A WiFi-enabled Indoor Air Quality Monitoring and Control System: the Design and Control Experiments

Xiaoke Yang, Lingyu Yang, and Jing Zhang

***Abstract*— Particulate matter pollution becomes an increasingly important problem in developing countries and poses a hazard to human health especially in indoor environment. At the same time, modern buildings are equipped with advanced sensing and control technologies, offering control and optimisation capabilities over a large range of parameters. This paper proposes an open platform of a WiFi-enabled indoor air quality monitoring and control system, which could be incorporated into such a 'smart building' structure. The complete software and hardware design of this system is presented, along with a series of control experiments. The proposed system operates over an existing WiFi wireless network utilising the MQTT protocol. It is capable of monitoring the indoor air quality as well as controlling an air purifier to regulate the particulate matters concentration. Experiment results under a real world office environment demonstrate the effectiveness of the proposed design.**

***Index Terms*— indoor air quality control, particulate matter pollution, smart building, Internet-of-Things, open-source hardware**

I. Introduction

"Outdoor air pollution is a major environmental health problem"[1] worldwide, especially in areas with low- and middle-income[1], [2]. Among all types of pollutants, "particulate matter affects more people"[1] and attracts increasing attention of researchers recently. Airborne particulate matter consists of solid and liquid, organic and inorganic substances and is usually grouped into coarse particles and fine particles. Fine particles sometimes refer to particles with aerodynamic diameters of and less than 2.5 μm, or simply PM2.5[3]. PM2.5 is "strongly associated with mortality and other endpoints such as hospitalization for cardio-pulmonary disease"[3]. Through infiltration, natural and mechanical ventilation, outdoor pollutants can go indoor with air movements. Furthermore, biomass burning, building materials, and furnishings, etc. may also contribute to indoor particulates, which directly affects the health and comfort of the building occupants[4].

Despite the aforementioned problems of outdoor and indoor air pollution, the other side of the coin is the fast technological development of smart structures, smart buildings, and the Internet of Things (IoT). Within such a 'smart' vision, a variety of sensors, actuators and subsystems are incorporated into the building and interconnected to deliver monitoring, management and optimisation functionalities on illumination, temperature and air quality, etc., in order to improve the health and productivity of the building occupants and reduce the environmental impact of the building[5], [6].

Concerning the problem of indoor air pollution, and riding such an IoT and smart technologies trend, this paper proposes an open platform of a WiFi-enabled indoor air quality monitoring and control system. With detailed hardware and software design elaborated in the paper, the openness of the system reaches further to its interconnectability, which is built upon an open and commonly-used IoT protocol and allows the system be incorporated into a larger management and optimisation structure. The hardware and software design in the paper is also publicly available at `https://github.com/teancake/Indoor-Air-Quality-Monitoring-and-Control-System`, released under the MIT licence.

There are quite a few similar but different works in literature. For example, [7] proposes a WiFi-based indoor air quality monitor with multiple sensors in order to provide a low-cost alternative to commercially available products. [8] describes a sensor network measuring vibration, temperature, humidity, illumination, as well as the electrical load of the air conditioning system, in order to achieve a balance between the building's electricity load and occupants' comfort level for sustainability and energy efficiency. [9] presents an indoor air quality monitoring system using a wireless sensor network (WSN) and a web interface for the ease of use and maintenance. In terms of air quality control, [10] and [11] utilise multivariate linear and nonlinear predictive control strategies respectively and [12] analyses the possible uncertainties in an air quality control system.

The structure of the rest of the paper is as follows. Section II describes the hardware design. Section III covers the software design. Detailed control algorithm and control experiments are presented in section IV, and section V summarises the paper and proposes possible further work.

II. Hardware Design

The hardware structure of the indoor air quality monitoring and control system is shown in Fig. 1. It has three individual units, the server, the sensor unit, and the control unit, interconnected via a WiFi network. An air purifier is connected to the control unit.

A. The Server Hardware

The server is a Raspberry Pi 2 Model B mini-computer installed with a GNU/Linux-ARM operating system. The server is connected to the WiFi network via a wireless router.

Xiaoke Yang, Jing Zhang, and Lingyu Yang are with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, 100191, China. E-mail: `dr.x.yang@ieee.org`, `zhangjing2013@buaa.edu.cn`, `yanglingyu@buaa.edu.cn`.

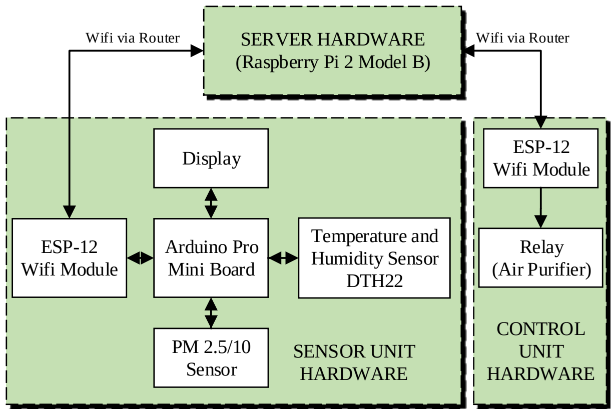
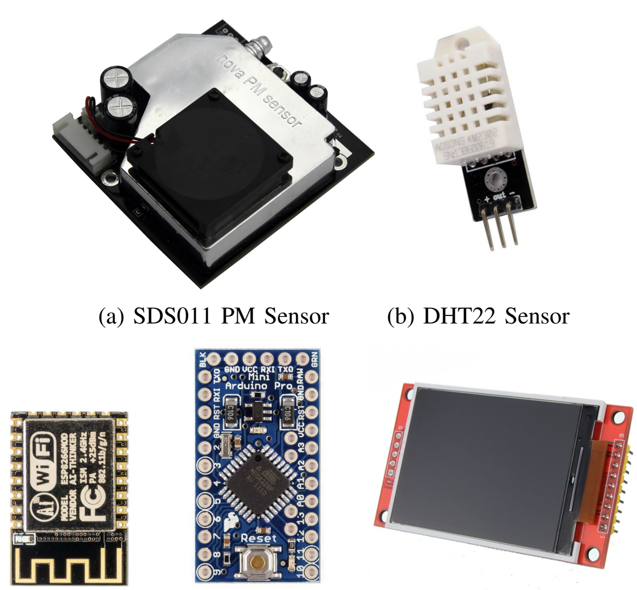


Fig. 1: Hardware structure of the indoor air quality monitoring and control system.

B. The Sensor Unit Hardware

As shown in Fig. 1, the sensor unit consists of a particulate matter (PM) sensor, a temperature and humidity sensor DHT22, a WiFi module, a microprocessor, and a display module. The individual modules used in the hardware design are shown in Fig. 2.



(a) SDS011 PM Sensor (b) DHT22 Sensor

(c) ESP-12F (d) Arduino (e) 2.8" LCD Screen

Fig. 2: Individual modules of the sensor unit hardware.

1) The PM sensor: The SDS011 PM sensor in Fig. 2a is capable of sensing particulates of diameters between 0.3 to 10 μm in the air. Both PM2.5 and PM10 concentrations within the range of 0-999.9 $\mu g/m^3$ can be captured and output with a serial interface at a frequency of 1 Hz. A fan mounted on top of the sensor actively takes in air for measurement[13].

The laser diffraction method as described in [14], [15] is the underlying working principle of the SDS011 PM sensor. The sensor has a built-in diode laser and a sensing photodiode aligned at a right angle with the laser beam. The laser beam illuminates the air in a duct and scattered by particulates floating in the air duct. The scattered light is picked up by the sensing photodiode as electric pulses, the height of which correlates with the particle size. The electric pulses are amplified and sampled by a micro-processor through an analogue-to-digital converter. The distribution of the pulses within a certain amount of time is computed and thus the distribution of the size of the particulates.

2) The temperature and humidity sensor: The DHT22 sensor (also called AM2302) in Fig. 2b consists of a capacitive humidity sensor and a thermistor, together with an 8-bit microprocessor inside for analogue-to-digital conversion and temperature calibration. It uses a single-line serial data interface which is capable of delivering measurement at around 2Hz. The sensor has a humidity sensing range of 0-100% with 2-5% accuracy and a temperature sensing range of -40°C to 80°C with 0.5°C accuracy.

3) The WiFi module: The ESP-12F in Fig. 2c is a WiFi module incorporating an ESP8266 chip and an on-board antenna. The ESP8266 chip is a system on chip (SoC) integrating an 80MHz 32-bit microprocessor Tensilica L106, 4MB flash memory, and WiFi functionalities supporting the standard IEEE 802.11 b/g/n agreement and the TCP/IP stack. The ESP-12F module can be programmed directly or be controlled by the Espressif AT command via the UART interface. The ESP8266 also has a number of GPIO pins.

4) The microprocessor and the display module: The microprocessor used in the hardware design is the 3.3V 8MHz version of the Arduino Pro Mini Board, and the display module is an SPI-interfaced LCD screen, as shown in Fig. 2d and 2e.

5) Hardware interconnections: Connections among the above modules are illustrated in Fig. 3. Apart from the DHT22 sensor, all other modules are powered by a 5V voltage source. Through various interfaces, e.g. UART, SPI, etc, the Arduino board samples the sensor measurements, manages the WiFi module, and transmits data over wireless connections. The finished sensor unit hardware is shown in Fig. 4.

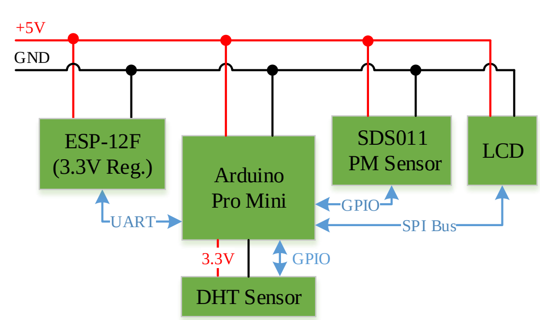


Fig. 3: Interconnections among modules of the sensor unit hardware.

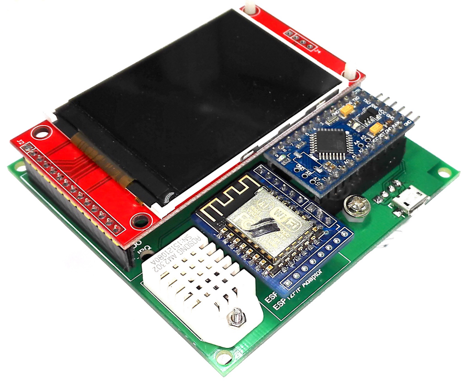


Fig. 4: The finished sensor unit hardware.

C. The Control Unit Hardware

The core component of the control unit is a relay controlled by the ESP-12F WiFi module. Since the control unit acts as a switch for the air purifier, it is convenient to draw power directly from the mains electricity. An AC-DC converter is employed to convert the mains current into the 5V DC current. The block diagram of the interconnections within the control unit is in Fig. 5 and the finished hardware is in Fig. 6.

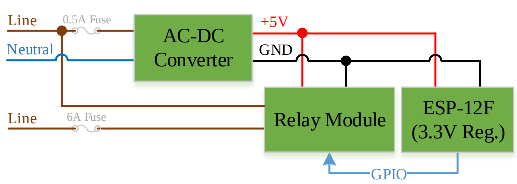


Fig. 5: Interconnections among modules of the control unit hardware.

Fig. 6: The finished control unit hardware.

III. Software Design

The software structure of the indoor air quality monitoring and control system is illustrated in Fig. 7. As shown in the figure, it consists of the server software, the sensor unit software, and the control unit software, each implemented on the corresponding hardware platform.

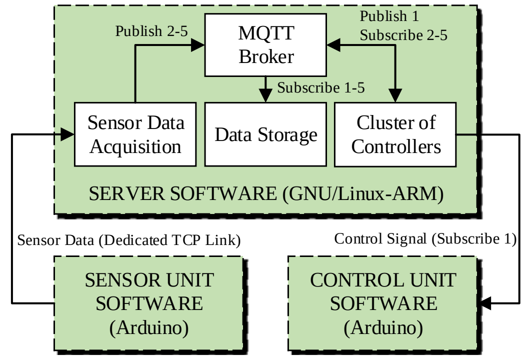


Fig. 7: Software structure of the indoor air quality monitoring and control system.

A. The Server Software

The server software runs on the Raspberry Pi board within the Arch Linux ARM operating system. The software, which includes a sensor data acquisition program, a data storage program, and a controller program, is built around an MQ Telemetry Transport (MQTT) broker as shown in Fig. 7. These three programs, together with the MQTT broker, run as system services in the GNU/Linux-ARM operating system, enabling the power-on-auto-start and restart-on-failure capabilities of the server programs.

1) The MQTT broker: The MQTT broker implements a message passing system using the MQTT protocol. The MQTT protocol is a lightweight message transport protocol commonly used in the Internet-of-Things (IoT) applications[16], [17] and normally runs on top of the TCP/IP stack. It has a central server called the broker, which manages connections from clients and distributes messages among them. Connections can either be open or authenticated by a user name and a password. The Transport Layer Security/Secure Sockets Layer (TLS/SSL) protocol can also be used to give encrypted communication over the TCP connections for improved security. The MQTT messages are grouped under topics. Clients can publish and subscribe messages of a certain topic. The broker then manages the publications and subscriptions and distribute incoming messages to target clients. This publish-subscribe architecture effectively decouples the publisher and the subscriber, enabling 'hot plugging' of the clients.

There are five messages defined in the system, as shown in Table I.

TABLE I: MQTT message topics of the indoor air quality monitoring and control system

Index	Topic	Type	QoS
1	office/switch/airpurifier	boolean	0
2	office/sensor/airquality1/pm2.5	float	0
3	office/sensor/airquality1/pm10	float	0
4	office/sensor/airquality1/temperature	float	0
5	office/sensor/airquality1/humidity	float	0

In Table I, the first message describes the status of the air purifier, which corresponds to a boolean variable, i.e. can be on or off. This message is published by the controller program of the server software and subscribed by the control unit software, so that commands from the controller can reach the control unit of the air purifier. Messages 2-5 describe the indoor air quality, represented by floating point variables. These messages are published by the sensor data acquisition program of the server software and subscribed by the data storage and the controller programs. The sensor unit software can also publish those messages directly, but due to hardware implementation constraints, the sensor data are transmitted to the server through a dedicated TCP link and published there. The publish and subscription flow of the MQTT messages are also depicted in Fig. 7.

Note that in Table I, the quality of service (QoS) level of all messages are set to be 0. This means that the delivery of the messages are carried out with a best effort or at most once. No acknowledgement or re-delivery of the messages will occur between the sender and receiver, implying that messages may get lost when the network traffic is heavy.

2) The sensor data acquisition program: The sensor data acquisition program establishes a TCP server, which accepts connections from the sensor unit software. The program then interprets the incoming sensor measurements and publishes measurement messages under topics 2-5.

3) The data storage program: The data storage program is a simple one which subscribes all messages in Table I and stores them in a database for visualisation and analysis.

4) The controller program: The flow chart of the controller program is shown in Fig. 8.

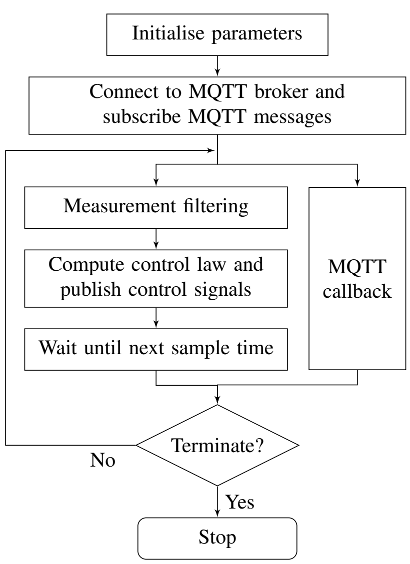


Fig. 8: Flow chart of the controller program of the server software.

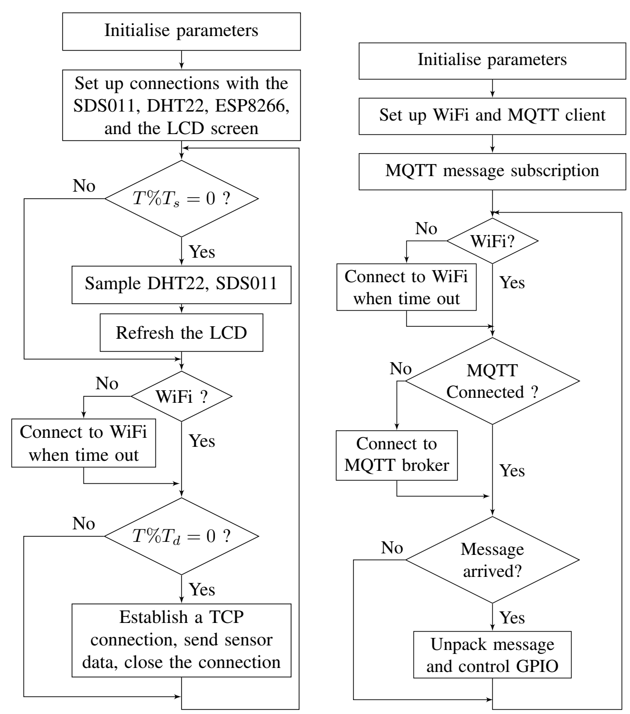
As shown in the chart, the program subscribes measurement messages from the MQTT broker. A measurement filtering block then applies a filtering algorithm to the measurement returned by the MQTT callback procedure. The control law can be any properly designed controllers, and an on-off controller was designed and will be discussed in detail in the next section.

B. The Sensor Unit Software

The sensor unit software lies in the AVR microprocessor on the Arduino board. Its flow chart is in Fig. 9a. As shown in the figure, the program manages connections with all the peripherals, including the sensors, the WiFi module, and the display. The values of the DHT22 and SDS011 sensors are sampled at a fixed interval T_s and displayed on the LCD screen. Data transmission over the WiFi operates at another interval T_d, which is much longer than T_s considering the slow dynamics of the environment parameters.

C. The Control Unit Software

The control unit software lies in the ESP8266 chip of the ESP-12F WiFi module. The code are compiled and programmed directly into the chip's flash memory, so that an additional microprocessor is not required to control the relay. The program utilises an MQTT client library and the flow chart is shown in Fig. 9b. As shown in the figure, the program simply waits for incoming control messages from the MQTT broker and manipulates the GPIO pin connected to the relay accordingly.



(a) Sensor unit (b) Control unit

Fig. 9: Flow charts of the sensor and control units software.

IV. Controller Design and Experiments

A. The Controller Design

1) The Measurement Filter: The measurement filter is a simple finite impulse response (FIR) filter, which takes the average of measurements within a fixed-length backward horizon, i.e. $m_{f,k} = \frac{1}{N}\sum_{i=0}^{N-1} m_{k-i}$, where N is the length of the horizon, k is the current time instant, $m_{f,k}$ is the filtered measurement.

2) The On-Off Controller: Since the internal control logic of the air purifier is not accessible, the control unit hardware described in the previous section is added in between the air purifier and the power supply. The relay within the control unit hardware either fully switches off the air purifier or operates it at a specific power level. In this situation, an on-off controller would be a simple yet ideal option[18].

In on-off control, a set point is determined for the controlled variable, in our case, the PM2.5 concentration. When the measurement exceeds the set point, the relay is switched on, otherwise it stays off. As a standard practice, our on-off controller includes hysteresis, or a deadband around the set point in which the status of the relay is kept without changing. The hysteresis avoids repeated on and off of the relay around the set point by converting the chattering into a low frequency oscillation. This won't be a problem for our air quality control system, since its dynamics is slow, and the control accuracy is not a major concern as long as the PM2.5 concentration is around the setpoint, despite the oscillations introduced. Implementation of the on-off controller is illustrated in Algorithm 1.

Algorithm 1: On-Off Control

Data: Set point m_{SP} and deadband width d of the PM2.5 concentration
Input : Control input of the previous time instant u_{k-1}, and the current filtered measurement of PM2.5 concentration $m_{f,k}$
Output: Control input of the current time instant u_k

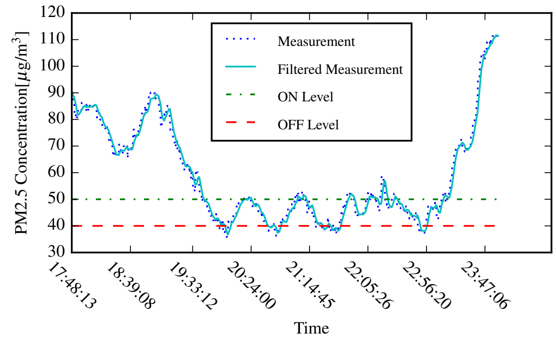
```
Compute the on and off levels of the PM2.5
  concentration, m_ON = m_SP + d/2, m_OFF = m_SP − d/2;
if u_{k−1} == 0 and m_{f,k} > m_ON then
  | u_k = 1;
else if u_{k−1} == 1 and m_{f,k} < m_OFF then
  | u_k = 0;
else
  | u_k = u_{k−1};
end
```

B. Control Experiments

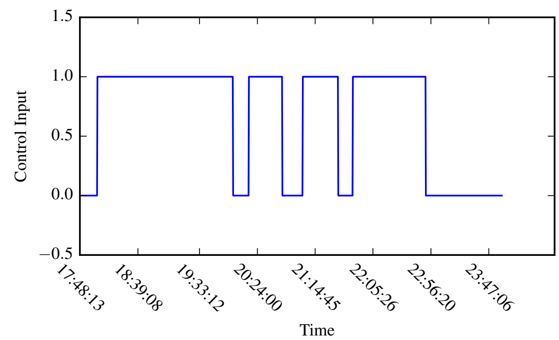
Experiments were carried out in the authors' office environment in Beijing. There were two experiments done, one with a shorter period of time of around 5 hours. The other lasted for about 5 days.

1) The first experiment: The first experiment started when the indoor PM2.5 concentration was around 85 μg/m^3. Fig. 10 shows the measurement of the indoor PM2.5 concentration, together with the filtered values and the control actions of the on-off controller.

In terms of the filter, it can be seen that the filtered measurement as represented by the solid line is less noisy than the raw measurement in the dotted line. In terms of the controller, it can be seen that after the air purifier kicked in under the control action, the indoor PM2.5 concentration dropped effectively. The bump between 18:39 and 19:00 on



(a) PM 2.5 concentration.



(b) Control input.

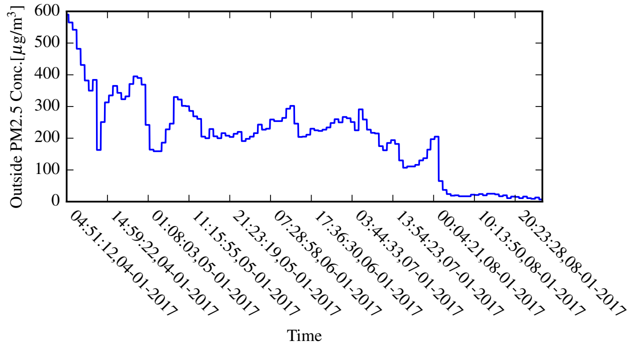
Fig. 10: PM2.5 concentration and the corresponding control input in the first experiment.

the measurement curve was caused by the opening of the door. After the door was closed, the PM2.5 concentration decreased continuously until the OFF level was reached as indicated by the dashed line. The controller then switched off the air purifier accordingly. Roughly 20 minutes after the switch-off of the air purifier, the PM2.5 concentration gradually built up and reached the 50 μg/m^3 ON level. Right after that the controller energised the air purifier.

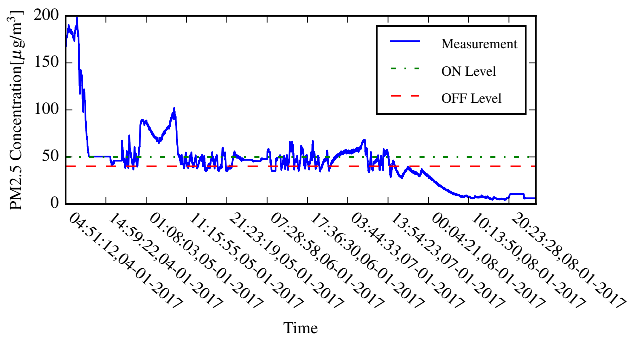
The controller kept turning on and off the air purifier until 23:00, when the controller was scheduled to be off. The PM2.5 concentration in the office then increased rapidly to a high level due to the seriously polluted air outside.

2) The second experiment: The second experiment was continuous operation of the system from 4 January 2017 to 8 January 2017, during which Beijing suffered from a long-duration haze. Since this experiment was carried out at a working environment, the door and the windows were opened and closed irregularly. The air purifier was also scheduled to be off during 11:00 pm and 7:00 am. The PM2.5 concentration outside was recorded and shown in Fig. 11a, together with the experiment results in Fig. 11b and 11c.

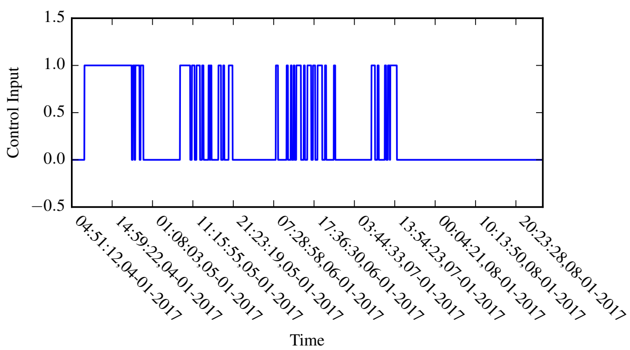
It can be seen that when the PM2.5 concentration outside reached above 400 μg/m^3 and the air purifier was off at night on 3 January, the indoor PM2.5 concentration surged to almost 200 μg/m^3. The level of PM2.5 was effectively brought down when the air purifier resumed operation at 7:00 in the morning on 4 January. The sensor measurement got stuck temporarily and the air purifier was on for most of the



(a) Outside PM 2.5 concentration.



(b) Indoor PM 2.5 concentration.



(c) Control input.

Fig. 11: PM2.5 concentrations and the corresponding control input in the second experiment.

day till late in the afternoon on 4 January. Then the system functioned properly by switching the air purifier on and off and regulating the indoor PM2.5 concentration within the given band around the set point. From 00:00 of 8 January, the PM2.5 concentration outside dropped significantly to almost 0, The measurement indoor also fell below the set point and the controller stayed off during the rest of the experiment. In this experiment, other than the 11:00 pm to 7:00 am interval, there were several points at which the indoor PM2.5 values stayed above 50 μg/m^3 for a period of time, those were the points when the door or the windows were opened.

V. Conclusions and Further Work

This paper discusses an open platform of a WiFi-enabled indoor air quality monitoring and control system. The detailed hardware and software design are presented and demonstrated by experiments in a real-world office environment. The current work focuses on the control of the particulate matters PM2.5 concentration, and can be extended to humidity control as well as networked control over wireless sensor and actuator networks.

References

[1] Anon., "Ambient (outdoor) air quality and health," Sep 2016. World Health Organisation, Fact sheet, http://www.who.int/mediacentre/factsheets/fs313/en/.

[2] Anon., "WHO global urban ambient air pollution database," 2016. World Health Organisation, http://www.who.int/phe/health_topics/outdoorair/databases/cities/en/.

[3] Anon., "Health aspects of air pollution with particulate matter, ozone, and nitrogen dioxide," Tech. Rep. EUR/03/5042688, World Health Organization, Bonn, Germany, Jan 2003.

[4] Anon., "An introduction to indoor air quality." United States Environmental Protection Agency, https://www.epa.gov/indoor-air-quality-iaq/introduction-indoor-air-quality.

[5] R. Harper, "Inside the smart home: Ideas, possibilities and methods," in *Inside the smart home*, pp. 1–13, Springer, 2003.

[6] D. Minoli, K. Sohraby, and B. Occhiogrosso, "IoT considerations, requirements, and architectures for smart buildings - energy optimization and next generation building management systems," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[7] Q. Yang, G. Zhou, W. Qin, B. Zhang, and P. Y. Chiang, "Air-kare: A Wi-Fi based, multi-sensor, real-time indoor air quality monitor," in *2015 IEEE International Wireless Symposium (IWS 2015)*, pp. 1–4, March 2015.

[8] D. Brunelli, I. Minakov, R. Passerone, and M. Rossi, "Smart monitoring for sustainable and energy-efficient buildings: A case study," in *2015 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS) Proceedings*, pp. 186–191, July 2015.

[9] S. M. Saad, A. R. M. Saad, A. M. Y. Kamarudin, A. Zakaria, and A. Y. M. Shakaff, "Indoor air quality monitoring system using wireless sensor network (WSN) with web interface," in *2013 International Conference on Electrical, Electronics and System Engineering (ICEESE)*, pp. 60–64, Dec 2013.

[10] Z. Wu, M. R. Rajamani, J. B. Rawlings, and J. Stoustrup, "Model predictive control of thermal comfort and indoor air quality in livestock stable," in *2007 European Control Conference (ECC)*, pp. 4746–4751, July 2007.

[11] M. del Mar Castilla, J. D. lvarez, J. E. Normey-Rico, F. Rodrguez, and M. Berenguel, "A multivariable nonlinear MPC control strategy for thermal comfort and indoor-air quality," in *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, pp. 7908–7913, Nov 2013.

[12] G. Baroni, C. Carnevale, G. Finzi, E. Pisoni, E. Turrini, and M. Volta, "Uncertainty analysis in air quality control systems," in *52nd IEEE Conference on Decision and Control*, pp. 6792–6797, Dec 2013.

[13] Nova Fitness Co., Ltd., *Laser PM2.5 Sensor Specification, Product model: SDS011*, October 2015. V1.3.

[14] A. Morpurgo, F. Pedersini, and A. Reina, "A low-cost instrument for environmental particulate analysis based on optical scattering," in *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pp. 2646–2650, May 2012.

[15] B. Giechaskiel, M. Maricq, L. Ntziachristos, C. Dardiotis, X. Wang, H. Axmann, A. Bergmann, and W. Schindler, "Review of motor vehicle particulate emissions sampling and measurement: From smoke and filter mass to particle number," *Journal of Aerosol Science*, vol. 67, pp. 48 – 86, 2014.

[16] "MQTT version 3.1.1." Edited by Andrew Banks and Rahul Gupta. 29 October 2014. OASIS Standard. http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html. Latest version: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html.

[17] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutorials*, vol. 17, pp. 2347–2376, Fourthquarter 2015.

[18] T. A. Hughes, *Measurement and Control Basics*. ISA: The Instrumentation, Systems, and Automation Society, October 2006. 4th Edition.