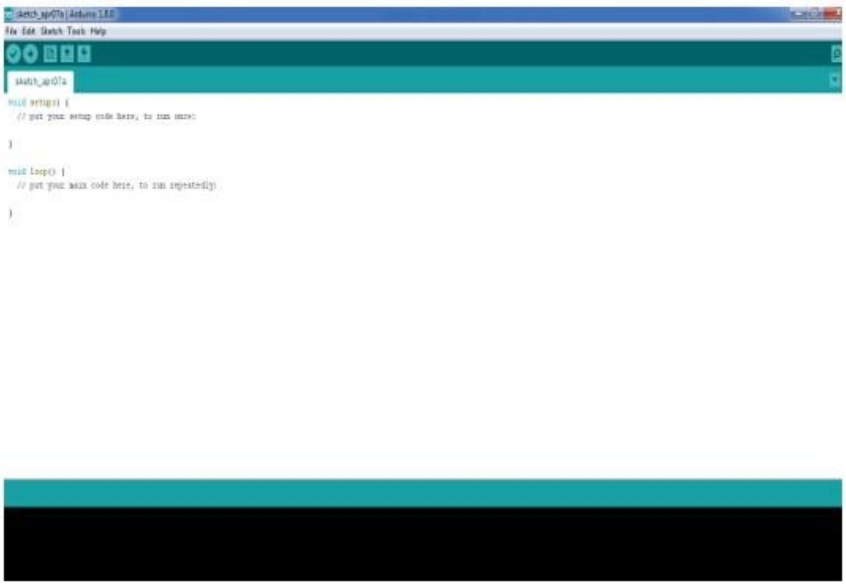
PROGRAMMING OVERVIEW :

When you open the ARDUINO program, you are opening the IDE. It is intentionally streamlined to keep the file is called a sketch – a sketch is where you save the computer code you have written things as simple and straight forward as possible. When you save a file in ARDUINO. The coding language that ARDUINO uses is very much like C Programming which is a common language in the world of computing. The code you learn to write for ARDUINO will be very similar to the programming languages. Code you write in any other computer language – all the basic concepts remain the same – it is just a matter of learning a new dialect should you pursue other programming languages.

The code you write is "human readable", that is, it will make sense to you (sometimes), and will be organized for a human to follow. Part of the job of the IDE is to take the human readable code and translate it into machine-readable code to be executed by the ARDUINO. This process is called compiling. The process of compiling is seamless to the user. All you have to do is press a button.

If you have errors in your computer code, the compiler will display an error message at the bottom of the IDE and highlight the

line of code that seems to be the issue. The error message is meant to help you identify what you might have done wrong – sometimes the message is very explicit, like saying, “Hey – you forget a semicolon”, sometimes the error message is vague. Why be concerned with a semicolon you ask A semicolon is part of the ARDUINO language syntax, the rules that govern how the code is written. It is like grammar in writing. Say for example we didn’t use periods when we wrote – everyone would have a heck of a time trying to figure out when sentences started and ended. Or if we didn’t employ the comma how would we convey a dramatic pause to the reader And let me tell you, if you ever had an English teacher with an overactive red pen, the compiler is ten times worse.



ANDROID:

Android is a mobile operating system based on a modified version of the Linux kernel and other open-source software, designed primarily for touch screen mobile devices such as smart phones and tablets. Android is developed by a consortium of developers known as the Open Handset Alliance, with the main contributor and commercial marketer being Google. Initially developed by Android Inc, which Google bought in 2005, Android was unveiled in 2007, with the first commercial Android device launched in September 2008. The current stable version is Android 10, released on September 3, 2019. The core Android source code is known as Android Open-Source Project (AOSP), which is primarily licensed under the Apache License. This has allowed variants of Android to be developed on a range of other electronics, such as game consoles, digital cameras, PCs and others, each with a specialized user interface. Some well-known derivatives include Android TV for televisions and Wear OS for wearables, both developed by Google. Android's source code has been used as the basis of different ecosystems, most notably that of Google which is associated with a suite of proprietary software called. Google Mobile Services (GMS) that frequently comes pre-installed on said devices. This includes core apps such as Gmail, the digital distribution platform Google Play and associated Google Play Services development platform, and usually apps such as the Google Chrome web browser. These apps are licensed by manufacturers of

Android devices certified under standards imposed by Google. Other competing Android ecosystems include Amazon com, Fire OS, or Lineage OS. Software distribution is generally offered through proprietary application stores like Google Play Store or Samsung Galaxy Store, or open-source platforms like Aptoide or F-Droid, which use software packages in the APK format. Android has been the best-selling OS worldwide on smart phones since 2011 and on tablets since 2013.As of May 2017, it has over two billion monthly active users, the largest installed base of any operating system, and as of January 2020, the Google Play Store features over 2.9 million apps.

GOOGLE HOME APPLICATION:

Google Home refers to two things: the original Google Home smart speaker, and the entire product line, including Google Home Hub, Google Mini, and other products.

The original Google Home device is essentially a single two-inch speaker and some computer hardware packaged up in a housing that looks something like an air freshener. It has Wi-Fi connectivity built right in, which it uses to access your Wi-Fi network and connect to the internet.

Google Home was originally designed to compete with the Amazon Echo. It has similar capabilities and functionality, but it's

built around Google Assistant instead of Amazon's Alexa virtual assistant.

In addition to the original Google Home smart speaker, Google has created a variety of other devices in the Google Home line that provide access to Google Assistant.

GOOGLE HOME IN SMART HOME:

If you're sold on the whole concept of talking to your virtual assistant, control your entire smart home through Google Home with voice commands. With Google Home as the centerpiece of your smart home, use voice commands to turn your lights on and off, control your television and other smart electronics, adjust your thermostat, and more.

Some smart home devices work natively with Google Home, and others require some kind of hub to act as a bridge. Check out our guide to what works with Google Home for more information.

SOURCE CODE:

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include "SinricPro.h"
#include "SinricProSwitch.h"

#include <map>

#define WIFI_SSID         "V2036"
#define WIFI_PASS         "sindhupriya"
#define APP_KEY           "8b8c8bc9-b958-4113-9085-0259c440d3c2"
// Should look like "de0bxxxx-1x3x-4x3x-ax2x-5dabxxxxxxxx"
#define APP_SECRET        "85da558a-f5b8-4992-aa0b-
21abb2938d47-dea1d2b3-ce20-4aa2-bd3d-37427170dd27"
//Enter the device IDs here
#define device_ID_1   "65115226ec78415303a5077d"
#define device_ID_2   "6511529aec78415303a5087c"
#define device_ID_3   "6511527aec78415303a507df"
#define device_ID_4   "xxxxxxxxxxxxxxxxxxxxxxxx"

// define the GPIO connected with Relays and switches
#define RelayPin1 5  //D1
#define RelayPin2 4  //D2
#define RelayPin3 14 //D5
```

```
#define RelayPin4 12 //D6

#define SwitchPin1 10  //SD3
#define SwitchPin2 0   //D3
#define SwitchPin3 13  //D7
#define SwitchPin4 3   //RX

#define wifiLed   16   //D0

// comment the following line if you use a toggle switches instead of tactile buttons
//#define TACTILE_BUTTON 1

#define BAUD_RATE   9600

#define DEBOUNCE_TIME 250

typedef struct {      // struct for the std::map below
  int relayPIN;
  int flipSwitchPIN;
} deviceConfig_t;


// this is the main configuration
```

```
// please put in your deviceId, the PIN for Relay and PIN for flipSwitch
// this can be up to N devices...depending on how much pin's available on your device ;)
// right now we have 4 devicesIds going to 4 relays and 4 flip switches to switch the relay manually
std::map<String, deviceConfig_t> devices = {
  //{deviceId, {relayPIN,  flipSwitchPIN}}
  {device_ID_1, {  RelayPin1, SwitchPin1 }},
  {device_ID_2, {  RelayPin2, SwitchPin2 }},
  {device_ID_3, {  RelayPin3, SwitchPin3 }},
  {device_ID_4, {  RelayPin4, SwitchPin4 }}
};

typedef struct {      // struct for the std::map below
  String deviceId;
  bool lastFlipSwitchState;
  unsigned long lastFlipSwitchChange;
} flipSwitchConfig_t;

std::map<int, flipSwitchConfig_t> flipSwitches;    // this map is used to map flipSwitch PINs to deviceId and handling debounce and last flipSwitch state checks
// it will be setup in "setupFlipSwitches" function, using informations from devices map

void setupRelays() {
```

```
  for (auto &device : devices) {           // for each device (relay, flipSwitch combination)
    int relayPIN = device.second.relayPIN; // get the relay pin
    pinMode(relayPIN, OUTPUT);             // set relay pin to OUTPUT
    digitalWrite(relayPIN, HIGH);
  }
}

void setupFlipSwitches() {
  for (auto &device : devices)  {                  // for each device (relay / flipSwitch combination)
    flipSwitchConfig_t flipSwitchConfig;           // create a new flipSwitch configuration

    flipSwitchConfig.deviceId = device.first;      // set the deviceId
    flipSwitchConfig.lastFlipSwitchChange = 0;     // set debounce time
    flipSwitchConfig.lastFlipSwitchState = true;   // set lastFlipSwitchState to false (LOW)--

    int flipSwitchPIN = device.second.flipSwitchPIN;  // get the flipSwitchPIN

    flipSwitches[flipSwitchPIN] = flipSwitchConfig;   // save the flipSwitch config to flipSwitches map
    pinMode(flipSwitchPIN, INPUT_PULLUP);             // set the flipSwitch pin to INPUT
```

```
  }
}

bool onPowerState(String deviceId, bool &state)
{
  Serial.printf("%s: %s\r\n", deviceId.c_str(), state ? "on" : "off");
  int relayPIN = devices[deviceId].relayPIN; // get the relay pin for corresponding device
  digitalWrite(relayPIN, !state);             // set the new relay state
  return true;
}

void handleFlipSwitches() {
  unsigned long actualMillis = millis();                                    // get actual millis
  for (auto &flipSwitch : flipSwitches) {                                    // for each flipSwitch in flipSwitches map
    unsigned long lastFlipSwitchChange = flipSwitch.second.lastFlipSwitchChange;  // get the timestamp when flipSwitch was pressed last time (used to debounce / limit events)

    if (actualMillis - lastFlipSwitchChange > DEBOUNCE_TIME) {  // if time is > debounce time...

      int flipSwitchPIN = flipSwitch.first;                                 // get the flipSwitch pin from configuration
```

```
    bool lastFlipSwitchState = flipSwitch.second.lastFlipSwitchState;
// get the lastFlipSwitchState

    bool flipSwitchState = digitalRead(flipSwitchPIN);
// read the current flipSwitch state

    if (flipSwitchState != lastFlipSwitchState) {                    //
if the flipSwitchState has changed...

#ifdef TACTILE_BUTTON

      if (flipSwitchState) {                                        // if the
tactile button is pressed

#endif

        flipSwitch.second.lastFlipSwitchChange = actualMillis;
// update lastFlipSwitchChange time

        String deviceId = flipSwitch.second.deviceId;                //
get the deviceId from config

        int relayPIN = devices[deviceId].relayPIN;                   //
get the relayPIN from config

        bool newRelayState = !digitalRead(relayPIN);
// set the new relay State

        digitalWrite(relayPIN, newRelayState);                       //
set the trelay to the new state


        SinricProSwitch &mySwitch = SinricPro[deviceId];
// get Switch device from SinricPro

        mySwitch.sendPowerStateEvent(!newRelayState);
// send the event

#ifdef TACTILE_BUTTON

      }

#endif
```

```
        flipSwitch.second.lastFlipSwitchState = flipSwitchState;
// update lastFlipSwitchState
      }
    }
  }
}

void setupWiFi()
{
  Serial.printf("\r\n[Wifi]: Connecting");
  WiFi.begin(WIFI_SSID, WIFI_PASS);

  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.printf(".");
    delay(250);
  }
  digitalWrite(wifiLed, LOW);
  Serial.printf("connected!\r\n[WiFi]: IP-Address is %s\r\n",
WiFi.localIP().toString().c_str());
}

void setupSinricPro()
{
  for (auto &device : devices)
```

```
  {
    const char *deviceId = device.first.c_str();
    SinricProSwitch &mySwitch = SinricPro[deviceId];
    mySwitch.onPowerState(onPowerState);
  }

  SinricPro.begin(APP_KEY, APP_SECRET);
  SinricPro.restoreDeviceStates(true);
}

void setup()
{
  Serial.begin(BAUD_RATE);
  pinMode(wifiLed, OUTPUT);
  digitalWrite(wifiLed, HIGH);
  setupRelays();
  setupFlipSwitches();
  setupWiFi();
  setupSinricPro();
}
void loop()
{
  SinricPro.handle();
  handleFlipSwitches();
}
```

SCOPE FOR THE SOLUTION:

Home Automation System (HAS) with low cost and wireless remote control module is designed. Internet of Things (IoT) is also implemented along with HAS for controlling the home appliances via World Wide Web. Home Automation System is **a space for the digital natives**. With the invention of lots of automation technologies featuring IOT and AI, home automation has become a reality. One can implement several of their tasks with just a single command of verbal instructions. We can make our smart home device work according to our day to day life works and habits.

APPLICATION:

- Home automation has been projected to target wide array applications for the new digital consumer.
- Some of the areas expected home automation led to IoT-enabled connectivity are:
- Lighting control
- Improved home safety and security
- HVAC
- Lawn/gardening management
- Natural language-based voice assistants

- Smart switches, locks energy meters
- Better infotainment delivery
- AI-driven digital experiences
- Home air and water quality monitoring

MERITS:

Managing all of your home devices from one place.

Flexibility for new devices and appliances.

Maximizing home security.

Remote control of home functions.

Increased energy efficiency.

Improved appliance functionality.

Home management insights.

CONCLUSION:

This project gives the basic knowledge of IOT, where all the appliances are controlled and can also update the status of the device. Here home automation is carried out using internet of things where in ESP 8266 Module plays an important role by send the order

performed by the used and helps to proceed further the following set of instructions or commands which has to be followed . The Voice input from the user using google assistant is dedected and the output is deliverred as per our input. ESP8266 is very useful in IOT industry where it is cost efficient, security, power is saved , compact in design and its performance is reliable The convenience is increased by accessing wherever and saves our valuable time, cost. ESP8266 would be no longer burned by the users heating bill and also it is cheap in cost for our home automation . In future scope, this technology of IoT can be used to monitor and control all home appliances remote through Wi-Fi network and smart devices by connecting appropriate relays and power supply .

REFERENCE:

- [1] Ms.M.Malathi , A.Gowsalya, M.Dhanushyaa, A.Janani, “Home automation on ESP 8266” SSRG International Journal of Computer Science and Engineering - (ICRTECITA-2017) - Special Issue - March 2017

- [2] K.Priyanka Srinivasulu Reddy on “Implementation of IOT with Esp8266 Part I - Creating A Prototype” - 2017 IJSRST | Volume 3 | Issue 1 | Themed Section: Science and Technology Syarif Hidayat and Syahrial Farid Firmanda on “Scheduler and Voice Recognition on Home Automation Control System” - 2015 3rd International Conference on Information and Communication Technology (ICoICT)
- [3] Tanish Sehgal, Shubham More on Home Automation using IOT and Mobile App - International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056 Volume: 04 Issue: 02 | Feb -2017 www.irjet.net p-ISSN: 2395-0072 , IRJET | Impact Factor value: 5.181 | ISO 9001:2008 Certified Journal
- [4] R.Aravindhan, M.Ramanathan, D.SanjaiKumar, R.Kishore on “HOME AUTOMATION USING Wi-Fi INTERCONNECTION “
- [5] International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056 Volume: 04 Issue: 03 | Mar -2017