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 $\star$ predstavlja številko vaše skupine - glavna datoteka je `main.tex`.
- Slike shranjujte v svoj direktorij.
- Vse labele začnite z znaki `g*:`, kjer $\star$ 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

[T1]fontenc makecell

Poglavje 1
Testing BME680 sensor for air quality

Metodija Bucevski, Jernej Koželj

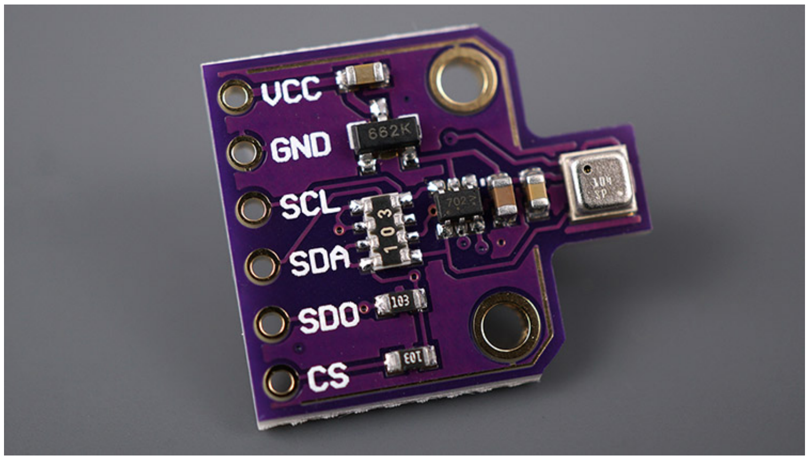
Povzetek The BME680 is a digital sensor for environmental data, that measures air pressure, gases, humidity and temperature. It can communicate with a microcontroller using SPI and I2C communication protocols. One of its specialities is detecting a range of gases, like volatile organic compounds or VOCs 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 nRF24L01 low power near distance communication 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 PCF8574.

1.1.1 BME680

This sensor 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.



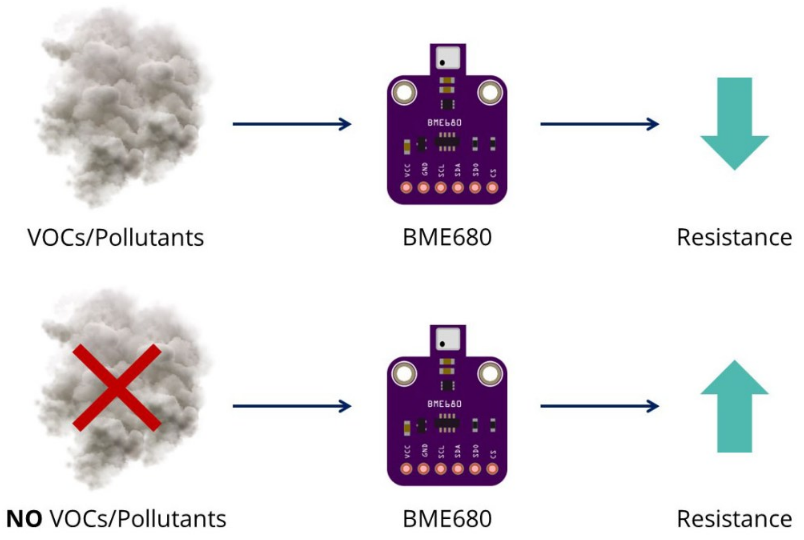
Slika 1.1: BME680 sensor from up close.

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).

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.



Slika 1.2: BME680 resistance with VOCs present.

Higher concentration of VOCs means lower resistance and lower concentration of VOCs means higher resistance. 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	+/- 1.0ºC
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 +85ºC
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 [3] 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 [4]. Included libraries:

- stdio.h
- stdlib.h
- esp/uart.h
- FreeRTOS.h
- bme680/bme680.h
- task.h
- esp8266.h
- timers.h
- espressif esp$_c ommon.hssid_c onfig.h$
- espressif esp$_s ta.hespressifesp_w ifi.h$
- semphr.h
- MQTTESP8266.h
- MQTTClient.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 [5].

1.2.1.1 Publish/Subscribe

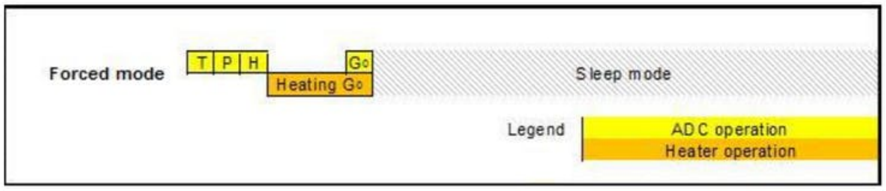
The MQTT protocol is based on the principle of publishing messages and subscribing to topics, or ”pub/sub”[4]. Multiple clients connect to a broker and subscribe to topics that they are interested in. Clients also connect to the broker and publish messages to topics. Many clients may subscribe to the 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. [4].

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.

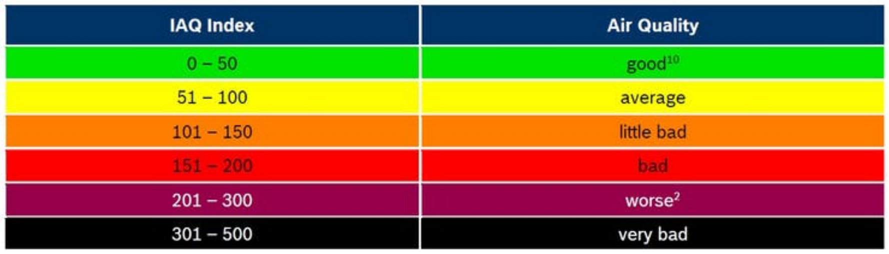
1.3 Air quality measurements

First the readings for temperature, pressure and humidity are taken. Then the hotplate is heated up. After the configured heating time, the gas reading is taken. Then the sensor sleeps for 2.x seconds.



Slika 1.3: Sequence of ADC and gas sensor heater operation.

This cycle takes 3 seconds including sleep time. Advantage for this, is that when taking the temperature reading, the sensor was not heated anymore. The IAQ library from BSEC (Bosch Software Environmental cluster) uses the raw data of the sensors, and spits out an IAQ index.



IAQ Index	Air Quality
0 – 50	good[10]
51 – 100	average
101 – 150	little bad
151 – 200	bad
201 – 300	worse[2]
301 – 500	very bad

Slika 1.4: BSEC Indoor Air Quality classification and color-coding.

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 [3]. Then we used nRF24L01 low power near distance communication 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 PCF8574.

List of global variables:

- I2C$_BUS0I2C_SCL_PIN14$
- I2C$_SDA_PIN13I2C_FREQI2C_FREQ_100K$
- PCF$_ADDRESS0x38led10xfe$
- threshold 60
- static bme680$_sensor_t * sensor = 0; floatgas_baseline;$
- float air$_quality_index; MQTT_HOST(”test.mosquitto.org”)$
- MQTT$_PORT1883MQTT_USERNULL$
- MQTT$_PASSNULLSemaphoreHandle_twifi_alive;$
- QueueHandle$_tpublish_queue; TASK_STACK_DEPTH256;$

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.

1.6 Solution

We start by setting usb connection communication to speed to 115200 baud/sec, initialize i2c communication, and on I2C bus number 0 we register the presence of our BME680 sensor that is addressed with 0x77. We create one-shot timer, in order to calculate gas baseline, which is the average of 50 consecutive measurements of gas resistance, that we obtain from the sensor. After termination of the timer we create three tasks, one for connecting to local network via wifi 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, and third for sending data to mqtt broker called mqtt_task, with priority 4. We also have a semaphore, which at first belongs to the first task. It belongs to it, until we establish successful connection or we don't establish connection after 30 times of trying. As long as we are connected, the semaphore is released, and can be taken by the second task. After calculation of Air Quality Index, we check whether this value is above certain threshold, and if yes, we write to PCF8574 register 0xfe, meaning that we will turn on LED1, and then after 100ms, we turn it off by writing to PCF8574 0xff. Also, using previously established wifi connection (using hot spot, i.e. using our mobile phone as access point) we published each Air Quality Index value to mqtt broker – "test.mosquitto.org". In order for our board to publish to mqtt server, we first create its own unique ID, using its MAC address. We connect to MQTT port 1883. We connect to the broker and subscribe to esptopic with quality of service 1, meaning we send message at least once, without confirmation required. Then, we wait for element to come in the queue, called publish_queue, from user_task, which contains Air Quality Index value, and if there is we publish it to this broker.

1.7 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 [6].

Literatura

1. W. source, “Iaq index.” https://www.transportenvironment.org/what-we-do/air-quality-and-transport/why-air-quality-so-important.
2. W. source, “Iaq index.” https://www.epa.gov/indoor-air-quality-iaq/volatile-organic-compounds-impact-indoor-air-quality.
3. W. source, “Bme680 driver.” https://github.com/SuperHouse/esp-open-rtos/tree/master/extras/bme680.
4. W. source, “Mqtt messaging protocol.” https://mosquitto.org/man/mqtt-7.html.
5. W. source, “Mqtt messaging protocol.” https://mqtt.org//.
6. 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.