A Cloud based System for Real-Time Remote Surveillance and Control based on the Wireless Multi-Sensor Network Inputs

Mr. Rohit Kumar Jhajharia[1], Mr. Dayal Chandra Sati[2], Dr. Vivek Kumar[3]

[1]M.Tech. Scholar, Dept. of Electronics & Communication Engineering, B.R.C.M.C.E.T., Bahal, Haryana, India
[2]Assistant Prof., Dept. of Electronics & Communication Engineering, B.R.C.M.C.E.T., Bahal, Haryana, India
[3]Head of Dept. of Electrical & Electronics Engineering, B.R.C.M.C.E.T., Bahal, Haryana, India

---

Abstract - *Here in this work efforts were made to exploit the potentials of a cloud based mobile app Blynk which is specifically designed for the IoT based applications. For this purpose, a NodeMCU based IoT system was designed and developed for real-time supervision of the measured outputs from multiple sensors and also enable the user to make decision accordingly to control the electrical load connected to it. As it was a wireless multiple sensor network, it utilizes the Wi-Fi local hotspot network as per the ssid and password credentials entered by the user in the firmware itself. The carefully designed hardware and the real-time supervision of measured sensor outputs as well as the relay on/off status observed over the Blynk App along with the real-time controlling of relays validated the work.*

Key Words: NodeMCU, WiFi, IoT, Cloud, ESP8266, Blynk, Wireless Multi Sensor Network, Surveillance, etc.

1. INTRODUCTION

1.1 Cloud

"The cloud" refers to servers that are accessed over the Internet and the software and databases that run on those servers. Cloud servers are located in data centers all over the world. By using cloud computing, users and companies don't have to manage physical servers themselves or run software applications on their own machines. The cloud enables users to access the same files and applications from almost any device, because the computing and storage takes place on servers in a data center, instead of locally on the user device. This is why a user can log into their Instagram account on a new phone after their old phone breaks and still find their old account in place, with all their photos, videos, and conversation history. For businesses, switching to cloud computing removes some IT costs and overhead: for instance, they no longer need to update and maintain their own servers, as the cloud vendor they are using will do that. This especially makes an impact for small businesses that may not have been able to afford their own internal infrastructure but can outsource their infrastructure needs affordably via the cloud. The cloud can also make it easier for companies to operate internationally, because employees and customers can access the same files and applications from any location.

1.2 Internet-of-Things (IoT)

The Internet of Things, or IoT, is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. An IoT ecosystem consists of web-enabled smart devices that use embedded systems, such as processors, sensors and communication hardware, to collect, send and act on data they acquire from their environments. IoT devices share the sensor data they collect by connecting to an IoT gateway or other edge device where data is either sent to the cloud to be analyzed or analyzed locally. Sometimes, these devices communicate with other related devices and act on the information they get from one another. The connectivity, networking and communication protocols used with these web-enabled devices largely depend on the specific IoT applications deployed.

1.3 NodeMCU

NodeMCU is a low-cost open source IoT platform. It is an open source firmware for which open source prototyping board designs are available. The name "NodeMCU" combines "node" and "MCU" (micro-controller unit). It runs on the ESP8266 Wi-Fi SoC (System on Chip) from Espressif Systems. Both the firmware and prototyping board designs are open source. ESP-12F Wi-Fi module is having a core processor ESP8266 and integrates ultra low power 32-bit MCU micro, with the 16-bit short mode, Clock speed support 80 MHz, 160 MHz, supports the RTOS, integrated Wi-Fi on-board antenna. The module supports standard IEEE802.11 b/g/n agreement, complete TCP/IP protocol stack. ESP8266EX offers a complete and self-contained Wi-Fi networking solution; it can be used to host the application or to offload Wi-Fi networking functions from another application processor.

1.4 Blynk

Blynk is a mobile platform with iOS or Android to control various microcontroller platforms like Arduino, Raspberry Pi, NodeMCU etc. by reading or writing bits wirelessly over the internet. Users can easily create the graphical interfaces for their projects by simply dragging and dropping widgets available in this app. It is a hardware-agnostic IoT platform with white-label mobile apps, private clouds, device management, data analytics, and machine learning.

2. OBJECTIVE

The objective of the study was to implement a cloud based system for the surveillance of certain parameters in a multisensory environment and that too remotely in real-time over the mobile phone using internet and it should also enable the user to control different electrical appliances remotely over the internet using the same cloud based application. In context of this statement a versatile system was required to establish communication between the different sensor nodes or clients and the central server node. The central node was expected to enable the user to remotely access the status of sensor parameters in real-time over the mobile using internet. The system was expected to acquire both types of signals analog as well as digital as per the sensor output. The aim was to collect the sensor data from the deployed units and send it to the cloud so that the user could access it remotely in real-time on an open source cloud platform based application. Wi-Fi protocol was to be utilized to establish communication between the hardware and the Mobile Application. A local hotspot was to be used to establish connectivity. Hardware prototype has validated the work.

3. PROBLEM FORMULATION

1. A survey was conducted to identify the potential environmental sensors and a high-performance, low-cost, IoT-integrated computational platform used in home automation systems
2. The detailed knowledge was gathered about the sensor specifications, their availability in market, operating principles, working procedures, driver circuits and also about their interfacing with microcontroller
3. To put efforts in selection of a type of IoT based monitoring application specifically to fulfill mobile-based surveillance and controlling needs
4. To study about establishing the wireless connection between multiple sensors and the cloud based mobile application via central node
5. To prepare the Bill-of-Material and make procurement for the required components from multiple sources
6. To finally implement the system over a self-designed PCB in a CAD tool and finally perform multiple iterations to test the system and calibrate it.

4. SYSTEM ARCHITECTURE

The implemented work was an IoT (Internet-of-Things) based home automation system to sense multiple environmental parameters using a multi-sensor network and accordingly make decision to control multiple electrical appliances over the internet using Wi-Fi protocol. The system design consisted of a central node i.e. NodeMCU to receive the measured analog as well as digital outputs from various sensors. Those measured values were processed by the NodeMCU and Blynk is a mobile platform with iOS or Android to control various microcontroller platforms like Arduino, Raspberry Pi, NodeMCU etc. by reading or writing bits wirelessly over the internet. Users can easily create the graphical interfaces for their projects by simply dragging and dropping widgets available in this app.

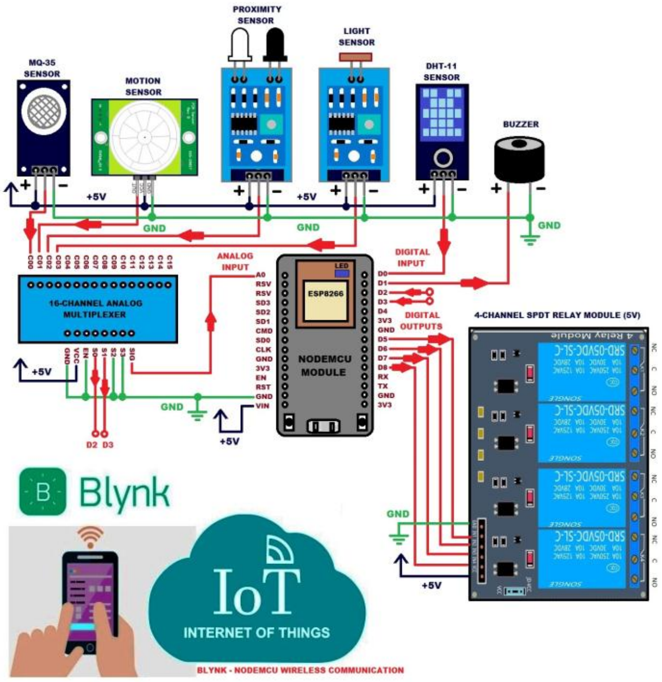

Fig.-1: System Architecture

5. METHODOLOGY ADOPTED

Here in this section the methodologies followed to design and implement this system using different hardware and software tools were discussed.

5.1 Hardware Prototyping

The hardware prototype board for the proposed system was designed in the EAGLE CAD tool. EAGLE is Easily Applicable Graphical Layout Editor by Autodesk. It is the worldwide most widely used PCB design tool. In this work methodology was adopted to design and validate the schematic first and then the layout was designed as per the schematic. Finally we used the PCB layout design to fabricate the PCB. Afterwards the hole drilling process was carried out followed by the component placement and soldering process. Finally we obtained the hardware prototype for the proposed system in the form of self designed PCB card. The schematic and board layout details could be accessed in the upcoming subsections.

5.2 Schematic Design

The schematic shown below consisted of a central node i.e. NodeMCU which is an integrated Microcontroller and ESP8266 Wi-Fi module and that supports most of the IoT based applications. There are two types of sensors interfaced to the NodeMCU depending upon their outputs. Some of those were the analog output sensors and others were the digital output sensors. As there were multiple digital input/output pins and only single analog input/output pin available on the NodeMCU board, we had interfaced a 16-Channel Analog Multiplexer to the analog pin of the NodeMCU. This was done to enable NodeMCU to receive the multiple sensor analog-output values one by one and process it. As it was a 16:1 multiplexer, there were four select lines (S3, S2, S1, S0) available for that purpose on-board to select each channel of the multiplexer. We had utilized just four channels of this multiplexer for four different analog output sensors. The digital output sensors and other digital devices were connected directly to the digital input/ output pins of the NodeMCU. The analog sensors used here were Light Dependent Resistor (LDR) module for sensing light intensity, MQ-5 module for sensing gas in the environment, Infrared Proximity sensor module for proximity sensing, Passive Infrared (PIR) sensor module for motion detection. The digital sensors here were DHT11 to sense humidity and temperature levels. The digital devices like buzzer and four-channel relay board were also interfaced to the NodeMCU with appropriate driver circuits. The whole system required +5volt dc power for its operation.

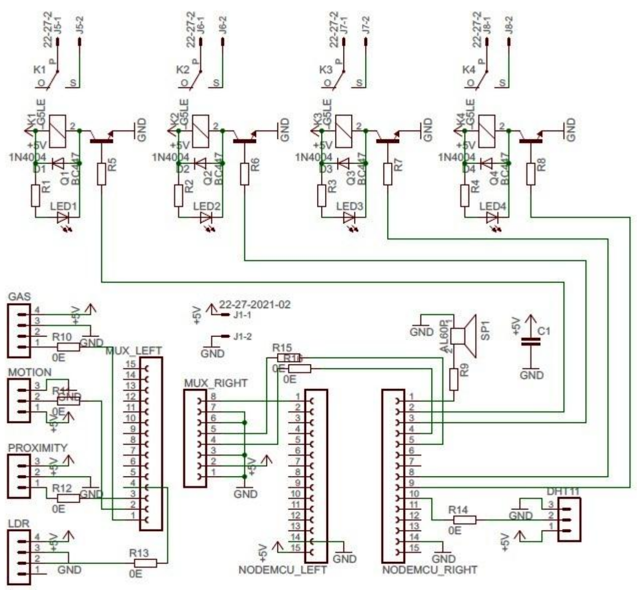

Fig.-2: System Architecture

5.3 Layout Design

The single layer pin-through-hole (PTH) PCB layout was designed in EAGLE. The layout editor took the reference of schematic design and generated component packages corresponding to each component chosen into the schematic design. Therefore, proper care was taken while choosing the components keeping in view of their package including parameters like type of package required (DIP, SMD, etc.) , pitch size, area, pad size, pad shape and drill hole, etc. The PCB tracks were designed in Bottom layer only. Component placement and routing is the most important aspect for a PCB design and there were other parameters too worth consideration like track size, track pitch, parasitic capacitance, track resistance, proper grounding, etc.

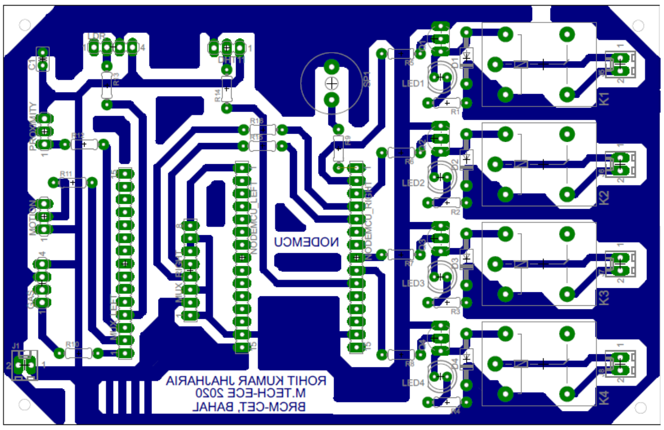

Fig.-3: System Architecture

Table-1: List of Components

Sr. No.	Component / Module	Specification	Quantity
1	NodeMCU Module	ESP-8266	1
2	IR Proximity Sensor Module	4-PIN	1
3	LDR Light Sensor Module	4-PIN	1
4	DHT11 Sensor Module	3-PIN	1
5	PIR Motion Sensor Module	3-PIN	1
6	MQ-5 Gas Sensor Module	4-PIN	1
7	Small Buzzer	+5V	1
8	CD74HC4067 Multiplexer 16:1	16 Analog Channels	1
9	SPDT Relay	+5V	4
10	IN4007 Diode	1A	4
11	Resistor 10K	Quarter Watt	6
12	Resistor 220E	Quarter Watt	6
13	Resistor 1K	Quarter Watt	4
14	Capacitor	2200uF/35V	1

	Electrolytic		
15	Capacitor Electrolytic	10uF/35V	1
16	Voltage Regulator IC	LM7805	1
17	Copper Clad	130 x 85 mm^2	1
18	Transformer	12-0-12 / 200mA	1
19	Connecting Cable	220V AC	1
20	Berg Strip	Male Connector	2
21	Berg Strip	Female Connector	2
22	Connector	2-PIN	4
23	LEDs	Red	4
24	NPN Transistor	BC-547	4

6. ALGORITHM

1. First of all before writing the firmware we had installed the required sensor libraries to the Arduino IDE and included those libraries at the initial position while starting to write the firmware. For Example: #include <BlynkSimpleEsp8266.h> and likewise others were required for this project. One can easily find these libraries over the internet.
2. Then we configured the NodeMCU digital as well as analog pins and defined those pins by assigning them appropriate names as per our convenience. Analog pin to read the incoming analog value from different sensors and Digital pins for read/ write purpose. This step was taken to minimize the probability of making errors while using specific pins in the firmware and makes source code easy to understand by anyone else.
3. Next Step was to assign digital pins of NodeMCU dedicatedly for the Selection pins (S0 & S1 in our case) of the multiplexer module. We had used it in our system to switch between different analog sensors and give data over a single available analog channel on the NodeMCU Board.
4. Declare variables and assign appropriate data types.
5. Then we included in our firmware the Unique Authorization Token provided by the Blynk App. One can receive this Authorization Token over his/ her registered e-mail id while signing up in Blynk App for a New Project. This is an important step.
6. Then another important step is to enter your WiFi credentials i.e. your local hotspot ssid and password. Set the password to "" for open networks.
7. There was a function that sends Arduino's up time every second to Virtual Pin (5). In the app, Widget's reading frequency should be set to PUSH. This means that you define how often to send data to Blynk App.
8. One can send any value at any time. Please don't send more that 10 values per second.

7. FIRMWARE

The firmware was edited and compiled in the Arduino IDE (Integrated Development Environment). The very first step before starting writing the code is to write an algorithm. So, we referred the algorithm at each and every step of firmware writing to minimize the possibility of error and minimize the time by writing an efficient source code for the system. There are certain key features of the firmware I found worth describing those below here.As an initial and necessary step the appropriate libraries were included at the beginning of firmware. The libraries used here were as shown here.

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>
```

Then suitable names were assigned to the NodeMCU analog as well as digital pins with the help of #define function. The names were assigned to the pins just for the ease of user's understanding about each pin's functionality; otherwise it could be difficult to remember each pin function by its pin number. The pin definitions as shown below were made as per the designed schematic diagram and printed circuit board (PCB).

```
/******** Pin Definition ************/

//Relays for switching appliances
#define Relay1          D6
#define Relay2          D2
#define Relay3          D1
#define Relay4          D5

//DHT11 for reading temperature and
//humidity value
#define DHTPIN          D7

//buzzer to use it for alert
#define buzzer          D0


//Selection pins for multiplexer module
//to switch between different sensors
//and give data on a single analog pin
#define S0              D3
#define S1              D4

//Analog pin to read the incoming analog
//value from different sensors
#define analogpin       A0
```

Then the source code was to demand from the user to enter an Authorized Unique Token ID provided by the Blynk App server. This token ID can be accessed by an authorized user only through a registered e-mail address. So, this step was considered very important here, as, to establish the communication between the NodeMCU and the Blynk this token id must be entered in the firmware as shown below.

```
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "XvnyBAxdH-Dp22GpHH53SOQv88DdjaV2";
```

Next important step was to enter the user's Wi-Fi Credentials like mobile's hotspot ssid and password to which the system needs to establish connection with.

```
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Redmi";
char pass[] = "abcd#1234";
```

For the Blynk here in this code some virtual variables (V0, V3, etc.) were assigned enabling it to communicate with different sensors and devices as shown below.

```
Blynk.virtualWrite(V0, t);
Blynk.virtualWrite(V1, h);
Blynk.virtualWrite(V2, proximity);
Blynk.virtualWrite(V3, gas);
Blynk.virtualWrite(V4, motion);
Blynk.virtualWrite(V5, light);
```

As per the 16:1 multiplexer's select line status of S0 and S1, each of the connected analog output sensor got NodeMCU's service. The other two status lines of the multiplexer i.e. S2 and S3 were grounded as we were demonstrating the system application with just four analog output sensors. The source code was as shown below.

```
// Address 00
digitalWrite(S0, LOW);
digitalWrite(S1, LOW);
gas = analogRead(analogpin);
Serial.print("Gas - "); Serial.println(gas);


// Address 11
  digitalWrite(S0, HIGH);
  digitalWrite(S1, HIGH);
  int raw_light = analogRead(analogpin);
  light = map(raw_light, 1024, 0, 0, 100);
  Serial.print("Light - "); Serial.println(light);

  Blynk.run();
  timer.run();
```

```
// Address 10
  digitalWrite(S0, HIGH);
  digitalWrite(S1, LOW);
  motion = analogRead(analogpin);
  if (motion > 512)
  {
    motion = 1;

  }
  else
  {
    motion = 0;

  }


// Address 01
  digitalWrite(S0, LOW);
  digitalWrite(S1, HIGH);
  proximity = analogRead(analogpin);
  Serial.print("Proximity - "); Serial.println(proximity);
  if (proximity < 512)
  {
    proximity = 1;
    digitalWrite(buzzer, HIGH);
  }
  else
  {
    proximity = 0;
    digitalWrite(buzzer, LOW);
  }
```

8. EXPERIMENTAL SETUP & RESULTS

The hardware set-up was initially assembled over a breadboard only and with the help of jumper wires all the sensors and the four-channel relay module were interfaced to the NodeMCU board and the power source. Initially the NodeMCU interfacing with each sensor was carried out individually and then at a later stage all the sensors were integrated in a single system along with the controlling relays. The firmware was uploaded to the NodeMCU and the system response was checked for the desired results. There was a power drop issue in the circuit and the reason was because of the significant amount of power drain by the system as all the sensors, relay and NodeMCU were connected at singe time and power demand increased. Hence, the power source rating was increased to meet the system demand. Multiple iterations were carried out to calibrate the system.

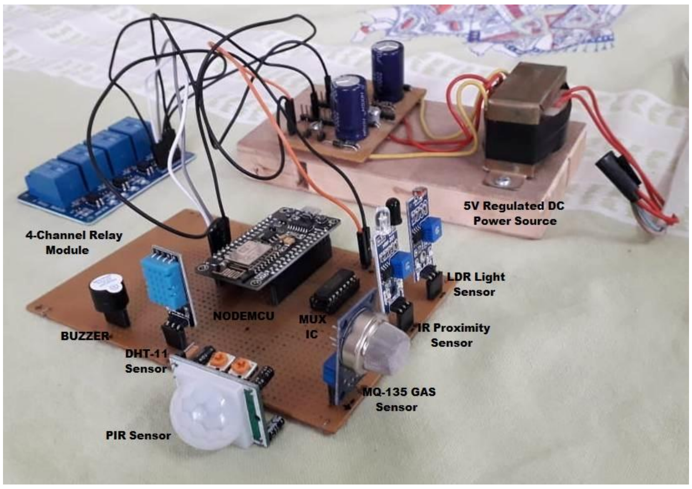

Fig -4: Experimental Setup

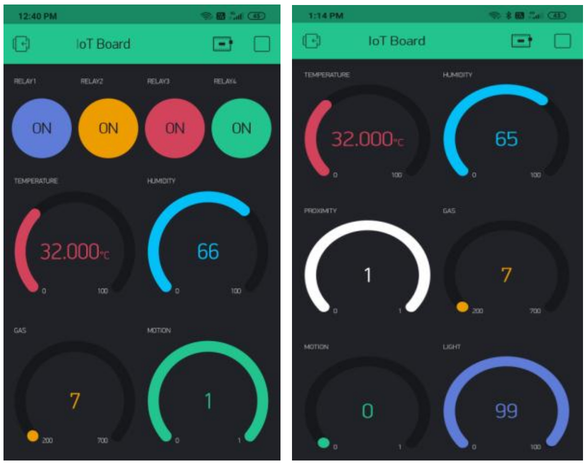

Fig -5: Outputs and Controls over the Blynk App

The firmware was edited, compiled and tested on Arduino IDE and multiple libraries were included to make the system run. The output response of the system was checked on serial monitor at each stage. To establish communication between the user and the developed IoT system various credentials were required from the user to be included at the firmware level only. The system access was limited to the Wi-Fi hotspot based local area network only. Multiple iterations were carried out to debug the firmware to get the desired results.

The results obtained from the final prototype board were observed on a mobile application known as Blynk. Blynk was downloaded and installed into the user's handset and an account was created there by entering the required user credentials. The user was assigned an authorization token ID over the e-mail which was entered in the firmware itself before uploading it to the NodeMCU. A new project was created in the graphical-user-interface window of the Blynk App by selecting some widgets from its wide library and configured those widgets as per the system hardware and firmware. All the sensors and the relay board along with the NodeMCU were connected and powered up. The NodeMCU based hardware and the Blynk over the mobile handset got connected with the help of Wi-Fi. The individual system outputs from each sensor were successfully monitored on the Blynk project screen and that too in real-time. Also the relays were triggered remotely by using this platform. Hence the work was tested and validated.

9. CONCLUSIONS

By the end of successful completion of this work it was concluded that the IoT is a new scenario of wireless communication devices. IoT is the development of existing internet facility to manage everything which exists in the world or exists in the future. Things having individualities and the simulated dispositions functioning in smart space using a smart interface to link and connect within the social environment and user context. The IoT also can be considered as global networks which give the communication between things to things, human to things and human to human. As per this work, surveillance is the procedure of close deliberate perception or supervision kept up over an individual, gathering, and so forth particularly one in care or under doubt. For the above mentioned purposes now a day's devices are equipped with various sensors as per application requirements. Sensors are communicating with each other using various topologies in IoT. Data travels locally or remotely from or in by each sensor node. As per application and requirements, sensor nodes may be of same type or different type. For a smart home, it is essential to combine sensor network with internet and intelligent real life objects. Integration of these sensors, smart objects, devices and network is IoT.

REFERENCES

[1] Dilip Kumar Sharma; Neeraj Baghel; Siddhant Agarwal, "Multiple Degree Authentication in Sensible Homes basedon IoT Device Vulnerability", 2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), IEEE.
[2] Satyendra K. Vishwakarma; Prashant Upadhyaya; Babita Kumari; Arun Kumar Mishra, "Smart Energy Efficient Home Automation System Using IoT", 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), IEEE.

[3] Kabita Agarwal; Arun Agarwal; Gourav Misra, "Review and Performance Analysis on Wireless Smart Home and Home Automation using IoT", 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), IEEE.
[4] Tushar Chaurasia; Prashant Kumar Jain, "Enhanced Smart Home Automation System based on Internet of Things", 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), IEEE.
[5] Tui-Yi Yang; Chu-Sing Yang; Tien-Wen Sung, "A Dynamic Distributed Energy Management Algorithm of Home Sensor Network for Home Automation System", 2016 Third International Conference on Computing Measurement Control and Sensor Network (CMCSN), IEEE.
[6] Waheb A. Jabbar; Mohammed Hayyan Alsibai; Nur Syaira S. Amran; Samiah K. Mahanadi, "Design and Implementation of IoT-Based Automation System for Smart Home", 2018 International Symposium on Networks, Computers and Communications (ISNCC), IEEE.
[7] Shradha Somani; Parikshit Solunke; Shaunak Oke; Parth Medhi; P.P. Laturkar, "IoT Based Smart Security and Home Automation", 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), IEEE.