IIUM ENGINEERING JOURNAL

AIMS & SCOPE OF IIUMENGINEERING JOURNAL
The **IIUM Engineering Journal**, published biannually (January and July), is a carefully refereed international publication of International Islamic University Malaysia (IIUM). Contributions of high technical merit within the span of engineering disciplines; covering the main areas of engineering: Electrical and Computer Engineering; Mechanical and Manufacturing Engineering; Automation and Mechatronics Engineering; Material and Chemical Engineering; Environmental and Civil Engineering; Biotechnology and Bioengineering; Engineering Mathematics and Physics; and Computer Science and Information Technology are considered for publication in this journal. Contributions from other areas of Engineering and Applied Science are also welcomed. The IIUM Engineering Journal publishes contributions under *Regular papers and Invited review papers*. It also welcomes contributions that address solutions to the specific challenges of the developing world, and address science and technology issues from an Islamic and multidisciplinary perspective.

REFEREES' NETWORK
All papers submitted to IIUM Engineering Journal will be subjected to a rigorous reviewing process through a worldwide network of specialized and competent referees. Each accepted paper should have at least two positive referees' assessments.

SUBMISSION OF A MANUSCRIPT
A manuscript should be submitted online to the IIUM-Engineering Journal website at
http://journals.iium.edu.my/ejournal. Further correspondence on the status of the paper could be done through the journal website.

INTERNATIONAL ADVISORY COMMITTEE

IIUM ENGINEERING JOURNAL

Volume 22, Issue 1, January 2021

Table of Contents

MATERIALS AND MANUFACTURING ENGINEERING

MECHATRONICS AND AUTOMATION ENGINEERING

MECHANICAL AND AEROSPACE ENGINEERING

MONITORING AND CONTROL SYSTEM WITH A CLIENT-SERVER MODEL BASED ON INTERNET OF THINGS (IOT)

SRI RATNA SULISTIYANTI, F.X. ARINTO SETYAWAN[*],
SRI PURWIYANTI, HELMY FITRIAWAN AND ADAM RABBANI ADNAN

[1]Department of Electrical Engineering, Faculty of Engineering,
Universitas Lampung, Bandar Lampung, Indonesia

[*]Corresponding author: fx.arinto@eng.unila.ac.id

ABSTRACT: Safety and comfort are needs for all human beings. Meanwhile, the crime rate is increasing. Therefore, a remote monitoring and control system is needed. This research offers a home monitoring and control system with a client-server model using NodeMCUESP-12E. The equipment design can be used to monitor the condition of the house through sensors installed in each room. Home monitoring includes motion detection using a motion sensor, detection of the condition of the house door using a magnetic sensor, and remote door locking using a solenoid. The system can be operated offline or online using an Android smart phone. The communication model used for client-server over the transport layer protocol is User Datagram Protocol (UDP), so the server can communicate simultaneously on two clients. The fastest average response time is 0.653 seconds. The communication model between a server and a cloud uses Transmission Control Protocol (TCP) so that the data sent or received by the server through the internet is more reliable. The cloud used is Firebase which has real-time database facilities and historical data. When online, sensor response time average is the fastest on an android at 3.898 seconds, response time control is the fastest average on a client at 7.157 seconds and the control response time average is the fastest on an android at 9.495 seconds.

ABSTRAK: Keselamatan dan keselesaan merupakan keperluan penting untuk manusia. Sementara itu kadar jenayah pula kian meningkat. Oleh itu, kita perlu pemantauan jarak jauh dan sistem kawalan. Kajian ini merupakan sistem kawalan dan pemantauan rumah dengan model pelanggan-pelayan menggunakan NodeMCUESP-12E. Peralatan yang dibina boleh digunakan bagi mengawasi keadaan rumah melalui sensor yang dipasang dalam setiap bilik. Pemantauan rumah adalah termasuk pengesan gerakan menggunakan sensor gerakan, pengesan keadaan pintu rumah menggunakan sensor magnet, dan pengunci pintu jarak jauh menggunakan solenoid. Sistem ini boleh digunakan secara luar atau dalam talian menggunakan telefon pintar Android. Model komunikasi yang digunakan pada pelanggan-pelayan menggunakan protokol lapisan pengangkutan adalah Protokol Datagram Pengguna (UDP), oleh itu pelayan dapat berkomunikasi secara terus dengan dua pelanggan pada purata masa respon terlaju sebanyak 0.653 saat. Model komunikasi antara pelayan dan awan adalah menggunakan Protokol Kawalan Penghantaran (TCP), dengan harapan data dapat dihantar dan diterima oleh pelayan melalui internet dengan lebih kebolehpercayaan. Awan yang digunakan adalah Firebase yang mempunyai kelengkapan pengkalan data waktu nyata dan data sejarah. Melalui keadaan dalam talian, purata masa sensor bertindak balas adalah paling laju pada Android sebanyak 3.898 saat, purata kelajuan kawalan masa bertindak balas pada pelanggan adalah 7.157 saat dan purata masa kawalan tindak balas adalah paling laju pada Android sebanyak 9.495 saat.

1. INTRODUCTION

Home is where we live to meet daily needs. Along with the development of current times, many are using electronic systems inside their houses or buildings. Home security technologies in use today are often web-based, so home security monitoring only occurs when the owner opens the web address. Homeowners do not know the condition of their house directly, so that when there is a danger, unwanted events cannot be prevented in a timely manner.

One solution for checking security conditions and home control each time it is needed is a tool that can monitor the application via Smartphone with an internet connection using the concept of the Internet of Things (IoT) so that users can know the state of the home anytime and at any distance via the Internet. The concept of IoT is where objects can communicate with other objects or any embedded sensors connected through an Internet network [1]. One of the services used in the concept of the IoT is Firebase. Firebase is a Cloud Service Provider (CSP) and a Backend as a Service (BaaS) owned by Google which allows users to store data and read data in real time [2,3]. As a result of using these technologies, features don't need to be built from scratch and focus can shift to easily developing IoT-based applications without the need to create a cloud.

Previous research has been carried out by Nama, who designed a surveillance system over the internet using a BCM 2835 microcontroller and sent notifications via a short message service [4], and Despa, who designed smart monitoring of electrical quantities based on single board computer BCM2835 [5]. Other research was conducted by Nurfaif who designed a smart home using Raspberry Pi through the Global System for Mobile Communication (GSM) network [6]. Other research has also been carried out by Kodali whose home security system uses the CC3200 IT microcontroller. The system works on a single microcontroller with sensors via cables and sends notifications via telephone calls [7]. Subsequent research was carried out by Malche who designed a monitoring system via the MQTT-based internet web [8].

This research, designed to make home security maintained by monitoring the condition of a house, turning on the blower when there was LPG gas, locking the door electronically, and controlling the electric light in each room through the Android application over the Internet (online) or a local network (offline).

2. THE PROPOSED SYSTEM

This research uses firebase to facilitate the communication of mobile and Web applications. Firebase is a Cloud Service Provider offered by Google to facilitate the development of mobile and web applications. Firebase has real-time database facilities and historical data. The block diagram of the monitoring and control equipment designed through this research is shown in Fig. 1.

The client consists of some sensors, each of which is connected to NodeMCU ESP-12E. Client 1 is the MC-38 magnetic sensor and HC-SR505 motion sensor, client 2 is a HC-SR505 motion sensor, and client 3 is a HC-SR501 motion sensor and MC-06 gas sensor. Each client is connected to NodeMCU ESP-12E which then connects to a server in the form of NodeMCU SP-12E as well.
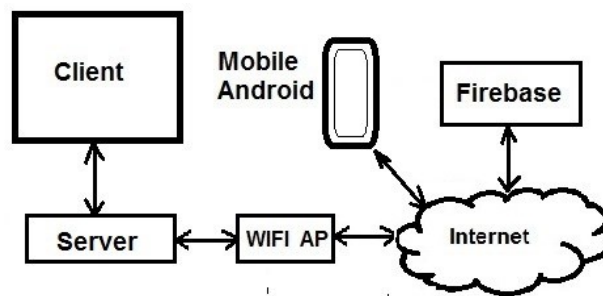
Fig. 1: Block diagram of system design.

2.1 Internet of Things

Internet of Things is a concept where certain objects have the ability to transfer data through the Internet [1]. This process does not require interaction from humans to humans or humans to a computer but can run automatically with the program. The constituent elements of IoT are artificial intelligence, sensors, connectivity, and various devices that are small in size.

Control and monitoring from very long distances can be done using IoT through the internet network. Sensors on the client side feel the phenomena that occur around them and send data to the microcontroller to be processed into information. Information from the microcontroller is then sent through the internet to the user's smartphone. Smartphone users can provide actions in the form of controlling remotely.

IOT devices can be connected to cameras installed in various locations in the city [9]. Cameras can be used for traffic monitoring, security for example, at banks, or face recognition to assist police. IOT applications in the health sector are used for remote monitoring and emergency notification systems [10]. This remote monitoring can range from monitoring blood pressure, heart rate, to sophisticated devices, namely monitoring special implants.

2.2 Firebase

Firebase is a Cloud Service Provider (CSP) and a Backend as a Service (BaaS). Firebase is the solution offered by Google to simplify mobile and Web application development [2]. This solution means that features don't need to be developed from scratch and focus can be placed on easily developing IOT-based applications without the need to create a dedicated cloud. Firebase has many SDKS that allow it to integrate with Android, iOS, Javascript, C++ to Unity.

2.3 Client-Server

Client-server is a work arrangement in accessing a server on a particular network between client and server. Data communications on the client-server are a range of interfaces that function as a place to run database applications. A client-server network is a computer network architecture where the client device will process the request for data, and the server has the task of responding in the form of data to the request [7]. The communication model of client and server network systems is shown in Fig. 2.

Client 1 used for controlling solenoid door locks in order to unlock and lock the doors, monitor motion, and monitor the condition of the door on the front porch of the House. Client 1 can be installed at the front door of the house. The goal is to be able to open and

close the door remotely and notify the homeowner if there is someone in front of the house. Client 2 is used to control the lights in the living room and monitor the movements in the House. Client 2 can be installed in the living room and used to monitor if there are people in the house.
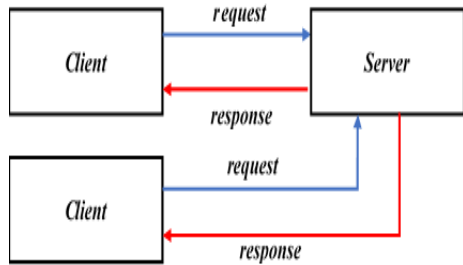
Fig. 2: Client-server network block diagram.   Fig. 3: Local network block diagram.
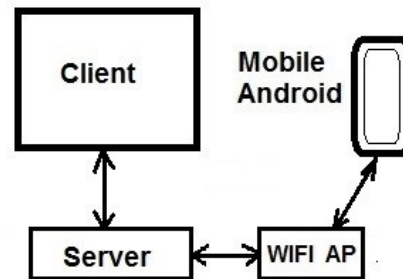
3. EXPERIMENTAL RESULTS

In this research, a comparison is made between the time response of the equipment in responding to commands when using either a local network or the Internet. The response time is determined using a stopwatch. A local network is a communication network using a Wi-Fi connection that gives commands to equipment over a maximum distance of only 10 meters. Communication or commands to equipment using the internet network can be done from anywhere with unlimited distance. Figure 3 shows a block diagram of a local network where communication is done between an Android mobile and equipment via a WiFi AP.

3.1 Client 1

Data response time on client 1 on the local network is shown in Table 1. The response time is the delay of the action that occurs on the sensor when given a command through the local network. Data is taken from five attempts with the initial conditions of both the door being closed or the door being open.

Table 1: Response time client 1 and android on the door with a magnetic sensor
(MC-38) when offline

No.	Response Time of Android (seconds)	The Initial State of The Door	The Final State of The Door
1	2.24	Closed	Open
2	3.55	Closed	Open
3	2.88	Closed	Open
4	2.65	Closed	Open
5	1.65	Closed	Open
6	1.78	Open	Closed
7	2.91	Open	Closed
8	3.28	Open	Closed
9	2.69	Open	Closed
10	1.58	Open	Closed

Table 1 shows the response time for magnetic sensor (MC-38) given the state of the door with certain conditions on the local network or offline. Response time where the client sends data or sensor feedback results to the server and the server data or sensor results broadcast feedback to android WiFi AP via a network. The value of the response time average for Android is 2.521 seconds.

On the server, the data is processed in the form of JSON data type and then the sensor or result data broadcasts feedback to the WiFi AP. The Android listens to data from the server via the WiFi AP network and receives the sensor results data.

On Android applications, each client listens to data sensor results from the server via WiFi AP. The client had an arranged time span to send data or sensor feedback results to the server with a range of 1000 ms and span of time servers for broadcast data or sensor feedback results to WiFi AP i.e. 1000 ms. So, the total minimal time span required from the client to the Android application through the local network or offline is 1000 ms.

There exists a delay in response time because when the client sends to the server. If the server is not processing data at that time, such as when it is still processing data from the cloud, this creates a span of 300ms for client listening. When a server sends broadcast data to Android, Android receives data every 1000ms and processes data from the Firebase cloud conditions.

Table 2 shows Android data response time on client 1, for magnetic sensors (MC-38) given the State of the door with a particular condition on the Internet network or online. The Android response time is where the client sends data or sensor feedback results to the server and the server is writing data to the sensor or the results of the feedback to the cloud through a network of WiFi AP Firebase. The value of the response time average android is 4.373 seconds.

Table 2. Response time client 1 and android on the door with a magnetic sensor (mc-38) when online

No.	Response Time of Android (seconds)	The Initial State of The Door	The Final State of The Door
1	4.04	Closed	Open
2	5.77	Closed	Open
3	2.84	Closed	Open
4	4.54	Closed	Open
5	3.61	Closed	Open
6	5.25	Open	Closed
7	5.99	Open	Closed
8	4.76	Open	Closed
9	2.47	Open	Closed
10	4.46	Open	Closed

On the server, the data results of the sensors or feedback received from the client is processed in the form of JSON data type and, if the data changes, then the server writes the results of sensor data or feedback to the Firebase network through cloud WiFi AP.

The Android app reads sensor data from each client from the cloud Firebase. The client has an arranged time span to send data or sensor feedback results to the server with a 1000 ms range and the span of time the server to write data or sensor feedback results to the cloud through a network of WiFi AP Firebase with a 4000 ms range. So, the total minimal time span required from the client to the internet network through the android application or online is 4000 ms.

There exists a delay in response time, because when the client sends data to the server at the time of that time, the server is still processing the data from the cloud and 300ms for listening span client and when the server sends a broadcast data to Android. Android receives data every 1000 ms and processes data from the cloud conditions Firebase.

The Android data response time on client 1 for motion sensors (HC-SR505) was given to the condition of the moving objects on the local network or offline and online shown by Table 3. The response time of the android is where the client sends data or sensor feedback results to the server and the server data or sensor results broadcast feedback to android WiFi AP via a network. In this research, data retrieval is carried out 10 times when the motion sensor senses the object moving. The value of the Android response time average is 2.3 seconds for a local network and 4.753 seconds.

The Android app reads sensor data of each client from the cloud Firebase. The client has arranged a time span to send data or sensor feedback results to the server with a 1000 ms range and the span of time for the server to write data or sensor feedback results to the cloud through a network of WiFi AP Firebase is 4000 ms range. So, the total minimal time spans required from the client to the internet network through the android application or online is 4000 ms.

Table 3: Response Time Client 1 and android on the sensor motion (HC-SR505) when online

No.	Response Time of Android when offline (seconds)	Response Time of Android when online (seconds)
1	3.8	4.45
2	1.85	5.75
3	3.02	3.54
4	3.09	5.08
5	2.64	5
6	1.85	4.32
7	1.29	3.54
8	1.78	5.62
9	2.03	4.52
10	1.65	5.71

There exists a delay in response time because when the client sends data results of a sensor to the server, the server is still processing the data from the cloud and the server has a server listening span of 300 ms. When the server sends data from the cloud conditions, Firebase and Android are reading data from a cloud firebase every 1000 ms.

Table 4 shows data for the response time of Client 1 and Android, when the solenoid door lock is executed with Open and Closed conditions on the local network or offline. The client response time is faster than Android because the response time of the client is the response time span between the Android data giving instructions to the server through a Network Access Point (AP) WiFi broadcast and the server data instructions to the client via the Wi-Fi network server. While the response time on Android is the response time of a client after the process and the Android listening time. Android listening time process is a time span of listening data from the server via the WiFi AP network. The value of the average client response time is 0.653 seconds and the value of the Android response time averages is 4.108 seconds.

On the server, data and instructions from android are treated in the form of JSON data and then the instruction data is sent to the broadcast client, so the client receives the instruction data and executes it. The server has an arranged span to send data to the client with an instructions range of 500 ms. So the minimum range of client response time via local network or offline is 500 ms.

Table 4: Response Time Client 1 and android on the solenoid door lock when offline

No.	Response Time of Client (seconds)	Response Time of Android (seconds)	The Initial State of The Lamp	The Final State of The Lamp
1	0.85	4.19	Closed	Open
2	0.8	3.84	Closed	Open
3	0.6	4.47	Closed	Open
4	0.61	3.36	Closed	Open
5	0.53	3.8	Closed	Open
6	0.74	6.15	Open	Closed
7	0.6	3.95	Open	Closed
8	0.66	3.95	Open	Closed
9	0.54	3.39	Open	Closed
10	0.6	3.98	Open	Closed

On Android, the client sends data measurement results or feedback data to a server with a span that is 1000 ms, the server listening to the client is set up with a span that is 300 ms, and server processes the data of each client in the form of JSON data then Server broadcast to android apps via the AP with WiFi within 1000 ms. So the minimum range of response time from android to the server to the client and that client to the server through a local network to the Android or offline is 1500 ms.

The existence of a delay in response time is because when Android data sends instructions to a server, the server is still processing the data from the cloud, a 300ms listening client span, and the time for the server to send broadcast data to Android. The Android application receives data every 1000ms and the server processes the data to the cloud conditions Firebase. When the client next sends the feedback data to the server, the server is still processing the data from the cloud and the server has a span of 300ms for client listening. When a server sends a broadcast data to Android, Android receives data every 1000 ms.

Table 5: Response Time Client 1 and android on the solenoid door lock when online

No.	Response Time of Client (seconds)	Response Time of Android (seconds)	The Initial State of The Lamp	The Final State of The Lamp
1	7.85	14.73	Closed	Open
2	10.19	16.71	Closed	Open
3	6.8	13.79	Closed	Open
4	7.85	14.39	Closed	Open
5	6.87	12.95	Closed	Open
6	5.88	10.28	Open	Closed
7	8.24	15.03	Open	Closed
8	6.94	11.44	Open	Closed
9	8.82	12.1	Open	Closed
10	8.58	13.81	Open	Closed

Table 5 shows data of the response time of a client and Android for Client 1, when the solenoid door lock executes commands with Open and Closed conditions over the Internet network or online. The response time of the client is faster than the Android response time because the response time on the client is the response where the time span between android writes instruction data to the Firebase cloud, the server reads the Firebase cloud, and the server broadcasts the instruction data to the client via the WiFi server network. While the response time on Android is the response time after the client response time process, the

server time process writes instruction data to the Firebase cloud, and the Android time process reads the Firebase cloud. The average response time of the client is 7.802 seconds and the average response time of the android is 13.523 seconds.

3.2 Client 2

Client 2 is a controller for living room lights and monitoring of moving objects in the house. Table 6 and Table 7 shows the data for the response time of a client and Android on Client 2, when executed with the final State of a particular lamp through the local network (offline) and Internet (online).

The average client response time for the living room lights is 0.986 seconds and the average value of the Android response time is 3.719 seconds on the local or offline network. While the average client response time for a living room lamp is 7.157 seconds and the average value of Android response time is 9.495 seconds on the internet or online.

Table 6: Response Time to Client 2 and Android on the living room light when offline

No.	Response Time of Client (seconds)	Response Time of Android (seconds)	The Initial State of The Lamp	The Final State of The Lamp
1	1.19	3.44	Off	On
2	0.93	2.62	Off	On
3	0.73	2.87	Off	On
4	1.12	4.61	Off	On
5	0.79	3.1	Off	On
6	1	4.83	On	Off
7	1.19	3.26	On	Off
8	0.73	4.71	On	Off
9	1.12	4.99	On	Off
10	1.06	2.76	On	Off

Table 7: Response Time to Client 2 and Android on the living room light when online

No.	Response Time of Client (seconds)	Response Time of Android (seconds)	The Initial State of The Lamp	The Final State of The Lamp
1	8.57	11.58	Off	On
2	7.19	9.38	Off	On
3	7.46	9.56	Off	On
4	6.74	9.12	Off	On
5	6.67	8.77	Off	On
6	5.96	8.51	On	Off
7	8.82	11.07	On	Off
8	8.25	10.33	On	Off
9	5.69	8.16	On	Off
10	6.22	8.47	On	Off

Table 8 shows the Android data response time on client 2, when the motion sensor (HC-SR505) found the condition of a moving object when used on a local network and the Internet. Response time of the android on the local network is the time response when the client sends data or sensor feedback results to the server and the server data or sensor results broadcast feedback to Android WiFi AP via a network. The value of the Android response time on the local network is 1.411 seconds.

Android response time on internet network is response time where the client sends data or sensor feedback results to the server and the server is writing data to the sensor or the results of the feedback to the Firebase network through cloud WiFi AP. On the server the data results of the sensors or feedback received from the client are processed in the form of JSON data type and, if the data changes, then the server writes data the results of sensor or feedback to the cloud through a network of WiFi AP Firebase. The value of the response time averages android using the internet network is 3.898 seconds

Table 8: Response Time to Client 2 and Android on the sensor motion (HC-SR505) when offline and online

No.	Response Time of Android on local network/offline (seconds)	Response Time of Android on Internet network/online (seconds)
1	1.58	3.74
2	1.19	4.83
3	1.33	3.35
4	1.06	3.68
5	1.45	3.37
6	0.92	3.73
7	1.85	3.89
8	1.57	4.4
9	1.39	3.99
10	1.77	4

3. CONCLUSION

On the designed system, client response time for offline usage is faster than online. The offline use does not require an Internet network and is done only through WiFi AP communication with Android mobile. Online use requires an Internet network to communicate with Firebase, which takes longer. The average response time of the magnetic sensor on client 1 for offline usage is only 2.521 seconds while online usage takes 4.373 seconds. The average response time of the motion sensor on Client 1 for offline usage is only 2.3 seconds while online usage takes 4.753 seconds. The average response time of the solenoid on the client-side for offline usage is only 0.653 seconds and on Android side is 4.108 seconds while online usage takes 7.802 seconds on the client-side and 13.523 seconds on the Android-side. For lamp control on Client 2, the average response time takes 0.986 seconds on the client-side and 3.719 seconds on the Android side on offline use, while online usage takes 7.157 seconds on the client-side and 9.495 seconds on the Android side. The average response time of the motion sensor on Client 2 for offline usage is only 1.411 seconds while online usage takes 3.898 seconds.

For further research, more sensors such as gas sensors, visual sensors (cameras), temperature sensors, and voltage sensors need to be added to provide more safety for the user.

ACKNOWLEDGEMENT

REFERENCES

[1]   Minerva R, Biru A and Rotondi D. (2015) Towards a definition of the Internet of Things (IoT). In Proceeding IEEE Internet Initiative: 1-86.

[2]   Moroney L. (2017) The definitive guide to firebase. A Press, California, Google.

[3]   Zuhelmi TP, Sulistiyanti SR, Setyawan FXA, Adnan AR. (2019) Smart home controlling and monitoring system using multiboard client-server internet of things (IoT). Journal of Engineering and Scientific Research, 1(2): 69-73.

[4]   Nama GF (2013) The design of an internet connection monitoring system at Lampung University based on a mini single board computer BCM2835. (Rancang Bangun Sistem Monitoring Sambungan Internet Universitas Lampung Berbasis Mini Single Board Computer BCM2835). In Proceeding National Seminar on Science & Technology V: 73-83.

[5]   Despa D, Kurniawan A, Komarudin M, Mardiana, Nama GF. (2015) Smart monitoring of electrical quantities based on single board computer BCM2835. In Proceeding of the 2nd International Conference on Information Technologies, Computer, and Electrical Engineering (ICITACEE 2015):315-320.

[6]   Nurfaif MB, Sulistiyanti SR, Komarudin M, Nama GF (2017) Telemetry and tele-control of electronic appliances for smart-home-system. In Proceeding International Symposium on Electronics and Smart Devices (ISESD):18-23.

[7]   Kodali RK (2016) IoT based smart security and home automation system. In Proceeding International Conference on Computing, Communication and Automation (ICCCA):1286-1289.

[8]   Malche T (2017) Internet of Things (IoT) for building Smart Home System.In Proceeding International Conference on I-SMAC Palladam-2017:65-70.

[9]   Bharati TS. (2019) Internet of Things (IotT: A Critical Review. International Journal Of Scientific & Technology Research, 8:227-232.

[10]   Madhavia MVDNS, Hemalathaa K, Sairamb PVS, Rajani D. (2016) Healthcare applications of the Internet of Things (IoT): A Review. Proceedings of 3rd International Conference on Emerging Technologies in Computer Science & Engineering (ICETCSE 2016):205-213.