International Journal of Scientific Research in Science, Engineering and Technology
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET

IoT Based Smart Saline Level Monitoring System

Aman S. Ramteke[1], Pranali A. Pardhi[1], Shivani N. Gandlewar[1], Khushbu C. Ingale[1], Dhawal Dongre[1], Ms. N. D. Sonwane[2]

[1]BE Students, Department of Computer Science & Engineering, Dr. Babasaheb Ambedkar College of Engineering & Research, Nagpur, Maharashtra, India

[2]Assistant Professor, Department of Computer Science & Engineering, Dr. Babasaheb Ambedkar College of Engineering & Research, Nagpur, Maharashtra, India

Article Info
Volume 8, Issue 3
Page Number: 152-157

Publication Issue :
May-June-2021

Article History
Accepted : 15 May 2021
Published: 23 May 2021

ABSTRACT

Saline, one of the most popular intravenous (IV) therapy plays a major role in the management of patients who are critically ill. Surveillance of saline bottle level is very important because when the bottle is emptied and the needle is not removed from the vein then the blood flows outward into the bottle. In hospitals, the nurses or caretakers are responsible for monitoring the saline bottle level. Mostly, due to negligence and any unusual condition, the exact timing of removing the needle from the patient's vein is ignored which causes a serious casualty and may lead to death as well. Furthermore, remote monitoring is a need to provide telehealth services. To prevent the accident due to the ignorance of caretakers and to provide remote surveillance in telehealth services, we have proposed the cost-effective smart saline level monitoring device which includes the combination of sensor and Internet of Things (IoT) technologies. We have built this system by using load sensor and ultra-low power low cost ESP32 WiFi System on Chip (SoC) microcontroller. The load sensor converts the weight of the bottle to a specific voltage. The ESP32 microcontroller generates and publishes a specific message based on the voltage received from the sensor. To publish and present the messages to the devices (e.g. smartphone, tablet, laptop etc.) of subscribers like doctors, nurses or caretakers, we have used MQTT-S publish/subscribe protocol which runs over TCP. This proposed monitoring system fulfills the reliable delivery of messages to the subscribers which is very important for healthcare.

Keywords : Intravenous, Internet of Things, System on Chip, MQTT-S

I. INTRODUCTION

The latest report of Global Health Observatory (GHO) data on the density of physicians per population states that globally the ratio of physicians is less than 1 per 1000 persons [1]. Building smart healthcare [2] including tele-health is a need so that the care must be reachable. In order to make the healthcare system smart, it is required to automate the function of diagnosis, treatment, management,

Copyright: © the author(s), publisher and licensee Technoscience Academy. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License, which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited

and decision, so that the services are available both for rural and urban people.

One of the important challenges related to the management of healthcare is to watch the saline level. Almost in all hospitals, a caretaker/nurse is responsible to keep an eye on the saline level and if they fail to monitor this, it is the patient who suffers. Saline bottle when emptied and if the needle is not removed from the patient's vein then due to the pressure difference, the blood flows outward into the bottle which may lead to serious casualty. So, it is the need to automate the surveillance in order to prevent such accident. Further, long distance monitoring by the clinician is also a requirement in telehealth [3] services.

Many authors [4], [5], [6] have addressed the above-discussed problem with the alert notification system. The authors [4] have used the buzzer sound to alert the nurse. Buzzer sound alert system creates noise which is not suitable for the hospital which requires a soundless environment. Moreover, this buzzer system is not expandable for telehealth. On the other side, the authors of [5], [6] have used SMS messaging system for sending an alert message to the nurses mobile number. This system has used GSM (Global System for Mobile Communications) technology [7], which is an open cellular technology, for transmitting data services.

However, the authors [8] have discussed that accessing cellular data is more expensive than WiFi. WiFi is a local area network (LAN) run in a local environment or in a distributed setting. WiFi network protocol is one of the leading communication technology used in the IoT world which supports low transmit power along with low cost [9]. For providing the cost-effective solution, we have proposed IoT based saline level monitoring system using ultra-low power low-cost WiFi technology. The components, we have used to build this system includes a load sensor [10] and ESP32 WiFi chip [11]. The load sensor converts the current weight of the saline bottle into a specific voltage. The ESP32 WiFi chip receives continuous voltage level from the load sensor and publishes notification message to the nurse station, doctor, nurse, caretaker etc., once the specified threshold saline level is reached. The application layer is responsible for message for-matting and publication. Hyper Text Transfer Protocol (HTTP)

[17] is a commonly used application layer protocol on the Internet for message formatting and presentation. However, the authors [9] have commented that the HTTP protocol is not suitable for resource-constrained environments because it arouses a large parsing overhead. They have discussed two alternative application layer protocols such as Constrained Application Protocol (CoAP) and Message Queue Telemetry Transport(MQTT) which have been widely used for IoT appli- cations. CoAP [13] incorporates optimizations for constrained application environments. MQTT[13] is a lightweight protocol and is designed for ultra-low power and low-cost devices. It is based on topic-based publish/subscribe [12] protocol. In topic-based publish/subscribe protocol, the clients act as publish-ers/subscribers and the server acts as a broker. The clients are connected to the broker via TCP which confirms the reliability of message delivery. The communication among publishers and subscribers takes place through the broker. The broker is responsible for coordinating the subscriptions. Clients can publish or subscribe to a topic name. However, MQTT uses text for topic names which increases its overhead and so MQTT-S protocol [14] is introduced. In MQTT-S, the topic names are replaced by topic IDs which reduces the overhead of transmission. Furthermore, MQTT-S protocol also supports buffering of published messages which can be read later by the offline client. This feature provides guaranteed delivery of the published message. Due to the added advantages of MQTT-S, we have utilized this application layer protocol for message formatting and presentation.

Section II presents the overview of the proposed system. Section III gives the architecture of the proposed system. In section IV, we have presented the algorithm which describes the communication among the components of the proposed system. Finally, section V gives the conclusion.

II. PROPOSED SYSTEM

The system is composed of four components: *Load Sensor*, *ESP32 WiFi Chip*, *MQTT-S Broker/Server* and *MQTT-S Client*. The load sensor is fixed on saline hanger and bottle is hung on it. This sensor converts the varying weight of the bottle into different voltages. So, each level of saline weight corresponds to some specific voltage. The output voltage from the load sensor is fed in the ESP32 WiFi chip. Out of the continuous input voltage received, when the specific voltage of interest is obtained, ESP32 will produce the suitable output message such as FULL, ALERT and CRITICAL. The messages are generated under the following conditions.

1) FULL, when the saline bottle is full.
2) ALERT, when the liquid level is reduced to 50%.
3) CRITICAL, when only 10% of the liquid level is left in the bottle.

The generated output message is then published through WiFi network. The proposed system supports asynchronous communication between ESP32 and end devices of doc- tor, nurse or caretaker like the smartphone, tablet, laptop etc. To obtain the asynchronous communication, we have adopted MQTT-S lightweight IoT protocol. This protocol is based on publish/subscribe paradigm [12]. Publish/Subscribe paradigm supports asynchronous messaging system in which the provider(called publisher) of information is decoupled from the clients(called subscriber) of that information. The communication among the provider and clients takes place through the broker.

In this system, ESP32 WiFi chip acts as MQTT-S publisher and devices such as the smartphone, tablet, laptop etc. are acting as MQTT-S clients/subscribers. MQTT-S broker is deployed on such configured machine which can handle any number of MQTT-S publishers and clients. MQTT-S clients are connected to the broker via TCP. When ESP32 publish a message, the MQTT-S broker will receive the message which is then forwarded to the subscribed MQTT-S clients. The general process of the proposed system is shown in figure 1. In the next section, we will describe each of the componentsin detail.

III. SYSTEM SPECIFICATION

In the proposed system, by the weight of the saline bottle, the level of liquid can be estimated so that when the liquid reaches to its minimum level, ESP32 WiFi module is used to send alerts to end subscribers via MQTT-S communication protocol. Below we describe the working of each module:

Load Sensor: It is comprised of a load cell[10] and a HX711 [18] chip. The load sensor works on the principle that on applying a mechanical load, the piezoelectric material of strain gauge in the load cell deforms that leads to change in resistance and thereby the output voltage will change. So, the load cell in a sensor acts as a transducer that converts load into measurable electrical output. This analog output is fed to the HX711 chip which amplifies and converts the analog voltage into a pulse width modulated (PWM) digital voltage. Further, the amplified output voltage from the HX711 chip is fed in one of the GPIO input pin of the ESP32 chip shownin figure 2.

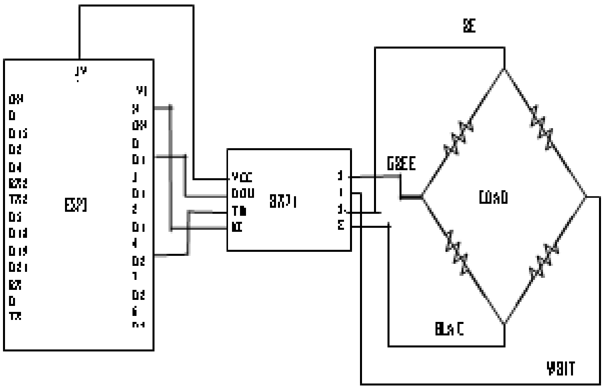

Fig. 1. Circuit Diagram

ESP32 WiFi Chip: This is a well known low-cost microcontroller having Wifi embedded on the chip. The WiFi is based on the IEEE 802.11 b/g/n standard, which operates in the gigahertz band (2.4 GHz) with data transfer rate upto 150 Mbps [11]. The continuous pulse width modulated digital voltage generated by HX711 is received on ESP32 GPIO input pin. Depending on the weight of the bottle, the load sensor will produce corresponding PWM voltage. The maximum capacity the load cell can take is 750g, and for this weight the HX711 chip will produce the value of 255 PWM voltage. In case of the saline bottle, generally the weight ranges up to 500g. We have observed that the load cell generates the following PWM output.

a) For 500g, the PWM value is 170.
b) For 250g, the PWM value is 85.
c) For 50g, the PWM value is reduced to 17.

We have programmed ESP32 in such a form that it will generate the suitable message corresponding to the PWM value described in table I. The message is then published through MQTT- S communication protocol to the MQTT-S broker. The MQTT-S broker further broadcast the received published message to all the subscribed MQTT-S clients such as the smartphone, smartwatch, tablet etc.

TABLE I
NOTIFICATION MESSAGE CORRESPONDING TO THE PWM VALUE

S.No	Weight	PWM Voltage	Message
1	500g	170	FULL!!
2	250g	85	ALERT!!
3	50g	17	CRITICAL!!

MQTT-S: It is a lightweight protocol specially designed for low power and low-cost devices [14]. It is based on topic-based publish/subscribe paradigm as shown in figure 3. In this paradigm, one node acts as a broker while all others as a client. Each client can either be a publisher or a subscriber and should be connected with a broker. Clients can publish or subscribe message with respect to any topic. The topic is identified by the unique identification number [15] named *topicID*. When a publisher (in this case ESP32) publishes, the message is sent to all the other clients subscribed to that topicID via a broker.

In this work, we have generated topic identification number by applying SHA-1 cryptographic hash function on *ESP32 WiFi MAC address* and *barcode of salinebottle* which is shown in equation 1.

$topicID = sha(ESP\ 32\, WiFiMACaddress, salinebarcode)$ (1)

ESP32 WiFi MAC address is the MAC address of the ESP32 WiFi chip in which the load sensor is connected and *barcode of the saline bottle* is the number used to uniquely identify the saline bottle. The value generated by hashing will be 160-bit and is represented by a hexadecimal number for example, f7ff9e8b7776543454 23bcedfdea34523432abf0. In MQTT-S, topics are preregistered and do not need registration [9].

Below, describe the components of MQTT-S utilized in this paper.

MQTT-S Broker: When ESP32 publishes a mes- sage for saline level with certain *topicID*, the MQTT-S broker acts as a mediator which can receive that message and further forward it to the clients subscribed to that topicID. In this work, we have considered a machine as the broker which can handle both publishers and subscribers and is also able to buffer the currently published mes- sages. Buffering of published messages is useful for offline clients who can later read that missed messages once they wake up [9].

MQTT-S Publisher/Subscriber: Here, ESP32 ac- companied with the load sensor acts as an MQTT-S publisher whose job is to publish a message ac- cording to the saline level described in table I. On the other side, the subscribers like the smartphone, tablet, LCD display etc. acts as MQTT clients who can subscribe to the preregistered topicID. For receiving the published message, it is required that the subscriber must register itself respective to the topicID on to the broker.

For registration, the subscriber gives its identi- fication number, for example, MAC address of the network interface, and topicID, to the broker. Whenever the broker receives the message, it can send that message related to the subscribed topicID to the registered subscriber, through online. When the subscriber is in offline or sleep state, then also MQTT-S protocol provides a provision that the subscriber can read that missed message from the buffer of the broker.

IV. DISCUSSION

The communications among the above described MQTT-S components are shown in algorithm 1. The procedure named *ESP32 WiFi Chip* acts as MQTT-S publisher runs on ESP32 WiFi microcontroller whose job is to continuous read digital voltage in its GPIO input pin i.e *GPIO Input Pin* received from the HX711 chip. As soon as the voltage reading reaches to the certain level like 170, 85 etc., *ESP32 WiFi Chip* publish the message to the corresponding *topicID* by invoking *Broker.Publish* function of *Broker* procedure.

On the other side, the procedure *Broker* which runs on the server machine accepts publication and subscription by *Publish* and *Subscription* procedure respectively. The pro- cedure *Subscriber* runs on each MQTT-S client's machine whose job is to display the published message. However, for receiving the publication related to certain *topic* from the *Broker*, the subscriber must register itself by invoking *Broker.Subscription* procedure of *Broker* along with pass- ing two parameters *myID* and *sTopicID* which represents subscriber identification number and the topic of its interest respectively.

After experimentation, we have observed that ESP32 Wifi chip generates the messages recorded in table I. Once the message is generated, the publishing process running in ESP32 WiFi chip publishes the message which will be buffered in the *Broker* represented by $buffer[i][j]$ where, $i \epsilon message$ and $j \epsilon topic\ ID$ (shown in algorithm 1). Due to buffering of published messages on *Broker*, it is possible that the subscriber can also read the message after waking up from the sleep state or when online after offline. For reading the message from the buffer, the *Subscriber* invokes the proce- dure *Broker.ReadBuffer* which in turn writes the message by calling procedure *Subscriber.Write*. This procedure then displays the message to MQTT-S client who was offline.

V. CONCLUSION

In this paper, we have proposed a cost-effective smart saline level surveillance system by which the level of the saline feeding to the patient can be monitored remotely by the nurse, caretaker, hospital staff, doctor etc. We have adopted MQTT-S protocol as it is efficient for low cost and low power devices. Furthermore, MQTT-S also provides guaranteed delivery of messages as it supports asynchronous communication using buffering of messages. We have believed that using this proposed monitoring system one can monitor the level of the saline bottle from a distant position which will aid in building smart healthcare system.

VI. REFERENCES

[1]. GHO: http://www.who.int/gho/health workforce/physicians density/en/

[2]. Lei Yu, Yang Lu & XiaoJuan Zhu, Smart Hospital Based on Internet of Things, Journal Of Networks, Vol 7, No. 10, 2012.

[3]. Telehealth: www.cchpca.org/what-is-telehealth

[4]. Priyadharshini.R, Mithuna.S, Vasanth Kumar.U, Kalpana Devi.S, Suthanthira Vanitha.N., Automatic Intravenous Fluid Level Indication System for Hospitals, International Journal for Research in Applied Science & Engineering Technology, Vol 3, Issue VIII, 2015.

[5]. S. Gayathri & C.S. Sundar Ganesh, Automatic Indication System Of Glucose Level In Glucose Trip Bottle, International Jornal of Multidisci- plinary Research and Modern Education, Vol 3, Issue 1, 2017.

[6]. Ram Kumar S, Saravana Kumar S, Sukumar M, Remote Monitoring The Glucose Bottle Level In Hospitals, International Conference on Emerging Trends in Applications of Computing, 2017.

[7]. GSM: https://www.gsma.com/aboutus/gsm-technology/gsm.

[8]. WiFi: https://www.verizonwireless.com/articles/whats-the-difference- between-wifi-data-and-cellular-data/

[9]. Pallavi Sethi and Smruti R. Sarangi, Internet of Things: Architectures, Protocols, and Applications, Journal of Electrical and Computer Engi- neering, 2017

[10]. I. Muller and R. M. de Brito and C. E. Pereira and V. Brusamarello, Load cells in force sensing analysis – theory and a novel application, IEEE Instrumentation Measurement Magazine, 13, 1, 15-19, 2010.

[11]. ESP32: https://www.espressif.com/

[12]. Eugster, Patrick Th. and Felber, Pascal A. and Guerraoui, Rachid and Kermarrec, Anne-Marie, The Many Faces of Publish/Subscribe, ACM Comput. Surv., vol 35, June 2003.

[13]. D. Locke, MQ telemetry transport (MQTT) v3. 1 protocol specification, IBM developerWorks Technical Library, 2010, http://www.ibm.com/developerworks/webservices/library/ws- mqtt/index.html.

[14]. Hunkeler U., Truong H.L., and Standford-Clark A., MQTT-S - a pub- lish/subscribe protocol for wireless sensor networks, in Proceedings of the 3rd IEEE/Create-Net International Conference on Communication System Software and Middleware (COM-SWARE 08), pp 791-798, Bangalore, India, January 2008.

[15]. Standford-Clark A., and Linh Truon H., MQTT for sensor networks (MQTT-S) protocol specification, International Business Ma- chines Corporation Version 1, 2008.

[16]. Eastlake, 3rd, D. and Jones, P., US Secure Hash Algorithm 1 (SHA1), RFC Editor, United States, 2001.

[17]. Berners-Lee, T., Fielding, R. and H. Frystyk, Hypertext Transfer Protocol– HTTP/1.0, RFC 1945, May 1996.

[18]. D. P. Utomo et al., ”CLC (Cellular Lightweight Concrete) brick making process using neural network and extreme learning method based on microcontroller and Visual Studio.Net,” 2017 International Symposium on Electronics and Smart Devices (ISESD), Yogyakarta, 2017, pp. 79-84

Cite this article as :

Aman S. Ramteke, Pranali A. Pardhi, Shivani N. Gandlewar, Khushbu C. Ingale, Dhawal Dongre, Ms. N. D. Sonwane, "IoT Based Smart Saline Level Monitoring System ", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 8 Issue 3, pp. 152-157, May-June 2021.

Journal URL : https://ijsrset.com/IJSRSET218330