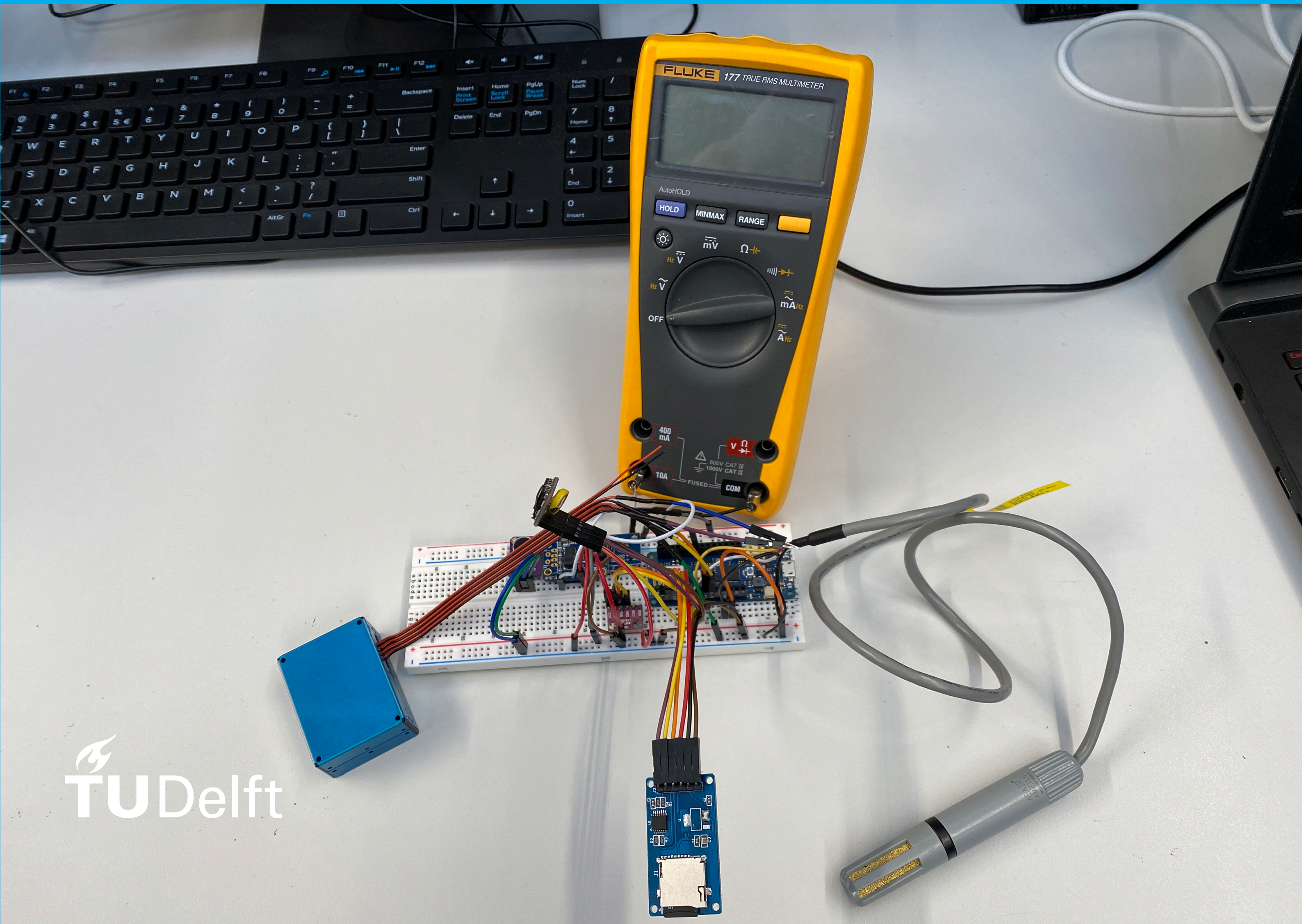
Solar Powered Weather Station

Sensors and Control

Roan Föllings and Henk van Grootheest

Group B3



Solar Powered Weather Station

Sensors and Control

by

Roan Föllings and Henk van Grootheest

To obtain the degree of Bachelor of Science.

Supervised by Dr. Patrizio Manganiello and Dr. Mirco Muttillo.
Proposed by Dr. Patrizio Manganiello.
Department of Electrical Sustainable Energy.

June 18, 2021.

Abstract

Increasing concerns surrounding the influence of urban climate on the liveability of cities and the health of citizens lead to a need for environmental data on a more localized scale. Self-powered weather stations that use the many WiFi networks spanning the city to transmit their data are a promising solution to increase the amount of data on localized urban climate. Conventional weather stations often measure wind, rain and sun radiance, phenomena that are heavily influenced by the many objects present in an urban environment, which is why they are usually placed outside of the city or on top of buildings. Self-powered weather stations placed on street level can be used to gather data on temperature, humidity, air pollution and noise pollution. If a good trade-off is made between the sensing capacities, the battery capacity and the solar panel size, a weather station like this can do frequent measurements throughout the year, surviving the winter months of low solar panel power generation due to reduced irradiation. In the simulations done for this project, it was found that a simple algorithm that linearly varies the relative measuring frequency of the most power-hungry sensors with the available battery charge greatly improved the average measuring frequency and lifetime of the weather station. Thanks to the increased power budget, additional sensors can be added, such as gas sensors. Furthermore, the weather station can be made more compact and affordable by reducing the size of the solar panel and/or the battery capacity of the weather station. The final setup can be seen in figure 5.2 and the web page can be seen in figure 7.1.

Contents

1 Introduction

1.1. System overview

The goal of this project is to make an autonomous weather station which only runs on solar energy. A large number of these stations are to be placed in and around urban areas where they will gather data about humidity, temperature, noise, and particulate matter. This information will be uploaded via WiFi to a database where it can be analysed. The results may be used in traffic planning, city-planning or to make weather predictions. In order to make it feasible to have a lot of these stations, the stations themselves should be maintenance-free for at least one year, as small as possible and affordable.

1.1.1. Structure

The project is divided into three groups, each focusing on a particular part of the station (summarised in figure 1.1).

The first group is responsible for the power generation of the Photovoltaic (PV) array and for the design of the housing and support structure of the station (including a mounting system to allow attachment to an object like a lantern pole).

The second group focuses on the power electronics that connects the output of the PV-array to the micro-controller. The main focus is maximum power point tracking (MPPT), to get the maximum power transfer between the panel and the controller. This group is also responsible for the power management when it comes to battery charging and discharging.

The third group takes care of the choice and implementation of the sensors. Based on the power generation of the solar panels and consumption of the system, an algorithm is made to optimize the frequency of measurement. After collecting the data, this group makes sure it is periodically uploaded to a database via WiFi. In this thesis, the design process of the third group is explained.

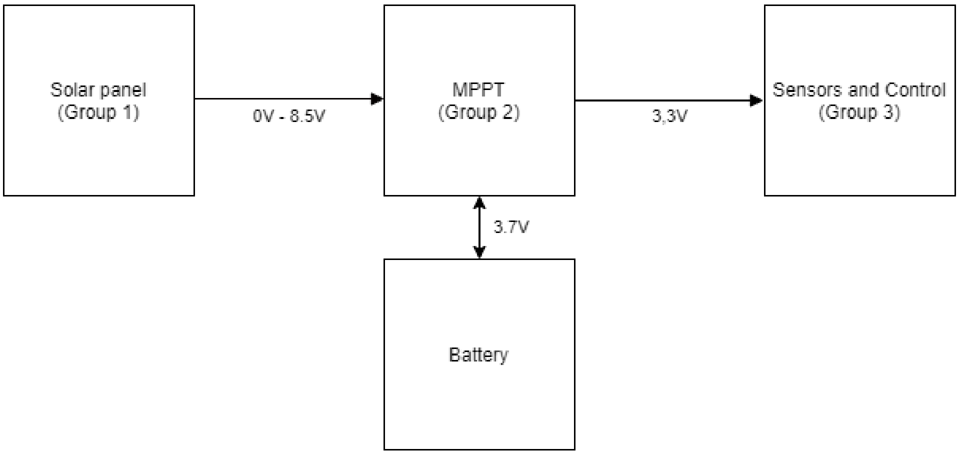


Figure 1.1: System overview with sub-group division.

2 Programme of requirements

Sections 2.1 and 2.2 describe the programme of requirements of the complete weather station. Section 2.3 translates these requirements to the relevant requirements for the Sensors and Control group.

2.1. Mandatory requirements

Functional requirements:

- The system must measure at an interval of at least one hour.
- The system must transmit the measured data at least once per week.
- The system must send the measured data over WiFi.
- The system must measure:
  - Temperature
  - Humidity
  - Air pressure
  - PM2.5/PM5/PM10 pollution

Non-functional requirements:

- The system must function in an urban environment for at least 1 year.
- The only external power source must be PV.
- The maximum size of the box must be 250x250x300mm.

2.2. Trade-off requirements

- The system should contain a mounting system.
- It is preferable to also measure:
  - NOx
  - O_3
  - SO_2

2.3. Programme of requirements Sensors and Control

The goal for this subgroup is to choose sensors for the weather station and interface them with a control unit. Based on the power generation of the solar panel and the power consumption of the control unit with the sensors, a suitable algorithm needs to be created to conserve energy in times of low generation (this will mostly concern the winter period) and to do extra measurements in times of over-generation. Furthermore, the obtained data needs to be written to an on-board storage and periodically (at least weekly) sent to a web server using WiFi. The entire system should fit within a 250 x 250 x 300 mm enclosure.

2.4. Problem definition

A selection of appropriate sensors needs to be made. These sensors should be able to measure their meteorological and environmental variables without being hindered by the urban siting of the weather station. It is also important that the sensors have a low power consumption to allow the system to do more frequent measurements and to work during prolonged times of low generation. Chapter 3 analyses the operational principles and upsides and downsides of different sensor technologies. Since the generation of solar panels is highly dependent on the season (especially in regions far away from the equator), the system needs to be adaptable to optimise power use. During times of over-generation (mostly during the day and summer), more frequent measurements can be performed, while energy needs to be conserved in times of under-generation (mostly during the night and winter). An algorithm can be created that tries to predict future power generation based on the generation of the past, while keeping track of the battery charge. This gives the system an idea of the power budget it can use for measurements.

3
Sensor technology

In order to make an informed decision on the type of sensors to implement in the weather station, some background on the sensor technology is required. This chapter gives a short description of different ways to measure physical quantities relevant to an urban weather station.

3.1. Temperature sensors

3.1.1. Metal resistance thermometers

Most pure metals have an almost linear temperature-resistance relationship within the temperature range relevant to the weather station [1]. Metal resistance thermometers are often made by winding a platinum wire around a dielectric core. Because of the low resistance of this type of sensor (typically between 100 and 1000 Ω), special care must be taken to ensure that the device does not heat itself up, influencing the measurement and increasing power consumption. The relatively low resistance can also cause thermal variations of the resistance of the lead wires that have a substantial influence on the temperature measurement [2]. Advantages of this type of sensor are the high sensitivity and wide operating range.

3.1.2. Thermistors

A thermistor is a type of temperature sensor that also uses the fact that the resistance of the device varies almost linearly with the temperature within a certain temperature range [1]. Because this type of device employs a metallic oxide instead of a metal, the device typically has a much higher resistance (in the kΩ range). This causes these devices to be less affected by self-heating and lead wire resistance variations. The downside of this type of sensor compared to the metal resistance sensor is the lower accuracy and the smaller semi-linear operating range [2].

3.1.3. Thermocouples

Thermocouples use the Seebeck effect to measure temperature. This effect causes a temperature-dependent voltage drop over two metal wires with different properties that are connected in a junction, shown in figure 3.1 [1]. Thermocouples require good read-out methods, since simply connecting a voltmeter introduces new junctions and thus a generates the Seebeck effect. Furthermore, thermocouples are relatively inaccurate. Advantage are the big temperature range and the high maximum temperatures that thermocouples can be used in [2].

3.2. Relative humidity sensors

3.2.1. Capacitive humidity sensors

The relative humidity (RH) of air influences the dielectric constant of certain materials. This property can be used to determine the RH by measuring the capacitance of a capacitor with a (polymer or metal oxide) dielectric whose dielectric constant varies with the relative humidity of the surrounding air [3]. A typical graph of the capacitance-RH relationship of such a device can be seen in figure 3.2.

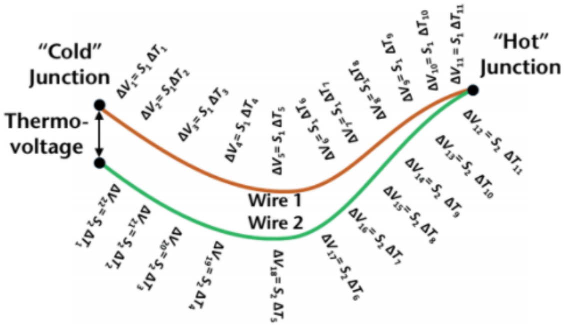

"Cold" Junction
Thermo-voltage
"Hot" Junction
Wire 1
Wire 2
$\Delta V_1 = S_1 \Delta T_1$
$\Delta V_2 = S_1 \Delta T_2$
$\Delta V_3 = S_1 \Delta T_3$
$\Delta V_4 = S_1 \Delta T_4$
$\Delta V_5 = S_1 \Delta T_5$
$\Delta V_6 = S_1 \Delta T_6$
$\Delta V_7 = S_1 \Delta T_7$
$\Delta V_8 = S_1 \Delta T_8$
$\Delta V_9 = S_1 \Delta T_9$
$\Delta V_{10} = S_1 \Delta T_{10}$
$\Delta V_{11} = S_1 \Delta T_{11}$
$\Delta V_{12} = S_2 \Delta T_{11}$
$\Delta V_{13} = S_2 \Delta T_{10}$
$\Delta V_{14} = S_2 \Delta T_9$
$\Delta V_{15} = S_2 \Delta T_8$
$\Delta V_{16} = S_2 \Delta T_7$
$\Delta V_{17} = S_2 \Delta T_6$
$\Delta V_{18} = S_2 \Delta T_5$
$\Delta V_{19} = S_2 \Delta T_4$
$\Delta V_{20} = S_2 \Delta T_3$
$\Delta V_{21} = S_2 \Delta T_2$
$\Delta V_{22} = S_2 \Delta T_1$

Figure 3.1: The Seebeck effect in thermocouples [1].

Capacitive RH sensors are appealing because of their low temperature coefficient, wide temperature operating range and full recovery from condensation [4].

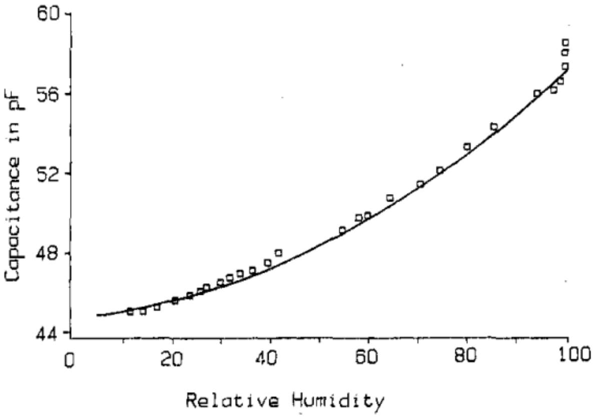


Figure 3.2: Typical graph of capacitance as a function of relative humidity [3].

3.2.2. Resistive humidity sensors

In resistive humidity sensors, a salt or polymer substrate whose resistive properties change with the RH of the surrounding air is used to measure the RH of the air [4]. A typical graph of the exponential response of such a sensor is shown in figure 3.3. Big downsides of this type of sensor are the significant temperature dependency and the weakness to condensation [4].

3.3. Air pressure sensors

3.3.1. Capacitive air pressure sensors

In capacitive air pressure sensors, the air pressure is measured with an elastic diaphragm. This diaphragm functions as one plate of a parallel-plate capacitor. Because the air pressure causes slight changes in the distance between the plates, the capacitance of the device has a relationship with the air pressure. Because the measurement is temperature dependent, the temperature needs to be measured to apply compensation techniques to obtain high accuracy air pressure measurements [1].

3.3.2. Piezoresistive air pressure sensors

The piezoresistive effect, where the resistivity of materials changes when a force is applied to them, can be used to measure air pressure. As in the capacitive air pressure sensor, a flexible surface is used to turn air pressure into movement. Resistors are placed on this flexible surface to form a measuring setup (a common setup is the 4-resistor Wheatstone bridge, shown in figure 3.4). Due to the bending of the surface, the resistors will be compressed or expanded, which slightly changes their resistive properties. These differences can be measured to determine the air pressure change.

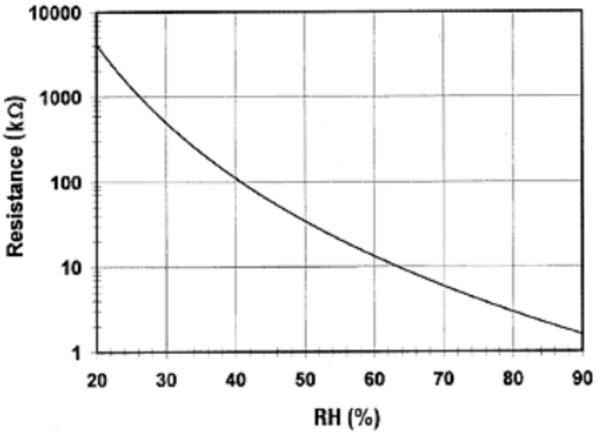


Figure 3.3: Typical graph of resistance as a function of relative humidity [4].

Because the resistive properties of materials also change with temperature, accurate measurements require temperature compensation [1].

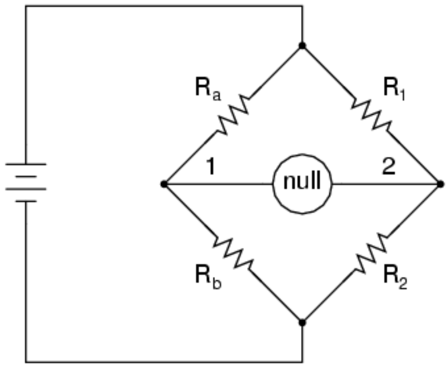


Figure 3.4: Wheatstone bridge circuit diagram [5].

3.4. Acoustic sensors

3.4.1. Capacitive microphones

Capacitive microphones use a stretched diaphragm to catch air vibrations. This diaphragm is one plate of a parallel-plate capacitor (the other plate being stationary). A schematic overview is shown in figure 3.5. Because the distance between the capacitor plates is directly dependent on the power of the air vibrations, the changes in capacitance can be measured as a voltage containing the information [6].

Advantages of capacitive microphones are that they can be made very small without significant loss of performance and that a flat frequency response can be obtained over a relatively wide band [7].

3.4.2. Inductive microphones

Inductive microphones, also called dynamic pressure microphones, catch air vibrations with a membrane that is attached to a coil. This coil can move freely along a permanent magnet. When the membrane is vibrating, the coil will move back and forth along the permanent magnet. A schematic overview is shown in figure 3.6. Because of electromagnetic conduction, the change in flux flowing through the coil will generate a potential difference between both ends of the coil. This potential difference is proportional to the air vibrations [8].

Advantages of inductive microphones are that they do not need an external power supply and that their measurements remain accurate in a wide variety of environmental conditions [8].

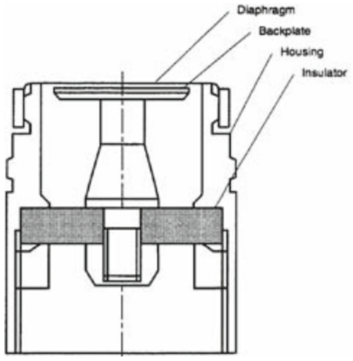


Figure 3.5: Schematic overview of a capacitive microphone [6].

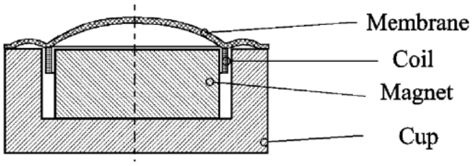


Figure 3.6: Schematic overview of an inductive microphone [8].

3.5. Gas sensors

Semiconductor gas sensors use the fact that the interaction between certain metal-oxides and gasses produce surface charge differences, resulting in a changing resistivity of the material. By measuring the resistance of a metal-oxide substrate, a measure of the gas concentration in the surrounding air is obtained. Different semiconductors react more or less strongly with different gasses, so a substrate that is sensitive to the gas(ses) that need to be measured needs to be chosen. Since the induced change of substrate resistance is strongly dependent on temperature (see figure 3.7), gas sensors often include a heater to bring the substrate temperature to the desired range. Downsides of semiconductor gas sensors are the high power use (because of the heater) and the fact that many substrates react to a range of different gasses (no perfect selectivity). An advantage of this type of sensor are the low cost of production and the small size of the sensor [9].

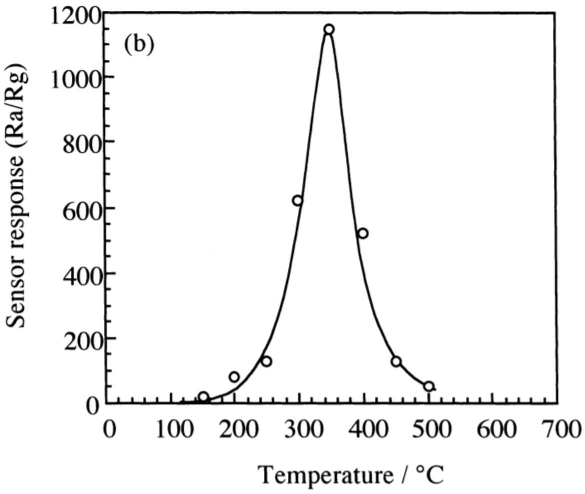


Figure 3.7: Temperature dependence of gas sensor response [9].

3.6. PM sensors

Affordable particulate matter (PM) sensors use laser scattering to measure particles of different sizes suspended in the air [10]. Because light rays get reflected by particles around the size of their wavelength, different frequency microwaves can be used to get an indication of the amount of particles of a particular size in air. It should be noted that different types of particles of the same size have different reflective properties, so the measurement should always be taken as an indication [11]. Furthermore, the reflective properties of suspended particles depend on the relative humidity of the air, where a higher relative humidity causes an underestimation of the particulate matter mass concentration [12]. A schematic overview of the working principle of a laser scattering PM sensor is shown in figure 3.8.

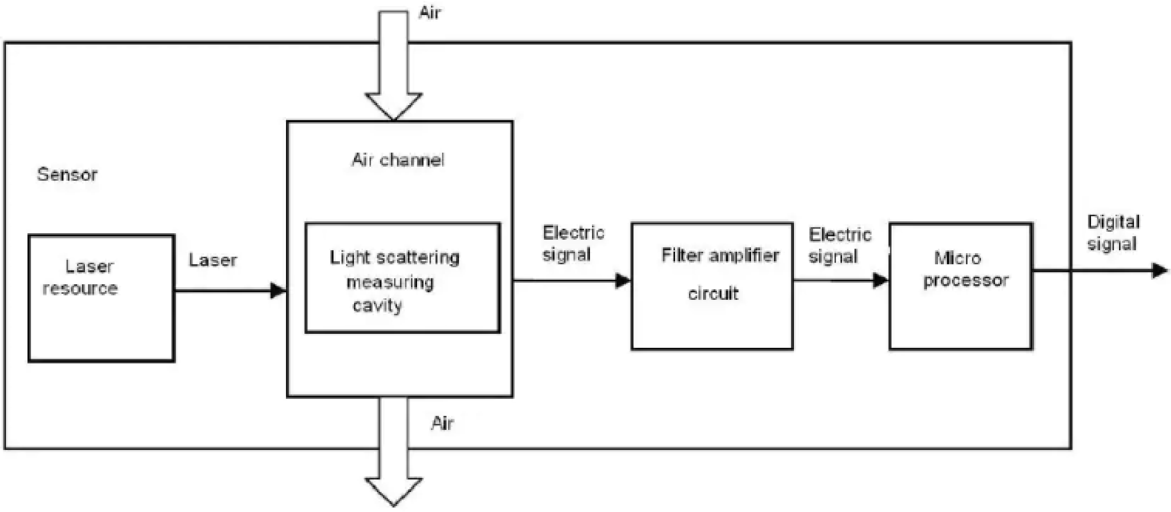


Figure 3.8: Schematic overview of a laser scattering PM sensor [13].

3.7. Wind, rain and light sensors

Because the measurement of wind, rain and light was determined to be impractical for an urban weather station, no deep literature study was done into the physics of such sensors. The difficulty in sensing these phenomena mostly has to do with the interaction between them and the urban environment, not with the sensor technology itself. A more detailed explanation of the choice not to include these sensors is given in chapter 4.

4 Choice of sensors and components

The weather station will be used in an urban environment. An important requirement for the placement is that the solar panel needs to face the south (assuming the weather station is used in the Netherlands), ensuring sufficient power generation. Another requirement is that the weather station is not placed on the ground, which could cause condensation or flooding problems (the weather station includes an attachment system to facilitate above-ground level placement).

According to the World Meteorological Organisation, it is crucial that the surrounding area and obstacles close to the instruments do not decrease the representativeness of the measurements [1]. Since the objective is to create a weather station for an urban environment, the design choices for the weather station need to be based on this prerequisite. The merit of an urban-localized weather station is that it could provide valuable information about the living conditions in a city. A big downside is the fact that buildings and other objects influence measurements, potentially rendering some measurements unusable. During the design phase of the weather station, special attention was given to the choice of suitable and useful sensors.

4.1. Control unit

The system is controlled using an Arduino MKR1010 WiFi unit (shown in figure 4.1), which was already available for the project, powered by an ARM Cortex-M0+ CPU [14]. The device features 13 PWM Pins, 8 digital I/O pins, 7 analog input pints and 1 analog output pin [15]. The size of the unit is 61.5 x 25mm, which makes it easy to incorporate in a small enclosure. The system features a low-power mode that can be used when the system is not doing a measurement or sending data over WiFi. To make sure that the device periodically wakes up from its low-power mode, a real-time clock (RTC) module is required for the system.

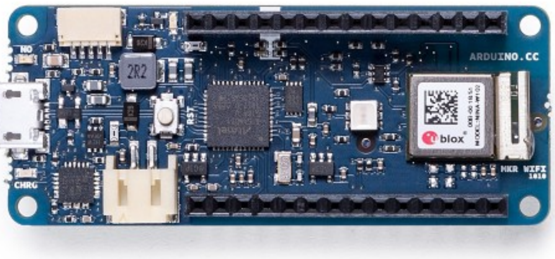


Figure 4.1: Arduino MKR1010 WiFi [15].

4.2. Arduino Oplà IoT carrier

For this project, an Arduino Oplà IoT kit (shown in figure 4.2) was available. The Arduino Oplà IoT carrier contains 5 useful components for the weather station project:

- Micro-SD card reader
- Battery holder
- Humidity sensor
- Temperature sensor
- Pressure sensor

The humidity and temperature sensors should be placed in contact with the outside air. Since this is in conflict with the required placement of the Arduino (which should be sheltered from the weather), only the micro-SD card reader, battery holder and pressure sensor could be used. Based on the data-sheet of the carrier [16], a power consumption of 0.1927 W is found. The measured idle power is 0.0638 W. This relatively high idle power use is due to the attached components that cannot be disconnected from the power source. On the market, a more suiting and more energy efficient micro-SD card reader, battery holder and pressure sensor were found. The power consumption of these components will be discussed in chapter6. For the reasons mentioned above, the weather station will not include the Oplà IoT carrier.

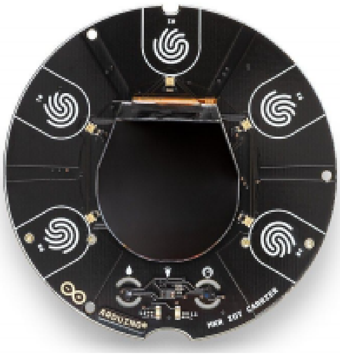


Figure 4.2: Arduino Oplà IoT carrier [16].

4.3. Real-time clock module

Because the system needs to periodically wake up from low-power mode, an real-time clock (RTC) module is required. This device keeps track of time without consuming as much power as the control unit would if it needed to be out of low-power mode. The RTC module used in this project is the DS3231 [17], which was already available for the project.

4.4. Micro SD-card module

Because the system needs to store the measured data, a Micro SD-card module is used. The Micro SD-card module used in this project is a 'CATALEX MicroSD Card Adapter', which was already available for the project. The SanDisk (8 GB) Micro SD-card was also already available for the project.

4.5. Light measurement

A potential use case for measuring light with the weather station is to use this information for the algorithm that decides how often to do measurements (if a lot of light is present, it could perhaps be assumed that the solar panel is generating a lot of power, which would allow the weather station to do frequent measurements). However, the internal electronics of the weather station allow it to measure the current coming from the solar cells. Since this is a far better indicator of available power (based on the light sensor, for instance, the weather station might think the sun is shining, while, in reality, the light is coming from the lantern pole it is attached to), this use case is not an argument to include a light sensor in the weather station. This is a broader problem with doing light measurements in an urban environment. It is very hard to interpret the data, because of the many light sources and obstacles casting shadows in a city.

Another interesting use of a light sensor is the quantification of light pollution. The impact of artificial light on humans and the ecological environment has been a growing concern [18]. A distributed system of light sensors would be attractive to map out and track light pollution over time. However, implementation in an urban weather station would put many constraints on the placement of the weather stations. Since the measured light pollution depends on height, zenith angle and azimuth angle, interpretation of the data would be very hard if the weather stations are not facing the same way and placed on the same height [19]. Putting this as a constraint on the placement of the weather stations would make the whole package less flexible and thus less attractive. Secondly, and most importantly, the problem of artificial light sources and obstacles in an urban environment makes measurements of light pollution nearly impossible, since light intensity varies greatly throughout an urban environment.

For the reasons mentioned above, the weather station does not include a light sensor.

4.6. Temperature measurement

Temperature influences the quality of living in a city. Because of limited urban vegetation (especially in city centers) and climate change, there are concerns that, especially during summer time, some parts of cities might get uncomfortably, or even dangerously, hot. Thermal comfort and heat stress will likely become a critical issue in many urban areas in and outside of the Netherlands [20]. For this reason, a temperature sensor would be an invaluable feature for the weather station. The urban siting does not introduce measurement problems due to buildings and other objects, because it is precisely the influence of these objects on the temperature that is the quantity of interest. A network of weather stations could be distributed over a city to identify ”Heat Islands” information that could be used by policy makers to make the city more habitable. The specific sensor that has been chosen is the AM2315 encased temperature and humidity sensor [21]. The temperature sensor is of the thermistor type described in section 3.1. Because of the small size and limited self-heating this type of sensor is attractive to use for the weather station. The biggest downside of thermistors, the relatively narrow semi-linear range, is not a big problem because of the small temperature range that it needs to measure (outdoor temperature). The AM2315 has a temperature range of -40 to 125 °C, with a typical accuracy of 0.1 °C. The protective casing of the AM2315 provides extra protection against pollutants and moisture.

4.7. Humidity measurement

An important meteorological factor concerning health and well-being is the relative humidity of the air. Global temperature and humidity are expected to rise in the 21st century, decreasing subjective well-being and increasing heat stroke incidences [22]. Since humans cannot lose as much heat in a humid environment, high temperatures become even more dangerous. To assess urban living quality, measuring humidity is an invaluable addition to measuring temperature. Accurate capacitive humidity sensors are widely available for low prices and their quiescent and active power use is very low. For these reasons, an air humidity sensor would be a good addition to the weather station without putting strain on the trade-off requirements (because of its low power use). As described in section 4.6, the sensor that has been chosen is the AM2315 encased temperature and humidity sensor [21]. The humidity sensor is of the capacitive type, which is more attractive to use in a weather station than the resistive type that is vulnerable to condensation (as described in section 3.2). The AM2315 has a humidity range of 0 to 100%, with a typical accuracy of 2% at 25 °C. The protective casing of the AM2315 provides extra protection against pollutants and moisture.

4.8. Wind measurement

The urban choice of siting makes accurate wind speed and direction measurements almost impossible. Since the device is designed to be attached to the south side of an object, wind coming from the north will be blocked. Furthermore, buildings and other objects on all other sides will significantly influence the measurements. According to the World Meteorological Organisation, wakes can easily downwind to 12 or 15 times the obstacle height and the requirement of 10 obstruction heights is an absolute minimum [1]. This means that an average house with a height of 7 meter requires the weather station to be at least 70 meter away from it. This is an impossible requirement for a weather station that is used in an urban environment. It could be argued that the influence of buildings and other objects on the wind is the variable of interest, but it would be nearly impossible to interpret the data from a network of weather stations throughout a city, let alone use this data for policy decisions or scientific research. The only way to obtain reasonably accurate measurements is to require the weather station to be placed on top of high buildings. This brings up multiple problems, the first being that the sensor needs to be placed at least one building width higher than the building to avoid turbulence distorting the measurement too much [23]. The second problem is that many other measurements are a lot less useful at this height, like temperature and air pollution (since most people spend most time outdoors at ground level). Since the steps required to make the weather station suitable for wind measurement defeat many of its purposes, the weather station will not measure wind speed and direction.

4.9. Rain measurement

According to the World Meteorological Organization, objects should not be closer to the rain gauge than a distance of twice their height above the rain gauge orifice [1]. If the example of the 7 meter high house is taken again, this would require the weather station to be at least 14 meter away from the house. This would make the location requirements for the weather station overly restrictive for an urban setting. Since the weather station will probably be attached to the south side of a wall or pole, the weather station would not be able to measure rain when (strong) wind is coming from the north if the weather station is not attached to the top of this wall or pole. The rain measurement feature suffers from a lot of the same downsides as the wind measurement feature. For this reason, the weather station will not measure rain.

4.10. Air pressure measurement

Air pressure can be measured from the inside of an enclosed object, which puts no additional constraints on the design of the enclosure or siting of the weather station. Furthermore, electronic air pressure sensors are very power efficient, which means that the inclusion of such a sensor does not significantly impact the power management of the weather station. Atmospheric pressure data can be useful for weather prediction or for studying correlation with other weather parameters. There are strong indications that changes in air pressure have a strong impact on atmospheric pollution [24], which means that measuring and analysing atmospheric pressure could yield information about urban living quality. The sensor that has been chosen for this project is the BMP280 [25], because the sensor has a temperature operating range of -40 to 85 °C, an absolute accuracy of 1.5 hPa and a low power use (as described in 6.1). Because the sensor is of the piezoresistive type (as described in section 3.3), it includes an internal temperature sensor to allow the required temperature compensation.

4.11. Noise measurement

The amount of noise within an urban environment impacts living quality and can be an indication of traffic intensity. Traffic intensity has a correlation with certain pollutants, like NO_2 and black smoke [26]. Since electric microphones are widely available and affordable, it can be a worthwhile addition to the weather station. Since the microphone will most likely be located within the enclosure (to protect it from moisture), special care should be taken when the measurements are interpreted. Frequency components that resonate with the enclosure, or low-frequency waves that travel through the wall or pole the weather station is attached to, will most likely be over-represented. The sensor used in this project is a CMA-4544PF-W microphone [27] with an MAX4466 [28] amplifier, which was already available for the project.. The microphone is of the capacitive type, because the small size and affordable price, as described in section 3.4. Since the main goal is to get an overview of noise pollution, instead of recording high-fidelity sound, an exactly flat-band response is not required.

4.12. Gas measurement

The effects of air pollution on public health is a growing concern. Certain gasses, like sulphur dioxide and nitric dioxide, can induce airway problems (especially in asthmatic or allergic/hypersensitive people) [29]. For this reason, a gas sensor would be a good addition to the weather station with regards to measuring urban living quality. However, affordable semiconductor gas detectors require a very high temperature to measure gasses (for which they use a resistor) and have long settling times. This makes a semiconductor gas sensor a big constraint on the energy budget. Other types of gas sensors are too expensive or impractical to implement in a small, power efficient weather station. Because of the high power use, the weather station does not feature a gas sensor.

The MQ-135 NOx sensor, widely available in Europe, needs a warm-up time of 60-120 seconds before an accurate measurement can be obtained. During this time, the heater consumes around 750 mW [30]. More power efficient NOx sensors are available (like the MiCS-6814 [31]), but due to the long lead times they were unfeasible for implementation in this project. Furthermore, the MiCS-6814 is one order of magnitude more expensive than the MQ-135.

It should be noted that the feasibility of a gas sensor was reconsidered after measurements of the power consumption of the system and power generation of the solar panel were obtained. A detailed discussion of the options of gas sensors in future designs is given in chapter 9.

4.13. Particulate matter measurement

One of the most harmful air pollutants is particulate matter (PM). This name stands for a large group of small airborne particles that can enter the lungs and/or bloodstream of humans. Chronic exposure to high PM levels can induce a number of adverse respiratory and cardiovascular health effects. This effect seems to be most pronounced with relatively small PM particles, hypothetically because of increased acidity and ability to penetrate into the lower airways [32]. Previous research shows that there are cheap solutions for PM measurement commercially available [33]. PM sensors do introduce a few constraints on the weather station. They consume a substantial amount of power when active (in the 250 mW range), mostly because of the intake fan required to measure air PM levels [13]. Furthermore, once the fan starts taking in air, it can take some time before an accurate measurement can be done. Since the power use is not nearly as high as the power use of a gas sensor, a PM sensor could (if correctly implemented with power-saving algorithms) be implemented in the weather station. The PM sensor chosen for this project is the PMS5003 [13]. This sensor is an attractive choice because of its low price of around 25 euro, its maximum measuring error of around 10%, and its wide range of operating conditions (temperatures between -10 and 60 °C and relative humidity between 0 and 99%). As described in section 3.6, this sensor uses laser scattering to estimate the presence of particles of a certain size. Because the PMS5003 requires a 5 V input, a DC/DC converter is required to increase the voltage from the battery. The TPS61090 boost converter is used for this task, because of its high efficiency (96%) and low price [34].

5 Implementation

5.1. Hardware

The different components are connected as depicted in figure 5.1. All components are directly connected to the 3V3 pin (VCC) of the Arduino, except for the PM sensor, which is connected to the boost converter. Besides the default pins for the three different communication protocols (IIC, SPI and Serial), one analog and 3 digital pins have been used. The analog pin is used for reading out the analog signal coming from the microphone, which is ranging from 0 to 3.3 V. One of the digital pins is used for reading the interrupt coming from the RTC, another digital pin is used to write to the enable pin of the boost converter. The third pin is used as chip select pin for the SPI communication of the SD-card module. Pull-up or pull-down resistors are not needed because they are already available inside the Arduino. The breadboard test setup can be seen in figure 5.2.

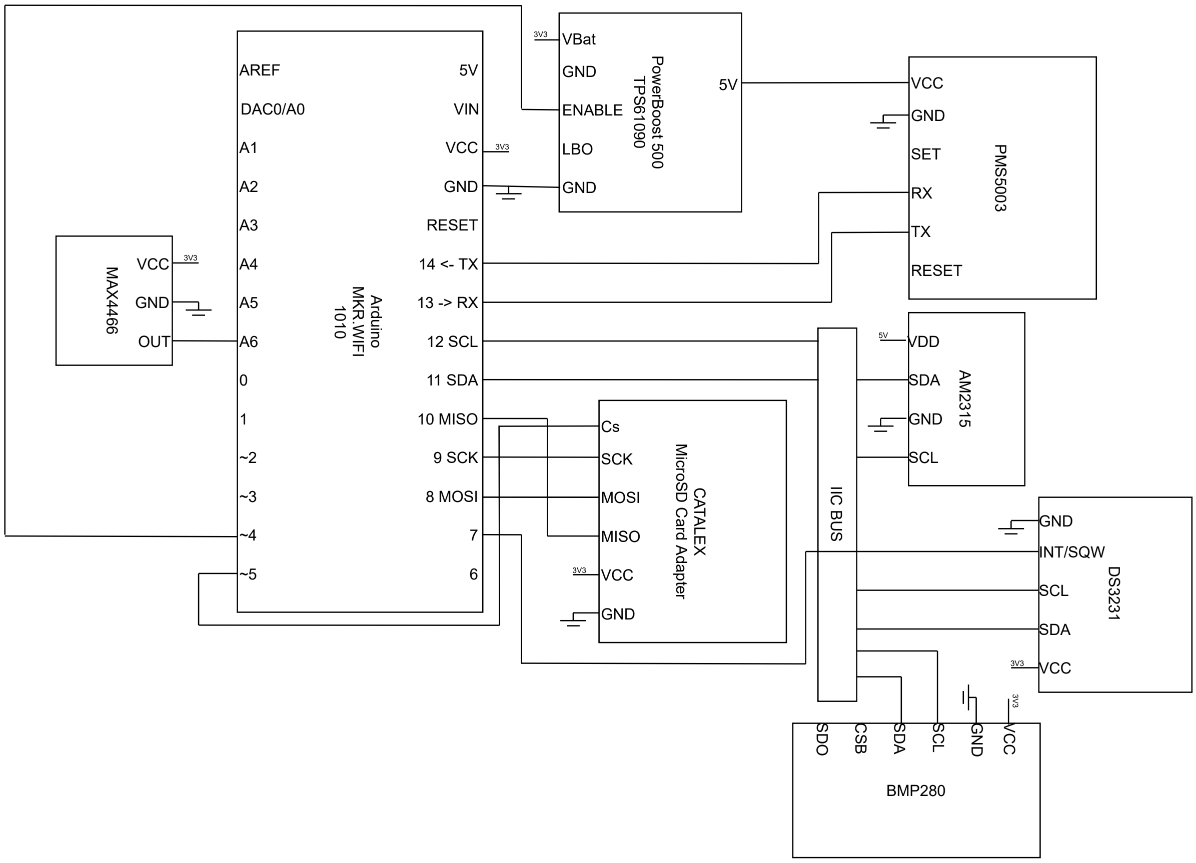


Figure 5.1: Block diagram of the main controller attached to the components.

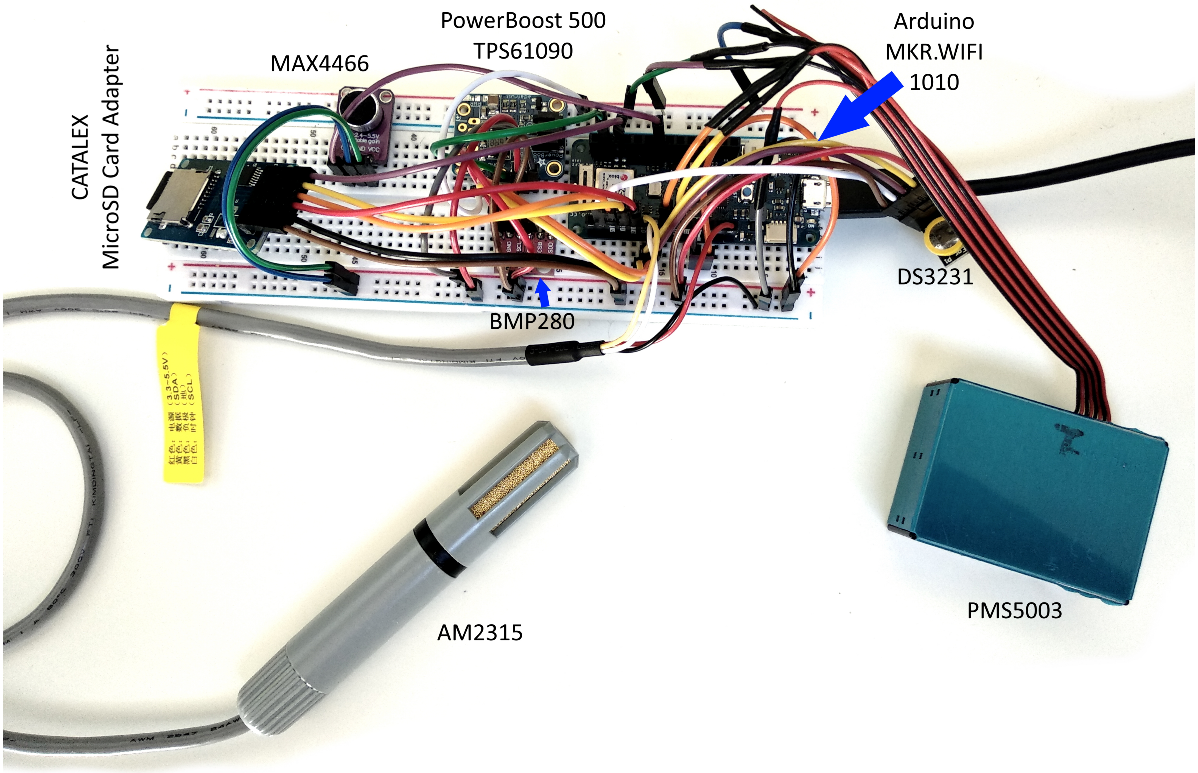


Figure 5.2: Breadboard setup of the main controller attached to the components.

5.2. Software

The main controller (Arduino MKR1010 WiFi [15]) has been programmed in C++ and compiled and uploaded with the Arduino IDE. The list of libraries used is (without their dependencies):

- Arduino `https://github.com/arduino/ArduinoCore-avr/tree/master/cores/arduino`
- Wire `https://github.com/arduino/ArduinoCore-avr/tree/master/libraries/Wire`
- SPI `https://github.com/arduino/ArduinoCore-avr/tree/master/libraries/SPI`
- ArduinoLowPower `https://github.com/arduino-libraries/ArduinoLowPower`
- SD `https://github.com/arduino-libraries/SD`
- WiFiNINA `https://github.com/arduino-libraries/WiFiNINA`
- Adafruit BMP280 `https://github.com/adafruit/Adafruit_BMP280_Library`
- Adafruit AM2315 `https://github.com/adafruit/Adafruit_AM2315`
- PMserial `https://github.com/avaldebe/PMserial`
- DS3231 `https://github.com/jarzebski/Arduino-DS3231`

The main function of the code looks roughly like this semi-pseudo code:

```
initialize everything
```

```
loop:
    if allowed to measure:
        date_time.read()

        Temperature_humidity.read()

        Pressure.wake()
        Pressure.read()
        Pressure.sleep()

        Microphone.read()

        if time to measure PM:
            if PM is off:
                PM.turn_on()
            PM.read()
        else:
            if PM is on:
                PM.turn_off()

        save_Measurements_in_files()
    if allowed to send:
        sendFile()

    setAlarm(5 seconds)
    deepSleep()
```

The conditions ”if allowed to measure:”, ”if time to measure PM” and ”if allowed to send:” are determined by the power algorithm in use. The function save_Measurements_in_files() saves the measured data in two files. One file is a CVS ASCII file of the current day, the other is a file that contains all binary data since the last time the data was send via WiFi. Once it is time to send the data, this file is loaded into a buffer and is transmitted to the web server in the data section of the http message. Afterwards, this file is emptied/deleted. The binary data of one measurement cycle is 36 bytes as can be seen in table 5.1. If a particular parameter is not measured, the value 0 is given, which will not give rise to confusion since there is no case in which there will actually be a value of 0. The implementation of the web server will not be elaborated here. In short, a Linux Apache PHP/MySQL web server is used with a client interface generated with the help of the JavaScript Library called APEXCHARTS (https://apexcharts.com/).

unix timestamp	Int (4 Bytes)
temperature_outside (AM2315)	Float (4 Bytes)
humidity	Float (4 Bytes)
pressure	Float (4 Bytes)
temperature_inside (BMP280)	Float (4 Bytes)
acoustic noise	Int (4 Bytes)
PM >0.3	Short Int (2 Bytes)
PM >0.5	Short Int (2 Bytes)
PM >1.0	Short Int (2 Bytes)
PM >2.5	Short Int (2 Bytes)
PM >5.0	Short Int (2 Bytes)
PM >10.0	Short Int (2 Bytes)

Table 5.1

6 Power consumption

6.1. Power measurements

When all chosen sensors were obtained, power measurements were done for the different sensors in different modes of operation.

The Arduino MKR1010 has a sleep mode in which a consumption of 53.5 mW is observed. By removing the power-on LED, the consumption is drastically reduced to 14.5 mW. When the Arduino is not in sleep mode, a power consumption of 50 mW is observed.

The air pressure sensor consumes 1.65 mW while measuring and less than 1 μW in sleep mode. The time required for a measurement is lower than 4 ms.

The TPS61090 boost converter has a power consumption of 27.36 mW. After removing the power-on LED, the consumption is reduced to just 2.11 mW. The TPS61090 can also be disabled by the Arduino, which reduces the power use to less than 1 μW.

The microphone module (including the MAX4466 amplifier) has a power consumption of 0.76 mW and no dedicated low-power mode. Since sound pressure varies drastically with time in an urban environment, a 50 ms long measurement is done and the average value is taken.

The real-time clock module has a constant power use of 0.66 mW.

The temperature/humidity sensor has power use of 4.95 mW during measurement and a power use of 0.13 mW when no measurement is being done. The time required for a measurement is a fraction of a second (about 14 ms).

The PM sensor has a quiescent power use of 30 mW (which is not an issue for power consumption in sleep mode, since the attached DC/DC converter can be turned off). When the PM sensor is turned on, it starts off with increased fan speed to refresh the air in the internal chamber, consuming 310 mW. After the first 20 seconds, the PM sensor consumes 221 mW. In order to decide how long a measurement needs to be to obtain stable PM data, measurements were done after a cold start-up, the results of which are shown in figure 6.1. Since the sensor shows some fluctuations without clearly converging to a stable point, the chosen procedure is to start measuring after 30 seconds and take the average of the measurements done between 30 seconds and 60 seconds. In this way, a measurement can be obtained that is less disturbed by short temporal variations. This means that the sensor is measuring for 60 seconds with an average power consumption of 250.7 mW.

Since the amount of data to be sent over WiFi is so small, WiFi communications take a few seconds at most. This means that despite the power use of around 600 mW during transmission, the power use of daily transmissions is negligible [35].

6.2. Generation and battery

The solar panel has a surface area of 0.0625 m^2 and an efficiency of around 15%. The maximum power point tracker has an efficiency of at least 75%. This means that about 10.5% of solar power incident on the solar panel is converted to electrical power. It should be noted that the efficiency of the solar panel decreases when the temperature of the panel rises. For this weather station, however, the winter period is the critical period where there is a risk of running out of battery charge, which makes this temperature dependency less relevant for design decisions.

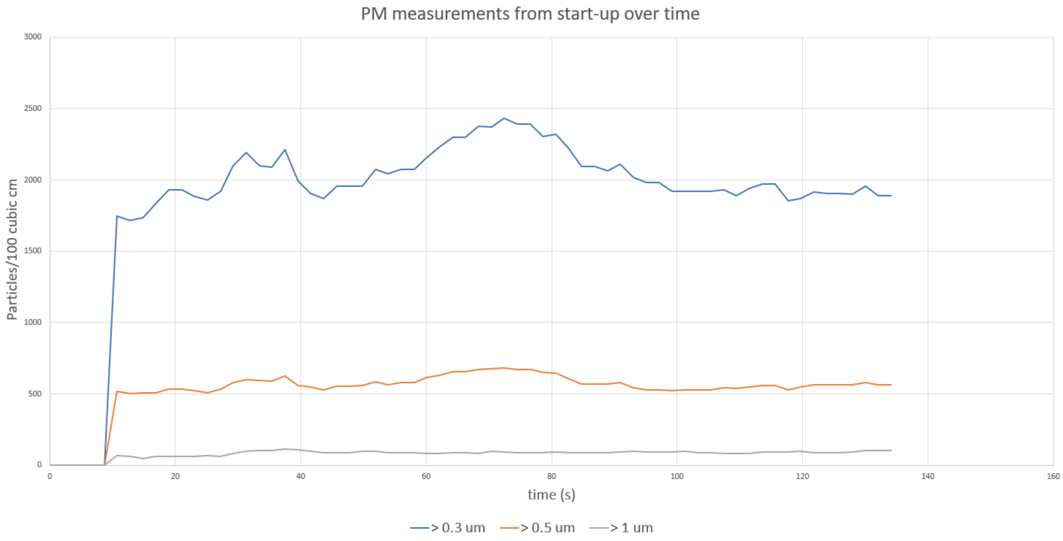


Figure 6.1: Graph of PM sensor measurements over time.

The weather station contains two rechargeable 10.4 Wh batteries of the Lithium-ion type. The allowed depth of discharge is 20%, which leaves 8.3 Wh of effective capacity. The self-discharge rate is estimated to be 3% of the full battery charge per month, which is a typical value for a Li-ion battery [36].

6.3. Mode of operation

The weather station will be in sleep mode for most of the time. Periodically, the RTC module wakes up the Arduino to initiate a measurement. A measurement without PM sensor takes 2 seconds, after which the system goes into sleep mode again. When a PM measurement is done, PM sensor is turned on for 60 seconds. At least on a weekly basis, the Arduino sends all collected data to a web server over WiFi. In order to safe power during times of low generation, an algorithm can be implemented that changes the frequency of measurement based on the available power (this will be shown in the next section).

6.4. Simulations

To determine the number of measurements that the weather station can do with the available power without running out of battery charge, simulations are done using generation data from the solar panel subgroup that is based on irradiance data from the Dutch city of Delft between 2012 and 2016 from the European Commission [37]. A distinction is made between full measurements and short measurements. Full measurements include a PM measurement and take 1 minute to perform, while short measurements do not include a PM measurement and take 2 seconds to perform. A full measurement consumes about 222 times as much energy as a short measurement. When the battery drops below 10%, the weather station turns off completely. The battery charge cannot go higher than 90%. For the first simulation, 15 full measurements and 225 short measurements are performed per hour. This means that the noise, temperature and humidity sensors take a measurement every 15 seconds, while the PM sensor takes a measurement every 4 minutes. The obtained data is sent over Wi-Fi daily and the weather station starts with 50% battery charge. When the battery charge is plotted against time between 2012 and 2016, the graph shown in figure 6.2 is obtained. It can be seen that the battery charge in the graph is above 80% most of the time. The battery charge shows significant drops only for short times during the winter months in case of a few weeks with barely any irradiance. A less cluttered 24-hour average battery charge plot can be seen in the appendix in figure A.1.

If the number of PM measurements per hour is slightly reduced, the weather station can survive on 1 battery. A simulation with 10 full measurements and 230 short measurements per hour is shown in figure 6.3 (the 24-hour average plot is shown in figure A.2).

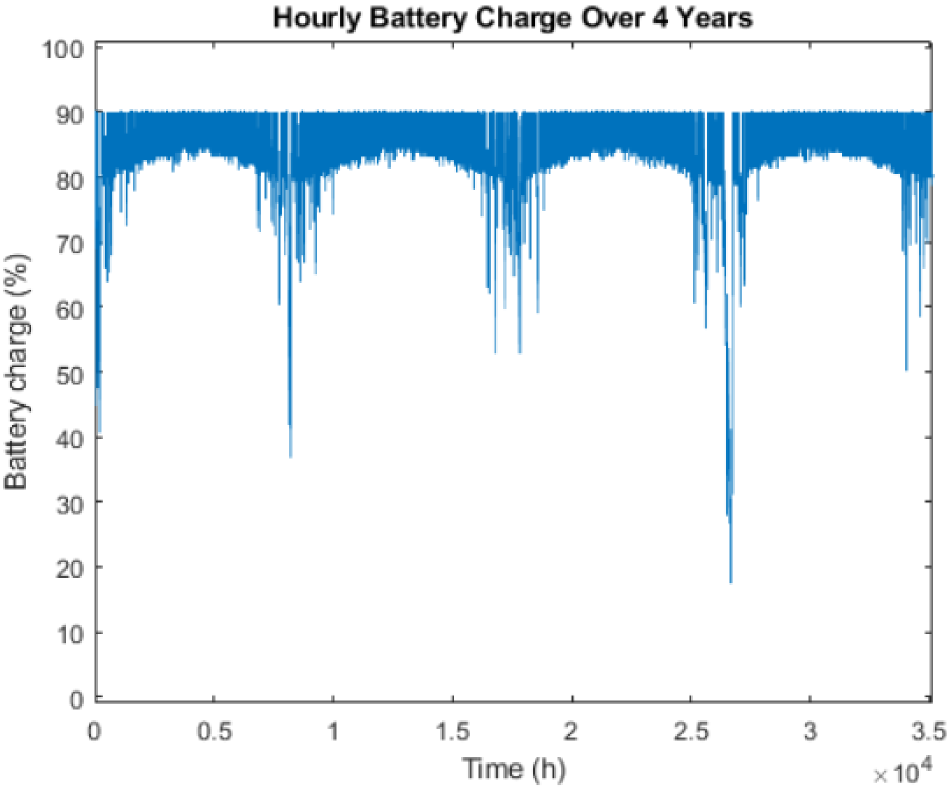


Figure 6.2: Simulation with 2 batteries, 225 short measurements and 15 PM measurements.

In order to make sure the weather station survives the winter months, an algorithm can be implemented that saves power in times of under-generation. One way to implement this is to make the measuring frequency dependent on the current battery charge. Since measurements that include the PM sensor consume about 222 times as much energy as short measurements, the frequency of full measurements determines the (biggest part of the) consumption. If a linear dependency is chosen, the graph of such an algorithm could look like the one shown in figure 6.4. This algorithm includes 4 parameters. A lower battery charge threshold is chosen below which the weather station should stop performing PM measurements (while continuing the short measurements) and an upper battery charge threshold is chosen above which the weather station should perform the maximum desirable number of measurements. Below 10% battery charge the weather station should stop measuring all-together (also no temperature, humidity, air pressure and noise measurements). The other two parameters are the minimum number of measurements (to be performed just above the lower threshold) and the maximum number of measurements (to be performed above the upper threshold). The maximum number of measurements can be configured to correspond to continuous sensing (which could be desirable to use excess power in times of over-generation). Furthermore, it can be decided that the weather station sends its data over WiFi more frequently in order to make use of excess power. It should be noted that the ”optimal” algorithm depends on the specifics of the implementation. If, for example, the weather station is used in a country with a higher generation in the winter (closer to the equator), the measuring frequency at lower battery charge can be higher and the algorithm thus less conservative. It is also not true that a linear algorithm is the optimal algorithm. If a relevant performance metric can be defined (such as number of measurements per year), different models can be fitted and evaluated. In order to optimise the power use and measuring frequency, generation data from the last days or years can be used to more accurately predict the available power in the coming weeks. Such an optimisation could become more important if the weather station has a smaller power budget due to a smaller solar panel and/or battery.

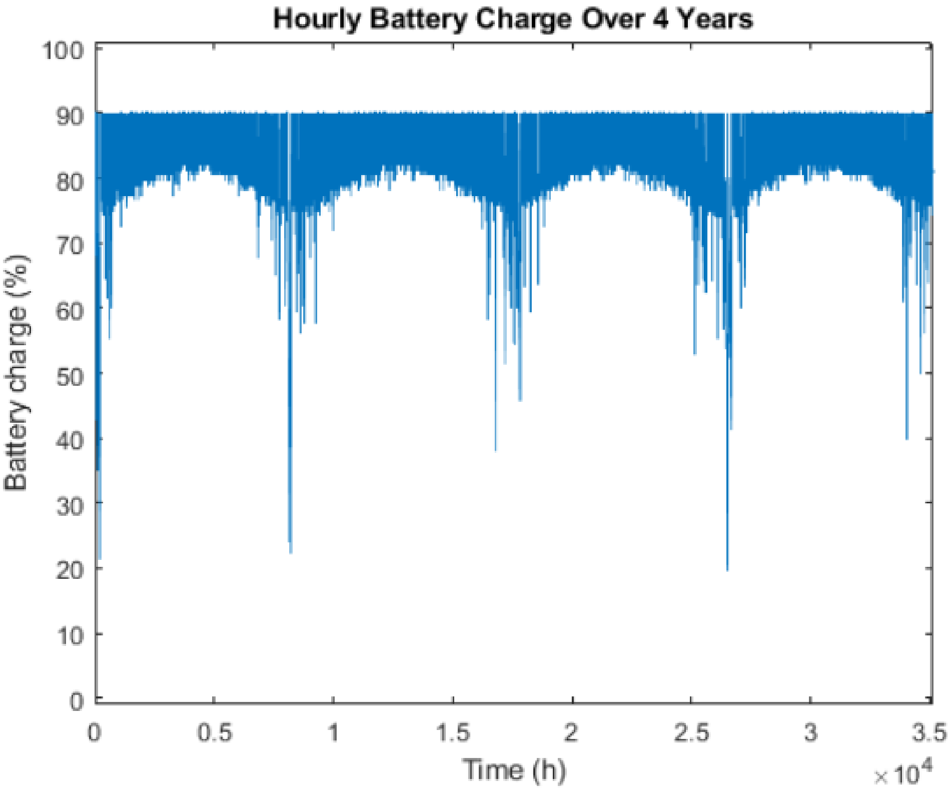


Figure 6.3: Simulation with 1 battery, 230 short measurements and 10 PM measurements.

In the next simulation, the linear algorithm described above is implemented with a lower threshold of 30% and an upper threshold of 85%. Above 85% battery charge, the weather station continuously measures and sends data over WiFi (never going into sleep mode). When the battery charge drops below 85%, the frequency of WiFi transmissions reduces to one transmission per day. When the battery charge is slightly above the lower threshold, the PM sensor is measuring 10% of the time (a relative frequency of 0.1 with respect to the maximum measuring frequency). The result of this simulation is shown in figure 6.5 (the 24-hour average graph is shown in figure A.3). The corresponding relative frequency of PM measurements can be seen in figure 6.6 and the associated power consumption of the system is shown in figure 6.7. It can be seen that the application of the algorithm greatly increased the measuring frequency of the system, since the measuring frequency of the first simulation that was shown in figure 6.2 corresponds to a relative frequency of 0.25 in figure 6.7.

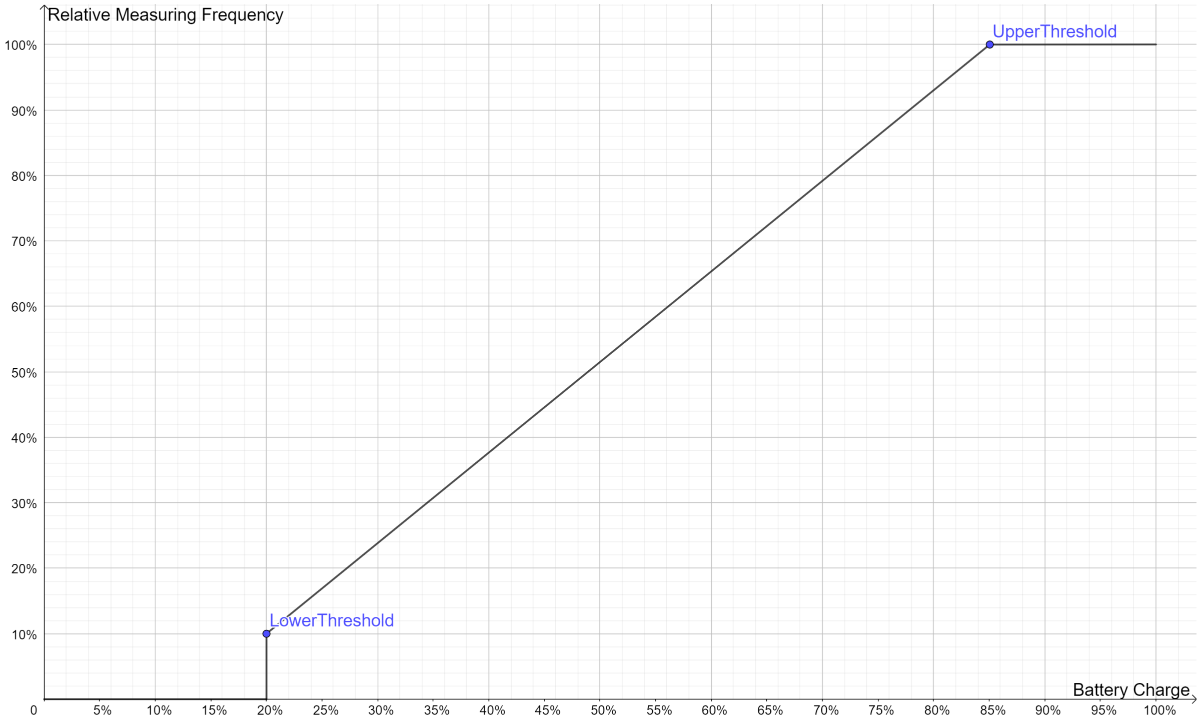


Figure 6.4: Linear algorithm that bases PM measuring frequency on current battery charge.

Since the idle power consumption of the system is lower than hypothesised because of the removal of LED's and the implementation of the algorithm, the addition of a gas sensor, such as the MQ-135, can be considered. The MQ-135 needs at least 1 minute to perform a measurement, while consuming around 750 mW [30]. This quadruples the power consumption of a full (PM + gas) measurement. When a simulation is done with the same settings as in figure 6.5 (but with addition of the gas sensor), the graph shown in figure 6.8 is obtained (the 24-hour average graph is shown in figure A.4). The corresponding relative measuring frequency shown in figure 6.9. It can be seen that the weather station has an average relative frequency of around 40%, which means that the system is doing full measurements 40% of the time. In a 96-hour close-up of January, shown in figure 6.10, the variations between day and night can clearly be seen. The variations in measuring frequency between day and night during 96 hours in January can be seen in figure 6.11. If PM and gas measurements are desired during the hours in which the graph in figure 6.11 drops to zero, the algorithm can be adjusted to be more conservative during the day, which leaves more charge for the night. In June, the relative measuring frequency does not drop to 0% during the night, as can be seen in figure 6.12. It can be concluded that the addition of a gas sensor is feasible if the algorithm is applied.

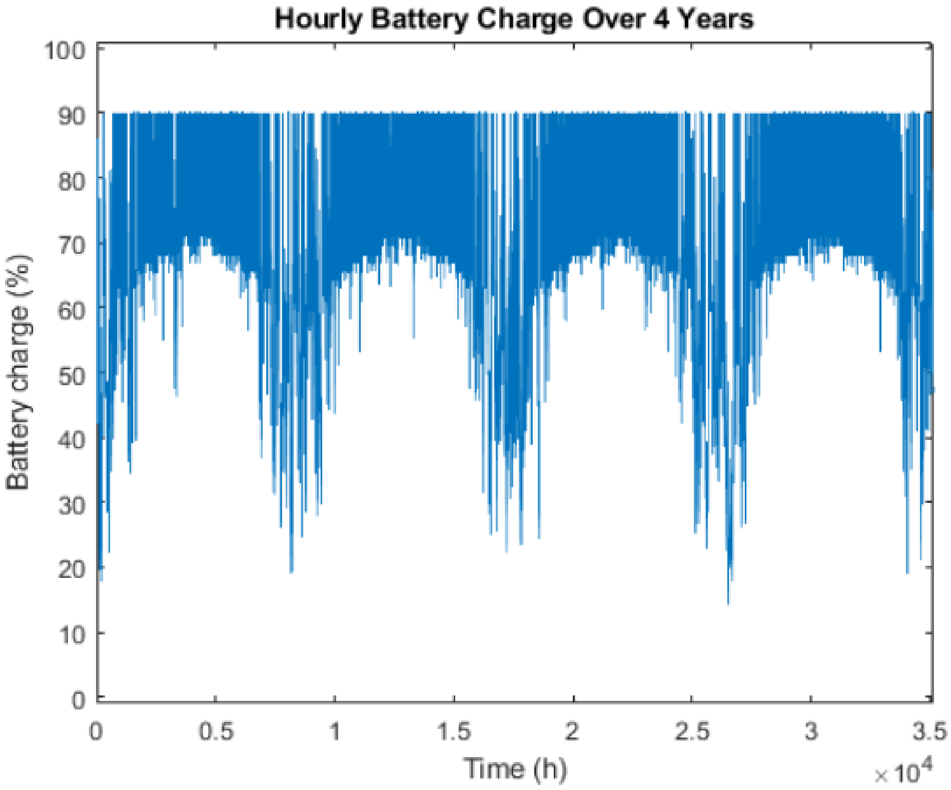


Figure 6.5: Simulation with 2 batteries, continuous measurements at full charge and algorithm enabled.

The increased power budget can also be used to reduce the size of the weather station by using a smaller PV panel or to remove a battery while still being able to do (relatively) frequent measurements. Figure 6.13 shows the result of the same simulation done in figure 6.8, but this time with only 1 battery and a PV panel with only one fourth of the surface area: 155 cm^2(the 24-hour average graph is shown in figure A.5). The relative measuring frequency of the PM and gas sensors is shown in figure 6.14. It can be seen that such an implementation is feasible, since the relative measuring frequency of the PM and gas sensors oscillates around 0.2 (12 measurements per minute). It should be noted that the weather station still measures temperature, humidity, noise and air pressure when the gas and PM sensors are turned off.

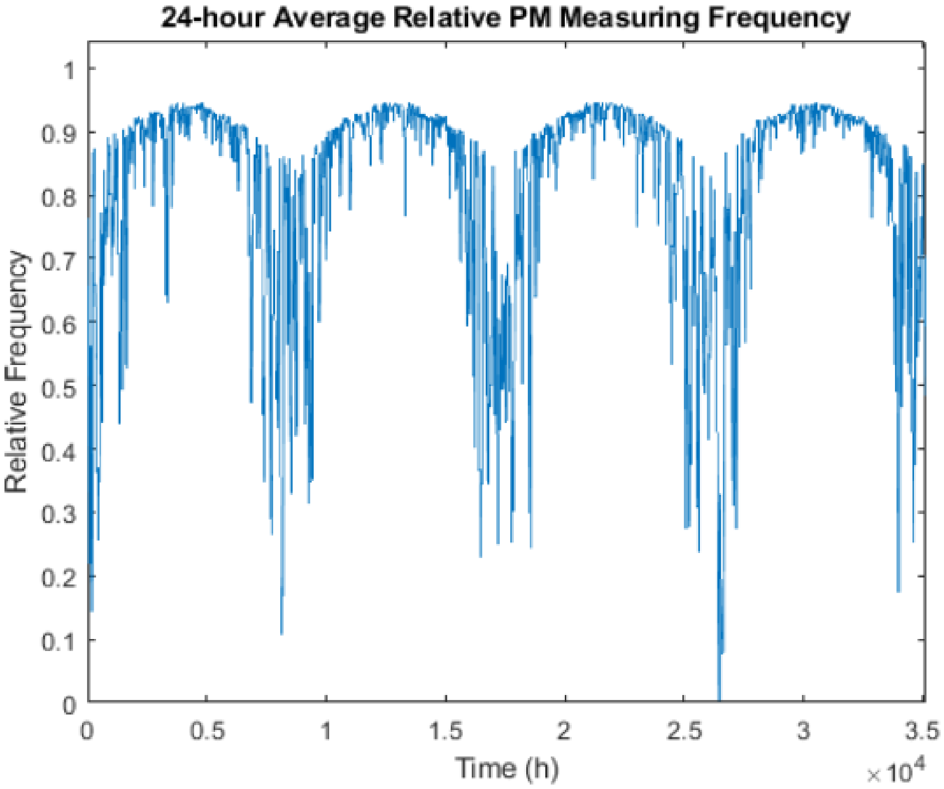


Figure 6.6: Relative PM measuring frequency of simulation with 2 batteries, continuous measurements at full charge and algorithm enabled.

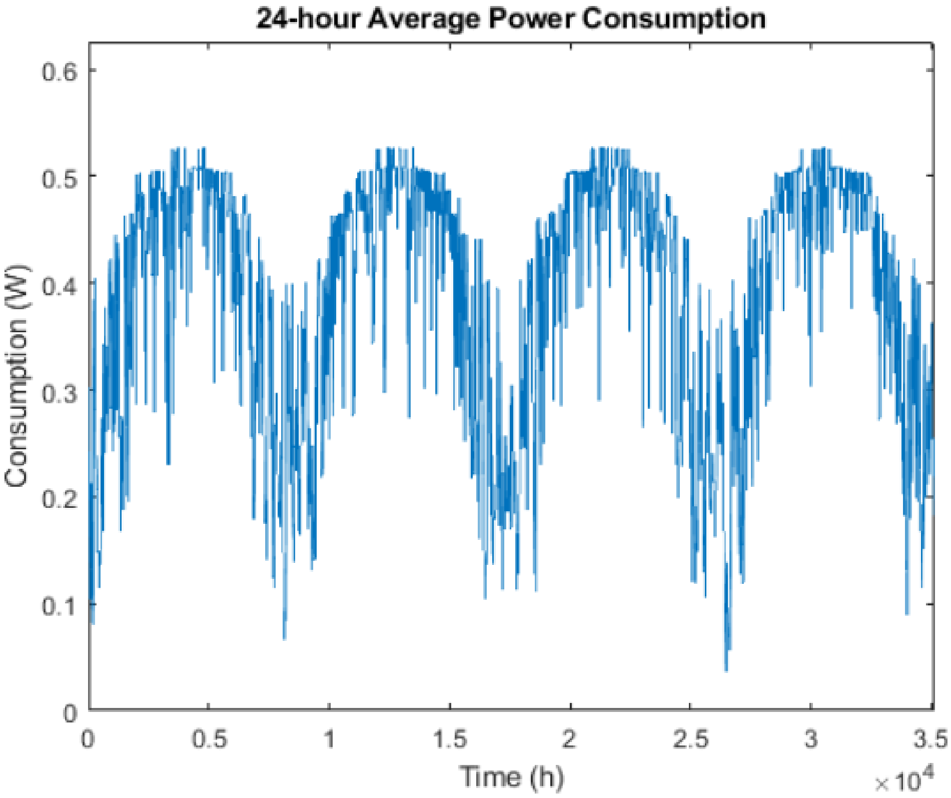


Figure 6.7: Power consumption with 2 batteries, continuous measurements at full charge and algorithm enabled.

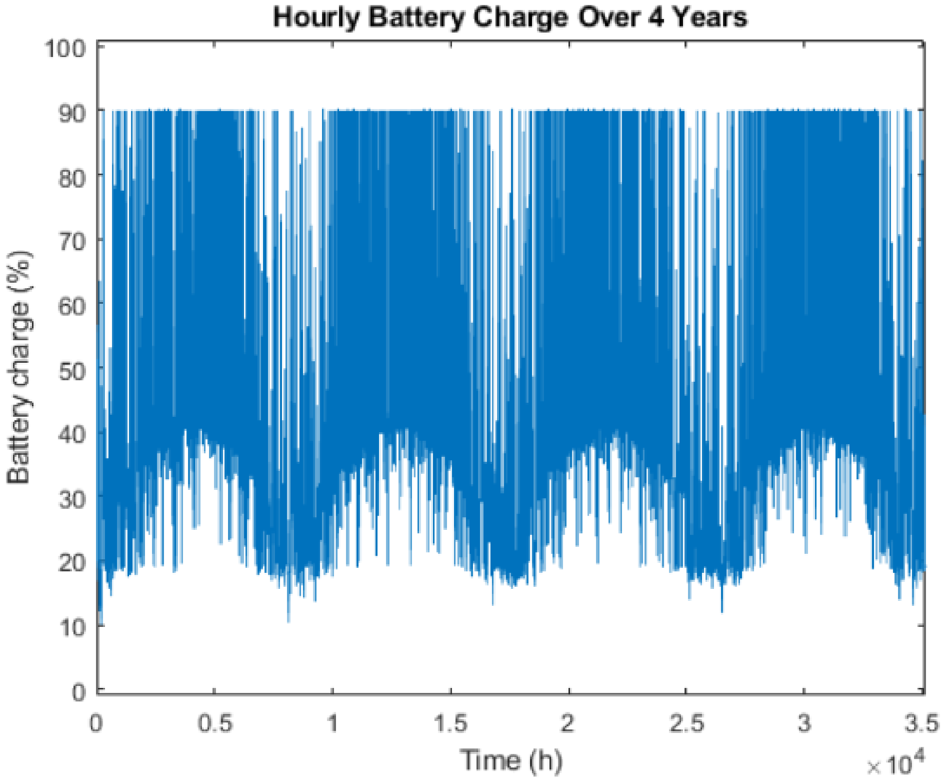


Figure 6.8: Simulation with 2 batteries, continuous measurements (including gas sensor) at full charge and algorithm enabled.

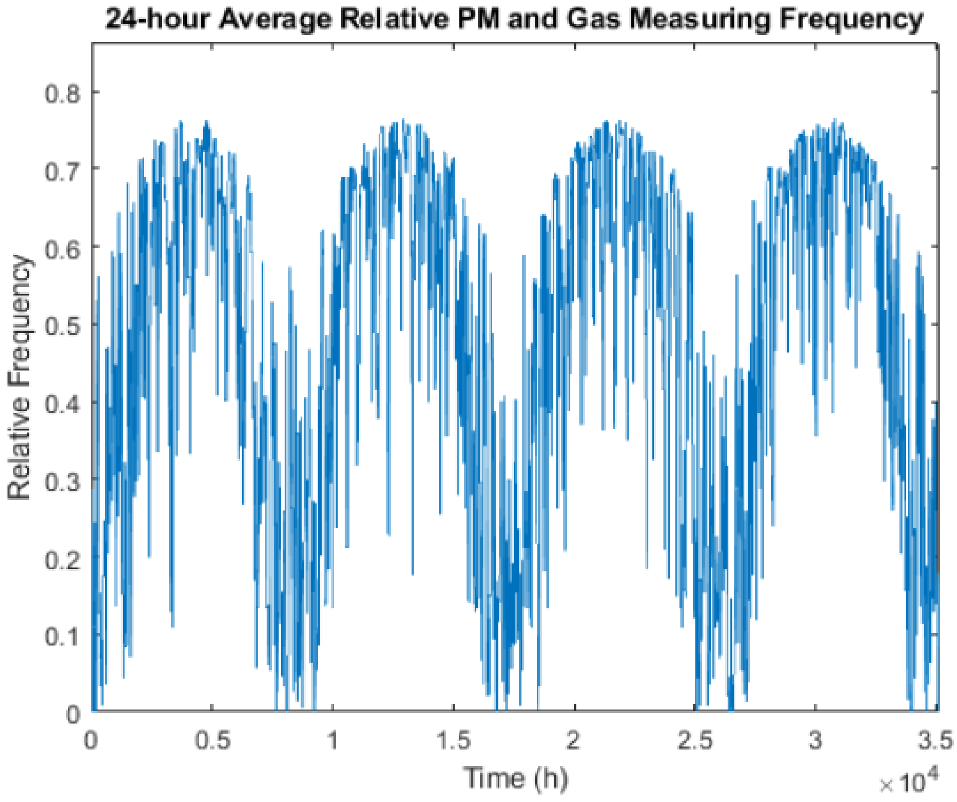


Figure 6.9: Relative PM and gas measuring frequency of simulation with 2 batteries, continuous measurements (including gas sensor) at full charge and algorithm enabled.

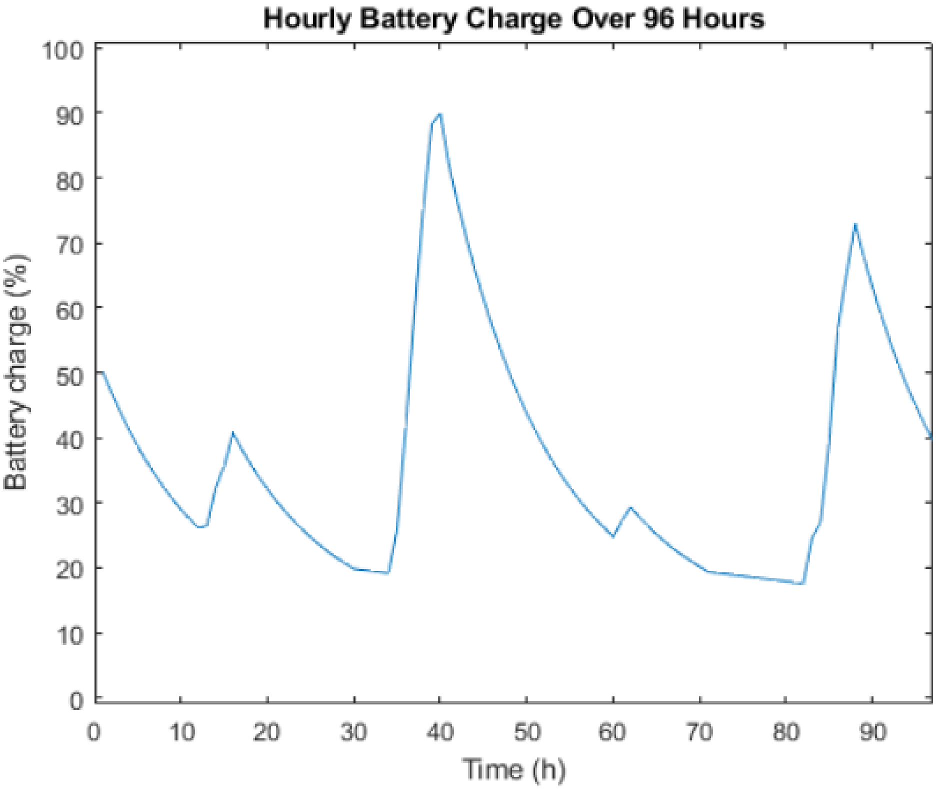


Figure 6.10: Simulation with 2 batteries, continuous measurements (including gas sensor) at full charge and algorithm enabled (96-hour closeup in Januari).

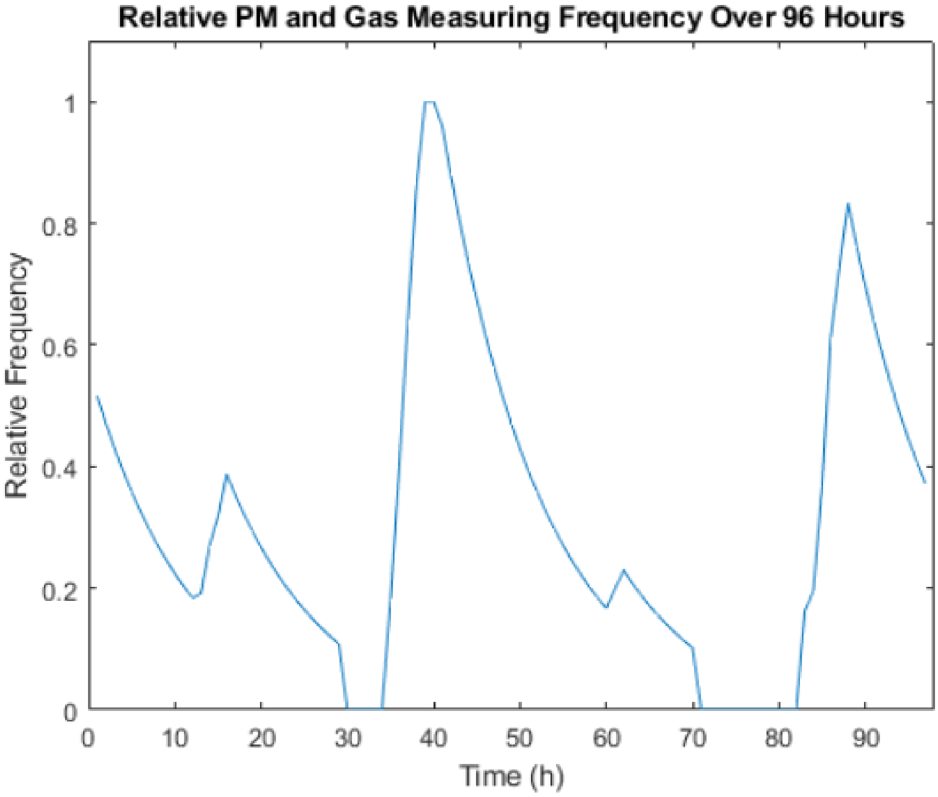


Figure 6.11: Relative measuring frequency of the gas en PM sensors during four days in Januari.

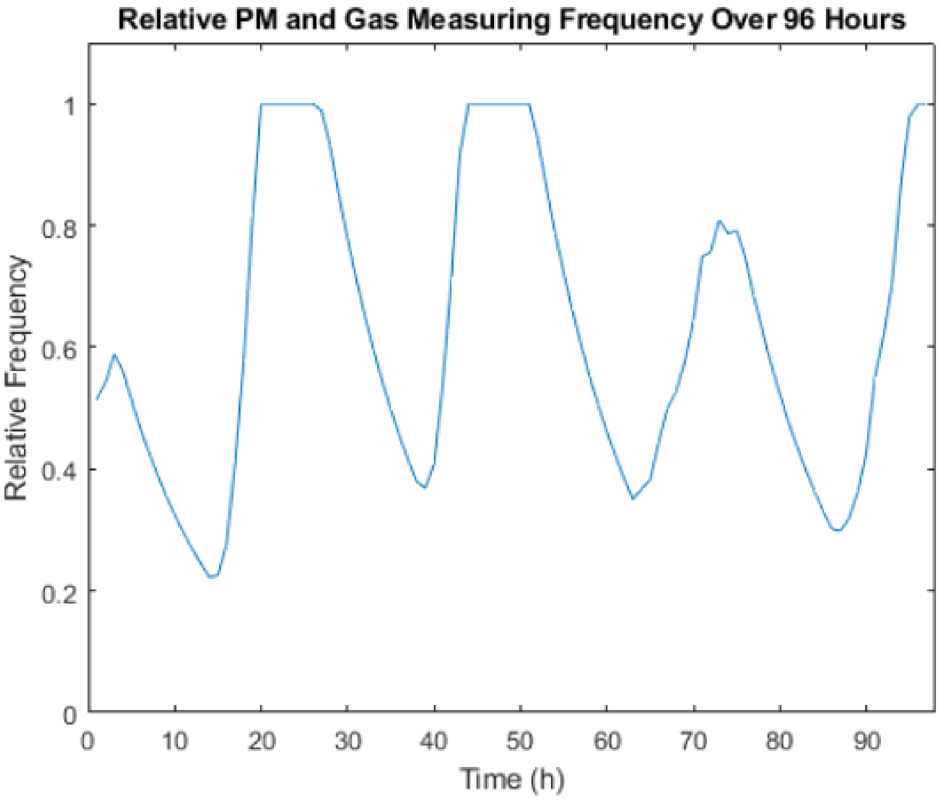


Figure 6.12: Relative measuring frequency of the gas en PM sensors during four days in June.

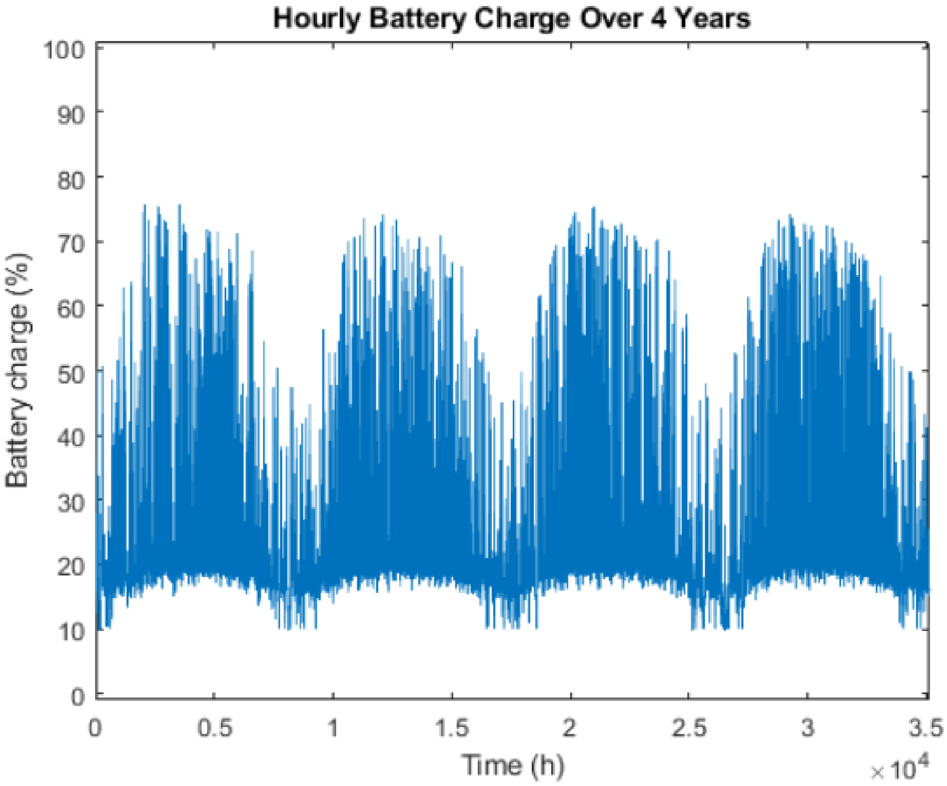


Figure 6.13: Simulation with 1 battery and 1/4th the PV area, continuous measurements (including gas sensor) at full charge and algorithm enabled.

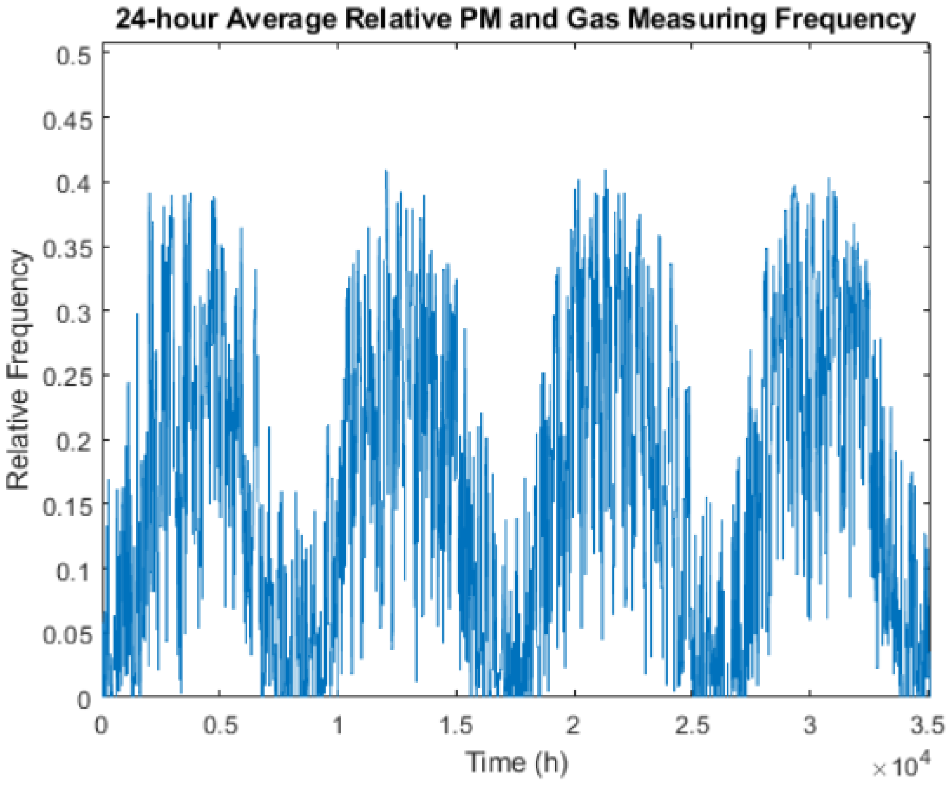


Figure 6.14: Relative PM measuring frequency of simulation with 1 battery and 1/4th the PV area, continuous measurements (including gas sensor) at full charge and algorithm enabled.

7 Results

One of the requirements for this project was to be able to send data to a web server. Additionally, a client web page was created to visualize the received data, which is depicted in figure 7.1. After visualisation, it seemed that the acoustic noise sensor and the temperature/humidity sensor were distorted/unreliable when the PM-sensor was turned on. This is probably due to the dropped VCC which also becomes noisy due to the fan of the PM sensor. The VCC voltage fluctuates from 2.95 to 3.17 volt while the PM sensor is turned on. In figure 7.1, the acoustic noise constantly ranges between 98% and 100% due to this error. For the making of this screenshot, the temperature/humidity sensor was connected to the 5V pin of the Arduino to avoid the distorted signal, but this pin is not available in the real weather station. Thoughts on ways to avoid this problem are shared in chapter 9.

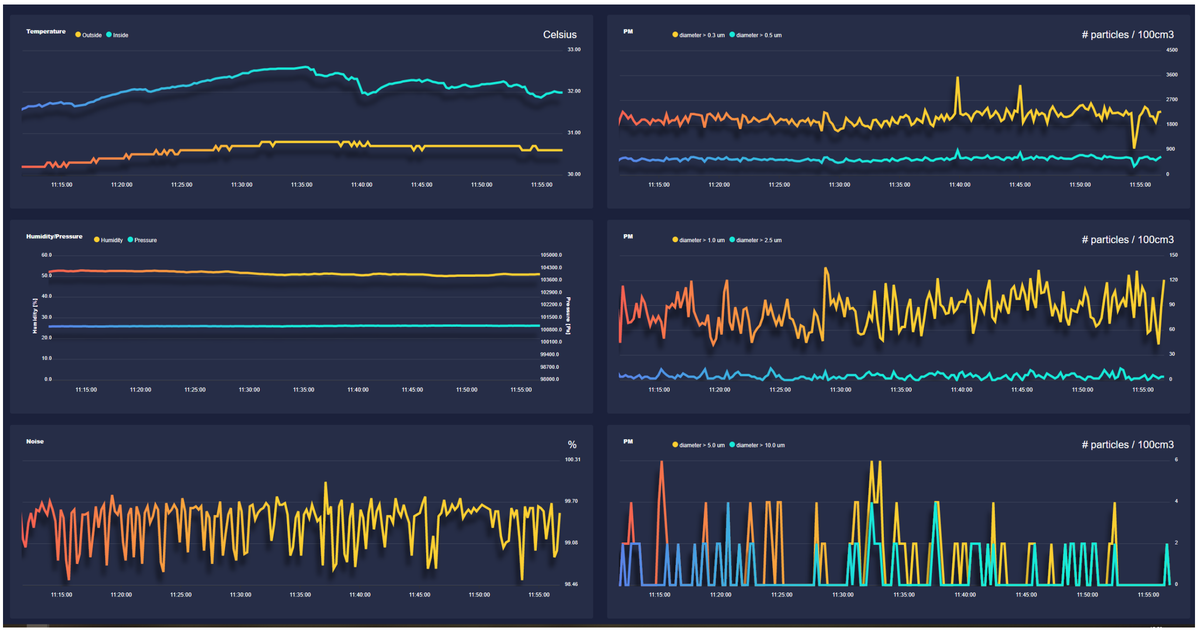


Figure 7.1: Screenshot of the web server with data tracking.

8 Conclusions

Because of the growing concerns surrounding the influence of urban climate on livability for citizens, self-powering weather stations can provide much-wanted information on relevant measures. The operating principles of the relevant sensors and the physical nature of the phenomena to be studied should be well known to make sensible decisions on the choice of sensors and the siting of the weather stations. During the literature study done for this project, it became clear that urban-sited weather stations are often unsuitable for measuring wind speed, rain and sun radiance (unless they are situated on the roof of a high building or on another object that is higher than the surrounding buildings, which makes measuring ground-level temperature or air pollution impossible). Weather stations measuring these phenomena are best situated outside of the city. The electrical components required for an urban weather station are widely available and their power use is low enough for a weather station to be self-sustaining (especially if the system can go into a low-power mode between measurements). Feasible sensors include temperature, humidity, air pressure, PM, gas and noise sensors. If an algorithm is implemented that can vary the measuring frequency of the most power-consuming sensors, more measurements can be done in times of high generation. This algorithm can use the current battery charge as an input, but it can also use generation data of the past days or years. It has been found that even a simple linearly dependent algorithm greatly improves the adaptability of the weather station.

The prototype that has been obtained meets the programme of requirements defined in section 2.3, since 1) appropriate sensors have been chosen and implemented, 2) an algorithm has been added to increase measuring frequency during times of high generation and 3) a link has been made between the weather station and a web server.

9

Recommendations

During this project, many ways were found to decrease the power consumption. Removing power-on LEDs reduced the power consumption by tens of milliwatts per removed LED. Thanks to the RTC module, the Arduino can be in low-power mode most of the time, which also significantly reduces power consumption. It was found that the weather station did not run out of power during the winter months when two 10.4 Wh batteries were used. This realization gives many options for future iterations of the weather station. A smaller battery in combination with a smaller solar panel could still be sufficient for the weather station to survive the winter months (this would especially be the case if the weather station is used in a sunnier country than the Netherlands), allowing the weather station to be more compact. The spare power could also be used to include more or better sensors. At the beginning of the project, a gas sensor was deemed too inefficient for the weather station, but later on the realization dawned that the power budget allows many types of gas sensors to be included in the weather station (even if the weather station would not use them during every measurement, in order to save power).
If an implementation with a more tight power-budget is chosen, the algorithm can be optimized for its specific purpose. Since the optimal algorithm depends on the specifics of the generation and power consumption of the sensors, each situation has to be analyzed in order to find an optimal algorithm.
If a relevant performance metric can be defined (such as number of measurements per year, possibly assigning more weight to nighttime and winter measurements), the result of fitting different models would be interesting. These models would try to optimise the defined performance metric by analysing relevant data, such as the current battery percentage, generation data of the last days or generation data of previous years.
Further research could also tackle the problem of the unstable 3.3V pin due to the fan of the PM sensor. This could possibly be solved by turning off the fan for a short amount of time during temperature, humidity and noise measurements.

A

Additional graphs

A.1. 24-hour average battery charge graphs

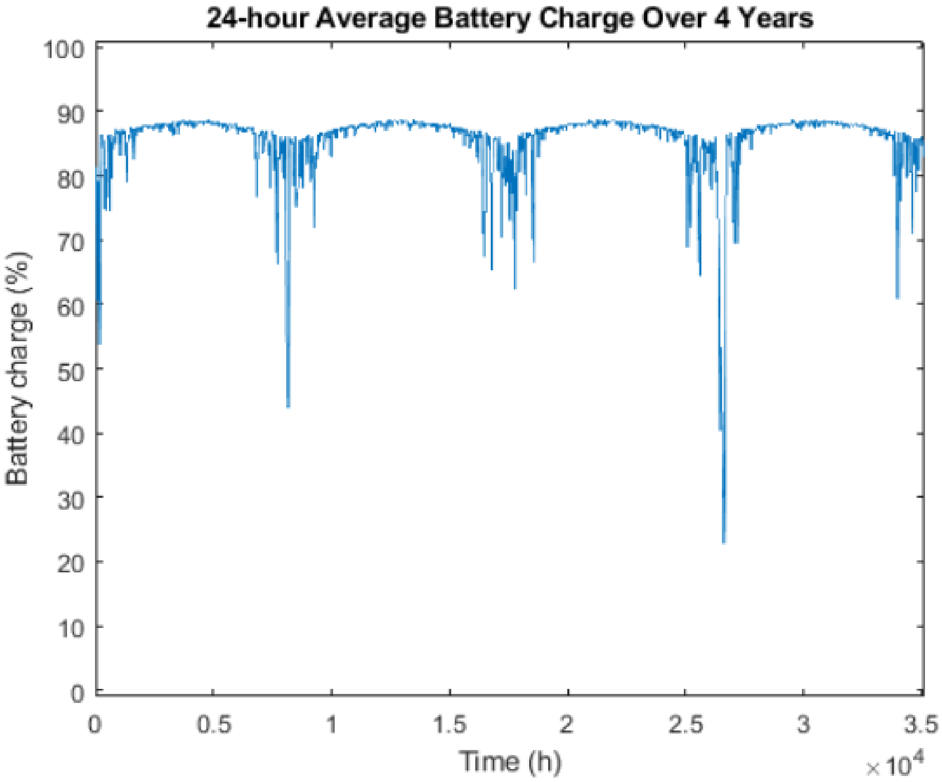


Figure A.1: Simulation with 2 batteries, 225 short measurements and 15 PM measurements.

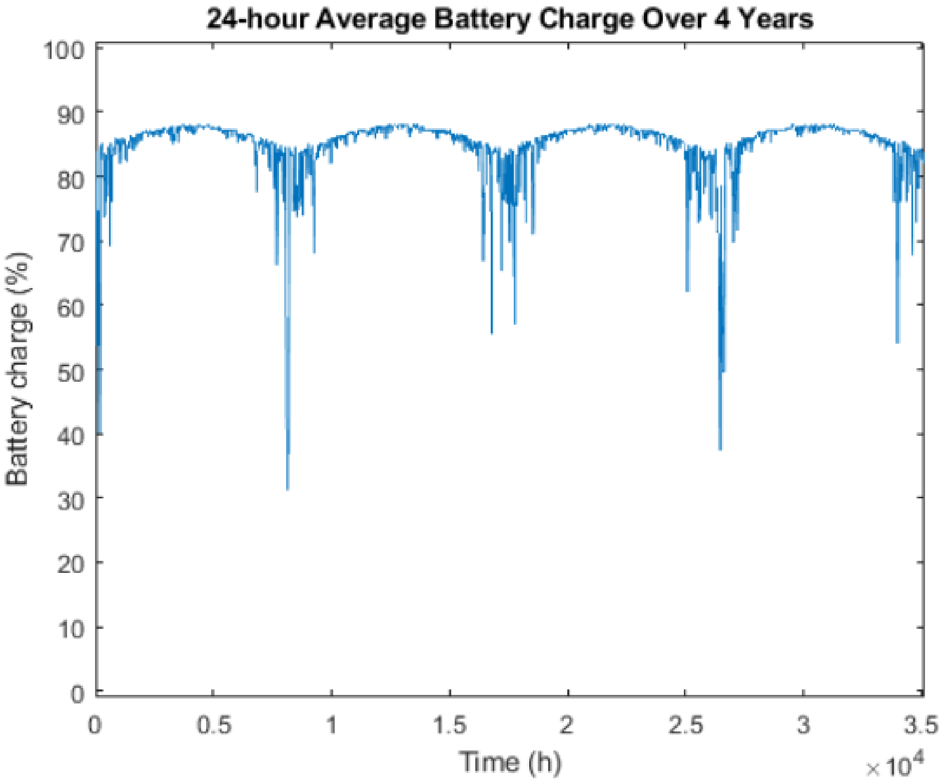


Figure A.2: Simulation with 1 battery, 230 short measurements and 10 PM measurements.

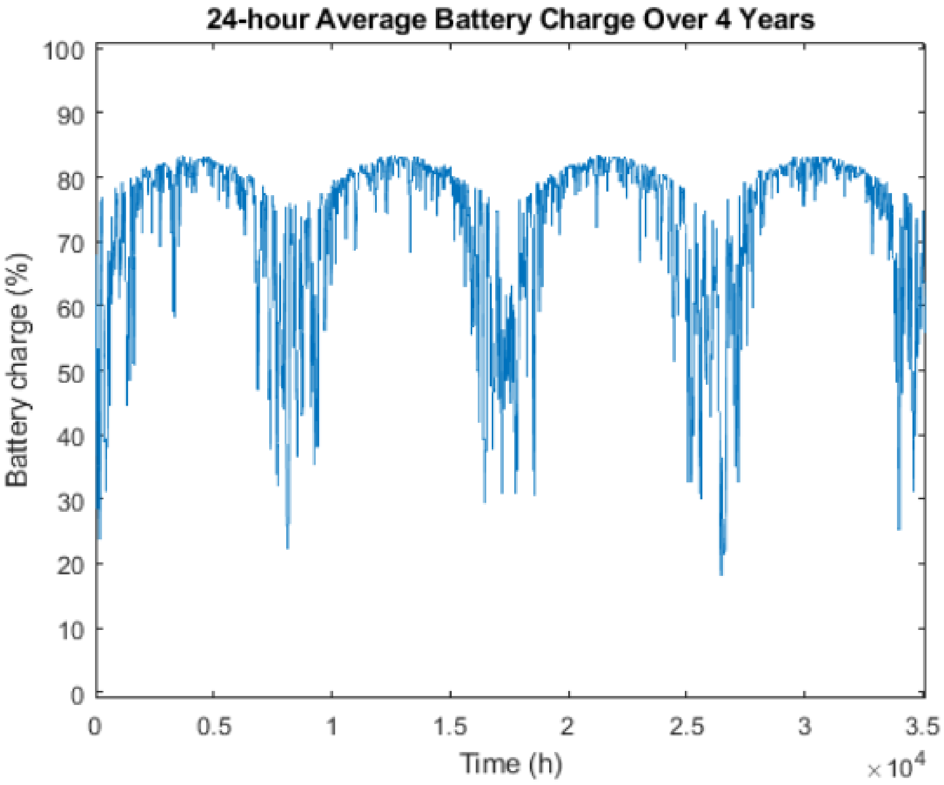


Figure A.3: Simulation with 2 batteries, continuous measurements at full charge and algorithm enabled.

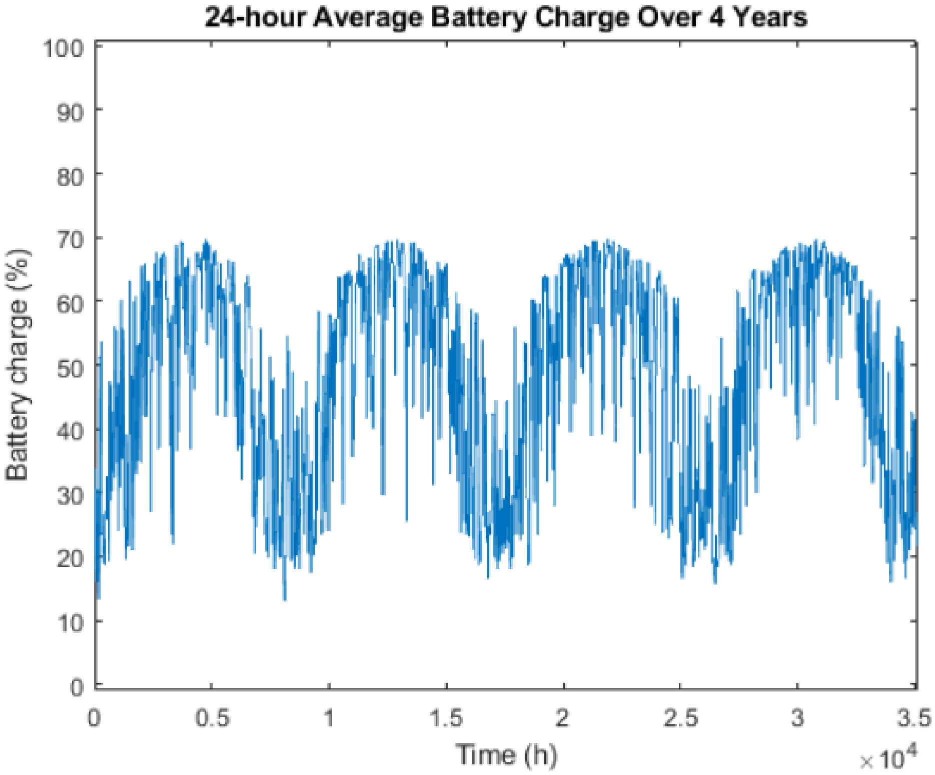


Figure A.4: Simulation with 2 batteries, continuous measurements (including gas sensor) at full charge and algorithm enabled.

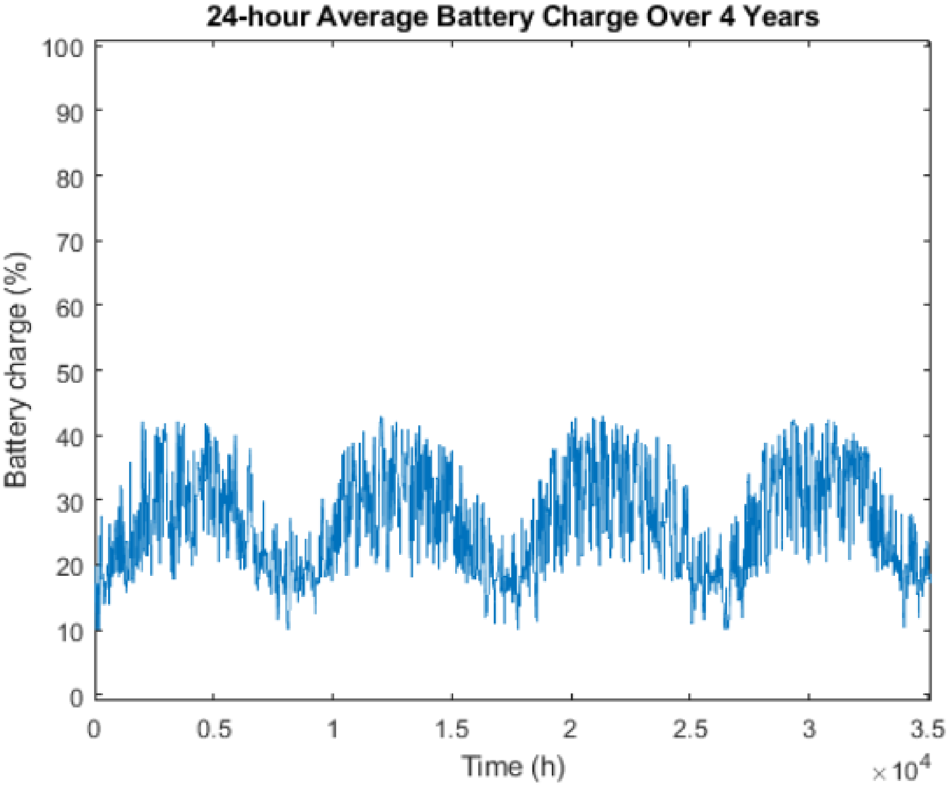


Figure A.5: Simulation with 1 battery and 1/4th PV area, continuous measurements (including gas sensor) at full charge and algorithm enabled.

B

Matlab Simulation script

Listing B.1: PowerConsumption.mlx

```
% Script written by Roan Föllings on June 17th, 2021
% This script is used to simulate the power consumption of a solar-powered weather station.

clear all;
load('Pmax_Data.mat'); % dataset containing 12 years of hourly power generation from a solar panel

% generation parameters
data = Pmax38(1:8784,9:12); % take the last 4 years of the dataset
data = reshape(data,[],1); % reshape it to 1D
mppt_efficiency = 0.75; % efficiency of the MPPT
production = mppt_efficiency*(data); % production in Watt

% consumption parameters, based on the measured power use of the sensors and Arduino
percentage_per_month = 3; % percentage of total battery charge dissipated per month
idle_consumption = 0.0165; % baseline power use
short_consumption = 0.0000231; % additional average power use for one short measurement per hour
pm_consumption = 0.005147; % additional average power use for one pm measurement per hour
gas_consumption = 0.01419; % additional average power use for one gas measurement per hour
complete_consumption = 0.02052; % additional average power use for one pm+gas measurement per hour

% tune the number of batteries and number of measurements to be done per hour
batteries = 2; % number of batteries
short_measurements = 0; % amount of measurements per hour without PM or gas sensor (takes 2 seconds), set to 1800 for continuous sensing.
pm_measurements = 0; % amount of measurements per hour with PM without gas sensor (takes 60 seconds), set to 60 to simulate continuous sensing.
gas_measurements = 0; % amount of measurements per hour without PM with gas sensor (takes 60 seconds), set to 60 to simulate continuous sensing.
```

```
complete_measurements = 60; % amount of measurements per hour with PM and gas sensor (takes 60 seconds), set to 60 to simulate continuous sensing.

% tune the algorithm
algorithm = true; % turn on the algorithm
full_battery_threshold = 0.85; % upper threshold for the algorithm
empty_battery_threshold = 0.2; % lower threshold for the algorithm
wifi_spam = true; % continuously send data over wifi above full battery threshold
wifi_consumption = 0.4; % consumption of wifi module if constantly sending
measuring_freq_factor = 0.1; % fraction of measurements to be done just above 10% battery
a = (1-measuring_freq_factor)/(full_battery_threshold-empty_battery_threshold); % slope of algorithm
b = measuring_freq_factor - a*empty_battery_threshold; % intersect of algorithm

battery_selfdissipation = 0.00296*batteries*percentage_per_month;
consumption = battery_selfdissipation + idle_consumption + short_measurements*short_consumption + pm_measurements*pm_consumption + gas_measurements*gas_consumption + complete_measurements*complete_consumption; % If the algorithm is turned off, consumption is constant
consumption_tracker(1) = consumption; % for a consumption plot
batt_cap_max = 10.4*batteries; % battery capacity in Wh
batt_cap(1) = 0.5*batt_cap_max; % battery charge at start
highest_allowed_charge = 0.9; % upper threshold of effective battery range
lowest_allowed_charge = 0.1; % lower threshold of effective battery range

for i = 1:length(data(:,1))
    if algorithm == true
        if batt_cap(i) > full_battery_threshold*batt_cap_max
            if wifi_spam == true
                consumption = battery_selfdissipation + idle_consumption + short_measurements*short_consumption + pm_measurements*pm_consumption + gas_measurements*gas_consumption + complete_measurements*complete_consumption + wifi_consumption;
            else
                consumption = battery_selfdissipation + idle_consumption + short_measurements*short_consumption + pm_measurements*pm_consumption + gas_measurements*gas_consumption + complete_measurements*complete_consumption;
            end
            measuring_freq = 1;
        elseif (batt_cap(i) >= empty_battery_threshold*batt_cap_max) && (batt_cap(i) <= full_battery_threshold*batt_cap_max)
            measuring_freq = a*batt_cap(i)/batt_cap_max + b;
            freq_tracker(i) = measuring_freq;
            consumption = battery_selfdissipation + idle_consumption + short_measurements*short_consumption + measuring_freq*(pm_measurements*pm_consumption + gas_measurements*gas_consumption + complete_measurements*complete_consumption);
        else
```

```
            consumption = battery_selfdissipation + idle_consumption;
            measuring_freq = 0;
        end
        measuring_freq_tracker(i) = measuring_freq;
    end
    batt_cap(i+1) = batt_cap(i) + production(i) - consumption;
    if batt_cap(i+1) < lowest_allowed_charge*batt_cap_max % weather station turns off if the battery charge gets below 10%
        batt_cap(i+1) = lowest_allowed_charge*batt_cap_max;
    elseif batt_cap(i+1) > highest_allowed_charge*batt_cap_max % battery cannot go above 90%
        batt_cap(i+1) = highest_allowed_charge*batt_cap_max;
    end
    consumption_tracker(i+1) = consumption;
end

batt_cap = batt_cap*100/batt_cap_max; % convert to percentage

% plot battery capacity
plot(batt_cap)
title(['Hourly Battery Charge Over 4 Years'])
xlim([0 length(data(:,1))])
ylim([-1 101])
xlabel('Time (h)')
ylabel('Battery charge (%)')
% make the graph more readable by taking the average of multiple hours
reduction_factor = 24;
for i = 1:(length(batt_cap)-reduction_factor)/reduction_factor
    temp_avg = 0;
    for j = 0:reduction_factor-1
        temp_avg = temp_avg + batt_cap(reduction_factor*i+j);
    end
    temp_avg = temp_avg/reduction_factor;
    for j = 0:reduction_factor-1
        avg_batt_cap(reduction_factor*i+j-reduction_factor+1) = temp_avg;
    end
end

% make the graph more readable by taking the minimum of multiple hours
reduction_factor = 24;
batt_cap = batt_cap*batt_cap_max/100; % convert to absolute
for i = 1:(length(batt_cap)-reduction_factor)/reduction_factor
    local_minimum = batt_cap_max + 1;
    for j = 0:reduction_factor-1
        if local_minimum > batt_cap(reduction_factor*i+j)
            local_minimum = batt_cap(reduction_factor*i+j);
        end
    end
    for j = 0:reduction_factor-1
        low_batt_cap(reduction_factor*i+j-reduction_factor+1) = local_minimum;
    end
end
low_batt_cap = low_batt_cap*100/batt_cap_max; % convert to percentage

% plot battery capacity
```

```
plot(avg_batt_cap)
title(['24-hour Average Battery Charge Over 4 Years'])
xlim([0 length(data(:,1))])
ylim([-1 101])
xlabel('Time (h)')
ylabel('Battery charge (%)')

plot(low_batt_cap)
title(['24-hour Low Battery Charge Over 4 Years'])
xlim([0 length(data(:,1))])
ylim([-1 101])
xlabel('Time (h)')
ylabel('Battery charge (%)')

% make the graph more readable by taking the average of multiple hours
reduction_factor = 24;
for i = 1:(length(consumption_tracker)-reduction_factor)/reduction_factor
    temp_avg = 0;
    for j = 0:reduction_factor-1
        temp_avg = temp_avg + consumption_tracker(reduction_factor*i+j);
    end
    temp_avg = temp_avg/reduction_factor;
    for j = 0:reduction_factor-1
        consumption_tracker(reduction_factor*i+j-reduction_factor+1) = temp_avg;
    end
end

% plot power consumption
plot(consumption_tracker)
title('24-hour Average Power Consumption')
ylim([0 max(consumption_tracker)+0.1])
xlim([0 length(consumption_tracker)])
xlabel('Time (h)')
ylabel('Consumption (W)')

if algorithm == true
    % make the graph more readable by taking the average of multiple hours
    reduction_factor = 24;
    for i = 1:(length(measuring_freq_tracker)-reduction_factor)/reduction_factor
        temp_avg = 0;
        for j = 0:reduction_factor-1
            temp_avg = temp_avg + measuring_freq_tracker(reduction_factor*i+j);
        end
        temp_avg = temp_avg/reduction_factor;
        for j = 0:reduction_factor-1
            measuring_freq_tracker(reduction_factor*i+j-reduction_factor+1) = temp_avg;
        end
    end

    % plot relative measuring frequency
    plot(measuring_freq_tracker)
    title('24-hour Average Relative PM and Gas Measuring Frequency')
    ylim([0 max(measuring_freq_tracker)+0.1])
    xlim([0 length(consumption_tracker)])
    xlabel('Time (h)')
    ylabel('Relative Frequency')
end
```

Bibliography

[1] WMO, “Guide to instruments and methods of observation,” vol. 1, 2018.

[2] A. Tong, “Improving the accuracy of temperature measurements,” *Sensor Review*, vol. 21, pp. 193–198, 2001.

[3] P. J. Schubert and J. H. Nevin, “A polyimide-based capacitive humidity sensor,” *Transactions on Electron Devices*, vol. 7, pp. 1220–1223, 1985.

[4] D. K. Roveti, “Choosing a humidity sensor: A review of three technologies,” 2001. [Online]. Available: https://www.fierceelectronics.com/components/choosing-a-humidity-sensor-a-review-three-technologies

[5] “Wheatstone bridge diagram,” accessed on 14-06-2021. [Online]. Available: http://www.ibiblio.org/kuphaldt/electricCircuits/DC/DC_8.html

[6] E. Frederiksen, *Handbook of Signal Processing in Acoustics*, D. Havelock, S. Kuwano, and M. Vorländer, Eds. Springer, 2008, chapter: Condenser Microphones.

[7] S. A. Zawawi, A. A. Hamzah, B. Y. Majlis, and F. Mohd-Yasin, “A review of mems capacitive microphones,” *Micromachines*, vol. 11, 2020.

[8] E. Werner, *Handbook of Signal Processing in Acoustics*, D. Havelock, S. Kuwano, and M. Vorländer, Eds. Springer, 2008, chapter: Dynamic Pressure Microphones.

[9] N. Yamazoe, G. Sakai, and K. Shimanoe, “Oxide semiconductor gas sensors,” *Catalysis Surveys from Asia*, vol. 7, pp. 63–75, 2003.

[10] K. E. Kelly, J. Whitaker, A. Petty, C. Widmer, A. Dybwad, D. Sleeth, R. Martin, and A. Butterfield, “Ambient and laboratory evaluation of a low-cost particulate matter sensor,” *Environmental Pollution*, vol. 221, pp. 491–500, 2017.

[11] J. Han, X. Liu, M. Jiang, Z. Wang, and M. Xu, “A novel light scattering method with size analysis and correction for on-line measurement of particulate matter concentration,” *Journal of Hazardous Materials*, vol. 401, pp. 1–7, 2021.

[12] J. Han, X. Liu, D. Chen, and M. Jiang, “Influence of relative humidity on real-time measurements of particulate matter concentration via light scattering,” *Journal of Aerosol Science*, vol. 139, 2020.

[13] *Digital universal particle concentration sensor*, PLANTOWER, 2016, rev. 2.3. [Online]. Available: https://cdn-shop.adafruit.com/product-files/3686/plantower-pms5003-manual_v2-3.pdf

[14] *Low-Power, 32-bit Cortex-M0+ MCU with Advanced Analog and PWM*, Microchip Technology Inc., 2020. [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/SAM_D21_DA1_Family_DataSheet_DS40001882F.pdf

[15] *Arduino® MKR WIFI 1010*, Arduino S.r.l., 2020. [Online]. Available: https://store.arduino.cc/arduino-mkr-wifi-1010

[16] *Arduino® MKR IoT Carrier*, Arduino S.r.l., 2021, rev. 1. [Online]. Available: https://content.arduino.cc/assets/MKR%20IoT%20Carrier%20-%20Datasheet.pdf

[17] *Extremely Accurate I2C-Integrated RTC/TCXO/Crystal*, Maxim Integrated, 2015, rev. 10. [Online]. Available: https://datasheets.maximintegrated.com/en/ds/DS3231.pdf

[18] T. Longcore and C. Richt, “Ecological light pollution,” *Frontiers in Ecology and the Environment*, vol. 2, pp. 191–198, 2004.

[19] P. Cinzano and F. Falchi, “Quantifying light pollution,” *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 139, pp. 13–20, 2014.

[20] L. van Hove, G. Steeneveld, C. Jacobs, B. Heusinkveld, J. Elbers, E. Moors, and A. Holtslag, “Exploring the urban heat island intensity of dutch cities,” 2011.

[21] *Digital temperatures and humidity sensor*, AOSONG. [Online]. Available: https://cdn-shop.adafruit.com/datasheets/AM2315.pdf

[22] E. D. Coffel, R. M. Horton, and A. de Sherbinin, “Temperature and humidity based projections of a rapid rise in global heat stress exposure during the 21st century,” 2018.

[23] J. Wieringa, “Description requirements for assessment of non-ideal wind stations,” *Journal of Wind Engineering and Industrial Aerodynamics*, pp. 121–131, 1983.

[24] Z. Chen, S. Cheng, J. Li, X. Guo, W. Wang, and D. Chen, “Relationship between atmospheric pollution processes and synoptic pressure patterns in northern china,” *Atmospheric Environment*, vol. 42, pp. 6078–6087, 2008.

[25] *Digital Pressure Sensor*, Bosch Sensortec, 2015, rev. 1.14. [Online]. Available: https://cdn-shop.adafruit.com/datasheets/BST-BMP280-DS001-11.pdf

[26] M. C. Roorda-Knape, N. A. H. Janssen, J. J. de Hartog, P. H. N. van Vliet, H. Harssema, and B. Brunekreef, “Air pollution from traffic in city districts near major motorways,” *Atmospheric Environment*, vol. 32, pp. 1921–1930, 1998.

[27] *Electret Condenser Microphone*, CUI Devices, 2020. [Online]. Available: https://www.cuidevices.com/product/resource/cma-4544pf-w.pdf

[28] *Low-Cost, Micropower, SC70/SOT23-8, Microphone Preamplifiers with Complete Shutdown*, Maxim Integrated, 2012, rev. 2. [Online]. Available: https://datasheets.maximintegrated.com/en/ds/MAX4465-MAX4469.pdf

[29] M. N. W. Chan-Yeung, “Air pollution and health,” *Hong Kong Medical Journal*, vol. 4, pp. 390–398, 2000.

[30] *MQ-135 gas sensor*, Hanwei. [Online]. Available: https://www.bitsandparts.nl/documentation/532/Datasheet_gassensor_MQ-135.pdf

[31] *The MiCS-6814 is a compact MOS sensor with three fully independent sensing elements on one package*, SGX Sensortech, rev. 8. [Online]. Available: https://www.sgxsensortech.com/content/uploads/2015/02/1143_Datasheet-MiCS-6814-rev-8.pdf

[32] K.-H. Kim, E. Kabir, and S. Kabir, “A review on the human health impact of airborne particulate matter,” *Environment International*, vol. 74, pp. 136–143, 2015.

[33] B. G. Marin B. Marinov, Stefan Hensel and G. Nikolov, “Performance evaluation of low-cost particulate matter sensors,” *International Scientific Conference Electronics*, vol. XXVI, 2017.

[34] *SYNCHRONOUS BOOST CONVERTER WITH 2A SWITCH*, Texas Instruments, 2003, rev. C. [Online]. Available: https://www.ti.com/lit/gpn/tps61090

[35] *ESP32 Series Datasheet*, Espressif Systems, 2021, rev. 3.6. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[36] M. Vetter and S. Lux, *Storing Energy*, T. M. Letcher, Ed. Elsevier, 2016, chapter: Rechargeable Batteries with Special Reference to Lithium-Ion Batteries.

[37] “Photovoltaic geographical information system,” European Commission, accessed on 20-05-2021. [Online]. Available: https://re.jrc.ec.europa.eu/pvg_tools/en/#MR