DOI: https://doi.org/10.33103/uot.ijccce.20.3.7

IoT Monitoring System Based on MQTT Publisher/Subscriber Protocol

Narges A-hussein[1], Ayman D. Salman[2]
[1,2] *Department of Computer Engineering, University of Technology, Baghdad, Iraq*
120633@student.uotechnology.edu.iq, 120008@uotechnology.edu.iq

Abstract— *In the past years, the scene has witnessed huge technological progress which made our lives simpler and flixper. After Wi-Fi and cellular communications networks' improvements, with the parallel optimization of numerous embedded devices, momentum has risen globally and today gave us a concept that is the IoT or the Internet of Things. Where several contemporary technologies have been utilized and the developers have been advancing structures to collect data from sensor systems that may be sent to any part of the world over the Internet. The Internet of Things can be used for many purposes like controlling, tracking and managing systems. In this study, we presented the work of the MQTT internet routing protocol to exchange sensor information between two different devices. The IoT platform is about monitoring temperature and humidity in a smart home based on an MQTT protocol which makes this connection possible. However, the MQTT protocol works in publishing/ subscribing mode. The proposed work used an Esp8266NodeMCU as a publisher and Raspberry pi3 model (B) as a subscriber. Also, it used a dht11 sensor to measure the temperatures and humidity. The measurements were collected from sensors and alternated between the two devices through the MQTT broker (server). Node-Red and ThingSpeak designed as a website to share the data.*

Index Terms— *Message Queuing Telemetry Transport protocol, Esp8266 NodeMCU, Raspberry pi3 model B.*

I. INTRODUCTION

In recent times, The concept of IoT resonates in information and communications technology. The Internet of Things is a collection of devices like cars, sensors, and electrical devices that are connected to the Internet through PCs, laptops, and portable devices [1]. One of the most significant characteristics of the IoT is that it is a smart environment and requires no human efforts in order to be managed. This is why the Internet of Things has become a major player in many applications. An Internet of Things application is the design of a smart city tracking and management system for a smart light environment to reduce restricted energy consumption [2]. Surveillance systems that rely on IoT technologies are many, such as the system to monitor the temperature and humidity during the shipment of goods, on the farm and in the agricultural environment[3,4]. The layout of those systems has contributed to lowering costs, shortening time and lowering human effort through the use of modern-day technology[5]. The most usual routing protocol in IoT is Message Queuing Telemetry Transport or MQTT protocol. It launched through IBM in 1999, to transfer data correctly according to shore parameters which include long network delay and low bandwidth network utilization[6]. MQTT characterized as open-source, low code size, for remote places and consumes a little of the network bandwidth. MQTT applications to work on mobile devices, like tablets and smart-phones. Telemetry is utilized as well for receiving the statistics from remote sensors and managing them [7]. The cause which made people and builders select MQTT is a flexible and light-weight protocol. The publish/subscriber mechanism designed to be clean to implement [8]. Those characteristics make it perfect for data and statistics, trade from one device to another. In this paper, a work proposed for the Internet of Things to monitor temperature and humidity in the smart environment by using the MQTT protocol. It measures temperature and humidity measurements using the dht11 sensor. So components of

Received 17 March 2020; Accepted 28 April 2020

the MQTT protocol in this proposed work are server(MQTT broker) and two clients. The first client is an ESP8266Node NodeMCU, which is a publisher. This microcontroller collects the records from the sensor and publishes it to the MQTT broker (server). The second client is a Raspberry pi3 model B(on-board computer)which is a subscriber. It subscribed to the records from a server and shared it via the internet to Node-Red webpage as an IoT dashboard.

II. MQTT ARCHITECTURE

The MQTT architecture can divide into two main parts as shown in figure- (1). This section presents a brief description of each part:

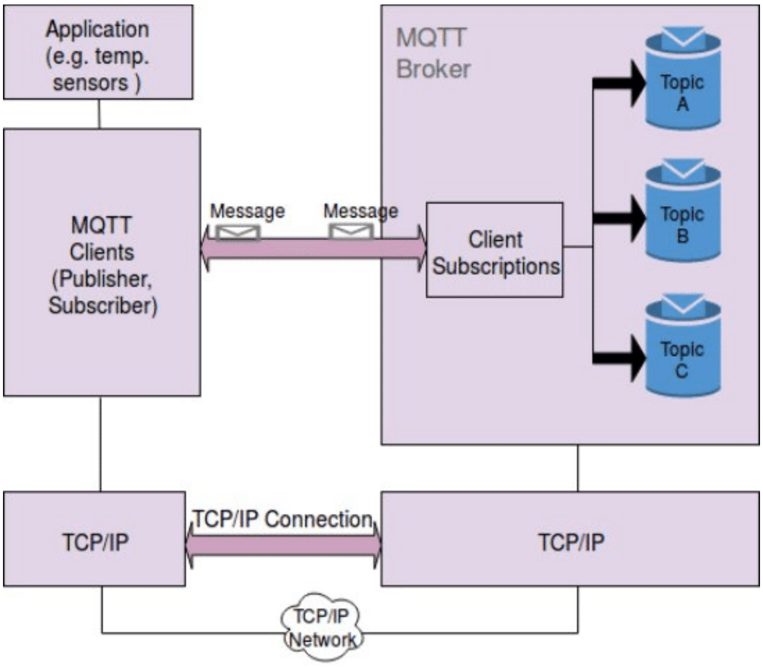


FIG.1. MQTT ARCHITECTURE.

A. Client:

The MQTT client may behave as a publisher or a subscriber and it's responsible about establishing the network connections to the server (MQTT broker) also it performs the next functions[9]:
1. Publish messages according to the device's demands.
2. Subscribe to favourite topics for receiving messages.
3. Unsubscribe to exit from shared topics.
4. cut-off connections from the broker.

B. Broker:

The MQTT broker is responsible for acceptance of every message beyond the publisher, distinguishes them, and decides who the client will subscribe to. Then re-sent the messages following the entire clients (subscriber).
1. Accept and receive all client requests.
2. Messages revealed in conformity with that from anybody device.
3.Deals including more than a few requests (subscribe, unsubscribe) from devices.
4. After arrival messages from the publisher sends it to all devices that request the messages.

Received 17 March 2020; Accepted 28 April 2020

A study aim: This study verifies the data transmission between two different devices based on the feature of the MQTT protocol to publish and subscribe pattern. The transmission is done at the lowest cost, fastest time and accurate measurements according to the principle of the IoT platform. It is a monitoring system for (temperature and humidity) in smart (home, buildings, or any place) indoor/outdoor to provide protection.

III. LITERATURE REVIEW

In ~~Ref~~.[10] the researchers introduced a temperature and humiditymonitoring system based on MQTT protocol. They used the following basic components in their system: Arduino UNO microcontroller board depend on the ATmega328P, an Ethernet guard for connecting Arduino to the internet, dht22 temperature and humidity sensor, router for sending sensor data to the internet and relay module for exchanging among an electronics device, on this paper, new hardware, and technologies proposed with the lowest cost. We used Esp NodeMCU. This microcontroller is greater than modern-day and has Wi-Fi onboard for connecting without needing an Ethernet guard. Also, use the adht11 sensor which is smaller and less expensive thandht22. It can degree temperature of 00-500 Celsius and humidity of 20%- 90% RH. Extensively used Raspberry pi3. It is an on-board computer that has many capabilities that permit us to deal with information as we want. In Ref.[11] The paper presented the home automation and monitoring system. Make HTTP connections to send data to the web and FTP servers, and the use of TCP / IP and UDP / IP sockets to connect to any server on the Internet, in this paper, we preferred the MQTT protocol on HTTP in publishing data on the website because HTTP is designed as a request/ response protocol for customer service conversation Which characterized via the bad use of power and plenty of interfering statistics. So, the MQTT protocol designed as a publish/subscribe protocol. It consumes power better than HTTP and there is no interference between the data. MQTT runs over TCP/IP network means it's work after the network connection completed. In Ref.[12] the authors have accomplished the environmental monitoring system to monitor temperature and humidity in real-time. The data sent through Wi-Fi to the cloud where the data is in real-time and it's displayed as a graphical analysis. In this paper, we used two types of IoT dashboard. one of them is a ThingSpeak website to share the statistics directly from ESP8266Node NodeMCU via the internet and Node-Red dashboard for sharing the same statistics but from Raspberry pi3. The data displayed as a chart and gauge on the IoTdashboard immediately because it is variable, non-stop, and offers real-time.

IV. The proposed work:

The main components of the proposed work show in figure-(2) below;

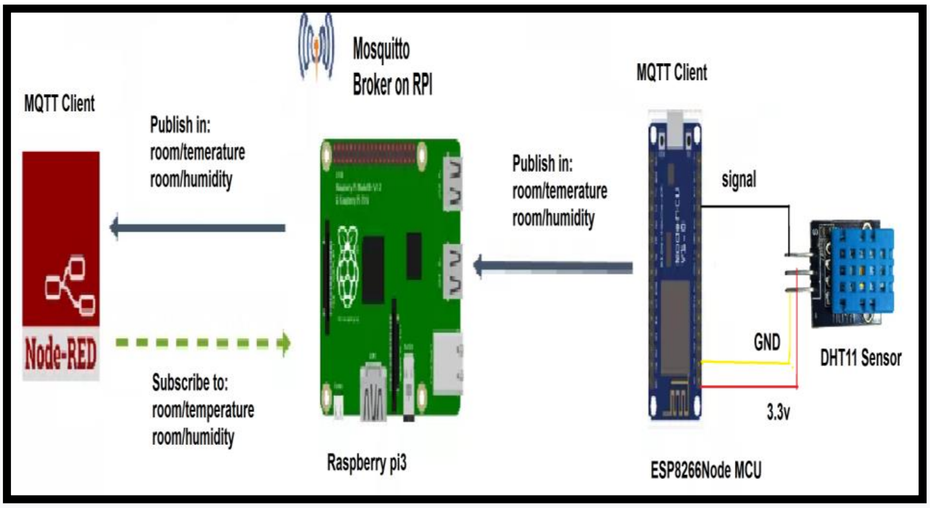


FIG.2. THE PROPOSED WORK.

Received 17 March 2020; Accepted 28 April 2020

A. Software Requirements.

1. Arduino IDE software**:** For ESP8266 NodeMCU programming.
2. Node-Red tools: It is a graphical programming tool for the JS node. It is used to connect devices, API, and online services in an exciting new way. In addition, it provides a browser-based browser that makes it easy to provide bulk streams. Wide nodes can be spread within the board for one-click playback time.
3. MQTT: Network protocol (publisher/subscriber protocol) used to transfer data between two devices (IoT communication).
4. ThingSpeak**:** Is an IoT analytics platform provider. That provides immediate visualizations of the statistics posted via your devices to ThingSpeak. It can carry out online evaluation and processing of the records as it comes in. ThingSpeak is regularly used for prototyping and evidence of an idea IoT structure that requires analytics.

B. Hardware Requirements.

1. ESP8266-NodeMCU**:** It is an inexpensive, compact, and modern micro-controller unit. It has been built around a less expensive chip system known as ESP8266. It has the basic features of the best PC with RAM, OS, CPU, and Wi-Fi connection. Those features make the NodeMCU an impressive array of IoT frameworks. Figure3 illustrates the configuration of the board and pin out of the ESP8266-NodeMCU.

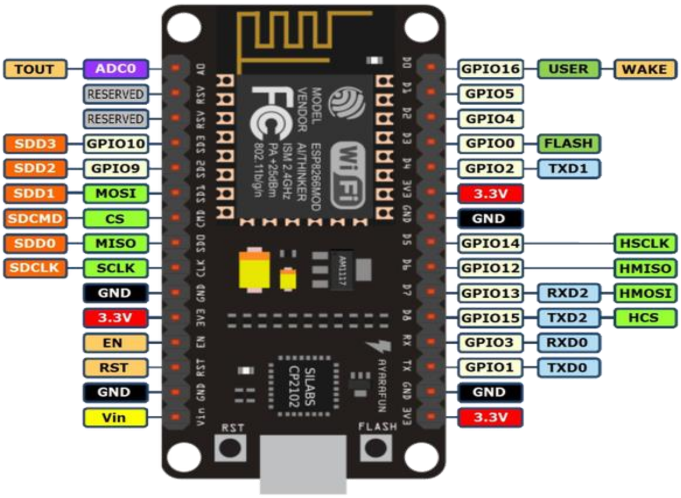


FIG.3. PINOUT OF ESP8266NODEMCU.

2. DHT11 sensor**:** An extraordinarily low-cost basic digital sensor of humidity and temperature. Capacitive humidity sensor and thermistor are used for measuring the ambient air and creating a digital signal to the information pin. It is easy to use but requires accurate timing to extract records. The simplest real drawback to this sensor is that you can exceptionally get new records from it every two seconds. Therefore, when using our library, sensor readings can reach two seconds. Figure- (4) shows the DHT11 sensor.

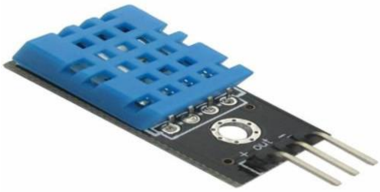


FIG.4. DHT11 SENSOR.

3. Raspberry Pi3 Model-B: Is the third generation of the raspberry pi. It is characterized by processor pace 1.25GHz, an ARM-based credit score card-sized SBC (Single Board Computer). With on-board Wi-Fi/ Bluetooth and a 64bit upgraded, switched electricity source up to 2.5Amps. It can now step forward

Received 17 March 2020; Accepted 28 April 2020

processor energy even with extra powerful devices over USB ports. This powerful single size credit card computer can be used in many projects of students and engineers. Figure- (5) Shows the external appearance of the Raspberry pi3 model B.

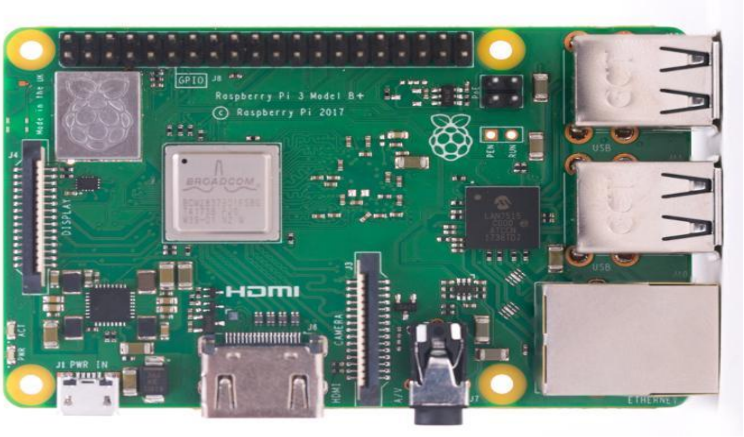


FIG.5. RASPBERRY PI3 MODEL-B

Note: The router and PC as an essential deviceswhich used in this framework.

C. The proposed work flowchart:

we can summarise the steps of the suggested work in a flowchart as can be seen in figure- (6) below;

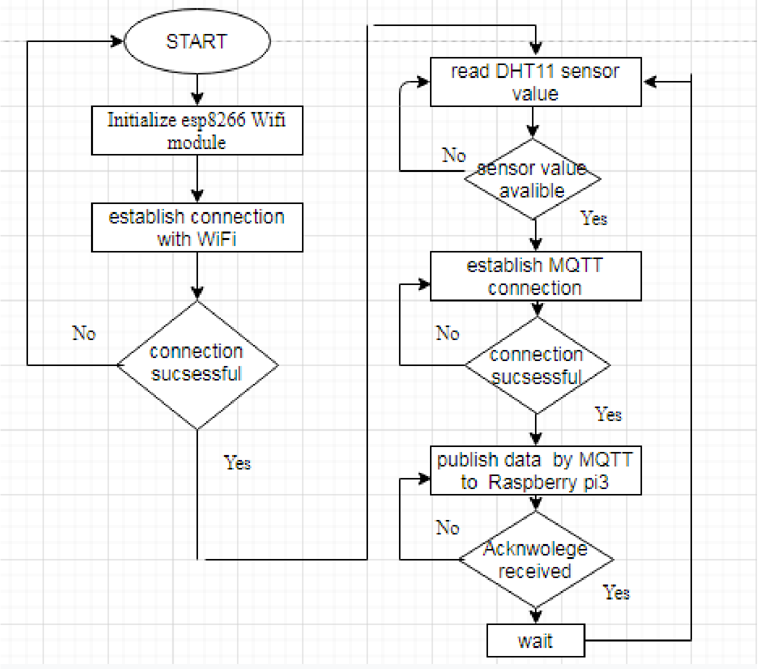


FIG.6. FLOWCHART OF PROPOSED WORK

D. The proposed work steps:

Step one: connect Esp8266NodeMCU to the Wi-Fi network. Then downloaded Esp8266 packages on Arduino IDE which allows us to select this board and programming it. The dht11 sensor has 3 pins(GND, SIGNAL, VCC). The signal pin connects to the digital pins of Esp8266Node MCU marked by D3(GPIO0). The GND and the VCC(3.3V) pins of sensor connected to the GND and VCC of a microcontroller. Figure- (7) illustrated the connection between the dht11 sensor, esp8266, and raspberry pi3.

Received 17 March 2020; Accepted 28 April 2020

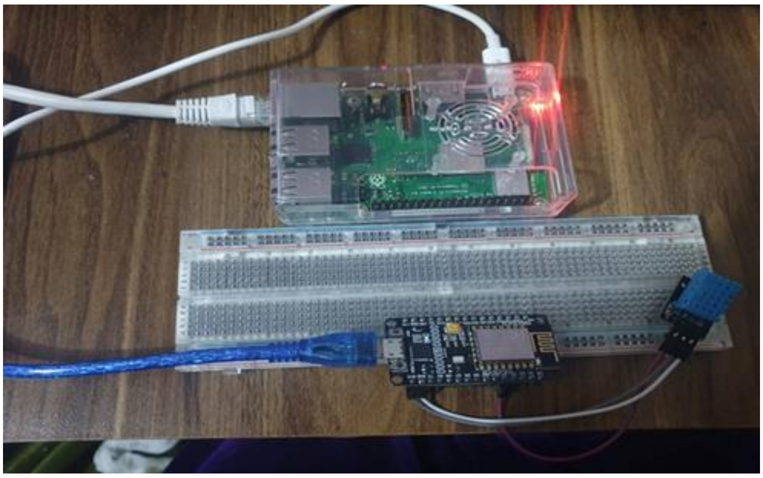
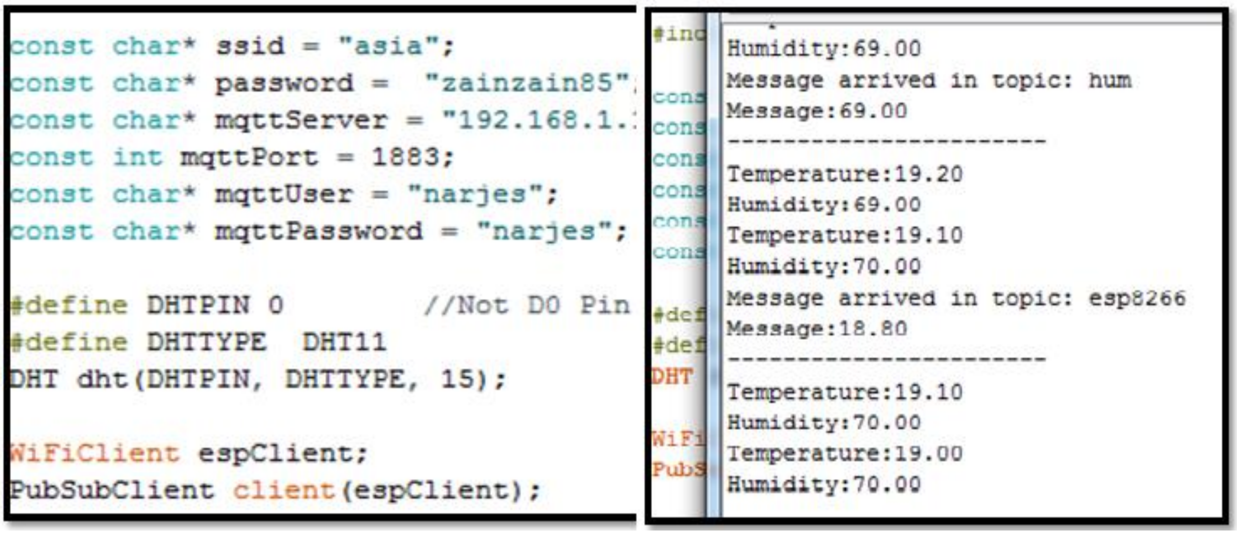


FIG.7. THE CONNECTION AMONG DHT11, ESP8266NODE MCU AND RASPBERRY PI3.

***Step two*:** We wrote code in the C programming language on the Arduino IDE to programming the microcontroller. This code defined the sensor and MQTT client for publishing data to MQTT broker (server) and Wi-Fi networks. After running the code, the measurement is fetched correctly as illustrated in the figure- (8) below;



```
const char* ssid = "asia";
const char* password =  "zainzain85";
const char* mqttServer = "192.168.1.:
const int mqttPort = 1883;
const char* mqttUser = "narjes";
const char* mqttPassword = "narjes";

#define DHTPIN 0         //Not D0 Pin
#define DHTTYPE  DHT11
DHT dht(DHTPIN, DHTTYPE, 15);

WiFiClient espClient;
PubSubClient client(espClient);
```

```
Humidity:69.00
Message arrived in topic: hum
Message:69.00
-----------------------
Temperature:19.20
Humidity:69.00
Temperature:19.10
Humidity:70.00
Message arrived in topic: esp8266
Message:18.80
-----------------------
Temperature:19.10
Humidity:70.00
Temperature:19.00
Humidity:70.00
```

FIG.8. READ MEASUREMENTS OF DHT11 SENSOR AT ESP8266NODE NODEMCU.

The Esp8266Node NodeMCU is a client that behaves as a publisher. It collects temperature and humidity data from dht11 sensor and published it to the MQTT broker (server) can also publish facts to websites such as thingSpeak as shown in figure-(9) below;

Received 17 March 2020; Accepted 28 April 2020

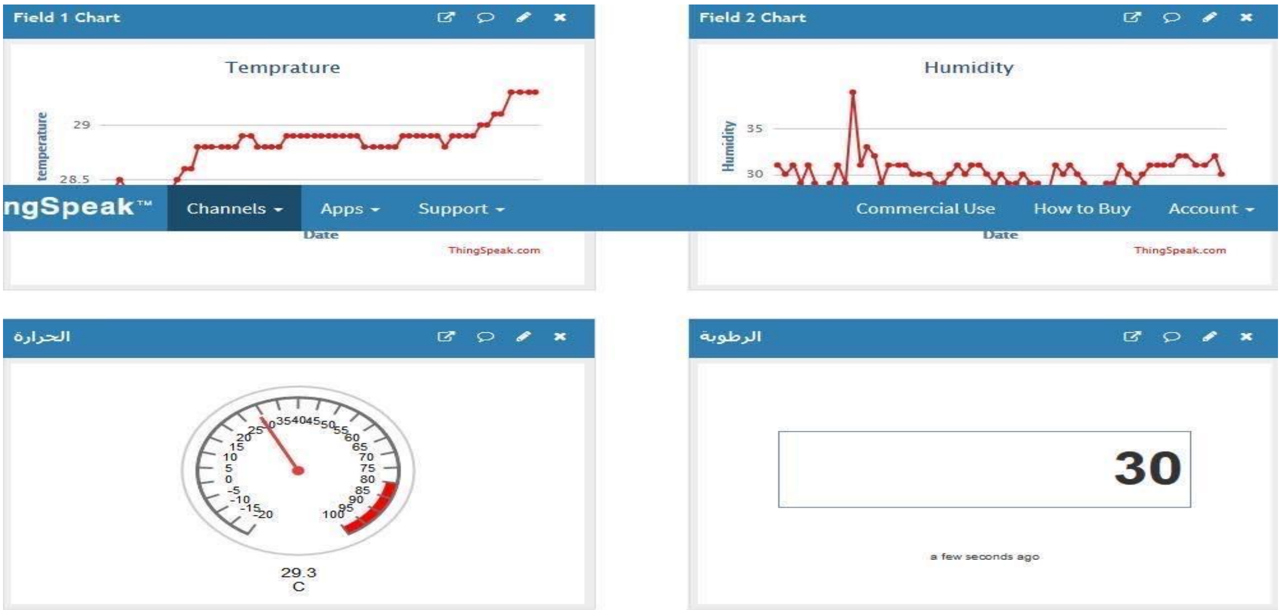


FIG.9. DATA PUBLISHED TO THINGSPEAK WEB PAGE

Step three: we established the MQTT broker(server) on raspberry pi3. the broker here acts as a server only for publishing/subscribing messages. MQTT broker work is shown in figure-(10). It published the messages from the client (microcontroller) and subscribed to those messages to the client(Raspberry pi3).

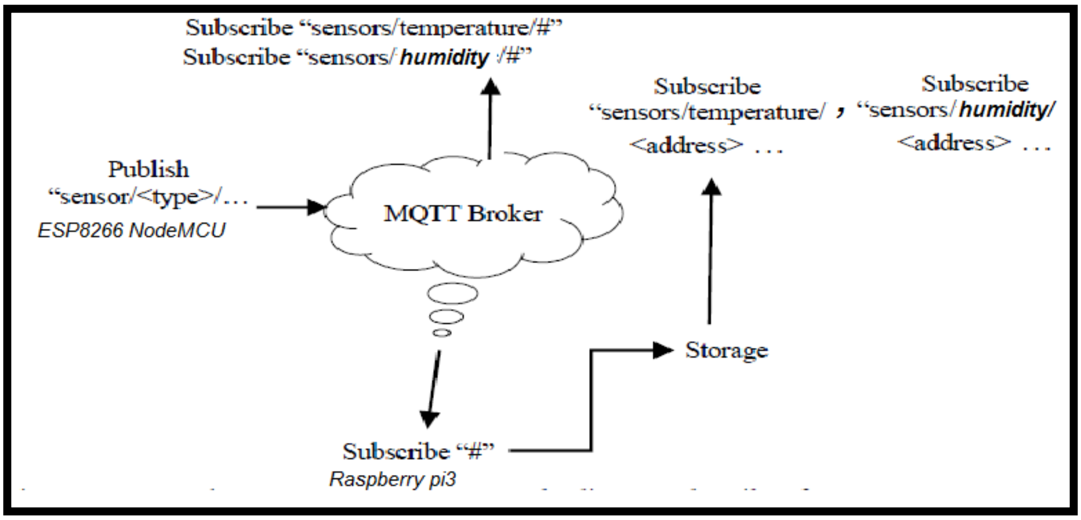

Subscribe “sensors/temperature/#”
Subscribe “sensors/ humidity /#”
Subscribe “sensors/temperature/ <address> ...
Subscribe “sensors/ humidity/ <address> ...
Publish “sensor/<type>/...
ESP8266 NodeMCU
MQTT Broker
Storage
Subscribe “#”
Raspberry pi3

FIG.10. MQTT BROKER WORK.

Step four: Raspberry pi3 subscribed to the information from a server. We used NODE_RED as Node Js graphical programming tools. Which is also installed on raspberry pi3. The IoT dashboard includes charts and gauges that show active values of temperature and humidity. The charts represent the measured sensor records as a function of time. raspberry pi3 will submit the readings of sensor information that subscribed it from an MQTT server on the node-red dashboard through the internet as shown in figure-(11) below;

Received 17 March 2020; Accepted 28 April 2020

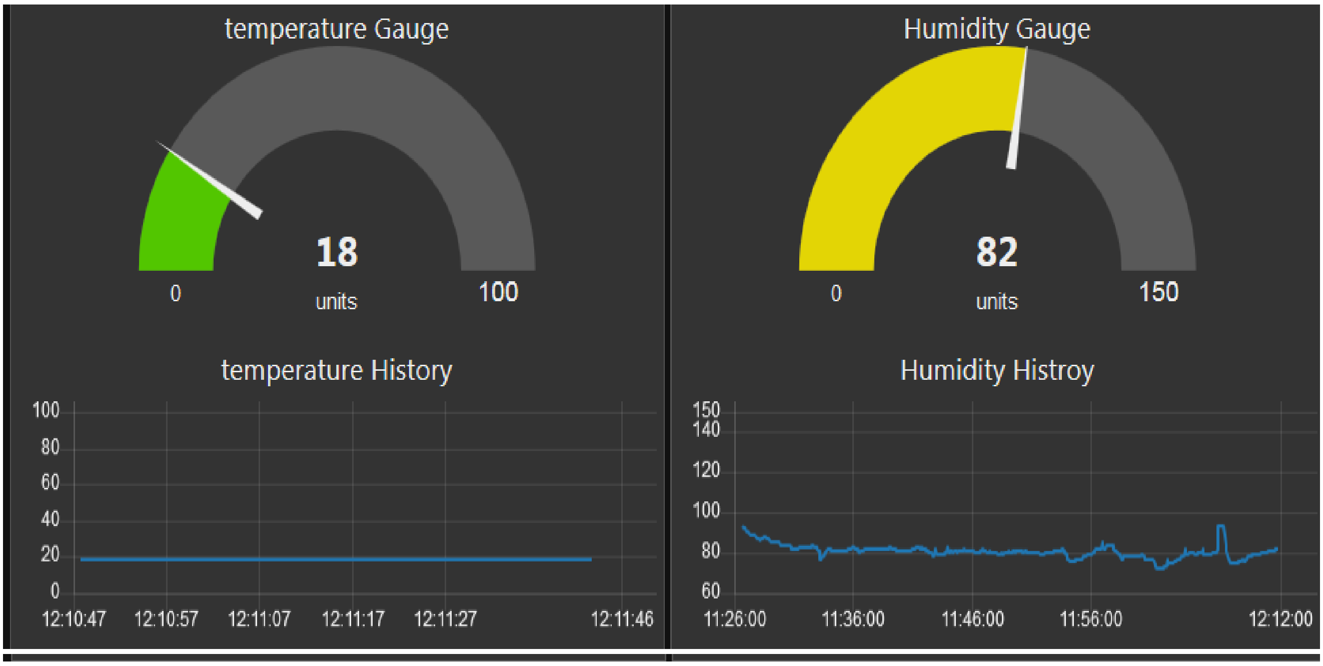


FIG.11. NODE-RED DASHBOARD.

E. Analysis of the results:

1. MQTT protocol allows us to determine which device is a publisher and which is a subscriber. This feature is very important because it makes the network flexible and possible to exchange the role of each device in the work.
2. It runs over TCP/IP connection meaning it's operating after the network connection is completed .
3. MQTT subscribes to the published message to all subscribers that request the message.
4. It is a light-weight protocol. That used to send the small message such as ON, OFF, sensor records and not for blocks of messages.
5. MQTT protocol based on IoT devices (machine to machine).
6. The code in MQTT protocol is a small size and not required a big size code to publish or subscribe to the message.

V. CONCLUSION

By searching at previous works and via this work we have come to the conviction that MQTT is more appropriate and an effective protocol for IoT applications. MQTT protocol may be more popular over the coming years. On this platform of work, we proposed a monitoring system using the MQTT internet routing protocol. ESP8266Node MCU was a publisher and raspberry pi3 was a subscriber. We see how sensors to record information transmitted from one tool to alternative devices. Also, we notice the statistics read from the sensor are accurate after comparing it with the readings of the thermometer. There is no difference in the readings and if any, it is very little that does not exceed above 0.5or less than -0.5. This system was a monitoring of the temperature and humidity constantly and display the outcomes on the website directly.

In future It is possible to rely on the MQTT protocol to build an integrated monitoring system in the smart home by installing it on a local server. The local server takes readings from the sensors or cameras which are connected with different smart domestic devices. Then transfers the data to the MQTT broker, which in turn publishes them on a web page. This allows you to screen the sensor records from everywhere within the world using your smartphone or tablet. While are away, could additionally secure your house and the alarm will sound when the event starts.

Received 17 March 2020; Accepted 28 April 2020

REFERENCES

[1] K.J. Reshma, J. Selvin Peter Paul, V. Swetha, A Study on MQTT based Environmental Parameters, Monitoring and Alarming System, Eurasian Journal of Analytical Chemistry

[2] ISSN: 1306-3057 OPEN ACCESS 2018 13 (SP): 323-327, 2018.

[3] Aymen DawoodSalman,Ghaidaa M. and Ula K., An adaptive smart street light in a smart city, Computational and Theoretical Nanoscience, 2019.

[4] Zaenurrohman, Arief M., Suryani A. Temperature and Humidity Monitoring on IoT Based shipment Tracking, Electrical Engineering Post Graduated Program, Sultan Agung Islamic Unversatile. Raya Kaligawe KM 4 Semarang 50112, Indonesia, Journal of Telematics and Informatics (JTI) Vol.6, No.1, pp. 27~36,ISSN: 2303-3703, March 2018.

[5] R.Mythili, M. Kumari, A. Tripathi, N. Pal. IoT Based Smart Farm Monitoring System, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-4, November 2019.

[6] Akash1, Amit B. IoT-based Temperature and Humidity Monitoring System for Agriculture M.Sc Student, Department of Electronic Science, University of Delhi, India1 Assistant Professor, Department of Electronic Science, University of Delhi, India2 International Journal of Innovative Research in Science, Engineering and Technology, July 2017.

[7] Kraijak, S., & Tuwanut, P., A survey on internet of things architecture, protocols, possible applications, security, privacy, real-world implementation and future trends, In Communication Technology (ICCT), IEEE 16th International Conference on (pp. 26-31), October 2015.

[8] Adnan S., Brennan C., Using the MQTT Protocol in Real Time for Synchronizing IoT Device State, Electrical and Computer Engineering Department, University of Michigan, Dearborn,The International Arab Journal of Information Technology, Vol. 15, No. 3A, Special Issue 2018.

[9] Valerie L., Weng T.L., Leonardo O., Sweta R., Nagesh S., Rong X., Gerald K., Neeraj K., Stefan F., Martin K., Dave L.. IBM redbooks building smarter planet solutions with MQTT and IBM WebSphere MQ telemetry, International Technical Support Organization, September 2012.

[10] Dipa S., Ashwin M., A SURVEY ON MQTT: A PROTOCOL OFINTERNET OF THINGS(IOT), CHARUSAT University, Gujarat, India, ResearchGate, April 2017.

[11] Sumit P., Sourav G., Sarasij B. Study and implementation of environment monitoring system based on MQTT. Department of Information Technology, Indian Institute of Engineering, Science and Technology, EESRJ, ISSN: 2369-5668(print), 2369-5676(online), Vol. 4, No. 1. March 2017, pp. 23-28.

[12] A. P. Vancea, I. Orha, Smart home automation and monitoring system, Electric, Electronic and Computer Engineering, Technical University of Cluj-Napoca, North University Center Baia Mare, Baia Mare, Romania, Carpathian Journal of Electronic and Computer Engineering 11/1 (2018) 40-43, 2018.

[13] Saima Z., Ghosia M., Rajaa B., Danish M., Khadija A., An IoT Based Real-Time Environmental Monitoring System Using Arduino and Cloud Service, Department of Electrical Engineering National University of Computer and Emerging Sciences, FAST-NU, Lahore Campus, Lahore, Pakistan, Engineering, Technology & Applied Science Research Vol. 8, No. 4, 2018, 3238-3242.

Received 17 March 2020; Accepted 28 April 2020