Realization of an IoT system for Real-Time Remote Surveillance of Multi-Sensor Network and Control of Multiple Appliances using 'Blynk' Cloud Server

Ms. Priya Sharma[1], Mr. Parveen Kantha[2]

[1]*M.Tech. Scholar, Dept. of Computer Science Engineering, B.R.C.M.C.E.T., Bahal, Haryana, India*
[2]*Assistant Professor & Head, Dept. of Computer Science Engineering, B.R.C.M.C.E.T., Bahal, Haryana, India*

Abstract - *In this work efforts were made to integrate an IoT system with the 'Blynk' cloud server. For this an IoT system was designed, developed and implemented with NodeMCU board and multiple sensors were interfaced to it for monitoring the status of all the measured parameters remotely over a cloud server platform i.e. Blynk. Another purpose of this work was to monitor the status as well as control the switching of multiple actuators connected to the NodeMCU remotely via the same cloud server Blynk. These actuators were further connected to electrical equipments. The NodeMCU utilized Wi-Fi local hotspot network to establish connectivity with the cloud. For this purpose appropriate Wi-Fi credentials were entered into the firmware along with a unique Blynk authorization token. The hardware prototype and serial monitor output validated the work.*

Key Words: IoT, NodeMCU, WiFi, Cloud, ESP8266, Blynk, WSN, etc.

1. INTRODUCTION

1.1 Wireless sensor network (WSN)

Wireless sensor network (WSN) refers to a group of spatially dispersed and dedicated sensors for monitoring and recording the physical conditions of the environment and organizing the collected data at a central location. WSNs measure environmental conditions like temperature, sound, pollution levels, humidity, wind, and so on. These are similar to wireless ad hoc networks in the sense that they rely on wireless connectivity and spontaneous formation of networks so that sensor data can be transported wirelessly. The more modern networks are bi-directional, both collecting data from distributed sensors and enabling control of sensor activity. Today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on. The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. After the sensor nodes are deployed, they are responsible for self-organizing an appropriate network infrastructure often with multi-hop communication with them. Then the onboard sensors start collecting information of interest. Wireless sensor devices also respond to queries sent from a "control site" to perform specific instructions or provide sensing samples. The working mode of the sensor nodes may be either continuous or event driven. Wireless sensor networks (WSNs) may provide various useful data and are being utilized in several areas like healthcare, government and environmental services (natural disaster relief), defense (military target tracking and surveillance), hazardous environment exploration, seismic sensing, etc. However, sensor networks have to face many issues regarding their communications (short communication range, security and privacy, reliability, mobility, etc.) and resources (power considerations, storage capacity, processing capabilities, bandwidth availability, etc.). While Internet evolution led to an unprecedented interconnection of people, current trend is leading to the interconnection of objects, to create a smart environment.

Due to the heterogeneity of the participating objects, to their limited storage and processing capabilities and to the huge variety of applications involved, a key role is played by the middleware between the things and the application layer, whose main goal is the abstraction of the functionalities and communication capabilities of the devices.

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.

2. OBJECTIVE

Here the objective of the work was to make an attempt to integrate two technologies having complementary characteristics relative to each other. One of these technologies is well known Internet of Things and other one is the Cloud Computing. Each technology has its own set of advantages and disadvantages so the integration of both could compensate for the shortcoming of each other. For this purpose an IoT system was designed and developed around a high performance computational platform i.e. NodeMCU. The objective was to implement an IoT system for the purpose of real-time remote surveillance of multi-sensor network and also enable the user to control the switching of multiple electrical appliances remotely via a smart-phone using a cloud server. The system was expected to acquire both types of signals analog and digital from the sensors. 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 a cloud server application 'Blynk'. 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 should be capable enough to validate the work.

3. PROBLEM FORMULATION

The advancements and convergence of micro-electro-mechanical systems (MEMS) technology, wireless communications, and digital electronics has resulted in the development of miniature devices having the ability to sense, compute, and communicate wirelessly in short distances. These miniature devices called nodes interconnect to form a wireless sensor networks (WSN) and find wide ranging applications in environmental monitoring, infrastructure monitoring, traffic monitoring, retail, etc. For the realization of a complete IoT vision, efficient, secure, scalable and market oriented computing and storage resourcing is essential. Cloud computing is the most recent paradigm to emerge which promises reliable services delivered through next generation data centers that are based on virtualized storage technologies. This platform acts as a receiver of data from the ubiquitous sensors; as a computer to analyze and interpret the data; as well as providing the user with easy to understand web based visualization. The ubiquitous sensing and processing works in the background, hidden from the user.

4. SYSTEM ARCHITECTURE

Here the design was checked and for its validation a hardware working prototype was implemented and sensors were deployed to get data about multiple environmental parameters. The system also encompassed the controlling features. This smart environment application domain required a small network size with very few users and usually energized by rechargeable batteries. The internet connectivity used here in such case was either WiFi, 3G or 4G LTE with data management taken care of by a local server. The IoT devices used here were WSN and some actuators with small bandwidth requirement. The system developed here around a high performance low power low cost WiFi powered computational platform i.e. NodeMCU. The NodeMCU read analog and digital outputs from multiple sensor nodes. As most of the sensors deployed here were analog in nature there was an acute requirement of multiple analog input pins on the board but the NodeMCU board is restricted to only one analog input channel i.e. A0. So, dealing with this extremely vital situation here, an external 4:1 multiplexer was interfaced to the A0 pin of NodeMCU board. There were multiple sensors deployed in the system for Temperature, humidity control, activity monitoring for energy usage management, CO_2 levels, and others were to detect any invasion or trespassing by sensing the human motion, to sense proximity and to sense light intensity levels in its surroundings. A buzzer was also interfaced to one of the digital pins of NodeMCU for alert purposes. That was the multi-sensor portfolio of the developed system for monitoring purpose. The system was also equipped with a multi-channel control of electrical appliances and for that purpose a 4-channel SPDT relay module was also interfaced to the digital output pins of NodeMCU. The whole system was operated on a 5V / 2A regulated DC power supply. A phone with an internet connection and Blynk App installed in it was also required to demonstrate the system application.

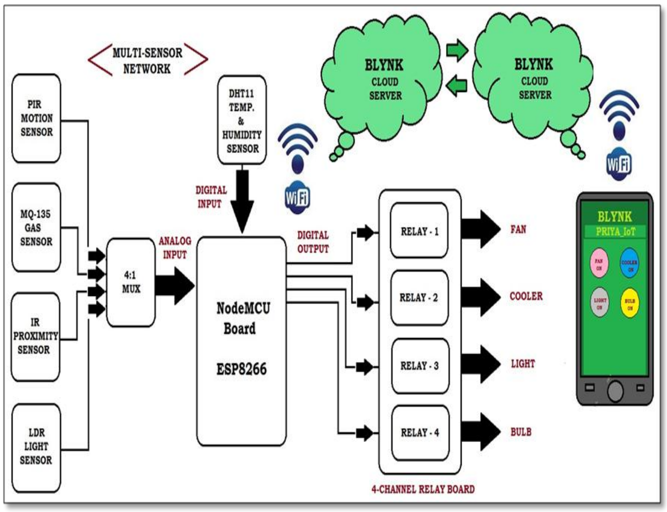

Fig.-1: System Architecture

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 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 4-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 4:1 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.

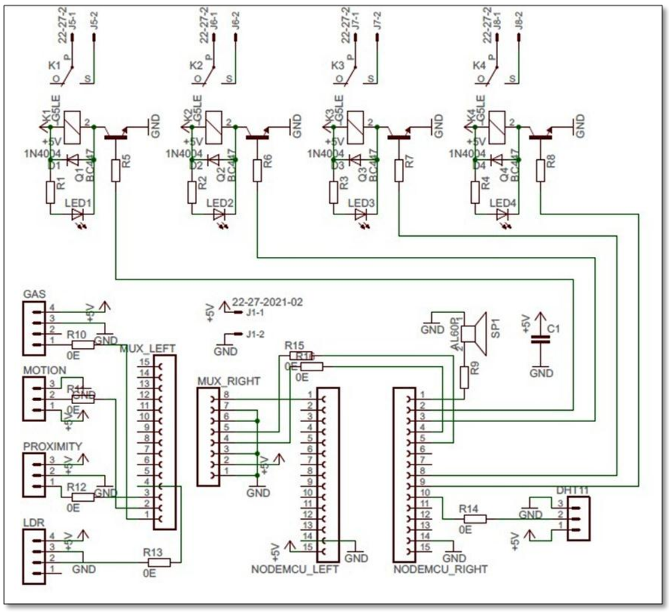

Fig.-2: System Schematic Design

6. 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);
  }
```

7. 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.

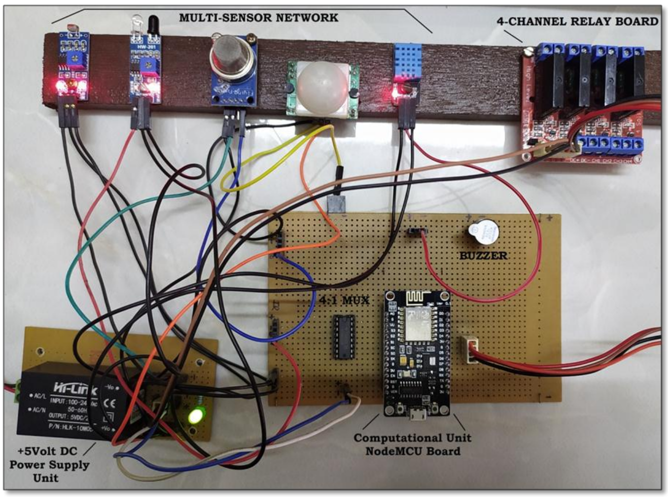

Fig -3: Experimental Setup

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.

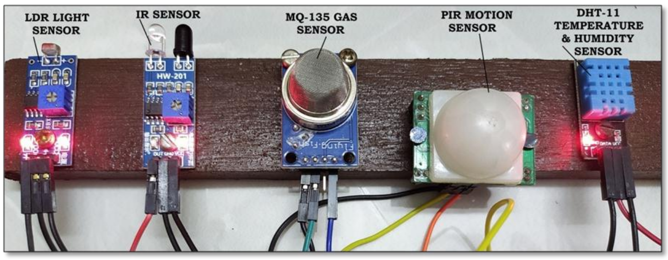

Fig -4: Outputs and Controls over the Blynk App

```
Connecting
.............
WiFi connected
IP address:
192.168.43.82
Connecting to MQTT... Failed to subscribe
Retrying MQTT connection in 5 seconds...
MQTT Connected!
Motion 5
...OK!
C02 6
...OK!
Light 4
...OK!
Failed to read from DHT sensor!
```

Fig -5: Outputs and Controls over the Blynk App

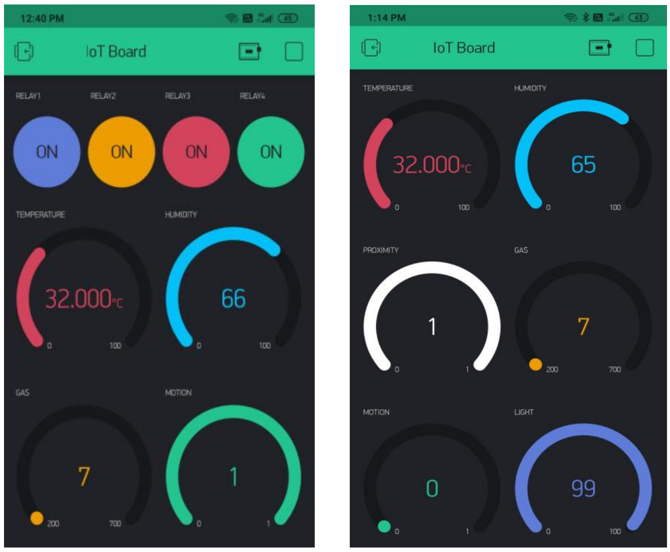

Fig -6: Outputs and Controls over the Blynk App

Fig -7: Outputs and Controls over the Blynk App

8. CONCLUSIONS

After putting rigorous efforts and successfully implementing this system some conclusions were drawn from this work.

1. **IoT:** With IoT one can connect to as many things as required. This will allow long-lasting applications to be developed and deployed using the available state-of-the-art protocols at any given point in time. The proliferation of devices with communicating–actuating capabilities is bringing closer the vision of an Internet of Things, where the sensing and actuation functions seamlessly blend into the background and new capabilities are made possible through access of rich new information sources.
2. **Data:** The data have to be stored and used intelligently for smart monitoring and actuation.
3. **Visualization:** Visualization is critical for an IoT application as this allows the interaction of the user with the environment. With recent advances in touch screen technologies, use of smart tablets and phones has become very intuitive. For a lay person to fully benefit from the IoT revolution, attractive and easy to understand visualization has to be created.
4. **Extraction of Information:** Extraction of meaningful information from raw data is non-trivial. This encompasses both event detection and

visualization of the associated raw and modeled data, with information represented according to the needs of the end-user.

5. **Application Domains:** There are several application domains which will be impacted by the emerging Internet of Things. The applications can be classified based on the type of network availability, coverage, scale, heterogeneity, repeatability, user involvement and impact.
6. **Use of Smartphone:** A Smartphone can be used for communication along with several interfaces like Bluetooth for interfacing sensors measuring physiological parameters. So far, there are several applications available for Apple iOS, Google Android and Windows Phone operating systems that measure various parameters.
7. **Cloud Centric Framework:** The proposed Cloud centric vision comprises a flexible and open architecture that is user centric and enables different players to interact in the IoT framework. It allows interaction in a manner suitable for their own requirements, rather than the IoT being thrust upon them. In this way, the framework includes provisions to meet different requirements for data ownership, security, privacy, and sharing of information.
8. **End Goal:** The end goal is to have Plug n' Play smart objects which can be deployed in any environment with an interoperable backbone allowing them to blend with other smart objects around them. Standardization of frequency bands and protocols plays a pivotal role in accomplishing this goal.

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.

[8] Arvind Arya; Akash Taliyan; Pradeep Chauhan; Anju Gautam, "Smart Kitchen with New Measurement, Web and Application Based with Affordable Design", 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), IEEE.

[9] Zulkhairi Mohd Yusuf; Md Masum Billah; Kushsairy Kadir; Abdul Malik Mohd Ali, "A Temperature-based Omnidirectional Solar Tracking System for IoT Application", 2019 IEEE International Conference on Smart Instrumentation, Measurement and Application (ICSIMA), IEEE.

[10] Mona Kumari; Ajitesh Kumar; Ritu Singhal, "Design and Analysis of IoT-Based Intelligent Robot for Real-Time Monitoring and Control", 2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), IEEE.

[11] Harsh Kumar Singh; Saurabh Verma; Shashank Pal; Kavita Pandey, "A step towards Home Automation using IOT", 2019 Twelfth International Conference on Contemporary Computing (IC3), IEEE.