iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** IV **Month of publication:** April 2018

DOI: http://doi.org/10.22214/ijraset.2018.4637

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue IV, April 2018- Available at www.ijraset.com

IOT based Home Automation using Arduino and NODEMCU

V. Siva Parvathi[1], R. Padma[2]
[1, 2] *Assistant professor, Electrical and Electronics Engineering, Lakireddy Balireddy College of Engineering*

Abstract: ***Internet of Things (IOT) related applications have emerged as a significant field for both engineers and researchers, reflecting the magnitude and impact of data-related problems to be solved in contemporary business organizations especially in cloud computing. This paper reviews some of the technical opportunities offered and the technical challenges faced by the IoT in the house hold applications. Arduino Board is one of the significant objects in this ecosystem, where it is being used by many as an open source platform to build electronics projects. You need to understand that Arduino is a microcontroller, not a full-fledged computer, so you need to integrate the Arduino with wireless modules, motor drivers etc. for the suitable functioning of the project. For this, the IoT application should be directly loaded into the flash memory of the Arduino itself and also you should connect other wireless modules to transform it to an IoT device.***

In the advancement of technologies controlling and monitoring electrical appliances using laptop, computer with the help of internet connection is possible. So it gives a more space at a home, university and industrial controlling electrical appliances everywhere in the world. By using Internet of Things we can control many devices such as light, power plug, Fan, computer, security system and etc. It reduces human efforts and power efficiency. The main objective of internet of things is used to help specially challenged people and old age people to control electrical appliances and security purpose. IoT is very useful for these people in crucial situations. There are two ways to access these process WIFI connectivity (or) it is connected to a router. This process is done in low cost & controlling many devices in a simple circuit.

To the Arduino, an ultrasonic sensor that detects the movement of a person is used in unison with a Piezo buzzer. An effort is being made to use a GSM module with the Arduino. This will be used to send and receive messages. A message would be sent to the authorised number whenever an unwanted movement is detected by the sensor. On detection of an unauthorised movement, first the buzzer starts sounds alarm at a particular tone and then the message is sent to the user/owner. In addition to this, an effort would be made to use a Wi-Fi module also. This Wi-Fi module is used to connect to the Internet in order to send and receive the data. The received data is projected onto a cloud based server dashboard to detect flow of intruder in the system. Lastly, a cloud storage system called pCloud.com is used to store the information, documentation and research related to this project.

Keywords: Ardunio, IOT, Home automation, blynk application, Ardunio IDE and ESP

I. INTRODUCTION

Home automation has been a feature of science fiction writing for many years but has only become practical since the early 20th Century following the widespread outline of electricity into the home, and the hasty advancement of information technology. Home automation refers to the application of computer and information technology for control of home appliances simply. It is an automation of the home, household activity. Home automation may contain centralized control of Light, Appliances, Temperature and other systems, to provide improved convenience. Comfort, energy efficiency and safety. Home automation for the elderly and disabled can deliver increased quality of life for persons who might otherwise require caregivers or institutional care. The popularity of home automation has been increasing greatly in recent years due to much higher affordability and simplicity through Smartphone and tablet connectivity. The perception of the "Internet of Things" has tied in closely with the popularization of home automation.

It is a popular open-source single-board microcontroller, descendant of the open-source Wiring platform, designed to make the process of using electronics in multidisciplinary projects more accessible. The hardware contains of a simple open hardware design for the Arduino board with an Atmel AVR processor and on-board input/output support. The software consists of a standard programming language compiler and the boot loader that runs on the board. Arduino hardware is programmed using a Wiring-based language (syntax and libraries), similar to C++ with some slight simplifications and modifications, and a Processing-based integrated development environment. The proposed system does not require a dedicated server PC with respect to similar systems and offers a novel communication protocol to monitor and control the home environment with more than just the switching

©IJRASET (UGC Approved Journal): All Rights are Reserved

functionality. To demonstrate the feasibility and effectiveness of this system, devices such as light switches, power plug, temperature sensor and current sensor have been integrated with the proposed home control system.

IOT describes about concerning the physical devices with the sensors to the Internet, via wired or wireless networks. These sensors can use several types of local area connections like RFID, NFC, Wi-Fi, Bluetooth, and Zigbee. Sensors can also have extensive area connectivity such as GSM, GPRS, 3G, and LTE.

A. The three C's of IOT

1) *Communication:* The leading intension of IOT is to offer a communication among the physical devices, systems and people. Each domain desires the exchange of information in one way or the other. For instance, the medical domain the information about the patients, in some cases the critical information must be sent, so that an immediate action could be taken. The critical information in the form of either blood pressure or the pulse rate could be measured with the support of sensors. In case of transport domain, a vehicle can be traced, which necessitates the enabling of the location of the device. In all these cases the communication plays a vital role.
2) *Control and Automation:* In the connected world, the business and the customer have an option to control the devices, either directly or remotely. For example, a consumer can use IoT to unlock their car or start the washing machine. Similarly, IOT can be used to check the movement of individuals in a area. It can be done by configuring a sensor which can detect the movement, and this can be finished remotely i.e. automatically, by sitting in some other place.
3) *Cost:* IOT is appreciated for automating the things and this would decrease the cost of the overall project. With new sensor information, IOT can help a company save money by minimizing equipment failure and allowing the business to perform planned maintenance. Sensors can also measure the driving behaviour, life style parameters, which can be used to reduce the cost of fuel usage and suggest for a healthier living.

B. Structure of IOT

In this fast-paced world, the need for security-based systems has improved with time. Smart systems working automatically without human interference have create high demand. Such smart systems can be created with the help of IOT technology. IOT is an upcoming technology that makes use of Internet to control/monitor physical devices connected to the Internet. IOT gives the user the ability to control more than one digital thing easily through a comfortable GUI over the Internet.

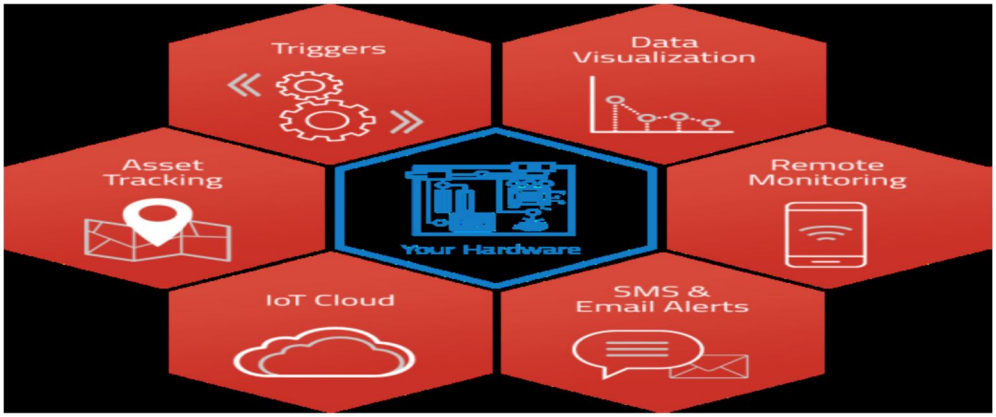


Fig 1 Structure of IOT

We realise that there are clearly assistances of IoT: several airlines have already improved significantly their operational efficiency by utilising Internet check-in, electronic boarding passes, RFID-enabled luggage handling, as well as e-enabled airport check-in and boarding. The implementation and deployment of smart devices is bound to improve their efficiency even further. Also, border control and airport security agencies can make use of these technologies to achieve a more precise and efficient screening process. From the passengers' perspective, improved convenience comes from reducing or even eliminating the need to carry and manage various pieces of documents, certificates and other sensitive assets. Although IoT will inevitably play a major role in improving upcoming air transportation, as it will in many other areas as well, there are critical issues to be recognized and considered in depth.

©IJRASET (UGC Approved Journal): All Rights are Reserved

Fig 2 IoT air travel scenario where various devices/cards/procedures are interconnected.

Unlike most of available home automation system in the market the proposed system is scalable that one server can achieve many hardware interface modules if it exists on Wi-Fi network coverage. System supports a wide range of home automation devices like power management components, and security components. The proposed system is better from the scalability and flexibility point of view than the commercially available home automation systems.

C. Flashing of "AT FIRMWARE" to ESP-01

Esp Wi-Fi module are work on firmware that was installed in it. We can create our own firmware and we can dump that firmware in Esp. But to work Esp with this AT commands we must flash the AT Firmware that was available on internet.

These are the steps we have to do in order to flash the Esp-01:

1) Firstly the Esp-01 does not having any UART so we cannot connect Esp directly to the computer.

a) For that we require Usb to TTL converter (FTDI). We must connect the Rx pin of FTDI is needing to be connected to the Tx pin of Esp. And Tx pin of FTDI is needing to be connected to the Rx pin of Esp.

b) Or, you can also use Arduino for flashing the esp. For that we must connect the Rx pin of Arduino is needing to be connected to the Tx pin of Esp. And Tx pin of Arduino is needing to be connected to the Rx pin of Esp. And make sure that a blank sketch is uploaded to your Arduino.

2) After connecting the Esp to the computer just download the AT firmware file and Esp flasher tool to flash the Esp-01.

3) Before flashing the AT firmware to the Esp you must ground the GPIO-0 pin of Esp. Then the Esp recognizes that the firmware is flashing and put Esp in flashing mode. Otherwise the firmware would not be installed.

II. PROPOSED SYSTEM

The system has two parts, namely; hardware and software. The hardware system contains of Arduino Uno board, Arduino wi-fi shield, sensors and home appliances. The software system consists of a java based android application also Arduino language is used to configure the arduino uno board and the sensors. In this system, the components used are Arduino uno board, arduino wi-fi shield, sensors (lm35, LDR). These hardware components are used to control the home appliances. Arduino uno board will help to develop an interface between the hardware and the software application. This system also consists of a software application which is developed using android. The arduino wi-fi shield will help in transmitting and receiving the input given by the user.

©IJRASET (UGC Approved Journal): All Rights are Reserved

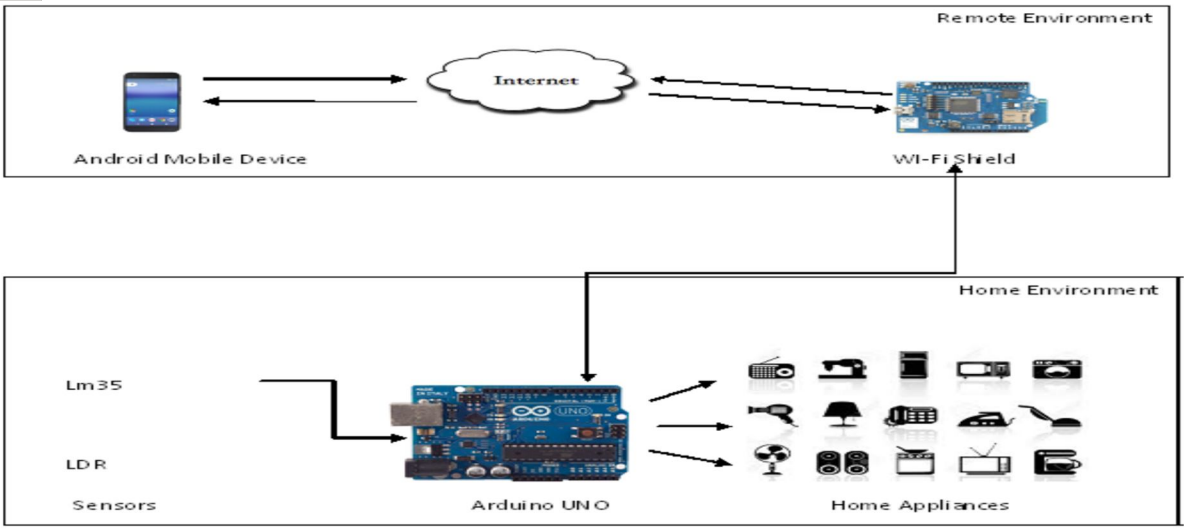


Fig 3 proposed system

Part1- The main path of appliance is controlling basic power using are android application which is made by us. The key part of Arduino is that it acts as interface between hardware component and software (application).
Part 2-After successfully implements of path 1 we can introduce one of significant aspect of project ARDUINO TEMPERATURE SENSOR LM35 which will check the room temperature and it is able adjust room temperature using the application. Plus light detection would play is role by adjusting light accordingly.
Part 3-Study of Light detection and temperature detection is carried out.
Part 4- Finally, Project will be compiled. Compilation of all the modules will be done. The entire system will be executed and will be able to run perfectly.

Table 1. Hardware and Software Requirements

HARDWARE REQUIREMENTS:	SOFTWARE REQUIREMENTS:
• "NODEMCU" OR "ARDUINO AND ESP8266" • TEMPERATURE AND HUMIDITY SENSOR (DHT11) • PIR MOTION SENSOR • RELAYS • RASPBERRY PI • BATTERIES • CONNECTING WIRE	• ARDUINO IDE • XAMPP • NOTEPAD++ • BLYNK APPLICATION

Fig: 4 Arduino UNO

©IJRASET (UGC Approved Journal): All Rights are Reserved

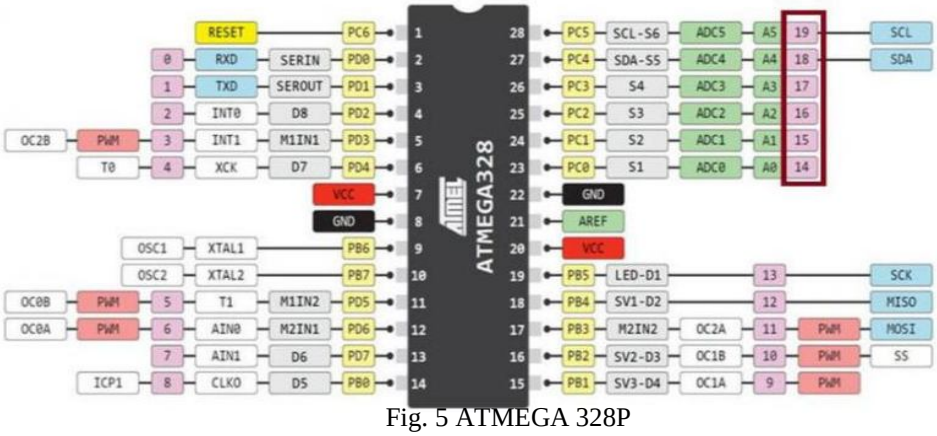


Fig. 5 ATMEGA 328P

Circuit Diagram for Nodemcu

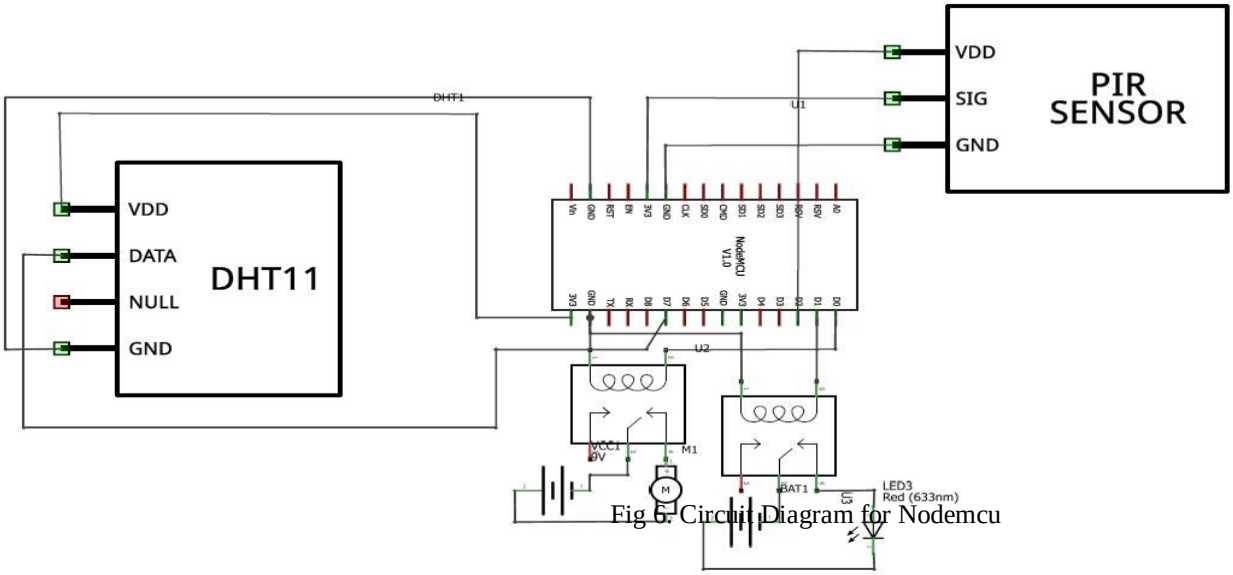


Fig 6. Circuit Diagram for Nodemcu

A. Program for proposed system:

Program for home automation using nodemcu

```
#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>
#include <ESP8266WiFiAP.h>
#include <ESP8266WiFiGeneric.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266WiFiScan.h>
#include <ESP8266WiFiSTA.h>
#include <ESP8266WiFiType.h>
#include <WiFiClientSecure.h>
#include <WiFiServer.h>
#include <WiFiUdp.h>
#define BLYNK_PRINT Serial
#include <DHT.h>
#include<ESP8266WiFi.h>
#include <Blynk.h>
```

```
#include <WiFiClient.h>
#include <ThingSpeak.h>
#include <BlynkSimpleEsp8266.h>
int fs, ls;
WiFiServer ser(80);
DHT dht(D7, DHT11);
int fan, lig = 0;
char ssid[] = "IoT";
char pass[] = "Lbrce@IoT";
char auth[] = "3a83ec4841574da9a5ce2165905a52e8";
unsigned long chn = 239268;
const char * api = "1EUHUMGUT60UK8DQ";
int calibrationTime = 30;
long unsigned int lowIn;
long unsigned int pause = 5000, tim = 0;
int stat;
boolean lockLow = true;
boolean takeLowTime;
WiFiClient c;
void setup() {
  pinMode(D0, OUTPUT);
  pinMode(D1, OUTPUT);
  pinMode(D2, INPUT);
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  WiFi.mode(WIFI_AP_STA);
  //WiFi.begin(ssid,pass);
  WiFi.softAP("BED ROOM", "lbrceiot@1");
  Serial.println("");
  Serial.println("CONNECTING TO THE WIFI PLEASE WAIT");
x:
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(200);
    Serial.print(".");
  }
  if (WiFi.status() == WL_CONNECTED)
  {
    Serial.print("\nCONNECTED TO THE WIFI\nYOU CAN ACCESS THROUGH ");
    Serial.print(WiFi.localIP());
    Serial.print(" or ");
    Serial.println(WiFi.softAPIP());
  }
  else
  {
    Serial.println("FAILED TO CONNECT RETRYING....");
    goto x;
  }
  ser.begin();
  Serial.println("SERVER IS STARTED");
```

©IJRASET (UGC Approved Journal): All Rights are Reserved

```
 ThingSpeak.begin(c);
 Serial.print("calibrating sensor ");
 //    for(int i = 0; i < calibrationTime; i++){
 //      Serial.print(".");
 //      delay(1000);
 //      }
 Serial.println(" Done");
 Serial.println("SENSOR ACTIVE");
 delay(50);
}
int hour = 0;
unsigned long prev = 0, pr = 0, g = 0;
void loop()
{
 Blynk.run();
 HTTPClient f, l, ti;
 c = ser.available();
 float tc = dht.readTemperature();
 float tf = dht.readTemperature(true);
 float h = dht.readHumidity();
 Serial.print(" Humidity:");
 Serial.println(h);
 Serial.print(" Temparature:");
 Serial.print(tc);
 Serial.print(",");
 Serial.println(tf);
 //f.begin("http://distinguishing-pens.000webhostapp.com/bedf.html");
 // l.begin("http://distinguishing-pens.000webhostapp.com/bedl.html");
 //f.GET();
 //l.GET();
 //  String fdata=f.getString();
 //  String ldata=l.getString();
 Blynk.virtualWrite(V1, h);
 Blynk.virtualWrite(V2, tc);
 Blynk.virtualWrite(V3, tf);
 if (stat==1)
 {
  if ((millis() - (tim * 1000)) >= 120000)
  {
    if(digitalRead(D2)==HIGH)
    {
       stat = 1;
       tim=millis()/1000;
    }
    else
       stat=0;
  }
 }
 if (digitalRead(D2) == HIGH && stat == 0)
 {
```

```
  digitalWrite(D0, HIGH);
  if (hour >= 18)
   digitalWrite(D1, HIGH);
  if (lockLow)
  {
   lockLow = false;
   Serial.println("---");
   Serial.print("motion detected at ");
   Serial.print(millis() / 1000);
   Serial.println(" sec");
   delay(50);
  }
  takeLowTime = true;
 }

 if (digitalRead(D2) == LOW && stat == 0) {
  digitalWrite(D0, LOW);
  digitalWrite(D1, LOW);
  if (takeLowTime) {
   lowIn = millis();          //save the time of the transition from high to LOW
   takeLowTime = false;       //make sure this is only done at the start of a LOW phase
  }
  if (!lockLow && millis() - lowIn > pause) {
   lockLow = true;
   Serial.print("motion ended at ");      //output
   Serial.print((millis() - pause) / 1000);
   Serial.println(" sec");
   delay(50);
  }
 }
 if ((millis() - prev * 1000) >= 120000)
 {
  ThingSpeak.setField(1, tc);
  Blynk.run();
  ThingSpeak.setField(2, tf);
  Blynk.run();
  ThingSpeak.setField(3, h);
  Blynk.run();
  ThingSpeak.writeFields(chn, api);
  Blynk.run();
  prev = millis() / 1000;
 }
 if ((millis() - pr * 1000 >= 120000) || pr == 0)
 {
  Blynk.run();
  ti.begin("http://api.thingspeak.com/apps/thinghttp/send_request?api_key=73AF7ZVL4M2BBQA
H");
  Blynk.run();
  ti.GET();
  String time = ti.getString();
```

©IJRASET (UGC Approved Journal): All Rights are Reserved

```
  for (int i = 0; i < time.length(); i++)
  {
   if (i == 0)
   {
    while (time[i] != ':')
    {
     hour = (time[i] - 48) + hour * 10;
     i++;
    }
   }
   if (time[i] == 'p' && hour != 12)
    hour = hour + 12;
  }
  Serial.println(hour);
  pr = millis() / 1000;
}
if (!c)
{
 return;
}
Serial.println("new client");
while (!c.available())
{
 delay(2);
}
String data = c.readStringUntil('\r');
Serial.println(data);
c.flush();
if (data.indexOf("/LIGHT=ON") != -1)
{
 digitalWrite(D1, HIGH);
 tim = millis() / 1000;
 stat = 1;
 lig = 1;
}
else if (data.indexOf("/LIGHT=OFF") != -1)
{
 digitalWrite(D1, LOW);
 stat = 0;
 lig = 0;
}
else if (data.indexOf("/FAN=ON") != -1)
{
 digitalWrite(D0, HIGH);
 tim = millis() / 1000;
 stat = 1;
 fan = 1;
}
else if (data.indexOf("/FAN=OFF") != -1)
{
```

©IJRASET (UGC Approved Journal): All Rights are Reserved

```
    digitalWrite(D0, LOW);
    stat = 0;
    fan = 0;
  }
  c.println("HTTP://1.1 200 OK");
  c.println("Content-Type:text/html");
  c.println("");
  c.println("<!DOCTYPE HTML>");
  c.println("<html>");
  c.println("<style>body{background-image:url(\"http://esp8266iot.orgfree.com/svet-cvet-linii-obem.jpg\")}</style><title>HOME AUTOMATION</title><link href=\'http://fonts.googleapis.com/css?family=Pacifico\' rel=\'stylesheet\' type=\'text/css\'><link rel=\"stylesheet\" href=\"http://esp8266iot.orgfree.com/style.css\">");
  c.println("<center><h1>WELCOME TO HOME AUTOMATION</h1></center><h2>CLICK THE RESPECTIVE BUTTONS FOR TURNING ON AND OFF</h2>");
  c.println("<center><h3>BUTTONS FOR FAN</h3><h3><a href=\"/FAN=ON\"class=\"action-button shadow animate green\">ON</a> <a href=\"/FAN=OFF\"class=\"action-button shadow animate red\">OFF</a></h3>");
  c.println("<br><h3>BUTTONS FOR LIGHT</h3><h3><a href=\"/LIGHT=ON\"class=\"action-button shadow animate green\">ON</a> <a href=\"/LIGHT=OFF\"class=\"action-button shadow animate red\">OFF</a></h3></center>");
  c.println("<h2>STATUS OF ROOM:");
  c.print("<h4>LIGHT IS ");
  if (digitalRead(D0) == HIGH)
    c.print("ON & ");
  else
    c.print("OFF & ");
  c.print("FAN IS ");
  if (digitalRead(D1) == HIGH)
    c.print("ON</h4>");
  else
    c.print("OFF</h4>");
  c.print("<h2>TEMPERATURE AND HUMIDITY OF THE ROOM</h2><h4>TEMPERATUR:");
  c.print(tc);
  c.print(" in celcius,");
  c.print(tf);
  c.println(" in farnhiet</h4>");
  c.print("<h4>HUMIDITY:");
  c.print(h);
  c.println("</h4>");
  c.println("</body>");
  c.println("</html>");
  delay(500);
}
BLYNK_WRITE(V0)
{
  fan = param.asInt();
  Serial.println(tim);
  if (fan == 1)
  {
```

©IJRASET (UGC Approved Journal): All Rights are Reserved

```
   digitalWrite(D0, HIGH);
   tim = millis() / 1000;
   stat = 1;
  }
 if (fan == 0)
  {
   digitalWrite(D0, LOW);
   stat = 0;
  }
}
BLYNK_WRITE(V4)
{

 lig = param.asInt();
 if (lig == 1)
 {
   digitalWrite(D1, HIGH);
   stat = 1;
   tim = millis() / 1000;
 }
 if (lig == 0)
 {
   digitalWrite(D1, LOW);
   stat = 0;
 }
}
```

B. Program for home automation using arduino and esp

```
#include<SoftwareSerial.h>
SoftwareSerial ser(2, 3);
void setup()
{
 Serial.begin(9600);
 ser.begin(9600);
 pinMode(11, OUTPUT);
 pinMode(4, OUTPUT);
 pinMode(5, OUTPUT);
 pinMode(6, OUTPUT);
 pinMode(7, OUTPUT);
 pinMode(8, OUTPUT);
 pinMode(9, OUTPUT);
 pinMode(10, OUTPUT);
 pinMode(A0, INPUT);
 pinMode(A1, INPUT);
 pinMode(A2, INPUT);
 pinMode(A3, INPUT);
digitalWrite(4, LOW);
digitalWrite(5, LOW);
digitalWrite(6, LOW);
digitalWrite(7, LOW);
```

©IJRASET (UGC Approved Journal): All Rights are Reserved

```
digitalWrite(8, LOW);
digitalWrite(9, LOW);
digitalWrite(10, LOW);
digitalWrite(11, LOW);
  sen("AT+CIFSR\r\n", 1000);
  sen("AT+CIPMUX=1\r\n", 1000);
  sen("AT+CIPSERVER=1,80\r\n",1000);
  ser.write("AT+CIPSTO=1000\r\n");
  delay(1000);
  sen("AT+CIPSTO?\r\n",1000);
}

void loop()
{
  int c = 0, p = 0;
  String cmd="AT+CIPSEND=";
  if (ser.available())
  {
    if (ser.find("+IPD,"))
    {
      c = ser.read() - 48;
      ser.find("pin=");
      {
      delay(1000);
      p = (ser.read() - 48) * 10;
      p += (ser.read() - 48);
      switch (p)
      {
        case 4: digitalWrite(4, HIGH);
          break;
        case 5:  digitalWrite(5, HIGH);
          break;
        case 6:  digitalWrite(6, HIGH);
          break;
        case 7: digitalWrite(7, HIGH);
          break;
        case 8: digitalWrite(8, HIGH);
          break;
        case 9:digitalWrite(9, HIGH);
          break;
        case 10: digitalWrite(10, HIGH);
         break;
        case 11:digitalWrite(11, HIGH);
          break;
        case 34: digitalWrite(4, LOW);
          break;
        case 35:digitalWrite(5, LOW);
          break;
        case 36:digitalWrite(6, LOW);
          break;
```

©IJRASET (UGC Approved Journal): All Rights are Reserved

```
case 37:digitalWrite(7, LOW);
  break;
case 38: digitalWrite(8, LOW);
  break;
case 39:digitalWrite(9, LOW);
  break;
case 140:digitalWrite(10, LOW);
  break;
case 41:digitalWrite(11, LOW);
  break;
case 20: cmd+=String(c);
        cmd+=",30\r\n";
        sen(cmd,1000);
        if(digitalRead(4)==HIGH &&digitalRead(5)==HIGH)
         ser.write("FAN IS ON LIGHT IS ON\r\n");
        else if(digitalRead(4)==HIGH &&digitalRead(5)==LOW)
         ser.write("FAN IS ON LIGHT IS OFF\r\n");
         else if(digitalRead(4)==LOW &&digitalRead(5)==HIGH)
         ser.write("FAN IS OFF LIGHT IS ON\r\n");
         else if(digitalRead(4)==LOW &&digitalRead(5)==LOW)
         ser.write("FAN IS OFF LIGHT IS OFF\r\n");
         ser.write("         ");
         delay(1000);
         cmd="AT+CIPCLOSE=";
         cmd+=String(c);
         cmd+="\r\n";
         sen(cmd,1000);
        break;
case 21: cmd+=String(c);
        cmd+=",30\r\n";
        sen(cmd,1000);
        if(digitalRead(6)==HIGH &&digitalRead(7)==HIGH)
         sen("FAN IS ON LIGHT IS ON       \r\n",1000);
         else if(digitalRead(6)==HIGH &&digitalRead(7)==LOW)
         sen("FAN IS ON LIGHT IS OFF        \r\n",1000);
        else if(digitalRead(6)==LOW &&digitalRead(7)==HIGH)
         sen("FAN IS OFF LIGHT IS ON       \r\n",1000);
         else if(digitalRead(6)==LOW &&digitalRead(7)==LOW)
         sen("FAN IS OFF LIGHT IS OFF      \r\n",1000);
         delay(1000);
         cmd="AT+CIPCLOSE=";
         cmd+=String(c);
         cmd+="\r\n";
         sen(cmd,1000);
  break;
case 22: cmd+=String(c);
        cmd+=",30\r\n";
        sen(cmd,1000);
        if(digitalRead(8)==HIGH &&digitalRead(9)==HIGH)
         ser.write("FAN IS ON\nLIGHT IS ON\r\n");
```

```
            else if(digitalRead(8)==HIGH &&digitalRead(9)==LOW)
             ser.write("FAN IS ON\nLIGHT IS OFF\r\n");
             else if(digitalRead(8)==LOW &&digitalRead(9)==HIGH)
             ser.write("FAN IS OFF\nLIGHT IS ON\r\n");
             else if(digitalRead(8)==LOW &&digitalRead(9)==LOW)
             ser.write("FAN IS OFF\nLIGHT IS OFF\r\n");
             delay(5000);
             cmd="AT+CIPCLOSE=";
             cmd+=String(c);
             sen(cmd,1000);
             break;
      case 23: cmd+=String(c);
            cmd+=",30\r\n";
            sen(cmd,1000);
            if(digitalRead(10)==HIGH &&digitalRead(11)==HIGH)
             ser.write("FAN IS ON LIGHT IS ON\r\n");
            else if(digitalRead(10)==HIGH &&digitalRead(11)==LOW)
             ser.write("FAN IS ON LIGHT IS OFF\r\n");
             else if(digitalRead(10)==LOW &&digitalRead(11)==HIGH)
             ser.write("FAN IS OFF LIGHT IS ON\r\n");
             else if(digitalRead(10)==LOW &&digitalRead(11)==LOW)
             ser.write("FAN IS OFF LIGHT IS OFF\r\n");
             delay(5000);
             cmd="AT+CIPCLOSE=";
             cmd+=String(c);
             sen(cmd,1000);
            break;
    }

    Serial.println(p);
    }
   }
  }

}

String sen(String command, const int timeout)
{
 String response = "";

 ser.print(command);

 long int time = millis();

 while ( (time + timeout) > millis())
 {
  while (ser.available())
  {
   char c = ser.read();
   response += c;
```

```
  }
 }
 Serial.print(response);
 return response;
}
```

C. Manual Sending Of At Commands To Esp-01

After installing the Firmware let's check whether this AT firmware is working or not by using manual testing. Now open the Arduino Ide software and upload blank sketch to Arduino. Now connect Rx pin of Arduino to the o the Tx pin of Esp. And Tx pin of Arduino to the Rx pin of Esp. So, the serial monitor can directly communicate with the Esp module. Now by entering the AT commands we can get the responses from Esp. The figure exposed below represents the responses of Esp to the AT commands. By using Arduino Ide or putty you can send this commands to esp-01. And the Esp module will responds to the commands as shown in below figures.

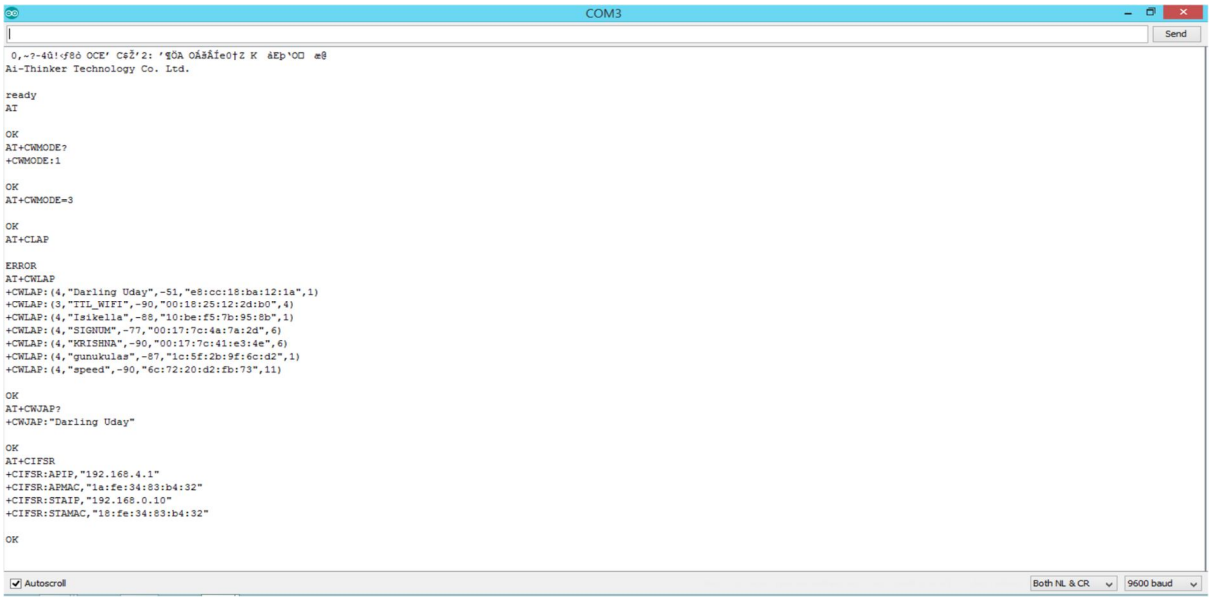


Fig 7 AT commands response from Esp

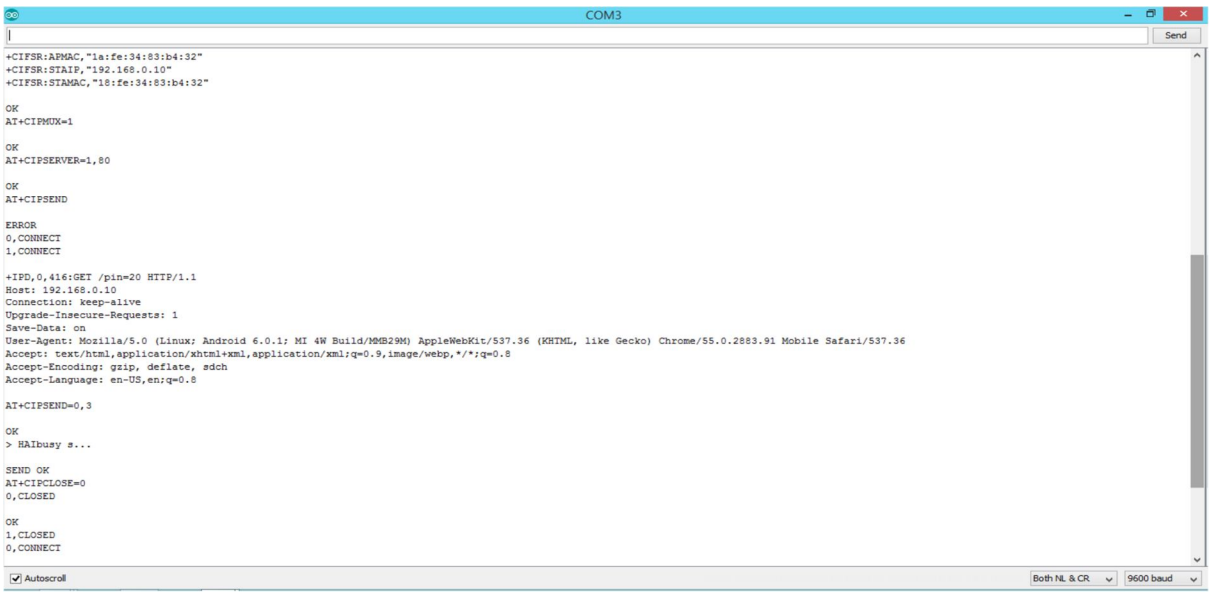


Fig 8 Sending data to client

©IJRASET (UGC Approved Journal): All Rights are Reserved

D. Receiving Data From Clients

As I said before Esp can works in 3 modes

1) STATION MODE
2) ACCESS POINT MODE
3) BOTH

a) In Station mode it will generates the wifi and can receive the data from the clients who are connected to the Esp'sWifi.
b) In client mode the Esp needs to be connected to external wifi through which the communication happens.
c) In 3rd mode we can generate as well as connect to another Wifi which is the special feature of Esp. So that we can connect to any of the wifi network and send the data to Esp.

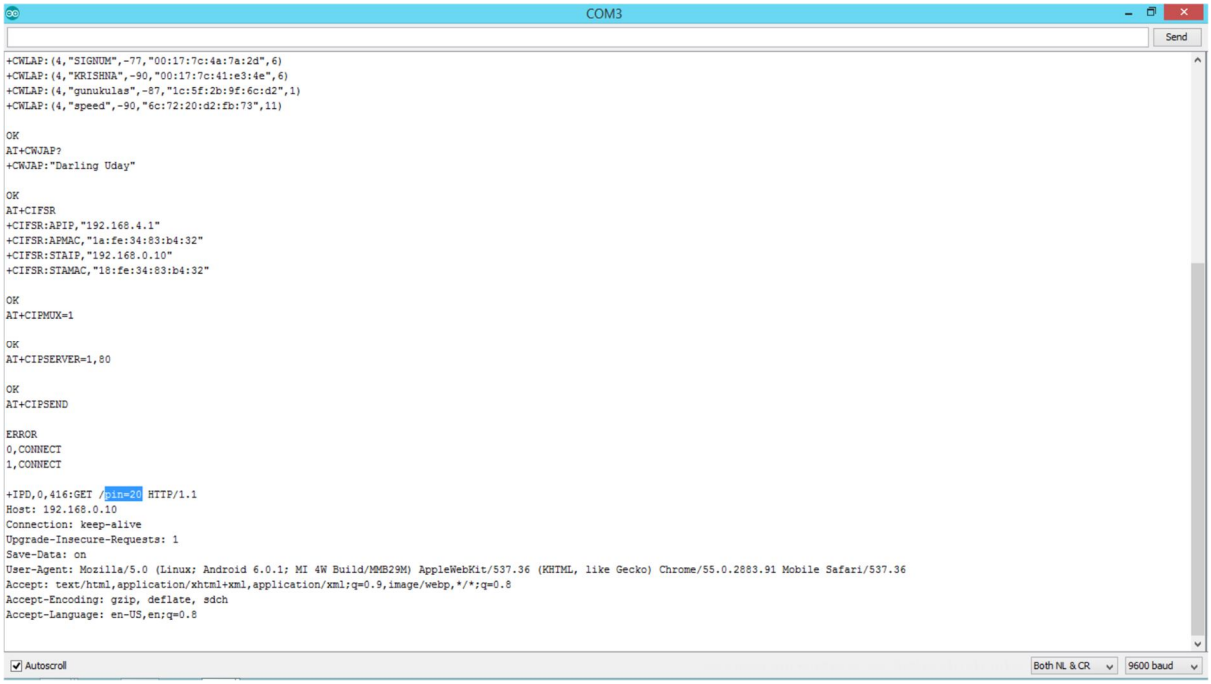


Fig 9 Esp received data from client

E. How can we send the data to Esp module?

It's very simple we can connect our mobile or laptop to Esp'swifi or to the Wi-Fi that Esp is connected. So that it will have an IP address with this IP address we can communicate to Esp. For sending data to Esp we must enter the data in search bar of browser like this "IP address/data". For example, I must send the data as "HAI" to Esp then I need enter like this
192.168.41. /HAI //in url box of browser then it will send HAI to Esp.

1) 192.168.4.1 is the default IP address of Station mode (Esp'sWifi).
2) Whatever the data send by client was received by Esp along with the channel number of the client. And it will be like +IPD, channel_numebr, no. of characters send: Receiving Method (Either GET or POST methods) / Data that was received

Ex:

+IPD,0,416:GET /HAI HTTP/1.1
Host:192.168.4.1
Connection: keep-alive……..
You can see the figure for more details. And in such a way we can receive the data from clients.

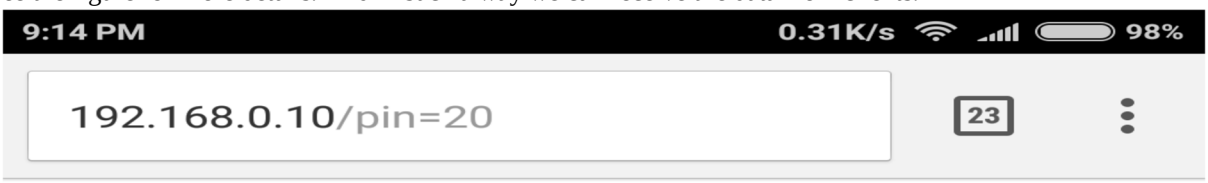


Fig 10 Sending data from mobile to Esp

©IJRASET (UGC Approved Journal): All Rights are Reserved

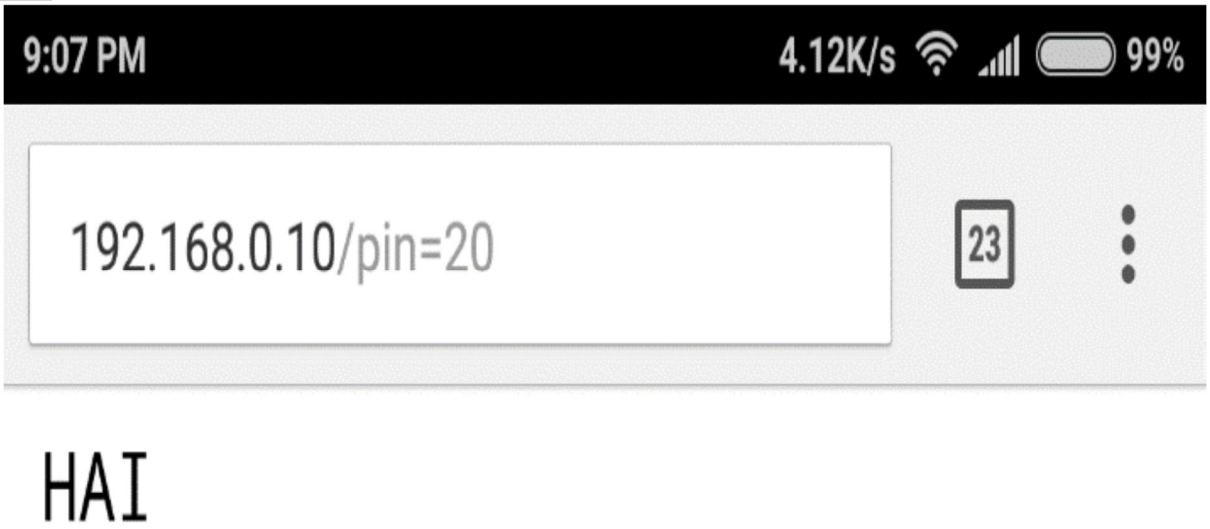


Fig 11 Receiving data from Esp to mobile

F. Automation The Esp By Using Arduino

With the help of Arduino, we can make sending of AT commands automatic and can receive the data from Esp and analyse the data and we can process the things according to the data received.

So, with the help of Esp we can make the Arduino work with internet. And we can communicate with the internet world with help of Esp through Arduino. With the help of Arduino whatever the things that was done manually was done automatically so that there is no need of man for giving AT commands for that we have to connect our Esp to Arduino and program Arduino according to that.

G. Linking The Devices To Internet

1) Introduction To Blynk

a) How Blynk Works: Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, visualize it and do many other cool things.

There are three major components in the platform:

i. *Blynk App* - allows to you create amazing interfaces for your projects using various widgets we provide.

ii. *Blynk Server* - responsible for all the communications between the smartphone and hardware. You can use our Blynk Cloud or run your private Blynk server locally. It's open-source, could easily handle thousands of devices and can even be launched on a Raspberry Pi.

iii. *Blynk Libraries* - for all the popular hardware platforms - enable communication with the server and process all the incoming and outcoming commands.

b) Why do I need to Blynk?

At this point you might be thinking: "Ok, I want it. What do I need to get started?" – Just a couple of things, really:

i. Hardware. : An Arduino, Raspberry Pi, or a similar development kit.

ii. *Blynk works over the Internet.* This means that the hardware you choose should be able to connect to the internet. Some of the boards, like Arduino Uno will need an Ethernet or Wi-Fi Shield to communicate, others are already Internet-enabled: like the ESP8266, Raspberri Pi with Wi-Fi dongle, Particle Photon or SparkFunBlynk Board. But even if you don't have a shield, you can connect it over USB to your laptop or desktop (it's a bit more complicated for newbies, but we got you covered). What's cool, is that the list of hardware that works with Blynk is huge and will keep on growing.

iii. *A Smartphone*. The Blynk App is a well designed interface builder. It works on both iOS and Android, so no holywars here, ok?

2) Introduction to thingspeak: The Internet of Things(IoT) is a system of 'connected things'. The things generally comprise of an embedded operating system and an ability to communicate with the internet or with the neighbouring things. One of the key elements of a generic IoT system that bridges the various 'things' is an IoT service. An interesting implication from the 'things' comprising the IoT systems is that the things by themselves cannot do anything. At a bare minimum, they should have an

©IJRASET (UGC Approved Journal): All Rights are Reserved

ability to connect to other 'things'. But the real power of IoT is harnessed when the things connect to a 'service' either directly or via other 'things'. In such systems, the service plays the role of an invisible manager by providing capabilities ranging from simple data collection and monitoring to complex data analytics. The below diagram illustrates where an IoT service fits in an IoT ecosystem.

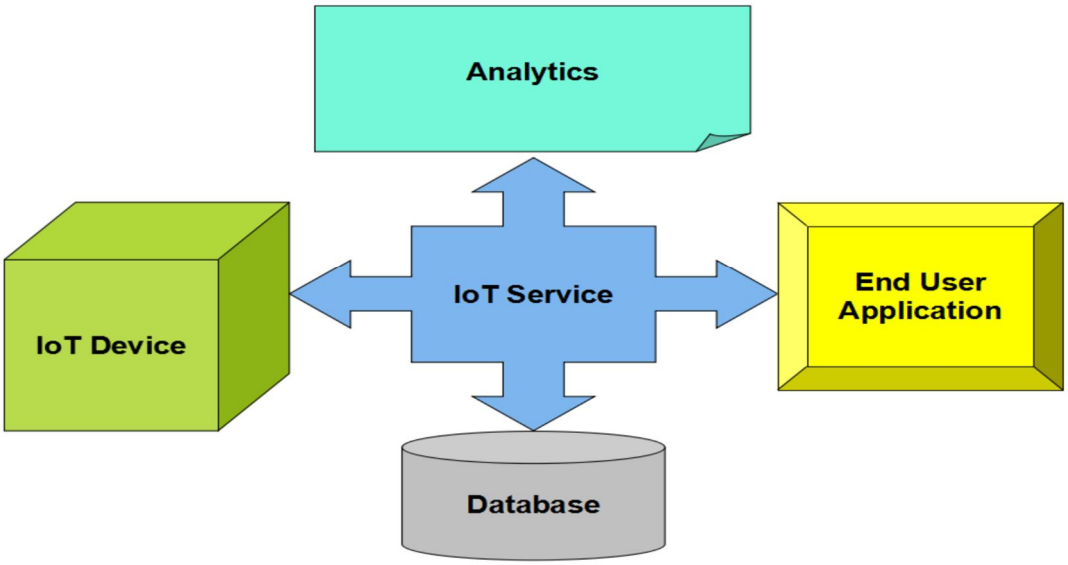


Fig. 12 Thingspeak Working

3) What is ThingSpeak

ThingSpeak is a platform providing various services exclusively targeted for building IoT applications. It offers the capabilities of real-time data collection, visualizing the collected data in the form of charts, ability to create plugins and apps for collaborating with web services, social network and other APIs. We will consider each of these features in detail below.

The core element of ThingSpeak is a 'ThingSpeak Channel'. A channel stores the data that we send to ThingSpeak and comprises of the below elements:

a) 8 fields for storing data of any type - These can be used to store the data from a sensor or from an embedded device.

3 location fields - Can be used to store the latitude, longitude and the elevation. These are very useful for tracking a moving device.

b) 1 status field - A short message to describe the data stored in the channel.

To use ThingSpeak, we need to sign up and create a channel. Once we have a channel, we can send the data, allow ThingSpeak to process it and also retrieve the same. Let us start exploring ThingSpeak by signing up and setting up a channel.

4) ThingSpeak Apps

ThingSpeak provides apps that allow us for an easier integration with the web services, social networks and other APIs. Below are some of the apps provided by ThingSpeak:

a) *Thing Tweet* - This allows you to post messages to twitter via ThingSpeak. This is a Twitter Proxy which re-directs your posts to twitter.

b) *Thing HTTP* - This allows you to connect to web services and supports GET, PUT, POST and DELETE methods of HTTP.

c) *Tweet Control* - Using this, you can monitor your Twitter feeds for a specific key word and then process the request. Once the specific keyword is found in the twitter feed, you can then use Thing HTTP to connect to a different web service or execute a specific action.

d) *React* - Send a tweet or trigger a Thing HTTP request when the Channel meets a certain condition.

e) *Talkback* - Use this app to queue up commands and then allow a device to act upon these queued commands.

f) *Time control* - Using this app, we can do a Thing Tweet, Thing HTTP or a Talkback at a specified time in the future. We can also use this to allow these actions to happen at a specified time throughout the week.

©IJRASET (UGC Approved Journal): All Rights are Reserved

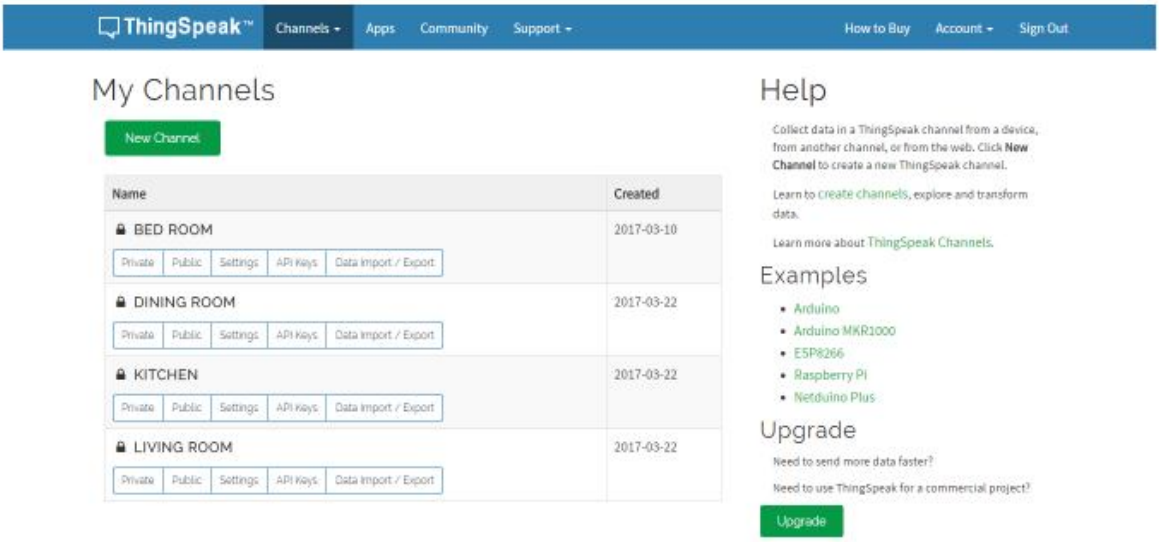


Fig. 13 Thingspeak Channels

III. SIMULATION AND RESULTS

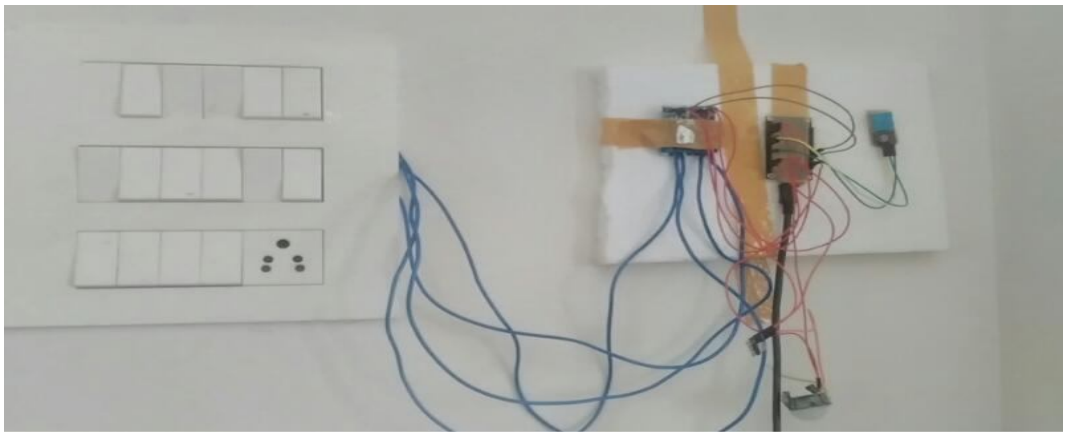


Fig 14 Proposed prototype using Nodemcu

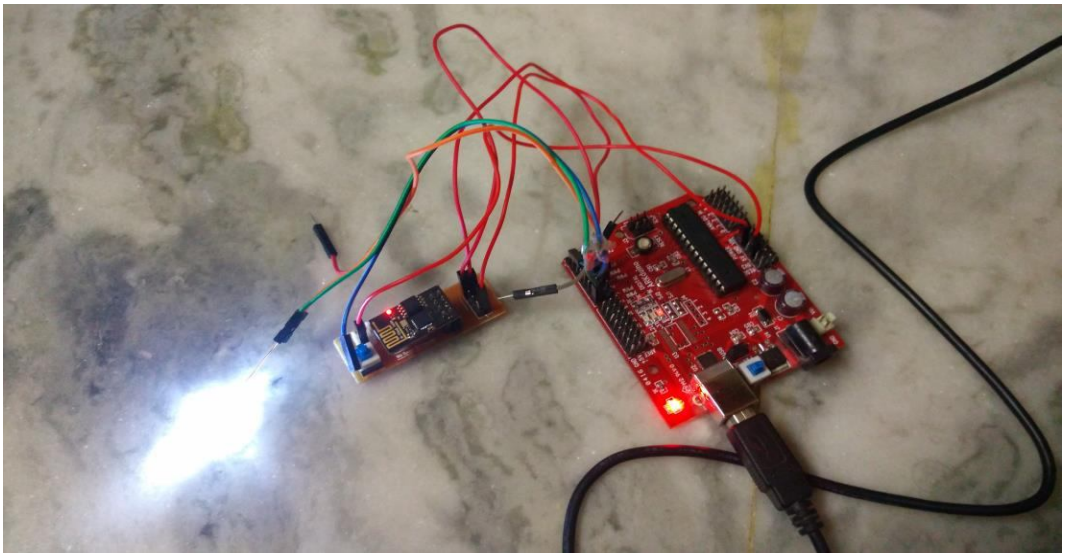


Fig 15 Proposed prototype using Arduino and Esp-01

©IJRASET (UGC Approved Journal): All Rights are Reserved

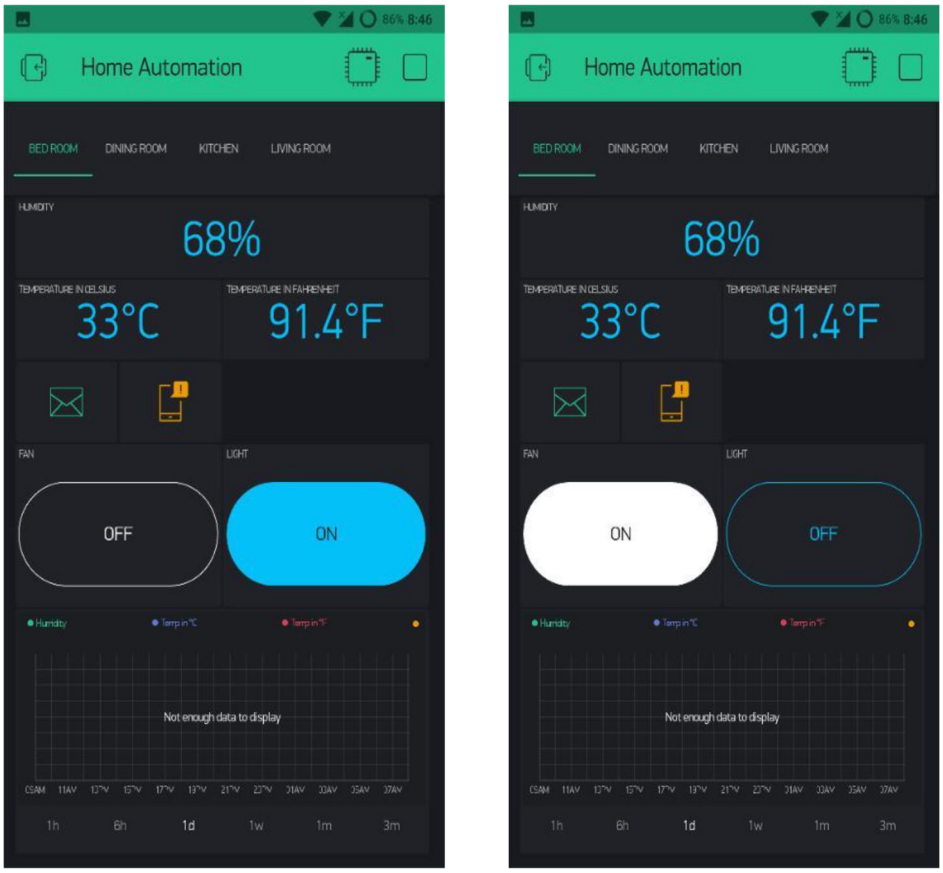


Fig. 16 (a) & (b) Blynk app for controlling the devices

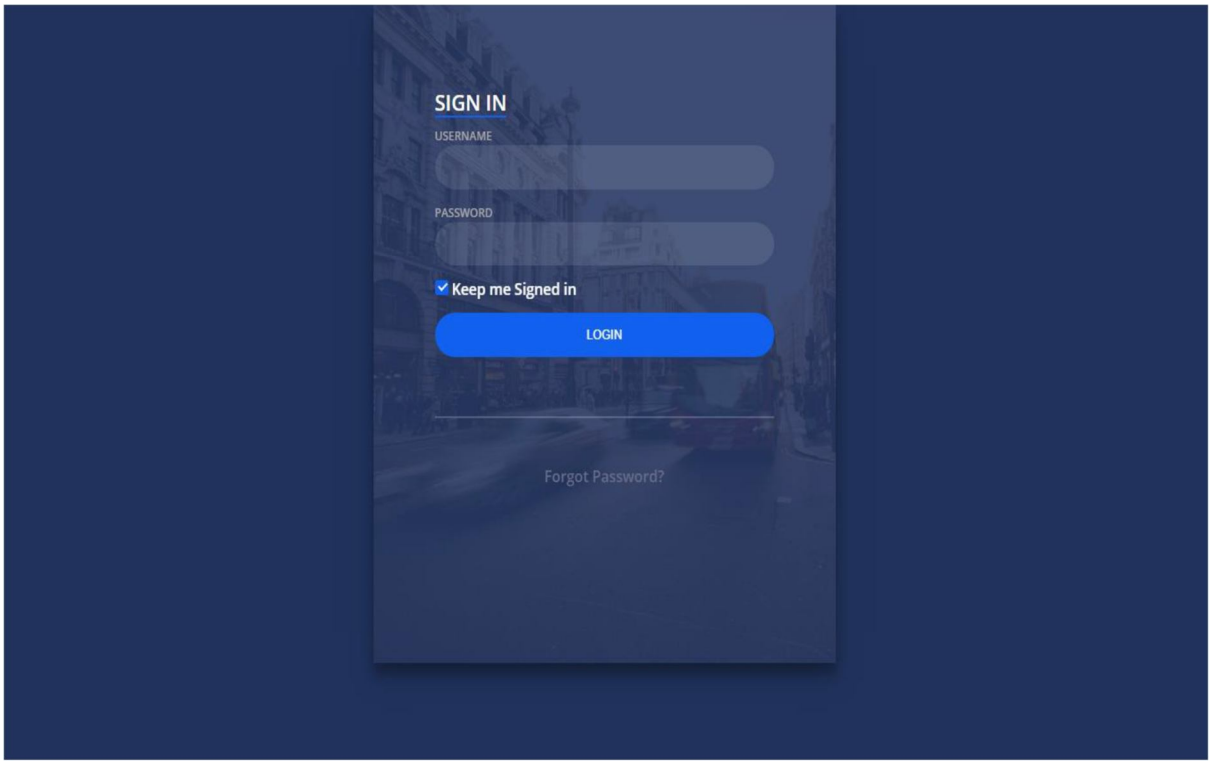


Fig.17 Login page for webpage

©IJRASET (UGC Approved Journal): All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue IV, April 2018- Available at www.ijraset.com

Fig. 18 Home page of web interface

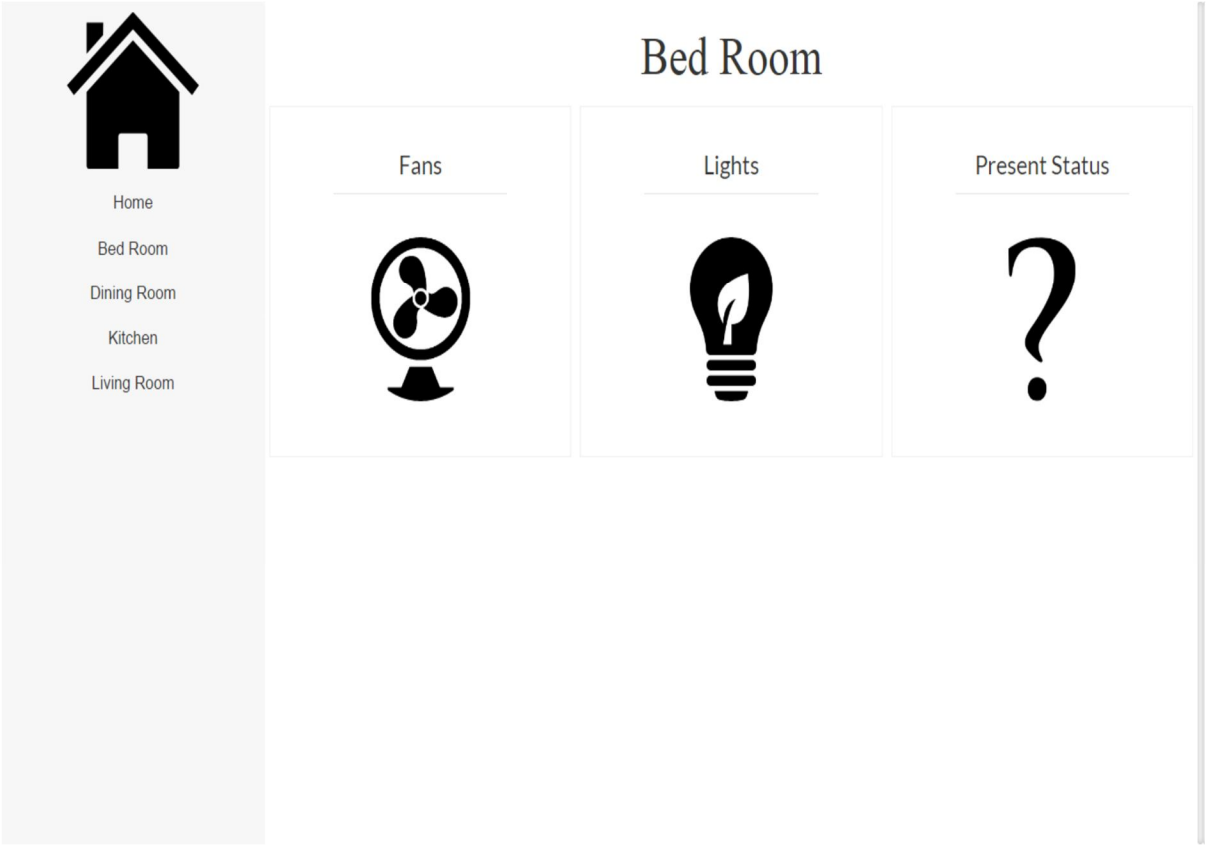


Fig. 19 Selection of devices in web page

©IJRASET (UGC Approved Journal): All Rights are Reserved

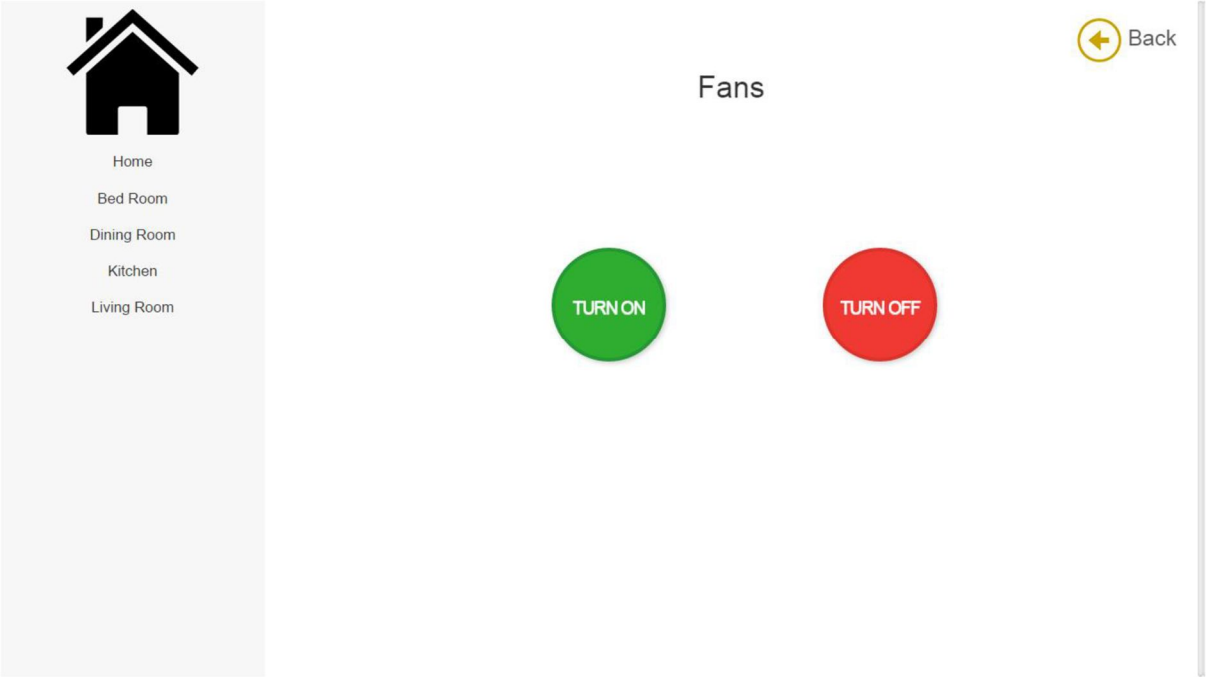


Fig. 20 ON & OFF Buttons for controlling in webpage

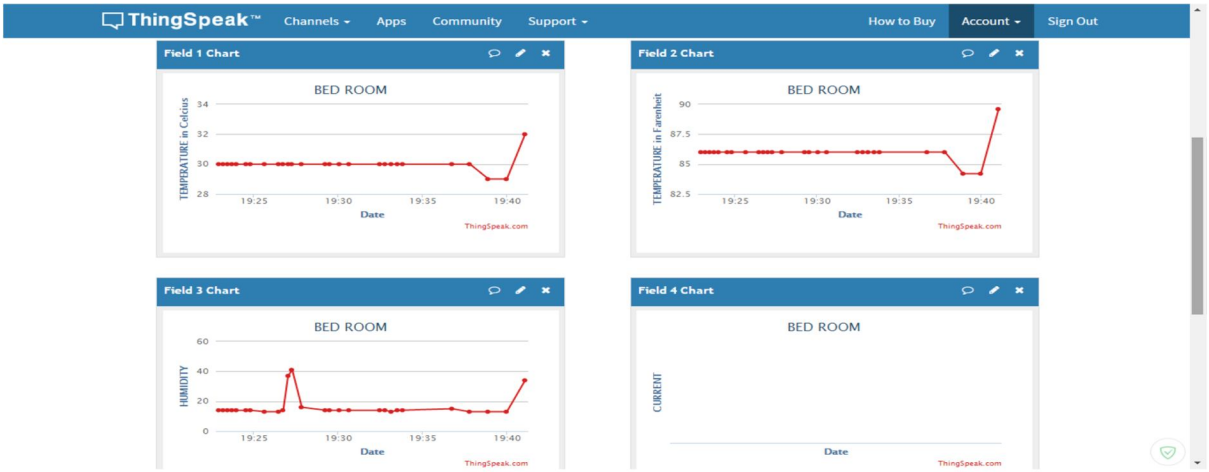


Fig. 21 Sensors Data in ThingSpeak

IV. CONCLUSION

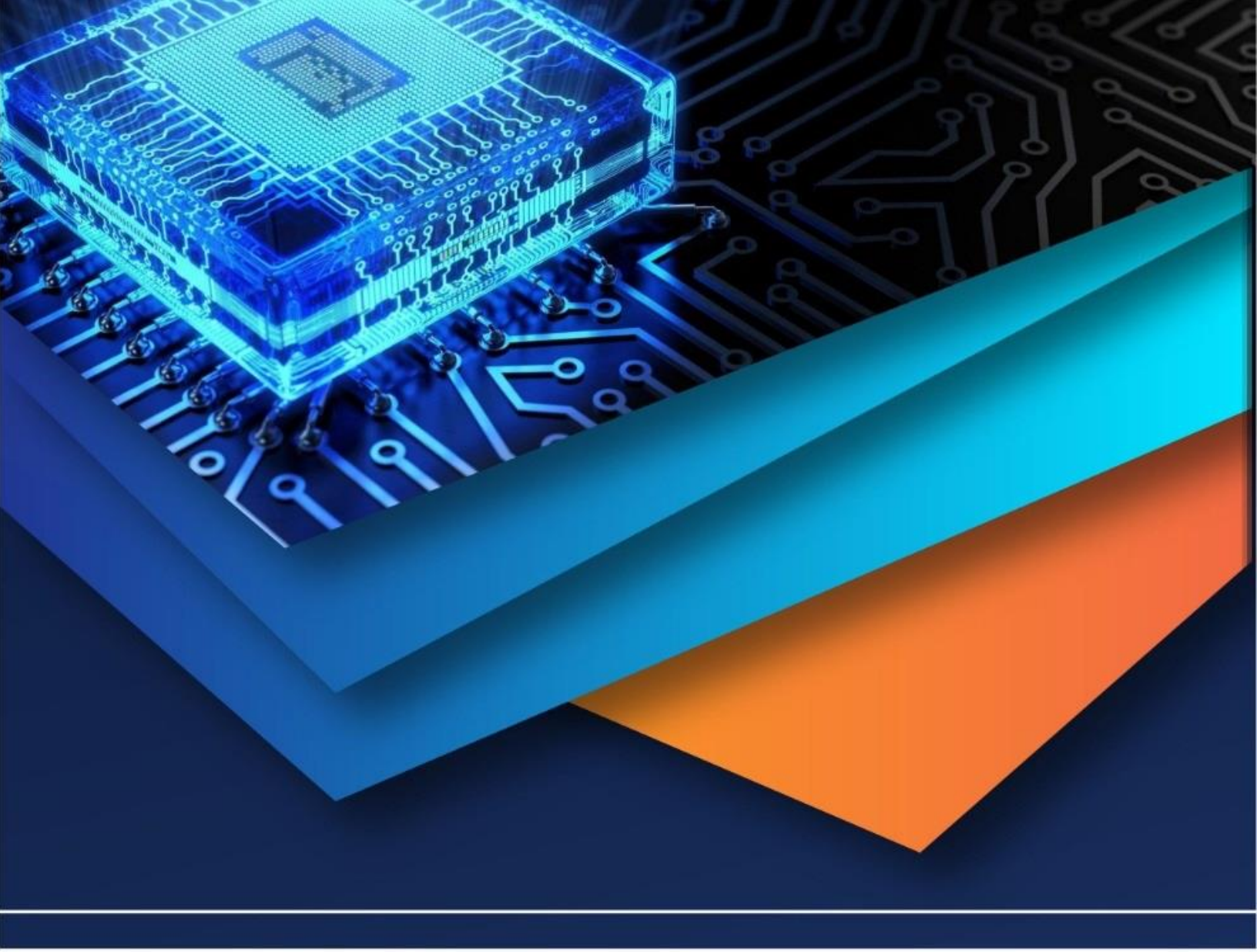
In this paper we focus on the process of operating or controlling several equipment, machinery, and other electrical and electronic appliances using various control systems remotely. This method of operating or controlling such applications is referred to as automation which has become an integral part of everyday life for human beings. The working model we designed has its focal point on home automation providing 100% efficiency. The model has its roots on an IOT platform that permits devices to synchronize with the IOT platform so that it can be controlled remotely. The platform uses the IOT technology to create a network among the main server and the other electrical and electronic appliances making home a smarter place to live in. The entire network consists of a single admin which makes our model a secure one as the admin only have the authority to access all the nodes present under each user. The model is quite economical as however there is only a single admin but the number of user under the admin may increase making a large complex network but a secure one. For future work we would try to increase the number of networks under a single server making a whole city automation using IOT.

©IJRASET (UGC Approved Journal): All Rights are Reserved

REFERENCES

[1] Ameetchauhan, Sangeetaagarwal "SMS based remote control System in International Journal of computer Science and Management, 2011.
[2] HOME AUTOMATION USING ATmega328 MICROCONTROLLER AND ANDROID APPLICATION, S. Anusha1, M. Madhavi2, R.Hemalatha3. International Research Journal of Engineering and Technology (IRJET) , Volume: 02 Issue: 06 | Sep-2015 www.irjet.net
[3] Internet of Things: Ubiquitous Home Control and Monitoring System using Android based Smart Phone, Rajeev Piyare. International Journal of Internet of Things 2013, 2(1): 5-11 DOI: 10.5923/j.ijit.20130201.02
[4] https://arduinohistory.github.io/ (here the history of arduino is given) 5.http://www.atmel.com/devices/atmega8.aspx
[5] Sirsath N. S, Dhole P. S, Mohire N. P, Naik S. C &Ratnaparkhi N.S Department of ComputerEngineering, 44, Vidyanagari, Parvati, Pune-411009,India University of Pune, "Home Automation usingCloud Network and Mobile Devices".
[6] Weimei Zhang, "Study about IOT's application in Digital Agriculture construction", Electrical and Control Engineering (ICECE) 2011 International Conference, pp. 2578-2581, 2011.

©IJRASET (UGC Approved Journal): All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)