Jurnal Teknik Informatika (JUTIF)
Vol. 3, No. 1, Februari 2022, hlm. 147-153

DOI: https://doi.org/10.20884/1.jutif.2022.3.1.172
p-ISSN: 2723-3863
e-ISSN: 2723-3871

FREERTOS BASED AIR QUALITY MONITORING SYSTEM USING SECURE INTERNET OF THINGS

Belen Septian[*1], Md. Misbahuddin[2], Fardhan Arkan[3]

[1]Teknik Elektro, Fakultas Ilmu Komputer & Rekayasa, Universitas Multi Data Palembang, Indonesia
[2]Electronics and Communication Engineering, Osmania University, India
[3]Teknik Elektro, Fakultas Teknik, Universitas Bangka Belitung, Indonesia
Email: [1]belen.septian@mdp.ac.id, [2]misbah426@gmail.com, [3]arkanfardhan@yahoo.com

(Naskah masuk: 17 Februari 2022, Revisi: 18 Februari 2022, diterbitkan: 25 Februari 2022)

Abstract

As the number of vehicles is increasing, the air quality in the environment becomes more polluted. This problem happens due to concentration of carbon dioxide (CO_2) is high in the air and it can lead to various health issues. The air quality meter is an important device to measure the concentration of CO_2 in the air. In this study, an air quality monitoring system is built using Arduino Nano along with IoT technology to measure parameters such as CO_2, temperature, humidity, and heat index. FreeRTOS is used to run multiple tasks consisting of display and transmit concurrently. MQ135 and DHT22 are used to sense CO_2, temperature, humidity, and heat index respectively. Data will be sent periodically to a web server using ESP8266 Wi-Fi module through secure HTTPS POST protocol. On the back-end side, a web server is employed to receive sensor parameters as well as to build a website application securely by which users can monitor it remotely. The system was tested to monitor CO_2 in 4 different locations of Pangkal Pinang city, Indonesia. The results showed that the averages of CO_2 concentration in Housing Indo Graha, Morning Market, Dea Lova Park, and Pasir Padi Beach are 411.37 ppm, 485.97 ppm, 416.45 ppm, and 444.43 ppm respectively. Based upon the results, public place i.e. Pasir Padi Beach has higher CO_2 concentration meanwhile public place with organic waste i.e. Morning Market has the highest CO_2 concentration.

Keywords: *Air Quality Monitoring System, FreeRTOS, HTTPS, IoT.*

1. INTRODUCTION

Environmental quality is affected by many factors, one of which is the air quality index. One of the important substances which influences air quality indicators is CO_2. CO_2 can be associated as an odorless, colorless, and non-combustible gas resulting from the oxidation of the carbon [1]. High-level CO_2 imposes very dangerous effects on either humans or the environment. Based on [2], Breathing with an excessive amount of CO_2 results in a high level of CO_2 in the blood related to the reduction of blood pH resulting in a health issue known as acidosis. The reduction of tissue and blood in pH produces effects on the central nervous systems, cardiovascular, and respiratory. Moreover exposure to a level between 2000 to 5000 ppm, humans may encounter various health issues including headaches, insomnia, nausea, and hypercapnia [1].

Currently, the average of CO_2 in the Earth's atmosphere is about 400 ppm and this index is still increasing [3]. This substance is the crucial one among the greenhouse gases as it absorbs infrared radiation on the Earth. This event leads to a significant change in temperature. The normal CO_2 level in the outdoor ranges 250-350 PPM and 350-1000 ppm as for indoor, some levels beyond which may harmful for the humans and environment [11].

In this study, it is proposed the air quality monitoring system that measures CO_2, humidity, temperature, and heat index. Humidity and temperature play important role in many biological processes in nature. Some changes in humidity and temperature affect various biological reactions [1]. IoT usage is also proposed in this study to enable the users to monitor air quality parameters anytime and anywhere in the world through the secure HTTPS protocol. This is to ensure that the data are safe from unauthorized parties. Additionally, FreeRTOS is employed to boost the performance of the system.

1.1. Related Studies

A study on the CO_2 monitoring system was done in [5]. CO_2 and CH_4 were measured using MOS sensors: MQ2, MQ4, TGS2611, and MQ135. Arduino Nano was used as a microprocessor and data were sent through a wireless network. The results showed the measurements of CO_2 and CH_4 in different places of Juliaca city. Wherein, places with a large number of vehicles have a high concentration of CO_2 and CH_4.

Another study in this field was done by [1], 5 parameters were measured such as CO_2, smoke, NO_2, SO_2, and LPG. These parameters were measured using the MQ135 sensor and data were shown on an internet-based platform. Data were stored in ThinkSpeak API via ESP8266 wireless module. The system is complex as it used only a single MQ135 sensor to measure different parameters. The main advantages of this system are a cost-effective and wide range of measurements.

Air quality monitoring system was developed using the Internet of Things to measure NO_2, SO_2, CO, and CO_2 [4]. MQ135 sensor was used along with NodeMCU V1.0 as a wireless-enabled microcontroller. The tolerance level was set to 1000 ppm thus if the value exceeds tolerance level, the system would send a message to users through the IFTTT application. To enhance IoT security, AWS IoT device SDK was used to authenticate messages over the internet network.

A different way to measure CO_2 by using the MH-Z19 NDIR infrared gas module was done by [3]. This system was developed using an Arduino Uno microcontroller interfaced with the LCD1602. The microcontroller is connected to MH-219 and AM2302 sensors to measure CO_2, temperature, and humidity respectively. The results of this study are plotted in the form of a line chart.

In this study, an air quality monitoring system is built using an Arduino Nano microcontroller. FreeRTOS library is employed to furnish multitasking capability to the system [6]. The system is divided into two tasks such as display and transmit to speed up the performance of the microcontroller, 4 parameters that will be measured including CO_2, temperature, humidity, and heat index using MQ135 and DHT22 sensors respectively [5,6,7]. These parameters will be sent to a cloud hosting server through a secure HTTPS protocol to ensure the security of the data [8].

1.2. MQ135 Sensor

This sensor consists of 4 terminals such as Vcc, Gnd, A0, D0. It also supports various low-end microcontrollers including the ESP series, STM series, and Arduino series. MQ135 sensor can sense various kinds of toxic gases consisting of Ammonia, Smoke, CO_2 Benzene, alcohol, and NOx [4]. However, due to the excessive amount of CO_2 compared to other gases, this sensor is used to measure CO_2 concentration. Before measurements, the sensor needs to be calibrated by using the calibration code in Arduino IDE. It has to be calibrated for 24 hours so that the precise value of resistance is obtained. However, the resistance value may change deliberately as the sensor is highly sensitive to the working environment. The specifications of this sensor are given in Table 1.

Table 1. Specifications of MQ135 [4]

Parameter	Value
Operating Voltage	5.0 +/- 0.1V AC (or) DC
Detection Range	10 – few thousands of PPM
Sensitive Material	SnO_2

1.3. DHT22 Sensor

DHT22 is a low-cost sensor used for measuring temperature and humidity. It makes use of a capacitor-based humidity and temperature sensor to measure the surrounding air. This sensor produces a digital signal that is to be transferred to a microcontroller digital pin. This sensor has better accuracy and performance than that of DHT11 in addition, it supports a broader range of measurements [9].

It consists of 4 pins including Vcc, Data, NC, Gnd from left to right respectively. It requires a pull-up resistor between Vcc and Data to hold the input in high logic. However, some manufacturers produced it with an in-built pull-up resistor thus there are only 3 pins available. Vcc pin can be supplied with DC voltage ranges from 3.3V to 6V.

1.4. HTTPS Protocol

HTTPS is an IP that allows websites and users to maintain security and integrity in their communication [8]. It uses SSL protocol and if the server needs to obtain HTTPS' trust, it is compulsory to install the credentials acknowledged by the SSL certificate. SSL is the combination of 4 protocols that provides security to upper-layer protocols such as FTP, HTTP, and any application layer protocols [10].

1.5. Internet of Things

The term Internet of Things was founded in the early 2000s to illustrate the range of Internet-connected objects as well as their uses. In the year 2005, the ITU officially acknowledged the term. The ITU describes the IoT as a system consisting of smart objects which connect to the internet and communicate with each other without the need for or minimal human intervention. These objects are connected throughout a gateway to an IoT platform, that consists of services and software tools furthermore collect data from controllers, sensors, and other devices [12].

For complex applications, IoT devices are equipped with actuators, embedded sensors, processors, and transceivers. IoT is not a sole technology instead, it is a collection of various technologies working together in a company. Actuators and sensors are devices that aid in interacting with the environment. The data accumulated by the sensors have to be processed and stored in the database to gain useful insight from it [13].

1.6. FreeRTOS Library

FreeRTOS is a favored open-source RTOS kernel by which embedded applications can be created to meet the requirements. This kernel receives over 75,000 downloads per year [6]. By using this, it is possible to sort applications as a collection of unconstrained tasks of execution. On one core processor, only one task is allowed to be processed at any time. The scheduler of the kernel is accountable for choosing which task should be executed by scrutinizing the priority assigned to each task. It is outlined to be compact, easy to port, maintainable, and simple.

2. METHODOLOGY

MQ135 is not specifically made to measure CO_2 concentration in the air, the sensor characteristics must be taken from datasheet [15] and have to be computed using a mathematical equation. According to [5], ppm can be computed using the power regression equation. Characteristics value is preprocessed using statistic software and the results are calculated using a regression model. The same has also been proposed in [14], this formula is also known as Gironi’s Formulation.

Another important algorithm in this proposed study is priority-based scheduling. It provides determinism by permitting the users to allocate the priority of each thread. Thus, the scheduler manages the threads using the priority to perceive which thread will be executed next [6]. In this study tasks are classified into 2 tasks: display and transmit, it is assigned based upon which priority is the highest.

2.1. Contoh Sub-Bab Pertama

The system in this study consists of two processing parts i.e. hardware processing and software processing. Hardware processing is classified into 3 main roles: input, process, output, wherein each role is carried out by sensors, Arduino Nano, and LCD respectively. Process role will be performed by Arduino Nano microcontroller to receive inputs from the sensors and display as well as send data to a web server using ESP8266 wireless module. Tasks are managed using an open-source FreeRTOS library [11] by which each task i.e. display and transmit will be divided based upon their priorities. All the components are centered towards the Arduino Nano microcontroller as shown in Figure 1. It is responsible to manage all of the sensor, display, and network connections, to put it simply it functions as the brain for this system.

Software processing continues processes that have been done by a microcontroller. Data are to be sent to a cloud hosting server through HTTPS POST protocol in the form of plain texts. At this point, the data will be processed using backend programming language and to be saved in the database. As shown in Figure 2, there is an interaction between the cloud hosting server and database as the data of sensors are recorded in the database after going through some processes in the cloud hosting server. These data can be retrieved from the server via a secure website application whenever users need it for observation.

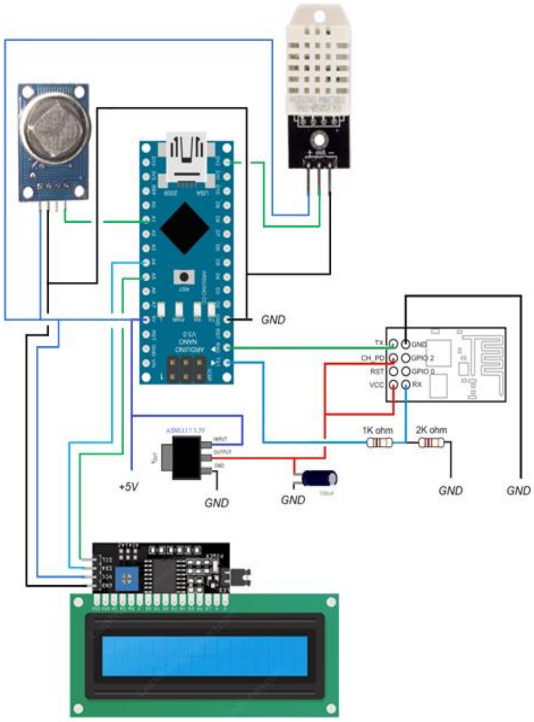

Figure 1. Schematic of System

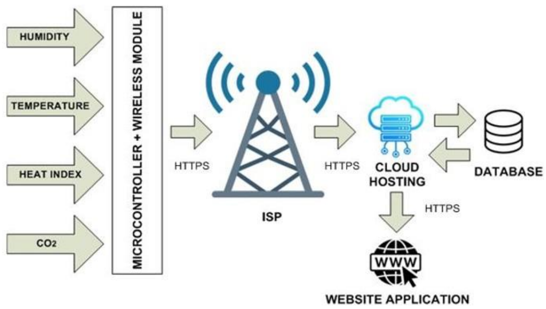

Figure 2. Architecture of System

2.2. Power Regression Model

MQ135 is a multipurpose sensor to measure numerous gases in the air. It is highly affected by SnO_2 [4] which makes the measurements become less accurate for other gases i.e. CO_2. In this study, the power regression equation is used to get scaling factor (a) and exponential factor (b) as shown in Equation 1 [14].

$$y = a.x^b \tag{1}$$

The sensor needs to be calibrated in the clean air prior to use. Using a and b values obtained from linear regression it can be calculated using Equation 2 [14].

$$Rs = Ro.sqrt(a/ppm,b) \tag{2}$$

Ro_limit value can be obtained using Equation 3 [14]. It is important to limit the value of CO_2 concentration based upon the recent trend. According to the NOAA-Earth System research laboratory, the average value of CO_2 in the air is 418.08 ppm as of July 3rd, 2021 [7]. Another problem to be concerned about is the sensor needs to be in preheat mode for over 24 hours [15] before the first usage to get an accurate measurements.

$$Rs/Ro\_limit = (ppm/a)^{1/b} \quad (3)$$

2.3. Priority Based Scheduling

By default FreeRTOS uses a preemptive algorithm, it can be also employed using preemptive priority-based scheduling to determine the priority of the tasks in which the developer can adjust the value of the priority. In FreeRTOS priority number starts from "0" which indicates that the task has the lowest priority amongst all [11]. Table 2. shows the priority of the tasks in this study consisting of display and transmit tasks.

Table 2. Priority of Task

Priority	Task	Remarks
0	void TaskDisplayParameters	To display sensor parameters on the LCD interface
1	void TaskTransmitParameters	To send data to a web server

Display task priority is set to "0" as it is less important than that of the transmit task. This task runs after transmit task finishes and displays all the sensors data to a 16x02 LCD. Transmit task priority is set to "1" as it takes more time to send the sensor parameters to a web server due to undesired issues such as network connection, web server issue, and latency. This is the main task as the system is expected to send the sensors data safely and periodically to the cloud hosting server.

For creating a task, the task is defined in the setup function and begins the task scheduler [6]. The complete functions of these two tasks are illustrated in the following lines.

a. xTaskCreate(TaskDisplayParameters, "Task 1", 128, NULL, 0, NULL); This task has priority "0" which is the lowest priority and responsible to display the output of the sensors on the LCD interface. Stack space must be assigned for each task, 128 indicates the number of words the stack can hold.

b. xTaskCreate(TaskTransmitParameters, "Task 2", 128, NULL, 1, NULL); This task has priority "1" which is the highest priority and it occupies the CPU immediately and begins running as soon as the system is turned on. This task is responsible to send sensor parameters to a web server whenever the system is being connected to a Wi-Fi network.

3. RESULTS AND DISCUSSION

The system is designed and built using free and open-source software. In addition, hardware parts i.e. sensors and microcontrollers were obtained through local and online markets. There are a few open-source software and libraries used in this study such as Arduino, FreeRTOS library, and Open Source database that is responsible for storing the data. Data collection was done in 4 different locations of Pangkal Pinang city, Indonesia. Data were taken for 15 minutes in each area and sent to the database periodically throughout the time. In this study, there exist 4 parameters that can be monitored including CO_2, temperature, humidity, and heat index. These parameters can be observed either via website application or LCD.

3.1. Management of Tasks

To begin with, a set of FreeRTOS libraries must be included in Arduino IDE to give the functionality of RTOS to Arduino Nano Microcontroller. It needs to be included before calling the main functions as shown in Figure 3.

```
#include <Arduino_FreeRTOS.h>
#include "SoftwareSerial.h"
#include "LCD.h"
#include "LiquidCrystal_I2C.h"
#include "DHT.h"

#define I2C_ADDR 0x27
#define Rs_pin 0
#define Rw_pin 1
#define En_pin 2
#define BACKLIGHT_PIN 3
#define D4_pin 4
#define D5_pin 5
#define D6_pin 6
#define D7_pin 7
#define DHTPIN 12
#define DHTTYPE DHT22

String ssid="freewifi";
String password="12345678";
```

Figure 3. Insertion of FreeRTOS Library

It is defined as "Arduino_FreeRTOS.h", this declaration needs to be included so that Arduino IDE recognizes FreeRTOS functions. In this study, there are 2 tasks managed by a kernel. These tasks are employed to display and send the data to a web server. Tasks are called in the form of functions in Arduino IDE as shown in Figure 4. As shown in Figure 4, "task1" contains ESP8266 functions to send data to a web server wherein "task2" contains functions to display sensor parameters on LCD.

Tasks are divided based upon their priorities as shown in Table 2. wherein "0" indicates the lowest priority and vice versa. As "TaskTransmitParameters" has "1" priority, it is called at the beginning of a cycle and subsequently "vTaskDelay" is called before the beginning of another task. According to FreeRTOS documentation [11], the kernel allocates idle task when "vTaskDelay" is called as shown in Figure 5. At this point, there is no task running in the system until the delay is cleared. As shown in Figure 4, "vTaskDelay" is called after the first task is finished followed by the second task and another delay. Delay is given 200ms and 300ms for transmit and display tasks respectively. This delay is utilized as a break for each task to maintain the stability of the system.

```
void TaskTransmitParameters(void *pvParameters)  {
  while(1)
  {
    task1();
    vTaskDelay( 200 / portTICK_PERIOD_MS );
  }
}
void TaskDisplayParameters(void *pvParameters)
{
  while(1)
  {
    task(2);
    vTaskDelay( 300 / portTICK_PERIOD_MS );
  }
}
```

Figure 4. Task Functions in Arduino IDE

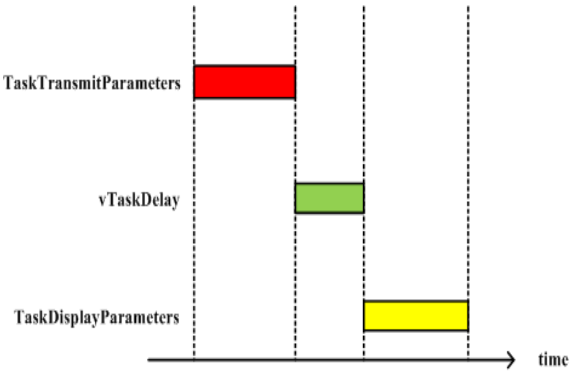

Figure 5. Allocation of Tasks by Kernel

3.2. Interface of System

The output of the sensors can be observed either via a website application or LCD. The requests are accessible using HTTPS POST protocol and it is to be redirected through a secure HTTPS URL. Monitoring via website can be done through the portal and users must be logged in before observation as shown in Figure 6.

Users have to input username and password recorded in the database, upon successful login they will be redirected to the main interface as shown in Figure 7. It shows the main portal interface which shows the daily chart of CO_2 concentration.

As shown in Figure 7, the line chart makes it easier for users to understand the characteristics of CO_2 measurements. Moreover, users can monitor real-time data as well as shown in Figure 8.

It displays all the sensor outputs such as CO_2, temperature, humidity, and heat index along with the date it was recorded in the storage. The observation can also be done through the LCD of the hardware as shown in Figure 9.

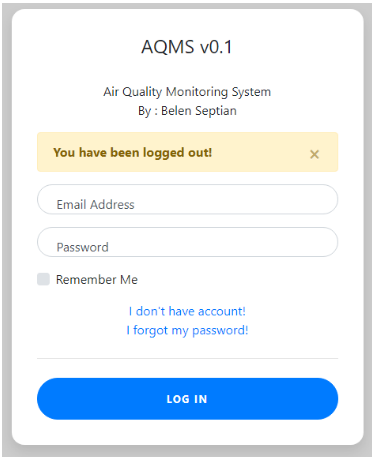

Figure 6. Login Interface of Portal

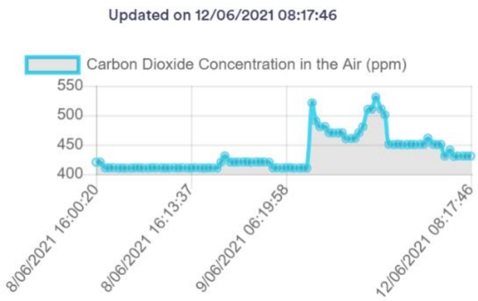

Figure 7. Main Interface of Portal

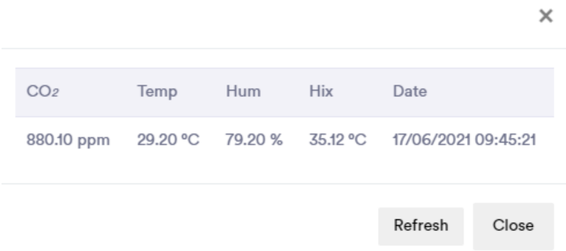

Figure 8. Real-Time Monitoring on Portal

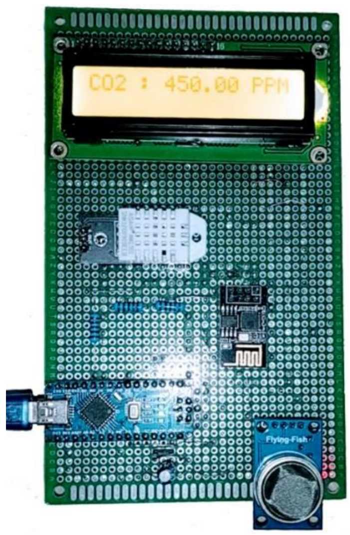

Figure 9. Monitoring Through LCD

Figure 9 shows the integrated hardware of this study, it consists of Arduino Nano, MQ135, DHT22, ESP8266, LCD, and passive components i.e. capacitor and resistors. LCD displays all the parameters alternately one after another, Figure 9 displays CO_2 concentration measured by the MQ135 sensor.

3.3. Analysis of Results

Data collection was taken place in Pangkal Pinang city, Indonesia wherein 4 different locations were chosen: Housing Indo Graha, Morning Market, Dea Lova Park, and Pasir Padi Beach. The results of the CO_2 average in these locations are shown in Table 3. and the places in which data were taken are pointed out on Google Earth as shown in Figure 10.

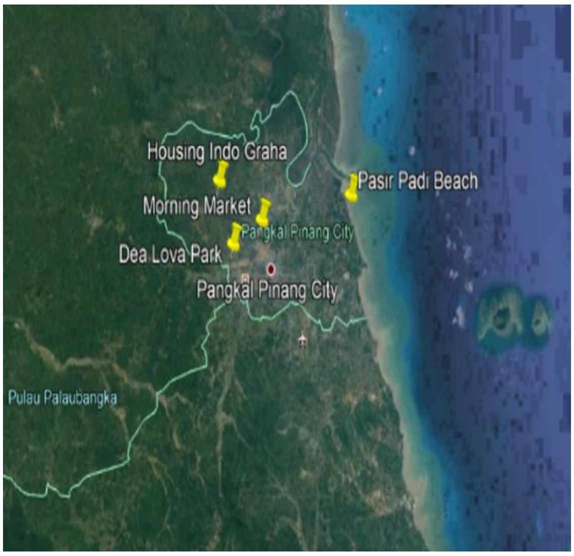

Figure 10. Location Mapped on Google Earth

Table 3. Averages of CO_2 Concentration in Each Location

Location	CO_2 Average
Housing Indo Graha	411.37 ppm
Morning Market	485.97 ppm
Dea Lova Park	416.45 ppm
Pasir Padi Beach	444.43 ppm

The results showed that the averages of CO_2 concentration in Housing Indo Graha, Morning Market, Dea Lova Park, and Pasir Padi Beach are 411.37 ppm, 485.97 ppm, 416.45 ppm, and 444.43 ppm respectively. Wherein places with a large number of people and vehicles have higher CO_2 levels such as Morning Market and Pasir Padi Beach. They are comparable with the research conducted by J. M. Montoya and J. Chilo [5], which stated that public places have higher CO_2 concentration meanwhile public place with organic waste i.e. Morning Market has the highest CO_2 concentration. These results are compiled based on the data stored periodically in the server. The number of data sent per minute for each location is shown in Table 4.

Table 4. Data Sent Per Minute for Each Location

Location	Data Sent Per Minute	Signal Strength
Housing Indo Graha	3	Strong
Morning Market	2	Medium
Dea Lova Park	2	Medium
Pasir Padi Beach	2	Medium

Table 4. shows that data sent per minute is proportional to the signal strength of the network. The results showed that Housing Indo Graha, Morning Market, Dea Lova Park, and Pasir Padi beach have received 3, 2, 2, and 2 data per minute respectively. Wherein, Housing Indo Graha has received the highest data per minute in respective of network signal.

4. CONCLUSION

CO_2 concentration should be monitored appropriately due to its harmfulness to humans and the environment. Thus, an air quality monitoring system is required by which people can monitor air quality parameters, especially CO_2. In this study, air quality parameters measured consist of CO_2, humidity, temperature, and heat index.

The system could be used successfully to monitor these parameters through a secure HTTP POST protocol. Besides, the system was able to manage two different tasks i.e. display and transmit successfully using FreeRTOS library. The observation was taken place in 4 different locations of Pangkal Pinang city, Indonesia. The results of measurements have been compared with the work done by [5] as shown in Table 5.

Table 5. Comparison of Current and Previous Study

Study	Location	CO_2 Average
Current Study	Housing Indo Graha	411.37 ppm
	Morning Market	485.97 ppm
	Dea Lova Park	416.45 ppm
	Pasir Padi Beach	444.43 ppm
Previous Study [5]	Andina University	400.00 ppm
	Centro commercial N° 2	500.00 ppm
	Cerro Huaynaroque	420.00 ppm
	Sandia Street	460.00 ppm

The results showed that the averages of CO_2 concentration in Housing Indo Graha, Morning Market, Dea Lova Park, and Pasir Padi Beach are 411.37 ppm, 485.97 ppm, 416.45 ppm, and 444.43 ppm respectively. Based upon the results, public place i.e. Pasir Padi Beach has higher CO_2 concentration meanwhile public place with organic waste i.e. Morning Market has the highest CO_2 concentration.

As IoT technology is always evolving, further improvements in such design can be made. Currently, 5G is a trend in the telecommunication field and such technology can be developed further for Internet of Things applications. Additionally, RTOS is necessarily required for embedded system applications as the complexity of the system increases.

REFERENCES

[1] B. Sahoo, et. al., “Low-Cost Air Sensing System,” in *3rd International Conference on Computing and Communication Technologies ICCCT 2019*, IEEE, 2019, pp. 258-267, doi: 10.1109/1742-6596/ICCCT2.2019.8824890.

[2] P. N. Bierwirth, “Carbon Dioxide Toxicity and Climate Change,” *Web Published: Research Gate*, 2020, pp. 1-22.

[3] P. D. Lapshina, S. P. Kurilova, and A. A. Belitsky, “Development of an Arduino-Based CO_2 Monitoring Device,” in *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, IEEE, 2019, pp. 595-597.

[4] R. K. Kodali, S. Pathuri, and S. C. Rajnarayanan, “Smart Indoor Air Pollution Monitoring Station,” in *2020 International Conference on Computer Communication and Informatics (ICCCI -2020)*, IEEE, 2020.

[5] J. M. Montoya and J. Chilo, “RealTime Wireless Monitoring System of CO_2 and CH_4 in Juliaca-Peru,” in *The 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, IEEE, 2019, pp. 464-467.

[6] L. D. O. Turci, “Real-Time Operating System FreeRTOS Application for Fire Alarm Project in Reduced Scale,” *International Journal of Computing and Digital Systems*, 2017, vol. 6, no. 4, pp. 198-204.

[7] Atmospheric CO_2, 2021. https://co2.earth/co2-for-print (accessed Jul. 10, 2021).

[8] C. Tseng, et. al., “An IoT-based Home Automation System Using Wi-Fi Wireless Sensor Networks,” in *2018 IEEE International Conference on Systems, Man, and Cybernetics*, IEEE, 2018, pp. 2430-2435.

[9] I. A. Abdulrazzak, H. Bierk, and L. A. Aday, “Humidity and Temperature Monitoring,” *International Journal of Engineering & Technology*, 2018, vol. 7, no. 4, pp. 5174-5177.

[10] A. Satapathy and J. Livingston, “A Comprehensive Survey on SSL/TLS and Their Vulnerabilities,” *International Journal of Computer Applications*, 2016, vol. 153, no. 5, pp. 31-38.

[11] FreeRTOS Documentation, 2017. https://www.freertos.org/Documentation/RTOS_book.html (accessed Aug. 20, 2021).

[12] Links, C, et. al. (2021) *Internet of Things for Dummies*. 2nd ed. New Jersey: John Wiley & Sons, Inc.

[13] P. Sethi and S. R. Sarangi, “Internet of Things: Architectures, Protocols, and Applications,” *Journal of Electrical and Computer Engineering*, 2017, vol. 153, no. 5, pp. 1-25.

[14] V. Kalra, et. al., “Influence of Temperature and Humidity on the Output Resistance Ratio of the MQ-135 Sensor,” *International Journal of Advanced Research in Computer Science and Software Engineering*, 2016, vol. 6, no. 4, pp. 423-429.

[15] MQ135 Datasheet, 2014. https://www.dreamgreenhouse.com/datasheets/MQ-135/index.php (accessed Jun. 13, 2021).