Design Of CO, CO₂, Temperature, Humidity, And Weather Monitoring System Based On Internet of Things (IoT) & Android

Christopher Calvin Adrianto¹, Yohanes Calvinus², Pono Budi Mardjoko³

¹Electrical Engineering Department, Faculty of Engineering Universitas Tarumanagara, Jakarta.
²Electrical Engineering Department, Faculty of Engineering Universitas Tarumanagara, Jakarta.
³Electrical Engineering Department, Faculty of Engineering Universitas Tarumanagara, Jakarta.

E-mail: christophercalvin1997@gmail.com

Abstract. Carbon monoxide (CO) gas can bind to hemoglobin in blood cells. Even if humans inhale CO gas at high levels, reaching 25 parts in a million (PPM) (PER.13 / MEN / X / 2011), the risk is death. Besides CO, there are other odorless and colorless gases, namely carbon dioxide (CO₂). Exposure to high levels of CO₂ concentrations reaching 1000 PPM or more can be bad (American ASHRAE 62-1989). Many individuals or groups in large buildings do not know the weather conditions outside because the building is too large and there are no windows to view the weather outside the building. This happens a lot in malls or large shopping centers. By utilizing the concept of internet of things (IoT) technology and Android applications, real-time remote monitoring can be realized.

1. Introduction
Carbon monoxide (CO) is one of the imperfect combustion gases from motor vehicles that can pollute the air. Carbon monoxide gas can be a pollutant in the room due to lack of air circulation, this can interfere with human health in the room. CO gas that pollutes the air can bind to hemoglobin in red blood cells, so that the binding of oxygen by the blood becomes disrupted. Even if humans inhale CO gas at high levels, reaching 25 parts per million (PPM) (PER.13 / MEN / X / 2011), the risk is death [1]. If it is in low levels, inhaling CO can cause headaches, dizziness, dizzy eyes, weakness and nausea. Besides CO, there are other odorless and colorless gases, carbon dioxide (CO₂). Exposure to high levels of CO₂ concentrations reaching 1000 PPM or more can be bad (American ASHRAE 62-1989). These conditions can cause unconsciousness, convulsions, or even death. Exposure to lower CO₂ concentrations can also cause a number of health problems. Many buildings still do not monitor the concentration of harmful gases to the air flow in a room inside. The impact, the air quality in the room becomes bad.

The weather condition is an important factor to determine whether an individual or group will leave the building or wait inside the building until the desired weather conditions occur. Many individuals or groups in large buildings do not know the weather conditions outside the building because the building
is too large and there are no windows to see the weather outside the building. This happens a lot in malls or large shopping centers.

The development of science and technology makes it possible to conduct remote monitoring in real time and automatically to the concentration of harmful gases, temperature, humidity, and weather conditions in a room. By utilizing the concept of internet technology of things (IoT) where an object that has the ability to be able to transmit data over a network without requiring human-to-human interaction or human to computer, this can be realized. In the IoT concept there are three important parts. The first is sensors and actuators, where sensors are used to get data on physical quantities and actuators are mechanical devices to drive or control a mechanism or system. The second is a microcontroller which is a chip that functions as an electronic circuit controller and can generally store programs in it. And the third is the internet network and applications used by users to obtain monitoring information on the system designed. The concept of IoT was chosen to monitor this system because the integration of internet-connected smart devices will help improve the safety of the concentration of harmful gases in the room through remote monitoring [2][3][4][5].

The tool is designed to function to monitor CO gas, CO$_2$, temperature, humidity, and weather conditions in an IoT and Android-based room. The hardware monitoring module requires input, i.e. analog or digital voltage data obtained from the sensor. The sensor consists of CO$_2$ (MQ135) and CO (MQ2) gas sensors as well as several sensors to add to the tool's features. The additional sensor is a temperature and humidity sensor (DHT22) used to detect the temperature and humidity of the area being monitored, and a water sensor (MD0127) is used to detect if there is rain. The hardware monitoring module can be placed in various rooms, but in this journal the test tool is placed on one side of the wall in the living room of a house. The living room was chosen because of the large scale social restrictions that were taking place at the time of writing this journal, and the living room was one of the places where many people gathered, so their comfort had to be considered. Laying the rain sensor on this tool is pulled from the device to the outside of the room so that if there is rain this sensor can detect rain water. The hardware monitoring module is packaged in a ventilated black box to get air samples in the room. Air samples around the room will be forwarded to the sensors contained in this module. After analog or digital data has been obtained these sensors then the data is sent to the microcontroller in the hardware monitoring module. When the microcontroller detects the input from sensors, the microcontroller will start processing. The processing produces an output in the form of gas content, temperature, humidity, and weather conditions. The microcontroller will send output to the LCD, Firebase, and buzzer. The LCD will display the output data that has been processed. The buzzer will sound when the gas level exceeds the maximum limit set. In addition to the LCD, there is an Android Application module that is connected to the internet network and Google Firebase database to display the results of monitoring this designed tool. The block diagram of the design of Monitoring System CO, CO$_2$, temperature, humidity and weather conditions based on IoT and Android can be seen in Figure 1.

![Figure 1. Block Diagram Design](image)

2. Realization Of Monitoring Hardware Module

Hardware monitoring module can be realized by using two NodeMCU 8266, one MQ2 sensor, one MQ135 sensor, one DHT22 sensor, one MD0127 sensor, two LCDs, and two I2C modules. The realization of the hardware monitoring program module is done by programming the microcontroller
with the Arduino Integrated Development Environment (IDE) programming language. If the processing module receives data from the sensor, the first process that will be performed is to read the analog signal from the analog pin connected to the sensor. The processing module in this tool is a microcontroller. The microcontroller used has an analog to digital converter (ADC) resolution of 10 bits so that it can detect as many as $2^{10}$ or 1024 analog levels (0 to 1023). Schematic design and printed circuit board (PCB) lines can be seen in Figures 2 and 3. Hardware monitoring module can be seen in Figure 4 [6][7][8][9][10][11].

3. Realization Of Firebase Cloud Module

Database realization on this tool uses Firebase. Firebase is an application development service created by Google. Database realization can be realized by creating a console at https://Firebase.google.com/. Next, you will get the authentication key and the hosting address. both of these will be used in Arduino programming to access the realtime database from Firebase. Figure 5 shows the hardware monitoring programming set up section so that it can communicate with Firebase.

```
1 #include <ESP8266WiFiClient.h>
2 #include <FirebaseArduino.h>
3 #include <ESP8266WiFi.h>
4 #define FIREBASE_HOST "skripsi-110lab.firebaseio.com"
5 #define FIREBASE_AGE "F2ogpE3HIL7e55In5Um3Zf4q6h4MyQF74p9M6az2"
6 #define WIFI_SSID "skripsi40" 
7 #define WIFI_PASSWORD "travis22"
```
Figure 5. Hardware Monitoring Module Set Up Programming for Firebase

The basic set up of programming on the hardware monitoring microcontroller to communicate with Firebase requires 3 basic libraries namely ESP8266HTTPClient.h, FirebaseArduino.h, and ESP8266WiFi.h. First, the hosting address must be defined as in the 5th line program. The hosting address can be accessed by defining the authentication key as in line 6 program. Hardware monitoring also requires internet access to be able to access Firebase. Programs on the 7th and 8th lines show access to hardware monitoring internet to WiFi with Nicholast SSID and their passwords.

There are five data sent by monitoring hardware to Firebase, namely CO, CO₂, temperature, humidity, and weather conditions. CO gas data will be sent to Firebase with the Firebase.setFloat ("coval", co) command. Where "coval" is a variable contained in Firebase which has a float data type, this variable will store "co" data, "co" is a hardware monitoring variable that stores PPM data for CO gas. CO₂ gas data will be sent to Firebase with the Firebase.setFloat ("co2val", co2) command. Where "co2val" is a variable contained in Firebase which has a float data type, this variable will store "co2" data, "co2" is a hardware monitoring variable that stores PPM data for CO₂ gas. Temperature data will be sent to Firebase with the Firebase.setFloat ("tval", t) command. Where "tval" is a variable contained in Firebase which has a float data type, this variable will store "t" data, "t" is a hardware monitoring variable that stores temperature data. Moisture data will be sent to Firebase with the Firebase.setFloat ("hval", h) command. Where "hval" is a variable contained in Firebase which has a float data type, this variable will store "h" data, "h" is a hardware monitoring variable that stores humidity data. Data on weather conditions if there is rain unread by monitoring hardware will be sent to Firebase with the Firebase.setString ("message", "rain") command. Data on weather conditions if there is no rain read by hardware monitoring will be sent to Firebase with the Firebase.setString ("message", "no rain") command. Where "message" is a variable contained in Firebase that has a string data type, this variable will store rain or no rain text. Figure 6 shows the realtime data base storage display of this designed tool.

Figure 5. Hardware Monitoring Module Set Up Programming for Firebase

Figure 6. Display Realtime Data Base Storage

4. Realization Of Android Application Module

The realization of android applications on this tool uses MIT App Inventor. The realization of the android application can be realized by doing visual block programming on https://appinventor.mit.edu/. There are two parts to making an application, namely the designer for the front end or for display, and the blocks for back end programming. In the designer part, several components are used. Figure 7 shows the components used in the design of the display android application.

Figure 6. Display Realtime Data Base Storage

4. Realization Of Android Application Module

The realization of android applications on this tool uses MIT App Inventor. The realization of the android application can be realized by doing visual block programming on https://appinventor.mit.edu/. There are two parts to making an application, namely the designer for the front end or for display, and the blocks for back end programming. In the designer part, several components are used. Figure 7 shows the components used in the design of the display android application.

Figure 7. Component Display Android Application
Display android applications that are designed using a screen only, can be seen in Figure 13, screen1. Label1 is used to write the name of the tool designer. Label2 is used to write down the student ID number (NIM) of the designer. Inside the horizontal arrangement1 there is label3 which is used to display the word "TEMPERATURE" on the screen, and tval is used to display temperature data obtained by the application from Firebase. Inside the horizontal arrangement2 there are labels4 which are used to display the word "HUMIDITY" on the screen, and hval is used to display moisture data obtained by applications from Firebase. Inside the horizontal arrangement 3 there is label 5 which is used to display the word "CARBON MONOXIDE" on the screen, and coval is used to display PPM CO data obtained by applications from Firebase. Inside the horizontal arrangement4 there is a label7 that is used to display the word "WEATHER CONDITION" on the screen, and the message is used to display rain or no weather data obtained by the application from Firebase. Inside the horizontal arrangement5 there is label9 which is used to display the word "CARBON DIOXIDE" on the screen, and co2val is used to display CO\textsubscript{2} data obtained by applications from Firebase. The FirebaseDB1 component is also used to access data that has been sent hardware monitoring to the Firebase real time database. This android application can access Firebase by using the token provided by the Firebase console, and the hosting address used. In the blocks the visual blocks are programmed so that the components in the fornt end are working properly. Figure 8 shows the visual block programming in the design of the Android application's back end tool.

![Figure 8. Designing Android Application Back End](image)

Basically the program in Figure 13 retrieves data on Firebase whenever there is a change in value. In the if then condition the first is to retrieve data on the Firebase tval variable. Then the tval label on the android application that has been provided will display the temperature value obtained from Firebase and add the °C unit. In the second if then condition, data retrieval is performed on the Firebase hval variable. Then the hval label on the android application that has been provided will display the moisture value obtained from Firebase and added%. In the third if then condition, data retrieval is performed on the Firebase coval variable. Then the coval label on the android application that has been provided will display the PPM CO value obtained from Firebase and the PPM unit is added. In the fourth if then condition, data retrieval is performed on the Firebase message variable. Then the message label on the android application that has been provided will display the value obtained from Firebase. In the fifth if then condition, data is taken from the Firebase co2val variable. Then the co2val label on the android application that has been provided will display CO\textsubscript{2} PPM values obtained from Firebase and add PPM units. The android application for this tool as a whole can be seen in Figure 9.
5. Results And Discussion

Hardware monitoring module testing is done to test the success of the module to monitor and display monitoring results on the LCD. Testing the hardware monitoring module is done by placing this module on one side of the room of a living room in the house. Testing will be done 4 times on the same day. This test is conducted in the afternoon, evening, night, and simulated rain conditions. This rain simulation is carried out due to weather conditions that don't rain. Rain conditions simulation is done by wetting the module that is tasked to detect the presence of rain. The data taken are PPM CO, PPM CO₂, temperature, humidity, and weather conditions. The results of testing the hardware monitoring module can be seen in Table 1 and Figure 10.

**Table 1. Test Results of Hardware Monitoring Module**

| Trial On       | CO (PPM) | CO₂ (PPM) | Temp. (°C) | Humidity (%) | Weather Con. |
|----------------|----------|-----------|------------|--------------|--------------|
| Daytime        | 0        | 662       | 29,6       | 55,6          | Kering       |
| Afternoon      | 0        | 667       | 29,6       | 54,8          | Kering       |
| Evening        | 1        | 588       | 29,7       | 56,1          | Kering       |
| Raining Sim.   | 0        | 607       | 30,1       | 49,4          | Hujan        |

Testing the Android Application module is done on a different day than the previous test, this test is carried out to test the success of the Android Application module to obtain data from hardware monitoring through Firebase and display the monitoring results in the appropriate column on the application screen. Android Application module testing is done by putting hardware monitoring in a room, hardware monitoring is placed on one side of a wall in the living room of a house. Testing will be done 4 times. This test is conducted in the afternoon, evening, night, and simulated rain conditions. This rain simulation is done by wetting the module which has the duty to detect the presence of rain. The data taken are PPM CO, PPM CO₂, temperature, humidity, and weather conditions. The results of testing the Android Application module can be seen in Table 2 and Figure 11.
Overall tool testing is carried out by combining all modules that are designed. Testing is done with a case of internet network variations. Overall tool testing table with variations of the internet network can be seen in Table 3.

| Trial On  | CO (PPM) | CO₂ (PPM) | Temp. (°C) | Humidity (%) | Weather Con. |
|----------|----------|-----------|------------|--------------|--------------|
| Daytime  | 1        | 573       | 31,9       | 56,9         | Tidak        |
| Afternoon| 0        | 576       | 30         | 56,7         | Tidak        |
| Evening  | 1        | 591       | 31         | 56,5         | Tidak        |
| Raining Sim. | 1    | 601       | 30,2       | 56,2         | Hujan        |

**Figure 11.** LCD Display Module Android Application when Testing

**Table 2.** Android Application Module Testing Results

| Trial On  | CO (PPM) | CO₂ (PPM) | Temp. (°C) | Humidity (%) | Weather Con. |
|----------|----------|-----------|------------|--------------|--------------|
| Daytime  | 1        | 573       | 31,9       | 56,9         | Tidak        |
| Afternoon| 0        | 576       | 30         | 56,7         | Tidak        |
| Evening  | 1        | 591       | 31         | 56,5         | Tidak        |
| Raining Sim. | 1    | 601       | 30,2       | 56,2         | Hujan        |

**Table 3.** Overall Tool Testing Results

| Output on LCD Hardware Monitoring | Internet Network | Output on Android Applications |
|----------------------------------|------------------|--------------------------------|
| ![LCD Hardware Monitoring](image1) | Same             | ![Android Application](image2) |
| ![LCD Hardware Monitoring](image3) | Different        | ![Android Application](image4) |
6. Conclusion
The hardware monitoring module was successfully designed to capture CO, CO$_2$, temperature, humidity and weather conditions, and display them on the LCD and send the data to Firebase. The Android Application Module can retrieve CO, CO$_2$, temperature, humidity and weather conditions data from Firebase, and display it in the appropriate column in the designed Android Application. The hardware monitoring module and the Android Application Module can use the same or different internet network connections and still be able to function properly.

7. References
[1] S. Apriyanti, Y. Fitrianingsih, and S. Pramadita, “Analisis Konsentrasi Karbon Monoksida pada Ruang Parkir Ayani Mega Mall Kota Pontianak,” J. Tek. Univ. Tanjungpura, vol. 1, no. 13, pp. 1–10, 2011.
[2] D. Prihatmoko, “Penerapan Internet of Things (IoT) dalam Pembelajaran di Unisnu Jepara,” J. Simetris, vol. 7, no. 2, pp. 567–574, 2016.
[3] A. Ikrom, “Telemetri Sensor Gudang Gas LPG Berbasis AT-MEGA 16,” J. Univ. Ahmad Dahlan, vol. 1, no. 1, pp. 2–12, 2016.
[4] Y. Liangdo and A. Wibowo, “Sistem Monitor dan Pengontrol Kadar Gas Karbon Monoksida (CO) dalam Ruangan,” Widya Tek., vol. 7, no. 1, pp. 155–167, 2008.
[5] S. Widodo, M. M. Amin, A. Sutrisman, and A. A. Putra, “Rancang Bangun Alat Monitoring Kadar Udara Bersih Dan Gas Berbahaya CO, CO2, dan CH4 di Dalam Ruangan Berbasis Mikrokontroler,” Pseudocode, vol. IV, no. 2, pp. 105–119, 2017.
[6] Y. S. Parihar, “Internet of Things and Nodemcu a Review of Use of Nodemcu ESP8266 in IoT products,” JETIR June, vol. 6, no. 6, pp. 1085–1088, 2019.
[7] B. B. L. Heyasa and V. R. K. R. Galarpe, “Initial Development and Testing of Microcontroller-MQ2 Gas Sensor for University Air Quality Monitoring,” J. Electr. Electron. Eng., vol. 12, no. 3, pp. 47–53, 2017.
[8] L. Thattazhi, C. Carvalho, S. Vernekar, N. Mascarenhas, and T. P. F. E. Fizardo, “IoT based Air Monitoring System,” Int. J. Sci. Res. Dev., vol. 6, no. 02, pp. 3909–3912, 2018.
[9] I. A. Abdulrazzak, H. Bierk, and L. A. Aday, “Humidity and Temperature Monitoring,” Int. J. Eng. Technol., vol. 7, no. 4, pp. 5174–5177, 2018.
[10] D. Perlanda, S. Alam, and S. Purwiyanti, “Alat Pengumpul Kopi Model Terhampar Secara Otomatis Berbasis Arduino Uno,” J. Rekayasa dan Teknol. Elektro, vol. 12, no. 1, pp. 1–9, 2018.
[11] P. Soni and K. Suchdeo, “Exploring The Serial Capabilities for 16x2 LCD,” Int. J. Emerg. Technol. Adv. Eng., vol. 2, no. 11, pp. 109–112, 2012.