Home Sync

Exploring Innovative Solutions in Home Automation Systems

Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science

By
Syed Taha Mohiuddin
Noman Khalil Siddiqui
Osama Jawed Khan

Under the Supervision of
Muhammad Zulqarnain Siddiqui
Department of Computer Science
Iqra University

Submission Date:
April, 2022

ABSTRACT

"This Project presents the overall design of Home Automation System (HAS) with low cost and wireless system. This system is designed to assist and provide support in order to fulfill the needs of elderly and disabled in home. Voice mode is used to control the home appliances. The main control system implements wireless technology to provide remote access from smart phone. The system intended to control electrical appliances and devices in house with relatively low cost design, user-friendly interface and ease of installation. In this project, we have designed, built a circuit which mainly meet the requirement of control the house appliances via any remote control device through voice controlling that is portable in the periphery of the room. The tool is able to control a load of excessive energy score from remote area."

1.1 Overview

We are now in a modern era technology is upgrading day by day and changing the user's life style of living, that time not far when user can easily control all their needs with their smart phone through internet. Now a day's people want to be modern themselves they have made technology as their part of life.

In this project, we have introduce a home automation system that will control several devices like Fan, TV, Light, motor. So in this project we are going to use Arduino board with wifi module. It will send instruction to the Arduino board with smart phone or remote for the programming we will use that default arduino IDE.[1]

1.2 Project Objective

The objective to create this project is to introduce a low cost and reliable home automation system that can easily accessible and controlled with smart phone through voice but this project is not only for home this technology can also use on industries and offices to controlled appliances by smart devices through voice. This project will save human energy and makes human life more easier. This can also be controlled from long range. This technology is very cheap and anyone can afford this technology easily and also it's not very difficult to use anybody can use this technology easily.

1.3 Project Scope

The scope of this project is to create the perfect home automation system that can easily control the home appliances with smart phone through voice. This project will also save the human time and energy as well this project can easily operate with smart phone also user can send commands with voice.

These are the following functions of this project are as follow:

- This automation system can be used on homes, offices, schools, malls, industries.
- To get access or control on this system internet is required.
- This technology is safe and friendly there is nothing harmful in this project in future user can add more devices to control easily.

COMPONENTS REQUIRED:

- NodeMCU ESP8266-12E Board
- Relay Module 5V 4-Chennel
- 3 Bulb Holder 220V
- 1 Fan
- 1 Female Socket 220V
- Jumper wires
- USB port

COMPONENTS ANALYSIS:

- NodeMCU ESP8266-12E Board:

Description:

ESP8266 NodeMcu is a popular and widely used development board based on the ESP-12E WiFi Module that combines elements of easy programming with Arduino IDE (C\C++) and WiFi capability. Through the build-in programmer and CH340G USB-to-Serial chip, flashing the ESP8266 and serial output on a PC, development and prototyping projects are done with ease. Just like Arduino boards, the ESP8266 NodeMcu has GPIO pins, voltage regulator, ADC, Micro-USB port (for flashing and serial output) – all on one board. On top of that the ESP8266 NodeMcu has a full WiFi that takes care of the WiFi communication to a server or client.

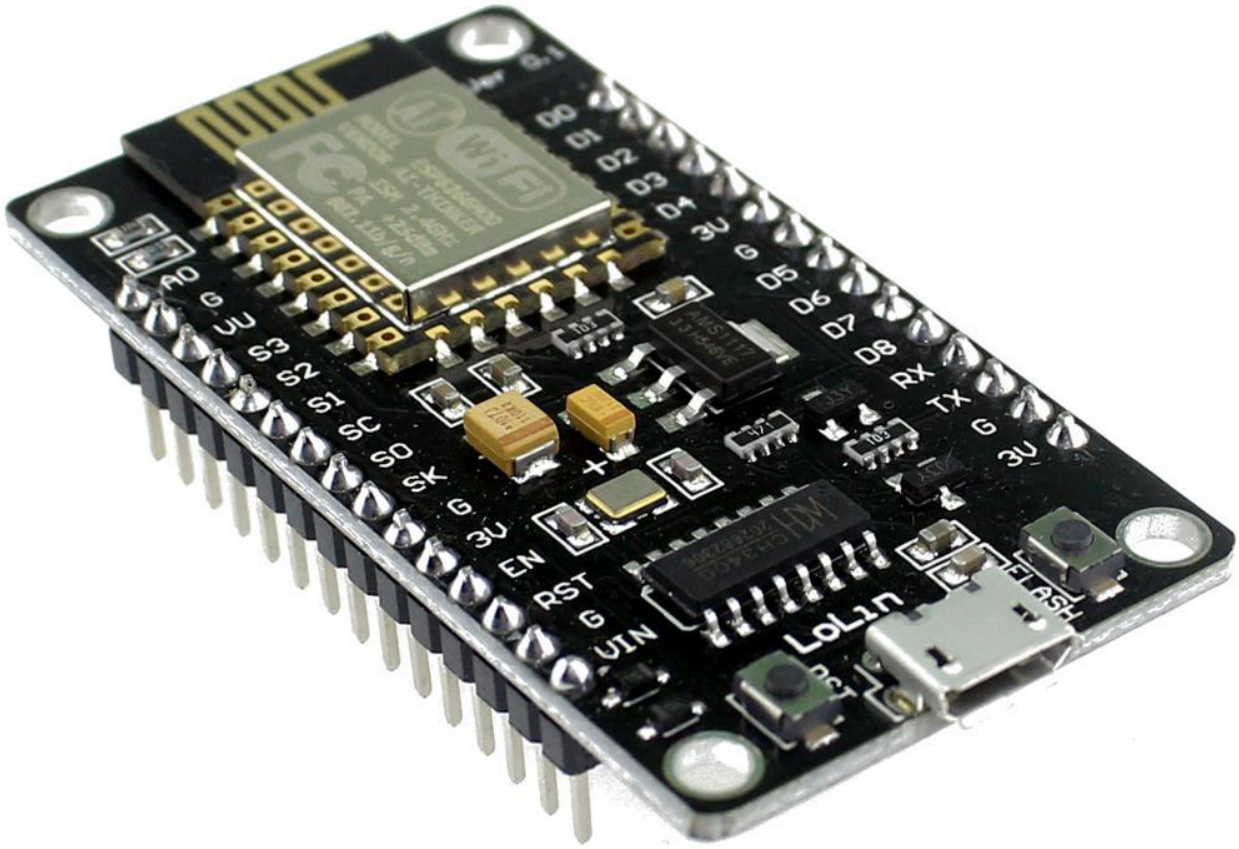
The ESP8266 is a System on a Chip (SoC), manufactured by the Chinese company Espressif. It consists of a Tensilica L106 32-bit micro controller unit (MCU) and a Wi-Fi transceiver. It has 11 GPIO pins* (General Purpose Input/Output pins), and an analog input as well. This means that you can program it like any normal Arduino or other microcontroller. And on top of that, you get Wi-Fi communication, so you can use it to connect to your Wi-Fi network, connect to the Internet, host a web server with real web pages, let your smartphone connect to it, etc … The possibilities are endless! It's no wonder that this chip has become the most popular IOT device available.

It contains a built-in 32-bit low-power CPU, ROM and RAM. It is a complete and self-contained Wi-Fi network solution that can carry software applications as a stand-alone device or connected with a microcontroller (MCU). The module has built-in AT Command firmware to be used with any MCU via COM port. The ESP8266 can be

flashed and programed using the Arduino IDE. Due to its large open source developer community, a large number of libraries for this popular microcontroller is available.

Processor:

The ESP8266EX integrates a Tensilica L106 32-bit RISC processor, which achieves extra-low power consumption and reaches a maximum clock speed of 160 MHz. The Real-Time Operating System (RTOS) and Wi-Fi stack allow 80% of the processing power to be available for user application programming and development.



- Relay Module 5V 4-Chennel:

Description:

The four-channel relay module contains four 5V relays and the associated switching and isolating components, which makes interfacing with a microcontroller or sensor easy with minimum components and connections. There are two terminal blocks with six terminals each, and each block is shared by two relays. The terminals are screw type, which makes connections to mains wiring easy and changeable.

The four relays on the module are rated for 5V, which means the relay is activated when there is approximately 5V across the coil. The contacts on each relay are specified for 250VAC and 30VDC and 10A in each case, as marked on the body of the relays.

The switching transistors act as a buffer between the relay coils that require high currents, and the inputs which don't draw much current. They amplify the input signal so that they can drive the coils to activate the relays. The freewheeling diodes prevent voltage spikes across the transistors when the relay is turned off since the coils are an inductive load. The indicator LEDs glow when the coil of the respective relay is energized, indicating that the relay is active. The optocouplers form an additional layer of isolation between the load being switched and the inputs. The isolation is optional and can be selected using the VCC selector jumper. The input jumper contains the main V-CC, GND, and input pins for easy connection using female jumper wires.

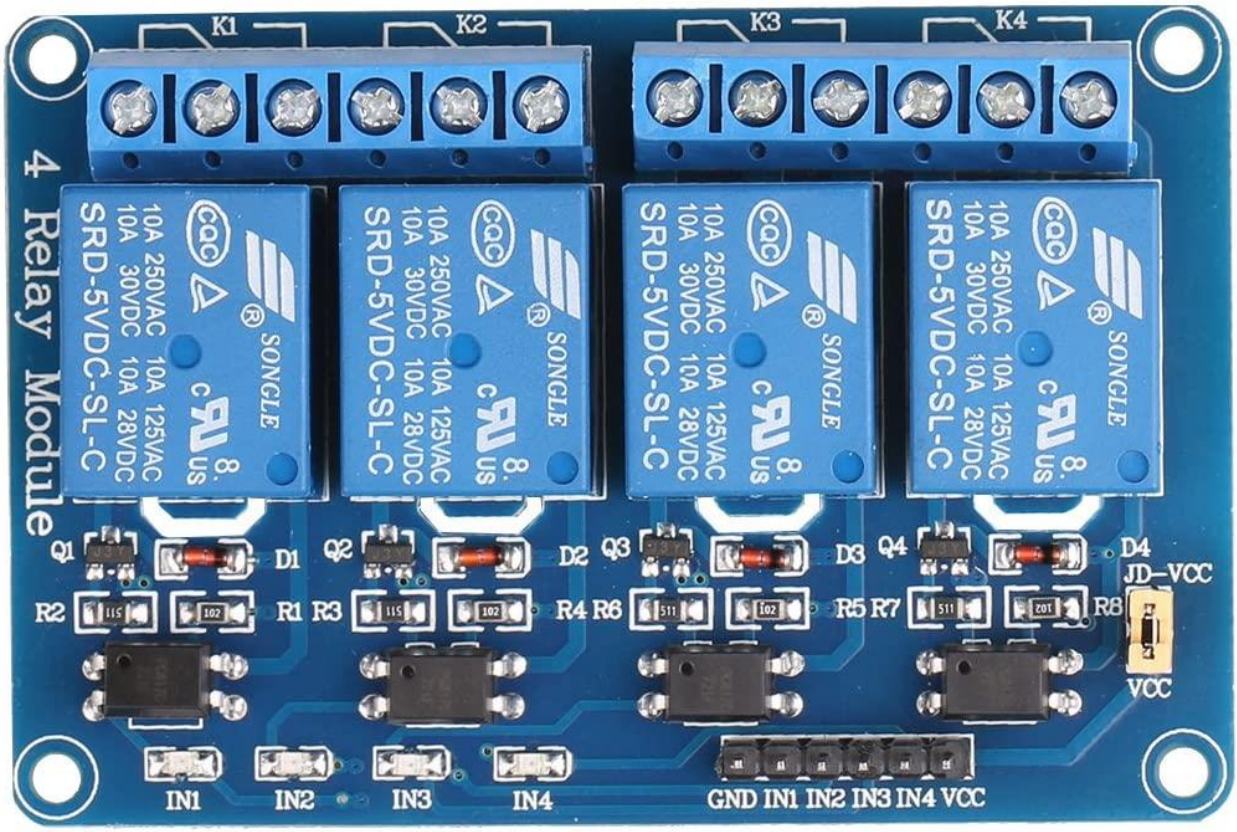
Components Present on a Four-Channel Relay Module:

Following are the major components present on the four-channel relay module, we will get into the details of this later in the article.

5V relay, terminal blocks, male headers, transistors, optocouplers, diodes, and LEDs.

Four-Channel Relay Module Specifications:

- Supply voltage – 3.75V to 6V
- Trigger current – 5mA
- Current when the relay is active - ~70mA (single), ~300mA (all four)
- Relay maximum contact voltage – 250VAC, 30VDC
- Relay maximum current – 10A



Functional Requirements:

- ARDUINO NODE MCU:

  The essential part of our project supports the home automation system of wireless device requirements.

- Voice Sensor :

  The most essential part of our project detect the user voice to control the lights and fan.

- Login ID:

  Admins and Users can sign up from a mobile application or log in from an existing ID.

- User verification:

  The system verify the user through mobile application and voice sensor.

- Remote Control:

  User can control the system with function of on or off remote control.

Non-Functional Requirements:

- Availability:

  In home automation system, the system should be fully operational constantly, so any client can get to the mobile application any time.

- Usability:

  The interface ought to give clients admittance to all important highlights effectively or as such, the system's interface should be easy to use.

- Reliability:

  If the person faces a connectivity issue that leads to system malfunction, the user will be provided a manual input option for further process.

- Accuracy:

  The project is comprised of different functions after detecting the user voice which is efficient to respond to any task given, to perform with high-speed internet is used, the application offers various options to look into details, the user panel is user friendly that does not create any panic.

- Efficiency:

  The system will be able to perform its operation with efficiency. The scope of the system must be to perform the home automation functions in system. The system will be able to show the related results in no more than a minute.

METHODOLOGY

A machine learning algorithm was proposed as a method in smart home automation to detect gas leakage, smoke, and fire in the home by Salhi et al. The approach presented by the authors uses a data flow model and data acquisition through sensors in the home for analytics and prediction. In conjunction with the sensed information, the data mining method was used to detect abnormalities in the air.Home automation is a method of controlling home appliances automatically for the convenience of users. This technology makes life easier for the user, and saves energy by utilizing devices according to strict requirements.Machine learning is a methodology of information analysis that automates analytical model building. These algorithms that iteratively learn from information, machine learning permits system to search out hidden insights while not being expressly programmed wherever to seem.Automation is a technique, method, or system of operating or controlling a process by electronic devices with reducing human involvement to a minimum. The fundamental of building an automation system for an office or home is increasing day-by-day with numerous benefits. Industrialist and researchers are working to build efficient and affordability automatic systems

After carefully analyzing different ideas for the project proposal, I came across with one of the interesting and indeed the most emerging idea that would really cause a huge impact on the people interact with the objects. It took me a while to go through various literature reviews and methodologies to bring up with something that could be much more efficient and of course sustainable.The application input is provided by the user using the Google Assistant through his mobile phone. Using this application the users can schedule the devices connected to Arduino and ESP8266. The input given by the users is fed into the Firebase. The Wi-Fi connected ESP8266 fetches the information from the firebase and turn on or off the devices depending upon the users requests. Any device can be scheduled independently using this application.The smart home environment offers to appropriate home setup where domestic appliances and devices connected together through internet which controlled by any mobile or remote device from anywhere. According to the house owner's desire, the facility of every smart home can be setup with help of service devices

Home Automation is done using Global System for Mobile communication. The features of GSM are that it uses the network of the mobile and is battery powered. It consists of two important components which are GSM modem and the microcontroller.The main purpose of the home automation system is to provide a cheap, secure and open source home automation which can be able to control all the home appliances through android device. The main advantages of home automation is that it provide security and flexibility through the android system, good range of scalabilityThe cost of investment: Compared to standard home automation,

intelligent home automation requires you to chip in a more significant investment.Reliance on an Internet connection. Because most smart home technology requires an Internet connection to operate, when the Internet goes down so does all of your smart home programming. Make sure that your smart home technology can be operated manually for these outages.If the smart technology is controlled by Wi-Fi like the IoT devices, you need to ensure that you are using a private WiFi network channel which can't be accessed from outside.

Agile

Home automation describes a system of Bluetooth and mobile application controllable that make our life easy. In this device there are several main parts Arduino, Bluetooth module, Relay drivers, ultrasonic sensor, servo motor, temperature sensor, android application and step-down transformer. Firstly, we provide power to the step-down transformer, it steps down the input voltage and given to the Arduino with VIN pin. The Bluetooth module is also connected with Arduino to Rx and Tx pin that provides the information to the microcontroller. Microcontroller reads the information and send to the relay drivers which work as a switch. In Arduino we upload the program as per requirement, then it performs some mathematical and logical operation to control the relay drivers. We also declare the water level and temperature for pump on/off and fan speed control. In the application we also set the light/fan button. Water motor on/off automatically by declaring water level. Also, we use a buzzer for motor for confirmation by hearing beep.

Objective

This project focuses on the design of a home automation system as an open source platform for all end users. The objectives of the project are as followsTo design and construct a microcontroller-based system that effectively controls and monitors devices in the home system To develop a remote control system that enables data transfer through wireless transfer medium To interface the system with a web browser enabling cross platform control To design a system that is user friendly and easy to use

Project goals:

The aim of this project is to design an open source, easy-to-use and affordable home automation system. For this reason the Arduino microcontroller is used, acting as the main controller for the system by sending signals to control electronic appliances. The project will be limited to a home model for prototyping purposes. A web browser works as the user interface while a standard wireless network is used as the medium between Arduino and web browser.

Goals of the Mobile Phone (Android) Application:

This has revolutionized the area of home automation with respect to an increased level of affordability and simplicity through the integration of home appliances with smart phone and tablet connectivity. Smart phones are already feature-perfect and can be made to communicate to any other devices in an ad hoc network with a connectivity options like Bluetooth . With the advent of mobile phones, Mobile applications development has seen a major outbreak. Utilizing the opportunity of automating tasks for a smart home, mobile phone commonly found in normal household can be joined in a temporary network inside a home with the electronic equipments. Android, by Google Inc. provides the platform for the development of the mobile applications for the

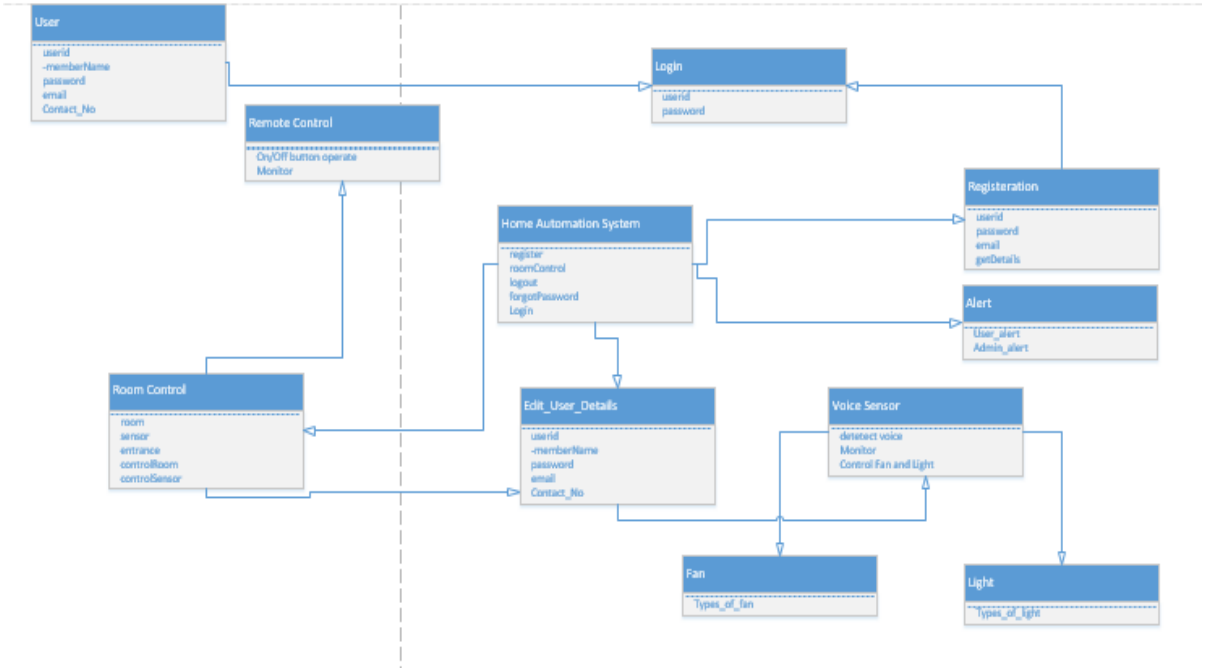
Android devices . Home automation system is a mobile application developed using Android targeting its vast market which will be beneficial for the masses. According to the International Data Corporation (IDC) Worldwide Quarterly Mobile Phone Tracker, Android maintained its leadership position in global market share . Bluetooth is a short-range wireless communication technology that comes in handy as the solution while communicating over an ad hoc network environment like the home environment for connecting the home appliances with the mobile phones . Bluetooth works over 2.4 GHz frequency range up to the range of 100 m with 1 Mbps speed, providing a safe and efficient solution for controlling home automation.

Implementation:

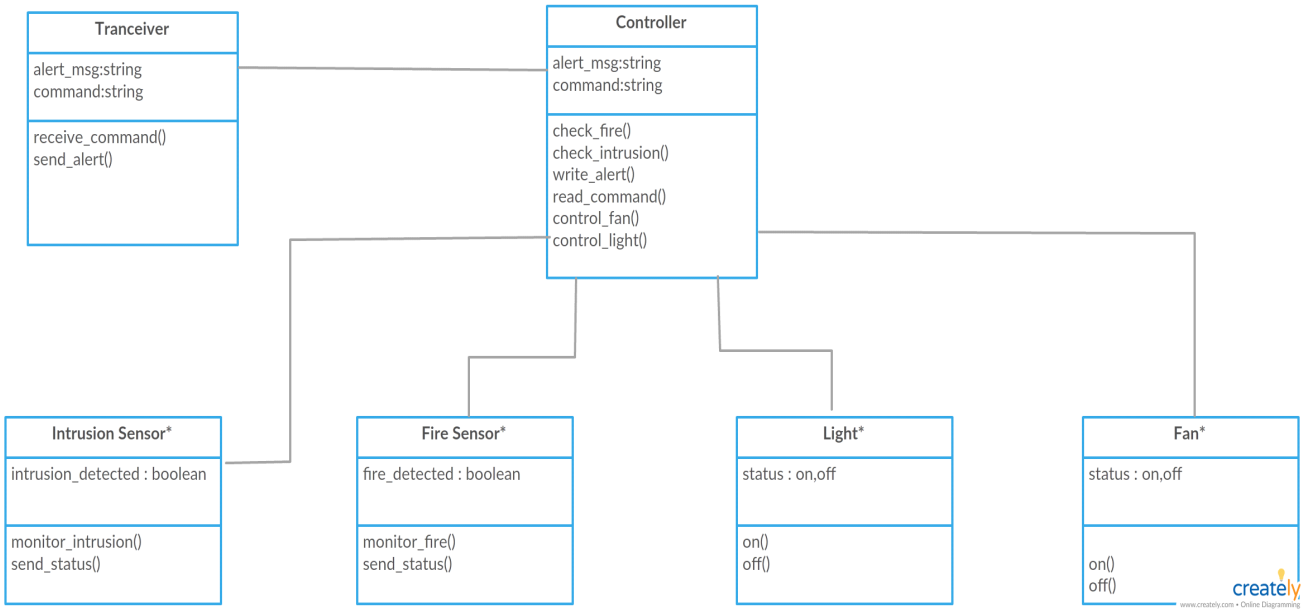
wireless communication of various machines and devices in mobile networks is a fast growing business and application area in industry, maintenance business, customer service, security, and banking areas. This paper presents design and implementation of remote control system by means of GSM cellular communication network. The design integrates the device to be controlled, the microcontroller, and the GSM module so that it can be used for a wide range of applications. Detailed description and implementation of each design element are presented. To verify the principle operation of the M2M design, two home applications are experimentally tested using PC-based environment.

Data Flow Diagram (DFD):

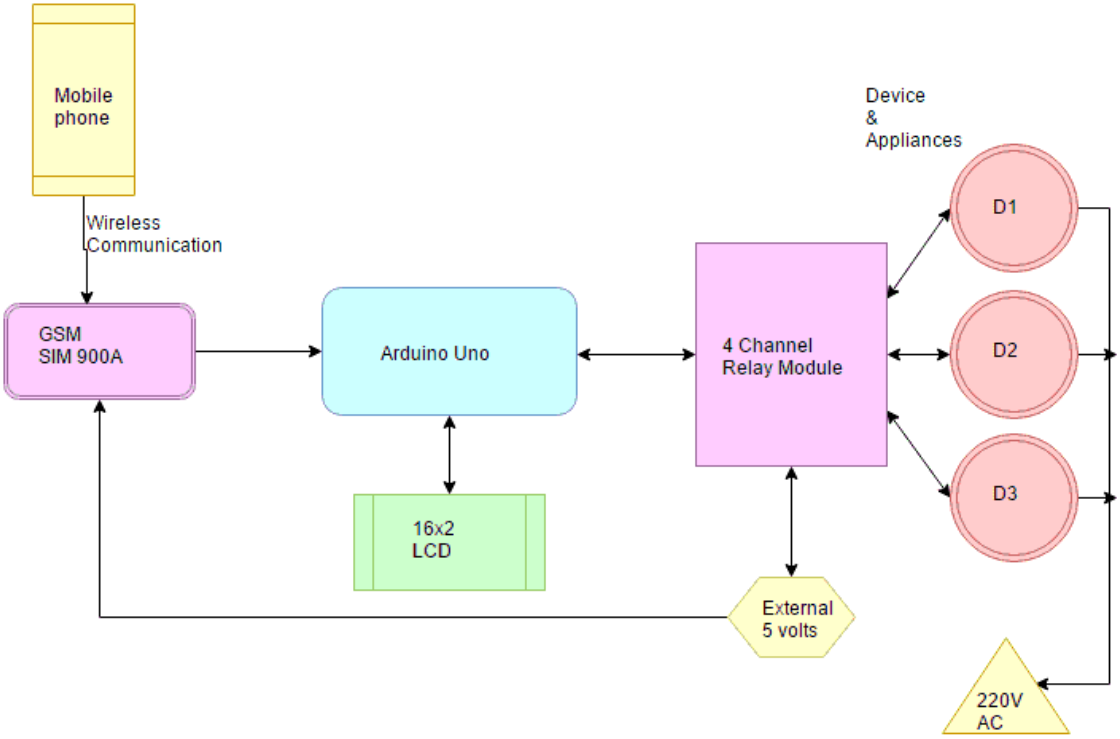
The Data Flow Diagram (DFD) depicts the logic models and expresses data transformation in a system. It includes a mechanism to model the data flow and supports decomposition to illustrate details of the data flows and functions. A Data Flow Diagram cannot present information on operation sequence**.**



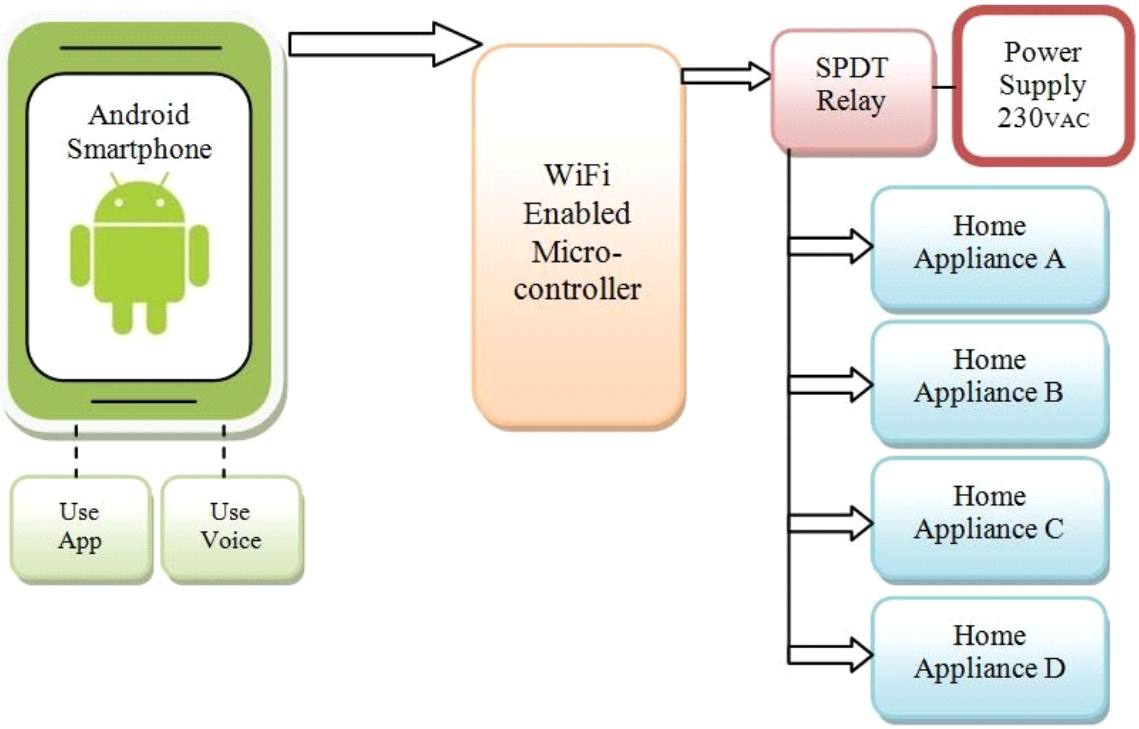
UML DIAGRAM:


Tranceiver
alert_msg:string
command:string
receive_command()
send_alert()
Controller
alert_msg:string
command:string
check_fire()
check_intrusion()
write_alert()
read_command()
control_fan()
control_light()
Intrusion Sensor*
intrusion_detected : boolean
monitor_intrusion()
send_status()
Fire Sensor*
fire_detected : boolean
monitor_fire()
send_status()
Light*
status : on,off
on()
off()
Fan*
status : on,off
on()
off()
creately
www.creately.com • Online Diagramming

INTERACTION DIAGRAM:

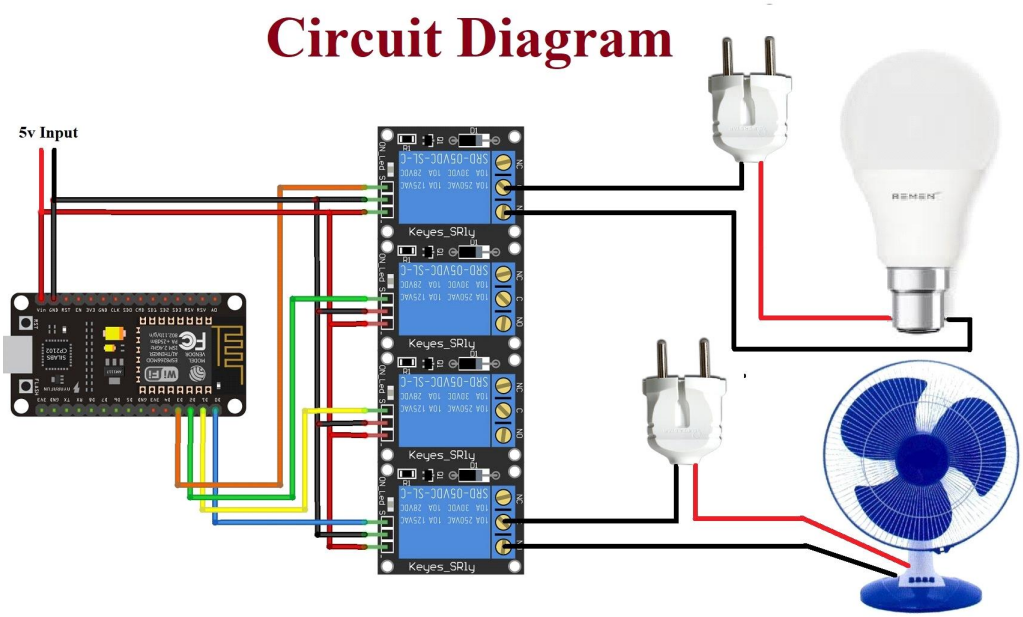


CUSTOMER:

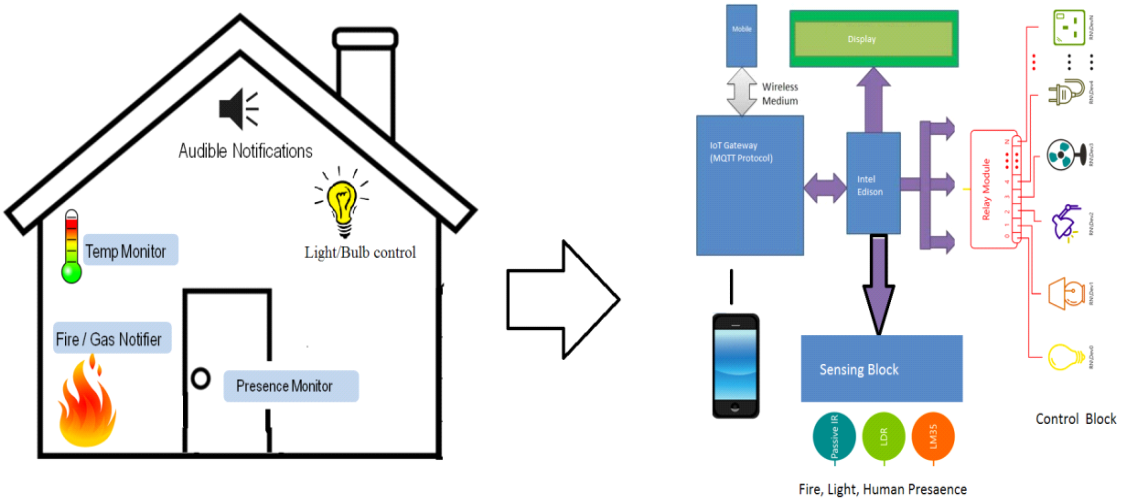

Android Smartphone
Use App
Use Voice
WiFi Enabled Micro-controller
SPDT Relay
Power Supply 230VAC
Home Appliance A
Home Appliance B
Home Appliance C
Home Appliance D

USE CASE DIAGRAM:

ACTIVITY DIAGRAM:



ARCHITECTURE DIAGRAM:



Conclusion:

In this project, we have introduced the event of a home management using Arduino, Node MCU and internet of things technology. The system is suitable for remotely controlling the home appliances. A smart home system integrates various electrical appliances in a home with each other using information devices automatically according the users need. For the mobile application the C language part is provided inside the program thus it doesn't require any other application to be developed for different gadgets. The automated mode makes life easier for users by complete automation of necessary appliances without any human effort.

Future scope:

Future scope for the home automation systems involves making homes even smarter. Homes can be interfaced with sensors including motion sensors, light sensors and temperature sensors and provide automated toggling of devices based on conditions. More energy can be conserved by ensuring occupation of the house before turning on devices and checking brightness and turning off lights if not necessary. The system can be integrated closely with home security solutions to allow greater control and safety for home owners. The next step would be to extend this system to automate a large-scale environment, such as offices and factories. Home Automation offers a global standard for interoperable products. Standardization enables smart homes that can control appliances, lighting, environment, energy management and security as well as the expandability to connect with other networks.

Source Code:

Hardware Code:

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#ifndef UNIT_TEST
#include <Arduino.h>
#endif
#include <WiFiManager.h>
long day = 86400000; // 86400000 milliseconds in a day
long hour = 3600000; // 3600000 milliseconds in an hour
long minute = 60000; // 60000 milliseconds in a minute
long second =  1000; // 1000 milliseconds in a second
long motions=0;
int seconds;
//const int s5=16; //D02
const int s4=14; //D02
const int s1 = 5;  //D21
const int s2 = 4;  //D22
const int s3 = 12;  //D3
long minutes;
int led[3]={13,16,15};
int Status = 12;  // Digital pin D6
const int ledon=LOW;
const int ledoff=HIGH;

int led_delay=200;

int sensor = D7;  // for PIR SENSOR


//const int s4 = 2;  //D4
```

```
//const int s6=0;//D7


int To_Min=0;


#define mqtt_server "tailor.cloudmqtt.com"
#define mqtt_user "blailqau"
#define mqtt_password "DN9Q3taI8Zay"
#define clientName "EspKH9NGAJRFEEyEi"

WiFiClient espClient;
PubSubClient client(espClient);


void red(){
   digitalWrite(led[1], ledoff);
  digitalWrite(led[2], ledoff);


  digitalWrite(led[0], ledon); //red


}

void blue(){
  digitalWrite(led[0], ledoff);
   digitalWrite(led[2], ledoff);
    digitalWrite(led[1], ledon);//blue



}
void green(){

  digitalWrite(led[0], ledoff);
  digitalWrite(led[1], ledoff);
```

```
  //digitalWrite(led[2], ledon); //green
}
void setup() {
  Serial.begin(115200, SERIAL_8N1, SERIAL_TX_ONLY);
  Serial.begin(115200);

  for(int i=0; i<3; i++) //for leds
  {
     digitalWrite(led[i],LOW);
    pinMode(led[i],OUTPUT);
    //digitalWrite(led[i],ledoff);

  }
  red();
  setup_wifi();
  blue();
  client.setServer(mqtt_server, 10623);
  client.setCallback(callback);
  pinMode(2, OUTPUT);
  pinMode(s1,OUTPUT);
  pinMode(s2,OUTPUT);
  pinMode(s3,OUTPUT);
  pinMode(s4,OUTPUT);

  pinMode(sensor, INPUT);
 // pinMode(s5,OUTPUT);
  //pinMode(s6,OUTPUT);


  digitalWrite(s1,HIGH);
  digitalWrite(s2,HIGH);
 digitalWrite(s3,HIGH);
 digitalWrite(s4,HIGH);
 //digitalWrite(s5,HIGH);
 //digitalWrite(s6,HIGH);
```

```
}

void setup_wifi() {
  delay(10);
  //WiFi.disconnect();
    WiFiManager wifiManager;
    wifiManager.autoConnect("Hassan Project");

    Serial.println("connected...yeey :)");

    delay(500);
  }


void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    if (client.connect(clientName, mqtt_user, mqtt_password)) {
 //client.publish("Connected", "");
       client.subscribe("topcity");
        client.subscribe("Connected");
        green();


    } else {
      // Wait 5 seconds before retrying
      delay(5000);
     // client.publish("EspKH9NGAJRFEEyEi", "Connected");
   //  client.publish("Connected", "");
    }
  }
}

bool checkBound(float newValue, float prevValue, float maxDiff) {
  return !isnan(newValue) &&
         (newValue < prevValue - maxDiff || newValue > prevValue + maxDiff);
```

```
}

void callback(char* topic, byte* payload, unsigned int length) {
  String a;
  String getTopic=topic;
 // getTopic="";
  a = "";

  for (int i = 0; i < length; i++) {
    a += (char)payload[i];
   // getTopic+=(char)topic[i];
  }
  Serial.println("Topic: ");

  Serial.println(getTopic);
  if(getTopic=="Connected")
  {
    Serial.println("Connected sent");

    client.publish("Success", "EspKH9NGAJRFEEyEi is Connected");

    getTopic="";
    // client.subscribe("Connected");
  }
  if (a == "L1:1") {
    Serial.println("Light is ON : ");
      digitalWrite(s1,LOW);
       client.publish("Success", "");
  }
  if (a == "L1:0") {
    Serial.println("Light is OFF : ");
      digitalWrite(s1,HIGH);
  }

   if (a == "L2:1") {
      digitalWrite(s2,LOW);
  }
```

```
  if (a == "L2:0") {
     digitalWrite(s2,HIGH);
  }


   if (a == "L3:1") {
     digitalWrite(s3,LOW);
  }
  if (a == "L3:0") {
     digitalWrite(s3,HIGH);
  }




     if (a == "FanOn") {
     digitalWrite(s4,LOW);
  }
  if (a == "FanOff") {
     digitalWrite(s4,HIGH);
  }









}
void time(){

long timeNow = millis();
 //Serial.println("Time now: ");
// Serial.println(timeNow);
int days = timeNow / day ;                            //number of days
```

```
int hours = (timeNow % day) / hour;                    //the remainder from days
division (in milliseconds) divided by hours, this gives the full hours
 minutes = ((timeNow % day) % hour) / minute ;        //and so on...
 seconds = (((timeNow % day) % hour) % minute) / second;

 // digital clock display of current time
// Serial.println(days,DEC);
 printDigits(hours);
 printDigits(minutes);
 printDigits(seconds);
 //Serial.print("Minutes: ");
 //Serial.print(minutes);

 Serial.println("seconds: ");
 Serial.print(seconds);
  if(seconds==59)
   {
    To_Min=To_Min+1;
   }

}

void printDigits(byte digits){
 // utility function for digital clock display: prints colon and leading 0
 //Serial.print(":");
 if(digits < 10)
   Serial.print("");
 //Serial.print(digits,DEC);
}

void loop() {


   time();
    delay(1000);

   //client.publish("EspKH9NGAJRFEEyEi", "Connected");
```

```
long state = digitalRead(sensor);

 Serial.println(state);
  if(state == HIGH) {
   // digitalWrite (Status, HIGH);
    Serial.println("Motion detected!");
     // client.publish("topcity", "Motion Detected");
    motions++;

  }
  if(state==LOW) {
  //  digitalWrite (Status, LOW);
    Serial.println("Motion absent!");
     // client.publish("topcity", "Motion Absent");

    }
    if(To_Min==2)
    {
      if(motions==0)
      {
        Serial.println("Alaram...! No motions no working no sallary :D");
        client.publish("topcity", "Alaram");

         //minutes=0;
         To_Min=0;
      }
      else
      {
        Serial.print("number of motion detected: ");
        //Serial.print(motions);
         client.publish("topcity","Motions Detected");
         To_Min=0;
```

```
        // minute=0;
         motions=0;

       }


     }

  if (!client.connected()) {
    reconnect();
  }
  client.loop();


}
```

App Main Code:

```
package com.example.homeautomation;

import androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextClock;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        TextClock clock = (TextClock) findViewById(R.id.c1);


        CardView cardView = findViewById(R.id.lr);
        cardView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, LivingRoom.class);
                startActivity(intent);
            }
        });
    }
}
```

Living Room:

```
package com.example.homeautomation;

import androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;

import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.snackbar.Snackbar;

public class LivingRoom extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_living_room);

        CardView cardView = findViewById(R.id.r1);
        cardView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(LivingRoom.this, roomone.class);
                startActivity(intent);
            }
        });

        FloatingActionButton fab = findViewById(R.id.fab1);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Snackbar.make(v, "Here is a snack bar", Snackbar.LENGTH_LONG)
                .setAction("Action", null).show();

            }
```

```
        });
    }
}
```

Room One:

```
package com.example.homeautomation;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class roomone extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_roomone);
    }
}
```

Example_ Instrumentat:

```
package com.example.homeautomation;

import android.content.Context;

import androidx.test.InstrumentationRegistry;
import androidx.test.runner.AndroidJUnit4;

import org.junit.Test;
import org.junit.runner.RunWith;

import static org.junit.Assert.*;

@RunWith(AndroidJUnit4.class)
```

```
public class ExampleInstrumentedTest {
   @Test
   public void useAppContext() {
      // Context of the app under test.
      Context appContext = InstrumentationRegistry.getTargetContext();

      assertEquals("com.example.homeautomation",
appContext.getPackageName());
   }
}
```

Example_ UnitTest:

```
package com.example.homeautomation;

import org.junit.Test;

import static org.junit.Assert.*;
public class ExampleUnitTest {
   @Test
   public void addition_isCorrect() {
      assertEquals(4, 2 + 2);
   }
}
```

Build.Gradle:

```
// Top-level build file where you can add configuration options common to all
sub-projects/modules.

buildscript {
   repositories {
      google()
      jcenter()

   }
   dependencies {
```

```
        classpath 'com.android.tools.build:gradle:3.4.2'

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        google()
        jcenter()

    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

App_Build.Gradle:

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 30
    buildToolsVersion "30.0.2"
    defaultConfig {
        applicationId "com.example.homeautomation"
        minSdkVersion 19
        targetSdkVersion 30
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
```

```
        proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'),
'proguard-rules.pro'
      }
   }
}

dependencies {
   implementation fileTree(dir: 'libs', include: ['*.jar'])
   implementation 'androidx.appcompat:appcompat:1.2.0'
   implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
   testImplementation 'junit:junit:4.12'
   androidTestImplementation 'androidx.test:runner:1.3.0'
   androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
   implementation 'androidx.cardview:cardview:1.0.0'
   implementation 'androidx.recyclerview:recyclerview:1.1.0'
   //noinspection GradleCompatible
   implementation 'com.android.support:design:28.0.0'
}
```