Brezžična senzorska omrežja

Uredila dr. Miha Janež in prof. dr. Nikolaj Zimic

april 2021

Navodila za izdelavo seminarske naloge

- Vaše datoteke se nahajajo v direktorijih `Skupina*`, kjer `*` predstavlja številko vaše skupine - glavna datoteka je `main.tex`.
- Slike shranjujte v svoj direktorij.
- Vse labele začnite z znaki `g*:`, kjer `*` predstavlja številko vaše skupine.
- Pri navajanju virov uporabite datoteko `references.bib`, ki se nahaja v korenskem direktoriju projekta.

V okviru seminarske naloge se boste ukvarjali z aplikacijami na področju brezžičnih senzorskih omrežij. Delo bo potekalo v skupinah z dvema članoma. Za skupinsko delo uporabljajte repozitorij, kot je npr. dropbox[1] ali git[2]. Cilj seminarja so izdelki in poročila, ki jih lahko ob zaključku združimo v celoto. Poročilo pišite v okolju LaTeX, kjer lahko za lažje skupinsko delo uporabljate okolje, kot je npr. overleaf[3]. Za iskanje virov uporabljajte iskalnike znanstvene literature[4]. Rok za izdelavo seminarske naloge je 14. 5. 2021. Predstavitve nalog bodo v zadnjih dveh tednih v semestru. Na predstavitvi seminarja bo imela vsaka skupina 10 minutno predstavitev svojega izdelka, nato bo sledila krajša diskusija.

Za predlogo uporabite strukturo znanstvenega članka, ki obsega poglavja Uvod, Metode, Rezultati, Zaključek in Literatura. Slike naj bodo v formatu PDF ali EPS, z ustreznimi viri (literaturo) polnite vašo BIB datoteko.

Oddana seminarska naloga naj vsebuje:

- izvorne datoteke poročila v LaTeXu (poročilo naj vsebuje približno 10 strani),
- poročilo v formatu pdf,
- prosojnice za predstavitev (predstavitev naj traja približno 10 minut),
- morebitno dodatno gradivo (programska koda itd.).

---

[1] `https://www.dropbox.com`

[2] `https://bitbucket.org`

[3] `https://www.overleaf.com/`

[4] `https://scholar.google.si`, `www.sciencedirect.com`, `https://www.scopus.com`, `https://arxiv.org`, `http://citeseerx.ist.psu.edu`

Kazalo

Poglavje 1
Testing BME680 sensor for air quality

Metodija Bucevski, Jernej Koželj

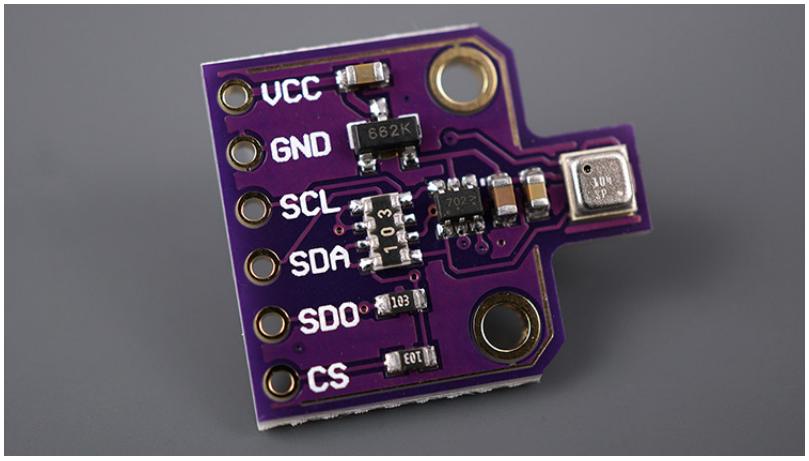
Povzetek The BME680 is a digital sensor for environmental data, that measures air pressure, gases, humidity and temperature. It can communicate with a microcontroller using SPI and I2C communication protocols. One of its specialities is detecting a range of gases, like volatile organic compounds or VOCs in short. Because of this it is widely used as an indoor air quality sensor. We measured air pollution inside with the help of an equation derived from humidity and air pressure. We compared results and informed for the air quality through flashing LED on the board. We came to the conclusion that this is a great tool for personal indoor air quality control.

1.1 Introduction and motivation

An adult breathes around 15,000 litres of air every day. When we breathe polluted air pollutants get into our lungs; they can enter the bloodstream and be carried to our internal organs such as the brain. This can cause severe health problems such as asthma, cardiovascular diseases and even cancer and reduces the quality and number of years of life [1]. So the BME680 is a great sensor for indoor air quality control, because it is a 4-in-1 digital sensor that measures temperature, humidity, air pressure and gases in form of VOCs. It presents as a great tool to inform you about poor air quality which may affect your health in the long run. We used I2C protocol for communication between BME680 sensor and esp8266 chip with the use of bme680 driver. We also demonstrate how to use Wi-FI to transfer data to MQTT server using I2C protocol. When measuring the air quality we turn on LED if air quality is above given threshold using I2C to communicate with PCF8574AN.

1.1.1 BME680

This sensor, shown on figure 1.1, is the first gas sensor that integrates high linearity and high accuracy gas, pressure, humidity and temperature sensors. It is specifically developed for mobile applications and wearables where size and low power consumption are critical requirements. In order to measure air quality for personal well being the gas sensor within the BME680 can detect a broad range of gases as volatile organic compounds VOC.



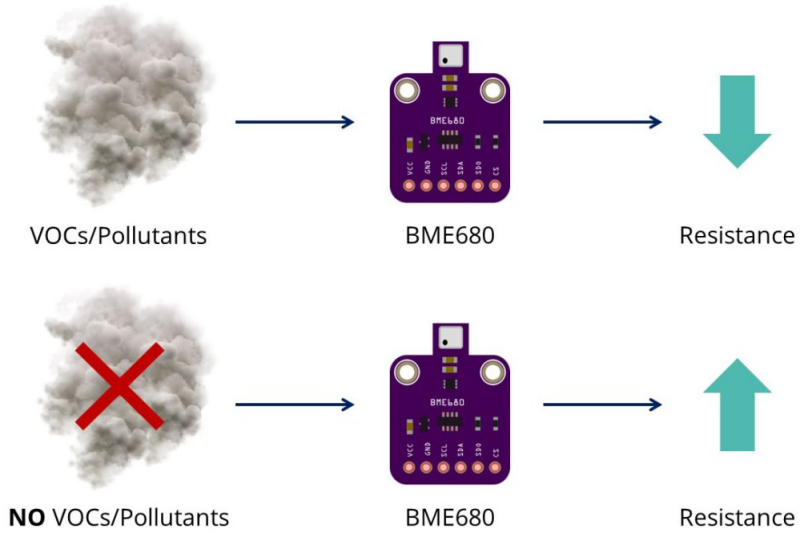
Slika 1.1: BME680 sensor from up close.

Volatile organic compounds (VOC) are organic chemicals that have a high vapour pressure at room temperature. High vapor pressure correlates with a low boiling point, which relates to the number of the sample's molecules in the surrounding air, a trait known as volatility. [2]

1.1.2 Gas sensor

The BME680 contains a MOX (Metal-oxide) sensor that detects VOCs in the air. This sensor gives you a qualitative idea of the sum of VOCs/contaminants in the surrounding air - it is not specific for a specific gas molecule. MOX sensors are composed of a metal-oxide surface, a sensing chip to measure changes in conductivity and a heater. It detects VOCs by adsorption of oxygen molecules on its sensitive layer. The BME680 reacts to most VOCs polluting indoor air (except CO2) [3].

When the sensor comes into contact with the reducing gases, the oxygen molecules react and increase the conductivity across the surface. As a raw signal, the BME680 outputs resistance values. These values change dut to variations in VOC concentrations [3].



Slika 1.2: BME680 resistance with VOCs present.

Higher concentration of VOCs means lower resistance and lower concentration of VOCs means higher resistance, as shown on figure 1.2. The reactions that occur on the sensor surface (thus, the resistance) are influenced by parameters other than VOC concentration like temperature and humidity.

The gas sensor on BME680 gives you qualitative idea of VOCs gasses in the surrounding air. So you can get trends, compare your results and see if the air quality is increasing or decreasing. To get precise measurements, you need to calibrate the sensor against known sources and build a calibration curve.

1.1.3 BME680 properties

1.1.3.1 Accuracy

Tabela 1.1: Accuracy of BME680 sensors.

Sensor	**Accuracy**
Temperature	+/- 1.0℃
Humidity	+/- 3%
Pressure	+/- hPA

1.1.3.2 Operation range

Tabela 1.2: operation range for the temperature, humidity and pressor sensors.

Sensor	**Operation range**
Temperature	-40 to +85℃
Humidity	0 to 100%
Pressure	300 to 1100 hPA

1.1.3.3 Pinout

Tabela 1.3: BME680 pinout

Pin	**Action**
VCC	Powers the sensor
GND	Common GND%
SCL	SCL pin for I2C communication, SCK pin for SPI communication
SDA	SDA pin for I2C communication, SDI (MOSI) pin for SPI communication
SDO	SDO (MISO) pin for SPI communication
CS	Chip select pin for SPI communication

1.2 Libraries and modules used

We used libraries from [4] for the support for the BME680 sensor. We also used MQTT messaging protocol which is useful for low power sensors, but is applicable in many scenarios [5]. Included libraries:

- stdio.h
- stdlib.h
- espressif esp_common.h
- string.h
- ssid _config.h
- espressif esp_sta.h
- espressif esp_wifi.h
- paho_mqtt_c/MQTTESP8266.h
- paho_mqtt_c/MQTTClient.h
- semphr.h
- bme680/bme680.h
- esp/uart.h
- FreeRTOS.h
- task.h
- esp8266.h
- i2c/i2c.h
- queue.h

1.2.1 MQTT

MQTT is an OASIS standard messaging protocol for Internet of Things (IoT). It is designed as extremely lightweight publish/subscribe messaging transport that is ideal for connecting remote devices with a small code footprint and minimal network bandwidth [6].

1.2.1.1 Publish/Subscribe

The MQTT protocol is based on the principle of publishing messages and subscribing to topics, or ”pub/sub”[5]. Multiple clients connect to a broker and subscribe to topics that they are interested in. Clients also connect to the broker and publish messages to topics. Many clients may subscribe to the same topics and do with the information as they please. The broker and MQTT act as a simple, common interface for everything to connect to. This means that you, if you have clients that dump subscribed messages to a database, to Twitter, Cosm or even a simple text file, then it becomes very simple to add new sensors or other data input to a database, Twitter or so on. [5].

1.2.1.2 Topics/Subscriptions

Messages in MQTT are published on topics. There is no need to configure a topic, publishing on it is enough. Topics are treated as hierarchy, using a slash (/) as a separator. This allows sensible arrangement of common themes to be created, much in the same way as a filesystem.

1.3 Solution

1.3.0.1 Code explanation

In makefile in entry EXTRA_COMPONENTS, we add the following drivers: extras/rboot-ota extras/i2c extras/bme680 extras/mbedtls extras/paho_mqtt_c.

1.3.0.2 Main code

Our sensor was at first set to measure without stopping for 24 hours. Those measurements did not have any impact on the final graphs, presented in Experiments and results subsections. But, they are the first step to getting relevant data.
We start by setting up USB connection communication with speed of 115200 baud/sec. Binary semaphore is created called wifi_alive, that assures the board first to connect to Wi-Fi, and then connect and send data to mqtt broker. Data is sent, only after successful Wi-Fi connection is established.

For I2C communication between esp8266 chip and PCF8574AN, pin 14 for SCL signal, and pin 12 for SDA signal will be used. First, we initialize I2C communication on I2C bus number 0, with the pair of pins, previously mentioned. BME680 sensor has address 0x76 (shared with BMP280) and 0x77. This sensor is initialized, on I2C bus number 0, with address 0x77.
Second set of measurements are also futile, but are important step towards getting relevant data. This measuring process last between 30 - 40 minutes.

Next, three tasks are created, one for connecting to local network via Wi-Fi called wifi_task, with priority 1, second is the user task where we get the measurements and out of them we calculate Air Quality Index with task priority 2, and third for sending data to mqtt broker called mqtt_task, with priority 3.

When talking about Air Quality Index, we think about indoor Air Quality Index, which is one of the many applications BME680 sensor is used for. Common way is to assume that relative humidity will have impact on final score with maximum 25% and gas resistance readings with maximum 75%. For hum_baseline we've set to be 40%, optimal humidity value for indoor environments. Anything that we obtain for

humidity, which is in range from 0% - 100%, that is lower or above 40% will impact the final score with less than 25%. Only if the measured humidity is 40%, the impact on the final score will be 25%.

When talking about the measured values for gas resistance, which is the second contributor to the final score, we first have to calibrate the sensor. This means to obtain the value for gas_baseline, which can be unique for each room, and also can vary in the same room, depending on multiple factors. Such factors can range from number of people and animals in the room to number of vehicles outside, if windows are opened. We make 50 gas resistance measurements and calculate their average value which is the gas_baseline.

To examine how gas resistance measurements impact the final score, we will assume that gas_baseline is 100 $k\Omega$ and the value for gas resistance is 50$k\Omega$, 0$k\Omega$, 100 $k\Omega$ and 200$k\Omega$. When gas resistance is 50 $k\Omega$, because it is half the value of gas_baseline it's impact on the final score will be 36.5%. When gas resistance is 0$k\Omega$, the impact on final score will be 0%. Any value for gas resistance that is equal or higher than gas_baseline, third and fourth scenario, will have impact on the final score with 75%.

The final score is in range from 0% to 100%, 0% meaning lowest possible quality of air and 100% meaning the best possible quality of air. For getting Air Quality Index, we subtract this score from 100 and this difference is then multiplied with 5. Air Quality Index, is in range between 0 and 500, when 0 means the best possible quality of air, and 500 means the worst quality of air.

For connecting to the local network, first the user must insert his/hers network SSID and password in /home/bso/(workspace_name)/include file named private_ssid_config.h file with the following lines:
#define WIFI_SSID "insert here mywifissid"
#define WIFI_PASS "insert here my secret password"
As long as we are connected to a network, the semaphore is released, and can be taken by the second task. Second task calculates Air Quality Index (AQI), by using the measurements obtained from the sensor BME680. After calculation of Air Quality Index, we check whether this value is above certain threshold, and if yes, we write to PCF8574AN register 0xfe, meaning that we will turn on LED1, and then after 300ms, we turn it off by writing to PCF8574AN 0xff.

Also, after calculation of AQI, that value is stored in a queue, named published_queue (pointer's name), that can be received in mqtt_task and sent to mqtt broker. Using previously established Wi-Fi connection (with user's home network or using hot spot, i.e. using for ex. mobile phone as access point) we published each Air Quality Index value to mqtt broker – "test.mosquitto.org", on port 1883. In order for the board to publish to mqtt server, we first create its own unique ID, using its MAC address. We connect to the broker and publish the message (if there is) recei-

ved from the publish_queue(pointer's name) with QOS1, meaning we send message at least once, without confirmation required. From command line we read the data send by the board to the MQTT broker, if we subscribe to the topic on which the board sends the data. In order to do that we have to install mosquitto-clients, with command: **mosquito_pub –h sudo apt-get install mosquito-clients** and then subscribe to the topic so that we can read the data with the command: **mosquito_sub –h test.mosquitto.org –t /beat –v**.

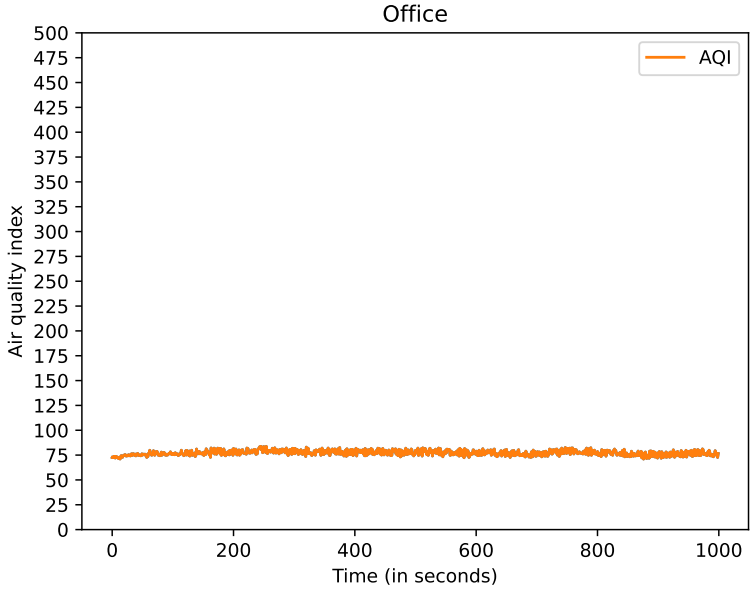
We measured air quality index in three different rooms, i.e. office, bedroom and living room. The idea was to see if the calculated air quality index is different. The lower the value the better as seen in the figure 1.3.

IAQ Index	Air Quality	Impact (long-term exposure)	Suggested action
0 – 50	Excellent	Pure air; best for well-being	No measures needed
51 – 100	Good	No irritation or impact on well-being	No measures needed
101 – 150	Lightly polluted	Reduction of well-being possible	Ventilation suggested
151 – 200	Moderately polluted	More significant irritation possible	Increase ventilation with clean air
201 – 250[9]	Heavily polluted	Exposition might lead to effects like headache depending on type of VOCs	optimize ventilation
251 – 350	Severely polluted	More severe health issue possible if harmful VOC present	Contamination should be identified if level is reached even w/o presence of people; maximize ventilation & reduce attendance
> 351	Extremely polluted	Headaches, additional neurotoxic effects possible	Contamination needs to be identified; avoid presence in room and maximize ventilation

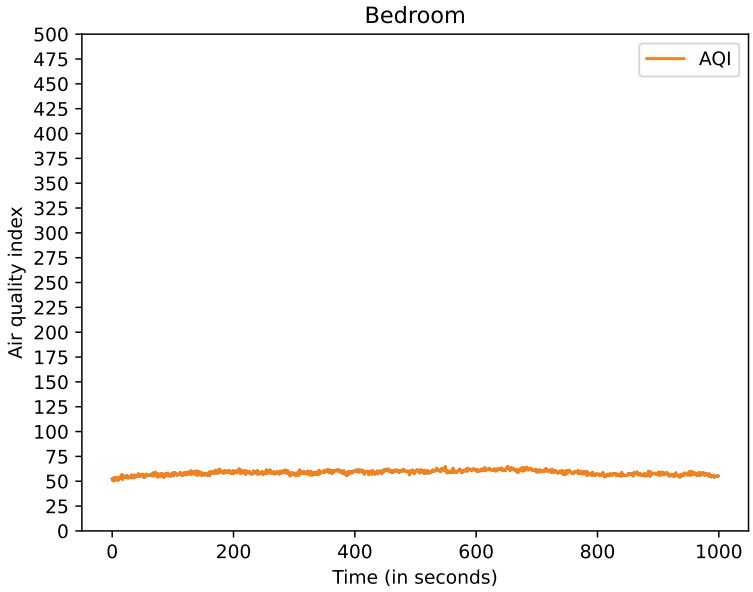
Slika 1.3: BSEC Indoor Air Quality classification

1.4 Experiments and results

In our work we tested the BME680 sensor for air quality, using the I2C protocol for communication between BME680 and esp8266 chip with the use of bme680 driver [4]. Then we used Wi-Fi to transfer data to MQTT server using I2C protocol. We then checked if air quality is above or below the given threshold and we turned on the LED accordingly using I2C to communicate with PCF8574AN. The air quality was the worst in the living room, because that is the place where all the people constantly change and mingle. The second to worst room was the office because we work in that room for the whole day, with not much air ventilation. The bedroom has the window always open, so the air quality was slightly better. The results are presented in the following figures, the measurements were for about 1000 seconds in each room. From the results we can see that the ventilation of the room can result in better air quality, so we must not forget to air the room every couple of hours. We measured the AIQ over the whole night in the office, that is the last graph shown in the figure 1.7. The AQI was around 50 in the bedroom with ventilation, and around 75 in the office. In the living room there was AQI around 80 and when we measured the office during the night the AQI was around 100. We purposely didn't ventilated the office, to get different results. The AQI in Celje on that day was around 43, so the measurements represent pretty good air quality indoors. Everything below 100 is known to be healthy.



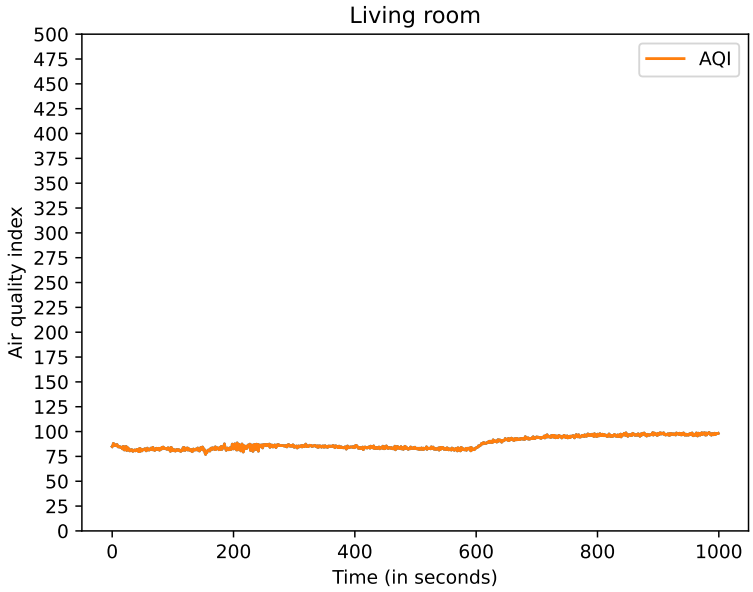
Slika 1.4: Air quality index in the office.



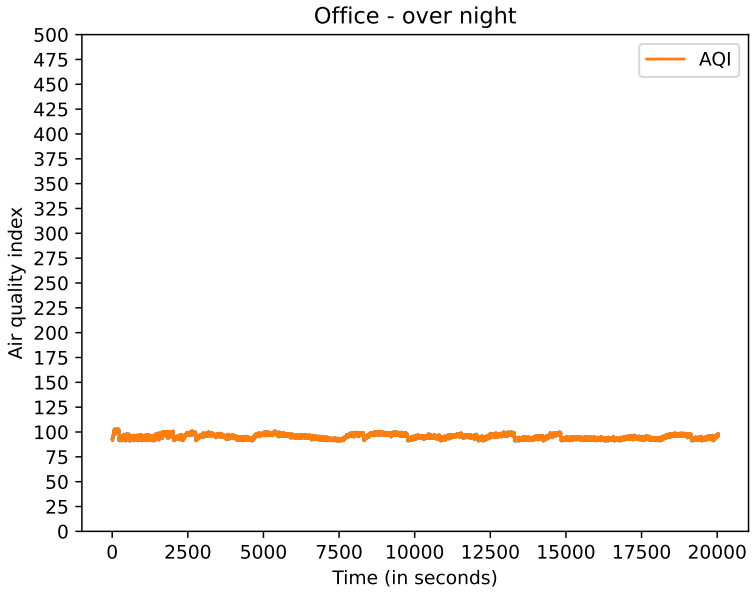
Slika 1.5: Air quality index in the bedroom.

1.5 Problems

During the project, we had several difficulties regarding this work. The first one was with Eclipse IDE, because it was very hard to make it work and actually compile and flash the board with our code.



Slika 1.6: Air quality index in the living room.



Slika 1.7: Air quality index in the office during the night.

1.6 Conclusion

From what we have done in this project and what we read, this type of sensors are a great way to keep you informed about the air quality in the room you are staying. But there also needs to be improvement, because when we read the articles, they pointed out, that these sensors need to be improved, especially if they are used in hospitals and such [7]. Before we can assume the gas measurements from the sensor to be relevant, we have to wait up to 24 hours [3]. When we perform one set of measurements, we wait additionally 30-40 minutes, then after that time we start assuming that the measurements are relevant for our further calculations.

There is a possibility to get better results, if we make the readings for obtaining the gas_baseline, over longer time period. Other possibility is when calculating baseline, we don't do that in the kitchen room, a room attached to the kitchen or room containing a boiler. Moreover, the room should be well-ventilated, windows should be closed if the windows are next to road where there is lots of traffic. Also, the sensor should not be close to heat sources because they affect the humidity in their close environment. Or, if the readings are higher for a certain period of time, we can shift the baseline to a new level. We have to take into account, humans and animals, because they also emit VOCs.
If we have short spikes in the readings, a moving average to smooth out the spikes could be better way for presentation of the air quality, although we lose a precision doing this.

Literatura

1. Web, “Air quality.” https://www.transportenvironment.org/what-we-do/air-quality-and-transport/why-air-quality-so-important.
2. Web, “Vocs.” https://www.epa.gov/indoor-air-quality-iaq/volatile-organic-compounds-impact-indoor-air-quality.
3. Web, “Bme680 guide.” https://randomnerdtutorials.com/bme680-sensor-arduino-gas-temperature-humidity-pressure.
4. Web, “bme680 driver.” https://github.com/SuperHouse/esp-open-rtos/tree/master/extras/bme680.
5. Web, “mqtt messaging protocol.” https://mosquitto.org/man/mqtt-7.html.
6. Web, “mqtt.” https://mqtt.org.
7. C.-C. Jung, P.-C. Wu, C.-H. Tseng, and H.-J. Su, “Indoor air quality varies with ventilation types and working areasin hospitals,” *Building and Environment*, vol. 85, 02 2015.