LoRa-Based Air Quality Monitoring System Using ChatBot

Yao-Chiang Kan*, Hsueh-Chun Lin†, Han-Yu Wu* and Junghsi Lee*
*Yuan Ze University, Taoyuan, Taiwan
E-mail: yckan@saturn.yzu.edu.tw Tel: +886-03-4638800 ext. 7302
†China Medical University Hospital and College of Medicine, Taichung, Taiwan
E-mail: snowlin@mail.cmu.edu.tw Tel: +886-04-22053366 ext. 6303

***Abstract*— The air quality index (AQI) monitoring of campus area becomes much more popular and cheaper thank to the emerging of low power wide-area network (LPWAN) technique including the LoRa and NB-IoT. This AQI information is more precisely related to the health status of the people who live in the area most of the time. In this paper, a convenient AQI query interface is built by using LineBot which is the most popular instant messaging APP in Taiwan. One LoRa-base station is installed on the top floor of No. 7 building at Yuan-Ze University. The whole campus is basically covered by the LoRa signal emitted by this station. In the front-end, the LoRa-based AQI sensors of PM2.5, CO2, and CO are deployed in different locations of this building. The sensor data are collected and stored in a database automatically by a program. User can query AQI information from a chatbot and the most update data are retrieved from the database. In addition, a sensor setup Android APP is developed to accomplish the automation of the sensor deployment and data updating on the Line Bot. The developed monitoring system is established and running more than years and provides much more convenient accessibility for people in campus. The proposed monitoring and query system can be ported to any locations by modify the webhook setting in chatbot software as long as the LoRa station is available.**

I. INTRODUCTION

The air pollution issue is always one of the hot topics in environment protection research area. In Taiwan, abroad and local pollution sources often significantly increase the concentration of particulate material with 2.5μm diameter ($PM_{2.5}$). More and more studies shows some disease have strong correlation with the high concentration of $PM_{2.5}$ [1]. For example, the number of people suffering from pulmonary diseases has gradually increased, although the smoking population has been significantly decreasing year by year [2][3][4]. The relationship between the temporal and spatial distribution of air pollution and the number of patients in the geographical distribution is also an investigation focus. The air quality monitoring stations provided by the environmental protection agency (EPA) in Taiwan are located at about 90 locations on the main island of Taiwan [5]. The deployment of the AQI monitor stations by EPA is designed for large geographical area, say only two stations in the XinJung area with more than 43 thousand population and about 120 Km^2 area. This larger area AQI monitoring scheme inherently ignores a lot of air pollution hot spots in regular living areas. The typical fix-location air pollution hot spot is the brunch restaurant in Taiwan since it usually located at 1F in the residence area. Thanks to the emerging of low power wide-area network (LPWAN) technique including the LoRa and NB-IoT. Then the AQI monitoring stations can be deployed in the smaller area, for example, resident building, campus, and industrial area, etc.. In that way, more precise correlation between air pollution and community can be studied.

Therefore, this study proposes and accomplish a small area AQI monitoring system with chatbot interface. The proposed system can be installed easily and simpler than other monitoring system [7]. The LoRa-based AQI monitoring system is initially setup in the building No. 7 in Yuan Ze University, an area of about 2.4 square kilometers. Three types of sensors: PM2.5, CO2, and CO with LoRa transmission are deployed in the indoor and outdoor of the building. In addition to provide real-time AQI data query, it can also provide historical data statistics. The system and network architecture of the proposed monitoring system is described in section II. Then the required software components are detailed in section III. The results is shown in section IV and followed by the conclusions.

II. SYSTEM AND NETWORK ARCHITECTURE

The system and network architecture of the proposed AQI monitor system mainly includes the front-end devices, a LoRa gateway, backend servers, network protocols employed and data flows as shown in Fig. 1. The front-end devices include sensor nodes with LoRa communication function, mobile devices running the sensor nodes setup APP, and the mobile phone using the chatbot by Line (LineBot) [4] to query the AQI data. To facilitate the AQI monitor system and query using LineBot, the MQTT broker service, self-developed RESTful API service, webhook service of LINE APP and database with time series support have to be provided. In this study, those services are implemented on a 1U rack-mount server running the CentOS operating system (OS). Several network protocols and three main data flows are involved in this proposed architecture. The employed protocols and data flows will be elaborated in the following paragraphs.

For data flow I in Fig. 1(b), the LoRa-based AQI sensors transmit the sensing data to the LoRa gateway via the LoRa protocol. After receiving the sensor data, the LoRa gateway adds other information such as MAC and current timestamp with the received data, forms a new packet, and then sends the packet to the MQTT broker. The MQTT broker then predefined topic. A node.js program is implemented to function as a MQTT client subscribed to the specific topic, retrieve the required information, and then write that information into various appropriate tables under an AQI database using SQL syntax. In this way, the sensed AQI information is stored in the database periodically with chronological order. The database is hosted by the open-source PostgreSQL server with time series data plugin.

The AQI data query and display function is implemented through data flow II in Fig. 1(b). The query is evoked by clicking on specific menu items in the LineBot, a predefine term indicated the query request shows in the chat widow and then it is sent to Line server. The Line server then sends that term to the webhook URL registered in the Line developer server. The webhook program on our API server picks up the term, query AQI data from internal SQL server, and package those information in so-called flex message format, a display format extension by Line Company. The packaged message is sent back to Line sever and a formatted AQI information is display in the chat windows finally. The flex message is similar to HTML but with more formats designed for displaying in chat window.

In order to have the sensing data of the new installed sensors display on the chat window right after it is deployed, a sensor setup APP is built. The MAC and type of the new installed sensors is keyed in the registration page of the APP and those information is sent to API server after user click submit button. A corresponding RESTful API on server picks up those information and write those into internal database server. A trigger of that database is setup to inform the webhook to update the sensor list. Then the information of the new installed sensor will be shown in chat window for next query request.

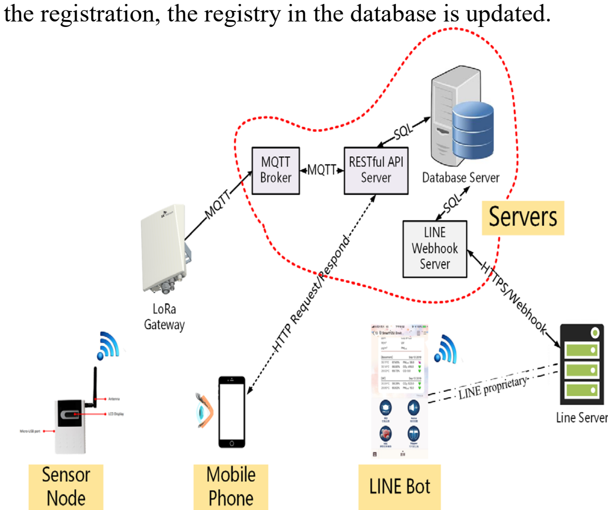
III. Software Developed

Follow the presentation in previous section, several software components have to be established to build a workable AQI monitoring system with LineBot query function. The first one is AQI data monitoring and storing service, the second is Line webhook service, and followed by the sensors setup APP.

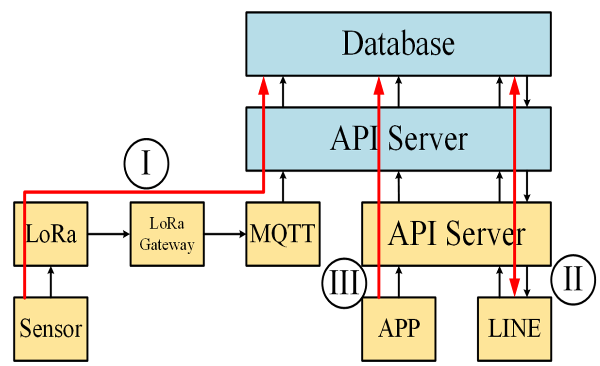
A. *AQI Data Monitoring and Storing Service (ADMSS)*

The AQI data monitoring and storing service mainly monitors the AQI sensing data that appears in the subscribed MQTT topic, parses data from the registered sensors and then writes those AQI data and timestamp information into the corresponding table of the database via SQL syntax. For the detailed procedure of the ADMSS is illustrated in Fig. 2.

In the initial stage, a database client is created to connect to the database server and prepare to retrieve data form it. Then the local online sensor registry (LOSR) will be synchronized with corresponding registry table in the database. The ADMSS then evokes a thread to listen the notification from the database. When the user setup a new sensor and completes the registration, the registry in the database is updated.



(a) Network architecture



(b). Protocols and data flow

Fig.1. System and network architecture

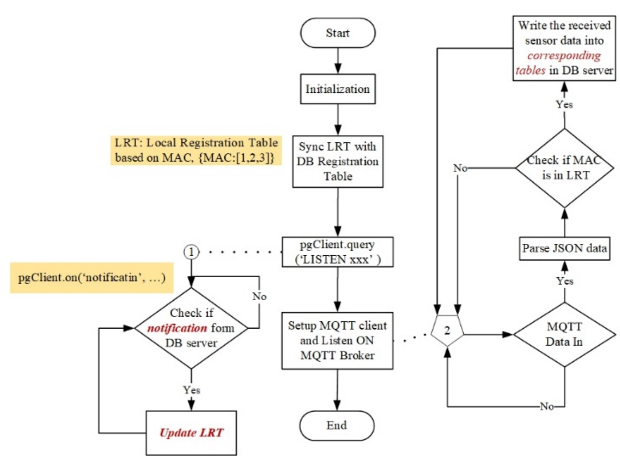


Fig. 2. The flow chart of AQI data monitoring and storing service

At the same time, this change in database triggers a notification to ADMSS. The sub-process ① in ADMSS update its LOSR by retrieving the information from the received notification. Finally, the ADMSS creates an MQTT client, subscribes to the predefined MQTT topic, listen to the incoming data, and evokes sub-process ② to parse the data from the registered sensors. As long as the published data show up, it is filtered, parsed and write into the database. In sub-process ②, the registered sensors are identified if the MAC address in the coming data is also in the LOSR.

B. *Line Webhook Service*

The Line webhook service deals with messages showing up in the chat widows of the corresponding LineBot. Those messages may be created by users input or menu item clicked from the chatbot menu, so-called rich menu in Line's terminology.

The webhook service is implemented by using RESTful API written in node.js programming environment. The RESTful API is basically accessed by using http or secure http (https) protocol. Each API should provide a http or https URL address and a path to the processing program. After the webhook is evoked, the node.js program create the https serverlet and listen to the incoming request from Line server. As long as there is data coming into the port, this program parse the incoming request packet and check if the sent message is the designed magic work, say &AQI pattern. If not, a help message showing commands is sent back to user's chat window. If the magic word is matched, the AQI query is sent to database server, responses from database server are parsed and reformatted into the flex message format. The reformatted messages is then send back to Line server and show on chat window of the LineBot finally.

C. *Sensors Setup APP*

The sensors setup APP provides a user interface (UI) for installer to key in the required information of sensors and send those data to API server. The corresponding RESTful API then write received data into database and evoke the LOSR trigger function. Hence, APP program and a RESTful API are required to accomplish this sensors setup APP.

The UI of the APP is shown in Fig. 4. The registration information page appears first and there are seven items listed in this page. Except the MAC field, other items comes with a button which show up the corresponding information if its button is clicked. For example, click button on the right side of field "Type", then the existing sensor types and their units are pop up as shown in Fig. 4(a). The above information for each item are actually retrieved from a RESTful API that query information about each one from the SQL server. As long as all fields are filled or chosen and the register button is clicked, the information is then sent to another RESTful API, data are write to database and response dialog is popup as shown in Fig. 4(b). As mentioned in section III.A, a trigger of the database is evoked and notification is sent to ADMSS. The new AQI information will be included in the message displayed on chat windows at next query request.

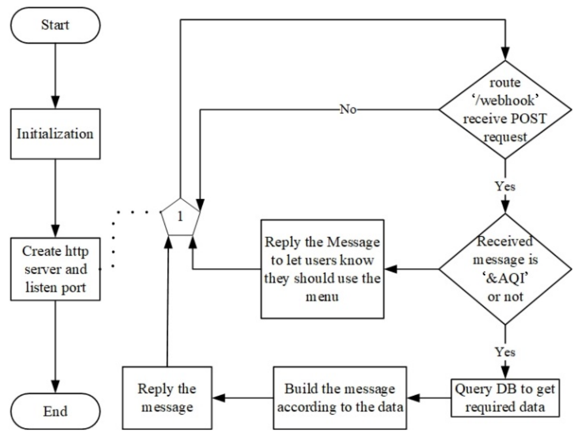
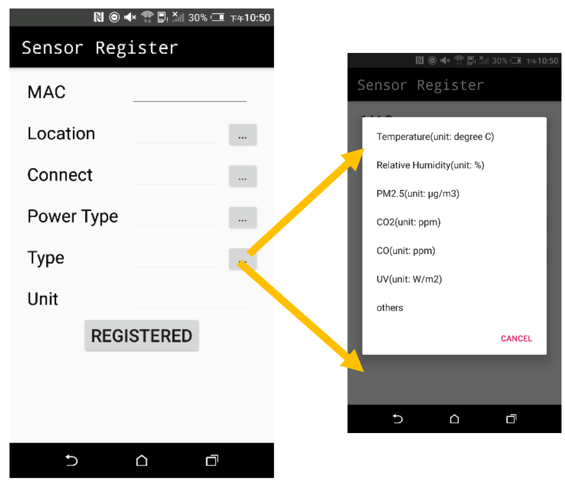
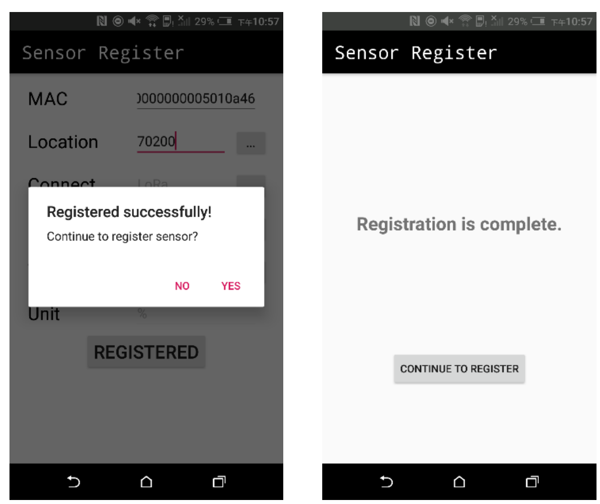


Fig. 3. The flow chart of LineBot webhook service



(a) Registration information page



(b) Response page after submit

Fig. 4. User Interface of the proposed APP

On the API server, two RESTful APIs are implemented to fulfill the functions required by the above APP. First one is to send back the information corresponding to each item on the APP's first page. The other one is to write sensor's registration information to suitable table in the database. The former is provided by using the http GET method with path /sensorData which retrieves data from database and send response in JSON format to APP end. The latter employs the http POST method with /newSensor which parses the request from APP and writes information into database

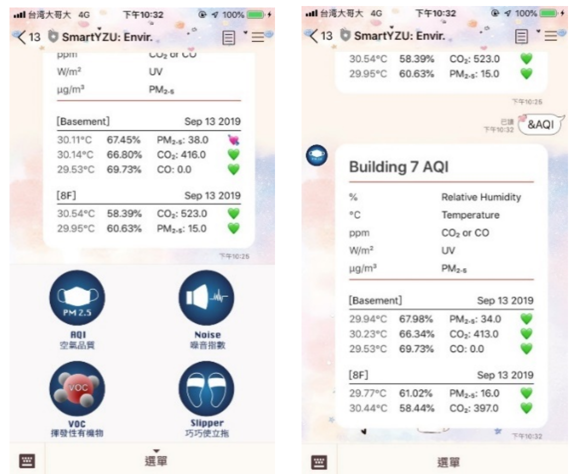
IV. Results

Two main user interfaces are designed in this system, sensor setup APP UI and LineBot chat window. The former UI is described in section III.C already. The latter UI basically presents the menu items for choosing and displays the AQI information after query. The menu in proposed LineBot is shown in Fig. 5(a). Four icons in this menu: AQI, Noise, VOC, and Slipper. Click the AQI icon and the current AQI information in building 7 at Yuan-Ze university is shown in the chat window as shown in Fig. 5(b). That information is formatted with a flex message format provided by Line. The top block in the message is units for different types of AQI parameters. Below the unit block, AQI information at different locations are listed by the order of number of floor. The temperature, humidity, AQI type with current values, and a heart icon in each row of AQI information block. Currently, the deployed sensors include $PM_{2.5}$, CO_2, and CO sensors. For each type of AQI, a threshold value can be defined. If the current value is higher than the threshold, then a heart with an arrow in red color is shown at the end of that row according to Fig. 5(a).

Since the AQI information is also stored in database by the ADMSS in section III.A, hence the history report can be evaluated. The hourly concentration tendencies of $PM_{2.5}$ at different floors on March 25 2020 are shown in Fig. 6. There are four $PM_{2.5}$ sensors deployed at semi-open spaces on 3F and two of those in stairwell locations at 1F and 3F. In the figure, the solid line with start marker indicated the hourly values from the nearest monitoring station provided by the Environmental Protection Administration (EPA) in Taiwan. Usually, there are two to three AQI monitoring stations by EAP in a city area. As shown in the figure, the locally measured concentration of $PM_{2.5}$ by the proposed system is higher than those by the nearest EPA station on that day. More analysis can be performed from database provided by the proposed AQI monitoring system.

V. Conclusions

A LoRa-based AQI monitoring and query system using chatbot is established. This system has been setup, runs more than years, and huge AQI information is stored. A sensor setup APP is also built to accomplish the data automatic update in LineBot query. The proposed system can be ported to any building or area by just change the LineBot webhook URL setting for different locations. However, there are still some functions can be developed in the future. For example, the summary report by a specified period is not implemented with the LineBot yet. Also, on-time notification of the malfunction sensor due to loss of power or hardware failure is another interesting one to be implemented.



(a) Menu (b) AQI Message

Fig. 5. Picture of the designed LineBot

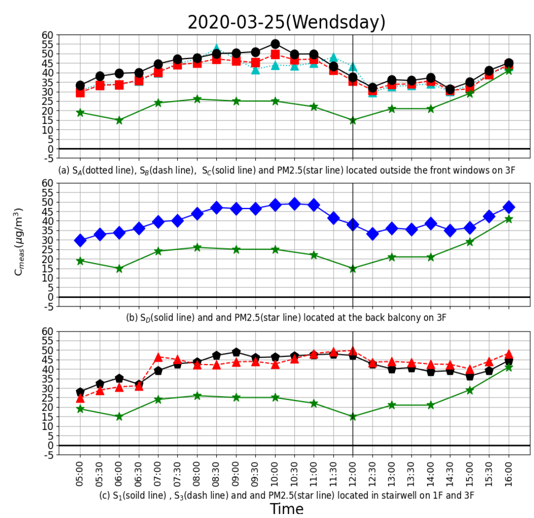


Fig. 6. The hourly concentrations tendency of PM2.5 at different floors on March 25 2020.

Acknowledgment

This work was funded by the Ministry of Science and Technology, Taiwan under the Grants MOST 106-2221-E-155-020, 107-2119-M-039-002, 109-2121-M-039-001 and Wan-Tz-Wong (萬磁王) project by the FarEastTone Company. The authors also appreciated Ms. W. M. Lin for her hard work and effort in helping building the APP.

References

[1] K.-C. Chu and M.-Y. Xiao, “A Study on the Correlation between Breast Cancer and Air pollution,” *IEEE/ACM Conference on Advances in Social Networks Analysis and Mining*, Sydney, Australia, July 31- August 03, 2017.

[2] S.-H. Wu, “Attention on Lung Health: Invisible Killer in The Air,” *National Health Insurance Magazine*, Taiwan, no. 116, pp. 38-41, May 2015. (In Chinese)

[3] O. K. Kurt, J. Zhang, and K. E. Pinkerton, "Pulmonary health effects of air pollution," *Current opinion in pulmonary medicine*, vol. 22, no. 2, p.138, 2016.

[4] G. S. Ajmani, H. H. Suh, and J. M. Pinto, "Effects of ambient air pollution exposure on olfaction: a review," *Environmental health perspectives*, vol.124, no. 11, p. 1683, 2016.

[5] Environment Protection Administration in Taiwan, Air Quality Monitoring Web Site. [Online]. Available: ttps://taqm.epa.gov.tw/taqm/tw/default.aspx

[6] Using Flex Messages - LINE Developers [Online]. Available: Retrieved from https://developers.line.biz/en/docs/messaging-api/using-flex-messages/

[7] Y.-L Chiang, C.-L. Hsieh, and etc., “Urban Area $PM_{2.5}$ Prediction with Machine Methods: An On-Board Monitoring System,” 12th *International Conference on Sensing Technology*, Limerick, Ireland, pp.25-30, 2018.