Smart Embedded-Analytics Sensors with Cloud-Based Measurement System for HVAC

N. S. M. Aseri[1, a)], N. Ali[1, 2, b)], M. N. M. Yasin[1], C. B. M. Rashidi[2], S. A. Aljunid[2], R. Endut[2], N. A. M. Ahmad Hambali[1] and M. I. Shapiai[3]

[1]*Semiconductor Photonics & Integrated Lightwave Systems (SPILS), School of Microelectronic Engineering, Universiti Malaysia Perlis, Pauh Putra Main Campus, 02600, Arau, Perlis, Malaysia.*
[2]*Advanced Communication Engineering, Centre of Excellence, Universiti Malaysia Perlis, 02600 Arau, Perlis, Malaysia.*
[3]*Department of Electronic Systems Engineering, Malaysia-Japan International Institute of Technology Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia*

[a)]Corresponding author: aida7002@yahoo.com
[b)]norshamsuri@unimap.edu.my

Abstract. HVAC system is a necessary component of environment to maintain the temperature and humidity to be kept at certain levels by using air taken from outside to ensure the indoor comfort. The purpose of the study is to reduce the electricity energy usage and cost from air conditioning by using smart embedded-analytics sensors to provide the automatic thermal control in an area. In this study, we used sensors such as temperature and humidity sensors to detect and read the currently temperature and humidity of an area monitored by a microcontroller. The cloud-based system and the sensors are connected via wifi in the presence of MQTT protocol. The protocol enables publish and subscribe method which provide the communication between sensors, cloud-based system and HVAC system. This communication can serve thermal control automatically thus resulting the optimize usage of energy from air conditioning according to the external environment temperature and humidity. The control of the temperature and humidity from air conditioning can be designed through the programming embedded in the microcontroller. The monitoring result can be displayed from the control panel to ensure how the system works.

INTRODUCTION

The purpose of heating, ventilating and air conditioning (HVAC) system are to provide comfort to the occupants in buildings [1]. The electricity energy consumed from this system can reach up to 50% in commercial and residential buildings [2]. This shows that the system is the major source of the energy consumption and the essential energy reduction is in demand [3]. It is important to further the research and development to increase the efficiency of energy used by HVAC system [3]. There were some approaches have been proposed previously to improve the energy saving while maintaining the occupants' comfort. However, these approaches could not be performed efficiently due to some limitation [1]. There were two categories of approaches been introduced, namely static control and dynamic control. Static control works at a fixed threshold value or time triggered value. The fresh air is injected to the specific area with variety of ventilation rate but this process resulting high energy consumption and increase humidity which affecting the indoor air quality. Dynamic control works with related data to adjust the ventilation rate accordingly by using the wireless sensors network technology integrated with context driven services. Dynamic control is better than static control as the approach resulting the most energy saving especially in building such as meeting areas or auditorium [1]. Generally building smart automation system can reduce the energy consumed by HVAC since the comfort requirements can be controlled and scheduled automatically according to environment factors. The smart automation system works by sensing and measuring the environment factor and optimizing the control based on the current environment [4].

The 2nd International Conference on Applied Photonics and Electronics 2019 (InCAPE 2019)
AIP Conf. Proc. 2203, 020050-1–020050-7; https://doi.org/10.1063/1.5142142
Published by AIP Publishing. 978-0-7354-1954-4/$30.00

According to the previous works, most of the approaches and the idea shared the same objectives which are to increase efficiency of energy consumption. One of the approaches introduced by Lachhab *et al.* [1] is by maintaining the indoor carbon dioxide (CO_2) concentration at the comfort setpoint with a good ventilation rate to keep the indoor air quality and reduce the energy consumption. The study compared the CO_2-based state feedback control with proportional-integral-derivative (PID) controller and ON/OFF controller and validated the performance by using Building Controls Virtual Test Bed (BCVTB) co-simulation framework with building environment parameter. The result from the simulation showed that the CO_2-based state feedback control is able to maintain CO_2 concentration in the comfortable zone and reduce the energy consumption better than PID controller and ON/OFF controller. The other approach introduced by Rahwan *et al.* [4], an occupancy-predictive HVAC control system in a low-cost embedded system is done to demonstrate the real time occupancy recognition using video processing and machine learning technique, analyze and predict the occupancy patterns and model predictive control for HVAC operations by real-time building thermal response simulation by using EnergyPlus simulator. The approach applied in the large public indoor such as mosque and resulting the energy saving. Another approach introduced by Guo *et al.* [5], is the IoT-based temperature and humidity monitoring system in the hospital. The collected sensor data about indoor humidity and temperature is submitted through wireless sensor network to the IoT gateway which later on uploaded through wifi or wired access network within the hospital to the background server. Once the report and control commands generated, the hospital staff can make real time control on the environment data by connecting their device (phone) to the server. The study shows the system success to run steadily with accurate data and ensure reliability. The approach introduced by Png *et al.* [6], the study present the IoT prototype which implement Smart-Token Based Scheduling Algorithm (Smart-TBSA) to reduce the energy from HVAC system in commercial building. It upgrades the legacy BAS in building to decentralize HVAC control.

The era of Internet of Things (IoT) been introduced and used widely nowadays which conclude the communication between physical and digital world through sensors and actuators [7]. The size of IoT is small that make them suitable to be attached to different type of sensors and actuators [8]. The example of IoT application include smart home (eg: turn on AC when humidity increase), smart city (eg: traffic light works when the vehicle is detected), smart healthcare (eg: view blood pressure through smart watch) and others [9]. IoT works when sensors collect the data such as temperature and humidity and send the data to the database server or display the data on the application interface [10]. IoT also allow the data gathering and data exchange among the connected devices using internet. These devices normally connected with microcontroller such as Arduino [11]. The IoT make tasks easy to monitor and collect the sensor data via internet [12]. Thus, the high quality data is needed to ensure the real time monitoring system with the compatible communication protocol. There are a lot of communication protocol developed for IoT and one of them is Message Queuing Telemetry Transport (MQTT) protocol [10]. MQTT is chosen because it only required small code footprint and easy to scale the sensor nodes. This is very suitable to be used since it is also require less bandwidth, real time response and low energy use because it is commonly used for small application [13]. The MQTT is chosen over the other IoT protocol such as CoAP because it support smartphones as client as the system interfaced on mobile based or web based application [10]. The concept of publisher and subscriber are used in this communication protocol which allow multiple devices to communicate with each other through wireless network. Message is being published or subscribed with the help of broker (server) in MQTT. Examples of broker in MQTT are Mosquitto, CloudMQTT, Adafruit and so on [11]. MQTT protocol is used in the top most layer Transmission Control Protocol/Internet Protocol (TCP/IP) which is Application layer. It is not necessary for both of publisher and subscriber to know each other as the MQTT send the data from a source to a destination implemented on the application layer [5]. Both of publisher and subscriber known as MQTT clients. There are always two types of agents in MQTT connection namely as MQTT client and MQTT broker. MQTT client refers to the devices connected to the network and take part in communication via MQTT [11]. MQTT broker is the central node where each MQTT clients need to refer to its broker to exchange message with other clients. The broker always comes with a topic [14]. A client can send message on a topic (publisher mode) or listen to one or more topics at the same time to get instant benefit from the message (subscriber mode). A publisher can send message and only subscriber who subscribe the topic can get the information. The broker allows the different clients to connect with each other. The broker responsible to acknowledge and send the message among the different clients associated with it [11].

There are so many ways to implement MQTT protocol as device as it works in the application layer which commonly supported by microcontroller such as Arduino, Raspberry Pi, ESP32 Wi-Fi module and so on. MQTT client needed to be installed on the devices [10]. The microcontroller such as ESP32 Wi-Fi module is a low power 32-bit CPU based where it contains its own system on chips with integrated Wi-Fi and dual mode Bluetooth. The module completed with integrated TCP/IP stack used as the application processor and allow the connected device to communicate via wireless network this module is chosen as it is powerful on-board processing and the storage

capabilities allow the devices such as sensors to interface each other through General-purpose input/output (GPIO) in minimal loading during runtime [8]. This ESP32 module is been used widely because of the Wi-Fi function provided [13-15]. Wi-Fi based automation utilize the localized system to manage the connected devices. The arrangement is a solution as they require complicated network traffic routing to operate [8]. Thus, this make up the reason of Wi-Fi is chosen over Bluetooth as the devices need to be connected through wire or built-in Bluetooth without internet connectivity. Bluetooth operates at a maximum range of about 100 m and this limit the system to cover long distance range [8].

METHODOLOGY AND STRUCTURE

In this paper, the setup is illustrated as shown in Figure 3. The temperature and humidity sensor are used to detect the temperature and humidity of the current environment. Temperature and humidity are needed as they are the common parameters to measure the environment condition. The sensor data is then processed and controlled by the ESP32 microcontroller via Arduino IDE software. At the same time, ESP32 microcontroller is connected to a MQTT broker named as CloudMQTT via wifi. The MQTT software has been installed in the ESP32 microcontroller to gain the MQTT protocol functionality. Equipment such as ESP32 Wi-Fi Module, DHT22 Temperature and Humidity Sensor, CLoudMQTT as MQTT broker or android smartphone are needed to implement the MQTT protocol usage.

For the temperature and humidity sensor, we use Si7021 sensor as the sensor in this project. It is widely used for HVAC applications and offer high accuracy temperature sensor in the range of up to 85□ with 0.4□ accuracy and precision Relative Humidity sensor in the range of up to 100% with 3% accuracy. The Si7021 requires supporting circuit in order to make use of it. However, the Si7021 is very tiny where the size is just 3x3mm and it is not suitable to be used as the prototype testing. Thus, we come up with solution to use DHT22 for the prototype since its features are closely related to Si7021. The DHT22 offers temperature range up to 125□ with 0.5□ accuracy and humidity range up to 100% with 2-5% accuracy, which is better than DHT11 that offers temperature range up to 50□ with 2□ accuracy and humidity range up to 80% with 5% accuracy. For Si7021 usage, we prepared the circuit for which will be used later once the prototype result is a success.

Si7021 sensor preparation

The Si7021 sensor schematic diagram is designed in the Altium software as shown in Fig. 1. Figure 2 depicts the PCB designed which is implemented by using the pin of ESP32 module. It will be two-sided PCB combined together to save the space usage.

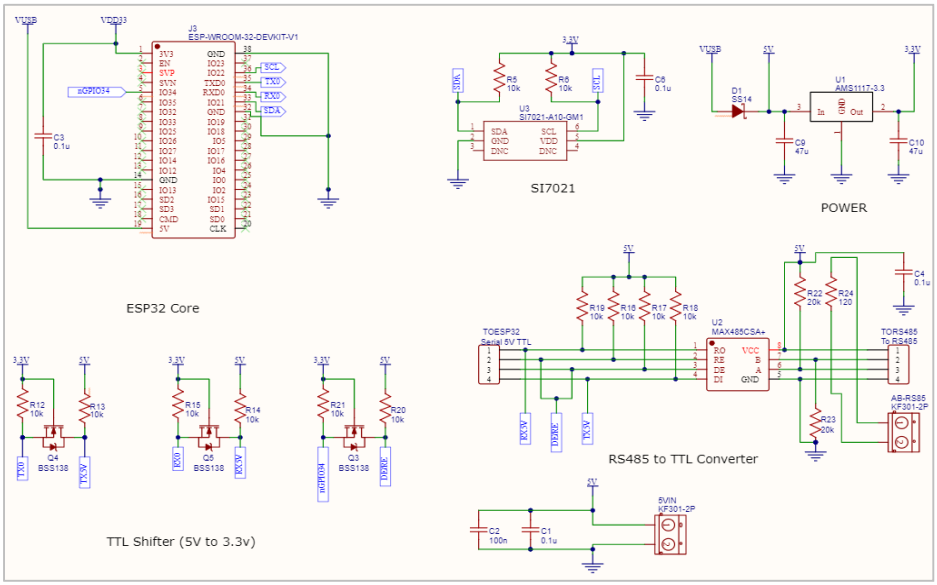

FIGURE 1. The schematic diagram of Si7021 sensor.

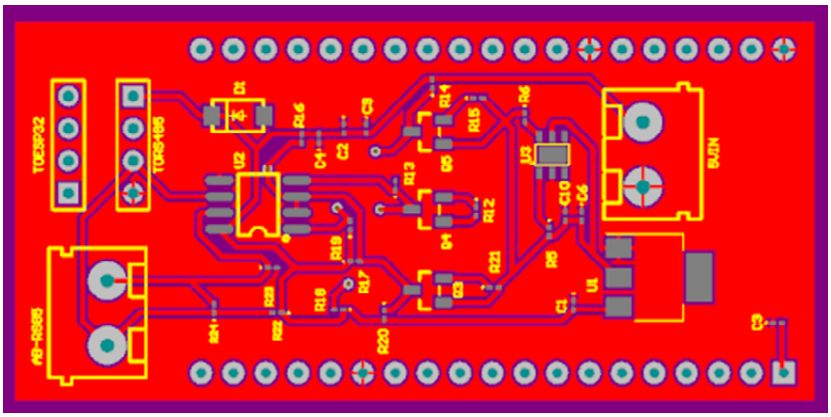

FIGURE 2. The PCB design view for SI7021 sensor.

Overview of overall prototype system

Since the Si7021 sensor is replaced with DHT22, thus the overall system is concluded as illustrated in Fig. 3 where ESP32 module is connected to DHT22 sensor. The program control for the whole project is designed in Arduino IDE software installed in the ESP32 module. Then the MQTT client such as sensors will publish the real time sensor data of the environment to the MQTT broker. The MQTT broker used is CloudMQTT which is an open source broker application of MQTT. The CLoudMQTT will then publish the desired sensor data to the MQTT client such as web application, smartphone or other devices who subscribe the topic published.

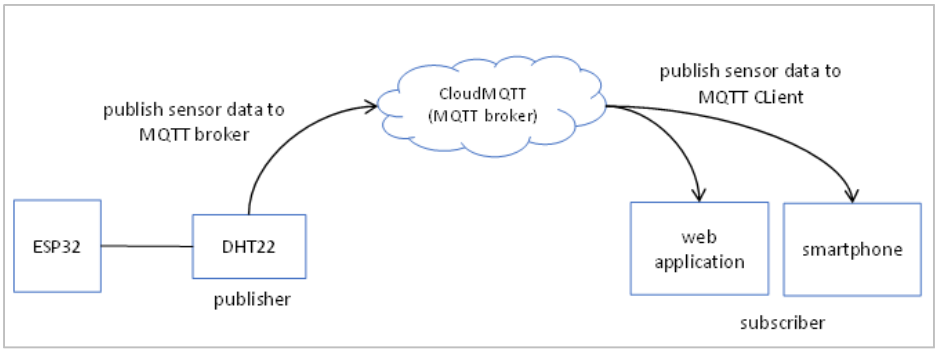

FIGURE 3. The prototype overview of the system.

Hardware Implementation

The hardware implementation for the project prototype only involve ESP32 Wi-Fi Module and DHT22 temperature and humidity sensor. The DHT22 has three pins named as DATA, GND and VCC. The DATA pin is connected to any GPIO pin from ESP32 as this is where the data will be read from the sensor. The ground pin of DHT22 is connected to the ground pin of ESP32. The VCC pin of DHT22 which operates at 3.3V is connected to the 3.3V pin of ESP32. The ESP32 gets the power through USB cable connected to the PC as shown in Fig. 4.

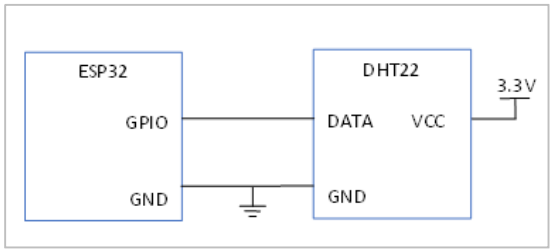

FIGURE 4. The physical connection of ESP32 module with DHT22 sensor

System Implementation

DHT22 serves as the publisher to transmit the current sensor data (temperature and humidity) to MQTT broker (CloudMQTT). The DHT22 measures the environment temperature and humidity. CLoudMQTT is the MQTT broker that responsible for the data exchange between publisher and subscriber or among mobile application, web application or hardware devices.

The software connection of the wifi, MQTT software and sensor are designed in Arduino IDE software where each of the software has been installed in ESP32 module, respectively. Internet is needed to connect the Wi-Fi Module where can use the mobile hotspot data. Table 1 shows the parameters needed for connection.

TABLE 1. Parameter needed for Wi-Fi connection.

Parameter	Data
Wi-Fi username	Nova 3i
Wi-Fi password	1234

To connect MQTT broker to the PC, the parameter shows in Table 2 below is required. These parameters and data can be obtained through the MQTT broker website page. Once the connection is established, the function of the publish and subscribe are ready to be used.

TABLE 2. Parameter needed for MQTT connection.

Parameter	Data
MQTT server	m11.cloudmqtt.com
MQTT port	14354
MQTT username	gnvurxxx
MQTT password	xmku-KOxxxx_

To observe the result of the project, a Google Chrome application named MQTTlens is used to test the functionality of the project. This application is connected to MQTT broker by using the parameter from Table 2. This application can subscribe and publish to MQTT topics accordingly.

RESULT AND DISCUSSION

Once the hardware and software implementation are successfully connected to each other, the results can be observed through the terminal. The terminal in this project is PuTTY and the data obtained through COM3 at 15200 baud rates.

The program written in Arduino IDE software is verified and uploaded. Once the program code is uploaded in the microcontroller, the PuTTY terminal displayed the result as shown in Fig. 5. The program will keep looping until the Wi-Fi is connected otherwise the whole connection is failed. Once the Wi-Fi is connected, the MQTT is ready to be connected. The last row shows in Fig. 5 indicates that the MQTT connection is successfully established.

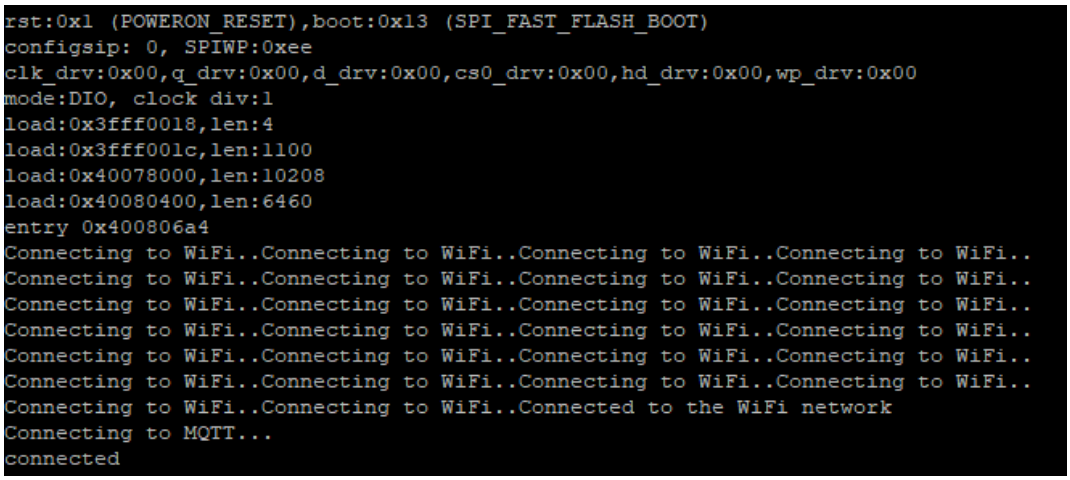

FIGURE 5. The result of the successful connection of Wi-Fi and MQTT.

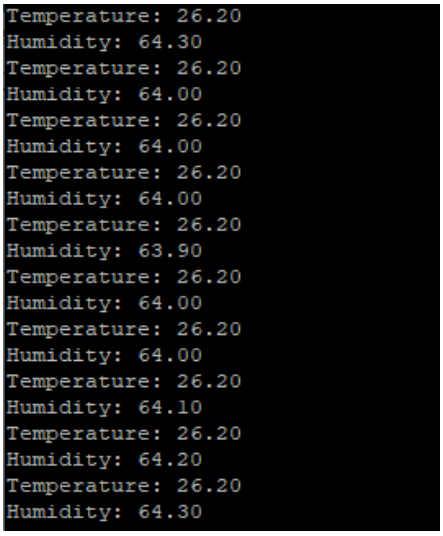

FIGURE 6. The result of the successful temperature and humidity data reading from sensor.

Figure 6 shows the result of the temperature and humidity data reading from the DHT22 sensor. The temperature and humidity sensor keeps collecting the new temperature and humidity reading every second to ensure the real time data collection.

CONCLUSION

In conclusion, the Wi-Fi and MQTT connection are success to gain. The publish and subscribe function is success to be implemented. The objectives of the prototype to get the sensors reading such as temperature and humidity from DHT22 is a success. Further development of the project is to replace the DHT22 with Si7021 sensor board, to connect more devices as the MQTT client, to add the energy meter to observe the energy consumption, to develop programming code for better monitoring and controlling.

ACKNOWLEDGMENTS

The authors acknowledge financial support by the School of Microelectronic Engineering, Universiti Malaysia Perlis (UniMAP). NA also acknowledges financial support by Universiti Malaysia Perlis STG Grant 9001-00494.

REFERENCES

1. F. Lachhab, M. Bakhouya, R. Ouladsine, and M. Essaaidi, "ScienceDirect ScienceDirect Towards an Intelligent Approach for Ventilation Systems Control Towards an Intelligent Approach for Ventilation Systems Control using IoT and Big Data Technologies using IoT and Big Data Technologies," *Procedia Comput. Sci.*, vol. 130, pp. 926–931, 2018.
2. L. Pérez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," *Energy Build.*, vol. 40, no. 3, pp. 394–398, 2008.
3. M. W. Ahmad, M. Mourshed, B. Yuce, and Y. Rezgui, "Computational intelligence techniques for HVAC systems : A review," no. 2016, pp. 359–398, 2020.
4. T. Rahwan, "Automatic HVAC Control with Real-time Occupancy Recognition and," no. November, 2017.
5. B. Guo, X. Wang, X. Zhang, J. Yang, and Z. Wang, "Research on the Temperature & Humidity Monitoring System in the Key Areas of the Hospital Based on the Internet of Things," vol. 10, no. 7, pp. 205–216, 2016.
6. E. Png, S. Srinivasan, K. Bekiroglu, J. Chaoyang, and R. Su, "An internet of things upgrade for smart and scalable heating , ventilation and air-conditioning control in commercial buildings," *Appl. Energy*, vol. 239, no. January, pp. 408–424, 2019.
7. S. Gusmeroli, H. Sundmaeker, and A. Bassi, "Internet of Things Strategic Research Roadmap," pp. 9–52.
8. T. A. Abdulrahman, O. H. Isiwekpeni, N. T. Surajudeen-bakinde, and A. O. Otuoze, "Design , Specification and Implementation of a Distributed Home Automation System," *Procedia - Procedia Comput. Sci.*, vol. 94, no. IoTNAT, pp. 473–478, 2016.
9. P. Sethi and S. R. Sarangi, "Internet of Things : Architectures , Protocols , and Applications," vol. 2017, 2017.
10. H. Search, C. Journals, A. Contact, M. Iopscience, and I. P. Address, "IoT real time data acquisition using MQTT protocol," vol. 012003.
11. M. Kashyap, V. Sharma, and N. Gupta, "Taking Taking MQTT MQTT and and NodeMcu NodeMcu to to IOT : IOT : Communication Communication in in Internet Internet of of Things Things," *Procedia Comput. Sci.*, vol. 132, no. Iccids, pp. 1611–1618, 2018.
12. A. Zanella *et al.*, "Internet of Things for Smart Cities," vol. 4662, no. c, 2014.
13. K. Chooruang and P. Mangkalakeeree, "Wireless Heart Rate Monitoring System using MQTT," *Procedia - Procedia Comput. Sci.*, vol. 86, no. March, pp. 160–163, 2016.
14. A. Schmitt, "ScienceDirect Dynamic Dynamic bridge bridge generation generation for for IoT IoT data data exchange exchange via via the the MQTT MQTT protocol protocol e," 2018.
15. S. A. Z. Murad, S. N. Mohyar, A. Harun, M. N. M. Yasin, I. S. Ishak and R. Sapawi, "Low noise figure 2.4 GHz down conversion CMOS mixer for wireless sensor network application," 2016 IEEE Student Conference on Research and Development (SCOReD), Kuala Lumpur, 2016, pp. 1-4.