International Journal of Environmental Science
http://www.iaras.org/iaras/journals/ijes

Developing reliable air quality monitoring devices with low cost sensors: Method and lessons learned

ELEFTHERIA KATSIRI
Electrical and Computer Engineering
Democritus University of Thrace
Building A, Kimmeria Campus, Xanthi 67100
GREECE

Abstract: Rapid advancements in affordable, miniaturised air pollution sensor technologies offer the potential to capture the high variability of personal exposure to air pollution during daily life with unprecedented spatial and temporal resolution. However, concerns remain regarding the suitability of these novel technologies for scientific and policy purposes due to their lack of reliability. On the other hand, large national networks that allow for the reporting of temporal variations in pollutant concentrations do not provide adequate spatial and temporal coverage and have very high price.

The aim of this work is the development of a common architecture of three types of highly-reliable, portable air quality devices using low-cost sensors. The devices monitor particulate matter, differential pressure and outdoor emissions (CO, CO_2, O_3 and $VOCs$) with high reliability and at a high temporal and spatial resolution while communicating measurements to the cloud in real time. This combination of features makes it possible to answer new questions about the underlying causes of poor air quality, ensure more accurate modelling and prediction at local scales, improve the ability to identify the links between air quality and human health or environmental degradation, identify potential air pollution “hot spots”, and enhance the ability to quantify the impacts of pollutant mitigation techniques.

The methodological approach followed in this paper is to develop at a high TRL level a flexible, modular hardware platform, delay and fault-tolerant middleware components and data-centric cloud services that together ensure reliability in all three places: at the sensor level, the device/edge level and at the cloud. Furthermore, devices are remotely configurable to reduce the maintenance burden from their deployment in real-time scenarios. To validate the usefulness of the devices in an industrial context, eighteen instances were deployed for eighteen months in a modern dairy plant and they were successful in contributing to checking conformance with health&safety guidelines, monitoring negative room pressure and discovering systematic sources of pollution thus protecting the personnel and the products.

Key-Words: low-cost sensors, air quality monitoring, internet of things, raspberry pi, arduino, wireless sensor networks, delay tolerant networking, middleware, publish/subscribe, industry

1 Introduction

Exposure to air pollution is the fifth most important cause of death in the world. According to WHO [1, 2] air pollution causes around 4.3 million deaths per year, while an additional 2.6 million early deaths are caused by indoor air pollution. Fine particles ($PM_{2.5}$) constitute the risk factor associated with one of the highest burdens of disease in the world, with more than 3.2 million deaths and more than 76 million DALYS [3, 4], while significant effects to health are caused by atmospheric NO, NO_2 and O_3.

The main precursor substances in the outdoor air are particulate matter (PM) of sizes up to 10 microns (PM_{10}) and up to 2.5 microns ($PM_{2.5}$), black carbon, carbon monoxide (CO), sulfur dioxide (SO_2) and nitrogen oxides (NO, NO_2). Secondary pollutants are produced in the atmosphere as a result of complex physico-chemical reactions, the most important of which is ozone (O_3) and higher nitrogen oxides (NO_x) together known as photochemical smog.

Recent developments link covid-19 to poor air quality. The authors of [8] found that when the daily concentration of PM2.5 rises above $50ug/m^3$, the number of cases doubles while [7] reports that during long term exposure, even $1ug/m^3$ raise in daily concentration of PM10 leads to an 8% increase in mortality rate with 98% confidence level. The impact of the lock down of the COVID-19 pandemic caused the air quality in many cities across the globe is investigated in [9].

Le Quéré et al. [10] mentioned that although before the Covid-19 pandemic, the emissions of CO_2 were increasing by about 1% in a yearly basis, the day-to-day global CO2 releases, was decreased by -17% (–11 to –25%) by April 2020, taking into account the emissions realised on 2019.

Indoor air is 2 to 5 and in some cases 100 times more polluted than outdoor air. Indoor air quality is influenced both from external air pollutants that penetrate in the building but also from internal sources. Several volatile compounds such as formaldehyde or other *VOCs* are emitted by building materials, paints, furniture, and cleaning products. Carbon monoxide, PM and black carbon can be produced from internal sources, i.e. burning of oil or solid fuels [5], whereas humidity and bad ventilation may produce molds, microorganism growths and release of toxic gases. Under the protective enclosure of the building envelope, pollutants mix and interact or grow (when considering microbes). The above issues make reliable measurement and control of air pollution a priority in the policy making agenda.

Although atmospheric air pollution constitutes an important public health hazard and raises the need for monitoring, there is a dearth of measurements due to the high cost of standard reference instruments. Large national networks allow for the reporting of temporal variations in pollutant concentrations however they do not provide neither adequate spatial coverage nor information on citizens' personal exposure to pollutants, as this can vary significantly in the micro-environments, especially in urban centers where traffic and building central heating remain significant pollution sources [6].

In recent years the development of *low-cost sensor* technology has been a critical factor that changed the pollution detection paradigm allowing for the measurement of air pollution ad hoc and in great spatio-temporal resolution [12], which is required for advanced processing and analytics functions, such as the estimation of aggregated exposure of individuals and their children, especially in underdeveloped areas.

Significant benefits stem from the successful adoption of low-cost sensor technology: outdoors, high spatio-temporal resolution is an enabler for accurate air pollution dispersion and prediction models, that can be exploited to produce novel applications such as pollution free path navigation for cyclists. Indoors, it can help identify systematic sources of pollutants, such as those coming inside when opening windows over busy roads. In the workplace, and especially in industry, high resolution air quality data can help identify not only breaches in health&safety protocols, e.g., heating left switched on or doors between adjacent rooms left ajar, but also identify equipment malfunction, e.g., ventilation not operating. Such information can be invaluable for the design and evaluation of indoor air quality mitigation techniques, proving adherence to legislation, improving citizen health and contributing to epidemiological research in the workplace.

Furthermore, timely response and mitigation of air quality has been proven to incur significant savings, as in the case of *on-demand ventilation* [23] i.e., operating building management systems as much as is necessary to restore air quality to the desirable levels rather than applying fixed ventilation and heating plans that are prone to resource inefficiency. To boot, the Sick Building Syndrome (SBS), a complex situation where people working or residing in a modern building experience several non-specific symptoms (e.g., headaches, fatigue, irritated and itchy eyes, runny nose, sore throats, rashes, etc) as a result of the presence of indoor air pollutants, also has problematic socio-economic consequences, which, in the USA only, are estimated at 10 to 70 billion USD for commercial buildings, while in the Education sector alone it costs the US economy over 6 billion a year, affecting 64 million teachers;

However, the suitability of low-cost measurement devices for reliable air quality sensing devices at a high TRL level is a contentious point:

Issue 1: the exploitation of low-cost sensors requires specialised knowledge about sensing principles, careful design of electronics, calibration in the laboratory and in the field, complex processing, monitoring and adaptive control, issues that hinder their adoption and their potential impact. Hence, only a limited number of works has been able to exploit this technology.

Issue 2: the great degree of heterogeneity (e.g., metal-oxide, electro-chemical, photoionisation, optical, analogue, digital sensors) and lack of standardisation (e.g., UART, SPI, I^2C and $4-20mA$ interface adaptors) makes it hard to create a common architecture that will suit every single low-cost sensing technology. As a consequence, the hardware configurations of the existing systems with real-time fine-grained air pollution information are usually fixed according to the applications (i.e., without any flexible modularity feature). It is very difficult or even impossible to modify or expand their sensing capabilities, or perform system upgrade and evolution.

Issue 3: an infrastructure to host real-time data needs to be provided addressing also the case where sensors operate in hardened environment where there is disrupted or total lack of internet connectivity or other fault or failure. None of the existing systems have dealt with the issue of practical issues in real-life deployment scenarios when thousands of sensor nodes will be deployed in the field. For example, to ensure seamless delay-tolerant operation in critical environments, optimum performance, maintenance on sensor nodes such as calibrating sensors, replacing components, etc., which are time-consuming and labor-intensive, are not negligible. Systems that are easy to maintain can dramatically reduce these costs.

1.1 Aims & Objectives

The aim of this work is to report on the learnings and best practices from developing three novel types of reliable air quality low-cost measurement devices for the workplace: *particulate matter, differential pressure, indoor gases.* It also has the following objectives:

Objective 1: *Reliability enhancement in three levels: sensor, device/edge* and *cloud.*

At the sensor level, the above devices integrate pre-calibrated sensors that have been certified by their individual providers, thus allowing for the measurements produced by the device to be considered *certified.* Here the manufacturers, by means of low-noise electronics and good design of both the sensors and their electrodes, achieve *parts-per-billion* (ppb, 10^{-9}) level sensitivity in laboratory conditions [14]. Furthermore, by calibrating the sensors, both external interference and cross-correlation issues were removed achieving a very small deviation with respect to the reference device [16, 14]. In this work, temperature and humidity sensors have been added to the sensor boards and compensation takes place in the embedded software for the case of CO, O_3, VOC using the provided calibration sheets. As a temperature sensor is already embedded in the PM sensor used in this work, the values reported are affected by the sensor enclosure and are generally higher than the ambient temperature. In this case, a correction is done manually in the software using a reference thermometer and the values are used indicatively. In this work, for analogue sensors, high-accuracy, high gain ADC components have been integrated to provide accurate digitised information at high resolution.

Objective 2: *Flexibiliy at the device platform level by developing two different modular architectures, each accommodating a multitude of sensors for the workplace and industry.*

Novel middleware components were developed that make the devices remotely configurable while leveraging *stream processing* and *store&forward* techniques for the real-time management of the large amounts of sensor data and notifications that are generated. This is particularly important when raw data contains information than can be used to identify transient sources of pollution e.g., a cloud of dust produced by a passing vehicle or a cleaning process in the adjacent production room.

Objective 3: *Scalability at the cloud level by developing an open cloud analytics and device configuration service.*

With sampling rates ranging from $1/10sec$ $(0,1Hz)$ to $1/sec$ $(1Hz)$ generating on average $<$ *25MB* of data per air quality sensing device per month, raw data from different sources needs to be integrated or compared to each other to provide rich context and enable decision making. An open-source *secure ftp server* based on the NetFTP project together with a *python middleware service* was also developed to accommodate cvs data sources of given format to the cloud.

The rest of this paper is structured as follows: Section 2 reports on related work while Section 3 discusses key technologies used as background for this work. Section 4 presents the methodology that was adopted in order to develop the new devices. More specifically, middleware aspects are presented in Section 5 while Section 6 explains how the end devices were tested and validated. Next, Section 7 introduces the applications in the workplace where it makes sense to deploy the devices and Section 8 discsusses selected results from deploying eighteen devices in an industrial scope. Section 9 concludes.

2 Related work

The plausibility of data from networks of low-cost measurement devices is a contentious point; although low cost sensors are commercially available for a large range of air pollutants, they have either been designed to measure concentrations 2 to 3 times higher than the ones that typically exist in the atmosphere and building interiors, or, they suffer in the field from temperature and humidity interference, and baseline drift, which is partly due to the deterioration of the sensing surface [15, 14]. To boot, NO and NO_2 sensors exhibit cross-correlation towards O_3 [15]. One manufacturer demonstrated that by using calibration models [16] both external interference and cross-correlation issues were removed from the *Oz-47* e2V sensor achieving a very small deviation ($< 5\%$) in O_3 measurement with respect to the reference device while a similar approach is mentioned by Mead et al [14] for the next-generation sensors of Alphasense Ltd. The work of [17] showed the possibility of compliance with the data quality objectives (DQO) defined by the European Air Quality Directive (2008/50/EC) for indicative measurements using calibration.

Most of the works reported in the literature have deployed low-cost-sensor devices outdoors, mostly in urban settings where there is significant pollution. Some of these works have used stationary networks (e.g., [24]) while many have investigated mobile ones, i.e., portable devices mounted on buses or trucks. Our work is one of the few that focuses mainly on indoor air and especially in an industrial setting. The work of [25] evaluated vehicle in-cabin air pollutant exposures in a highway and tunnel setting; it showed on average in-cabin exposure to particle number and PM_{10} for the open windows condition was seven times greater when compared to closed windows and air conditioning on. Most mobile air quality monitoring studies to date have been relatively short-term campaigns and provide insufficient repetitive frequency to reveal long-term spatial air quality trends in a city [27]. Apte et al [28] conducted one of the first long-term mobile air quality monitoring studies with a routine fleet of vehicles. Their study used high-quality reference PM air quality monitors on Google Street View cars to repeatedly sample every street in Oakland, CA, over the course of a year.

On the other end of the spectrum of air quality monitors, the use of low-cost monitors (costing less than USD $3000) is increasing [29, 30, 31], and several community-based mobile monitoring studies have used these low-cost instruments [42, 33]. Mahajan and Kumar (2020) [34] have evaluated the use of low-cost sensors for quantifying personal exposure. Anjomshoaa et al [35] compared the utility scheduled vehicles, such as trash-trucks, with non-scheduled vehicles, such as taxis, as urban air quality sensing platforms in cities. The authors of [36] deployed two low-cost optical particle counters on trash-trucks in the city of Cambridge, MA in order to identify $PM_{2.5}$ hotspots.

A method using raw aerosol size distributions from multiple, surface-based low-cost particle counters deployed in Nairobi combined with satellite imaging allows for the derivation of surface aerosol concentrations for particles as small as 0.1 µm in diameter [37]. The work of [39] reports on changes in nitrogen dioxide emissions over Greece due to the lockdown. Unfortunately, this approach is not directly applicable to indoor air quality monitoring.

Previous research has successfully constructed some monitoring systems that provide real-time micro-level air pollution information, flexible data access, and user-friendly data visualization by utilizing the low-cost sensors and WSNs together with mobile web apps. Systems proposed in [40, 41, 42, 43] were mainly focused on the system architecture and implementation. An urban scale wireless networking testbed was proposed and implemented in [41] based on 100 Wi-Fi enabled Linux PCs equipped with a set of low-cost sensors, mounted on the streetlight poles and powered by electrical grid. In [43], a real-time air monitoring system was proposed and implemented by utilizing the stationary low-cost sensors, powered by solar panels, and the General Packet Radio Service (GPRS) wireless link. In [42], a handheld sensor node with six environmental sensors was implemented. This sensor node was able to transmit data tagged with Global Positioning System (GPS) information to user's smartphone through Bluetooth or directly upload the data to a server through GPRS or 802.15.4 radio. In [40], a NoSQL database was chosen for the backend server considering the need for storing the time sensory data at any time from many sources. All of these systems are able to provide fine-grained air pollution information through mobile apps in real-time.

A neural network structure was proposed and implemented in [44] to improve the data quality of the acquired pollution data by considering the relationship between the pollution data and the ambient conditions. The GasMobile [45] system achieved high data accuracy by calibrating the low-cost sensors from time to time with the collocated conventional monitoring stations. Three novel techniques (i.e., a temporal n-gram augmented Bayesian room localization method; an air exchange rate based indoor air quality sensing method; and a zone-based proximity detection method for collaborative sensing) were proposed and implemented to improve the data accuracy and energy efficiency of the Mobile Air Quality Sensing (MAQS)[46] system. The energy consumption of the system proposed in [47] was optimized on sensor level, node level, and network level by performing dynamic gas sampling, people presence sensing, and nodes cooperating, respectively. A Clustering Protocol of Air Sensor (CPAS) network was proposed in [48] to improve the energy efficiency, life-time, and data rate of the network. In [49], in order to correct the impacts of ambient conditions (i.e., temperature and relative humidity) on the electrochemical sensors, a multi-parameter correction algorithm was adopted.

Issues and challenges like privacy problem, user behavior, and data visualization were studied in [50, 51, 52].

Our focus is far less on the performance of any individual module and far more on an overall architecture that supports the prototype, pilot, and production stages of design, and preserves the artifacts and learnings accumulated along the way. Reliability, delay-tolerance, real-time processing and fault-tolerance, coupled with the Internet of Things, are key pillars of our systematic approach. Furthermore, almost all the above works focus on outdoor monitoring, whilst we validate our architecture with real-data in an industrial indoor domain that has strict requirements.

3 Background

3.1 Single-board computers (SBC)

A single-board computer is a complete computer built on a single circuit board, with microprocessor(s), memory, input/output (I/O) and other features required of a functional computer. Single-board computers integrate some of the most popular microprocessors and microcontrollers, e.g., Expressif's esp32, esp8266, ARM Cortex series, Atmel's ATMega series, STMicroelectronics' STM32, to name a few.

3.1.1 Raspberry Pi (RPi)

The *Raspberry pi 4* (RPi4) is a single board computer with key features that include a high-performance 64-bit quad-core processor (Broadcom BCM2711), 4GB of RAM, dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0, Gigabit Ethernet and USB 3.0. One disadvantage is that its more powerful CPU and GPU consume more power and produce more heat especially when required to run for lengthy periods. Although subsequent version of firmware updates claim to reduce the temperature, it has not been possible to apply them during this work, because of backwards compatibility with the middleware libraries which requires a stable OS version. For this reason, the RPi3 was used instead in some cases.

Another disadvantage of the RPi4 is that it is more expensive than previous models (€60 for the bare 4GB version). It also consumes at a minimum 3A current requiring a more powerful adaptor, which supports a USB-C interface, rather than the older micro-USB; Alternatively, a high powered USB port that is capable of providing 3A current will also work. *Secure Digital*, is a proprietary non-volatile memory card format developed by the SD Association for use in portable devices, such as RPI4. SD cards come in three physical sizes and RPi4 uses the smallest one (microSD). Since the OS is stored in an SD card, choosing a high-quality micro-SD card (such as high-quality Panasonic and SanDisk cards) rated at a suitable speed and at least 8GB in size, is vital to ensuring that the RPI4 runs smoothly. High quality SD cards provide resilience to rewrites (to large numbers of which take place during the OS updates and when writing data to an embedded database). In this work the Raspian Buster operating system version was selected as it supported best the middleware libraries and especially Processing 3.

3.1.2 Arduino UNO Rev 3 (ARD)

The *Arduino UNO Rev 3* (ARD) is a popular micro-controller that is based on the ATmega328P and has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. In order to set it up, the ARD was connected using a USB AB cable to the MacBook Pro running the Arduino Studio IDE. Middleware services targeted for the ARD were downloaded and installed via the UART interface.

Grafana [64, 65] has a pluggable data source model and supports many of the most popular time series databases as well as SQL databases like MySQL and Postgres. Each sensing device is associated with a Grafana dashboard that provides time-series graphs of all pollutants, temperature and relative humidity over a period of time; the latter is defined by a widget on the top right of the screen. Visual analytics are possible by means of these dashboards. InfluxDB [66] is a timeseries database [67], i.e, a database optimized for time-stamped data communicated by the sensing devices through the IoT architecture.

3.2 Communication protocols

For the communication between the devices and the cloud, two different paradigms are supported in the literature: *Message-oriented (publish/subscribe) and store-and-forward*. The former is realised through the curl and MQTT protocols, respectively; each single measurement is directly communicated to the cloud, subject only to delays in the end-to-end communication path; Although this leads to shorter response times, MQTT has stronger security as it uses a public key infrastructure between the device (client) and the cloud (server). However, there is no end-to-end reliability so some messages might be lost. *Store-and-forward* methods on the other hand, provide delay and disruption tolerance by first aggregating data locally and then checking whether an internet connection is available in order to proceed by sending the bundles to the cloud. This makes the response time of each measurement is dependent on network availability. The choice of communication paradigm depends therefore on the requirements of the application domain regarding response time and data reliability as well as the requirements of third party IOT cloud services such as AWS and 5G infrastructure where MQTT remains the prevailing supported protocol.

3.3 Interface adaptors

For the communication between the sensors and the microprocessor or microcontroller several communication protocols are used, out of which, the most popular are UART, SPI and I^2C.

3.3.1 Universal Asynchronous Receiver-Transmitter (UART)

UART [53] stands for *Universal Asynchronous Receiver-Transmitter*, and is also commonly known as serial. It sends data bits one by one, from the least significant to the most significant, framed by start and stop bits so that precise timing is handled by the communication channel. The electric signaling levels are handled by a driver circuit external to the UART. Two common signal levels are RS-232 and RS-485. UART has three modes: simplex, which only allows one device to communicate to the other; full duplex, which uses two lines, allowing both nodes to be sending and receiving at the same time; semi- duplex, which uses a single line, with the two nodes switching being sender or receiver. The line has an idle of high, with a low bit signaling a start of a message. Depending on the code set that is being used, there are five to ten bits following this, followed by a high bit that signals the message has ended (Figure 1).

Starting in the 2000s, most IBM PC compatible computers removed their external RS-232 COM ports and used USB ports that can send data faster. For users who still need RS-232 serial ports, external USB-to-UART bridges are now commonly used. They combine the hardware cables and a chip to do the USB and UART conversion. Cypress Semiconductor and FTDI are two of the significant commercial suppliers of these chips [?]. Although RS-232 ports are no longer available to users on the outside of most computers, many internal processors and microprocessors have UARTs built into their chips to give hardware designers the ability to interface with other chips or devices that use RS-232 or RS-485 for communication.

3.3.2 Serial Peripheral Interface (SPI)

SPI ([55], [56], [57], [58], [59], [60]) is a synchronous serial data protocol used by micro-controllers for communicating with one or more peripheral devices quickly over short distances. With an SPI connection the micro-controller is a master device which controls the PM sensor via the following four lines:

- MISO (Master In Slave Out) - The Slave line for sending data to the master
- MOSI (Master Out Slave In) - The Master line for sending data to the peripherals
- SCK (Serial Clock) - The clock pulses which synchronize data transmission generated by the master
- SS (Slave Select) - the pin on each device that the master can use to enable and disable specific devices.

The clock is controlled by the slave, and is used as the control for when data is sent. The slave selects a master using the master select lines, writing it to zero. Then, during every clock cycle, the slave uses the MOSI line to transmit one bit to the master, and the master uses the MISO line to transmit one bit to the slave. When the slave is done with the communication, it will stop using the clock line, and deselect the master (Figure 2).

3.3.3 Inter-Integrated Circuit (I^2C)

I^2C [61] is a serial communication protocol that combines the best features of SPI and UARTs. With I^2C, you can connect multiple slaves to a single master (like SPI) and you can have multiple masters controlling single, or multiple slaves. This is really useful when you want to have more than one micro-controller logging data to a single memory card or displaying text to a single LCD. Like UART communication, I^2C only uses two wires to transmit data between devices:

- SDA (Serial Data) – The line for the master and slave to send and receive data.
- SCL (Serial Clock) – The line that carries the clock signal.

I^2C uses the clock line (SDL) to control the line, while using the data line (SDA) to send data. The devices use a logic zero on SCL to tell the system that SDA has been changed. When SCL holds logic one, SDA should stay constant. This is only violated for the STOP and START signals. During the START signal, the SCL line is at one, and SDA transfers from one to zero. For the STOP signal, the SCL line is at one, and SDA transfers from zero to one (Figure 3). The START and STOP signals are based on the clock signal.

3.4 Printed circuit board design

One of the main difficulties when designing a Printed circuit board (PCB) is that the connections between the components and the microcontroller cannot cross each other over a single layer (surface). This requires a careful design and optimisation of the layout of connections in the available space. The free Fritzing tool offers a validation option whereby connections are redeployed in different layouts, by a brute force algorithm until all overlaps have been eliminated. However, often a valid solution is not possible and either the board needs manual redesigning or some of the components need removing. Happily several ARD pins have redundancy (e.g., copies of *Vcc* and *GND* pins are located in opposite ends of the device GPIO) in order to aid connection routing. Another option is to outsource some connections to the bottom layer of the PCB. Fritzing only supports a two-layer PCB design while other manufacturing tools such as Eagle allow for more layers in between the top and the bottom. The designer also has the option to insert fixed connectors, as proxies for e.g., *Vcc* and *GND*, placed strategically to the components to aid routing.

Through-hole technology [62] is a mounting scheme used for electronic components that involves the use of leads on the components that are inserted into holes drilled in printed circuit boards (PCB) and soldered to pads on the opposite side either by manual assembly (hand placement) or by the use of automated insertion mount machines

Surface-mount technology (SMT) [63] is a method in which the electrical components are mounted directly onto the surface of a printed circuit board.An electrical component mounted in this manner is referred to as a surface-mount device (SMD). In industry, this approach has largely replaced the through-hole technology construction method of fitting components, in large part because SMT allows for increased manufacturing automation which reduces cost and improves quality. It also allows for more components to fit on a given area of substrate.

Flux is a chemical cleaning agent used before and during the soldering process of electronic components onto circuit boards, both in manual hand soldering as well as the different automated processes used by PCB contract manufacturers. The main purpose of the flux is to prepare the metal surfaces for soldering by cleaning and removing any oxides and impurities. Oxides are formed when metal is exposed to air and may prevent the formation of good solder joints. The flux also protects the metal surfaces from re-oxidation during soldering and helps the soldering process by altering the surface tension of the molten solder.

3.5 Wireless Sensor Networks & Middleware

Pressure sensors convert input pressures to electrical outputs to measure pressure, force and airflow. Pressure sensors provide an output based on physical pressure input and they are categorized, in part, by the type of pressure they measure: *gauge, differential, absolute, or vacuum gauge.*

Differential pressure is the difference in pressure between two pressure sources. Differential pressure sensors are designed to simultaneously accept two independent pressure sources, so they have two pressure ports. The output is proportional to the pressure difference between the two sources. When one of these two pressure sources is ambient pressure, this is then called gauge pressure while absolute pressure is measured with respect to a vacuum reference.

Five main programming logic functions are implemented in the middleware aiming to address challenges that arise at the device level:

- *Data acquisition:* This usually involves translating voltages to physical units by applying one or more mathematical functions. The output of this function a *tuple*, a finite, ordered list of raw sensor values.
- *Data calibration:* Transforming measurements from one range to another so that their values make sense. The output of this function is a calibrated tuple where at least some of the original raw values were updated according to the calibration function. This is an optional function.
- *Data merging:* The output of this function is the combination of individual calibrated tuples to form a a complex tuple type such as a list and merging of lists to form a *Matrix*, i.e., a table or a two-dimensional array-like structure in which each column contains values of one pollutant and each row contains one set of values from each pollutant.
- *Data streaming:* The management, storing and mining of potentially infinite volumes of

continuous data at variable rates and their immediate processing so that temporary storage of data at the device level does not fill up the file-system. The output of this function is a sliding window of fixed size on the previous Matrix.

- *Data communication:* Methods for communicating instances of Matrix data streams from the device to the cloud over one or more communication media; this includes store-and-forward methods, messaging and publish/subscribe methods.
- *Data high-availability:* refers to Methods for device management such as scheduled and nonscheduled reboots and remote control.

4 Methods/Development

Having reviewed related work (Section 2) and presented the background (Section 3), Section 4 discusses new work that was carried out in order to develop the three novel device types. The overall IoT system architecture is shown in Figure 4. It consists of *hardware, embedded software* and *cloud services* while it allows for a secure, delay-tolerant communication channel between the devices and the platforms as well as a scalable data storage, aggregation and visualization. Last, to ensure greater flexibility in integration of third-party systems, elements such as application programming interfaces (APIs) are provided.

4.1 Reliable air quality sensing device architecture

Three types of reliable air quality measurement devices, *Particulate Matter (PM)*, *Differential Pressure (DP)*, *Air quality Outdoor Gases (AQG)* were developed as shown in Table 1 where limits indoors are also mentioned where they exist. The devices can either be deployed in a network structure providing a holistic solution where this is desirable, for example to comply with workspace regulation or ISO standards. They can operate in either an *online* mode, sending measurements to the cloud for visualisation, alerts and further analytics as shown in Figure 4 or in an *autonomous* mode, storing data locally to a database where they can be retrieved through a USB interface using a flash drive. All devices are suitable for both indoor and outdoor use. They support wireless communication via WiFi and Bluetooth 4.0 as well as Ethernet. They are powered by mains and are also portable by integrating a power-bank which lasts for approximately 24 hours before requiring re-charging. They can be accessed and controlled remotely through an embedded TeamViewer micro-server. Temperature and humidity sensors are included in all devices and the specifications are shown in Table 2 and Table 3.

Table 1: Device air quality sensor specification

Device	Sensor	Range	limit
PM	PM1	0.38 to 1 ug/m^3	na
PM	PM2.5	0.38 to 2.5 ug/m^3	25 ug/m^3 [1]
PM	PM10	0.38 to 40 ug/m^3	40 ug/m^3 [2]
DP	DP.	-125 to 125 Pa	n/a
AQG	CO	1 ppb to 50 ppm	na
AQG	CO_2	0 ppm to 5000 ppm	1000 ppm
AQG	O_3	0 ppb to 20 ppm	na
AQG	VOC	1 ppb to 50 ppm	na
AQG	NO_2	0 ppb to 20 ppm	na

Table 2: Temperature sensor specification

Device Name	Range (T°C)	Accuracy
TH	-40°C to 125°C	1°C
PM	-10°C to 50°C	2°C
DP	-40°C to +85°C	3°C
AQG	-30°C to 50°C	0.3 °C[3] to 2°C

Table 3: Humidity sensor specification

Device Name	Range (RH%)
TH	10 to 90%
PM	0 to 95%
DP	40°C dew point

The overall system comprises two types of reliable sensing device architectures:

- Store-and-Forward (S&F) Architecture
- Publish/Subscribe (P/S) Architecture

The P/S architecture (Figure 6) is suitable for outdoor gas sensors, such as those found in urban (NO, NO_2,O_3, VOC, industrial, (SO_2, HC_4) and agricultural environments (NH_3). It is a plug-and-play architecture where existing sensors can be replaced by other electro-chemical or metal-oxide, three electrode sensors with minor adaptation in the software. The S&F architecture (Figure 5), on the other hand, is more generic and can accommodate a more diverse set of sensors, i.e., differential pressure, particulate matter, hydrogen peroxide (H_2O_2). It is an extensible architecture that can accommodate any sensor by modifying only the data collection and data calibration software functions and creating a custom sensor board that integrates the electronic components that drive the sensors as well as a connection interface to each sensor.

To set up the RPi4, a Mac Book Pro (late 2015) was used running Mac OS Yosemite. A Raspian Buster image was first downloaded from the Raspberry Pi foundation and was next downloaded onto the SD card using the Balena Etcher application. Using a configuration script, parameters that enable the appropriate communication interfaces (I2C, SPI) were set, middleware services and their software dependencies were initialised, and WiFi passwords on all known networks were added. This configuration method was preferred to making a bootable clone of a working installation with the required software pre-installed as it was not successful. One possible reason for this was the fact that OS partitions expand in the SD card so cloning them even for identical devices is not straightforward. Once RPi4 is up and running it is added to the administrator's TeamViewer account and it is secured by a password.

An open cloud analytics service was developed based on the Grafana visualisation platform. Underpining grafana cloud is an InfluxDB timeseries database. Furthermore, an open-source *secure ftp server* based on the NetFTP project together with a *python middleware service* were also developed. The service, that runs once every hour, first converts csv files to the correct *line-protocol* format that is required by InfluxDB and then imports the converted file into the database so that the data appears in Grafana and can be easily visualised and analysed.

4.2 Store&Forward architecture

The micro-processor is an RPi4 with an external 32GB flash memory card where a mySQL database is installed along with Rasbian Buster. The microprocessor is the Arduino Uno Rev 3. Custom sensor boards were developed for each of the supported sensors in the form of Arduino shields and they were integrated with the ARD using the GPIO interface, i.e., similarly to the RPi4 *hat* architecture. These boards integrate the electronic components that drive the sensors as well as a connection interface to each sensor.

4.3 The PM Sensing Device

The PM sensing device measures particulate matter by means of the OPC-N3 optical particle counter. It detects the scattering light from individual particles that are floating in an air current when they are hit by a laser beam. Based on the light scattering measurement the size of each particle is calculated (which depends on the amplitude of the scattered light and is based on the Mie theory) as well as their concentration.

The interface between the PM sensor and the micro-controller (RPi4) is based on the *Serial Peripheral Interface (SPI)* protocol. Initialisation, sleep intervals, laser intensity and sampling rate are configurable via HEX commands while the specific values depend among others on the actual concentrations in the environment.

As one of the PM sensor pins operates on $3.3V$ while the ARD produces $5V$, a *voltage divider* circuit was constructed using two resistors (R_1, R_2) connected in series, with the input voltage applied across the resistor pair and the output voltage emerging from the connection between them according to Equation 1.

$$V_{out} = R_2/(R_1 + R_2) * V_{in} \tag{1}$$

Both the SPI interface and the voltage divider were implemented as as a *PCB sensor board* (Figure 7).

4.3.1 The PM sensor board

Both SMT and TH technologies were investigated. In order to make the board as robust by manual assembly)and not caring yet at this point to its being produced in mass), TH technology components were selected for the design. Although Fritzing offers a variety of generic component libraries, specialised components such as the sensors used in this work do not exist in the library and had to be created. Creating a new component is not trivial as it requires editing a series of files including a 3D-representation of the component. In order to avoid this process, and since the PM sensor was to be connected to the board indirectly, via a cable that would allow it to be placed closer to the external air flow, the (cable) pico-molex cable-to-board receptor was selected instead. However, as this was only available as an SMD component in the library and could not be used, a screw-terminal was chosen instead.

Once the board was received by the manufacturers it was assembled. Each component was soldered on the board using a high quality soldering iron with a $0.64mm$ nose and high-quality flux. The flux was applied according to the selective soldering method, using precise drop jet process and was instrumental in making the connections.

The loose ends of the molex sensor cable were inserted in the terminal holes and secured with screws. This ensured that the connections were resilient; once the board had been tested using the process explained in Section 6.

4.4 The DP Sensing Device

The DP sensing device developed in this work, aims to provide a solution for all three above applications. It comprises the SDP800 highly-sensitive CMOS MEMs differential pressure sensor with a range from $-125Pa$ to $125Pa$. The sensor is fully calibrated and temperature compensated offering highest accuracy and resolution, especially at very low differential pressures sensors.

The sensor uses a thermal mass flow sensing element based on a calorimetric measurement principle. This sensing element is composed of two temperature sensors and a small heating element. The difference between the two temperature sensors correlates with the mass flow passing the chip. A differential pressure across the sensor ports induces a tiny gas flow through the sensor, which is measured by the sensor element. The mass flow sensing element is integrated on a tiny CMOS chip allowing the signal to be amplified and digitized on chip for best performance.

The communication between the DP sensor and the micro-processor is implemented as a I^2C connection.

4.5 The DP sensor board

Similarly to the PM sensing device, a through-hole printed circuit board was designed for driving the DP sensor (Figure 8) and implementing the I_2C connection with the ARD. As with the PM sensor, the DP sensor is linked to the sensor board with a custom cable that is available by the sensor provider. The sensor end of the cable terminates in a molex adaptor whereas the other end is connected to the screw terminal on the sensor board. This allows the sensor to be placed closer to the enclosure so that it is exposed as much as possible to the air-flow. The DP sensor is first inserted on a plastic adapter which has holes for screwing it on a 3D-printed surface that is fastened inside the enclosure while it leaves an opening for the cable. The DP sensor also has two input ports that together measure the difference in pressure between two points. The board also comprises the HIH-4000 accurate analogue temperature and relative humidity sensor. The HIH-4000 sensor is powered by $5V$ while the DP sensor is powered with $3.3V$.

The board was manufactured and tested using the testing process described in Section 6 that also includes the compensation of differential pressure drop on the hose. As a result five instances of the DP device were developed.

4.6 Publish/Subscribe Architecture

The micro-processor is an RPi3B with an external 32GB flash memory card where a mySQL database is installed along with Rasbian light. The Digital Front End (DFE) is a SouthCoastScience board in the form of a RPi3B shield; It connects to the RPi3B via the GPIO interface, abiding by the *hat* architecture. It includes an ADC, amplifiers, a sensitive digital temperature sensor for providing compensation for the effect of external temperature as electro-chemical sensors come calibrated at 25C. AFE stands for the 810-0020-00 4-Way Analog Front End; it includes amplifiers and electronics that condition the sensor signal as well as a PT100 temperature sensor. Next, four four-electrode MOS sensors are mounted on the AFE while an NDIR CO_2 sensor is connected via serial interface directly to the RPi3B; Last, the Adafruit Pi Foundation 7" Touchscreen Display for Raspberry Pi was mounted on the RPi3B to provide a graphical interface for the device as well as a user interface for controlling it.

4.7 The AQG Sensing Device

The AQG Sensing Device embodies this architecture using sensors for CO, CO_2, O_x and VOC, air pollutants that are found mostly in outdoor air. What is special in this device, with respect to the PM and DP ones, is its a multi-sensor nature, integrating 2 MOS, one PID, and one Non-dispersive Infrared (NDIR) sensor in one architecture. The sensors were elected for their appropriate working range and sensitivity:

CO A4 Carbon Monoxide 4-Electrode Sensor (for concentrations between 10ppb and 2000 ppm), Ox A4 Ozone 4-Electrode Sensor (for concentrations between $10ppb$ and $50ppm$), PID-AH2 VOC Sensor (for concentrations between $1ppb$ to $50ppm$ (Isobutylene)) and an IRC-A1 Carbon Dioxide Sensor (for concentrations between 0 to $5,000ppm$).

The CO, O_x and VOC sensors are mounted on a 810-0020-00 4-Way Analog Front End, which allows the power source and signal connections to be easily achieved. Furthermore, the AFE ensures low-noise electronics and achieves the temperature stability of the sensors -by means of a simple heater drive bridge circuit - whose performance is strongly dependent on the operating temperature.

The analogue signal is processed by the DFE, an electronic interface between electrochemical sensors and the RPi3B.It rejects exactly the upper frequency noise that typically disrupts attempts to read gas MOS sensor. This digital signal was then transferred to the RPi3B, via the GPIO interface.

The CO_2 NDIR sensor is mounted on a separate transmission board that offers both a $4-20mA$ and USB outputs. The USB output is connected to the USB port of RPi3B that also powers the NDIR sensor. However, the total current that was drawn by the DFE-AFE-sensor assembly and the NDIR sensor and transmission board, was too large for the operation of the RPi3B, that as a result showed a yellow lightning icon on the screen. Furthermore, the NDIR sensor occasionally reported too high values especially at start up.

5 Middleware

Efficient middleware components are tailored to the hardware architecture on which they run. For this reason the generic middleware functions (Section 3) were specialised per architecture and component and were structured as a service-oriented architecture.

5.1 Store-and-forward (Line Protocol)

The store-and-forward middleware operates on sensor data streams, structured as csv files, formatted according the Line Protocol. The csv files contain a persistent circular buffer of 4 to 8 data points that is continually updated according to the sensor sampling rate. The size of the window was chosen empirically in order to be near-real-time while representing either half a minute long or one minute long PM data time-series. The communication of the sensor data files to the the cloud, is structured as a service (using Telegraph). *systemd* is an init system and system manager that has widely become the new standard for Linux distributions and was utilised in this work while the *systemctl* command, is the central management tool for controlling the init system. Furthermore, auxiliary services for creating and maintaining the Line Protocol circular buffer of data are scheduled periodically as cron jobs in order to simulate the stream processing behaviour. With respect to data stream processing, no external software was used; rather, streams were programmed manually in Processing 3.0 [69] (java), where they are created and Python 3, where they are consumed. The circular buffer (sliding window) is modelled using the *java.util.Table* class (Processing 3.0) and the *pandas.Dataframe* two-dimensional, size-mutable, potentially heterogeneous tabular data data-structure (Python 3.0).

Table 4: Mapping of functions to S&F services

Function	Service
data acquisition	0
data merging	1
data streaming (and calibration)	2
data communication	3
data high-availability	4 to 6

The mapping of the generic functions to to specialised middleware services is shown in Table 4.

Service 0 runs continually on the micro-controller: it is an embedded C script that reads raw data-points from the sensor as these are generated. After pre-processing, it writes them to the serial output port.

Service 1 runs continually on the micro-processor: it is written in the Processing programming language and it collects the sensor data from the serial input port. Programs written in Processing have by default an implicit event() and drawing() loop, which continuously calls the program's *draw()* function. This has been modified as follows: Thread one (*event()*) continually listens to the micro-controller serial port in order to collect each new data-point. Next it checks its integrity by counting the number of bytes and compares it with the specs and then converts timestamps from millisecond accuracy to nanosecond accuracy before it adds it to a sliding window data-structure. Thread 2 (draw) periodically writes the stream from the sliding window to the micro-processor filesystem for persistence.

Service 2 runs continually on the micro-processor. It is is a python script that continually tails the stream on disk and after it converts it to the line protocol (required by InfluxDB) it writes it to a new file. This file is overwritten at each pass.

Service 3 runs continually on the micro-processor as a modified Telegraph service. It first reads the previous file from disk and next checks whether a connection to the cloud service is available, whereby it inserts the stream in a remote InfluxDB instance using the API. Otherwise, it caches the stream for up to 12 hours in memory and retries periodically. This process is configurable through the telegraph configuration file.

Service 4 runs continually on the micro-

Table 5: Mapping of functions to P/S services

Function	Service
data acquisition	0
data calibration	0
data merging	0
data communication	0
data high-availability	1 to 2

processor and it is an system script that coordinates Services 1 to 3 through the cron scheduling system.

Service 5 runs continually on the micro-processor and it is an system script that periodically restarts Service 3, (every 12 hours) and also restarts the device once in 24 hours, to prevent it from becoming stuck, either for lack of connectivity or if the flash card gets full.

Service 6 runs continually on the micro-processor and it is a remote-desktop type of service based on an embedded TeamViewer server.

5.2 Publish-subscribe (MQTT)

The publish-subscribe middleware is service-oriented and it operates on a data stream window of size: one vector of 5 (concurrent) data points. With respect to the service oriented architecture, a Linux-based operating system is required, and services are implemented as cron jobs. They invoke either Python 3.0 scripts that each runs in an indefinite loop or other services. Single data-points are communicated to the cloud as *events* using the MQTT (publish/subscribe) protocol and AWS cloud API. At the same time the GMAIL API is used to dispatch notifications of values that have exceeded a specified threshold. An SQLite embedded database is used to store measurements persistently. The mapping of the generic functions to to specialised middleware services is shown in Table 5. All services run on the micro-processor.

Service 0 is a Python script running continually, reading raw data from the sensors, compensating noise from the electrodes at zero value measured and aggregating the data in a JSON string that get's MQTT'ed to the cloud platform:

$\{"rec" : "2018 - 08 - 31T14 : 30 : 29.668 + 03 : 00", "val" :$
$\{"Ox" : \{"weV" : 0.605572, "aeV" : 0.554571, "weC" : null, "cnc" : null\},$
$"CO" : \{"weV" : 0.095376, "aeV" : 0.099502, "weC" : null, "cnc" : null\},$
$"VOC" : \{"weV" : 0.110252, "weC" : null, "cnc" : null\}$
$\}\}$

Table 6: Mapping of functions to PlatformIO

Function	Service number
data acquisition	0
data communication	0
data high-availability	-

where (weV) stands for the working electrode and (aeV) stands for the auxiliary electrode values of each of the 4-electrode MOS sensors, discussed later in the paper. The service also integrates an MQTT client to send the data-point vector to the AWS cloud while also logging it in the SQLite database. Last, it calls the GMAIL API to send alerts whenever a predefined threshold is exceeded and to notify the user of canceling these alerts when the values are restored below the threshold.

Service 1 is a scheduled system script that periodically restarts the device once in 24 hours, to prevent it from becoming stuck, either for lack of connectivity or if the SD card gets full.

Service 2 is a remote monitoring service based on TeamViewer that allows remote control of the device. For reasons of security only a remote screen is allowed so that if an external monitor is plugged on the RPi4, it will not work.

5.3 Messaging (HTTP / PlatformIO)

A specialised version of the P/S middleware that is appropriate for low-power embedded platforms that do not have an operating system, such as the popular ESP 8266 and newer ESP3210 micro-controllers, is *Messaging*; it uses the HTTP protocol to post data-points to the InfluxDB cloud service. As a trade-off it does not offer any support for data high availability as shown in Table 6.

PlatformIO is a cross-platform, cross-architecture, multiple framework, professional tool for embedded systems engineers. In this work, a native PlatformIO IDE extension for Microsoft VSCode editor was used to program the middleware component, that connects PM sensor to a WiFi network and streams data to an InfluxDB database using the provided credentials. The wifi and InfluxDB credentials are read from an embedded configuration file.

Service 0 runs on an embedded micro-controller and is implemented as an Arduino sketch that consist of a *setup()* and *loop()* functions. During the *setup()* function PM sensor and the serial and WiFi interfaces are initialised while during the *loop()* phase two things happen:

- Sensors are polled
- Sensor data are reported

As with the case of the P/S middleware, first a vector (message payload) of four values $(CO2, PM_{2.5}, Temp, RH\%)$ is constructed and then, it is posted using the HTTP protocol to the cloud where it is inserted into InfluxDB using the API (Figure 9).

6 Testing & maintenance

A four stage testing plan was devised and carried in order to test the new devices. It comprises four phases, namely: *development, staging, production (rollout)* and *maintenance.*

During the development phase, the sensors were tested individually, on-site, performing the following tests:

1. Testing the hardware architecture
2. Testing the end-to-end connectivity (device-to-cloud)
3. Testing the effectiveness of detection in the presense of pollutants

while during the Staging phase, in-network calibration took place, i.e., making sure that different devices measured air pollution in a consistent way, allowing for comparability.

6.1 Development

The first two test cases are explained in detail Section 4 for all device types. For the third test case, the work is analysed per type of device below.

6.1.1 PM sensor

Sensor-to-breadboard subsystem. The PM sensor was connected to a breadboard by means of a molex cable that on the sensor end had a molex receptor and at the breadboard end, was cut at one end to reveal the naked wires, which were in turn inserted in the breadboard.

Next, the ARD was connected both to the breadboard via its GPIO (and jumper wires) as well as to a Mac Book Pro running the Arduino Studio IDE so that the sensor output could be were observed through the IDE's serial monitor; Because the wires were much thinner than the breadboard holes, soldering was applied in the breadboard holes using the solder tip to secure each wire in place. Although this process caused the plastic around the holes to melt thus creating an "untidy" effect, it was however necessary: the sensor stopped communicating its output if any of the wired connections were even a little loose. The following behaviour was observed:

- If the *Vcc* pin was disconnected, the sensor observed output comprised only null values and was automatically restored when the connection was restored.
- If the *GND* pin was disconnected, the sensor observed output comprised of in-valid values such as the 255 value; it too was automatically restored when the connection was restored.
- Failure of one of the SPI-bus GPIO pins caused the sensor to stop working all permanently. The only way to reset it was by connecting it to the proprietary vendor software interface for this sensor running on a Windows 10 PC, using also the vendor-provided USB adapter. This caused the PM sensor to be reset and re-initialised so that it could be re-connected to the experimental board and continue operating.

PM Sensor board. Solder joints on assembled board was tested using a multi-meter. Tiny extensions were placed at the edge of the *Vcc* and *GND* electrodes in order to make contact with the solder joins. The following systematic tests were carried out.

- Printed connections were tested by measuring voltage between the two end of the connection on the board.
- Voltage drop on the resistors was measured by connecting one electrode of the multimeter (configured as a voltmeter) to the ARD header pin on one end of the resistor and the other to the opposite end of the same resistor.
- Soldered-on headers were tested by measuring contact between a) the metallic ring around the board hole where each header pin was inserted and the inside of the respective header hole, b) the metallic ring and the header pin. For this test the multi-meter was configured to beep when both electrodes measured the same voltage.
- Last, the solder joints of the pico-molex receptor was tested by the above method.

End-to-end connectivity. The PCB was first mounted on the ARD. Once the breadboard-to-ARD connection was validated, the system was connected to the RPi4 via UART to test whether the sensor data came through to the serial interface as explained earlier. Next, data collection (*S&F Service 1*), data stream processing (*S&F Service 2*) and communication to the cloud were tested (*S&F Service 3*).

Despite the successful testing, in some cases the serial connection between the RPi4 and the ARD seemed to hang, as if it were out of sync and only some of the 24 histogram bins that form part of the PM sensor output were actually produced. This was corrected by placing additional code in the draw() function in Service 1 running on RPI4 in order to checking the number of bytes that were submitted every second. If these where less than the expected size, the data-point was ignored.

Effectiveness of pollutant detection. The PM device was tested among others, in the proximity of a kitchen while a steak was being cooked. The result showed very quick response time in picking up the activity, very high concentrations of particulate matter and a long post-cooking poor air quality. As a consequence, the temperature within the enclosure rises by 3-4 degrees °C during operation, affecting the temperature sensor that is included in the device. In one of the devices, a fan was used to lower the temperature by a few degrees °C while the temperature sensor was placed closer to the outside air, however it was not possible to full isolate it thermally. This was compared to having the lid of the enclosure off and comparing the temperature values (Figure 10).

6.1.2 DP sensor

For the case of the DP sensor, the tests for the hardware architecture were similar to those presented for the PM Sensor.

Effectiveness of differential pressure detection. According to Hagen-Poiseuille law, a hose acts as a linear flow restrictor for air flowing through the hose, inevitably generating a pressure drop between the hose inlet and outlet (Figure 12) For long hoses (> 1 m) the pressure drop of the hose might not be negligible any more and needs to be compensated for an accurate differential pressure measurement using Formula 6.1.2.

$$dp = 2 * dp_{hose} + dp_{sensor} \tag{2}$$

while the pressure drop Δp of non-turbulent air or gas in a long narrow hose with circular cross section, is:

1. proportional to the Length L and flow m (Figure 13.)
2. inversely proportional to the 4th power of the diameter D.

A tube of diameter $6mm$ was procured from which pairs of pieces of two, three and five meters long respectively, were cut to be used as air hoses at different distances from the device. The particular diameter was chosen after testing with $2mm, 8mm$ and $10mm$ diameters and different degrees of tube hardness; flexible tubes not only did not fit securely on the sensor but also they were easily bent into sharp angles, preventing air from going through. On the other hand, rigid tubes were not easily handled through air vents of the enclosure or at different angles of the sensor inputs. A medium flexible PVC tube that was used as accessory for aquarium pumps was selected that had an airtight fit with the input port of the sensor. However, this involved first warming up with a lighter the end of the tube to take the shape of the connector. A version of the sensor with tube connectors rather than manifold connectors was chosen.

The pressure compensation formula was implemented in the implementation of Service 0 for the DP Sensor with $3m$ and $5m$ long hose, respectively. To test the compensation, first, all three devices were placed in the same location next to each other and the measurements were compared. All values were very close to one another, which indicated that the drop along the hose was negligible. Next, the two devices with hoses longer than one meter were tested first without the hose on and then with the hose on without compensation. The measured values were practically identical. This led to the conclusion that the drop was negligible, possibly due to the large diameter of the hose ($6mm$).

6.1.3 CO and CO_2 sensor

Effectiveness of pollutant detection. To test the CO sensor, initially a mosquito repellent coil was ignited that caused a peak in the CO concentration and a slight peak in the CO concentration. To test the CO_2 sensor response time, the author breathed periodically very close to the sensor, which caused a significant narrow peak $>$ *1000ppm* to the CO_2 concentration. Furthermore, tests took place within a room where initially windows and doors were shut, and then the door was left ajar. It was observed that CO_2 concentration dropped progressively from over 600ppm to around 350ppm where it stabilised. Figure 14 shows an increase on CO_2 from repeated toasting of bread in the kitchen during breakfast.

6.2 Staging

Five DP Devices were placed at the same location next to each other and their measurements in terms of differential pressure, temperature and humidity were compared while the enclosure lid was off, on and under different lengths of tubes as shown in Figures 10, 11.

As can be seen from these figures, three devices, that are either without enclosure (DP_1), or without lid (DP_3) or without tubes (DP_5) have most similarities in all types of measurements. DP_4 with *1m* tubes comes closer next while DP2 with $5m$ tubes has more distance than the rest. However, the differences are still very small around *0.3Pa*.

6.3 Rollout & Maintenance

After successful staging, fifteen PM devices and three DP devices were constructed and tested using the above methods before being deployed in a dairy plant in Greece that produces milk and yoghurt. The results are discussed in the Section 8.

Each of the eighteen devices were added to the author's account in TeamViewer so that they were accessible remotely. As long as the device was connected to the internet a remote desktop was available offering full access to the middleware components as well as information on the logfiles, the strength of the WiFi signal etc. Whenever the internet was disrupted, the high-availability middleware services, through scheduled reboots and other service restarts, contributed in 99% of the cases to connectivity revival. For the 1% of the cases where connectivity was not restored, the health&safety officer went on the field and connected with an Ethernet cable the isolated device to a laptop connected to the Wireless Lan, thus giving the device an IP address so that it was again remotely accessible through TeamViewer and it could be reconfigured. In two cases, the sensors stopped working and had to be replaced. During the first year-and-half of operation, no additional on-field calibration was required.

7 Case studies

A dairy in Greece was used as a case study for the evaluation of the devices. It consists of three production areas: *Mixing, Asceptic filling* and *Packaging* which, in terms of environmental health, range from *hygienic environments* to *critical environments* that are highly sensitive to contaminants. In the former type, air quality needs to be as good as possible while in the latter, high volume and pressure in terms of air flow is required on top. Furthermore, the Asceptic filling area has slightly lower pressure than the surrounding spaces, in order to prevent microorganisms from entering. This is known as *negative pressure.* Although the plant is equipped with a modern *Programmable Logic Controller (PLC)* system that undertakes the monitoring of several parameters, it did not include sensors for particulate matter, or differential pressure, while more dense measurements of temperature and humidity were needed than were available by existing industrial sensing instruments. Because of the very high price of the latter, it was not possible to procure any additional sensors than were included in the initial installation of this equiopment. It was this gap that was fulfilled by the novel devices. After discussion with the plant stakeholders including the health&safety team, it was decided that the following aspects would be monitored by means of the devices developed in this work.

1. Monitor in real-time the concentration of $PM_{10}, PM_{2.5}, PM_{1.0}$ in fifteen points of interest in the three production areas. Three of these locations are outdoors and twelve are indoors.
2. Calculate mean values and respect legislation limits
3. Real-time value of differential pressure between six points of interest in the three areas. More specifically, it was decided to:
    (a) Monitor in real-time the differential pressure between at the center of a new extension of the plant, adjacent to the Mixing area and an existing ventilation point in the Mixing area, in order to understand whether the extension was adequately ventilated from this source.
    (b) Monitor in real-time the negative pressure in the Asceptic Filling area in order to check conformance with the desirable threshold ($5Pa$).
    (c) Monitor in real-time the the change of differential pressure between the adjacent Mixing and Asceptic filling areas that is caused by the opening of a door connecting the two.

4. Develop visual analytics in order to discover systematic sources of pollution.

8 Results

Eighteen sensing devices were deployed at the dairy (shown in Figure **??**). Three PM devices were deployed outdoors, north, south and west of the plant, respectively, and twelve indoors, in the three distinct production areas. Furthermore, three DP sensing devices were deployed in the same areas.

For the monitoring of air flow, two DP sensing devices were deployed in the Mixing area. Device DP_1 measures airflow at the end of the extension building (bottom) while Device DP_2 measures flow in proximity of a sealed door that connects this area with the Asceptic one. Device DP_3 monitors negative pressure in the Asceptic area.

8.1 Real-time ambient PM monitoring

Each PM sensing device was associated with a Grafana dashboard that shows time-series of three classes of particles, $PM_{10}, PM_{2.5}$ and $PM_{1.0}$ (red, orange and blue colour lines, respectively in Figure 15). Temperature and humidity are shown in separate graphs (yellow and purple, respectively). The time period of the graph is configurable by a widget on the top right of the dashboard. Visual analytics are possible by means of these dashboards. Figure 15 shows that the concentration of all particles close to the strainer, located in the Mixing area, was unusually high between 3am and 12pm on April 15, 2020, while temperature decreased and humidity increased during the night until approximately 8:30am, as was expected.

8.2 Trend and seasonality

Furthermore, the timeseries from each monitoring device were decomposed using python. The hourly mean, trend and seasonality of the monitored PM1.0 data in the center of the Mixing Area for August 2020 are shown in Figure 16 (top, middle and bottom). Hourly mean values are normally very low, around 5 ug/m^3 while they can increase up to 20 ug/m^3, which is medium, especially if the peak has small duration as shown in this graph. There is no legislation limit for PM1.0, yet.

A clear daily seasonality can be observed, which makes sense for the plan operating cycle that is higher during the day. Furthermore, an increased trend is noted during August 3, 17, 24, 30, which happen to be Mondays, indicating that the peak of the plan operational activity is at the begining of the week (Monday-Wednesday). Furthermore, between August 10 and 17 most of the personnel was on summer vacation, which corresponds with reduced PM concentration values observed in this period.

8.3 Real-time pressure monitoring

A timeseries of DP measurements in the Asceptic area over 6 and a half days, from the 8/4/2020 to 14/4/2020 is shown in Figure 17. It can be observed that between 8/4 and 10/4 at noon, the average pressure is approximately at -5Pa, which is the desirable value. However, from 10/4 to 14/4, the average pressure drops to -7.5Pa, which is lower than the excpected value. Furthermore, several peaks, e.g., on 9/4, 10/4 and 14/4 in the afternoon, indicate some unusual activity. On the other hand, positive peaks could be attributed to the door connecting this area with the adjecent area, opening for a few seconds. The 6-hour average trend is also shown in blue in this Figure.

8.4 Discovery of systematic sources of pollution

Two examples are provided for illustration purposes. In the first example, the observation that one of the devices reported unusually high and raising temperature led, after inspection by the health&safety officer, to the discovery that the air-condition in one of the production areas had not been turned on by accident. In the second example, the placement of neighbouring devices in two disjoint production areas proved helpful for determining a door that was left open by accident; this door separates the areas and as a result a heavy steam that was generated from cleaning a machine in one area leaked into the other. This was observed by the obvious relationship of the matching peaks in particulate matter concentration in the two rooms (Figures 18,19)

9 Conclusions and future work

This work proposes a systematic approach for developing highly reliable air quality sensing devices using low-cost sensor technology and IoT. Reliability is enhanced in three locations: sensor, device/edge and cloud level, by means of novel, flexible hardware platform and delay and fault tolerant middleware components. A scalable cloud service for the visualisation of the measurements is also presented.

Two versions of the hardware platform are presented, each tailored for a different subset of air quality sensors requirements. Similarly, the middleware consists of three components: store-and-forward and publish-subscribe are tailored to the architectures presented, while HTTP-over-platform.io is appropriate for embedded systems (without operating system). Each middleware component is structured as a service-oriented architecture, comprising a set of services that implement one or more of the generic middleware functions. One of the middleware services in each stack, is responsible for the remote reconfigurability of the devices thus decreasing the maintenance burden of the devices in remote and large installations.

Eighteen instances of three novel sensing device types, embody of the above hardware and software novel components, each measuring either particulate matter or differential pressure or outdoor gases. Their development, testing and assembly at a high TRL level is described next, concluding in their validation in a state-of-the art industrial setting where they have been operating for 18 months, at the time or writing this paper. During this time it was proven that they had significant contribution to several use-cases such as, checking conformance with health&safety guidelines, monitoring negative room pressure and discovering systematic sources of pollution thus protecting the personnel and the products.

As next steps, there is ongoing work on analysing the sensor measurements from the pilot cases and mining knowledge pertaining to the plant operation as well as corellating the pollutant concentrations with other pollutants and micro-organisms. This is the subject of an upcoming paper.

References:

[1] Krzyzanowski, M., Martin, R. V., Van Dingenen, R., van Donkelaar, A., and Thurston, G. D., Exposure assessment for estimation of the global burden of disease attributable to outdoor air pollution, *Environ. Sci. Technol.*, Vol.46, 2012, pp. 652–660.

[2] WHO- Regional office for Europe, Review of evidence on health aspects of air pollution – REVIHAAP project: final technical report, 2013.

[3] Samet J.M., Dominici F., Curriero F.C., Coursac I., Zeger, S.L., Fine Particulate Air Pollution and Mortality in 20 U.S. Cities, 1987–1994, *N Engl J Med* , Vol. 343, 2000, pp. 1742-1749

[4] WHO- Regional office for Europe, Health effects of particulate matter. Policy implications for countries in eastern Europe, Caucasus and central Asia, 2013.

[5] Bano, N., Assessment of indoor environmental impacts on human health (Case study: Glass city, Firozabad (India), *Pollution*, Vol. 3, No.2, 2017, pp. 175-183

[6] Setton, E., Marshall J. D., Brauer M., Lundquist K. R., Hystad P., Keller P., and Cloutier-Fisher D., The impact of daily mobility on exposure to traffic-related air pollution and health effect estimates, *Journal of Exposure Science and Environmental Epidemiology*, Vol. 21, No. 1, 2011, pp. 42–48.

[7] Setti, L., Passarini, F., de Gennaro, G., Di Gilio, A., Palmisani, J., Buono, P., Fornari, G., Perrone, M., G., Piazzalunga, A., Barbieri, P., Rizzo, E. and Miani, A., Evaluation of the potential relationship between Particulate Matter (PM) pollution and COVID-19 infection spread in Italy:first observational study based on initial epidemic diffusion, *BMJ Open*,Vol.10, No.9, 2020;

[8] Wu, X., Nethery, R. C., Sabath, M. B., Braun, D., Dominici, F., Air pollution and COVID-19 mortality in the United States: Strengths and limitations of an ecological regression analysis, *Science Advances*, Vol. 6, No. 45, 2020

[9] Saadat, S., Rawtani, D., Hussain, CM., Environmental perspective of COVID-19, *Science of The Total Environment* Vol. 728, 2020, pp.138870

[10] Le Quéré, C., Jackson, B.R., Jones, W.M., Smith, P.J.A., Abernethy, S., Andrew, M.R., De-Gol, J.A., Willis, R.D., Shan, Y., Canadell, G.J., Friedlingstein, P., Creutzig, F., Peters, P.G., Temporary reduction in daily global CO2 emissions during the COVID-19 forced confinement *Nat. Clim. Chang.* Vol.10, 2020, pp.647–653

[11] Weissert, L. F., Alberti, K., Miskell, G., Pattinson, W., Salmond, J. A., Henshaw, G., Williams, D. E., Low-cost sensors and microscale land use regression: Data fusion to resolve air quality variations with high spatial and temporal resolution. *Atmos. Environ.* 2019, Vol. 213, pp.285-295.

[12] Masiol, M., Squizzato, S., Chalupa, D., Rich, D. Q., Hopke, P. K., Spatial-temporal variations of summertime ozone concentrations

across a metropolitan area using a network of low-cost monitors to develop 24 hly land-use regression models, *Science of The Total Environment*, Vol.654, 2019, pp.1167-1178.

[13] Delgado-Saborit J. M., Use of real-time sensors to characterise human exposures to combustion related pollutants, *J Environ Monit.* , Vol.14, No.7, 2012, pp. 1824-37.

[14] Mead, M. I., Popoola, O. A. M., Stewart, G. B., Landshoff, P., Calleja, M., Hayes, M., Baldovi, J. J., et al., The use of electrochemical sensors for monitoring urban air quality in low-cost, high-density networks, *Atmospheric Environment*, Vol. 70, 2013, pp. 186–203

[15] Afzal, A., Cioffi, N., Sabbatini, L., Torsi, L., NO_x sensors based on semiconducting metal oxide nanostructures: progress and perspectives. *Sensors and Actuators B: Chemical*, 171, 2012, pp. 25-42

[16] Hasenfratz D., Saukh O., Thiele L. *On-the-Fly Calibration of Low-Cost Gas Sensors.* In: Picco G.P., Heinzelman W. (eds) Wireless Sensor Networks. EWSN 2012. Lecture Notes in Computer Science, Vol. 7158, 2012

[17] Borrego, C., Ginja, J., Coutinho, M., Ribeiro, C., Karatzas, K., Sioumis, Th., Katsifarakis, N., Konstantinidis, K., De Vito, S., Esposito, E., Salvato, M., Smith, P., André, N., Gérard, P., Francis, L. A., Castell, N., Schneider, P., Viana, M., Minguillón, M. C., Reimringer, W., Otjes, R. P., von Sicard, O., Pohle, R., Elen, B., Suriano, D., Pfister, V., Prato, M., Dipinto, S., Penza, M., Assessment of air quality microsensors versus reference methods: The EuNetAir Joint Exercise - Part II, *Atmospheric Environment*, Vol. 193, 2018, pp: 127-142

[18] Singhal, A., and Cowie, M. R., The Role of Wearables in Heart Failure, *Current heart failure reports*, Vol.17, No.4, 2020, pp.125–132.

[19] Williams, David E., Low Cost Sensor Networks: How Do We Know the Data Are Reliable?, *ACS Sensors*, Vol.4, No.10, 2019, pp.2558-2565

[20] *Lightweight Machine to Machine Technical Specification*, Open Mobile Alliance http://www.openmobilealliance.org/release/LightweightM2M/V1 0-20170208-A/OMA-TS-LightweightM2M-V1_0-20170208-A.pdf

[21] Katsiri E., Bacon, J., and Mycroft, A., Linking Sensor Data to Context-Aware Applications using Abstract Events, *Journal of Pervasive Computer and Communications*, special issue on Managing Context Information in Mobile and Pervasive Environments, Vol.3, No.4,2007, pp.347-377

[22] Katsiri, E., and Lopez-Ipina, D., An eca rule-matching service for simpler development of reactive applications, *IEEE Distributed Systems Online*, 2001

[23] Kumar, P., Martani, C., Morawska, L., Norford, L., Choudhary, R., Bell, M., Leach, M., Indoor air quality and energy management through real–time sensing in commercial buildings, *Energy and Buildings*, Vol.111, 2016, pp.145-153

[24] Hagan, D. H., Gani,S., Bhandari, S., Patel, K., Habib, G., Apte, J., Hildebrandt Ruiz, L., and Kroll, H. J., Inferring Aerosol Sources from Low-Cost Air Quality Sensor Measurements: A Case Study in Delhi, India, *Environ. Sci. Technol. Lett.*,Vol.6, No.8, 2019 pp.467-472

[25] Yazdi, M. N., Arhami, N., Delavarrafiee, M., Ketabchy, M., Developing air exchange rate models by evaluating vehicle in-cabin air pollutant exposures in a highway and tunnel setting: case study of Tehran, Iran, *Environ Sci Pollut Res Int.*, Vol.26, No.1, 2019, pp.501-513

[26] deSouza, P ., Lu, R., Kinney P., Zheng, S., Exposures to multiple air pollutants while commuting: Evidence from Zhengzhou, China, *Atmospheric Environment*, Vol.247, 2020, pp.118168

[27] Bukowiecki, N., Dommen J., , Prévôt, A.S.H., Richter, R., Weingartner, E., and Baltensperger U., A mobile pollutant measurement laboratory—Measuring gas phase and aerosol ambient concentrations with high spatial and temporal resolution, *Atmospheric Environment*, Vol.36, 2002, pp. 5569-5579

[28] Apte, J.S., Messier, K.P., Gani, S., Brauer, M., Kirchstetter, T.W., Lunden, M.M. et al., High-resolution air pollution mapping with google street view cars: Exploiting big data, *Environmental Science & Technology*, Vol.51, 2017, pp. 6999-7008

[29] Kumar, P., Morawska, L., Martani, C., Biskos, G., Neophytou, M., Di Sabatino, S.,

et al. (2015) *The rise of low-cost sensing for managing air pollution in cities* Environment International, 75 (2015), pp. 199-205, 10.1016/j.envint.2014.11.019

[30] Morawska, L., Thai, P.K., Liu, X., Asumadu-Sakyi, A., Ayoko, G, Bartonova,A., et al. (2018) *Applications of low-cost sensing technologies for air quality monitoring and exposure assessment: How far have they gone?* Environment International, 116 (2018), pp. 286-299, 10.1016/j.envint.2018.04.018

[31] Snyder, E.G., Watkins, T.H., Solomon,P.A., Thoma, E.D., Williams, R.W., Hagler, G.S.W., et al. (2013) *Applications of low-cost sensing technologies for air quality monitoring and exposure assessment: How far have they gone?* Environmental Science & Technology, 47 (2013), pp. 11369-11377, 10.1021/es4022602

[32] Dutta, P., Aoki, P.M., Kumar, N., Mainwaring, A., Myers, C., Willett, W., et al., Common sense: Participatory urban sensing using a network of handheld air quality monitors, *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, 2009, pp. 349-350,

[33] Elen, B., Peters, J., Poppel, M.V., Bleux, N., Theunis, J., Reggente, M., et al., The aeroflex: A bicycle for mobile air quality measurements, *Sensors*, Vol.13, 2013, pp. 221-240

[34] Mahajan, S., and Kumar, P.,Evaluation of low-cost sensors for quantitative personal exposure monitoring, *Sustainable Cities and Society*, Vol.57, 2020, pp. 102076

[35] Anjomshoaa, A., Duarte, F. , Rennings, D., Matarazzo, T.J., deSouza, P., Ratti, C., City scanner: Building and scheduling a mobile sensing platform for smart city services, *IEEE Internet of Things Journal*, Vol.5, 2018, pp. 4567-4579

[36] DeSouza, P., Anjomshoaa, A.,Duarte, F., Kahn, R., Kumar, P., Ratti, C., Air quality monitoring using mobile low-cost sensors mounted on trash-trucks: Methods development and lessons learned, *Sustainable Cities and Society*, Vol.60, 2020, pp.102239

[37] deSouza, P., Kahn, A. R., Limbacher, A. J., Marais, A. E., Duarte, F., and Ratti, C., Combining low-cost, surface-based aerosol monitors with size-resolved satellite data for air quality applications, *Atmospheric Measurement Techniques*, Vol.13, No.10, pp.5319-5334

[38] Koukouli, M.E., Skoulidou, I., Karavias, A., Parcharidis, I., Balis, D, Manders-Groot, A.M.M., Segers, A.J., Eskes, H., van Geffen, J., Sudden changes in nitrogen dioxide emissions over Greece due to lockdown after the outbreak of Covid-19, *Atmospheric Chemistry and Physics*, Vol.21, No.21, pp.1759-1774

[39] Koukouli, M.E., Skoulidou, I., Karavias, A., Parcharidis, I., Balis, D, Manders-Groot, A.M.M., Segers, A.J., Eskes, H., van Geffen, J., Sudden changes in nitrogen dioxide emissions over Greece due to lockdown after the outbreak of Covid-19, *Atmospheric Chemistry and Physics*, Vol.21, No.21, pp.1759-1774

[40] Capezzuto, L.; Abbamonte, L.; De Vito, S.; Massera, E.; Formisano, F.; Fattoruso, G.; Di Francia, G.; Buonanno, A. A maker friendly mobile and social sensing approach to urban air quality monitoring. *In Proceedings of the 2014 IEEE on SENSORS*, Valencia, Spain, ; pp. 12–16.

[41] Murty, R.N.; Mainland, G.; Rose, I.; Chowdhury, A.R.; Gosain, A.; Bers, J.; Welsh, M. Citysense: An urban-scale wireless sensor network and testbed. *In Proceedings of the 2008 IEEE Conference on Technologies for Homeland Security*, Waltham, MA, USA, 12–13 May 2008; pp. 583–588

[42] Dutta, P.; Aoki, P.M.; Kumar, N.; Mainwaring, A.; Myers, C.; Willett, W.; Woodruff, A. Common sense: Participatory urban sensing using a network of handheld air quality monitors. *In Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, Berkeley, CA, USA, 4–6 November 2009; ACM: New York, NY, USA, 2009; pp. 349–350.

[43] Kadri, A.; Yaacoub, E.; Mushtaha, M.; Abu-Dayya, A. Wireless sensor network for real-time air pollution monitoring. *In Proceedings of the 2013 1st International Conference on Communications, Signal Processing, and their applications (ICCSPA)*, Sharjah, United Arab Emirates, 12–14 February 2013; pp. 1–5.

[44] Postolache, O.A.; Pereira, J.M.D.; Girao, P.M.B.S. Smart sensors network for air quality monitoring applications. *IEEE Trans. Instrum. Meas.* 2009, 58, 3253–3262.

[45] Hasenfratz, D.; Saukh, O.; Sturzenegger, S.; Thiele, L. Participatory air pollution monitoring using smartphones. *Mobile Sensing* 2012, 1, 1–5.

[46] Jiang, Y.; Li, K.; Tian, L.; Piedrahita, R.; Yun, X.; Mansata, O.; Lv, Q.; Dick, R.P.; Hannigan, M.; Shang, L. MAQS: A personalized mobile sensing system for indoor air quality monitoring. *In Proceedings of the 13th International Conference on Ubiquitous Computing*, Beijing, China, 17–21 September 2011; ACM: New York, NY, USA, 2011; pp. 271–280.

[47] Jelicic, V.; Magno, M.; Brunelli, D.; Paci, G.; Benini, L. Context-adaptive multimodal wireless sensor network for energy-efficient gas monitoring. *IEEE Sens. J.* 2013, 13, 328–338.

[48] Mansour, S.; Nasser, N.; Karim, L.; Ali, A. Wireless sensor network-based air quality monitoring system. *In Proceedings of the 2014 International Conference on Computing, Networking and Communications (ICNC)*, Honolulu, HI, USA, 2014, pp. 545–550.

[49] Sun, L.; Wong, K.C.; Wei, P.; Ye, S.; Huang, H.; Yang, F.; Westerdahl, D.; Louie, P.K.; Luk, C.W.; Ning, Z. Development and application of a next generation air sensor network for the Hong Kong marathon 2015 air quality monitoring. *Sensors* 2016, 16, 211.

[50] Honicky, R.; Brewer, E.A.; Paulos, E.; White, R. N-smarts: Networked suite of mobile atmospheric real-time sensors. *In Proceedings of the Second ACM SIGCOMM Workshop on Networked Systems for Developing Regions*, Seattle, WA, USA, 18 August 2008; ACM: New York, NY, USA, 2008; pp. 25–30

[51] Lane, N.D.; Miluzzo, E.; Lu, H.; Peebles, D.; Choudhury, T.; Campbell, A.T. A survey of mobile phone sensing. *IEEE Commun. Mag.* 2010, 48, pp. 140–150.

[52] Helbig, C.; Bauer, H.S.; Rink, K.; Wulfmeyer, V.; Frank, M.; Kolditz, O. Concept and workflow for 3D visualization of atmospheric data in a virtual reality environment for analytical approaches. *Environ. Earth Sci.* 2014, 72, pp. 3767–3780.

[53] Adam Osborne, An Introduction to Microcomputers Volume 1: Basic Concepts, Osborne-McGraw Hill Berkeley California USA, 1980 pp. 116–126

[54] "FTDI Products". www.ftdichip.com. Retrieved 22 March 2018.

[55] Leens, F., An Introduction to I^2C and SPI Protocols *IEEE Instrumentation & Measurement Magazine*, 2009, pp. 8-13

[56] Irazabel, J.M. and Blozis, S., Philips Semiconductors, I^2C-Manual Application Note ref. AN10216–0 , 2003.

[57] Leens, F., Solutions for SPI Protocol Testing and Debugging in Embedded System Byte Paradigm's White Paper, 2008pp, pp.1-9.

[58] Bacciarelli, L., et al., Design Testing and Prototyping of a Software Programmable I^2C/SPI IP on AMBA Bus, *Conference on Ph.D. Research in MicroElectronics and Electronics* (PRIME'2006),2006, pp. 373-376

[59] Hanabusa, R., Comparing JTAG SPI and I^2C, Spansion's application note, 2007, pp. 1-7.

[60] Myers, P. Interfacing Using Serial Protocoles: Using SPI and I2C. *In Proc. ESP 2005*, 2007, pp. 1-9

[61] I2C-bus specification Rev 6" (PDF). NXP Semiconductors, 2014, (Archived (PDF) from the original on April 26, 2021).

[62] Horowitz, Paul; Hill, Winfield. The art of electronics (PDF) (2nd ed.). Cambridge [u.a.]: Cambridge Univ. Press. 1989

[63] Garner, R.; Taylor, D. "Surface mount packaging". *Microelectronics Journal.* 17 (3), 1986, pp.5–13.

[64] *Grafana: The open observability platform*

[65] Adrienko, G., Adrienko, N., Drucker, S., Fekete, J.D., Fisher, D., Idreos, S., Li, G., Ma, K.-L., et al., Big Data Visualization and Analytics: Future Research Challenges and Emerging Applications, *3rd International Workshop on Big Data Visual Exploration and Analytics*, 2020 https://grafana.com

[66] *Purpose-built, open-source time-series database: InfluxDB* https://www.influxdata.com

[67] Fadhel, M., Sekerinski, E., Yao, S., A Comparison of Time Series Databases for Storing Water Quality Data, *Mobile Technologies and Applications for the Internet of Things*, Vol.12141, 2018, pp.371-383

[68] *Elasticsearch* https://www.elastic.co/what-is/elasticsearch

[69] Reas, C., and Fry, B., *Processing: A Programming Handbook for Visual Designers*, 2007, pp. 710