IoT-Based Air-Pollution Hazard Maps Systems for Ho Chi Minh City

Phuc-Anh Nguyen, Tan-Ri Le, Phuc-Loc Nguyen, and Cuong Pham-Quoc(✉)

Ho Chi Minh City University of Technology, VNU-HCM, Ho Chi Minh City, Vietnam
cuongpham@hcmut.edu.vn

Abstract. Hazard map is one of the major parts in smart city model. It gives citizens an overview of a particular hazard in different areas within the city. In this paper, we research and implement an IoT-based air-pollution hazard map system for Ho Chi Minh City. The system consists of sensor and gateway nodes, a server, and maps. The collected sensor values include temperature, dust, carbon monoxide (CO), and carbon dioxide (CO_2). Data collected by sensors is transmitted to gateway nodes and forwarded to the server. The server saves received data to database and queries according to request from users. A web application have been built to display the data and give users an overview of air pollution state around them.

Keywords: Internet of Things · Hazard maps · LoRa

1 Introduction

The fourth industrial revolution (industry 4.0) has brought to our life technologies which take part in changing the world. One of the most significant factors of Industry 4.0 is Internet of Things (IoT). The appearance of IoT applications has led to the release of smart devices like smart TV, smart home, or even smart city. Among these innovations, smart city is the main target that Ho Chi Minh city has planned on its way heading to Industry 4.0.

The goals of smart city are to enhance living quality of life for citizens and improve service quality of the local government. A simple example of smart city is an air pollution monitoring system which helps citizens to easily get information about environment quality of their current location. Air pollution is one of biggest threats for the environment and affects everyone's health. The pollution is also the reason of climate change and global warming. In the past few years, Ho Chi Minh city's air pollution has reached an extremely high level. However, people living in this city do not have many channels to access the information of how polluted is the air. In this paper, we design and implement IoT-based air-pollution hazard maps systems for Ho Chi Minh city which is a part of building a smart city and also a contribution to citizens's quality of life improvement.

Components of our system include sensor and gateway nodes, a server, and maps. Sensor nodes collect 4 types of sensing data: temperature, dust, carbon monoxide (CO), and carbon dioxide (CO_2). Sensed data will then be transmitted to gateway in an AES-128 encrypted packet whose length is 13 bytes by using LoRa. Gateway is based on Raspberry Pi 3 Model B board running Android Things (AT) OS. When gateway receives the packet, it will validate data in it and create an AES-256 encrypted JSON string to send 4 sensing values to server through HTTP protocol. Server built by Nodejs is responsible for receiving and decrypting packets sent from gateways, interacting with MongoDB database, handling request from users, and displaying sensing data to web application. Users can view air pollution data on web application developed using ReactJS and OpenLayers library.

2 Background and Related Work

In this section, we introduce an overview of similar systems in other cities and discuss background knowledge used for our work.

2.1 Related Work

Deployed in the cities of Belgrade and Pancevo, EkoBus system has been developed in collaboration between Telekom Serbia, City of Pancevo, and Ericsson. Aim of the project is to monitor a set of environmental parameters over a large area by using public transportation vehicles. It uses instruments mounted onto existing public transportation vehicles in order to monitor temperature, relative humidity, CO, CO_2, NO_2, and vehicle location [1]. Sensor nodes make measurements and periodically send collected results to the server application with the help of GPRS data communication for further analysis and database storage.

The management of air quality in Canada's national capital region requires the identification of small-scale pockets of air pollutants in an area of over 5,000 km^2 and with a population of more than a million inhabitants. In order to cover this large area, AURA earth satellite observations, supported by local air quality sampling and modelling, were applied to provide comprehensive information about the spatial distribution and dispersion of air pollutants. A combination of information from fixed, mobile and portable air quality monitoring stations at ground level as well as the Aura instruments including the Ozone monitoring instrument from space was selected as the most effective approach for mapping local air quality. Hourly mapping of six pollutants was complemented by local information on transportation and stationary emission sources, land use, commuting patterns and meteorology. The end result was hourly displayed in the Canadian Geospatial Data Infrastructure with the following parameters: NO_2, NO, NOx, PM2.5, O_3, and CO [8].

2.2 Background

LoRa: LoRa (Long Range) is a wireless technology developed by Semtech. It has become the most popular LPWAN technology all over the world [4]. Advantages of LoRa are numerous: ability to transmit data in a long range (can reach kilometers), low power consumption, high reliability, Doppler robustness, etc. LoRa operates on an ISM band (Industrial, Scientific and Medical radio bands) which consists of free license radio bands reserved internationally for the use of industrial, scientific, and medical purposes. The frequencies of the band vary by region and data can be sent on 433, 868, or 915 MHz frequency band. Maximum payload length of a LoRa packet is allowed in the world varies from 51 to 222 bytes which depends on the data rate [3]. Despite having low data rate, LoRa is a reasonable solution for IoT applications that transmit small data packets between sensor nodes and a server/cloud.

Components of Sensor Nodes: For sensor nodes, we use Nucleo STM32L053R8 board to control operations. This board has an STM32L053R8 microcontroller whose core is ARM® Cortex® M0+ that has high performance and low power consumption. The chosen microcontroller has high-speed embedded memory with 64 Kbytes of Flash memory, 2 Kbytes of data EEPROM, and 8 Kbytes for RAM. Moreover, the microcontroller also supports traditional peripherals such as I2C, SPI, I2S, USART, LPUART, and crystal-less USB. In the operation mode, it consumes 88 µA/MHz while also offering several sleep modes to reduce the power consumption.

Temperature sensor that we choose for the system is DS18B20. The sensor has digital output to achieve noise resistance ability and high precision. At the lowest resolution as 9-bit, DS18B20 only takes maximum 93.75 ms to convert temperature value. It can measure temperatures from $-55\,^\circ$C to $125\,^\circ$C with $\pm 0.5\,^\circ$C accuracy within range from $-10\,^\circ$C to $85\,^\circ$C.

CO sensors are MQ-7. This sensor can detect the presence of CO in the air with high sensitivity. Sensor's output supports both digital and analogue signals.

We choose SenseAir S8 LP as the CO_2 sensor. This sensor comes with miniature size and low power consumption. The sensor is based on non-disperive infrared (NDIR) technology to monitor CO_2. It is also built with self-correcting ABC algorithm. The algorithm constantly keeps track of the sensor's lowest reading over preconfigured time interval and slowly corrects for any long-term drift detected as compared to the expected fresh air value of 400ppm CO_2 [6].

Dust sensor which we choose for our sensor nodes is Sharp GP2Y1010AU0F. The sensor has low power consumption with 20 mA at max and 11 mA for average. It comes with an optical sensing system to detect dust in air and effectively detect fine particle. Output of the sensor is digital signal with typical sensitivity at $0.5\,\text{V}/0.1\,\text{mg/m}^3$.

We also integrate solar power in sensor nodes to extend life cycle of the nodes. Specifically, solar power management module DFR0559 is used to provide power for node's operation getting from LiPo battery and also to charge for the battery with power producing from solar panel. The module is controlled by CN3065 IC which has MPPT (Maximum Power Point Tracking) to maximize solar energy conversion efficiency under various sunlight. Battery can be charged by power from solar panel or through Micro USB port with maximum charge current at 900 mA. It also offers various features like Over charge/Reverse connection/Short circuit/Over current protections. Power output of the module is 5 V 1 A.

LoRa module Ra-02 is used for transmitting LoRa packets. It based on SX1278 chip from Semtech with receiving sensitivity as low as −141 dBm. The module has maximum transmit power at 18 dBm and 10 mW of output power.

Raspberry Pi 3 Model B: Pi is a mini computer which runs on operating systems with Linux core. Mainboard of Pi 3 already has components to make it like a normal computer such as CPU, GPU, RAM, MicroSD slot, Wi-Fi card, Bluetooth, 4 USB 2.0 ports, HDMI, etc. Raspberry Pi is developed by Raspberry Pi Foundation - a non-profit organization which targets to build multi-functional system that everyone can use for different purposes [5]. There are 3 OEMs responsible for manufacturing Raspberry Pi, including Sony, Qsida, and Egoman. Element14, RS Components, and Egoman are in charge of distribution. With a good specification and ability to be compatible with embedded operating systems, Raspberry Pi 3 is a suitable option for Gateway in a data collecting system.

Android Things: On May 7th, 2018, AT 1.0 was released at Google IO 2018. This is a OS that supports for IoT devices like smart speaker, CCTVs, etc with an Internet connection. Basically, AT was an update and renew version of Brillo - an OS based on Android for smart devices and IoT applications launched in the previous year. Similar to the original Android, AT was built as an open platform to support OEMs skipping the initial ecosystem development step and move straight to completing product process [2].

Below are some features of AT:

– Applications run on AT are coded with Java - one of the most popular object-oriented programming language in the present.
– The development process of an AT application is similar to mobile applications.
– Life cycle of an AT application is the same as an Android application.
– Similar to OTA (Over the air) updates on Android devices, developers can release updates for IoT devices in the same way with basic architecture that Google uses for their devices and services.
– Google Cloud Platform, Firebase, can easily be integrated into AT.
– AT is well compatible with Raspberry Pi 3 Model B. It provides APIs to use GPIO pins to communicate with other peripherals.

AES: AES (Advanced Encryption Standard) is an encryption algorithm which has been adopted by U.S Government. The algorithm is built based on the Rijndael Block Cipher developed by two Belgian cryptographers, Vincent Rijmen and Joan Daemen. AES works with a block size of 128 bits and cipher key with different lengths: 128, 192, and 256 bits. Expanded keys in AES are derived from the cipher key using Rijndael's key schedule. Most of operations in AES are performed with a finite field of bytes. Each 128 bits input block is divided into 16 bytes, arranged in 4 columns. Each column has 4 elements or a 4×4 matrix of bytes called state matrix. Depend on cipher key's length, the algorithm has different number of loops.

3 Proposed System

In this section, we introduce our proposed system with the aforementioned components. The system is suitable for deploying Ho Chi Minh City with reasonable cost.

3.1 Sensor and Gateway Nodes

Operations of our sensor node are controlled by STM32L053R8 microcontroller (as depicted in Fig. 1). The solar power management module DFR0559 is connected to a solar panel, a LiPo battery, and Nucleo board. The battery provides power for sensor node operations while solar panel produces power to charge the battery. These processes are managed by the solar power management module. Output of the module is provided for a Nucleo board which also supplies power for sensors. The flowchart of activities in sensor node is illustrated in Fig. 2.

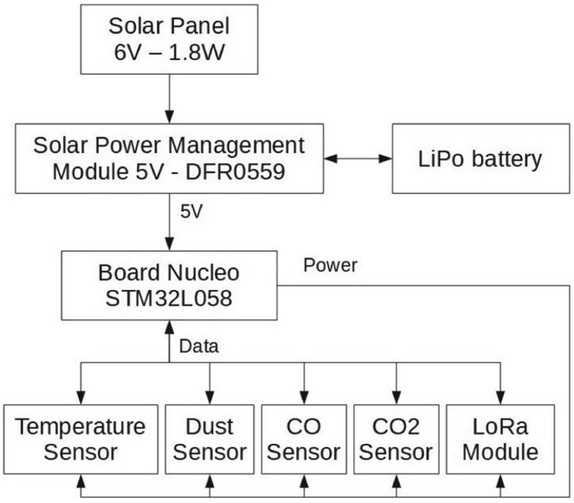


Fig. 1. Block diagram of sensor node

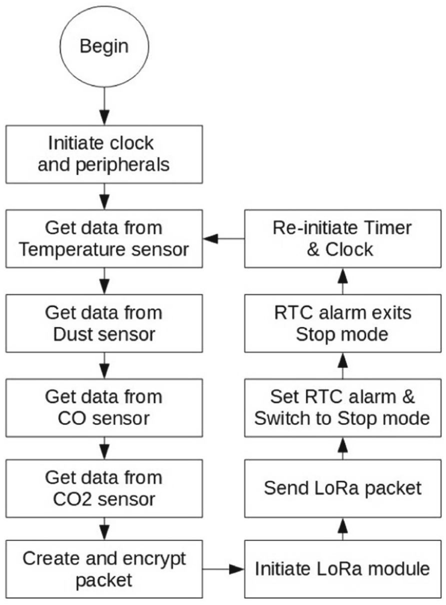

Begin
Initiate clock and peripherals
Get data from Temperature sensor
Get data from Dust sensor
Get data from CO sensor
Get data from CO2 sensor
Create and encrypt packet
Initiate LoRa module
Send LoRa packet
Set RTC alarm & Switch to Stop mode
RTC alarm exits Stop mode
Re-initiate Timer & Clock

Fig. 2. Operation flowchart of sensor node

To optimize transmitting time of a LoRa packet, payload size should be minimized. Currently, Cayenne LPP [7] is applied by the Things Network for packaging information in payload. Cayenne LPP follows maximum payload size requirement and can be reduced to 11 bytes. As illustrated in Table 1, data is divided into 3 fields: Data channel, Type, and Value. Our version forked from Cayenne to follow the way value is encoded. We modify the packet structure to 13 bytes in length that contains 4 types of data including temperature, CO, CO_2, and dust concentration as shown in Table 2. The packets are encrypted with AES-128 and sent by LoRa to gateway.

Table 1. Example of Cayenne LPP payload in a device with 1 temperature sensor

Payload (HEX)	01 67 FF D7	
Data channel	Type	Value
01 $\Rightarrow$ 1	67 $\Rightarrow$ Temperature	FFD7 $= -41 \Rightarrow -4.1\,^{\circ}$C

The main responsibilities of gateway are to communicate with sensor nodes to gather environmental data and to forward to the server (as shown in Fig. 3). The major requirement for gateway is to have wireless communication within 100 m with sensor nodes. Furthermore, the main requirement for gateway is having ability to connect to the Internet by technologies like GSM or WiFi for sending

Table 2. Payload structure of packet from sensor node

1 byte	1 byte	2 byte	1 byte	2 byte	...
SensorID	Data1 Type	Data1	Data2 Type	Data2	...

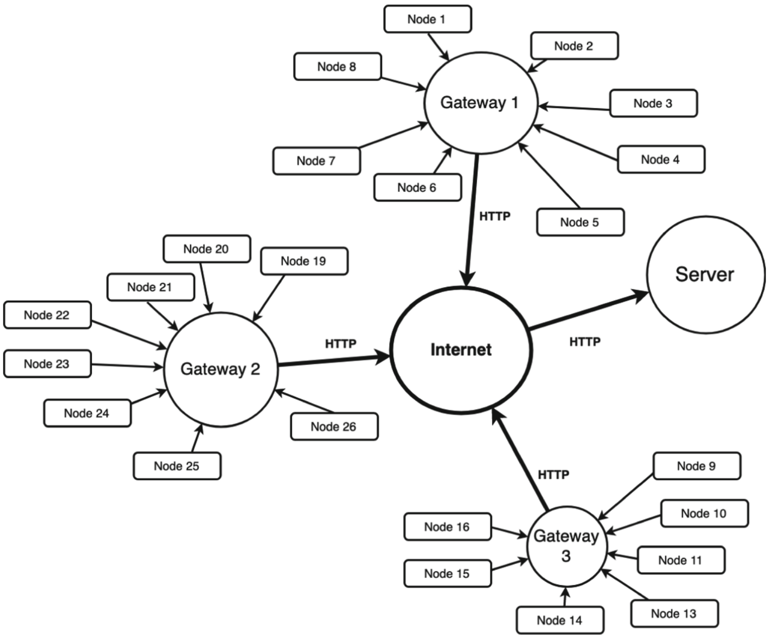


Fig. 3. Overview of gateway system

data to the cloud. In order to lower processing pressure on server, gateway is designed to filter received data. Only packets decrypted successfully and follow format will be sent to server.

Details of operations in gateway are illustrated in Fig. 4. At first, gateway registers to the server by sending a `POST` request that contains information about gateway (id, address, location, and name). Next, the `initLoRa` function is called to export necessary values to SPIdevices for setting LoRa module to running state. Most of the time LoRa module is in the sleep mode and will be woke up in 1 ms. The `onReceive` function is called to receive data. If there exists arriving data, the function sequentially saves data bytes until reaching 13 bytes of data as designed. In case received packets are shorter or longer than 13 bytes, gateway will skip these packets. When all 13 bytes of data received is valid, gateway will decrypt the packet and divide it into fields: sensorID, temperature, dust concentration, CO, and CO_2 concentration. Data will then be encoded to JSON string. Subsequently, gateway encrypts this string with AES-256 and sends to the server.

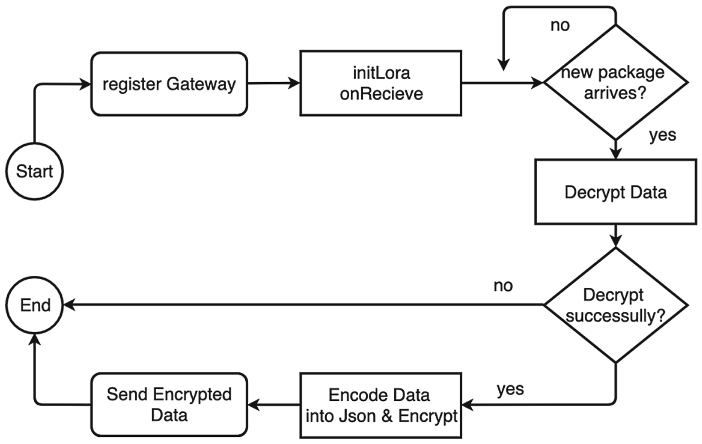


Fig. 4. Operation flowchar of gateway

3.2 Server and Maps

Data, collected from sensor nodes after being received by gateway, will then be transmitted to the server through the HTTP protocol. The server is responsible for gathering data from gateways, data decryption, saving them to database, and providing APIs to access data for clients.

The server architecture includes 3 modules (as shown in Fig. 5) with each module is in charge of distinctive function but can be combined to handle requests from client side. The three main modules in the server include:

- *Route module* defines API endpoints, pre-process attached data (decode, decrypt, ...), then call the Controller module to continue processing.
- *Controller module* interacts with database and accesses data based on inputs from the Route module.
- *Model module* is responsible for defining schemas in database. It combines with the Controller module to guarantee accessing right data from database.

To implement server as the above design, we use Nodejs to build the server and MongoDB for database that archives collected data. Using Nodejs helps the server to respond faster and gain better processing power. These advantages make Nodejs a right solution for frequently receiving data and supporting in real-time for client applications. Additionally, with collected data is discrete and does not have many relational constraints, the use of non-relational database like MongoDB speeds up queries and easily be expanded as the system grows.

After the implementation of the server and storing collected data, we next move to build a simple web application to display collected data as well as clarify development for the hazard map. ReactJS is used to build web application quickly and simply in combination with OpenLayers library to display a digital map. The application consists of 3 screens as follows.

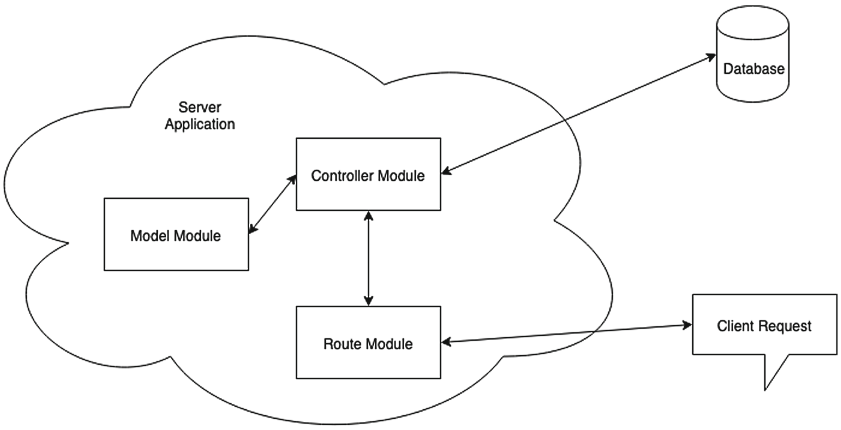
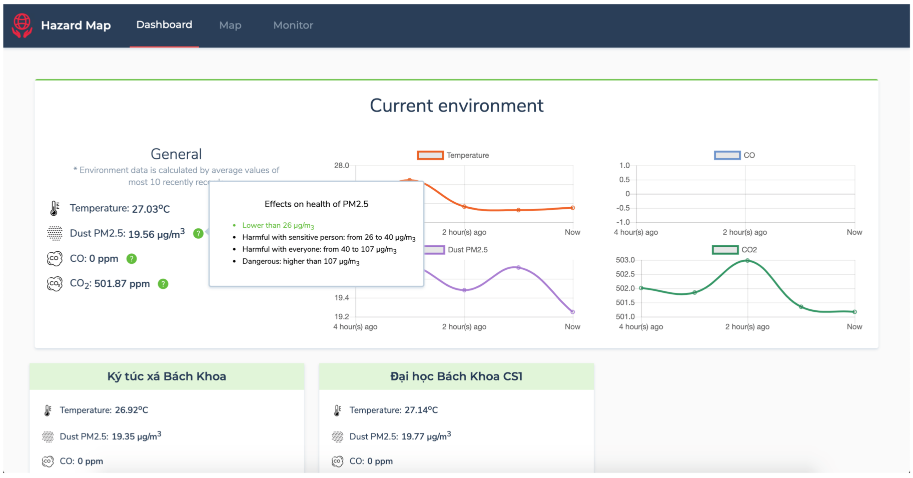


Fig. 5. Block diagram of server

Dashboard screen displays overview information extracted from gateways (as shown in Fig. 6). Information displayed includes average values of environmental parameters (based on 10 recent data collections), standard scale of health effects of parameters (will be showed when clicking on question mark ?́ícon), 4 graphs corresponding to 4 parameters (temperature, dust - PM2.5, CO, CO_2 concentration). Besides, the bottom gives a view of the list of gateways and average values based on 20 recent data collections. This section allows users to quickly get information of any gateway.



* Environment data is calculated by average values of most 10 recently rec

Temperature: 27.03°C

Dust PM2.5: 19.56 μg/m^3

CO: 0 ppm

CO_2: 501.87 ppm

Effects on health of PM2.5

- Lower than 26 μg/m_3
- Harmful with sensitive person: from 26 to 40 μg/m_3
- Harmful with everyone: from 40 to 107 μg/m_3
- Dangerous: higher than 107 μg/m_3

Ký túc xá Bách Khoa

Temperature: 26.92°C

Dust PM2.5: 19.35 μg/m^3

CO: 0 ppm

Đại học Bách Khoa CS1

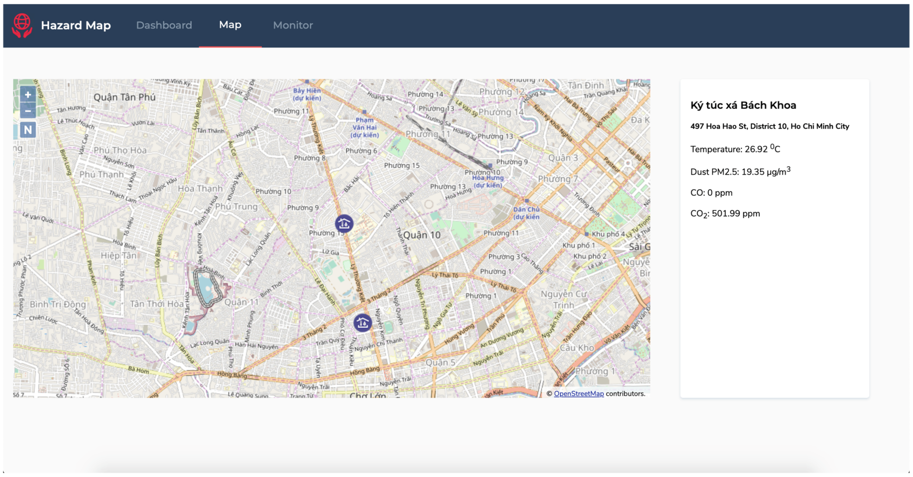
Temperature: 27.14°C

Dust PM2.5: 19.77 μg/m^3

CO: 0 ppm

Fig. 6. Dashboard screen

Map screen displays a digital map by calling APIs of OpenLayers (as depicted in Fig. 7). The default location is set to the main campus of Ho Chi Minh city University of Technology. We use APIs supplied by the library to implement some features, including displaying gateway's location on the map or locating current location of user. With a digital map, web application will give users an intuitive view of the around environment status and easy access to parameters of nearby gateways.



Ký túc xá Bách Khoa

497 Hoa Hao St, District 10, Ho Chi Minh City

Temperature: 26.92 ^{0}C

Dust PM2.5: 19.35 μg/m^3

CO: 0 ppm

CO$_2$: 501.99 ppm

Fig. 7. Map screen

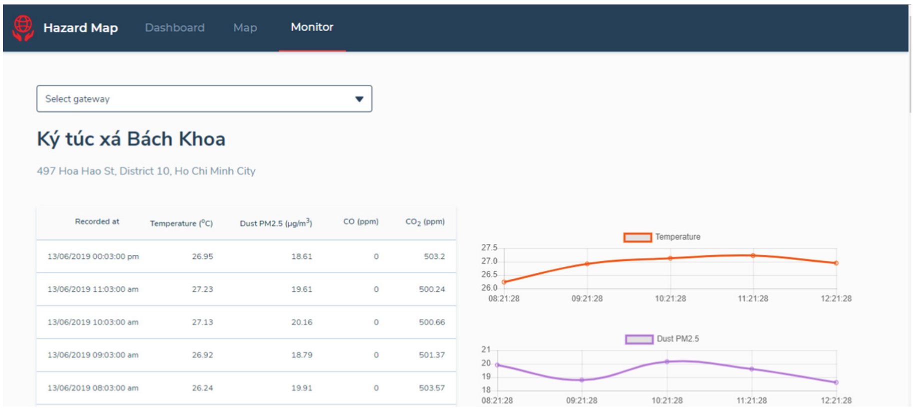
Monitoring screen displays collected data of each gateway in real-time (as shown in Fig. 8). Because the amount of data collected is huge, we limit this screen to show 20 recent data collections. The screen also shows 4 graphs corresponding to 4 parameters that enable user of monitoring instant change.

4 Deployment and Experiments

In this section, we describe our first prototype version deployed in Ho Chi Minh City.

4.1 Deployment

Sensor nodes (as illustrated in Fig. 9(a)) collects data of 4 sensors and sends to gateway through LoRa with AES-128 encryption. When gateway receives data, it will decrypt and validate data in the arrival packet. When data is valid, gateway creates a JSON string containing data of sensors and encrypts with AES-256. Gateway then sends the packet to server by HTTP protocol. Packets arrived at server will be decrypted and checked to store in database. New data arrived will be displayed on monitor screen on the web application as shown in Fig. 9(b).

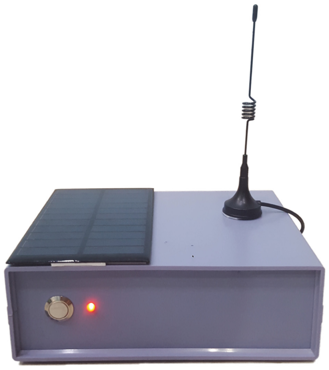


Ký túc xá Bách Khoa

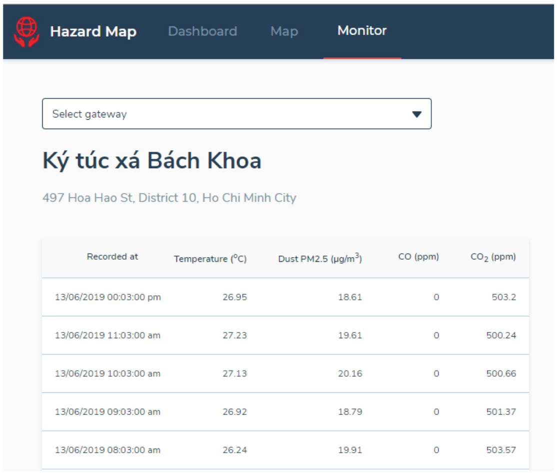
497 Hoa Hao St, District 10, Ho Chi Minh City

Recorded at	Temperature (°C)	Dust PM2.5 (μg/m^3)	CO (ppm)	CO$_2$ (ppm)
13/06/2019 00:03:00 pm	26.95	18.61	0	503.2
13/06/2019 11:03:00 am	27.23	19.61	0	500.24
13/06/2019 10:03:00 am	27.13	20.16	0	500.66
13/06/2019 09:03:00 am	26.92	18.79	0	501.37
13/06/2019 08:03:00 am	26.24	19.91	0	503.57

Fig. 8. Monitoring screen



(a)



Ký túc xá Bách Khoa

497 Hoa Hao St, District 10, Ho Chi Minh City

Recorded at	Temperature (°C)	Dust PM2.5 (μg/m^3)	CO (ppm)	CO$_2$ (ppm)
13/06/2019 00:03:00 pm	26.95	18.61	0	503.2
13/06/2019 11:03:00 am	27.23	19.61	0	500.24
13/06/2019 10:03:00 am	27.13	20.16	0	500.66
13/06/2019 09:03:00 am	26.92	18.79	0	501.37
13/06/2019 08:03:00 am	26.24	19.91	0	503.57

(b)

Fig. 9. (a) Sensor node; (b) Data received at server and displayed on Web Application

4.2 Maximum Transmitting Range of Sensor Node

We have set up 2 scenarios to test the maximum range that sensor nodes can send LoRa packets successfully to the gateway. In the first scenario, data is transmitted in an environment with buildings around. Maximum transmitting range recorded in this scenario is 120.65 m (as shown in Fig. 10(a)). In the second scenario, data in transmitted in a clear environment without obstacles. Maximum transmitting range reached 181.89 m in this scenario (as illustrated in Fig. 10(b)).

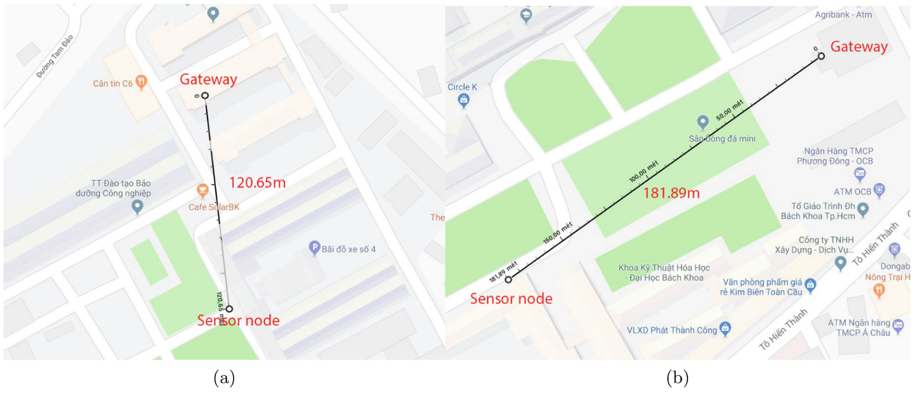


Fig. 10. (a) Experiment in environment with buildings around; (b) Experiments in clear environment

4.3 Power Consumption and Lifetime Estimation

Supply power for sensor node from a 5 V power source (V_{dd}) with a 1Ω resistor in between. We measure power consumption of the node by measuring power flows through the resistor using an Oscilloscope. Results are $V_T = 0.07485$ V for the average voltage in operation mode and $V_{sleep} = 0.04845$ V in sleep mode. In a period of 1 h, sensor node operates for 20.37 s and sleeps for the rest of time. The average voltage in 1-hour period (3600 s) is summarized in Eq. 1.

$$V_{\text{mean}} = \frac{V_T \times 20.37 + V_{\text{sleep}} \times (3600 - 20.37)}{3600} = 0.0486V \quad (1)$$

where V_{mean} is the average voltage in 1-hour period. Applying the Ohm's law, the average current is 0.0486 A.

Capacity of sensor nodes (P_{node}) is calculated from capacity of the circuit ($\text{P}_{\text{circuit}}$) and capacity of resistors (P_{R}), as depicted in Eq. 2.

$$P_{\text{node}} = P_{\text{circuit}} - P_{\text{R}} = V_{dd} \times I - R \times I^2 = 0.24301 \text{ W} \quad (2)$$

Let's call A is the battery capacity (Ah), T is total time using power from battery (hour), V is the battery voltage (V), η is the ratio of real battery capacity to capacity in specification (0.8 in normal). Life estimation of sensor node with power supply from a fully charged 1500 mAh 3.7V Li - Po battery is estimated in Eq. 3.

$$T = \frac{A \times V \times \eta}{P_{\text{node}}} = \frac{1.5 \times 3.7 \times 0.8}{0.24301} = 18.2707 \text{ h} \quad (3)$$

Assume that a Li - Po battery may last 500 cycles, the total time that sensor nodes can normally operate using power supply from a Li - Po battery and solar panel before battery replacement needed is estimation in Eq. 4.

$$T_{\text{bat}} = \# \text{cycles} \times \text{T} = 500 \times 18.2707 = 9135.33 \text{ h} = 380.639 \text{ days} \quad (4)$$

where T_{bat} is the battery life in hours.

5 Conclusion

Hazard maps systems have become a comprehensive analytical tool to provide information of vulnerability and risk for all users. From the government perspective, the system enables human operators to monitor the risk in a wide area that would help them on planning responses. As a citizen, hazard maps are the tool to actively deal with hazardous events. In this paper, we propose IoT-based Air-Pollution Hazard Maps Systems for Ho Chi Minh City to give people living in the city an overview of one of the biggest threats for health in urban areas. To achieve this aim, several design and techniques such as optimized payload structure or various screens for the web application have been used to complete the system design. By using LoRa to send packets between sensor and gateway nodes, sensor nodes do not need to be put in a close range with gateway like in systems using Wi-Fi or Bluetooth but still reach low power consumption.

Acknowledgment. This research is funded by Department of Science and Technology of Ho Chi Minh City under grant number 09/2018/HD-QKHCN.

References

1. Brković, M., Sretović, V.: Smart solutions for urban development: potential for application in Serbia. In: Proceedings of Regional Development, Spatial Planning and Strategic Governance (RESPAG) 2nd International Scientific Conference (2013)
2. Google Developers: Android things. https://developer.android.com/things/
3. LoRa Alliance: LoRaWAN^{TM} 1.1 Regional Parameters, January 2018
4. Mekki, K., Bajic, E., Chaxel, F., Meyer, F.: A comparative study of lpwan technologies for large-scale iot deployment. ICT Express **5**(1), 1–7 (2019). https://doi.org/10.1016/j.icte.2017.12.005
5. Raspberry Pi Foundation: What is a Raspberry PI? (2015). https://www.raspberrypi.org/help/what-is-a-raspberry-pi/
6. Senseair: Product Specification Senseair S8 LP Miniature CO2 sensor module width NDIR technique
7. The Things Network: Cayanne LPP. https://www.thethingsnetwork.org/
8. Urquizo, N., Spitzer, D., Pugsley, B., Robinson, M.: Mapping small scale air pollution distribution using satellite observations in a large Canadian city. In: 11th Conference on Atmospheric Chemistry of the Annual Conference of the American Meteorological Society, January 2009