

Raspberry Pi as Sensor Node and Hardware of the Internet of Things (Iot) for Smart Home

Md. Muminur Rahman, Monir Hossen, and Md. Taybur Rahama

Department of Electronics and Communication Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh

Abstract: *In the present days, home automation using wireless communication and network capabilities made the monitoring of remote parameters of sensor nodes very easy. In smart home concept, monitoring the sensor data plays an important role in detecting any abnormality in the house and taking proper response in time. However, all the existing sensor nodes required high cost and have lower processing capability and may use proprietary operating systems. That is why, these nodes do not provide optimum solution of the modern Smart Home systems. This paper proposes an implementation of cheap, fully customizable, and programmable Raspberry Pi board as gateway for the sensors and it also uses Linux as operating system which is free and open source. The proposed architecture has three main parts: a sensor data aggregator, i.e., Raspberry Pi, a notification system (SMS and Email), and a real time sensor data observation system using PubNub web service. The proposed architecture provides lower cost and user friendly environment than the existing architectures of sensor nodes.*

Keywords: *Raspberry Pi; Wireless Sensor Nodes(WSN); Smart Home; Internet of Things*

1. INTRODUCTION

Smart Home is the smallest unit of the future autonomous city where home appliances will be equipped with sensor nodes remotely monitored by the users. In the event of any abnormality in the home, a notification system will inform the user through various means, e.g., SMS to the owner's cell phone and Email notification. The Smart Home or home automation system must overcome four main issues [1]. These issues are high cost of ownership, inflexibility, poor manageability, and difficulty in achieving security. Different existing methods implemented for Smart Home and home automation for both short range, e.g., Bluetooth, Zigbee, etc., and long range through internet connected communication. Internet and wireless fidelity (WiFi) based Smart Home implementation is introduced in [2]. A web based wireless Smart Home solution with a personal PC or webserver as the central node was introduced in [3]. In [4], power line communications and wireless sensor networks (WSNs) based Smart Home control network with ZigBee, Bluetooth and WiFi to form the local network is introduced. A home gateway is needed between the local network and the internet. The use of a web server as a gateway for every home is costly and unrealistic. The authors in [5] proposed ZigBee based Smart Home system which requires lower cost and power. However, the main problem of this Zigbee based Smart Home is poor data rate and both the detection of abnormality and notification system need short response time. The authors in [6] proposed low cost Bluetooth controlled Smart Home system with challenging point to point communication and it also provides low data rate. Home monitor and alarm for fire, temperature, power, and motion etc. using the Cellular SMS also has been proposed in [7]. But the cellular SMS cost will make this system more expensive. To solve these problems new node architecture and operating system is required.

In this paper, we propose Raspberry Pi based sensor architecture for the Smart Home applications. Our proposed system is based on Twilio's free web SMS service as well as email notification which make the notification system less expensive. The proposed architecture uses PubNub web service for real time graphical representation of the temperature and humidity sensors' data and if the temperature exceeds a certain threshold it sends both SMS and email notification to the owner of the home. Home owner can remotely monitor the sensor graph in real time from any internet enabled device, e.g., mobile phone, tablet, computers, etc. using PubNub web service. In contrast, the social network integration for home automation [8] is based on Cacti and Facebook group notification systems.

Critical warning delivered to a Facebook group has limitation because it depends on user’s online status. So comparing the cost of sensor architecture and notification system with the existing architectures the proposed Raspberry Pi based architecture provides lower cost and efficient notification with lower latency.

This paper is organized as follows. Section 2 provides an overview of various components used in the proposed architecture of smart home system. Section 3 presents overview of the system architecture. Section 4 explains the experimental results and discussion. Finally, section 5 deals with conclusion and future works.

2. COMPONENTS OF THE SMART HOME SENSOR NODE

Unique system architecture is designed and implemented based on the two main components: Raspberry Pi, a DHT22 temperature and humidity sensor. Some subsidiary components also have been used for this architecture. This section presents details of all the components used in this Smart Home system.

2.1. Major Components

2.1.1. Raspberry Pi

Raspberry Pi is a single board computer. According to the application, the choice of single board computer varies. Raspberry Pi is chosen as the main processor because of its low cost, high processing capability, sufficient amount of random access memory (RAM), RJ45 port and WiFi connectivity for Internet of Things (IoT) application. Table I gives the comparison of Raspberry Pi with other single board computers. From the comparisons, we can see Raspberry pi is the cheapest single board computer with smaller size and lower cost per node that makes it suitable for the sensor node of the Smart Homes. Raspberry Pi has 40 general purpose input output (GPIO) pins through which it sends or receives data to or from sensors or home appliances. Raspberry Pi is connected with a temperature and humidity sensor, i.e., DHT22. Raspberry Pi also acts as a webserver and stores the data of temperature and humidity sensor in its MySQL database periodically. Raspberry Pi is responsible for decoding the sensor data. It requires internet connection via either local area network (LAN) or home WiFi for sending SMS and Email notifications. Fig. 1 shows a Raspberry Pi version 2 model B single board computer.

Table1. Comparison Between Raspberry Pi And Other Iot Hardwares

Features	Raspberry Pi 2 Model B	BeagleBone Black	Udoo Quad
Processor	Quad-core, 900 MHz	Single-core, 1 GHz	Quad-core, 1 GHz
RAM	1 GB	512 MB	1 GB
Cost Per Node	35 \$	55\$	\$135
Area	45 cm ²	45 cm ²	93.5 cm ²

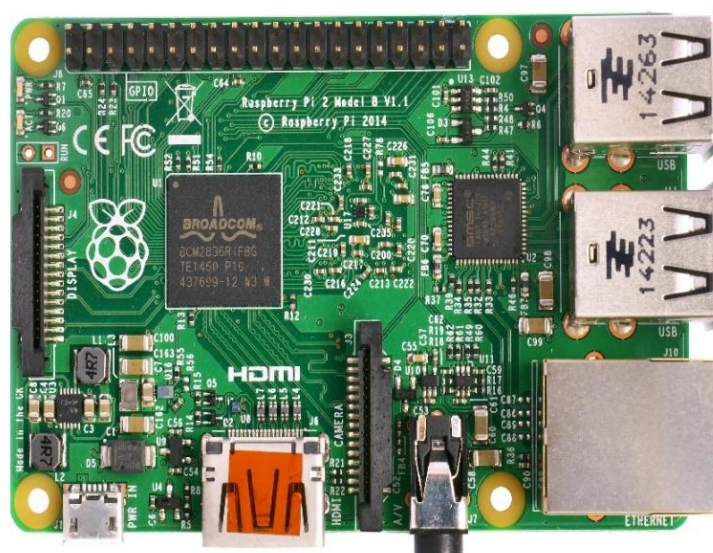


Fig1. The Raspberry Pi 2 model B.

2.1.2. DHT22 Sensor

Fig. 2 shows the DHT22 sensor which is an inexpensive digital temperature and humidity sensor and it uses a capacitive humidity sensor and a thermistor to assess the surrounding air and sends out digital signal to the data pin. It requires minimum of 3V to operate which makes it usable with Raspberry Pi's GPIO pin with 3.3V output voltage. The DHT22 draws a maximum of 2.5 mA current. It can measure humidity in the range of 0% to 100% with 2 – 5% accuracy and temperature in the range of -40°C to 80°C.

2.2. Subsidiary Components

A portable power bank (output current rating of 2A and output voltage rating of 5V) is used as a power source for the Raspberry Pi and DHT22 sensor. A monitor, a HDMI cable to connect the monitor with the Raspberry Pi, USB keyboard and mouse are used as subsidiary components during the design of the Smart Home sensor node.

3. THE PROPOSED SYSTEM ARCHITECTURE

In this section, we explain about the experimental scenario of the proposed system architecture of Raspberry pi including the notification system and graphical representation of the real time sensor data.

3.1. Experimental Scenerio

The DHT22 temperature and humidity sensor is used because in the Smart Home the temperature threshold should be determined if the fan or air conditioner need to turn on or off automatically. The temperature sensor data may also indicate fire in the house and notify the users through SMS and email notification. In addition, Raspberry Pi has 40 GPIO pins which can be used to connect with home appliances and control them based on the sensor data.

3.2. System Architecture

The proposed architecture is designed, implemented and verified to evaluate quality of service (QoS) parameters, i. e., packet loss and ping latency or round trip time (RTT).

Fig. 3 shows the system architecture of the Raspberry Pi as a sensor node for the Smart Home system. It consists of 3 parts: (i) Raspberry Pi as sensor node, (ii) a notification system using email and Twilio's free web SMS service [9], and (iii) Real time graphical representation of DHT22 sensor data using PubNub web application [10].

3.2.1. Raspberry Pi as Sensor Node

Raspberry Pi is connected to the DHT22 to sense temperature and humidity. Sensor data from the DHT22 is decoded using Adafruit's DHT22 Python library [11] and converted to user readable format for both the temperature and humidity, i.e., temperature in °C and humidity in %. The sensor data is then stored in a MySQL database which is configured in the Raspberry Pi with proper credentials. The value of the sensor data is then compared with the threshold value that we have specified in the Python program. Two temperature thresholds have been specified one for maximum positive temperature value and another for maximum negative temperature value. If the temperature exceeds these thresholds, then it sends an email and SMS notification by using Raspberry Pi as means to send the alert to the users.

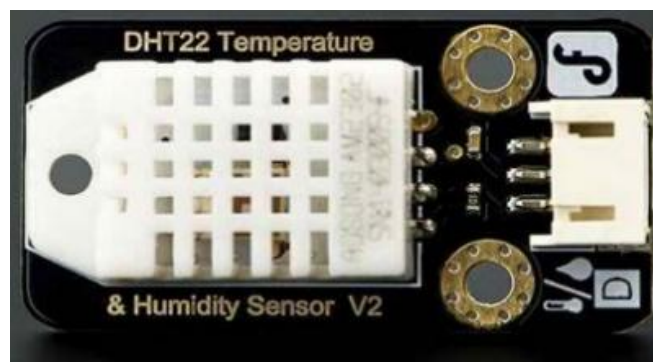


Fig2. DHT22 temperature and humidity sensor.

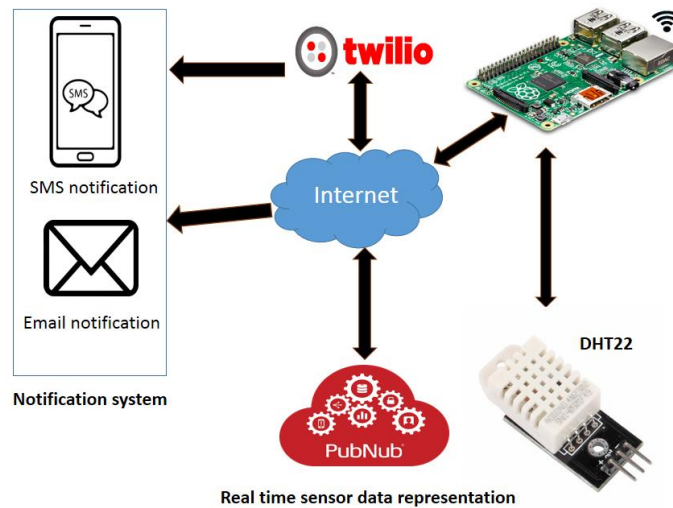


Fig3. Proposed system architecture of the Raspberry Pi as a sensor node for the Smart Home.

This feature can be used to turn on heating system in the Smart Home automatically if the room temperature drops below a certain limit or it can also be used to turn on fan or air conditioner if the temperature rises above the set limit. All data processing and sensor data evaluation are done by the Raspberry Pi. Each sensor has been given unique ID to identify a sensor specifically from a group of sensors. So the system can be easily adapted for integration with multiple types of sensors, e.g., gas sensor, pressure sensor, acoustic sensor etc. Here, the Raspberry Pi acts as both the sensor node and gateway. The Raspberry Pi is connected with internet via WiFi and establishes communication between the home network and any external network which is located far away. This facilitates user's home monitoring from anywhere anytime using any internet enabled device, e.g., tablets, smart phones or computers. The Raspberry Pi is configured as a web server and hence the temperature and humidity values from the database can be fetched and showed in a user dashboard. The Raspberry Pi is configured to allow secure shell (SSH) connection from LAN and remote network through internet. We have used a computer with Ubuntu 14.04 that gives command to the Raspberry Pi. So users can control the Raspberry Pi from anywhere in the world and monitor data from different sensors connected with it.

3.2.2. Notification System

A notification system also has been configured on the Raspberry Pi by using Twilio's application programming interface (API). The Twilio is a web service that enables sending free SMS from the internet. The users need to register in the Twilio website and verify their phone numbers. The users' credentials, i.e., account subscriber ID and authentication token, that can be used by the Raspberry Pi while sending SMS alarms. When the Raspberry Pi detects that the temperature is exceeding the threshold values, it runs the Python program for Twilio and an SMS is sent with the current temperature and humidity values to the owner's phone number. The Twilio allows a free trial of its service so we could use the SMS notifications free of cost. An email notification is also sent to the owner's email that is configured on the Raspberry Pi. This notification system can also be used to determine the event of fire in the house.

3.2.3. Graphical Representation of Real-Time Sensor Data

For real-time data representation of a sensor we have used PubNub web service. To identify for whom the message is sent, the PubNub web service gives unique subscription key to the user while signing up. Two channels are created in the Raspberry Pi for two types of data, i.e., temperature and humidity. Now, users need to subscribe those two channels to see the real time sensor data from the PubNub web service. All the channels are identified by their name and the subscription key. The Raspberry Pi continuously receives the sensor data and sends them to the PubNub through these channels and from the PubNub user interface the real time graphs for both the temperature and humidity can be observed. A line graph from our PubNub dashboard shown in Fig. 4 is used for the temperature data. The PubNub receives real time temperature updates from the Raspberry Pi and plots it as the line graph. A gauge chart from the PubNub dashboard shown Fig. 5 represents the humidity data. As the humidity data changes the gauge chart gets updated in real time.

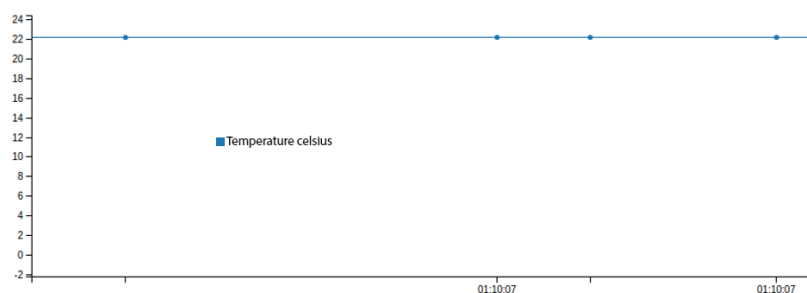


Fig4. The sensor data as graph generated by PubNub for temperature.

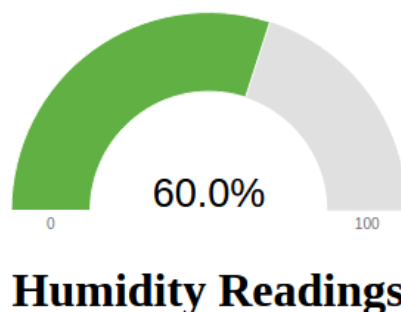


Fig5. Real time humidity reading generated by PubNub web service.

4. RESULTS AND DISCUSSION

This section presents the experimental results of our proposed system architecture and comparison table with the existing architectures of the Smart Home system.

Fig. 6 shows the column and row of temperature data table of the MySQL database configured in the Raspberry Pi to store the sensors data locally. This figure shows the date and time of the observation, ID of the sensor, and the actual values of temperature and humidity those are continuously stored in the database by running a Python program in the background. If needed it is possible to set that the average values of temperature and humidity of one week are also sent to the owner via email.

When temperature of the house rises above the trigger limit set by the owner, the Raspberry Pi will instantly send SMS and Email notifications to the owner. The time of sending mail, sensor ID, triggered limit, and the last sensed value are also stored in the database of the Raspberry Pi for

Fig. 8 shows the SMS notification received from the Raspberry Pi when the sensed temperature has exceeded the threshold value. The email notification received from the Raspberry Pi is shown in Fig. 9.

The QoS parameters like ping latency, i.e., RTT, and packet loss are shown in Fig. 10. In this figure, the horizontal axis represents local time. Here, the data acquisition time starts from 12:18 and ends at 12:23. In contrast, the vertical axis represents the RTT in ms. The RTT is taken for 5.0minutes' duration by the packets sent from the Raspberry Pi to the PubNub web server (IP 123.108.240.145) is on average 51.83 ms while the maximum latency is 142.47 ms. The packet loss is only 2.94%.

```
mysql> select * from temperaturedata;
```

dateandtime	sensor	temperature	humidity
2015-12-24 17:01:07	Temp&Humidity1	23.6	55.1
2015-12-24 17:03:01	Temp&Humidity1	23.7	54.1
2015-12-24 17:42:16	Temp&Humidity1	23.7	54.5
2015-12-24 17:54:12	Temp&Humidity1	23.7	54.5
2015-12-24 18:09:35	Temp&Humidity1	23.7	54.8
2015-12-24 18:09:43	Temp&Humidity1	23.8	54.5
2015-12-25 00:13:00	Temp&Humidity1	23.8	54.4
2015-12-25 00:15:34	Temp&Humidity1	23.8	54.6
2015-12-25 00:15:36	Temp&Humidity1	23.7	54.6
2015-12-25 00:15:38	Temp&Humidity1	23.7	54.6
2015-12-25 00:15:40	Temp&Humidity1	23.7	54.5
2015-12-25 00:15:42	Temp&Humidity1	23.7	54.6
2015-12-25 18:30:37	Temp&Humidity1	23.1	50.5
2015-12-25 19:14:01	Temp&Humidity1	23.2	50.8
2015-12-25 19:21:51	Temp&Humidity1	23.2	50.7
2015-12-25 19:33:29	Temp&Humidity1	23.1	50.7
2015-12-25 19:33:34	Temp&Humidity1	23.2	50.8
2015-12-25 19:40:08	Temp&Humidity1	23.2	50.8
2015-12-25 19:40:21	Temp&Humidity1	23.2	50.9
2015-12-25 19:41:50	Temp&Humidity1	23.1	50.9
2015-12-26 21:12:21	Temp&Humidity1	22.9	53

21 rows in set (0.00 sec)

Fig6. MySQL database table of sensor data configured on the Raspberry Pi.

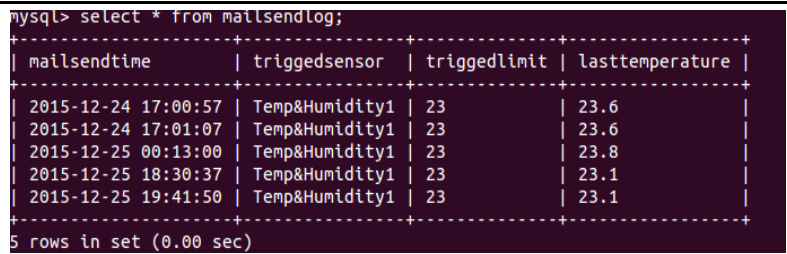


Fig7. Screenshot of mailsendlog table of Raspberry Pi MySQL database.

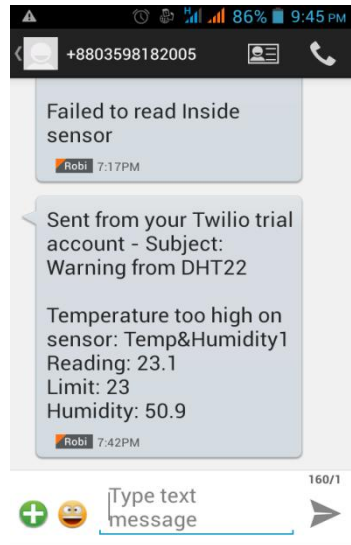


Fig8. Screenshot of SMS notification sent from the Raspberry Pi using Twilio API.

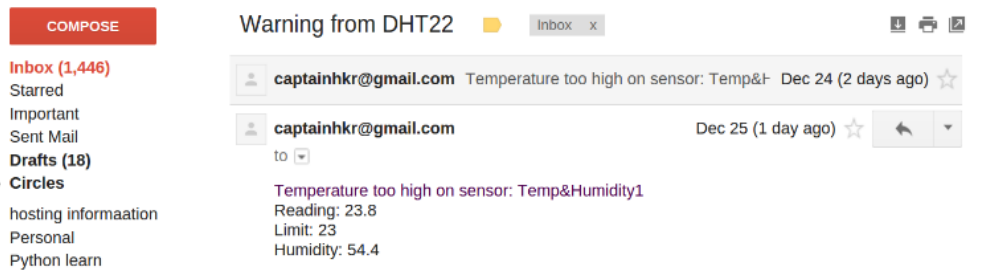


Fig9. Screenshot of the Email notification indicating temperature exceeding the threshold value.

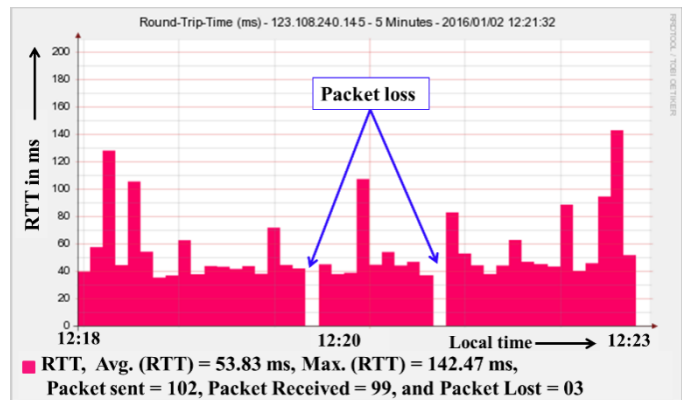


Fig10. Ping latency (RTT) and packet loss graph from PubNub.

Table II shows the comparison between the proposed and the existing architectures in [8]. In this existing architecture, the authors have proposed a home automation framework using the Raspberry Pi where the notification system is based on Cacti and a Facebook group of authoritative members. From the table we can see that the average ping latency of the existing architecture is 638.40 ms and the maximum ping latency is 1190 ms that is observed by using Cacti web service. In contrast, our proposed architecture shows the ping latency in average is 51.83 ms and the maximum latency is 142.47 ms. So from the comparison it is clear that the proposed architecture provides significantly less RTT delay than the existing architecture.

Table2. Comparison Between the Proposed and Existing Architectures

Quantity	Existing Model	Proposed Model
Average latency	638.40 ms	51.83 ms
Maximum latency	1190.00ms	142.47 ms
Cost	Same	Same
Notification system	Facebook group	SMS and Email

5. CONCLUSIONS

In this paper, the implementation of Raspberry Pi as a sensor node for the Smart Home system has been proposed and the QoS parameters of the proposed architecture also have been observed. High performance to cost ratio of the Raspberry Pi is the main reason of choosing it as a sensor node for the Smart Home. The Raspberry Pi is as small as a credit card and hence it is suitable for compact and portable applications such as sensor node in the Smart Home. In this system, the Raspberry Pi collects the data from digital temperature and humidity sensor and stores the data into the database and also sends the data to the PubNub web service. The owner of the Smart Home can monitor their home in real time using the PubNub web dashboard from the remote place. The advantages of Raspberry Pi using as a sensor node is that it has high processing capability in the cheapest price range. Also the expandable memory option of the Raspberry Pi and the internet connection made it suitable for the home automation and IoT applications. In our proposed architecture, only one DHT22 temperature and humidity sensor is used but it can be extended to multiple sensors of different kinds as the identity of each one easily identifiable from their unique ID name stored in the database. Further, we plan to apply encryption scheme for the sensor data that will provide the owner’s privacy and security of the network of the Smart Home system.

REFERENCES

[1] A. J. Brush, B. Lee, R. Mahajan, S. Agarwal, S. Saroiu, and C. Dixon, “Home automation in the wild: challenges and opportunities”, The SIGCHI Conference on Human Factors in Computing Systems, pp. 2115-2124, New York, USA, 2011.

[2] Y. Wenbo, W. Quanyu, G. Zhenwei, “Smart Home implementation based on internet and WiFi technology”, 34th Chinese Control Conference (CCC), pp. 9072-9077, Hangzhou, China, 2015.

[3] A. Z. Alkar, U. Buhur, “An internet based wireless home automation system for multifunctional devices”, IEEE Transactions on Consumer Electronics, vol. 51, no. 4, pp.1169-1174, Nov. 2005.

[4] M. Li, H. J. Lin, "Design and implementation of Smart Home control systems based on wireless sensor networks and power line communications", IEEE Transactions on Industrial Electronics, vol. 62, no.7, pp. 4430 - 4442, Jul. 2015.

[5] D. Yan, Z. Dan, “ZigBee-based Smart Home system design”, International Conference on Advanced Computer Theory and Engineering (ICACTE), vol. 2, pp. 650-653, Chengdu, China, Aug. 2010.

[6] K.Y. Lee, J. W. Choi, “Remote-controlled home automation system via bluetooth home network”, SICE Annual Conference, vol. 3, pp. 2824-2829, Fukui, Japan, Aug. 2003.

[7] H. ElKamchouchi, A. ElShafee, “Design and prototype implementation of SMS based home automation system”, IEEE International Conference on Electronics Design, Systems and Applications (ICEDSA), pp. 162-167, Malaysia, Nov. 2012.

[8] W. Warodom, K. Amatawit, W. Aekawat, “Design of home automation framework with social network integration”, J. Networking Technology vol. 5, no. 4, pp. 120-124, Dec. 2014.

[9] Twilio API, [Online] Available: <http://www.twilio.com/>

[10] PubNub Web Application, [Online] Available: <http://www.pubnub.com/>

[11] Adafruit’s DHT22 Python library, [Online] Available: https://github.com/adafruit/Adafruit_Python_DHT

AUTHORS' BIOGRAPHY



Md. Muminur Rahman, is currently working with REVE Systems Ltd. as a Software Engineer since 2016. He completed B.Sc. Engineering in Electronics and Communication Engineering from Khulna University of Engineering & Technology, Khulna, Bangladesh in 2016. His research area lie in the area of Wireless Networks, Internet of Things (IoT), Network Security, Artificial Intelligence, Software Defined Networking (SDN) etc. He enjoys Programming and solving problems in easier ways. He won Cisco NetRiders Competition in 2015 from Bangladesh.



Dr. Monir Hossen, received B.Sc. degree in Electrical and Electronic Engineering from Khulna University of Engineering & Technology (KUET), Bangladesh in 2002. He completed his M.Sc. in Electronics Engineering at Kookm in University, Korea in 2010. He received Ph.D. from University of Yamanashi, Japan in 2014. Currently, he is working as an associate professor in the Department of Electronics and Communication Engineering. His present research focuses on DBA algorithm of PON, PON-based hybrid networks and their bandwidth allocation algorithms.



Md. Taybur Rahama, He was born in Bangladesh in 1994. He received the B.Sc. Engineering degree in Electronics and Communication Engineering from Khulna University of Engineering & Technology, Khulna, Bangladesh in 2016. He joined in the software services department of REVE Systems, Bangladesh, in 2016 as an Engineer. His research interests are wireless sensor network, passive optical network, Smart grid, open access network etc.