Introduction to Internet of Things

Design Project

IoT Enabled Home Automation System

Submitted by:

M.Moaz Awan	19L-1367
Ahmed Hanif	19L-1316

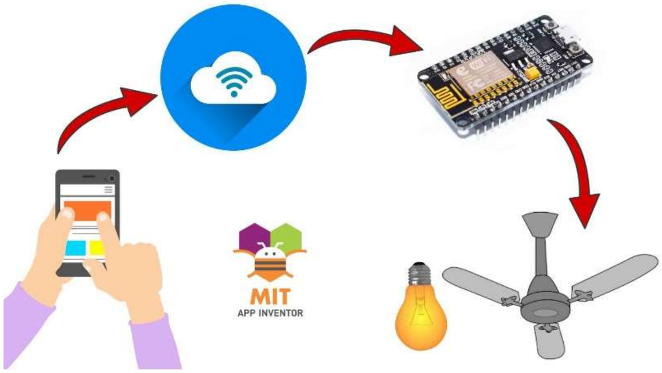
Submitted to: Ms Khizra Farooq

7 November, 2022

Department of Electrical Engineering
National University of Computer and Emerging Sciences, Lahore

Table of Contents

Introduction



A load controlled by computer systems has many advantages compared with manual controlled loads. Nowadays there are many programs and applications help to control things better using codes or new algorithms in artificial intelligence projects. In order to save energy and make loads monitored easily, this research suggests smart home project based on IoT technology. This smart home is an Internet of Things (IoT) project that controls loads with internet connection via Wireless Fidelity WIFI connection. A smart phone connected to internet with MIT App Inventor application as a control panel, and NodeMCU microcontroller kit in other side as a controller that receives control commands via WIFI signal. NodeMCU kit is built with ESP8266 WIFI receiver that able to process and analyze WIFI signal to input the microcontroller. The WIFI receiver and microcontroller are built in one kit to be used as IoT project. It’s called NodeMCU.

To connect the system to the Internet, needs a WiFi receiver. In my case I used ESP8266 that is connected as built-in in the NodeMCU board that contains a firmware runs with the ESP8266. The firmware is a low-level control computer software.

The NodeMCU is coded via Arduino Integrated Development Environment (IDE) with the Universal Serial Bus port (USB) to tell the NodeMCU what to do, I want to make the NodeMCU controls two-channel relay kit by App invented using MIT App Inventor hand phone application and shows the status of LED, FAN , Door Lock Status.

Problem Analysis

Our Problem is to design an automated home system to control the light, fan and door lock based on the real time data obtained from sensor. We have to upload the data on cloud using a IOT platform. Further we gave to Interface ultrasonic sensor with NodeMCU to detect the person and the door should get open as soon as a person is detected in front of door and remain close otherwise. If the door gets open turn the light and fan ON and upload the status of your components on cloud. In the end we have to design an android app to create a user interface using MIT app invertor.

Solution to the Problem:

We are going to follow these steps in order to solve the given problem:

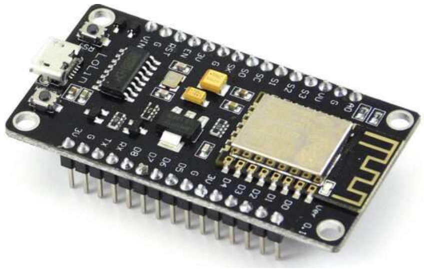
1. First we will interface ultrasonic sensor with NodeMCU and display data on serial monitor.
2. After interfacing ultrasonic sensor we will get distance from it start interfacing led and fan and servo motor with our nodeMCU.
3. In order to demonstrate Door lock we will use servo motor in such a way that if door is unlocked nodemcu will set servo motor to 0° and if door is locked NodeMCU will set servo 180°.
4. After interfacing all these things we will send the status of your interfaced components on cloud. We will choose Firebase as an IOT platform because it is very fast in order to read or write data.
5. In the End, we will design a mobile app to create a user interface for our design.Our app will be linked with your Firebase in order read data from cloud and also write data to the cloud. Our App will show status of LED,FAN and Door lock and also we will be able to update the status of LED, FAN, Door lock using this App.

Design Requirements

Hardware Requirements:

NodeMCU ESP8266:

NodeMCU is an open-source firmware and development kit that helps you to prototype or build IoT products. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.



LED :

The mini LED will be used to demonstrate a bulb.

DC Motor :

This mini DC motor will be used to demonstrate Fan.

Servo Motor :

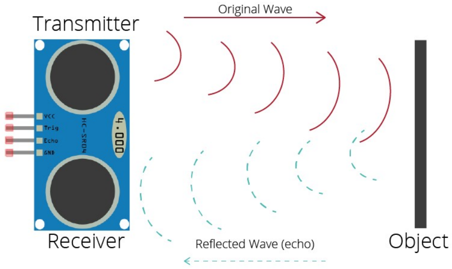
This Servo Motor HD-1800A will be used to demonstrate the Door Lock . The HD-1800A analog servo from Power HD is a great general-purpose micro servo at a low price.

Two Channel relay :

In order to drive dc motor we require a dc motor driver or a relay so,we choosed relay because it is cheap and easily available and we can connect AC and bulb with it. It is a 5V operated Relay.

Design Requirements

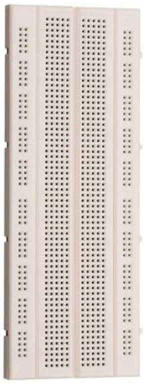
Ultrasonic Senor:



The HC-SR04 ultrasonic sensor uses sonar to determine the distance to an object. This sensor reads from 2cm to 400cm (0.8inch to 157inch) with an accuracy of 0.3cm (0.1inches), which is good for most hobbyist projects. In addition, this particular module comes with ultrasonic transmitter and receiver modules.

Jumper wires :

We will use Male to-Male and Female to Male jumper wires.



BreadBoard :

A breadboard will be required to implement the complete circuit.

Software Requirements:

Arduino IDE:

Arduino IDE is an open-source software, designed by Arduino.cc and mainly used for writing, compiling & uploading code to almost all Arduino Modules. It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.

Google Firebase:

Firebase is a set of hosting services for any type of application. It offers NoSQL and real-time hosting of databases, content, social authentication, and notifications, or services, such as a real-time communication server.

Design Requirements

MIT APP Inventor:

MIT App Inventor is a web application integrated development environment originally provided by Google, and now maintained by the Massachusetts Institute of Technology (MIT). It allows newcomers to computer programming to create application software (apps) for two operating systems (OS): Android, and iOS, which, as of 8 July 2019, is in final beta testing. It is free and open-source software released under dual licensing: a Creative Commons Attribution ShareAlike 3.0 Unported license, and an Apache License 2.0 for the source code.

Feasibility Analysis

This project is economically, technically, and operationally feasible since it can be done utilising low-cost electronic and software technology.

Feasibility from a financial standpoint :

This project uses an Android phone and a few low-cost electronic components such as the NodeMCU ESP8266 Microcontroller, 2 channel relay module, led, DC motor, Ultrasonic senor and some jumpers to make it cheaply feasible to build.

Feasibility from a technical standpoint:

This project is based on wireless technologies, hardware and software based that are quite contemporary in terms of technology. As a result, technology favours it greatly.

Feasibility of Operation:

The Arduino IDE software and MIT App inventor is a very simple, user-friendly interface, making it pretty nearly usable by anyone. Controlling components using ultrasonic sesnor and also with an mobile Application could be useful for physically challenged people as well. As a result, it is practicable.

Low cost operations:

The cost of an IoT deployment is determined by a number of factors, including the hardware and software costs. A smart home automation system's software does not cost a money. An android app can be build without any cost using roughly 4-5 man hours to develop. A Cloud server Firebase is also free in our case and IoT hardware, such as a sensor, board, microcontroller and relays, would be required to host the application.

Possible Solutions

Possible solutions in a our case are many in which some of them are as follows:

For Operation of LED and FAN:

In order to operate fan we had two choices :

1. **Interface a Motordriver with NodeMCU**

   If we use a motordriver then we can only drive dc motors through this but we will not be able to connect AC Fan or Led with it or any other home appliance so it will not be a good choice to interface it with NodeMCU.

2. **Interface a Relay Module with NodeMCU**

   We choosed a 2 channel relay module beacause it is the best possible choice in order to implement it in a real time scenario, we can interface any home appliance with max 10 amperes and 250Volts AC.

For Operation of Door Lock:

In Order to demonstrate Door LOCK we used Servo motor HD-1800A because it is powerful as compared to other motors available in the market a way it will be used in such a way that if door is unlocked nodemcu will set servo motor to 0° and if door is locked NodeMCU will set servo 180°.

For Cloud Storage:

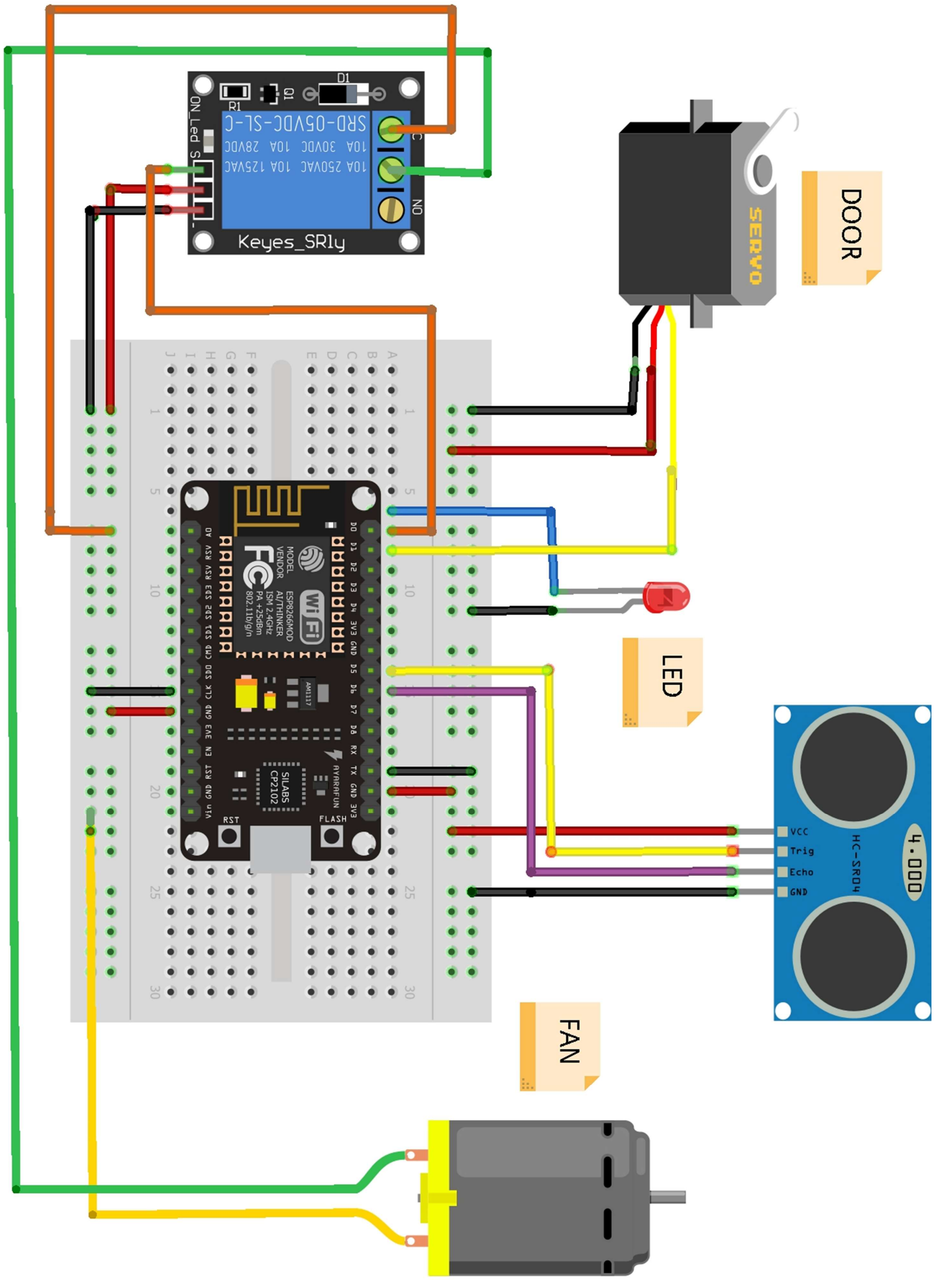
For storing the status of Led, Fan and door rather it is on or off we have to store data on cloud fromwhere we can read, write or update it . So, for the cloud stroage there are many available IOT API’s like Thingspeak,Thingsboard and Google Firebase. We choosed google Firebase because it is very fast and easy to use with NodeMCU.

For Andriod Application:

In order to control the status of interface compnents we need an mobile application from where we can update it, that app will update read ,write or update data to Firbase so the possible solutions could be to build an andriod application from Scratch or we can use an app inventor so we choosed MIT App Inventor to build a mobile application and connected it with Firebase.

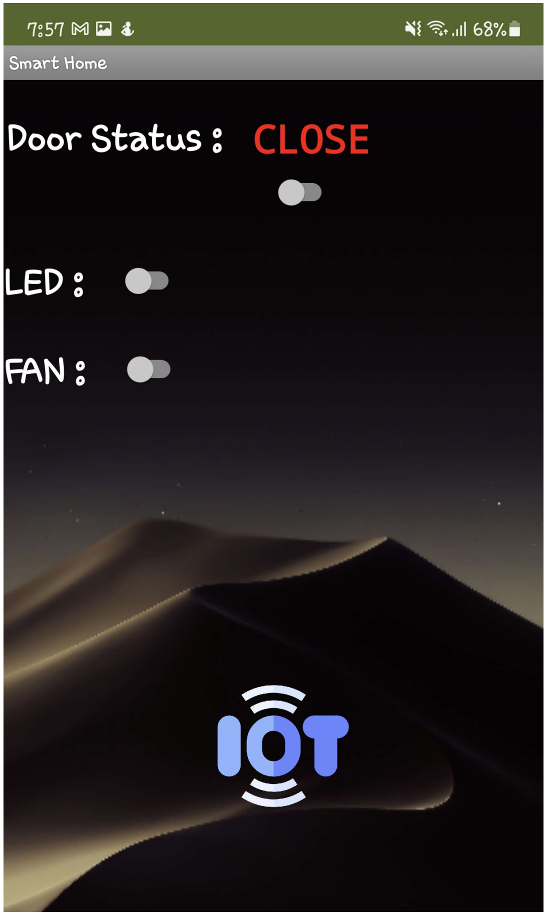
Preliminary Design

Hardware Design:

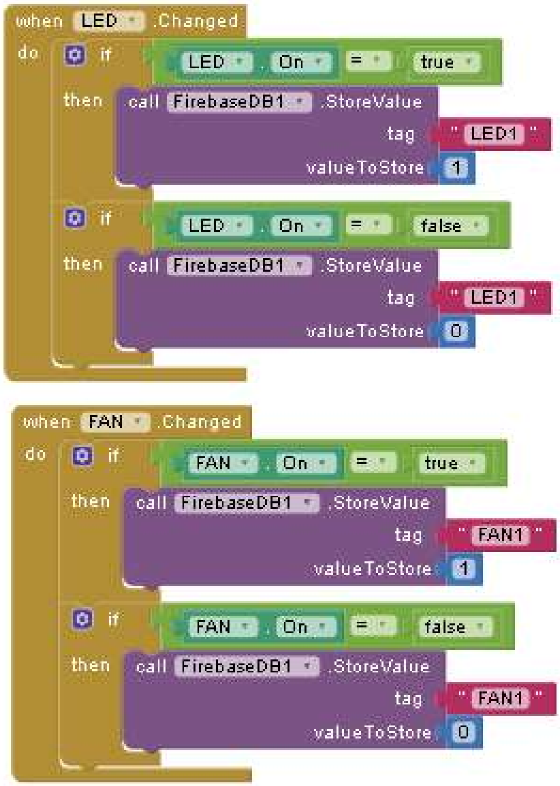


Preliminary Design

Software Design:

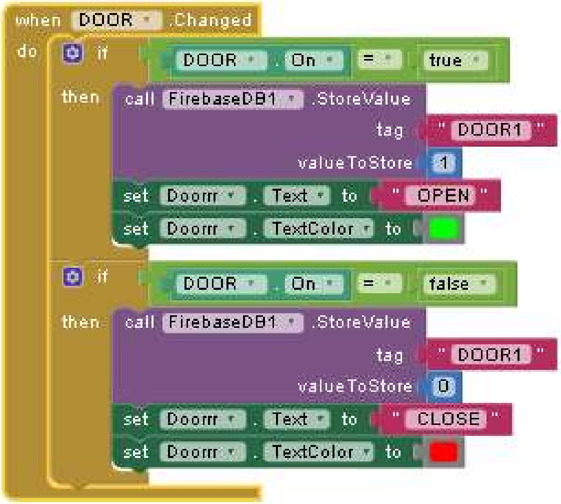


Mobile App GUI



```
when LED .Changed
do  if   LED . On = true
    then call FirebaseDB1 .StoreValue
                          tag  " LED1 "
                  valueToStore 1
    if   LED . On = false
    then call FirebaseDB1 .StoreValue
                          tag  " LED1 "
                  valueToStore 0

when FAN .Changed
do  if   FAN . On = true
    then call FirebaseDB1 .StoreValue
                          tag  " FAN1 "
                  valueToStore 1
    if   FAN . On = false
    then call FirebaseDB1 .StoreValue
                          tag  " FAN1 "
                  valueToStore 0
```

Mobile App Backend

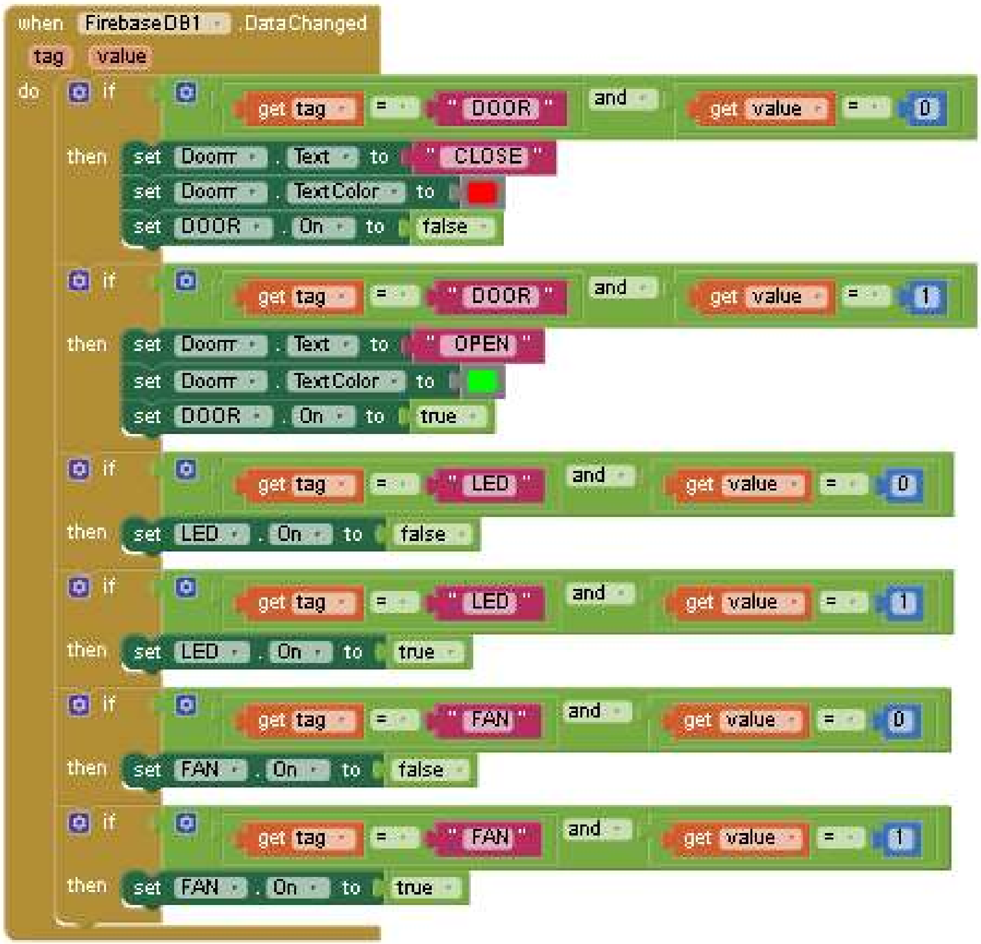


```
when DOOR .Changed
do  if   DOOR . On = true
    then call FirebaseDB1 .StoreValue
                          tag  " DOOR1 "
                  valueToStore 1
         set Doorr . Text to " OPEN "
         set Doorr . TextColor to 🟩
    if   DOOR . On = false
    then call FirebaseDB1 .StoreValue
                          tag  " DOOR1 "
                  valueToStore 0
         set Doorr . Text to " CLOSE "
         set Doorr . TextColor to 🟥
```

Mobile App Backend

Preliminary Design

Firbase mobile app logics:



```
when FirebaseDB1 .DataChanged
  tag  value
do  if  get tag = " DOOR " and get value = 0
    then  set Doorr . Text to " CLOSE "
          set Doorr . TextColor to 🟥
          set DOOR . On to false
    if  get tag = " DOOR " and get value = 1
    then  set Doorr . Text to " OPEN "
          set Doorr . TextColor to 🟩
          set DOOR . On to true
    if  get tag = " LED " and get value = 0
    then  set LED . On to false
    if  get tag = " LED " and get value = 1
    then  set LED . On to true
    if  get tag = " FAN " and get value = 0
    then  set FAN . On to false
    if  get tag = " FAN " and get value = 1
    then  set FAN . On to true
```

Mobile App Backend

Design Description

Connections with NodeMCU:

First set setup the NodeMCU in the BreadBoard and then we will make the connections:

1. LED light : Connected with D0 pin
2. Relay : Connected with D1 pin
3. Futher Dc motor is connected with relay
4. Servo Motor : Connected with D2
5. Ultrasonic triggerPin : Connected with D5
6. Ultrasonic triggerEcho : Connected with D6
7. VCC : All Required Vcc's are connected with the Vcc of nodemcu
8. Ground: : All Required gnd's are connected with the Vcc of nodemcu

Procedure:

After making all the connections very precisely then we make the logic's of source code, In the source code we first we include all the libraries for the components we are using i.e: <FirebaseESP8266.h> , <Servo.h> etc . After that we set set up our NodeMCU with FireBase in order to read or write to the firebase and also connect it up with the wifi. After that we define all the input and output GPIO pins that we are going to use in our design. After setting up with these things we further moved on to the code. We made the logic of code in such a way that in microcontroller loop first wrote the program for the ultrasonic sensor that is giving us the distance in variable after getting that distance we check weather the distance is greator or less then o a threshold we set up. If the distance is less then threshold then we say the person is detected and we turn on the door,led, fan for some time and send data to the Firebase, From firbase the app get updated and show the status of led fan and door on the mobile application.

If the distance is less greater then threshold then we go to the else satement where we are continously checking weather on a Firebase the data is updated by the user from mobile application or not , if the user update the status of Led,Fan or Door from the Mobile app then send it to Firebase and then Nodemcu will read it and update according to the user.

Source Code

```
#include <ESP8266WiFi.h>
#include <FirebaseESP8266.h>
#include <Servo.h>
Servo servo;

#define FIREBASE_HOST "iotlab-d3c41-default-rtdb.firebaseio.com"
                                   //Your Firebase Project URL goes here without "http:" , "\" and "/"
#define FIREBASE_AUTH "mQOnkPZWV1Irz62vpjuSztFYtx7K3Krf09YzajCF"
                                    //Your Firebase Database Secret  goes here

#define WIFI_SSID "PTCL-BB"                      //WiFi SSID to which you want NodeMCU to connect
#define WIFI_PASSWORD "00001111"               //Password of your wifi network

// Declare the Firebase Data object in the global scope
FirebaseData firebaseData;

  int val1=1;
  int val0=0;
String firedoor,fireled,firefan;
const int trigPin = 14;  //D5
const int echoPin = 12;  //D6

long duration;
int distance;
void setup()
{
pinMode (16, OUTPUT);        //LED D0
pinMode (5, OUTPUT);        //FAN D1
//pinMode (4, OUTPUT);       //DOOR D2
pinMode(trigPin, OUTPUT);  // Sets the trigPin as an Output
pinMode(echoPin, INPUT);  // Sets the echoPin as an Input

servo.attach(4);         //DOOR D2
servo.write(0);

   digitalWrite (16, LOW);
    digitalWrite (5, HIGH);

 Serial.begin(115200);                                   // Select the same baud rate if you want to see the datas
on Serial Monitor

  Serial.println("Serial communication started\n\n");

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);                                    //try to connect with wifi
  Serial.print("Connecting to ");
  Serial.print(WIFI_SSID);



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


  Serial.println();
  Serial.print("Connected to ");
  Serial.println(WIFI_SSID);
  Serial.print("IP Address is : ");
  Serial.println(WiFi.localIP());                                            //print local IP address
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);   // connect to firebase

  Firebase.reconnectWiFi(true);
  delay(1000);


      if ((Firebase.setInt(firebaseData, "/DOOR", val0)) && (Firebase.setInt(firebaseData, "/LED", val0)) &&
(Firebase.setInt(firebaseData, "/FAN", val0) ) )
         {    // On successful Write operation
               Serial.println("Strating...\n");
         }
}

void loop()
{

digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);

// Calculating the distance
distance= duration*0.034/2;
// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
```

Source Code

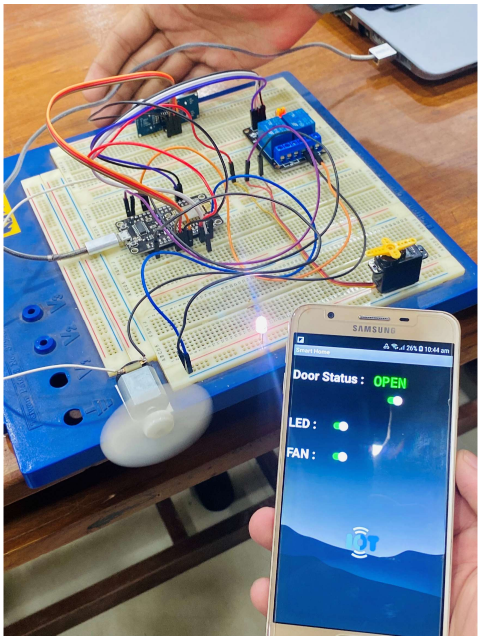
```
Serial.print(distance);
Serial.print("cm\n");
delay(400);


firedoor=Firebase.getString(firebaseData, F("/DOOR1"))? firebaseData.stringData() :
firebaseData.errorReason().c_str();
fireled=Firebase.getString(firebaseData, F("/LED1"))? firebaseData.stringData() :
firebaseData.errorReason().c_str();
firefan=Firebase.getString(firebaseData, F("/FAN1"))? firebaseData.stringData() :
firebaseData.errorReason().c_str();
;




if (distance < 15 )
{
         digitalWrite (16, HIGH);
         digitalWrite (5, LOW);
         servo.write(0);

         if ((Firebase.setInt(firebaseData, "/DOOR", val1))  && (Firebase.setInt(firebaseData, "/LED", val1)) &&
(Firebase.setInt(firebaseData, "/FAN", val1)) )
         {    // On successful Write operation, function returns 1
               Serial.println("\nPerson Detected");
               Serial.print("Updated data on FireBase\n\n");

         }

         delay(8000);

              if ((Firebase.setInt(firebaseData, "/DOOR", val0)) && (Firebase.setInt(firebaseData, "/LED", val0)) &&
(Firebase.setInt(firebaseData, "/FAN", val0) ) )
         {    // On successful Write operation, function returns 1
               Serial.println("Closing...");
         }

         digitalWrite (16, LOW);
         digitalWrite (5, HIGH);
         servo.write(180);

  }

  else
  {
```

Source Code

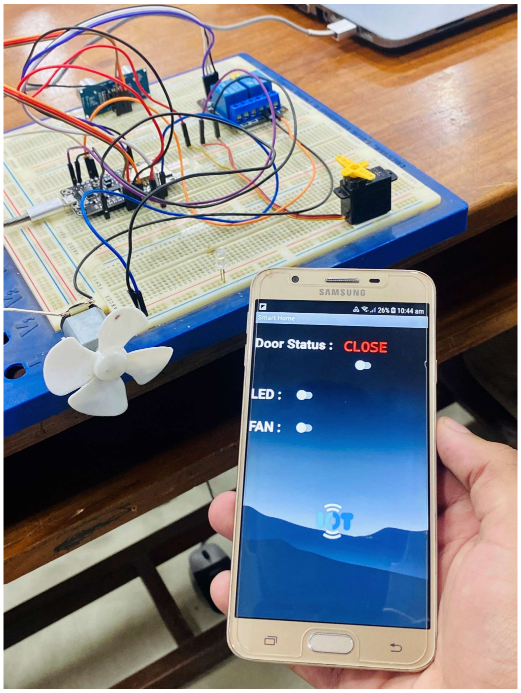
```
      if(firefan=="1")
      {
        digitalWrite (5, LOW);
      }

     if(fireled=="1" )
     {

        digitalWrite (16, HIGH);
     }
      if(fireled=="0")
      {
           digitalWrite (16, LOW);
      }

       if(firefan=="0")
      {
        digitalWrite (5, HIGH);
      }

          if(firedoor=="0")
      {
            servo.write(180);
      }

       if(firedoor=="1")
      {
         servo.write(0);
      }
    }
}
```

Experimental Results

Person Detection:



Controlling from Mobile Application:



Performance Analysis

We compared the performance and cost of four microcontroller boards (ESP8266). We find that the ESP8266 is the best performer. The Galileo board is too costly. Arduino Uno uses more current and voltage and also needs extra board for the extension.

This project presented is a low cost and flexible home control and monitoring system using Node MCU Board with internet and various sensors remotely controlled by Android OS smart phone. In this, Node MCU micro controller is used as an interface between user and hardware components. It is programmed and connected to several components according to the requirements. A mit app inventor is used as an application layer for communication between remote users and home devices, security systems. This entire system communication is enabled through internet. Notifications are sent to user through the app installed in smart phone.

Future Scope (Optional)

Smart Homes In the coming years, fully automated smart homes will surely become a reality as the home automation is developing rapidly. Due to good user convenience, smart homes are appealing a wide range of people all over the globe. The User can check for the electricity usage, the condition of his devices and get notification accordingly

More smartness can be added to this proposed project for making this smart home highly automated by using artificial intelligence. A camera can also be connected to micro controller so that suspect photograph can be taken and can be forwarded to the police if needed. Also voice call feature can be included to this system through which user can control the home appliances.

Today most of IoT prototyping is limited to remotely turning on/off the system’s components connected remotely from any web application or mobile phones using the wireless or wired networks. In future there are possibilities of making the system to work with artificial intelligence and include facilities of machine learning which will make the system more widely usable and more efficient. In future banks, hospitals, home, office and school automations will consider this technology as a necessary part of their infrastructure.

Conclusion

The main barrier towards the acceptance of home automation presently is its high cost. This report has studied and reviewed the presently available home automation system. These systems require additional network devices like hubs for their working, which in turn increase their cost. By the use of NodeMCU and the IoT platform, these devices can be made cost-effective. Above all, it will provide great user convenience as it will be possible to control the devices from a remote location. Using a web page or a mobile application, the system has been made platform independent. There is no need for any particular operating system so as to operate this system. The system will provide optimal results

[1] https://www.arduino.cc/en/software

[2] https://firebase.google.com/

[3] https://www.nodemcu.com/index_en.html

[4] https://appinventor.mit.edu/

[5] https://www.instructables.com/Interface-Relay-Module-With-NodeMCU/#:~:text=Relay%20is%20basically%20act%20a,based%20on%20NO%2DNC%20configuration.

[6] https://randomnerdtutorials.com/esp8266-nodemcu-firebase-realtime-database/