INTERNATIONAL JOURNAL OF INTEGRATED ENGINEERING
ISSN: 2229-838X e-ISSN: 2600-7916

IJIE

Vol. 16 No. 3 (2024) 78-91
https://publisher.uthm.edu.my/ojs/index.php/ijie

IoT-Based Embedded System for Streamlined Thermal Comfort Data Collection in Buildings

Farzana Sharmin Nila[1], Wooi-Haw Tan[1], Chee-Pun Ooi[1]*, Yi-Fei Tan[1]

[1] *Faculty of Engineering,*
Multimedia University, Persiaran Multimedia, 63100 Cyberjaya, Selangor, MALAYSIA

*Corresponding Author: cpooi@mmu.edu.my
DOI: https://doi.org/10.30880/ijie.2024.16.03.008

Article Info

Received: 28 November 2023
Accepted: 19 February 2024
Available online: 30 April 2024

Keywords

IoT, embedded system, data collection, thermal comfort, MQTT

Abstract

Thermal comfort refers to the process of determining a pleasant working temperature by considering various environmental, occupational, and personal factors. With the widespread adoption of Internet of Things (IoT) technologies, there is a potential to develop an IoT-based embedded system that directly integrates sensor data with building information to collect the necessary environmental factors for thermal comfort. This paper outlines the development and implementation of an embedded system for remote temperature monitoring, which gathers data such as temperature and humidity using sensors installed within the building and transmits this information to the cloud in real-time via a private Message Queuing Telemetry Transport (MQTT) server. An IoT-based embedded system makes the data collection process more efficient, automated, and integrated, which ultimately leads to a faster and smoother experience in gathering and processing the necessary information to optimize thermal comfort. Following data collection, a machine learning model may be trained using the acquired data to automatically adjust the thermal comfort level in the building, improving overall comfort and energy efficiency.

---

1. Introduction

Nowadays, people spend most of their time indoors, significantly impacting their comfort and quality of life. A high level of comfort is associated with productive performance. Unsafe behavior is more likely to occur when a person is uncomfortable due to heat or cold. They become less adept in decision-making and performing tasks manually. Human bodies have a fundamental requirement to thermoregulate or maintain a steady internal temperature within a specified range, which can be challenging if the interior environment is too hot or too cold. Thermal comfort is a mental state assessed subjectively and reflects comfort with the thermal environment [1]. In its simplest form, thermal comfort is the state in which a person does not feel overly hot or cold. As demonstrated by scientific research, the thermal environment is crucial for human health as well as comfort. It is not easy to make everyone comfortable [2].

Nevertheless, it is possible to take steps to ensure that the vast majority of people feel relaxed. Given its complexity, the human thermal condition cannot be explained in terms of degrees. In addition, the permissible variation in temperature remains insufficient to distinguish them. It is a personal feeling that differs from person to person in the same place and is influenced by a variety of variables. It can be challenging to attain thermal comfort. However, it is possible to increase the number of thermally comfortable inhabitants through thoughtful design, construction, and maintenance. An embedded system, a microprocessor-based computer hardware and software system, is developed for a particular task [3]. Embedded systems may operate separately or as part of a large system. An integrated circuit is built to perform processing at the core of the embedded system for real-time operations. For example, embedded systems can be used to monitor a room's temperature through IoT and adjust air conditioning settings. IoT is a concept that enables data transmission across a network without the use of a computer or a human [4]. It has the potential to be a monitoring system, making it advantageous and easier for the user to get information and manage a device [5]. Sensors are crucial for monitoring environmental changes and gathering real-time data for embedded systems. A sensor is an instrument that senses or measures a physical parameter—such as motion, heat, humidity or light and converts it to a functional equivalent analogue or digital representation. It transforms detected physical qualities into an electrical signal that can be measured [6].

2. Related Works

When it comes to embedded system construction and data storage through IoT, several researchers have contributed by bringing forth cutting-edge smart technologies and techniques for gathering environmental thermal comfort parameters such as temperature, humidity, air speed, and other factors. The methods of collecting the environmental data based on previous researchers' works are discussed below.

To provide an appropriate, thermally pleasant atmosphere, the research work of [7] suggested an air-conditioning smart IoT based System. A variety of sensors and modules were included with the smart air conditioner. The suggested design uses the LM 35 temperature sensor, a PIR motion sensor, and an ATmega328 microcontroller that receives temperatures and motion data as input and shows them on a display monitor.

In their work, [5] suggested a temperature monitoring and control system for fermentation. The proposed solution included a thermoelectric actuator, a microcontroller (NodeMCU ESP8266), a DHT22 temperature and humidity sensor as an input, and a Blynk application as the user interface. The suggested data transmission system utilized the IoT at the same time.

[8] described in their paper how to use the IoT and Tsukamoto's fuzzy inference system (FIS) technique to create an effective and efficient AC control system for regulating room temperature and humidity. The DHT22 sensor, directly connected to the ESP32 microprocessor, measured room temperature and humidity. This control system was remotely monitored through IoT and a mobile application interface.

[9] focused on designing and deploying a remote IoT interior monitoring approach. The physical components of the system were an Arduino Uno development board, a DHT11 temperature and humidity sensor, and an MQ-135 gas sensor. A web-based interactive system, ThingSpeak, was also used to display the data graphically.

[10] studied raising healthy chickens, lowering the mortality rate and increasing productivity in Brunei. This setup had three crucial modules: the humidity and temperature sensor, the air quality sensor and the feeder sensor. An Arduino Mega 2560 R3 development board was used to gather thermal data from the temperature and humidity modules. The food concentration and wind quality conditions were collected from the feeder and air quality sensor modules. These data were then sent to a database via a NodeMCU ESP8266.

To enable the automation of home appliances and remote monitoring of home conditions, [11] created a prototype known as IoT@HoMe. The data from the sensors in this system was updated on an Adafruit IO cloud server using a Wi-Fi-based gateway called a NodeMCU. Regardless of the user's location, If This Then That (IFTTT) can be installed on their devices to access data received from a variety of sensors, including temperature, humidity, radio-frequency identification, ultrasonic, gas, and PIR sensors, through IoT.

[12] described a Machine learning based IoT system to identify a model of personal thermal comfort. The system collected temperature and humidity data using DHT11 room sensors, a wearable fitness band, Arduino Mega 2560 and a Raspberry Pi, a smartphone. Additionally, a wind speed sensor was used to record air speed. One of the two gateways in this project, a Raspberry Pi B+, had a connection to the Arduino Mega 2560. This IoT solution provided sufficient precision to increase the accuracy of thermal comfort forecasts by nearly 50%.

[13] used Raspberry Pi 3 to monitor and control humidity and temperature data, leading to automated temperature and humidity management through IoT. The humidity and temperature readings were relayed to the Internet using the Raspberry Pi. This initiative produced a workable automated prototype for temperature and humidity control.

The system suggested by [14] offered a Graphical User Interface (GUI) to monitor and control IoT connected devices through embedded microcontrollers web server called Arduino Yun, temperature and humidity sensors, and motion sensors.

Based on the benefits and comfort provided by IoT technology, [15] put IoT into practice by using a mobile application to regulate and observe the conditions of room temperature. The Flutter framework was used to create the application in [16]. Real-time database access was made possible via the Wi-Fi module included in the ESP32 by [17]. The database developed by Google in use is called Firebase Realtime Database.

Several IoT-based humidity and temperature control systems have been developed in earlier research for a variety of applications, including server rooms [18], greenhouses [19] , thermal systems [20]. IoT applications may make a significant contribution to enhancing customer ease. The outcomes of IoT implementation are quite promising since they may raise user security, safety, intelligence, and comfort, according to earlier studies, which bolster this claim [11].

However, there are some issues that need to be addressed that were not covered by the other research papers, such as security and safety concerns, easy integration, power consumption and verification and assessment, which is essential for enhancing system performance. This study presented the development of an embedded MQTT system and a calibrated sensor for the collection of thermal comfort data effectively using a problem-solving methodology. This system focuses on collecting data from a large number of devices, which is subsequently transmitted to a private MQTT server for further analysis.

3. Methodology

Figure 1 shows the IoT-based embedded system architecture based on a four-layered IoT architecture. IoT architecture is a complex network of components which includes sensing devices, actuators, cloud-based services, regulations, and layering [21]. It is divided into layers to allow administrators access, continuous monitoring, and reliability maintenance. Data from sensor-connected devices and microcontrollers are sent across a network and onto the cloud for processing, analyzing, and storing in this IoT architecture. The IoT is ready to expand with new enhancements, offering users fresh and enhanced experiences.

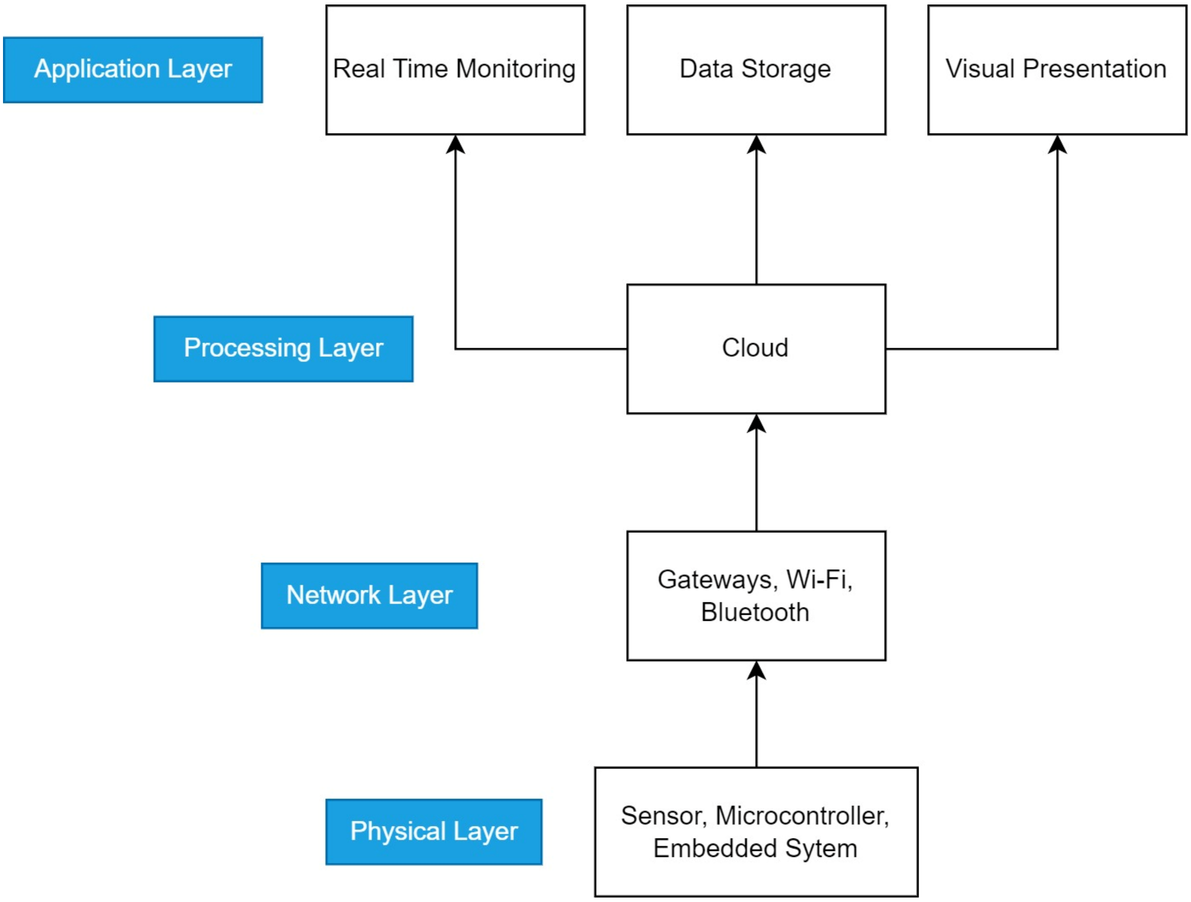


Fig. 1 *IoT-based embedded system architecture*

The physical layer is the base layer of an IoT system. It consists of sensors, microcontrollers, embedded systems and actuators that can gather, accept, and analyze data from the environment through a network [22]. Wireless or wired connections can be used to connect sensors and actuators [23].

The network layer is in charge of moving data sent from the physical layer to the processing layer and vice versa using networks such as Wi-Fi, LAN, Bluetooth, etc. [24].

The processing layer holds, processes, and converts information from the transport layer. It receives unprocessed sensor data and transforms it into information using cloud services and big data modules.

The user-centric application layer is designed to meet the demands for fully automated graphical visualization, real-time monitoring, and statistical analytics. The advent of servers and cloud technology has made it possible to store data effectively [25].

3.1 Proposed Layout

All the experiments were conducted in room A of the Digital Home Lab at Multimedia University (MMU), Cyberjaya, for streamlined thermal comfort data collection. The rectangular-shaped living room was divided into nine grids, according to the layout in Figure 2. Nine setups used for this experiment were placed into these partitions. The room is 5.96 meters in height and 9.86 meters in width. It has two air conditioners (AC) installed solely for the purpose of this experiment. One embedded system was placed in each of the nine partitions to collect the data.

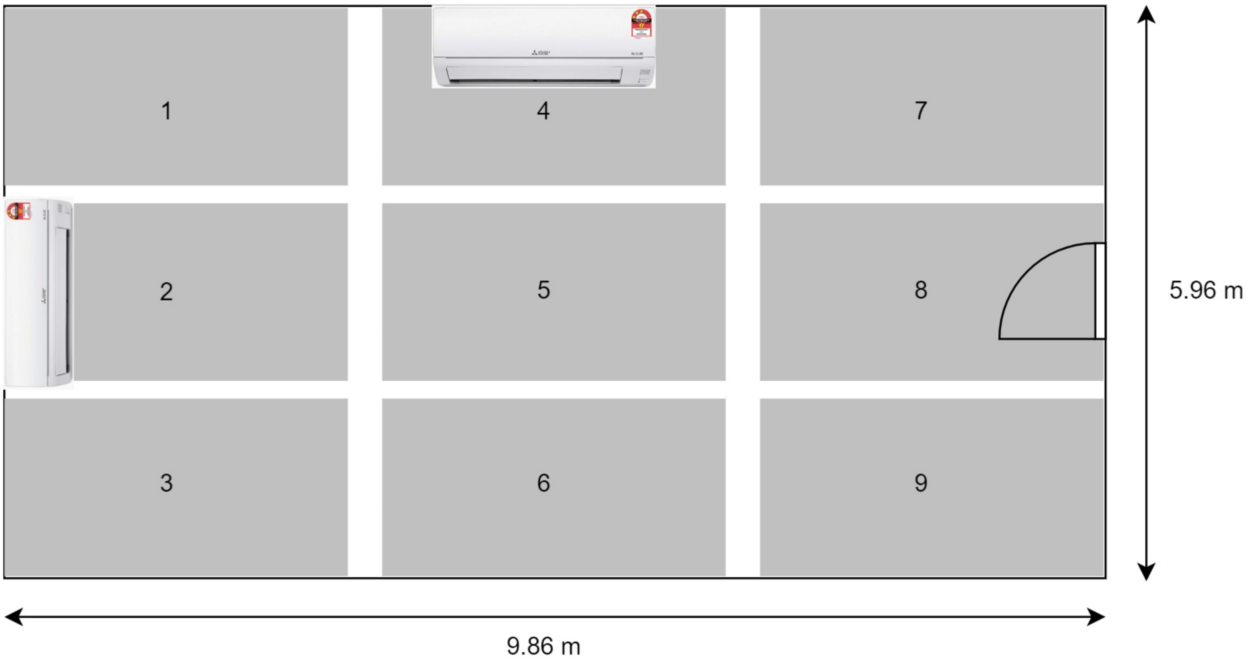


Fig. 2 *Layout of digital home lab*

3.2 Proposed Setup

Figure 3 explains the proposed experimental setup deployed in Digital Home Lab. In this experiment, ESP8266 WEMOS D1 mini development boards, DHT22 temperature and humidity sensors, dual base expansion board and breadboards were employed as components for data collection. Four AAA batteries enclosed in battery holders were used as power sources.

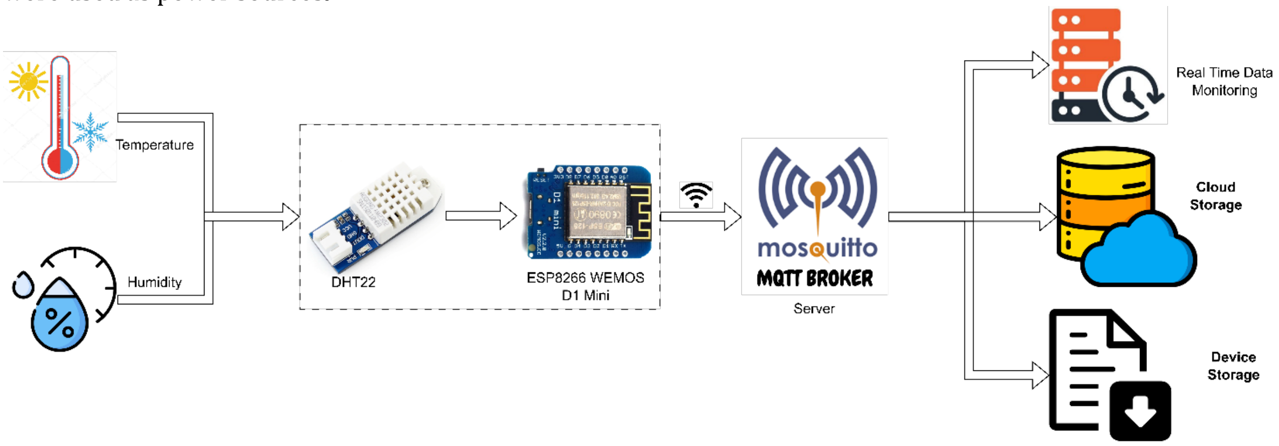


Fig. 3 *Experimental setup of humidity and temperature*

The data-gathering algorithm was uploaded into the microcontroller by connecting the WeMos D1 Mini board to a computer's USB port and running the Arduino IDE programme. Digital Home Lab's temperature and humidity were measured with nine DHT22 sensors interfaced with the WeMos D1 Mini boards. These Wi-Fi-enabled boards were connected to a private and secured MQTT broker using the publish/subscribe communication model to send the data. Prior to data acquisition, all sensor data underwent a process of calibration. The offset was calculated by deducting the sensor values from the reference reading, which was gathered using the UT333BT temperature and humidity meter. The difference between the two values was subsequently incorporated into each sensor reading within the MQTT subscriber code to generate the calibrated value.

The proposed experimental set-up to gather wind speed data from the digital home lab is illustrated in Figure 4. An anemometer, the UT363 BT, was used to measure the wind's speed. This data was then exported from the app to the device as a CSV file.

Fig. 4 *Wind speed experimental layout*

3.2.1 WeMos D1 Mini ESP8266 Microcontroller

Table 1 presents the specifications [26] of the WeMos D1 Mini development board. ESP8266 is a cost-effective System-on-a-Chip (SoC) which serves as a key component of an open-source Node Micro Controller Unit (NodeMCU). It consists of all the essential components of a computer, including the CPU, RAM, networking (Wi-Fi). As a result, this makes it an excellent choice across all kinds of IoT-related projects. The WEMOS D1 mini has a Wi-Fi module and is powered by a 32-bit ESP8266EX microcontroller. It is possible to programme this board using the Arduino IDE or Lua programming language. Over-the-air (OTA) and serial programming are also supported to program the development board.

Table 1 *Technical specifications of WeMos D1 Mini*

Microcontroller	ESP-8266EX
Operating Voltage	3.3Vdc
Digital I/O Pin	11
Analog Input Pin	1
Clock Speed	80MHz/160MHz
Flash	4M bytes
Board Dimension	68.6mm x 53.4mm
Weight	25g

3.2.2 DHT22 Temperature & Humidity Sensor

The DHT22 humidity and temperature sensor is employed in this study. DHT22 is a flexible and economical sensor used to measure ambient temperature and relative humidity for a variety of IoT-based embedded system applications. It measures the humidity and temperature using capacitive humidity detection components and thermistors, respectively. This sensor generates an electrical signal whose strength is proportional to an environmental physical parameter [14] , [27]. Additionally, its presence does not significantly alter the values being measured, allowing for more accurate and reliable measurements. DHT22 sensor has a high sensitivity to the property being measured but maintains a low sensitivity to all other properties. Table 2 lists all the technical parameters of the DHT22 sensor [28].

Table 2 *Technical specifications of DHT22*

Model	DHT22
Power supply	3.3-6V DC
Output signal	Digital signal via single bus
Sensing element	Polymer capacitor
Operating range	Humidity 0-100%RH. Temperature -40~80Celsius
Accuracy	Humidity +-2%RH (Max +-5%RH); Temperature <+-0.5Celsius
Resolution or sensitivity	Humidity 0.1%RH. Temperature 0.1Celsius
Repeatability	Humidity +-1%RH. Temperature +-0.2Celsius
Humidity hysteresis	+-0.3%RH
Long-term Stability	+-0.5%RH/year
Sensing period	Average: 2s
Interchangeability	Fully interchangeable
Dimensions	Small size 14*18*5.5mm. Big size 22*28*5mm

3.2.3 Mosquitto MQTT Broker

Machine-to-machine communication is facilitated by Message Queuing Telemetry Transport (MQTT), a publish-subscribe messaging protocol [29]. Smart sensors, wearables, and IoT devices generally need to send and receive data via a network with constrained resources and capacity. This study utilized Eclipse Mosquitto, an EPL/EDL-compliant open-source message broker that supports MQTT versions 5.0, 3.1.1, and 3.1 [30]. Mosquitto is a bridge that links to other MQTT-based message servers. MQTT communication operates based on the publish and subscribe model. Devices publish messages on a specific topic. The messages are sent to every device that has subscribed to that topic. Its primary uses include publishing and reading data from sensor nodes, sending messages to control outputs, and many more. Mosquitto is portable and adaptable for a wide range of equipment, from large servers to single-board computers with small power requirements.

3.2.4 UT363BT Anemometer

The mini anemometer, UT363BT, shown in Figure 5, is a wind speed and temperature meter that can detect wind speeds up to 30 m/s [31]. This measured data can be transferred through Bluetooth to a customized mobile application of UNI-T (iENV) for further processing, storage, and exporting. Table 3 summarizes all the specifications of UT366BT anemometer.

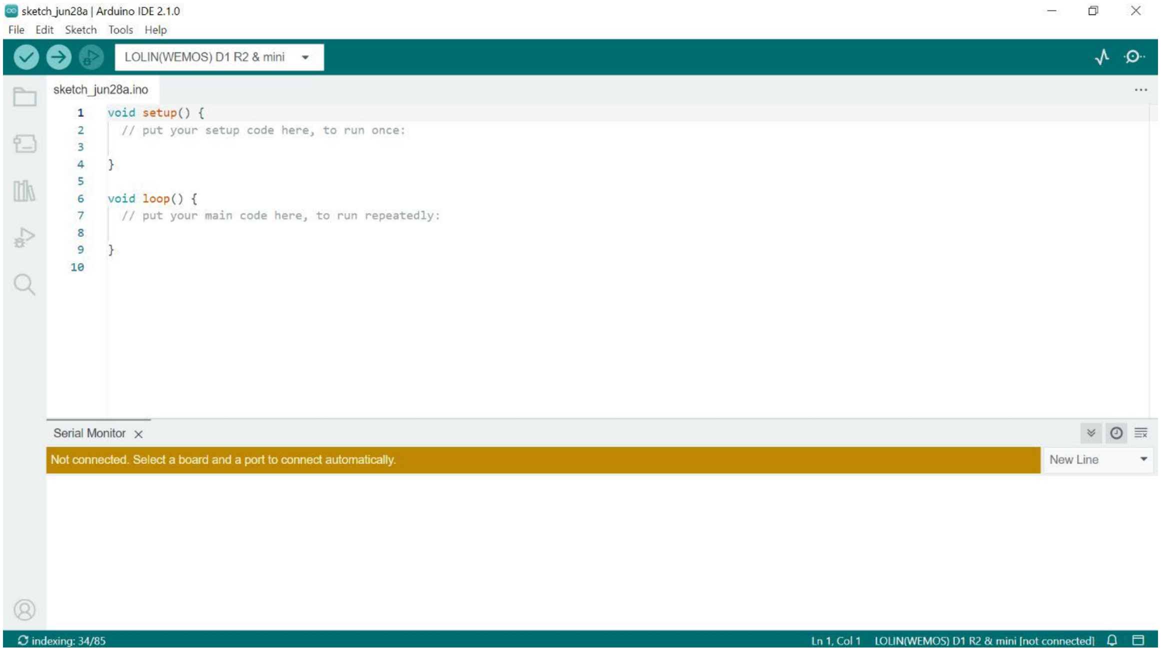
Fig. 5 *UT363BT Anemometer*

Table 3 *Specifications of UT363BT anemometer [31]*

Wind speed	0~30m/s
Wind speed accuracy	±(5%rdg+0.5)
Wind speed resolution	0.1m/s
Temperature	-10~50°C; 14~122°F
Temperature accuracy	±2°C/±4°F
Temperature resolution	0.1°C/0.2°F
Wind scale	Level 0~12
Wind scale accuracy	±1
Sampling rate	0.5s
App	Yes
Bluetooth	Yes
MAX/AVG	Yes
Data hold	Yes
Power	1.5V battery (R03) x 3
Display	32mm x 26mm
Product size	160mm x 50mm x 28mm
Product net weight	118g

3.2.5 Arduino IDE

The Arduino Integrated Development Environment (IDE), as depicted in Figure 6, is a freely accessible software officially released by Arduino.cc. It is mostly used to edit, compile, and upload code to Arduino and NodeMCU devices. It uploads a binary file and loads it through the designated port of the specified board. This software also has a serial monitor which displays serial output transmitted from the Arduino board via the USB or serial port.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}

```

Not connected. Select a board and a port to connect automatically.

indexing: 34/85   Ln 1, Col 1   LOLIN(WEMOS) D1 R2 & mini [not connected]

Fig. 6 *Arduino IDE*

3.3 Experimental Setup

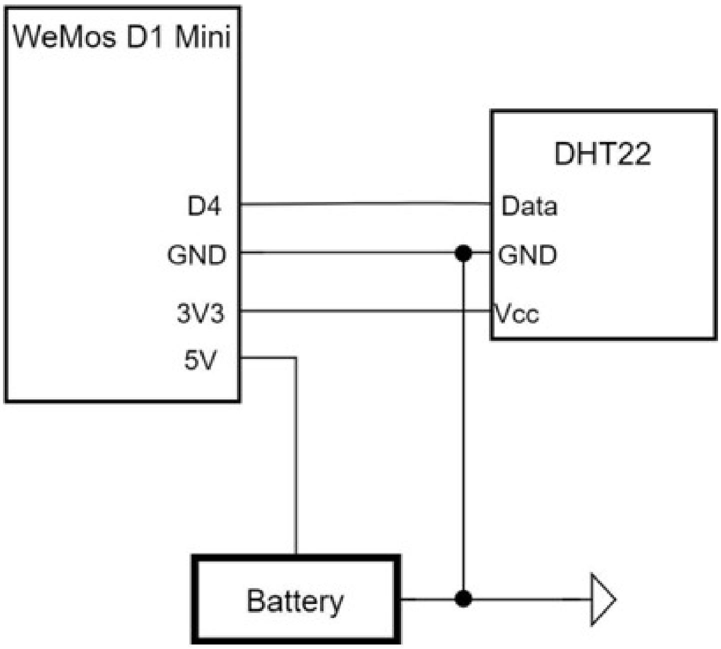
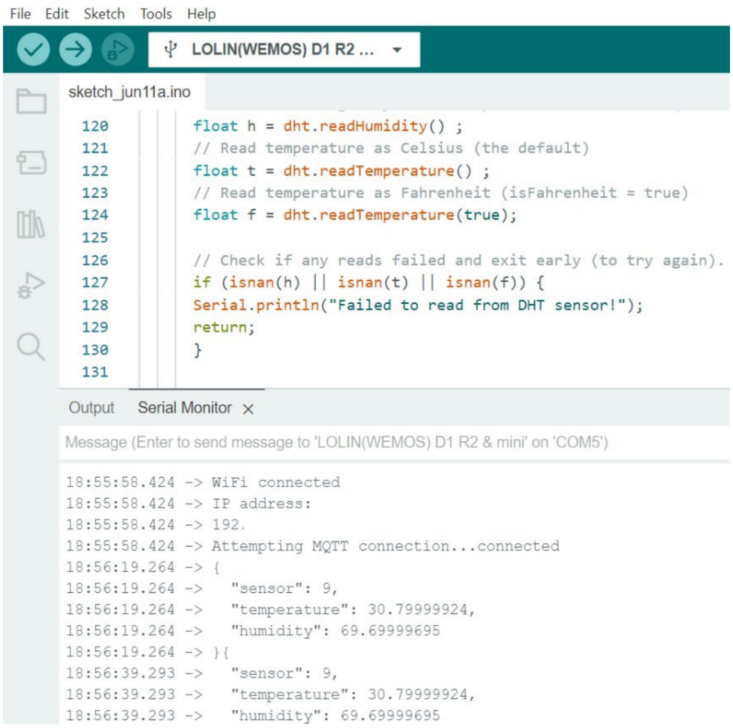


Fig. 7 *Experimental setup block diagram*

Figure 7 depicts a block representation of the experimental setup in this study, which includes connections between the WeMos D1 Mini development board and the DHT22 sensor. This setup was designed and installed inside the Digital Home Lab to collect humidity and temperature readings. The D4 pin of the WeMos D1 Mini was wired to the Data pin of the DHT22 sensor for transferring the readings. The positive terminal of the battery was connected to the 5V pin on the WeMos D1 Mini. This input voltage was regulated down to 3.3V, which ultimately powered the DHT22 sensor. The negative connection of the battery was used to ground the circuit's GND pins. The ground is a safety precaution to protect building occupants and equipment.

4. Results and Discussion

Sensor data delivered by the development board connected to the USB port can be seen on the serial monitor of the Arduino IDE software, as shown in Figure 8.



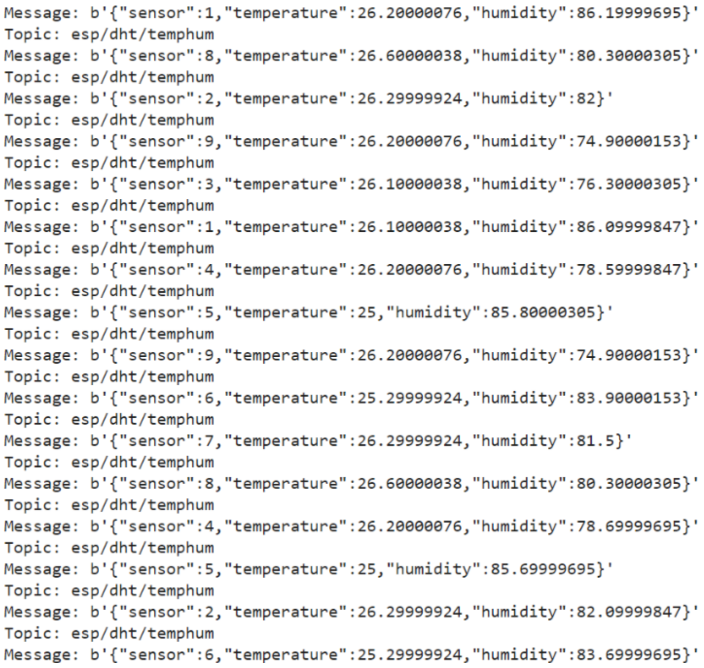
```
float h = dht.readHumidity() ;
// Read temperature as Celsius (the default)
float t = dht.readTemperature() ;
// Read temperature as Fahrenheit (isFahrenheit = true)
float f = dht.readTemperature(true);

// Check if any reads failed and exit early (to try again).
if (isnan(h) || isnan(t) || isnan(f)) {
Serial.println("Failed to read from DHT sensor!");
return;
}
```

```
18:55:58.424 -> WiFi connected
18:55:58.424 -> IP address:
18:55:58.424 -> 192.
18:55:58.424 -> Attempting MQTT connection...connected
18:56:19.264 -> {
18:56:19.264 ->   "sensor": 9,
18:56:19.264 ->   "temperature": 30.79999924,
18:56:19.264 ->   "humidity": 69.69999695
18:56:19.264 -> }{
18:56:39.293 ->   "sensor": 9,
18:56:39.293 ->   "temperature": 30.79999924,
18:56:39.293 ->   "humidity": 69.69999695
```

Fig. 8 *Arduino IDE serial monitor*

Figure 9 is a screen capture of the Mosquitto MQTT display monitor, which streams real-time temperature and humidity data using IoT. Table 4 shows data exported from an MQTT server and stored in a CSV file. Every sensor in this experiment has had its temperature and humidity calibrated. From this experiment, a total of 1000 data samples were collected.



```
Message: b'{"sensor":1,"temperature":26.20000076,"humidity":86.19999695}'
Topic: esp/dht/temphum
Message: b'{"sensor":8,"temperature":26.60000038,"humidity":80.30000305}'
Topic: esp/dht/temphum
Message: b'{"sensor":2,"temperature":26.29999924,"humidity":82}'
Topic: esp/dht/temphum
Message: b'{"sensor":9,"temperature":26.20000076,"humidity":74.90000153}'
Topic: esp/dht/temphum
Message: b'{"sensor":3,"temperature":26.10000038,"humidity":76.30000305}'
Topic: esp/dht/temphum
Message: b'{"sensor":1,"temperature":26.10000038,"humidity":86.09999847}'
Topic: esp/dht/temphum
Message: b'{"sensor":4,"temperature":26.20000076,"humidity":78.59999847}'
Topic: esp/dht/temphum
Message: b'{"sensor":5,"temperature":25,"humidity":85.80000305}'
Topic: esp/dht/temphum
Message: b'{"sensor":9,"temperature":26.20000076,"humidity":74.90000153}'
Topic: esp/dht/temphum
Message: b'{"sensor":6,"temperature":25.29999924,"humidity":83.90000153}'
Topic: esp/dht/temphum
Message: b'{"sensor":7,"temperature":26.29999924,"humidity":81.5}'
Topic: esp/dht/temphum
Message: b'{"sensor":8,"temperature":26.60000038,"humidity":80.30000305}'
Topic: esp/dht/temphum
Message: b'{"sensor":4,"temperature":26.20000076,"humidity":78.69999695}'
Topic: esp/dht/temphum
Message: b'{"sensor":5,"temperature":25,"humidity":85.69999695}'
Topic: esp/dht/temphum
Message: b'{"sensor":2,"temperature":26.29999924,"humidity":82.09999847}'
Topic: esp/dht/temphum
Message: b'{"sensor":6,"temperature":25.29999924,"humidity":83.69999695}'
```

Fig. 9 *Real-time data MQTT*

Table 4 *Data from MQTT stored in CSV file*

Date	Time	Sensor	Temperature (C°)	Relative Humidity (%)	Wind Speed (m/s)
5/25/2023	18:23:29	1	25.00	70.30	1.17
5/25/2023	18:23:29	8	25.50	70.27	1.17
5/25/2023	18:23:29	2	24.90	73.03	1.05
5/25/2023	18:23:29	3	25.20	72.90	1.17
5/25/2023	18:23:29	4	24.60	71.60	1.29
5/25/2023	18:23:29	9	25.47	73.60	1.40
5/25/2023	18:23:29	5	23.73	72.93	1.29
5/25/2023	18:23:29	6	23.20	74.17	1.52
5/25/2023	18:23:29	7	24.90	72.03	1.63
5/25/2023	18:23:28	8	25.40	70.37	1.52
5/25/2023	18:23:28	4	24.70	71.50	1.63
5/25/2023	18:23:28	5	23.63	72.53	1.87
5/25/2023	18:23:28	2	24.90	72.73	1.63
5/25/2023	18:23:28	6	23.20	74.07	1.75
5/25/2023	18:23:28	3	25.00	73.10	1.52
5/25/2023	18:23:28	7	25.00	72.13	1.63
5/25/2023	18:23:28	1	24.90	71.10	1.40
5/25/2023	18:23:28	9	25.40	70.27	1.52
5/25/2023	18:23:27	2	24.90	73.03	1.29

The data table (Table 4) in the document provides a detailed view of the sensor data collected during the experiment. The table includes the following columns:

- Date and Time: These columns indicate the exact moment the data was collected. This is important for tracking changes over time and identifying any patterns or trends in the data.
- Sensor: This column indicates the sensor number from which the data was collected. This allows for the comparison of data across different sensors.
- Temperature (C°): This column shows the temperature readings from each sensor. The temperature was measured in degrees Celsius.
- Relative Humidity (%): This column shows the relative humidity readings from each sensor. The humidity was measured as a percentage.
- Wind Speed (m/s): The value in this column, in meters per second (m/s), indicates the speed at which the air conditioner fan is circulating air.

The data in the table indicates that the experiment collected a variety of temperature and humidity readings from different sensors at different times. This data was analyzed to understand the performance and reliability of the sensors, as well as to gain insights into the environmental conditions during the experiment.

Additional data was gathered using a DHT22 sensor and UT363BT anemometer to investigate the relationship between the AC and the indoor climate's temperature, humidity, and wind speed. For more effective visualization, the readings recorded while the air conditioners were off are summarized in Table 5 and displayed in Figure 10.

Table 5 *Readings with AC turned off*

Time	Temperature (C°)	Relative Humidity (%)	Wind Speed (m/s)
15:22:20	30.17	65.2	1.17
15:22:40	30.17	65.5	1.17
15:23:00	30.17	65.5	1.05
15:23:20	30.17	65.6	1.17
15:23:40	30.17	65.4	1.29
15:24:00	30.17	65.4	1.40
15:24:20	30.27	65.4	1.29

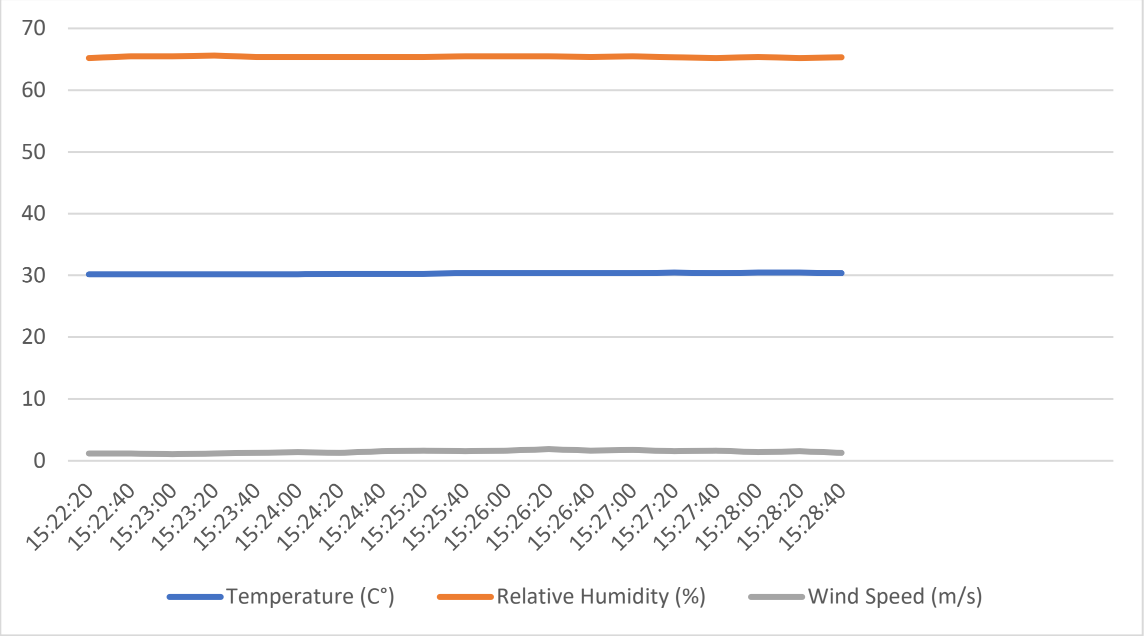


Fig. 10 *AC switched off*

The sensor readings in Table 6 were recorded while both ACs operated. The settings of the ACs were kept constant throughout the data collection for this experiment. For instance, the fan speed was highest, and the temperature was maintained at 24 °C. Figure 11 further depicts the readings from Table 6 in more detail.

Table 6 *Readings with AC turned on*

Time	Temperature (C°)	Relative Humidity (%)	Wind Speed (m/s)
15:57:00	26.17	62.3	3.17
15:57:20	26.17	62.2	4.17
15:57:40	26.27	61.7	3.05
15:58:00	26.17	61.7	4.17
15:58:20	26.07	61.6	4.29
15:58:40	26.17	61.6	4.40
15:59:00	26.17	61.3	4.29
15:59:20	26.27	61.0	4.52
15:59:40	26.27	60.9	4.63
16:00:00	26.27	61.0	4.52
16:00:20	26.27	60.8	4.63
16:00:40	26.27	60.7	4.87
16:01:00	26.37	60.6	4.63
16:01:20	26.37	60.7	4.75
16:01:40	26.37	60.3	4.52
16:02:00	26.47	60.3	4.63
16:02:20	26.37	59.9	5.40
16:02:40	26.47	59.8	5.52
16:03:00	26.37	59.6	5.29

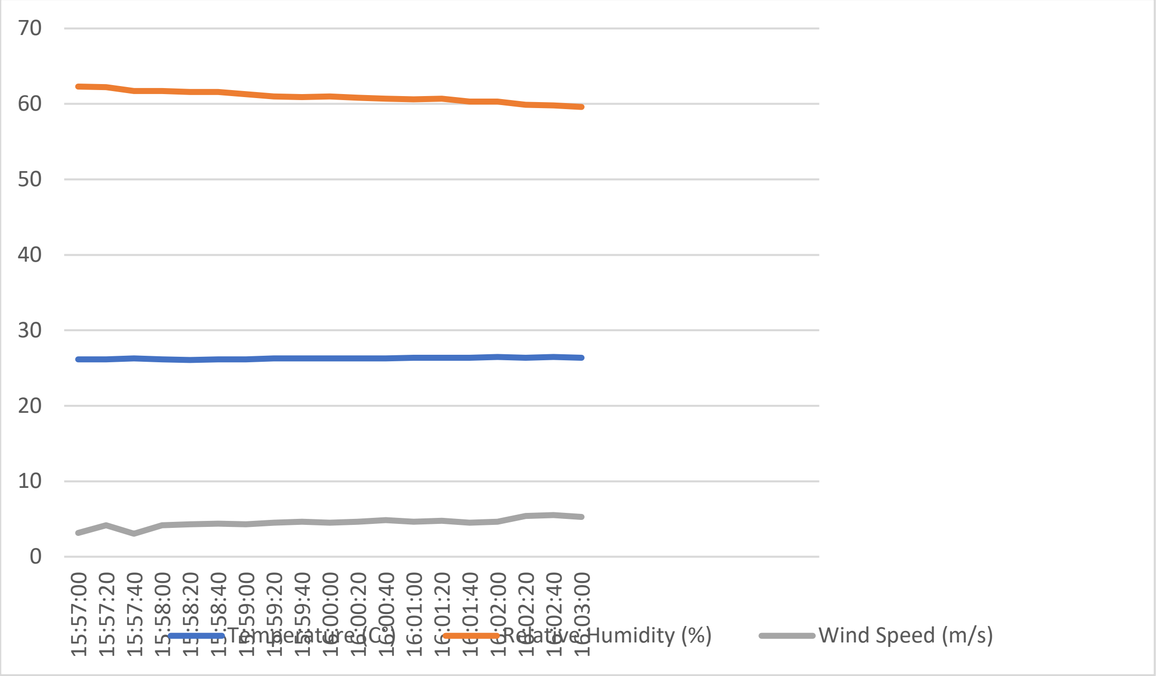


Fig. 11 *AC switched on*

From the above Figures 10 and 11, it is visible that when the air conditioners were turned on, the temperature and humidity measurements were lower than when they were turned off. It is also evident that the wind speed increased and significantly improved as the air conditioners were switched on. Given that both air conditioners were running, there was more airflow inside the digital home lab.

A total of 1000 temperature, humidity and air speed data were collected from this experiment, and no data loss was found. The results of this experiment demonstrate the potential of using IoT devices for real-time data collection and streaming, which can be applied in various fields such as environmental monitoring, industrial control systems, and home automation.

5. Conclusion

The proposed IoT-based embedded system incorporates real-time cloud communication using appropriate embedded technologies and programming interfaces. This system has many advantages with its minimal interconnections. Due to having fewer functionalities, these setups are cheaper to design and construct. This system runs on batteries as the microcontroller used in the experimental setup does not require a lot of processing power. The DHT22 sensor used in this study for data gathering has excellent reliability and long-term stability, making it an ideal choice for various applications, including indoor air quality monitoring systems. These sensors are compact, affordable, and power-efficient. A private MQTT broker offers more control over data privacy and security. Private MQTT brokers, which can be set up in the cloud, are often used for secure and dependable communication between IoT devices within an organization. Leveraging real-time remote monitoring through IoT can enhance workplace productivity. By diligently tracking environmental factors such as temperature and humidity, we can ensure thermal comfort for individuals, leading to increased performance. From this study, a total of 1000 temperature, humidity, and wind speed data samples were successfully collected. A thorough examination was conducted to rule out missing values, packet losses or time delays, but none were discovered. For future work, an IoT dashboard can be built to improve user interaction via graphs and widgets.

A future thermal comfort model can be created using machine learning using the collected temperature, humidity, and other environment and personal variables. Continuous data transmission could exert considerable pressure on the server's memory and storage capacity. As such, solutions can be developed as part of future work to address this data volume issue.

Acknowledgement

The authors would like to acknowledge the support of the TM R&D Research Grant (RDTC/221055), TM R&D, Multimedia University, Malaysia.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Chee-Pun Ooi, Wooi-Haw Tan;* ***data collection:*** *Farzana Sharmin Nila;* ***analysis and interpretation of results:*** *Wooi-Haw Tan, Yi-Fei Tan, Farzana Sharmin Nila;* ***draft manuscript preparation:*** *Farzana Sharmin Nila, Wooi-Haw Tan, Yi-Fei Tan. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] S. C. Turner et al., "American society of heating, refrigerating and air-conditioning engineers," *Int. J. Refrig., vol. 2*, no. 1, pp. 56–57, 1979, doi: 10.1016/0140-7007(79)90114-2.

[2] M. E. Emetere, "Chapter 3 - Typical environmental challenges," M. E. B. T.-N. M. in E. D. A. Emetere, Ed., Elsevier, 2022, pp. 41–51. doi: https://doi.org/10.1016/B978-0-12-818971-9.00004-1.

[3] Y. Nureni, "OVERVIEW OF EMBEDDED SYSTEM & ITS APPLICATION Computer Engineering Department Yaba College of Technology Computer Technology Department , Yaba College of Technology," no. June, 2022.

[4] M. García-Monge et al., "Is IoT monitoring key to improve building energy efficiency? Case study of a smart campus in Spain," Energy Build., vol. 285, p. 112882, 2023, doi: https://doi.org/10.1016/j.enbuild.2023.112882.

[5] B. A. Pramudita, R. C. Nugroho, P. Pangaribuan, and P. Megantoro, "Temperature Monitoring and Controlling System In Cassava Fermentation Room based On Internet of Things," vol. 030007, no. February, p. 30, 2023.

[6] A. Sharma, S. Sharma, and D. Gupta, "A Review of Sensors and Their Application in Internet of Things (IOT)," *Int. J. Comput. Appl., vol. 174*, no. 24, pp. 27–34, 2021, doi: 10.5120/ijca2021921148.

[7] A. Gupta, A. Mishra, M. Srivastava, and B. P. Dwivedi, "SMART INTERNET OF THINGS (IOT) BASED AIR-CONDITIONING SYSTEM," no. 05, pp. 520–526, 2023.

[8] F. Furizal, S. Sunardi, and A. Yudhana, "Temperature and Humidity Control System with Air Conditioner Based on Fuzzy Logic and Internet of Things," J. Robot. Control (JRC); Vol 4, No 3 (2023)DO - 10.18196/jrc.v4i3.18327 , May 2023, [Online]. Available: https://journal.umy.ac.id/index.php/jrc/article/view/18327

[9] Desai, P. Moga, and R. Patwardhan, "Design and Implementation of IOT Based Smart Library," … Res. J. …, no. May 2023, 2022, [Online]. Available: https://search.proquest.com/openview/e9fb078bccfcc92729e5978094e66f6b/1?pq-origsite=gscholar&cbl=5314840

[10] M. F. H. Hambali, R. K. Patchmuthu, and A. T. Wan, "IoT Based Smart Poultry Farm in Brunei," 2020 8th Int. Conf. Inf. Commun. Technol. ICoICT 2020, 2020, doi: 10.1109/ICoICT49345.2020.9166331.

[11] W. A. Jabbar, S. Member, T. K. Kian, R. M. Ramli, V. Shepelev, and S. Alharbi, "Design and Fabrication of Smart Home with Internet of Things Enabled Automation System," *IEEE Access*, vol. XX, pp. 1–9, 2017.

[12] E. Laftchiev and D. Nikovski, "An IoT system to estimate personal thermal comfort," 2*016 IEEE 3rd World Forum Internet Things, WF-IoT 2016*, pp. 672–677, 2017, doi: 10.1109/WF-IoT.2016.7845401.

[13] M. Lavanya, P. Muthukannan, Y. S. S. Bhargav, and V. Suresh, "Iot Based Automated Temperature and Humidity Monitoring and Control," *J. Chem. Pharm. Sci., no. 5,* pp. 86–88, 2016.

[14] A. Rajurkar, O. Shinde, V. Shinde, and B. Waghmode, "Smart Home Control and Monitor System Using Power o f IoT ' s," I*nt. J. Adv. Res. Comput. Commun. Eng., vol. 5*, no. 5, pp. 398–400, 2016, doi: 10.17148/IJARCCE.2016.5593.

[15] M. N. Bhuiyan, M. M. Rahman, M. M. Billah, and D. Saha, "Internet of Things (IoT): A Review of Its Enabling Technologies in Healthcare Applications, Standards Protocols, Security, and Market Opportunities," *IEEE Internet Things J., vol. 8,* no. 13, pp. 10474–10498, 2021, doi: 10.1109/JIOT.2021.3062630.

[16] R. Mamoun, M. Nasor, and S. H. Abulikailik, "Design and Development of Mobile Healthcare Application Prototype Using Flutter," *Proc. 2020 Int. Conf. Comput. Control. Electr. Electron. Eng. ICCCEEE 2020,* pp. 1–6, 2021, doi: 10.1109/ICCCEEE49695.2021.9429595.

[17] M. Babiuch, P. Foltynek, and P. Smutny, "Using the ESP32 microcontroller for data processing," P*roc. 2019 20th Int. Carpathian Control Conf. ICCC 2019*, no. March, 2019, doi: 10.1109/CarpathianCC.2019.8765944.

[18] M. O. Onibonoje, P. N. Bokoro, N. I. Nwulu, and S. L. Gbadamosi, "An IoT-Based Approach to Real-Time Conditioning and Control in a Server Room," *2019 Int. Conf. Artif. Intell. Data Process. Symp. IDAP 2019*, pp. 1–6, 2019, doi: 10.1109/IDAP.2019.8875880.

[19] A. F. Subahi and K. E. Bouazza, “An Intelligent IoT-Based System Design for Controlling and Monitoring Greenhouse Temperature,” I*EEE Access, vol. 8,* pp. 125488–125500, 2020, doi: 10.1109/ACCESS.2020.3007955.

[20] R. Rakhmawati, Irianto, F. D. Murdianto, A. Luthfi, and A. Y. Rahman, “Thermal optimization on incubator using fuzzy inference system based IoT,” *Proceeding - 2019 Int. Conf. Artif. Intell. Inf. Technol. ICAIIT 2019,* no. October, pp. 464–468, 2019, doi: 10.1109/ICAIIT.2019.8834530.

[21] S. A. Al-Qaseemi, H. A. Almulhim, M. F. Almulhim, and S. R. Chaudhry, “IoT architecture challenges and issues: Lack of standardization,” *FTC 2016 - Proc. Futur. Technol. Conf., no. December,* pp. 731–738, 2017, doi: 10.1109/FTC.2016.7821686.

[22] P. P. Ray, “An Internet of Things based Approach to Thermal Comfort Measurement and Monitoring,” 2016.

[23] M. H. B. Reddy and S. Talasila, “Internet of Things Protocols for Heterogeneous Devices and Cloud Services : Layered IoT Architectures,” no. November, 2020.

[24] S. Khare and M. Totaro, “Internet of Things: An Overview,” *Adv. Intell. Syst. Comput., vol. 1118*, no. March, pp. 71–79, 2020, doi: 10.1007/978-981-15-2475-2_8.

[25] Pallavi Sethi and Smruti R., “Internet Of Things: Architecture, Issues and Applications,” *Int. J. Eng. Res. Appl., vol. 07,* no. 06, pp. 85–88, 2017, doi: 10.9790/9622-0706048588.

[26] Powera, “Handson Technology,” Handson Technol., pp. 1–22, 2017, [Online]. Available: http://www.handsontec.com/pdf_learn/esp8266-V10.pdf

[27] F. Sarry and M. Lumbreras, “Gas discrimination in an air-conditioned system,” IEEE Trans. Instrum. Meas., vol. 49, no. 4, pp. 809–812, 2000, doi: 10.1109/19.863929.

[28] T. Liu, “Datasheet Digital-output relative humidity & temperature sensor/module(DHT22),” Aosong Electron. Co ., Ltd, vol. 22, pp. 1–10, 2013.

[29] P. R. Egli, “MQTT-Message Queueing Telemetry Transport Introduction to MQTT, a protocol for M2M and IoT applications,” no. September, 2017, doi: 10.13140/RG.2.2.13210.54721.

[30] “Mosquittopub Man Page.” *https://mosquitto.org/man/mosquitto_pub-1.html* (accessed May 28, 2023).

[31] L. B. Segerbäck, “General characteristics,” Orchid. Niger., pp. 1–7, 2020, doi: 10.1201/9781003079002-1.