A low-power scenario for IOT-based indoor air quality monitoring system at workplace

Fepri P Panghurian, Nico Surantha, Amalia Zahra

Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480

Abstract. This paper presents the development air quality monitoring system for indoor workplace that send measurement CO, CO2, PM2.5, temperature and humidity and send it via IEEE 802.11 b/g wireless communication. The problem is many sensor node for indoor air quality should not always send data to server in normal situation. It could spend power consumption and the data would send it periodically. Although the data send periodically, the sensor must handle emergency situation like fire. The results show that with controlled transmission interval not only spend power consumption but the device also good respond in emergency situation and user could monitor air quality remotely via web application.

Keyword: internet-of-things, indoor air quality monitoring, CO, CO2, PM2.5

1. Introduction

Most people with productive age spend their time working. In this day, many people work indoors and use air conditioner. The problem with this working condition may lead to illness like Sick Building Syndrome (SBS). Temperature in indoor workplace was one parameter cause Sick Building Syndrome[1] Not only temperature, measurement air quality involve organic compound, carbon monoxide, carbon dioxide are parameter for indoor air quality.[2] In this paper presents the development of open hardware system that monitor temperature, humidity, carbon monoxide level, carbon dioxide level and send the data with wireless infrastructure based on IEEE 802.11b/g standards.

Internet of Things (IoT), a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.[3] IoT make a new technology that make our world more smarter.[4] There are many implementation of IoT like controlling home appliance in smart home system[5][6], health monitoring system[7], agriculture[8], etc.

The device developed with Arduino platform with characteristic open device, easy for prototyping, high flexibility, support many sensor, and has huge user-based community. The device will attach with 4 sensors. The device using carbon monoxide sensor, carbon dioxide sensor, Pm2.5 sensor, temperature and humidity sensor. CO2 and CO using MQ7 and MQ 135 provided by flying fish. Based on datasheet from MQ7, the sensor maximum power consumption only 350mW, so it’s only 0.7 mA for this sensor working correctly. [10] MQ 135 based on datasheet less than 800mW, so it’s only 1.6 mA for this sensor working. The sensor also sensitive on air quality change. It works between 20 -
20000 ppm. [10] This sensor is also easy to calibrating and there are many libraries that work with this sensor. In this paper we will setup a system with low-power scenario that adaptive with working environment because the sensor not necessary always sending data to the server if there is not in emergency situation. It could spend more power that wireless sensor node use. There is web application dashboard, so building management or employee could see their building assets remotely. In this paper there are many sections. In section 2 discuss about previous paper that related to this paper. We will improve low-power method that previous paper is not handle it. Section 3 discuss about methodology. It tells about design prototype that we propose to achieve our target. Section 4 discuss about evaluation. After the system we build, we should test and evaluate if our system achieving our target. We will discuss about data that system generate and analyze it. Section 5 discuss about conclusion. After we evaluate our system, we will discuss about conclusion this research and future work about this research.

2. Related Works
Several solution have been made before, but with limited functionalities and features like a few sensor attached to device. Like this solution represent the subject paper, Measuring air quality using self-powered devices [9]. This device in solution attached with many sensor and for power it’s using solar panel. It was good idea implement solar panel in device. In this paper, the device was used for outdoor environment, power consumption that device used and solar panel generate wasn’t calculated and there wasn’t scenario how reduce power consumption.
Another solution that inspire this research with reduce power consumption with turning off the sensor when it is not used. The title of this paper was A Low-Power Wide-Area Network Information Monitoring System by Combining NB-IoT and LoRa. [11] It was built with LoRa as wireless sensor network and NB-IoT as data aggregator and processing. In software solution design in this paper implement if the device was in idle state and sensor not used for measuring, it would turn it off.
Another solution that considered with power consumption calculated and build it with proprietary device. The title of this paper was A Low-Power Wireless Sensor for Online Ambient Monitoring [12]. This paper was good to explain how to measure power consumption the device. In this solution, it has drawbacks on this paper, it lack on implementation the sensor because the device was proprietary and the device was soldered by manufacturer.
Based on previous IoT and low-power system solution. There was low-power strategy like using solar-panel that could power up many sensors attached it in WSN, implement turning off sensor logic if the device in idle state, using proprietary device that could powered up three years. So, we want to add low-power scenario if interval could adjust based on previous data and could setting based on situation in indoor workplace environment but this device should have good response when in emergency situation.

3. Research Methodology
To make the system, we should make system requirement about system design. There are many subsections in this section. General design approach, it explains about general architecture the system. It also explains about communication between component in the system. System hardware design, it explains about wiring component in wireless sensor node. System software design, it explains about flow logic the system especially firmware in wireless sensor node.
3.1 General Design Approach
For low power consumption, a system will implement logic in Arduino microcontroller to control sensor power, time for capturing air quality parameter and time for sending air quality data from WSN to database server. Regularly, the server makes simple analysis from historical data that stored in server and update time transmitting data configuration in WSN firmware.
Microcontroller firmware will have programmed with C/C++ programming language that support Arduino microcontroller IDE. We will be programming with Arduino IDE and it can downloaded free
at Arduino website. Microntroller can communication with sensor through analog or digital port connectivity based on sensor communication specification in datasheet.[13]

Figure 1. General Architecture Diagram

Figure 1 show general overview how component in architecture communication. First, sensor gathers air quality data and send it raw data to server via wireless network. Server will be processing data like processing analog value to air quality unit (ppm) and after that keep the data in database. Server also processing analysis for changing interval time in WSN configuration file. User could access web application, server will respond with information about air quality in the workspace. This server also installed web socket, so when there is new data submitted to server, user necessary will get the new data.

3.2 System Hardware Design

Figure 2. Wireless Sensor Network Schematic Diagram
Figure 2 tell about WSN schematic diagram based on Arduino board. Except temperature sensor, all sensor powered with digital PWM to control sensor if in sleep mode for power efficiency. Analog communication will be use to all sensor except temperature sensor. SD card storage is always on because microcontroller read configuration parameter that stored in SD card. SD card also storage temporary storage for WSN before they will send it to server. Before we use ESP8266 WiFi module, we must flash it with Arduino WiFi library module and ESP8266 WiFi module also simplified with adapter so it could fit in schematic and could use standard WiFi library that Arduino provide.

| Component            | Name                  | Power Specification  |
|----------------------|-----------------------|----------------------|
| CO2 Sensor           | MQ7                   | 5V max 0.7 mA        |
| CO Sensor            | MQ135                 | 5V max 1.6 mA        |
| PM 25 Sensor         | Sharp GP2Y1010AU0F    | 5V max 20mA          |
| Temperature and Humidity Sensor | DHT22              | 5V max 2.5mA         |
| WiFi Module          | ESP8266               | 5V 10μA-170mA        |
| Microcontroller      | Arduino UNO R3        | 5V                   |
| SD Card              | DF Robot SD card module | 3.3V 0.06 – 1.8 mAh |

Table 1 is list of all component in WSN. For microcontroller we will use Arduino UNO R3 because it is easy to custom and load firmware into Arduino platform. There are many sensor that support Arduino UNO R3 so in this research we use all module that comply with Arduino platform. For CO2 and CO Sensor we will use MQ 7 from flying fish. For dust or PM 25 Sensor we will use Sharp GP2Y1010AU0F from Sharp. Sharp GP2Y1010AU0F is optical sensor, when the was particle trapped into the sensor it will blocked the optic so it will affect resistor value. WiFi module we will use ESP8266. ESP8266 have model low-power mode when in idle state. Temperature and Humidity sensor, we will use DHT 22. For SD Card module we will use DF Robot SD card module Arduino. SD card will store all configuration and temp air quality data.
3.3 System Software Design

![Wireless Sensor Network Firmware Flow Diagram](image)

**Figure 3.** Wireless Sensor Network Firmware Flow Diagram

Figure 3 show block diagram about general overview how wireless sensor network working. First WSN read configuration in SD card. In SD card there is configuration file and temporary database. Configuration file consist of sensor read interval, transmission time interval, and temperature threshold. After read configuration file, WSN checking if there is a request for changing configuration. If there is a request, WSN will write new configuration into SD card. After that write new configuration or there is not request, WSN continue for checking sensor read interval for sensor read the environment. After that sensor checking emergency situation based on temperature threshold. In emergency situation, sensor always sending data to server so it will skip next step for checking transmitting time. If it not in emergency situation, WSN will go to transmitting step. In transmitting step WSN check if it the time for transmitting the data, if it yes it will send it to server or it will append the read data to SD card. In sending data to server, WSN always read all data in SD card, send it to server then clearing data in SD card.

4. Evaluation

After designing and implementation, the system should be evaluated. We will evaluate power consumption on device, web application and response the system on emergency situation. There are many things to do before we do evaluation the system, these jobs are:

- The device will capture amperage data for many intervals every 2, 6, 10, 14, 20, 30 minutes.
- Capture air quality data and amperage data with regular interval (every 2 minutes) for 3 days and conditional interval for 3 days.

After all data we get, we will analyze it and evaluate it.
**Power Consumption Evaluation**

Based on WSN schematic with many sensors attached, we measure power consumption based on interval when transmitting data to server. Based on configuration schema and interval, this is the result.

| Interval (Minutes) | Power Used (mA) | Lifetime With 20000mAh (hours) |
|-------------------|----------------|-------------------------------|
| 2                 | 264.01         | 75.75                         |
| 6                 | 253.56         | 78.88                         |
| 10                | 251.47         | 79.53                         |
| 14                | 251.74         | 79.45                         |
| 20                | 250.53         | 79.83                         |
| 30                | 249.28         | 80.23                         |

Table 2. show about power consumption based on interval when WSN send data to server. For example, if we have power bank with capacity 20000 mAh, so with interval 2 minutes that power bank only powered the device around 76 hours but if we used interval 10 minutes, we would power the device around 80 hours. Based that data, we could see the longer interval less power that device use but it will influence with report in application. In this research statically, we will use following configuration in Table 3 for the device.

| Condition          | Interval Used (Minutes) |
|--------------------|-------------------------|
| Peak Hour          | 2                       |
| Non-Peak Hour      | 10                      |
| Midnight           | 30                      |

Table 3. Configuration Decision for Peak Hour Transmission

After several days gather air quality data and power consumption the device. In first day when the device switch on, the device will gather the data and transmit it to the server every 2 minutes. In day 2, the server will analyze working hour in previous data and send it to WSN and become condition decision in Table 3. Midnight interval is configuration for time more than 18:00 until 08:00 on the next day. For peak and non-peak hour, device will decide it start use interval for peak hour if in previous data show the trend will go up and it pass the threshold and it decide is the end of peak hour if the trend go down and using interval non-peak hour. After analyze data, the server will send new configuration to WSN. WSN will get new configuration and rewrite his own old configuration with the new one.
Figure 4 shows line graph about CO2 in the workplace every 2 minutes. Based on mean formula, it shows that in that office average value is 246.77 (line green), that value Based on mean value. With processing data in the server based on average value around 9:56 and 9:58 am is start point for peak hour so on that hour, device is always update intensively and for the end of time 14:51.

![Figure 4. CO2 Air Quality Data](image1)

Figure 5 shows about power comparation data between every 2 minutes interval (left side) and scenario interval (right side). It shows that on the midnight, device only transmitting every 30 minutes it showed with when it transmitting it turn up and when idle with shallow line and when in peak hour it same as every 2 minutes interval it denser than on another time. On non-peak hour, it shows that a bit loose.

![Figure 5. Power Consumption Comaparion](image2)

| Interval Used    | Mean Current (mA) | Lifetime With 20000mAh (hours) |
|------------------|-------------------|---------------------------------|
| Every 2 minutes  | 256.02            | 78.12                           |
| Conditional      | 247.46            | 80.82                           |
Table 4 is shows about mean value for each interval used. Interval 2 minutes mean value consume more current than conditional. Power consumption for conditional interval less than 2 minutes interval. If we powered the device with powerbank 20000 mAh, for 2 minutes interval will still on for 78 hours but for conditional interval will stay around 80 hours.

**Air Quality Web Application**

Figure 6 show about air quality AQI on web application. The data that stored in database are calculated with AQI formula that show on Error! Reference source not found. on that day then it will show on web application with gauge meter chart.

\[
I = \left( \frac{(PM_{obs} - PM_{min})(AQI_{max} - AQI_{min})}{(PM_{max} - PM_{min})} \right) + AQI_{min}
\]  

(1)

Figure 7 show about real time monitoring air quality dashboard. After WSN send data to server, server that have websocket. It will broadcast to all subscriber and then the client will show new data that sensor posted.
**Emergency Situation Evaluation**

We simulate when emergency situation occurred with set on fire near sensor like on Figure 8 on midnight when the sensor was in 30 minutes interval mode. When sensor near the fire source, the temperature will high and when it near the threshold, the sensor will immediately send data to server without save it to SD card and send email to building manager like on Figure 9.

![Figure 8. Set Fire Near Temperature Sensor](image)

With good response detect abnormal temperature and alerting system via email, the system seems works very well and responsive.

**5. Conclusion**

Based on result and discussion, without always sending data from WSN to server in normal working environment. We able save the power consumption without loss information in system. Although not always sending data to server but in emergency situation it could monitoring the environment. With web application interface, employee or building management could monitor air quality the workspace remotely and know situation at workspace and analyze it via web application. For future work, this system should define the best interval for peak hour and non-peak hour automatically and air quality parameter that fit environment and also best for power consumption too.
Reference

[1] Yuliani Setyaningsih S & S 2003 Hubungan Antara Kualitas Udara Dalam Ruangan Berpendingin Sentral Dan Sick Building Syndrome Sains Kesehat. 16 374

[2] Kosa K H 1988 Indoor air quality. The Letest Sampling and Analytical Methods. vol 30

[3] International Telecommunication Union 2012 Overview of the Internet of things Ser. Y Glob. Inf. infrastructure, internet Protoc. Asp. next-generation networks - Fram. Funct. Archit. Model. 22

[4] Malche T 2017 Internet of Things (IoT) for building Smart Home System 65–70

[5] Putra L, Michael, Yudishtira and Kanigoro B 2015 Design and Implementation of Web Based Home Electrical Appliance Monitoring, Diagnosing, and Controlling System Procedia Comput. Sci. 59 34–44

[6] Khan A, Al-Zahrani A, Al-Harbi S, Al-Nashri S and Khan I A 2018 Design of an IoT smart home system 2018 15th Learn. Technol. Conf. 1–5

[7] Pradeep Kumar S, Samson V R R, Sai U B, Malleswara Rao P L S D and Kedar Eswar K 2017 Smart health monitoring system of patient through IoT Proc. Int. Conf. IoT Soc. Mobile, Anal. Cloud, I-SMAC 2017 551–6

[8] Athani S and Tejeshwar C H 2017 Suhas Athani*, CH Tejeshwar* 43–8

[9] Reilly K M, Birner M T and Johnson N G 2015 Measuring air quality using wireless self-powered devices Proc. 5th IEEE Glob. Humanit. Technol. Conf. GHTC 2015 267–72

[10] Arduino 2015 MQ Gas Sensors Datasheet Arduino

[11] Zhang X, Zhang M, Meng F, Qiao Y, Xu S and Hour S 2018 A Low-Power Wide-Area Network Information Monitoring System by Combining NB-IoT and 4662

[12] Folea S C and Mois G 2015 A low-power wireless sensor for online ambient monitoring IEEE Sens. J. 15 742–9

[13] Haouyu Z, Guomin L and Yaru L 2018 A Home Environment Monitoring Design on Arduino IEEE 2018 Int. Conf. Intell. Transp. Big Data Smart City 6–9