DELIVERABLE

D3.2 Sensor device functional and technical design report

Project Acronym:	COMPAIR	
Project title:	Community Observation Measurement & Participation in AIR Science	
Grant Agreement No.	101036563	
Website:	www.wecompair.eu	
Version:	1.0	
Date:	31 August 2022	
Responsible Partner:	SODAQ	
Contributing Partners:	Telraam IMEC	
Reviewers:	Karel Jedlička (UWB) Otakar Čerba (UWB) Martine Van Poppel (VITO) Carolina Doran (ECSA) Beatriz Ortega (ECSA)	
Dissemination Level:	Public	X
	Confidential, only for members of the consortium (including the Commission Services)	

This project has received financial support from the European Union's Horizon 2020 Programme under grant agreement no. 101036563

Revision History

Version	Date	Author	Organization	Description
0.1	15.07.22	Ajay Jamodkar Daria Yakushkina	SODAQ	Initial structure
0.2	11.08.22	Burcu Zijlstra	IMEC	Input added
0.3	12.08.22	Kris Vanherle	Telraam	Input added
1.0	17.08.22	Daria Yakushkina	SODAQ	Final version

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Definition
API	Application Programming Interface
FOTA	Firmware Over the Air
LCS	Low-Cost Sensor
OGC	Open Geospatial Consortium
OTAU	Over The Air Update
IP	Ingress Protection
IoT	Internet of Things
NPU	Neural Processing Unit
UI/UX	User interface (UI) - user experience (UX)
PCB	Printed Circuit Board
PCBA	Printed Circuit Board Assembled
MoSCoW	Must Should Could Won't
PCBA	Printed Circuit Board Assembled
DIY	Do It Yourself

Executive Summary

The D3.2 Sensor device functional and technical design report is a part of the COMPAIR deliverables in line with Annex 1 to the Grant Agreement. The current document presents the selection of the type of sensors - with a focus on low-cost sensor units - for the COMPAIR project, their specifications and integration possibilities. The report also pays special attention to the technical and functional design of these sensors for Citizen Science projects and how low-cost sensors can be integrated in a smart city environment as an intelligent data source.

The first sections of the document provide a summary of the requirements set by the pilots in Athens, Plovdiv, Sofia, Berlin and Flanders, and derive functional as well as non-functional technical requirements from them together with the set of parameters needed to be measured. Next, based on these requirements and a literature study, low-cost sensors that would be a suitable fit for the pilot projects are suggested.

Finally, the main sections elaborate on the product concept and calibration and standardization that will be undertaken in the future. It describes the compatibility of the data with other existing air quality sensor data, what kind of open standards will be used and how the metadata will be described. The re-usability of the sensor data includes a description of how data from the sensors will feed into the COMPAIR products (API use).

In terms of the current COMPAIR project, air quality sensors and traffic monitoring sensors are the main focus based on the selection results. This selection was made based on the product concepts and initial requirements. Annex 1 of this document contains additional information on the operational behaviour of the proposed design product concepts.

The sensor strategy section links to the pilot use cases, outlining what needs to be measured and how to use those sensors, generating open data for citizen science purposes and for local policy making.

1. Introduction

COMPAIR is an innovation project designed to bolster citizens' capacity to monitor, understand, and change their environmental impact, both at a behavioural and policy level. It unlocks the power of the wider public, including people from lower-socio economic groups, to provide broad granular data around a central theme of air quality, complementing and improving the quality of official datasets and making new information useful for research purposes, policy making and behavioural change.

The objective of this report is to develop the required technical innovation to enhance the use of citizen science initiatives to address the demands of the scientific community and policymakers in the field of environment and smart cities. This includes developing and improving Do It Yourself (DIY) air quality sensors and traffic count sensors to serve the citizen science (CS) pilots, and the development of novel data processing techniques and open data integration, and analytics for policy modelling and scientific analytics.

The D3.2 Sensor device functional and technical design report focuses on developing and improving DIY low-cost sensor units to serve the needs of citizen science initiatives in the project. Two main categories of sensor units are considered to measure, respectively, air quality and traffic count. For air quality measurement, the task will first focus on selecting the appropriate low-cost off-the-shelf sensors for PM (Particulate Matter) and NO2 (Nitrogen Dioxide). The selection criteria will include sensitivity and expected accuracy, and cost and form factor trade-offs to serve the specific application. These sensors will then be configured and integrated into DIY devices for the different CS pilots. Novel (more) reliable, low-cost DIY NO2 devices will be developed to complement the existing CS PM sensor network used in sensor.community. Moreover, low-power mobile personal devices will be developed for dynamic exposure measurements. Telraam DIY traffic count sensors, proven successful in several citizen science initiatives for traffic count sensors, will be further improved to enhance user-friendliness.

Based on the strategy and requirements (functional and technical) that were described in the D3.1 Sensor strategy and requirements report, a functional and technical design of the needed sensors considering the needed levels of accuracy and how they will be achieved is covered in this report.

2. User Stories & Requirements

2.1. Air Quality

Sensor requirements for air quality sensors were adjusted to a more measurable format after analyzing them for their technical feasibility. The tables below, presents the agreed sensor requirements and air quality parameters that will be measured by the Air Quality sensors. Table 1 shows the Air quality parameters that were agreed as feasible parameters to be measured at the end of deliverable D3.1. The parameters which must indicate those will be measurable by the Air quality sensor devices. During the development process effort would be spent to see if the parameters mentioned with could be accommodated but are not guaranteed. The parameters with won't will not be available. Table 2, shows the original requirements and comments on the constraints as well as mentions achievable requirements.

***Table 1**:* Air parameters & pollutants

Description			Unit	Requirment (MoSCoW)
Air parameters & pollutants	PM10	Particulate matter <10um (micrometers)	µg/m3 (micrograms per meter cube)	Must
	PM2.5	Particulate matter <2.5um	µg/m3	Must
	NO_2	NO_2 gas (respiratory pollutant)	µg/m3	Must
	BC	Black carbon	µg/m3	Could
	O3	Ozone	µg/m3	Could
	ultrafine PM	Particulate matter <0.1um	µg/m3	Won't
	Benzene		µg/m3	Won't
	VOC	Volatile organic compounds	µg/m3	Won't
	PAH	Polycyclic aromatic hydrocarbons	µg/m3	Won't
	SO2	Sulphur dioxide	µg/m3	Won't
	T	Temperature	degC	Must
	H	Relative humidity	%	Must
	Wind speed and direction		deg, m/s	Won't

***Table 2**: Air quality sensor requirements original (D3.1) vs updated based on technical constraints*

User Persona	Original Requirement	Updated Requirement (measureable)	Comments	MoSCoW	SODAQ Air	SODAQ NO2
As a researcher	I can rely on the accuracy of the sensor to meet the indicative label of VMM		This will be evaluated via VMM benchmarking tests	Must	In progress (VMM)	In progress, (VMM)
As a citizen	I want to get notified when sensor is out of spec and intervention is needed	I want to know when the sensor is working visually that the device is turned on (green/red light)"	Maybe a requirement for PMD, not sensor?	Must	Yes	No, in scope for the 2nd iteration (open testing)
As a citizen	I can count on the device to be fully charged within half an hour	Device should be rechargeable using a regular USB Charger	Charging within half hour is ambitious and needs to be confirmed (keeping price factor of the device)	Must	Yes (USB-C)	Yes (Micro USB)
As a citizen	I can count on the device to last at least 24 hours without charging	I can count on the device to last at least 4 hours without charging. Using a powerbank you can continuously measure	SODAQ Air ensures only 5h. Sodaq NO2, allows 4 hours	Must	Yes	Yes
As a citizen	I can carry the device while exercising without it bothering me (maximum similar to carrying a smartphone in terms of size and weight)	See Comment.	Comparing it to a smart phone is ambitious in terms of weight and dimensions. Not technically feasible. Air 250 gms, NO2 is 950 gms (target is to reduce it), personally carrying NO2 device would be difficult due to weight.	won't	No	No
As a citizen	I can expose the device to all normal weather conditions (e.g. rain, wind, hail, cold & warm temperatures)	The device should be exposable to IP62/IP63 conditions without damaging the operations of the device (Not waterproof).	As devices need to be in contact with outside air, it is not possible to make it completely IP65. Hence the device has a Dry Area (IP65) and wet area. Dry area has the main microcontroller and battery. Wet Area hosts sensor modules that are exposed to outside conditions but has no IP ratings.	Must	Yes	Yes

As a citizen	I can attach the device to various objects (e.g. bicycle, clothing) and easily attach and detach it	The device can be attached to a bike or a metallic bar for gathering measurements.	These two requirements are kind of contradictory	Must	Yes	Yes
As a citizen	I can securely attach the device to e.g. my bike	This requirement conflicts with user friendliness. So removed.		won’t	No	No
As a citizen	I can see if the device is working via a simple status light	Same requirement as before: "I want to know when the sensor is working visually that the device is turned on (green/red light)"		Should	Yes	No, in scope for the 2nd iteration (open testing)
As a citizen	I want to put the sensor outside on my car	Not achievable due to high wind speeds interfering with PM and NO2 measurements.	The wind speeds could damage the PM and NO2 sensors (if driving too fast) and also the reeadings might not be reliable at such a high speed. Needs to validated through experimentation. Not sufficient data available	won’t	No	No
As a citizen	I want to measure air pollution even at home		the sensor will measure everywhere when it is in active mode: but imec calibration will not work indoors	should	YES (but no calibration)	YES (but no calibration)
As a citizen	I want to know if a sensor is inside or outside	This can be a dashboard requirement	indoor vs outdoor is difficult to meausre. COuld be achieved in cloud based on GPS coordinates or daily deltaT, deltaRH compared to other sensors: PMD requirement?	should	Better through PMD	Better through PMD

2.2. Traffic counting

The user-defined requirements are focusing mostly on the air quality sensor and are discussed in the previous section. The user requirements for the Telraam sensor focus on the type of traffic data being collected and the temporal resolution. Unlike for the air quality sensor, there is no list of explicit users stories defined, other then “as a citizen, I want to know the traffic in my street”. Speed of traffic is not considered a key requirement by the users.

Building on the learnings from WeCount and other Telraam-users, the requirements for the sensor itself can be summarised as follows:

1. Ease of installation to ensure non-technical users can self-install the sensor
2. Avoid error-prone user handling; avoid user handling in critical installation steps such as wifi-setup and camera-angle setting and ideally also location setting
3. Elegant and inviting/playful design to engage users.

We added requirements from a useability point of view and updated the MoSCoW table as presented in D3.1.

***Table 3**: Traffic sensor requirements*

MoSCoW	Description
Must have	• Easy physical mounting procedure • Data connectivity without user involvement • Accuracy of car countings > 90%
Should have	• Unit “fly-away” cost <150€, covering cost of the sensor itself, packaging and any other cost. • Accuracy of car countings during daytime with sufficient light > 95% - other modes > 85% • Power consumption <5 Watt • (basic) counts when dark - at least for cars • Split between more categories (at least light/heavy freight, bus, motorcycle) • Elegant design
Could have	• Automatic location & orientation fix • Interactive screen (touchscreen/LCD/e-ink) • Speed calculation
Won’t have	• High-end hardware (e.g. Jetson Nano or Google Coral) • Other indicators apart from traffic counts & speeds • Power independent (i.e.. solar/battery powered)

3. Product concept - Air quality (SODAQ)

3.1. AIR Sensor

SODAQ AIR (Also referred as AIR in this document) has been designed keeping in mind that the device will be used by participants from all age groups to collect air quality data about their surroundings in both mobile and static conditions.

Figure 1: SODAQ AIR for air quality monitoring

The device resembles a slightly larger bicycle bell and records the timestamps, location of the device, concentration of fine particulate matter (PM2.5,1.0,10), temperature and humidity. The data is then transmitted to the cloud via LTM/NB-IoT (Cellular networks). The frequency of sending data depends on whether the device is moving or stationary. The device can act as a static environmental sensor to measure air pollution from a fixed location over an extended period of time, making it the ideal solution to be used at home or in the garden. LED lights on the device instantly display the local conditions and notify the user of the air quality. Each data point is not associated to a device id due to data protection considerations.

3.1.1. Hardware

The hardware of the SODAQ AIRcomprises main electronics board, sensor units (PM & Temperature-humidity) and a supercapacitor that powers up the device.This section will focus on the importance of each of these units and their role.

Main Electronics Board (PCBA)

The main Printed Circuit Board Assembled (PCBA) hosts the main microcontroller, power management circuitry, GPS module and other auxiliary components to gather and process data. The microcontroller used is Nordic NRF9160 which also acts as the modem for LTE-M/NB-IoT cellular connectivity. LTE-M antenna from Ignion is used which is one of the best providers in this sector. The GPS is collected using the UBLOX GPS module. The power circuitry ensures that the device can be charged using USB C and protects the battery from over discharging. There is an accelerometer onboard that is used for detecting whether the device is in motion or not. It also has the hall effect sensor that detects the presence of the magnet in the bike mount to activate or deactivate the device.

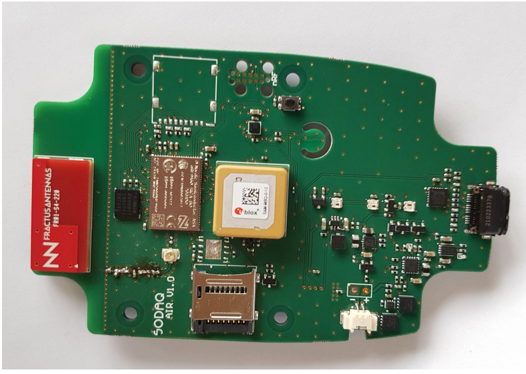


Figure 2: SODAQ AIR Main PCBA (Source: SODAQ)

Supercapacitor

Unlike traditional IoT devices, SODAQ AIR uses a 1Ah supercapacitor as a source of power. The main reason is extended lifetime and better performance as compared to Lithium Ion batteries. A supercapacitor generally can go through 10 times more charge discharge cycles as compared to a Li-ion battery.

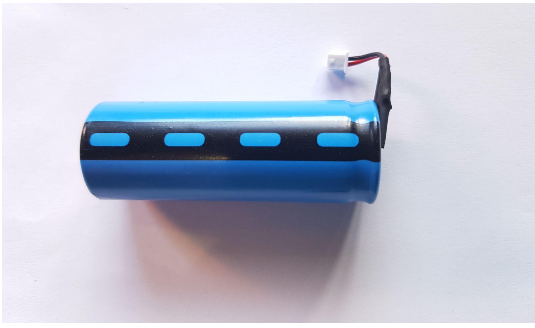


Figure 3: 1AH supercapacitor power source (Source: SODAQ)

Sensor units

SODAQ AIR consists of two main sensor units. The first one is SPS30 which is used for measuring the PM values and the second one is SHT31 which is used to measure the outside temperature and humidity. Both units are exposed to the outside air and are present on the bottom side of the device so that they produce accurate readings and are least influenced by the heat generated by the device itself.

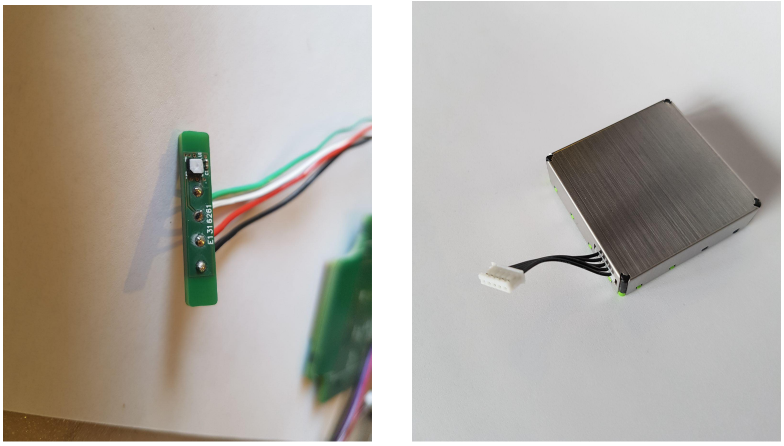


Figure 4: a) SHT31 sensor unit b) SPS30 sensor unit (Source: SODAQ)

Casing and Mounting

The AIR device casing has two compartments namely wet area and dry area. The wet area is the part where the sensor units are housed and is sealed from the dry area by the bottom part of the main PCBA. The wet area is called so because the part is exposed to outside air and thus also gets exposed to the water during rainy conditions. To prevent any damage to the smaller SHT31 sensor PCB, it has been coated with a water resistant layer during production.

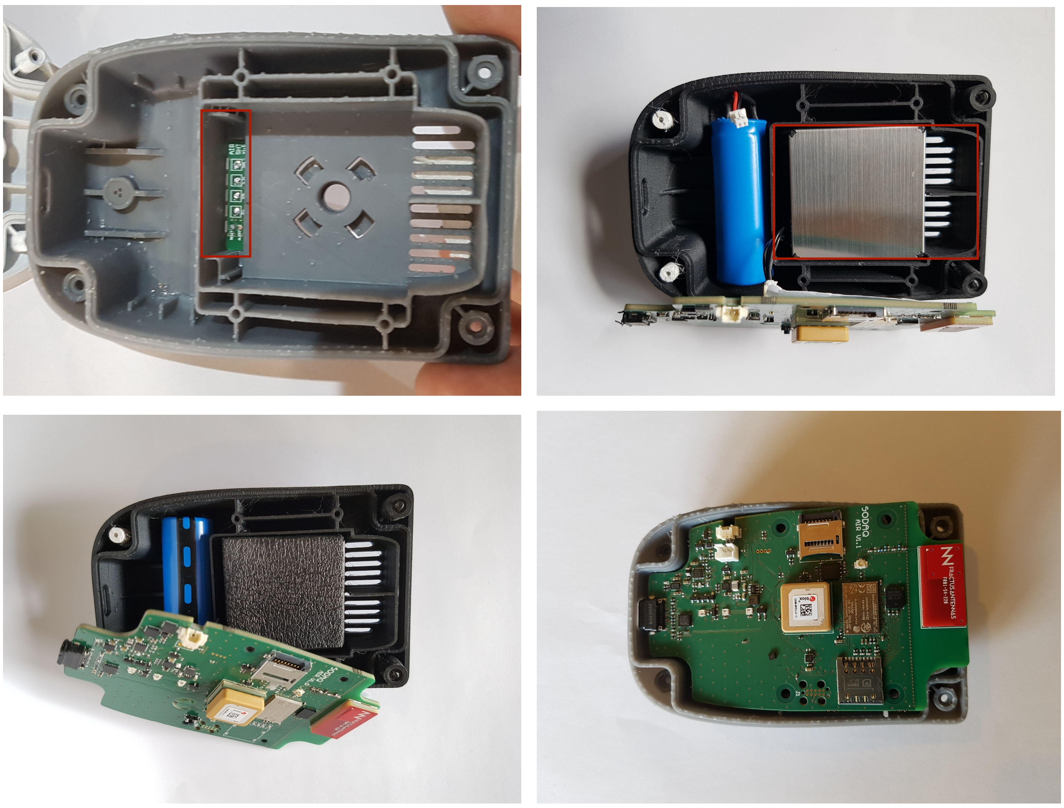


Figure 5: SODAQ AIR components (Source: SODAQ)

a) SHT31 sensor unit

b) placement of SPS30 PM sensor where the red box indicates the wet area

c) image showing use of foam and the back side of the pcba

d) half assembled area device

Another important part of the AIR device is its mount that consists of a magnet. The mount can be used to easily attach the device to a bicycle or a pipe for stationary purposes. The magnet in mount ensures that once you attach the device on the magnet it activates the device. Activating the device wakes it up from sleep mode and the device then starts measuring data. The frequency of data measurement depends on its motion status. The bottom casing also has a ¼ inch screw head that can be used for special mounting scenarios.

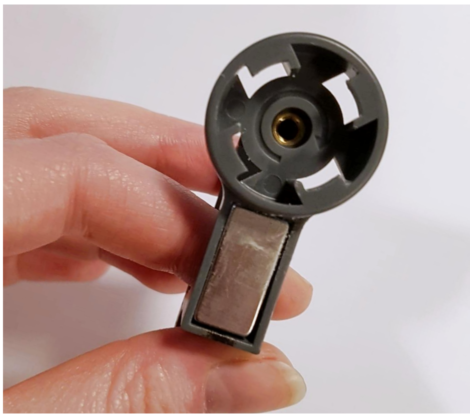


Figure 6: AIR bicycle mount and it’s usage

Charging

The device can be charged using a USB C cable and any mobile phone charger. It takes around 2 hours to completely charge the device. The device lasts for almost 4 hours while continuously in motion and more than 12 hours when used as a static sensor. The longer lifetime while being static is because the device sends data every 5 minutes as compared to every 10 seconds when in motion.

3.1.2. Software

The software of the AIR devices is designed keeping in mind that the device will be mostly used by non experts. This section describes the key technical features of the software and then how these are used during the functioning of the software.

Firmware Over The Air (FOTA)

As the name suggests, Air is capable of updating its firmware over the air without any user intervention. This gets triggered from the backend Over the Air Update (OTAU) server whenever a new updated software version is available. There are are two ways the device can receive an update:

- When the device powers on or resets because of user intervention, it is going to perform a FOTA(Firmware Over the Air) operation. If there is no new update then the device will perform its normal operation.
- Another input to the device is when it gets connected to the power. At that point the device will try to perform a FOTA update. If a FOTA is available then the device will enter the FOTA state and will be completed in less than 5 minutes.

Deep Sleep and Active Mode

In order to conserve power, whenever the device is not activated the microcontroller shuts down its peripherals and goes into sleep mode. It also goes to deep sleep, in the time between measurements when the device is activated but lying static. This helps with extending the battery usage of the device. The deep sleep mode is also referred to as an Idle state.

PM Sensor Warm UP

The PM sensor needs to run for at least 30 seconds before it starts producing stable readings. Therefore, a warm-up time has been put in place when the device starts after a break. The warm time is around 1 minute to be on the safe side.

Magnet Input

The device has a hall sensor (a sensor to detect presence of magnetic field) that can detect the presence of a magnet (refer figure 6) and is used as a mechanism to activate/deactivate the device. This is to ensure that the users can control when they would like to measure AIR quality.

User Reset

The device can be manually reset by pressing a small button present on the bottom side of the device with the help of a paper clip. The reset is mostly used when the device freezes for an unknown reason or needs to be updated. It is required by users if an upgrade is needed for a critical bug on the software. Also, it doesn't affect previous data as that is sent to the cloud when collected.

Device startup

When the device is started after a complete power down, it initialises all the sensors and reads out the reset cause. If the reset cause is reset from the user, then the device will go into the FOTA state, where it will attempt to connect to the OTAU server and check if there is any new update. Once the device completes the FOTA operation, it will go to the Idle (Deep sleep) state, where it will wait for the magnet activation. Similarly, if the reset is caused by some other reason, then the device will restart and go to the Idle state, where it will stay until a trigger from a Magnet input is received (Fig 7).

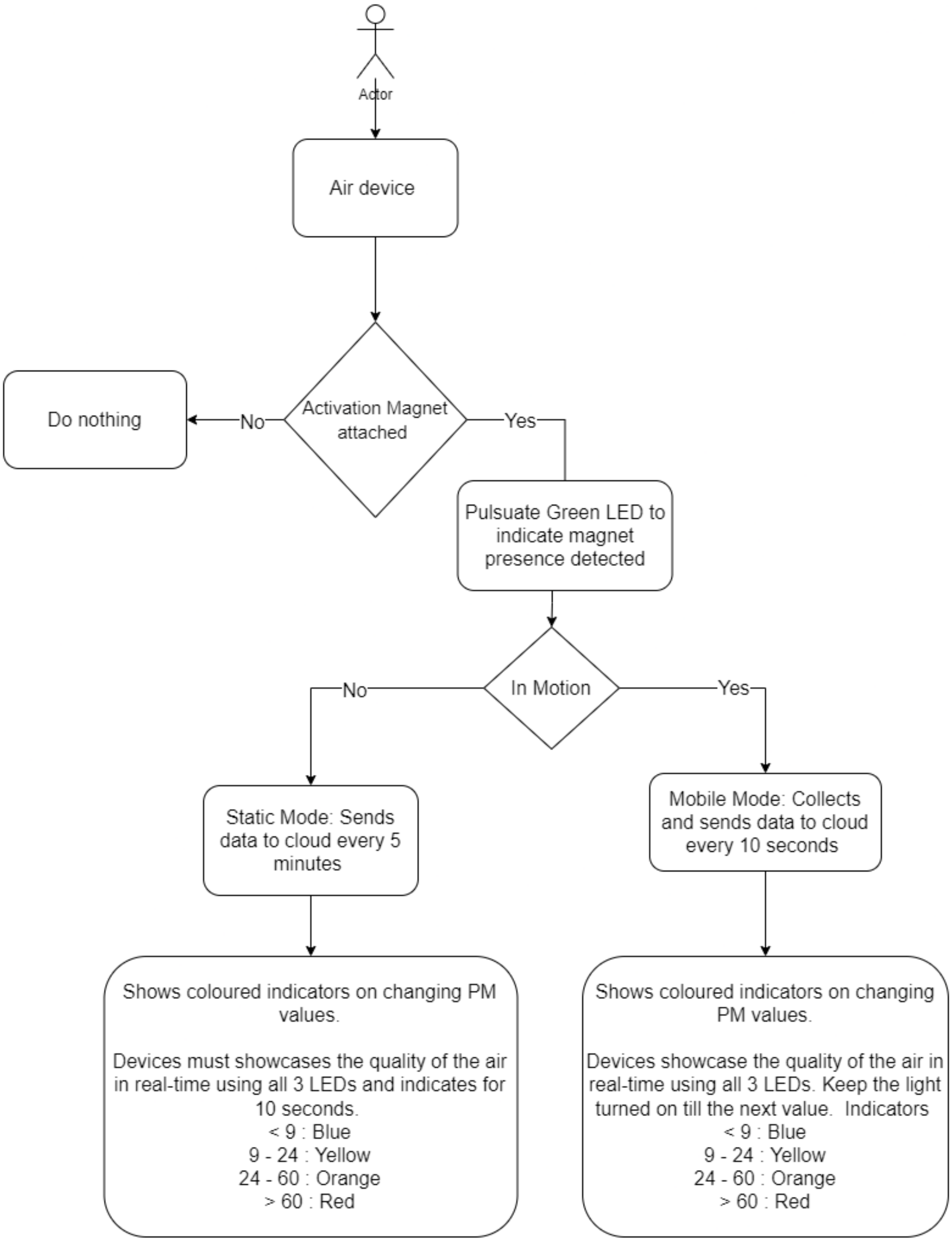


Figure 7: AIR Device startup

3.1.3. Data

Data Format

The data currently includes hardware-related information, version, timestamp, battery voltage, microclimate measurements, PM measurements, satellite and location-related information, and motion status, as observed below.

***Table 3**: Data output structure*

Variable	Example	Meaning	Unit
imei	352656107228127	Hardware ID or unique id for identifying the device	
timestamp	1.65E+12	It is the timestamp as sent by the device which is maintained by the device based on cellular time (UTC)	ms since epoch
voltage	3346	This is the battery voltage	mV
temperature	24.55	External temperature measured by the device	deg C
humidity	25.68	External humidity measured by the device	%RH
pm_1p0	22.44	particulate matter <1um	µg/m3
pm_2p5	23.84	particulate matter <2.5um	µg/m3
pm_10p0	23.99	particulate matter <10um	µg/m3
sats	17	Number of satellites seen that were used to get the Lat lon of the location	
h_acc	1361	This is the horizontal accuracy estimate of the position	m
v_acc	1753	This the vertical accuracy estimate of the the GPS position	m
lat	52.0757963	latitude	
lon	5.1858017	longitude	
rsrp	46	This is the signal strength of the Cell signal (LTE-M)	dBm
motion_status	0/1	Motion status, 0 is for idle and 1 is for moving	
gps_utc_epoch_time	1.65E+12	EPOCH time based on the GPS time. It is time received from the GPS signal	
imsi	232031721761704	International Mobile Subscriber Identity(IMSI) is a unique identifier for the sim chip	
iccid	89430301722117617041	Integrated Circuit Card Identification Number (ICCID).This is a unique identifier for sim chip	
name	352656107228127	sensor ID is same as the imei of the modem	
device_timestamp	1648117543	Timestamp sent by the device which is kept in sync with GPS time	

Based on future developments and upgrades the device data might also get updated and it might send some more information. This will be communicated to all the partners so that they can also upgrade their system if required. The device sends its data over LTE-M/NB-IoT which are cellular protocols designed specifically for low powered IoT devices.

3.2. NO2 Sensor

One of the key requirements of the CompAir project is the ability to measure NO2 related air parameters. This can be achieved by using the NO2 sensor developed by SODAQ in the past as shown in Figure below. The NO2 device is also equipped with other sensors such as PM, temperature and humidity sensors. This makes it an independent unit that sends sensor data and meta-data to SODAQ's cloud platform. As the NO2 sensor is in prototype phase, under the CompAir project, effort would be taken to improve its user experience and make it production friendly thereby reducing its cost.

Figure 8: SODAQ's NO2 sensor (source: SODAQ)

In the previous deliverable, a device concept was suggested for a AIR+NO2 device but due to technical limitations the concept was not found to be feasible a decision was made to continue with the NO2 device as it is to meet the time frame of the project. The main issue with developing the AIR+NO2 device was combining all the sensors in one casing leading to a comparable weight and form factor as that of the suggested prototype device. Further, following the regular design process, developing a fully tested device would have taken at least 1 year. As there is already a ready to use design for NO2 measurements, it makes more sense to use the available one and further, use the development time available to incorporate the necessary features in the NO2 device itself. The sections below elaborates on the hardware, software and data features of the product.

3.2.1. Hardware

The device consists of several electronic parts which have either been designed in-house or purchased from a manufacturer. The components are then placed in two different housing sections which are interconnected through a junction which also allows cables to pass through. These two different housings are also called Dry and Wet compartments. As the name suggests, the wet compartment houses the necessary sensors which are in direct contact with the outside air whereas the wet compartment is shielded from the external influence and houses the main microcontroller and battery. The image below shows the electronic components inside the two compartments. The top box marked by a red rectangle is the Dry compartment housing the main microcontroller, charging circuit and the battery. The bottom compartment marked by a blue rectangle is the wet compartment which houses sensors and supporting electronics.

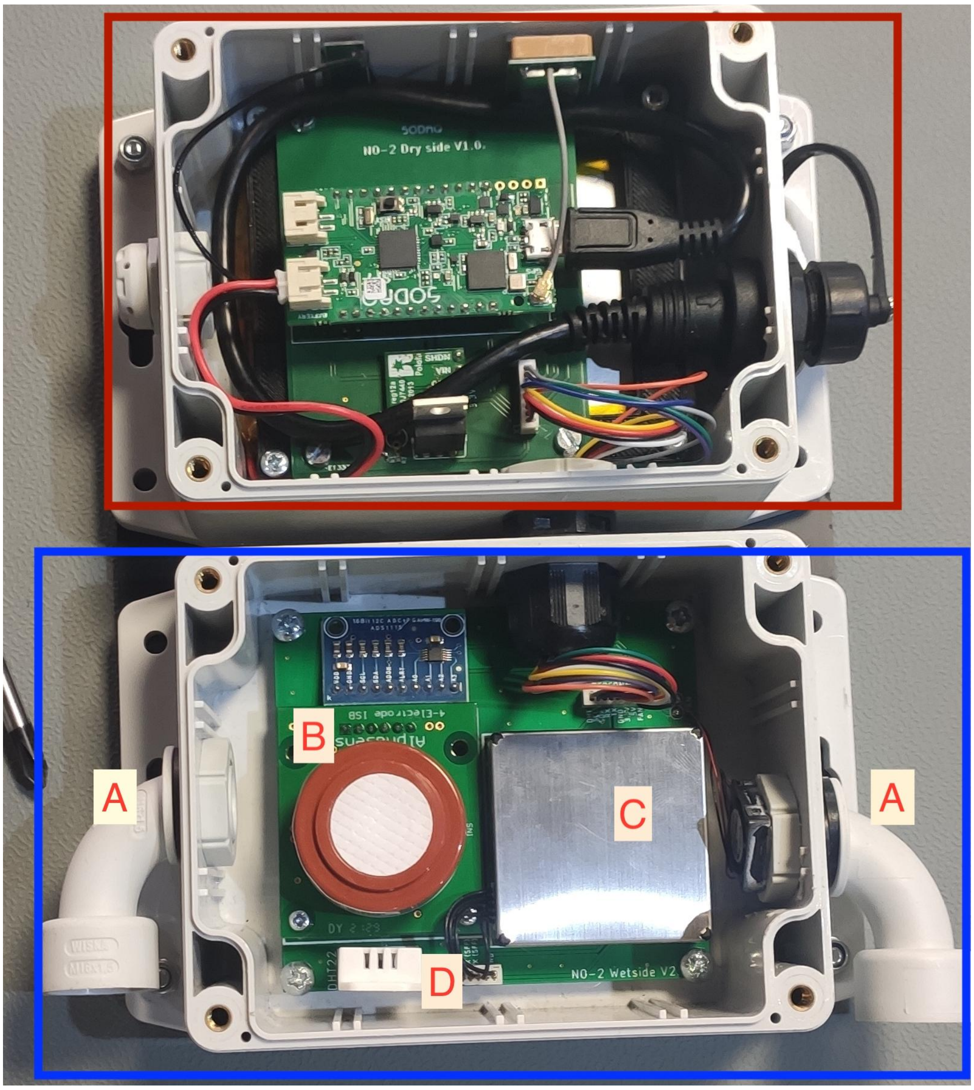


Figure 9: SODAQ NO2 sensor (Source: SODAQ)

a) vent for getting fresh air in the compartment
b) NO2 sensor from Alphasense
c) PM sensor
d) temperature and humidity sensor

As shown in figure 9, there are 3 main sensors in the device namely PM Sensor (SPS 30), NO2 sensor (Alphasense NO2-B43F) and temperature and humidity sensor (DHT 31). Details about the PM and NO2 sensors are discussed in detail in the first deliverable D3.1. The temperature and humidity sensor used in the device to provide temperature near the NO2 sensor which is required for calibrating its data.

There are two base PCBAs in each compartment which are used for routing the necessary connections and providing the mechanical support to connect all the components and keep them structured. The table below shows the list of main hardware components used in the NO2 device.

Table 4: *Hardware components of NO2*

Major Features	Component
Microcontroller	SODAQ SFF (ATSAMD21G18, 32-Bit ARM Cortex M0+)
Modem	Ublox SARA R410M
GPS	UBlox EVA 8M/M8M
PM Sensor	SPS30
NO2 Sensor	Alphasense NO2-B43F
Temperature and Humidity sensor	DHT 31
Battery power management	Pololu 5V U1V11F5
Power Source	Lithium Ion Battery 6Ah

Power Source

The NO2 device uses a 6Ah Li-ion Polymer Battery as shown in the figure below.

Figure 10: Li-ion battery used for NO2 device (Source: SODAQ)

Table 5: Specifications of the Li-ion battery

Feature	Specifications
Full Charge time	60-120 min (charge cycle dependent)
Charge cycle life	300-1000
Self discharge / month	3%
Safety	Safety consideration required
Fire Hazard	Medium
Operating temperature °C	-20 to +60
Charging temperature °C	+5 to +45

Device Charging

The device can be charged using a micro-USB cable. It takes around 12 hours to fully charge the device. The device continues to operate while still on charging thus making it suitable to be used in static use cases.

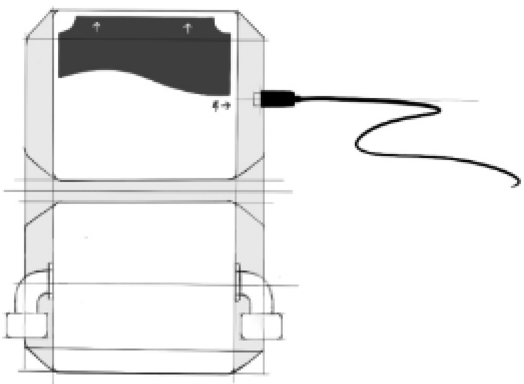
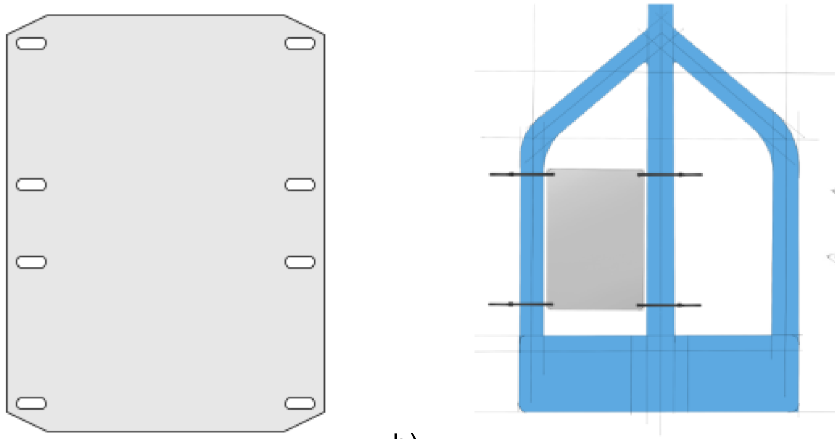


Figure 11: NO2 device charging (Source: SODAQ)

Casing and Mounting

The device uses two pieces Multicomp boxes which are available in the market. The boxes were chosen to expedite the design of the NO2 devices and as they are available readily. The boxes have very good plastic quality and have an IP rating of IP65. The two small bent pipes are used to get outside air inside the box, so that the box structure has no impact on the measurements.

Figure 12: Multicomp box for NO2 (Source: SODAQ)



a) b)

Figure 13: a) back mount for the boxes
b) mounting the device to the bike with the help of tie-wraps (Source: SODAQ)

3.2.2. Software

The NO2 device software is developed for taking care of all the necessary functionalities. It is an Arduino based firmware. The firmware has two modes of operations namely debug mode and operational mode. The debug mode is accessible when some core settings on the device needs to be changed and can only be accessed by opening the device.

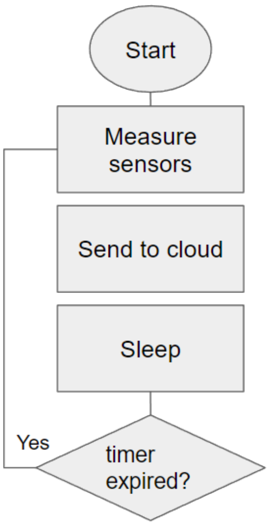


Figure 14: Software algorithm of NO2 device

While in operation the device always boots up in the operational mode, where it checks its battery power. If the battery is sufficient to operate then it starts with warming up the PM sensor and checks if all the other sensors are present. Once the data from the sensors is measured it sends data to the cloud. Next, it goes to the deep sleep mode where it stays for 1 minute. Therefore the data sending frequency of the NO2 device is every minute. Sometimes, if the device is not able to get a valid GPS fix within 1 minute, the difference between two consecutive readings could be more. The time interval between two readings can be adjusted using the boot menu. Currently the device is also not very power efficient as it doesn't have a low power sleep state when it is not measuring any data.

There are currently no user indications present on the device through means of LEDs. In the next iteration of the device, this would be evaluated to see if it is possible to make the device more user friendly through means of visual indicators.

3.2.3. Data

The table below shows the various data points sent by the device and its description. The device currently sends only minimum information that is essential for the operation of the device. The details about exact protocol and API calls are not yet there and are currently under development.

Table 6: *Data output structure of the NO2 device*

Data Value	Example	Unit	Description
IMEI	354679091436066		Unique ID for the device
Software Version Major	1		Major version of the software
Software Version Minor	0		Minor version of the software
Software Version Revision	1		Info about revision of the software
Timestamp	1661352634304	epoch	Timestamp from the device
Battery	4.19	Volts	Battery voltage
Board Temperature	28	Celsius	Temperature near the board
Latitude	52.0757963		GPS latitude
Longitude	5.1858017		GPS Longitude
Altitude	15		GPS altitude
Speed	0	km/u	Speed given by GPS
Satellite count	5		Satellite count as seen by GPS
Time to fix	15		Time required by GPS to get a valid fix
PM 1	0.07	0,3 - 1 µg/m3	particulate matter <1um
PM 2.5	0.08	0,3 - 2,5 µg/m3	particulate matter <2.5um
PM 10	0.08	0,3 - 10 µg/m3	particulate matter <10um
NO2	228	µg/m3	NO2 concentration as measured by the board
NO2 WE	223	mV	voltage of the working electrode
NO2 AE	-16	mV	voltage of the auxiliary electrode
Temperature	21	Celsius	Outside temperature
Humidity	45	%	Outside humidity

3.3. Main component & Costs

Air has been designed to be a repair friendly device and if required most parts could be easily replaced without damaging the device. The casings have been designed at SODAQ and use minimal components to avoid waste at the end of life cycle of the device. This also helps to reduce the cost of the device

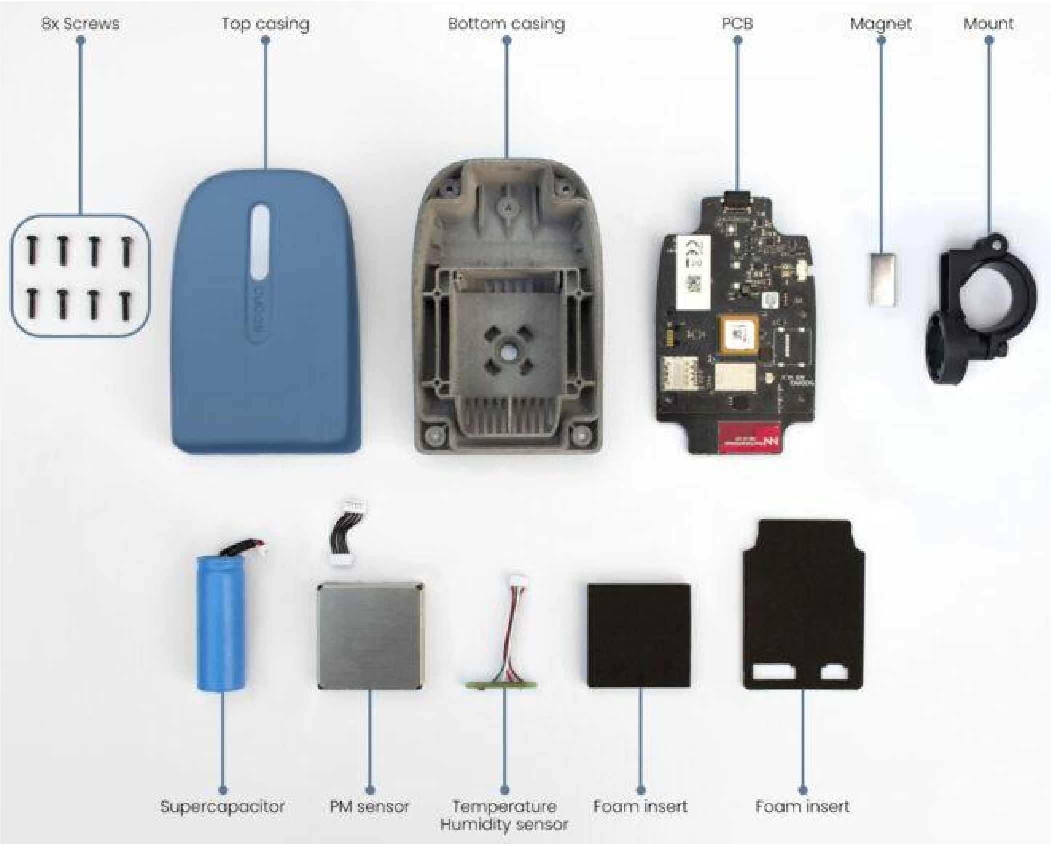


Figure 15: Components of the Air device

The AIR was created with a Creative Commons (CC) licence, to increase the reach and spread of the global air network. By creating the AIR with the CC licence, access to the complete schematics of the device and data are free of charge to individuals, learning institutions, and nonprofits. This allows other parties to manufacture or modify the sensor themselves and use the data to create changes for the betterment of air quality. Further, SODAQ is continuously trying to reduce the cost of the device by streamlining the production process and working with better production partners. As the production quantities are not very high, the device costs are higher than expected. The device costs also include connectivity costs and data package.

On the other hand, NO2 devices are still in the prototype stage, which means it has a very high production cost. The BOM costs alone are around 500 euros and it requires at least 5 hours to complete and test one full device. Through the CompAir project one of the goals is to improve the production process for the device, thereby directly reducing the production costs. Further, it will be evaluated which components in the device can be replaced or procured from other partners to reduce the BOM costs.

Summary AIR and NO2

As there are several similarities between the two devices even though the form factor is quite different, information about them has been summarised in the two tables below. They show the comparison between the AIR and NO2 devices and their capabilities when it comes to measuring the air quality.

Table 7: *The summary of Air Quality Parameters and device capabilities*

Air Quality Parameter	Description	Units	AIR	NO2
PM10	Particulate matter <10um (micrometres)	µg/m3 (micrograms per meter cube)	Yes	Yes
PM2.5	Particulate matter <2.5um	µg/m3	Yes	Yes
NO2	NO2 gas (respiratory pollutant)	µg/m3	No	Yes
T	Temperature	degC	Yes	Yes
H	Relative humidity	%	Yes	Yes

Table 8: *Physical characteristics of the two sensors*

Characteristic	Air	NO2
Photo		
Dimensions	height: 35 mm length: 85 mm width: 55 mm	height: 57,5mm length: 220 mm width: 160 mm
Weight	185 gms (200 gms including bike mount),	935 gms (1.05 Kg with mounting plate)
Charger Type	USB C	Micro USB B
Charging Time	2 Hours	12 Hours
Visual Indicator (LEDs)	Yes	No
Status	Production-ready	Beta version
IP Rating (Dry Areas)	IP 65	IP 65

***Table 9**: Requirements and their compatibility with the two devices*

Requirements	Comments
I want to get notified when sensor is out of spec and intervention is needed	Achievable through dashboard
I can easily charge my device daily (e.g. via USB, socket)	Possible to charge AIR using USB-C and NO2 micro USB . AIR devices charges with 2 hours whereas NO2 takes 12 hours
I can count on the device to last at least 4 hours without charging	Yes, that is possible for AIR and NO2 devices. Using a powerbank you can continuously measure.
I can carry the device while exercising without it bothering me	AIR 250 gms, NO2 is 950 gms (target is to reduce it by 100g or so), personally carrying NO2 device would be difficult due to weight.
I can expose the device to all normal weather conditions (e.g. rain, wind, hail, cold & warm temperatures) [Not waterproof]	Dry area is IP65. wet area is not but it is designed so that the sensor won't get damaged during rain.
I can attach the device to various objects (e.g. bicycle, clothing) and easily attach and detach it	Bike mount is available for both devices. For NO2 due to weight constraints it should be attached to the back side of bike using tie wraps
I can see if the device is working via a simple status light	Possible for AIR and not yet supported by NO2
I want to measure air pollution even at home	Possible but without Calibration
I want to know if a sensor is inside or outside	Will be possible through the dashboard

4. Product concept - Traffic counting (Telraam)

In this chapter, we elaborate on the technical design approach for the new and improved Telraam sensor, from a technical point of view. We distinguish between the hardware, the software and data output.

In brief, the Telraam sensor consists of a printed circuit board (PCB) encased in a custom-built casing and LCD-screen. The sensor runs an adapted and retrained AI-algorithm to detect passing traffic. Data communication is condensed to aggregated counts per 15’, allowing for low-cost data transmission to the cloud using the cellular network.

This approach allows for a small, easy to install traffic counting device at minimal cost.

4.1. Hardware

Main PCBA

The key element to highlight here, is that we opt to design our own PCB which is specifically designed for the functionalities the Telraam sensor needs to be able to do such as xxx. This means we can keep the component cost low, only requiring those components that are needed.

The choice of NPU (neural processing unit) is a crucial part of this project and must be reviewed and considered in detail if it is sufficient for the use case. This has a direct impact on the choice of camera and the mechanical integration but the choice of NPU does not impact the design and testing of the rest of the sensor such as the modem and sensors.

Therefore it is important to split the image processing HW from the rest of the electronics so the design of the IOT part with interface, sensors and modem can continue (with possible testing of the UI/UX and BE) separately.

The only requirement would be that the NPU talks SPI in a master/slave setup and the data structure is predefined (JSON, CSV, ...). Power and other lines such an interrupt, reset and others can be checked later.

For the NPU chip, we opt for the K210. AI SIPEED NPU "cores" are equipped with 67 pin M.2 edge connectors, in combination with NRF9160 chip set governing the operations and communications. This allows for an integrated design of the PCB with sensors (compass, GPS), modem and detection via the NPU (K210)

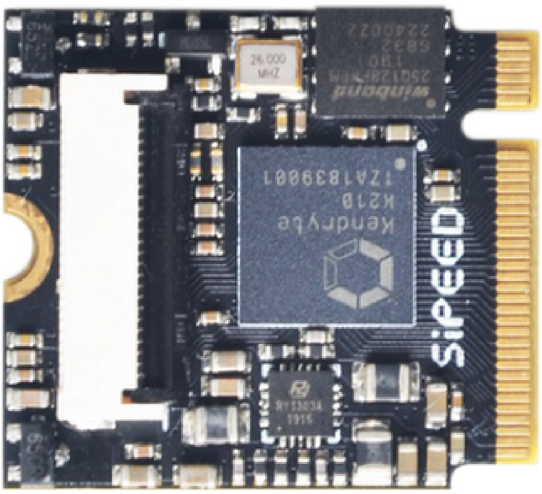


Figure 16: K210 Chip

The NRF9160 is the 2nd important chipe the “master” chips that governs the sensor, communication and the K210. We’ve chosen for this chip-type mainly because of the communication capabilities:

- Quad-band 850/900/1800/1900MHz
- GPRS multi-slot class 12/10
- GPRS mobile station class B
- Compliant to GSM phase 2/2+

- Class 4 (2 W @ 850/900MHz)
- Class 1 (1 W @ 1800/1900MHz)
- Support GPS/GLONASS/BD
- Weight: 1.5g
- Control via AT commands (3GPP TS 27.007, 27.005 and SIM Com enhanced AT Commands)
- Supply voltage range 3.4 ~ 4.4V
- Low power consumption
- Operation temperature:-40℃~85℃

The NRF is only 2G-capable, so is “old tech” but 2G is sufficient if we only work with small data packages. Moreover, this chip is available in large stocks, is proven technology in other projects, reliable and cheap.

The K210 & NRF are the key components on the PCB. Other components are more general in nature (SIM-card, connector for the display). The PCB will also include a compass module and GPS, for automatic location and orientation, so there's no need for the user to select the road segment where the sensor is active, thus avoiding another user intervention.

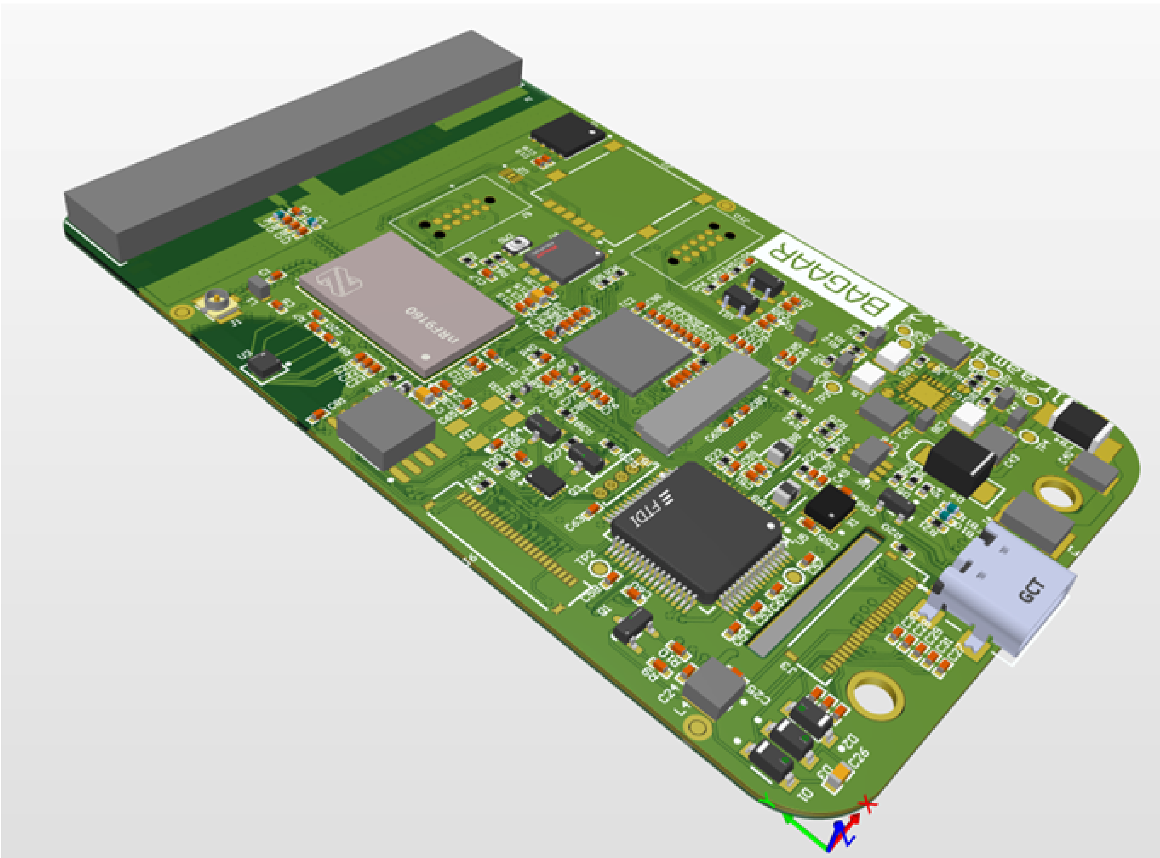


Figure 17: 3D render of the customised PCB

Display

The sensor will include a basic display and button allowing the user to interact directly with the sensor. As we are limited with the amount of memory available to display on the screen, only basic information such as total counts of the past day and a simple QR-code to initiate the installation will be displayed.

We summarise the content of the display requirements as follows:

1. 5 counting screens defined.
    a. One "overview counting screen" contains all 4 traffic categories (pedestrians, bicycles, cars, heavy trucks).
    b. One counting screen per category.
2. The "registration screen" consists of the QR code (unique per device) and corresponding text.
3. The "GPS searching" screen is defined but might become obsolete in the future
4. Telraam asked to include a screen to inform the user of a firmware upgrade
5. The "no-vision" screen shows a cat with a box on its head, and indicates that the camera does not have a good vision (e.g. low contrast).
6. The "Error" screen can be used to display an error message. The actual error message can be raised by the device or by the backend.
7. The date (e.g. "Fri Feb 25") will be displayed on some screens (the 5 counting screens) once it is available.

Optics

We choose a fixed wide angle camera-lens, integrated in the sensor so the sensor consists of a single unit as opposed to the old Telraam sensors which had a computing unit and camera separately. More importantly, the use of a wide angle lens allows for covering the full Region of Interest (RoI) of the street where counting needs to be done. Because of this, there is no need for the user to set the camera angle manually, as the detection of the RoI will be done on the software side automatically, again taking away an important user-interaction step, prone to error in the old sensor.

Casing

Given the use of a custom PCB and integrated camera-lens in the sensor, the only option is to design a casing that is customised to the sensor. No off-the-shelf casing exists for this type of configuration. Multiple 3D-printed options are available, but to keep unit-cost low, it is best to work with injection moulded plastic casings. However, this does require the design and procurement of an (expensive) mould a priori.

The sensor casing design is such that the least amount of individual components are needed with a front, a back , a separate button and shackle (for mounting on the window)

Pictures below give an impression of the complete hardware set, with the PCB, the integrated lens and the casing:

Figure 18: Telraam v2 sensor prototype

4.2. Software

The key decision here is the use of a completely new detection algorithm compared to the existing Telraam sensor. The concept is still to perform full edge-computing (i.e. immediate and local processing on the device itself), both for reasons of performance and managing communication requirements as well as privacy reasons. As with the current sensor, edge-computing ensures only privacy-secure counting data can leave the sensor and no privacy-sensitive images are transmitted to a central database for central processing. Whereas with the current Telraam sensor, the processing is split into conversion to (privacy-secure) object properties such as size, axis ratio of the object (device-side) which are then converted to traffic type counts using more complex algorithms (central database), the new sensor will immediately convert objects to counts on the device itself. So the concept of edge-computing is maintained, but the detection algorithm on the device fundamentally changes.

Instead of using Python openCV background correction, using open libraries, we develop an AI-based detection algorithm, based on Yolo, yet adapted and re-trained so the trained model fits on the (constrained) NPU-chip (the K210). In short, the flow of software services on the sensor for the detection of traffic can be summarised in picture below:

In brief, live pictures are immediately processed using a (pruned) Yolo v3 neural network to identify traffic on each frame. The model distinguishes the following traffic types:

1. Car
2. Bike
3. Pedestrian
4. Large truck
5. Light truck
6. Bus

It's imperative to maintain a high frame-rate in the detections to ensure proper counts as sensors can be installed quite close to the road and passing traffic can be fast, thus allowing only very little time to count passing traffic. There is a tradeoff to be made between performance and accuracy, both in terms of the size of the neural network as well as the size of the model allocated to detecting (per frame) and vs. the script governing the tracking (over frames).

As such, after detection on a per frame basis, the second step is linking of the objects over multiple frames (tracking), ensuring that the object identified in frame$_{T0}$ is the same as the object detection in frame$_{T+1}$.

The identified objects are communicated (internally between the K210 and NRF-chip) as counting data and stored/governed on the sensor in the NRF-chip and then sent to the cloud via the 2G-network using Thingstream as our IoT-communication partner (details in the next section). To avoid large data packages being sent, data is converted to (encrypted) binary data and only transmitted to the cloud every 15'.

To avoid overly technical documentation of the description of the software in this report, this section is deliberately kept light. The development of the initial detection script was done in cooperation with Kapenrikov; more technical details of the software approach of the detection script are included in this blogpost.

To conclude the section on the software, it is worth mentioning that firmware updates are possible Over-The-Air (OTA), allowing us to regularly update the detection script from our central cloud platform to the sensors and keeping track of version on the deployed sensors.

4.3. Data

Telraam is an integrated combination of the sensor(s) collecting data and the web-platform to publish the data. In this sense, the data from the new sensor(s) is FIRST sent to a central database, governed by Telraam, and SECONDLY available for third party use via the open API[1], calling the Telraam cloud, not the sensors directly. The connectivity of (Telraam) sensor to (Telraam) cloud is -in principle- not important for third parties. Nevertheless, in this section we explain in brief the approach for the communication between sensor and cloud.

[1] https://telraam-api.net/

As indicated in the previous section, we opted to collaborate with an existing IoT communication partner called Thingstream. Thingstream is an IoT "Communication-as-a-Service" service provider, agnostic about the approach to getting data between IoT devices and the enterprise. Thingstream builds on top of the foundation of high-performance MQTT and develops a Data Flow Manager, allowing simple processing, transformation, and integration of messages into the enterprises, such as Telraam. The communication protocol we've chosen for Telraam is the cellular network, not LORA. Currently, cellular networks have better coverage and reliability worldwide compared to LORA.

The data communication between the Telraam sensors and the Telraam cloud is closed and not directly accessible for third parties. Therefore, it is key to have a well documented and complete API on the Telraam cloud that gives access to all sensor data, indirectly.
As indicated in "D3.1 Sensor Strategy, Requirements Report", the Telraam data-infrastructure designed for the current Telraam sensor, will be used as is and further adapted to also ingest and process data from the new sensor. The current API is used by third parties and is well documented and will be extended with new API end-points.

The new sensor will collect data on a temporal granularity of 15' so this requires some (minor) adaptations to the API.

Documentation of the API[2] is done in PostMan. An example end-point for the raw traffic counting data is copied below. The call returns a JSON data table, with the following columns:

- **instance_id**: the instance identifier for "instance" level calls ("-1" for "segment" level calls)
- **segment_id**: the segment identifier for "segment" level calls (in the future when "instance" calls are implemented too, this will read "-1")
- **date**: ISO timeflag (date and UTC time) of the reporting interval (beginning of the interval)
- **interval**: can be "hourly" or "daily" for hourly or daily aggregate data, respectively
- **uptime**: between 0 and 1, represents the portion of the reporting interval (hour or day) that was actively spent counting the traffic (background calculation intervals in ohurly periods, and the night time in daily periods contribute to values being less than 1)
- **heavy**: the number of heavy vehicles (called lorry in older APIs, but all stand for the same: anything larger than car) on this day (and in this hour)
- **car**: the number of cars
- **bike**: the number of cyclists
- **pedestrian**: the number of pedestrians Then the counts for left and right directions are also reported separately, followed by
- **direction**: "1" - disregard, this is an internal consistency value making sure that when multiple cameras on different sides of the street are aggregated then the left and right directions are handled properly

[2] https://telraam-api.net/

- **timezone**: The name of the Time zone where the segment can be found, which can be used to convert reported UTC timestamps to local times
- **car_speed_hist_0to70plus**: the estimated car speed distribution in 10 km/h bins from 0 to 70+ km/h (in percentage of the total 100%)
- **car_speed_hist_0to120plus**: the estimated car speed distribution in 5 km/h bins from 0 to 120+ km/h (in percentage of the total 100%)
- **v85**: the estimated car speed limit in km/h that 85% of all cars respect (15% of drivers drive faster than this limit). Just like all other speed related measurements, the accuracy of this value is likely not better than +/-10%.

The API will further be developed in the project and more endpoints will be added and adapted as per the requirements of the dashboards etc. For example, a flag will be added to distinguish between V1 & V2 sensors and more traffic categories will be included.

4.4. Main component & Costs

At this point, we don't know yet what the final cost of the sensor will be. Cost can be broken down into the following components:

1. Electronics component cost (Bill-Of-Material or BOM)
2. PCB assembly cost including testing
3. Casing - either split by a fixed cost for the mould and the plastic used, or with a single unit cost per casing, including the write-off for the mould assuming a minimum order quantity
4. Packaging/logistics/shipping costs/storage/handling
5. Data communication & cloud-hosting.

BOM-cost expected to be well below 50€ per unit as we specifically have chosen low cost components. The most expensive component is the K210 (10-20€); all other individual components, including the LCD screen do not exceed 10€ per unit.

For the PCB assembly, the cost is unclear. We expect 20-70€ per unit, depending on the order quantity. For the casing, the cost of plastic is expected to be well below 1€ per unit. The cost of the casing is in the mould which has to be purpose built and is a unique piece. A variety of moulds exists, with a large variety of quality and performance (i.e. low cost moulds for max 2000 units or high cost moulds for 10.000+ units). Common moulds (steel) for this application are expected to cost in the range of 25.000€. The per unit cost then greatly depends on the amount of units that will be ordered.

Also, it is not uncommon to offset the investment in the mould into the unit price of the casing, taking into account a minimum purchase obligation. For example, a mould of 25k€ with a minimum purchase obligation of 1000 units, translates to 25€ for the casing on a per unit basis. To be on the safe side, we expect a cost for the casing between 15-35€ per unit.

Packaging (basic cardboard) and handling/storage represent a small cost as well. It is not possible at this stage to determine a per unit cost, as it will depend greatly on the amount of sensor ordered/handled/shipped. Conservatively, we expect a per unit cost of packaging and all handling at 10-15€/unit.

Summing all components up, while final cost estimates are impossible at this point, we do expect the unit cost of the sensor to be in the range of: 75€-175€ per unit.

Finally, the above components relate to the cost of the sensor itself, not the cost of operations. Both sending the data via the 2G network as well as processing and storing the data in the cloud are expected to cost 10-20€ per sensor per year in total.

5. Calibration and standardisation

5.1. Air quality sensor calibration strategy

Compact low-cost air quality sensors used in citizen science projects have simple measurement principles, and as such, suffer from sensitivity to environmental influences (temperature, humidity, interfering pollutants), as well as drift and sensitivity changes during their deployment.

Default calibration methods such as factory calibration involve measuring sensor response to a known pollutant concentration. Using 2 pollutant concentrations (e.g. 0 and 100ppm), sensor response is measured and the sensor is calibrated using a linear signal - pollutant concentration response curve. This calibration method may lead to accurate estimation of pollutant concentration under similar conditions as those used during the lab tests, but leads to a decrease in accuracy the longer the sensor is deployed in outdoor conditions with varying environmental influences.

Another method to calibrate sensors involves co-locating all low-cost sensors to be deployed at a reference station with high-end measurement equipment; such as a beta attenuation mass monitor for particulate matter or a chemiluminescence monitor to detect NO2 gas. After a co-location period of typically a few weeks, sensors are placed at the location of interest. Ideally, after the measurement at location the sensors are placed back to the reference station for a second co-location period. The data gathered during the co-location periods from the low-cost sensors (LCS) and reference sensors are used to train a model, usually multilinear model with raw sensor data, temperature, humidity and cross-interfering pollutants of interest as variables. In the case of particulate matter additional correction can be performed to account for droplet formation around the pollutant particle of interest which leads to overestimation of particle size, according to Köhler theory. While co-location calibration leads to improved concentration estimations than factory calibration in outdoor conditions, a disadvantage is that moving the sensors can be labour-intensive, and due to the limited sensor lifetime co-location periods decrease the time sensors can be used in the region of interest. Additionally, seasonal changes can significantly affect the calibration validity [3], so the algorithm trained in one set of environmental conditions has varying performance when applied to sensor data from a different set of conditions.

In CompAIR, we plan to use a novel sensor calibration method: distant calibration. Firstly, a few co-located sensors and several sensors in the region of interest are deployed simultaneously. This ensures that they go through a similar process of ageing in the field.

Pollutants that are generated to a great extent due to human activity such as particulate matter and NO2 have lower and relatively stable concentrations during nighttimes due to low human activity, and are fairly homogeneously distributed in an area compared to the daytime. The distant calibration algorithm, previously validated using several types of particulate matter and NO2 sensors [1,2], filters out out-of-range sensor data ("sanity checks") and uses a 34-day moving window of nighttime LCS data, combined with ground truth sensor data for training a multilinear model. Ground truth sensors are reference sensors located at a distance to the field sensors, which are selected if they are within a kilometre radius (typically 15km, extended if there are insufficient number of reference stations in close proximity), have high correlation with the LCS measurements.

Then, the parameters extracted from the training model are applied to calibrate LCS data of the next day in real-time using the cloud environment. This ensures that dynamic changes in the microenvironment around the sensors have a minimum effect on the calibration performance, as the calibration takes place in real-time. Performance of the calibration is evaluated using the reference co-located sensors; which allows corrections for drift and ageing (loss of sensitivity) into account. When performing the calibration on the co-located sensors, reference sensor next to the LCS is excluded as an option as the ground truth sensor. A schematic representation of the calibration algorithm workflow is in the the following figure:

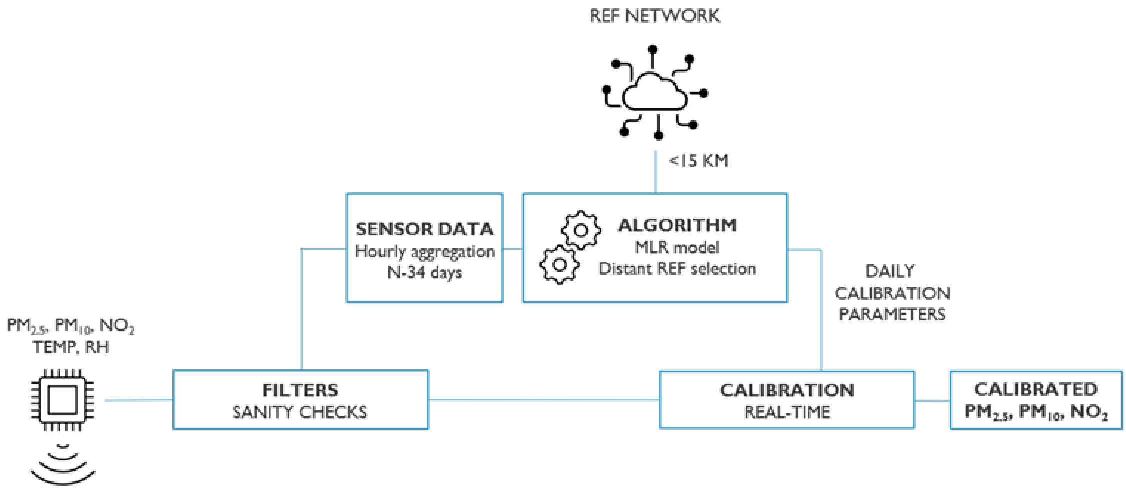


Figure 19: Schematic representation of the calibration algorithm workflow

This workflow and algorithm was developed and validated in projects performed in Belgium and The Netherlands. [1,2] Within CompAIR, sensors will be deployed in 4 distinct pilot regions with varying microclimates (warm and dry in Sofia/Plovdiv and Athens compared to mild Berlin and Flanders), and varying availability and sparser distribution of reference sensor data. Therefore we plan to validate the distant calibration approach, evaluate its added value and improve it for CompAIR use cases. Different training approaches, including machine learning will also be evaluated.

5.2. Data standardisation

A common pitfall in research is data management: when the generated data is locally saved, insufficiently labelled and missing context information which is a barrier against the re-use of data. Data standardisation brings sensor data into a common, consistent format that enables users to interface with, process and analyse the data. As mentioned in D3.1, in CompAIR we intend to use the non-proprietary [Open Geospatial Consortium (OGC)] SensorThings API data model for air quality data. Below is a diagram that shows how different components of sensor data and metadata are included in the SensorThings API data model:

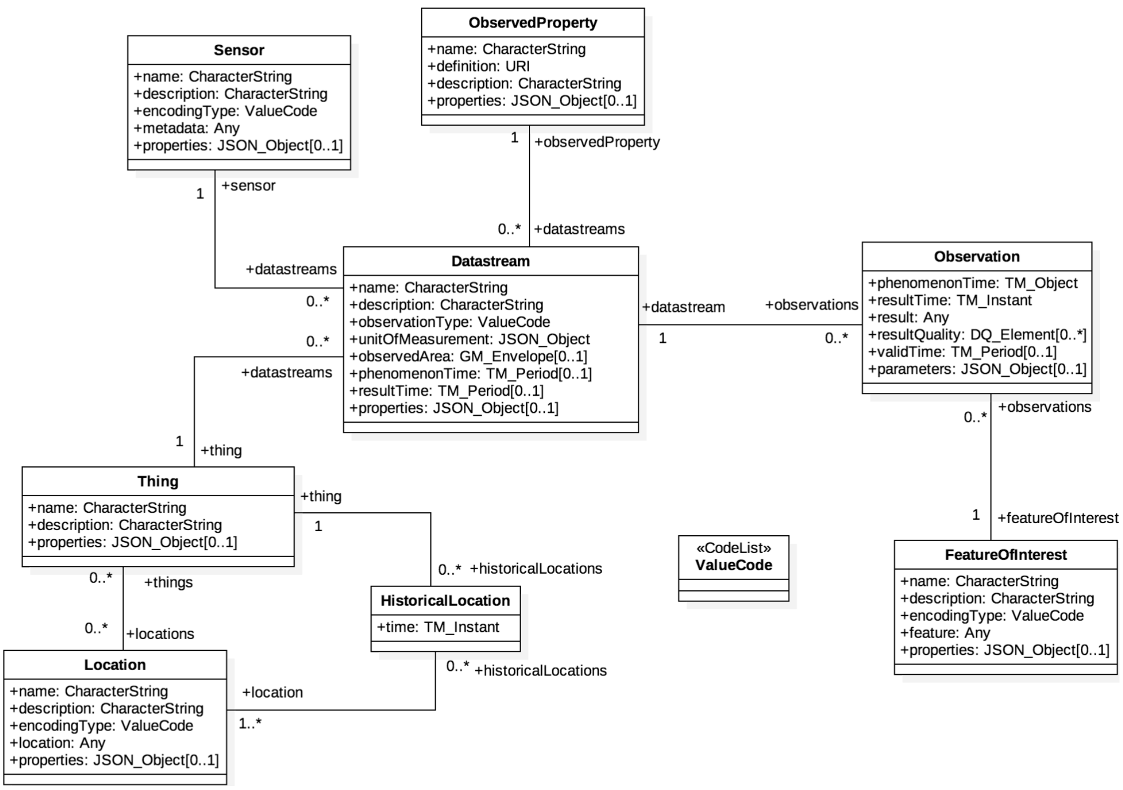


Figure 20: Components of sensor data and metadata

In SensorThings API data format, a **Thing** (such as an IoT device), is linked to a Datastream, which is a collection of **Observations** grouped by the same **ObservedProperty** and **Sensor**. A Thing can have a **Location**, **HistoricalLocation**, multiple **Sensor**s and **Datastreams**. An **Observation** is an act that produces a resolt whose value is the estimate of a target of interest (**FeatureOfInterest**). By applying this format to SODAQ Air data, we obtain the example data from a PM1 sensor datastream (Appendix 3).

As shown, the data format allows explicitly specifying sensor location, measured unit, measured property, type of sensor, location, observation and timestamp. SODAQ Air and NO2 data will be converted to the OGC SensorThings API format within the imec calibration platform. Using the data model will allow easy sharing of data accompanied by the metadata and querying the data using different attributes as needed, regardless of sensor type.

6. Conclusion

One of the key goals of the COMPAIR project is to develop IoT devices for citizens which can be used to measure air quality and to monitor traffic in an urban environment. In this deliverable, based on the requirements set by the pilots, a sensor and data integration strategy has been proposed. There are two main IoT devices that need to be developed for COMPAIR pilots to be used by the citizen science participants. First one is a portable air quality monitoring device and the second one is a traffic monitoring device that can be easily installed at home on windows.

In the case of the Air Quality monitoring device, after analysing the requirements set by the pilots for various parameters to be measured, it was seen that Particulate Matter (PM), NO_2 , temperature and humidity are the most important ones. The choice of sensors and requirements set by the pilots led to the conclusion that the current version of the SODAQ AIR and SODAQ NO2 are the best product concepts for the Air quality monitoring device.

Next in terms of development of the traffic monitoring solution, it was found that Telraam offers many advantages over the currently available traffic counting options used in a citizen science setting. The latter are mostly manual counting using mobile or web-apps. Providing the existing Telraam sensor is changed, focusing on using it more user friendly and accessible for non-expert users, a new Telraam sensor is found to the most reliable and scalable option.

As part of the COMPAIR, IMEC develops air quality sensor calibration algorithms based on combining low-cost sensor data, environmental data and reference grade sensor data located at a distance to the low cost sensors. Next, the data from the Air quality sensors and Traffic monitoring will be processed in the cloud and information would be made available to COMPAIR visualisation platforms. In terms of data standardisation, a suggestion is made to implement the Open GeoSpatial Consortium (OGS) SensorThings API which is a non-proprietary, platform-independent international standard to interconnect IoT devices, data and applications over the web.

Finally, it can be seen that the products' concepts of described in this report of combined air quality and traffic monitoring solution will be used by citizen science participants. In the next phase of the project, all devices will be tested in the closed testing round performed by pilot projects.

7. References

[1] Hofman J., Nikolaou M., Stroobants C., Weijs S., Shantharam S.P., La Manna V.P. (2021). Distant calibration of low-cost PM and NO2 sensors; evidence from multiple sensor testbeds . Env Sc &Tech.

[2] Hofman J, E. Nikolaou M, Huu Do T, Qin X, Rodrigo E, Philips W, et al. (2020). Mapping Air Quality in IoT Cities: Cloud Calibration and Air Quality Inference of Sensor Data. IEEE SENSORS 2020 Conference Proceedings. IEEE; 2020.

[3] Ratingen, S. van, Vonk, J., Blokhuis, C., Wesseling, J., Tielemans, E., & Weijers, E. (2021). Seasonal influence on the performance of low-cost no2 sensor calibrations. Sensors, 21(23), 7919. https://doi.org/10.3390/s21237919

Annex 1: Operation behaviour SODAQ AIR

The device has two different types of operational mode depending on whether the device is in a state of motion or static. Further, the behaviour will also be affected by the fact if the device is connected to a charger. One of the key requirements for the device to function is the presence of a magnet to activate the sensor. Once the device receives the Magnet input, it will transition further to the PM sensor Warm up state, where it will start preparing to collect and send data. When the magnet is not present, then there is no motion detection as well. The diagram below shows the main functionality of the device.

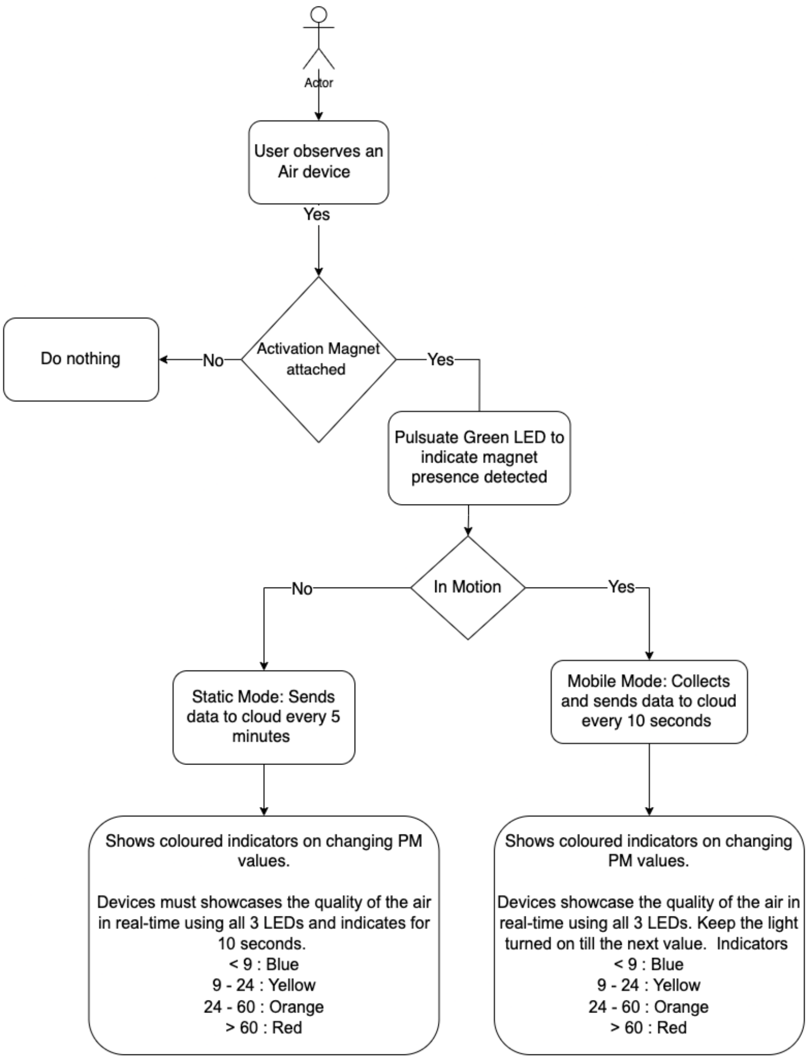


Figure 21: Functioning of Air device

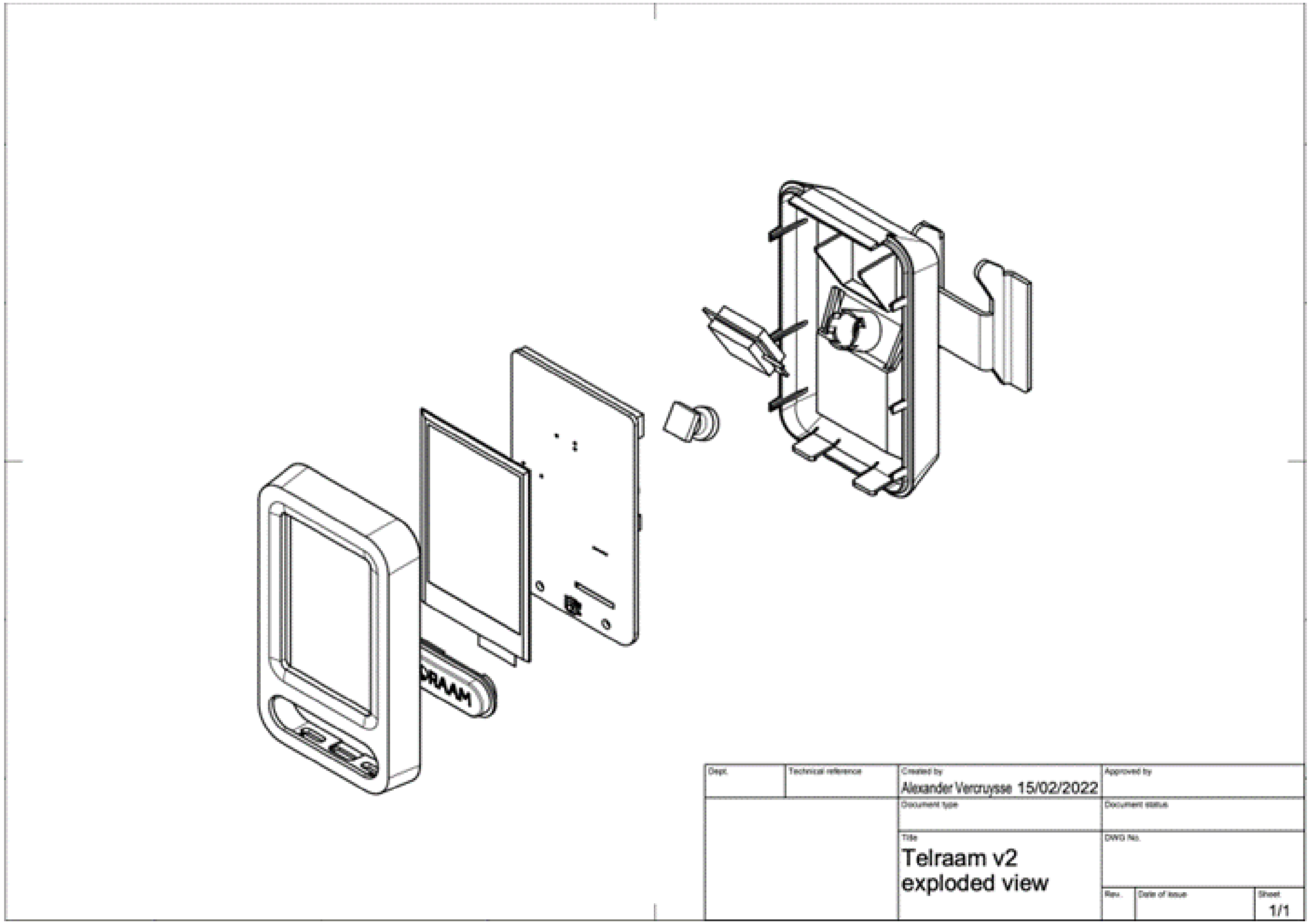
If the device is connected to a charger while also being activated it affects the LED indication (discussed below). In addition to the charging indication, it will also show the PM value indications. Also, it is important to note that the magnet input is also used for the motion detection.

LED behaviour

The Air device is a user friendly device and utmost attention has been given to LED indications to help the user

- The LED will pulsate WHITE while it is trying to perform FOTA and throughout the FOTA process.
- When the magnet is present and the device is static the following colours will be displayed for 10 seconds, depending on the PM reading:
    - PM < 9 : Blue
    - 9 < PM <24 : Yellow
    - 24 < PM < 60 : Orange
    - PM > 60 : Red
- When the magnet is present and the device is moving the following colours will be displayed until a new measurement is made:
    - PM < 9 : Blue
    - 9 < PM <24 : Yellow
    - 24 < PM < 60 : Orange
    - PM > 60 : Red
- When the device is charging:
    - If the magnet is not present then the device will display a pulsating green light throughout the charging. It will stop indicating the colors when the battery is full.
    - If the magnet is present and the device is static, it will then pulsate a green light interrupted from the PM level light indication described above, for 10 seconds.
    - If the magnet is present and the device is in motion, then it will display the green pulsating light indication only during the warm-up phase of the PM sensor and then it will go on displaying the PM level colors from above.
- When the device is not charging:
    - Display battery status LED, only when the device starts to move, while the device is warming up the SPS30 sensor
    - If the battery is too low then the LED indicator will pulsate red. Battery is considered to be low below 3150mV.

Annex 2: Telraam v2 exploded view



Annex 3: Example data from a PM1 sensor


```
{
        "name": "352656108696991",
        "description": "SODAQ sensor box for CompAir project",
        "properties": {
                "owner": "SODAQ",
                "maintainer": "SODAQ",
                "confidentiality": "public",
                "project": "CompAir",
                "areaDescription": "Mobile sensor",
                "deploymentCondition": "Mounted on a bicycle",
                "imei": 352656108696991,
                "imsi": 232031721761704,
                "iccid": 8943030172211761704 1,
      "version": "v0.1.0",
      "versionMajor": 0,
      "versionMinor": 1,
      "versionRevision": 0
        },
        "Locations": [
                {
                        "name": "Mechelen",
                        "description": "City and municipality in the province of Antwerp in the
Flemish Region of Belgium",
                        "encodingType": "application/vnd.geo+json",
                        "location":
                        {
                                "type": "Point",
                                "coordinates": [51.020430674538794, 4.483268910354823]
                        }
                }
        ],
        "Datastreams": [
                {
                        "name": "pm_1p0 raw",
                        "description": "Raw sensor output of sensor for particulate matter ≤ 1.0
µm",
                        "observationType":
"http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement",
                        "unitOfMeasurement":
                        {
                                "name": "Microgram per cubic meter",
                                "symbol": "µg/m3",
                                "definition": "https://qudt.org/vocab/unit/MicroGM-PER-M3"
                        },
```

```
				"Sensor":
				{
					"name": "Sensirion SPS30",
					"description": "Sensirion particulate matter sensor",
					"encodingType": "application/pdf",
					"metadata":
"https://sensirion.com/media/documents/8600FF88/616542B5/Sensirion_PM_Sensors_Datasheet_SPS30.pdf"
				},
				"ObservedProperty":
				{
					"name": "MassDensity",
					"definition": "https://qudt.org/vocab/quantitykind/MassDensity",
					"description": "The mass density or density of a material is its mass per unit volume."
				},
				"Observations":
				[
					{
						"phenomenonTime": "2022-07-18T12:00:00Z",
						"resultTime": "2022-07-18T12:00:00Z",
						"result": 30.0
					},
					{
						"phenomenonTime": "2022-07-18T13:00:00Z",
						"resultTime": "2022-07-18T13:00:00Z",
						"result": 15.0
					}
				]
			},
	]
}
```