

POLYTECHNIC OF TURIN

Master's Degree course in Electronic Engineering

Master's Degree Thesis

A sampling device for Particulate Matter monitoring

An innovative optical solution



Supervisor:

prof. Marco Parvis

Co-Supervisors:

prof. Simone Corbellini

Candidate:

Francesco Vitiello

April 2018

To my parents

Abstract

The quality of the air we breathe every day is a fundamental factor for the wellness of the environment and human health. Suspended in the air, composed mainly of Nitrogen and Oxygen, we can find some particles, mostly in the solid state, which take the name of *Particulate Matter (PM)*. They have very different physico-chemical characteristics and can cause various problems when they penetrate the respiratory system and react chemically with the surrounding tissues.

Nowadays, the pollution issue presents a high level of criticality due to the increase in anthropogenic activities that directly introduce polluting particles into the atmosphere (industries, domestic heating, vehicular traffic, etc.). On the other hand, it is understood that it is fundamental to have a monitoring network for the concentration of atmospheric particulate, in order not to exceed the levels considered dangerous for health. One of the many problems with this type of measurement is that it is strongly local. In fact, by its nature, the particulate tends to fall to the ground close to the source of production, specifically if the particle weight is high. This means that in the same area it is possible to obtain significantly different values of PM concentration depending on the sensor position. This phenomenon is increased in the absence of windy currents or with a high humidity level.

The aim of this work is the development of a new technology for the measurement and characterization of atmospheric particulate. In fact, the realized device provides data not only regarding concentration but also its dimensional distribution and a spectrographic analysis of the single particles. Not less important is the economy and wireless connectivity: in fact it is possible to create a monitoring network much more widespread than those installed today, in order to obtain more detailed information on the air pollution level.

During the development phase a measurement campaign was conducted in two different cities: Turin and Caserta. For each of them 11 samples were collected, analyzed and compared with the official data provided by the respective Regions. Starting from these data, the system was calibrated and set up.

From the analysis of the data obtained it is possible to notice that there is a quite significant correlation between the data obtained and the actual levels of pollution. On the other hand, different results have been obtained for the density calculation, confirming the need to make further improvements to the system in order to get more precise data.

Contents

List of Figures	iii
List of Tables	v
I General overview	1
1 Introduction	2
2 The Particulate Matter: background	5
2.1 Definition and classification	5
2.1.1 Dimensional classification	6
2.1.2 Chemical composition	9
2.2 Sources of production	10
2.2.1 Primary sources	11
2.2.2 Secondary sources	12
2.3 Pollution effects	13
2.3.1 Effects on human health	13
2.3.2 Effects on vegetation and ecosystems	17
2.3.3 Effects on climate and micro climate	18
2.3.4 Effects on visibility	19
2.3.5 Effects on materials	20
2.4 Regulatory framework	20
3 PM measurement techniques: state of art	23
3.1 The gravimetric techniques	24
3.1.1 Working principles	24
3.1.2 Measurement procedure	25
3.2 OPC	28
3.3 DMA, CPC and DMPS	30

II	Case study	33
4	PM monitor:	34
4.1	Device overview	34
4.1.1	Part list	35
4.1.2	Working cycle	40
4.2	Control unit	41
4.2.1	RaspberryPi Zero W	41
4.2.2	Interface board	42
4.3	Image processing software	43
4.3.1	Preprocessing	44
4.3.2	Detection	49
4.4	Grid architecture	53
5	Calibration and data analysis	55
5.1	Calibration	55
5.1.1	Pump calibration	56
5.1.2	Software calibration	59
5.1.3	System trans-characteristic	64
5.2	Data analysis	64
6	Conclusions	71
A	Drawings	73
B	Software: client application	76
B.1	Screenshots	76
B.2	Main.py	77
C	Software: server application	90
C.1	Main.py	90
C.2	Analyzer.py	92
C.3	Camera.py	93
C.4	Motor.py	94
C.5	Pump.py	95
C.6	Leds.py	96
C.7	Client.py	97
C.8	Utils.py	97
	Bibliography	99

List of Figures

2.1	Aerodynamic diameter.	8
2.2	Particulate matter dimension. [1]	9
2.3	PM 2.5: decrease in life expectancy in 2000s. [2]	14
2.4	PM penetration in the respiratory system. [3]	15
2.5	Leaf damage after exposure to 1.5 ppm of SO ₂ for 7 hours. [3]	18
2.6	Palazzo Rivaldi in Rome: example of black crusts. [4]	20
3.1	Impactor head.[5]	25
3.2	Measurement cycle of gravimetric technique. [6]	27
3.3	OPCs working principles.	29
3.4	DMA schematic diagram. [7]	31
3.5	CPC schematic diagram. [8]	31
4.1	Device block diagram.	34
4.2	Portable device	38
4.3	Automatic device	39
4.4	RaspberryPi Zero W.[9]	41
4.5	Control unit	43
4.6	OpenCV logo.[10]	44
4.7	Additive RGB trichromy of a real image.[11]	45
4.8	2D Gaussian distribution.[12]	46
4.9	Gaussian blurring with different σ values.[13]	47
4.10	Detection steps.	52
4.11	Output histogram.	52
4.12	Measuring grid architecture.	54
5.1	Calibration instrumentation.	56
5.2	Measurement with the pump connected to the collecting chambers.	57
5.3	Pump vacuum profile.	58
5.4	First test images.	60
5.5	Second test images.	60
5.6	Size calibration test.	61

5.7	Data collected in Caserta.	65
5.8	Data collected in Turin.	65
5.9	Analysis of data trends.	67
5.10	Electronic microscope photos.	70
5.11	Sample chemical composition.	70
B.1	Client software screenshots.	76

List of Tables

2.1	General characteristics of the natural and anthropogenic particulate. .	10
2.2	Main components of primary particulate ($d < 2.5$ m).	11
2.3	Main components of primary particulate ($d > 2.5$ m).	12
2.4	Main precursors of secondary particulate.	12
4.1	Light source and correspondent output channel.	45
5.1	Vacuum profile data.	58
5.2	Second test: data	62
5.3	First test: data	63
5.4	α evaluation from data collected in Caserta.	66
5.5	α evaluation from data collected in Turin.	67
5.6	Density ρ evaluation from data collected in Caserta.	68
5.7	Density ρ evaluation from data collected in Turin.	69
5.8	Chemical analysis data.	70

A sampling device for Particulate Matter monitoring

An innovative optical solution

Candidate: Francesco Vitiello

Supervisor: Marco Parvis

Co-supervisor: Simone Corbellini

Summary

I. INTRODUCTION

The air is a mixture of gases composed mainly of Nitrogen, Oxygen, Carbon Dioxide plus small quantities of other gases such as Argon and Helium. Suspended in this mixture, we find mostly solid small particles that take the name of *Particulate Matter (PM)*.

They have very heterogeneous physico-chemical characteristics and represent the main source of pollution in urban and industrial areas. The effects related to these substances are different and range from climatic variations to damage to vegetation and architectural works, but the most important is related to human health.

Nowadays is fundamental considering the air pollution issue: the development of a device that collect and analyze some of the different aspects of the particulate matter is the first step to improve the quality of the air and, consequently, our lives.

II. PM MEASUREMENT

The M.D. 60/02 provides a reference method for the particulate matter measurements. It is based on the gravimetric technique and defines all the technical characteristics and operating procedures to be performed.

After being conditioned to 20 °C and 50% relative humidity, a fiberglass filter is weighed by means of an analytical balance and subsequently exposed to a forced air flow for a period of about 24 hours. Once the exposure is finished, the filter is reconditioned at the initial temperature and humidity and weighed again. The PM concentration is calculated from the difference in weight of the filter and on the volume of air to which it has been exposed. Data are usually provided in [$\mu\text{g}/\text{m}^3$].

Although this is the standard procedure, considered the most reliable and precise, it is however subject to important sources of error: the filter in all its movements can be easily contaminated, during the second conditioning phase some particles deposited on the filter can evaporate altering the measurement, the impactor head can get dirty easily preventing the correct selection of the particles. A further disadvantage is that the data are not real-time available but only after a few days and each sensor is designed to measure only a specific particle size. Furthermore the device cost is high and its dimensions large, preventing the installation and maintenance of a large number of measurement sites.

Simultaneously to the gravimetric technique several technologies have been developed for particle measurement, the most important are the Opacimeters (OPC) and the Differential Mobility Analyzer (DMA). The former exploits the interaction between the particle under examination and a light source, usually a laser beam. The DMA instead is based on particle separation according to their electrical mobility.

One of the many problems with this type of measurement is that it is strongly local. In fact, by its nature, the particulate tends to fall to the ground close to the source of production, specifically if the particle weight is high. This means that in the same area it is possible to obtain significantly different values of PM concentration depending on the sensor position. This phenomenon is increased in the absence of windy currents or with a high humidity level.



Figure 1: Commercial gravimetric particulate meter (Instrumex IPM-FDS- $2.5\mu/10\mu$).

III. AN INNOVATIVE SOLUTION

The device made at the *Politecnico di Torino* laboratories is a hybrid between an optical particle counter and a more classic gravimetric meter. Specifically, it

adopts an optical system to count and recognize the particles while it uses the techniques and the filters of the gravimetric systems to capture them. The device was made in two different versions: the first one portable and manually operated, the second one is completely automatic and complies with the IP55 standard.

In both devices an air flux is forced from the external environment on a sampling fiberglass filter by a diaphragm pump. An optical system, composed by a camera and a Macro close-up lens, takes photos of the filter surface at different back-light wavelength. Finally, a dedicated software running on a RaspberryPi Zero analyzes the photos and detects the particles. The collected data can be transmitted on a remote server via *WiFi 802.11* connection, or stored on the internal memory.

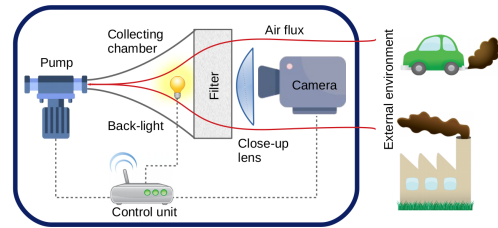


Figure 2: Device block diagram

The innovation of this technology is to provide data not only with regard to the concentration of particulate matter, but also its dimensional distribution and a spectrographic analysis of the single particles. Not less important is the economy and wireless connectivity: in fact it is possible to create a monitoring network much more widespread than those installed today, in order to obtain more detailed information on the air pollution level.

IV. DATA ANALYSIS

During the development phase a measurement campaign was conducted in two different cities: Turin and Caserta. For each of them 11 samples were collected, analyzed and compared with the official data provided by the respective Regions. Starting from these data, the system was calibrated and set up. The graph in Figure 4 shows the data obtained from the measurement campaign. The red line represents the data provided by the device while the blue one are those provided by the Regions. It is possible to notice that there is a quite significant correlation between the data obtained and the actual levels of pollution except for a value probably due to an analysis error.

The software provides the data in $[\mu\text{m}^2/\text{m}^3]$, it is therefore necessary to find a conversion factor that reports the data in the most common form $[\mu\text{g}/\text{m}^3]$. This coefficient was obtained by calculating, for each measure, the ratio between the reference value supplied by the Region and that obtained by the device. Then the arithmetic mean of the results was performed. The mean value is $\bar{\alpha} = 2.72 \cdot 10^{-4}$ and its standard deviation is $\sigma_{\alpha} = 1.67 \cdot 10^{-4}$.

This factor must be used only to compare the trend of the data obtained by the instrument with those provided by the Regions. It is not a direct conversion factor.

The trans-characteristic of the system is:

$$C = \left[\frac{Q \cdot A_p}{A_e} \cdot \frac{t}{1000} \right]^{-1} \cdot P_x \cdot \alpha \quad (1)$$

Where Q is the pump flow-rate, A_p and A_e the photographed and the exposed

area respectively, t the time exposure in minutes, P_x the total area in square micrometers of all detected particles and α the conversion factor.

V. CONCLUSIONS

The results obtained are very satisfactory, also considering that the device is still under development. The system features make this solution a valid low-cost alternative to the commercial air sampling devices. In order to improve the quality and resolution of the system, new optical solutions are being currently tested. At the same time, a new measurement campaign is underway to gather new data to be used for system calibration.

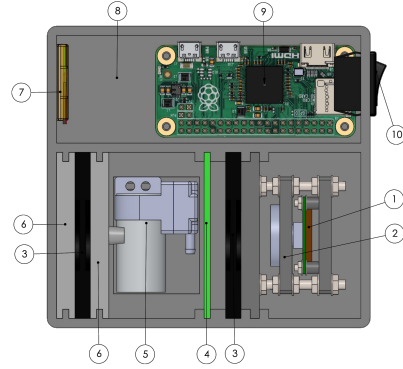


Figure 3: Portable device. Camera(1), Close-up lens (2), Filter (3a-3b), Leds (4), Air pump (5), Collecting chamber (6a-6b), Battery (7), Box (8), Control unit (9), Power switch (10)

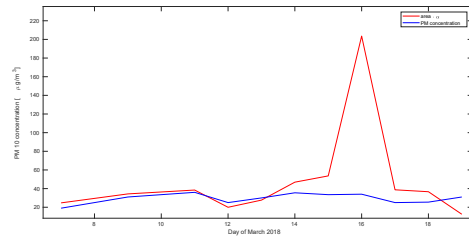


Figure 4: Collected data

Part I

General overview

Chapter 1

Introduction

The air that we commonly breathe is essential for the life of living organisms: it is in fact a source of the oxygen necessary for the processes of energy production that are the basis of life and cellular activity.

It consists of 78.9% of Nitrogen (N_2), 20.9% of Oxygen (O_2) and 0.2% of Carbon Dioxide (CO_2) and noble gases such as Argon (Ar). The remaining impurities are Hydrogen (H_2), Methane (CH_4) and Helium (He).

Air pollution is determined by the presence in the air of one or more undesirable or foreign substances, in quantity and for a duration such as to alter the healthiness of the air itself and constitute a danger to health.

Referring to the global quantity of all pollutants emitted, we note that five of them alone contribute more than 95% of the total. These pollutants are: Carbon Monoxide (CO), Sulfur Dioxide (SO_2), Nitrogen Oxides (NO_x), hydrocarbons and particulate matter. The concentration of these five pollutants, in addition to that of Ozone (O_3), is used as an index of air quality and the laws set the maximum values that these concentrations can reach .

The growing sensitivity of public opinion and institutions towards environmental issues has stimulated in recent years the development of methodologies for the collection and analysis of atmospheric pollutants.

A very important area of research is the one related to fine particles that, according to the APAT (Agenzia nazionale per la Protezione dell'Ambiente e servizi Tecnici, National Agency for Environmental Protection and Technical Services), represent the most homogeneously dispersed pollutant in the area.

Because of their intrinsic characteristics, they can remain for a longer time in the atmosphere, representing a danger to the health of humans, animals, plants and goods. In fact, there is a vast scientific literature including epidemiological, clinical and toxicological studies that testify the dangers of these substances on health both for short and long exposure times in correspondence with typical levels found in

urban areas of industrialized and non-industrialized countries.

If we consider the quantity of air that is daily breathed by a man, we can better understand its importance for health and the risks related to the respiration of polluted air.

In fact, an adult man breathes:

- **In rest condition:** from 6 to 9 liters of air per minute
- **During a moderate physical activity:** 60 liters per minute
- **During intense physical activity:** 130 liters per minute

In many studies in the literature, the particles in suspension (and especially the smaller fractions such as PM10 and PM2.5) are the indicator of air quality most frequently associated with a series of adverse health effects .

Studies have shown a measurable excess of clinical outcomes due to particles pollution. These effects are both acute, ie they occur in the population on days when the concentration of pollutants is higher (aggravation of respiratory and cardiac symptoms in predisposed subjects, acute respiratory infections, bronchial asthma crisis, circulatory and ischemic disorders), are chronic, that is, they occur as a result of long-term exposure (chronic respiratory symptoms such as cough and phlegm, pulmonary capacity decrease, chronic bronchitis, etc.).

Furthermore, studies conducted in the United States and in many European countries have highlighted an association between the levels of air pollutants and the daily number of deaths or hospitalizations due to respiratory and cardiovascular causes.

The measurement of air quality is useful to ensure the protection of the health of the population and the protection of ecosystems. The Italian legislation, built on the basis of the so-called European mother directive (Directive 96/62/EC implemented by Legislative Decree 351/99), defines that the Regions are the competent authority in this sector, and provides for the subdivision of the territory into areas and agglomerations on which assessing compliance with target values and limit values. One of the major problems affecting the measurement of particulate concentration in the atmosphere is its "locality". In fact it is possible to have very different levels of concentration in a relatively small geographical area. This condition requires a monitoring infrastructure consisting of a network of sensors that extends throughout the territory in question.

Today this is not possible because the techniques used and recognized as valid and reliable by law have very expensive equipments and laborious procedures of use and maintenance. Thus, increasing the detection locations would increase costs exponentially.

The aim of this work is to develop a new type of sensor for measuring atmospheric particulate concentration.

It captures the airborne particles through the filters typically used in the gravimetric technique while the analysis is performed optically. The device in question is able to provide as output data on both the concentration per unit of air and the dimensional distribution.

During the entire development process, we aimed for the realization of a device that had the following characteristics: economy, easy reproducibility, low power consumption and finally long operating autonomy.

Chapter 2

The Particulate Matter: background

The aim of this chapter is to introduce the problem of pollution due to particulate matter by defining its characteristics, production sources and the impact on human health. Particular attention is also paid to the legislative aspects that regulate the maximum acceptable levels both in Europe and in Italy.

2.1 Definition and classification

With the term particulate (Particulate Matter, PM) or Total Suspended Powders (TSP) reference is made to the set of solid and liquid particles dispersed in the atmosphere with a diameter between some nanometers (nm) and tens / hundreds of micrometers (μm).

The particulate consists of a complex mixture of substances, of an organic or inorganic nature, present in the solid or liquid state which, due to their small size, remain suspended in the atmosphere for longer or shorter times; among these we find different substances such as sand, ash, powders, soot, silicae of various kinds, vegetable substances, metallic compounds, natural and artificial textile fibers, salts, elements such as carbon or lead, etc.

According to the nature and size of the particles we can distinguish:

- **Aerosol:** consisting of solid or liquid particles suspended in air with a diameter of less than 1 micron.
- **Mists:** given by droplets with a diameter of less than 2 microns.
- **Exhalations:** consisting of solid particles with a diameter of less than 1 micron and released usually from chemical and metallurgical processes.

- **Smoke:** given by solid particles usually with a diameter of less than 2 microns and transported by gas mixtures.
- **Powders:** solid particles with diameters between 0.25 and 500 microns.
- **Sand:** given by solid particles with a diameter of more than 500 microns.

There is no an unambiguous way to describe the properties of the PM, neither for its classification. It is a heterogeneous pollutant whose characteristics mainly depend on:

- Size
- Chemical composition
- Production source

Following are described some of the most common classification approaches.

2.1.1 Dimensional classification

The dimensional classification is certainly one of the most used when trying to characterize the particulate matter. Before defining in detail its characteristics it is important to dwell on the two main actors: the particle and the fluid in which it is immersed, the air.

Assuming to have a particle of spherical shape immersed in a fluid, it is subject to a weight force and Archimedes' one expressed by eq. 2.1 and eq. 2.2 respectively:

$$F_g = mg = \frac{\pi d^3}{6} \rho_p g \quad (2.1)$$

$$F_A = -\frac{\pi d^3}{6} \rho_f g \quad (2.2)$$

where m , d and ρ_p are the mass, the diameter and the density of the particle, ρ_f is the fluid density and g the gravitational acceleration.

The resultant force is given by the sum of the two previous ones:

$$F_{tot} = F_g + F_A = \frac{\pi d^3}{6} (\rho_p - \rho_f) g \quad (2.3)$$

If the particle is in conditions of laminar motion with the fluid in which it is immersed, with a Reynolds' number less than 10^4 , it is subjected to a viscous friction force defined by the Stokes' law:

$$F_d = -6\pi\eta r v = -3\pi\eta d v \quad (2.4)$$

where r and d are the radius and the diameter of the particle respectively, η is the viscosity and v is the relative speed between fluid and particle. At steady-state condition the resultant of the forces is null:

$$F_g + F_A + F_d = 0 \quad (2.5)$$

From the equation 2.5 it is possible to obtain the speed as a function of the particle diameter:

$$\frac{\pi d^3}{6}(\rho_p - \rho_f)g = 3\pi\eta dv \rightarrow v_c = \frac{d^2 g(\rho_p - \rho_f)}{18\eta} \quad (2.6)$$

The Stokes' equation 2.4 is valid only for particles that see the fluid as if it were continuous, ie for those particles whose dimensions are much larger than the mean free path λ . For smaller particles a correction factor is needed, the Cunningham coefficient:

$$C = 1 + \frac{\lambda}{d} \left[2.51 + 0.80 \exp \left(-0.55 \frac{d}{\lambda} \right) \right] \approx 1 + 2.51 \frac{\lambda}{d} \quad (2.7)$$

$$F_d = -3\pi\eta d \frac{v}{C} \quad (2.8)$$

In the air at 25°C, $\lambda = 70$ nm:

$$v_c = 2.97 \cdot 10^{-5} \text{ m s}^{-1} (d/\mu\text{m})^2 \left(1 + \frac{0.17}{d/\mu\text{m}} \right) \quad (2.9)$$

For $d \gg 0.17$ the Cunningham factor is negligible (laminar viscous flow regime):

$$v_c = 2.97 \cdot 10^{-5} \text{ m s}^{-1} (d/\mu\text{m})^2 \quad (2.10)$$

For $d \ll 0.17$ the 1 factor in the parenthesis is negligible (molecular flow regime):

$$v_c = 5.00 \cdot 10^{-6} \text{ m s}^{-1} (d/\mu\text{m}) \quad (2.11)$$

Considering a particle having $\rho_p = 1 \text{ g cm}^{-3}$ and $d = 80 \mu\text{m}$, we are at laminar flow limit conditions and the critical gravitational speed is:

$$v_c = 2.97 \cdot 10^{-5} \text{ m s}^{-1} (d/\mu\text{m})^2 \approx 0.2 \text{ m s}^{-1} \quad (2.12)$$

this means that:

- if $d > 80 \mu\text{m}$ the particles sinks quickly
- if $d < 80 \mu\text{m}$ the particles persist suspended in the atmosphere for a long time

The considerations made so far refer to a perfectly spherical particle immersed in a fluid. What happens in reality is that the particulate does not have a regular shape and therefore it is necessary to define the dimensional variable that we want to use in a univocal way: the aerodynamic diameter.

"The aerodynamic diameter of an irregular particle is defined as the diameter of the spherical particle with a density of $1000 \text{ kg} \cdot \text{m}^{-3}$ and the same settling velocity as the irregular particle."[\[14\]](#)

Generally we have:

$$0.25 < \frac{D}{d} < 2.5 \quad (2.13)$$

where D is the real diameter and d the aerodynamic one.

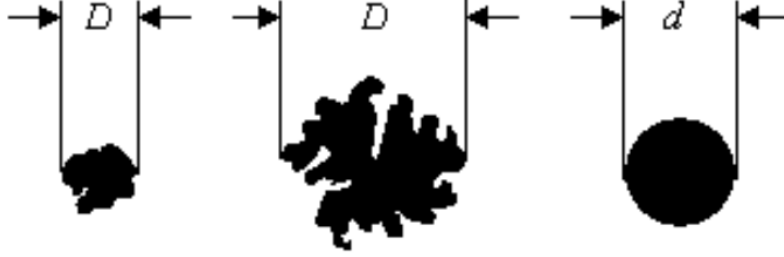


Figure 2.1: Aerodynamic diameter.

With the introduction of the aerodynamic diameter it is possible to define a first dimensional classification of the PM:

- **PM₁₀**: particles with aerodynamic diameter less than $10 \mu\text{m}$
- **PM₅**: particles with aerodynamic diameter less than $5 \mu\text{m}$
- **PM_{2.5}**: particles with aerodynamic diameter less than $2.5 \mu\text{m}$

It is also a convention to divide the atmospheric particulate according to the diameter aerodynamic in the following fractions:

- **Ultrafine**: aerodynamic diameter between 0.01 and $0.1 \mu\text{m}$; generally these particles consist of products of homogeneous nucleation of supersaturated vapors (SO_2 , NH_3 , NO_x).
- **Fine**: aerodynamic diameter between 0.1 and $2.5 \mu\text{m}$; their formation occurs by coagulation of ultrafine particles and through the gas-particle process or by condensation of gas on pre-existing particles in the accumulation interval.

- **Coarse:** aerodynamic diameter between 2.5 and $100\ \mu\text{m}$; essentially produced by mechanical processes (erosion, mechanical or wind resuspension, grinding), they contain elements present in the soil and in the sea salts; being also relatively large they tend to settle in a few hours or minutes, often finding themselves close to the emission sources according to their height.

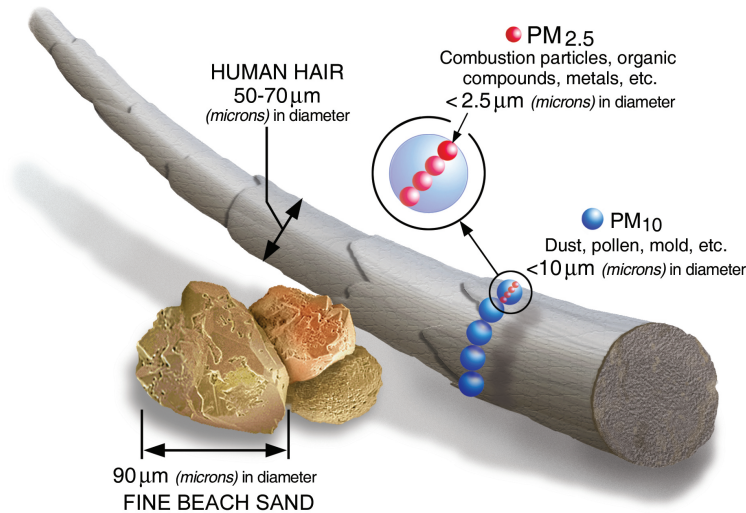


Figure 2.2: Particulate matter dimension. [1]

2.1.2 Chemical composition

The particulate, unlike other types of pollutants, is not uniquely characterized from the chemical point of view. In fact it is composed of a multiplicity of substances present in the atmosphere.

Among these substances the main ones are hydrogen ions, sulphates, nitrates, ammonium, carbon, water and other materials of the earth's crust. The elements listed above are not uniformly distributed in the dimensional analysis of the particulate. In fact the fine fraction of the particulate is mainly composed of sulfates, hydrogen ions, ammonium, elemental carbon and organic compounds. The larger particles instead are composed mainly of calcium, aluminum, silicon, magnesium, iron and pollen.

The chemical composition and the size of the particles produced also depends on the particulate sources. Taking for example Potassium: it is present in the fine fraction of the particulate if it is generated as a result of fires while it is present in the coarse fraction if it is the product of soil erosion. In the same way, the nitrate present in the fine particles is the product of the reaction between nitric acid and

ammonia in the gaseous phase, while in the coarse particles it is due to the reaction of the nitric acid with the pre-existing solid particles.

Some studies conducted to determine the metal take-up in the atmosphere have shown marked differences in concentration between urban, industrial and rural areas.

2.2 Sources of production

The atmospheric particulate can be divided into three large groups: primary PM, secondary PM and re-suspended PM. They differ mainly in the method of generating aerosol particles and for the chemical composition of the latter.

Analyzing the sources of production, however, they can be divided into two categories:

- **Natural sources:** are, for example, the rock and soil particles eroded, raised or resuspended by the wind, organic matter and ashes from forest fires or volcanic eruptions, plants (pollen and plant debris), spores, marine sprays, insect remains, etc.
- **Anthropogenic sources:** they are mainly due to the use of fossil fuels (production of energy, domestic heating), emissions from motor vehicles, tire wear, brakes and road surfaces, various industrial processes (refineries, chemical processes, mining operations, cement works) and waste disposal (incinerators). Large quantities of powders can also arise as a result of various agricultural activities. These sources are further subdivided into stationary and mobile.

The table 2.1 shows the main characteristics of the particulate generated by both types of sources.

Table 2.1: General characteristics of the natural and anthropogenic particulate.

	Natural	Anthropogenic
Physical properties	coarse diameters irregular shapes	fine diameters regular forms
Chemical composition	carbonate salts iron and aluminum oxides silica compounds crust minerals	sulphates nitrates organic compounds of the lead hydrocarbons
Production processes	Erosion wind transport marine spray	building agriculture industrial activities combustion

Both natural and anthropogenic sources can be divided into primary and secondary, generating primary and secondary particulate respectively.

2.2.1 Primary sources

For primary sources we mean those sources able to generate and to introduce particulate particles directly into the atmosphere. The generated particles is in turn divided into two granulometric fractions: particles with an aerodynamic diameter $d > 2.5 \mu\text{m}$ and particles with $d < 2.5 \mu\text{m}$.

The former are called coarse particles while the latter are fine particles.

Table 2.2: Main components of primary particulate ($d < 2.5 \mu\text{m}$).

Components	Natural sources	Anthropogenic sources
Sulphate SO_4^+	marine spray	combustion of fossil fuels
Nitrate NO_3^-	-	-
Ammonium NH_4^+	-	-
Organic Carbon CO	natural fires	motors food cooking
Elementary carbon	natural fires	wood combustion
Minerals	erosion	diffused sources
Metals	volcanic activity	foundries brake consumption
Bioaerosol	viruses and bacteria	-

Table 2.3: Main components of primary particulate ($d > 2.5$ m).

Components	Natural sources	Anthropogenic sources
Sulphate SO_4^+	marine spray	-
Nitrate NO_3^-	-	-
Ammonium NH_4^+	-	-
Organic carbon CO	humus	tire and asphalt wear
Elementary carbon	-	tire and asphalt wear
Minerals	erosion	diffused sources
Metals	erosion	-
Bioaerosol	pollen bacteria	-

2.2.2 Secondary sources

The secondary particulate is that which is formed as a result of chemical-physical reactions in the atmosphere. The secondary sources are therefore the sources that directly enter the precursors of the secondary particulate into the atmosphere.

Table 2.4: Main precursors of secondary particulate.

Components	Natural sources	Anthropogenic sources
Sulphate SO_4^+	oxidation of sulphides volcanism and fires	combustion of fossil fuels (SO_2 oxidation)
Nitrate NO_3^-	soil erosion solar radiation natural fires	motors combustion of fossil fuels
Ammonium NH_4^+	wild animals (NH_3) soil erosion	farm animals (NH_3) waste water fertilized soils
Organic carbon CO	vegetation (hydrocarbon oxidation)	motors wood combustion

2.3 Pollution effects

2.3.1 Effects on human health

Atmospheric particulate today is one of the main risk factors for human health, especially in urban and industrial areas. In the last decades, numerous studies have been carried out on the link between air pollution and health status.

Most of the pathologies found concern the respiratory, cardiovascular, neurological, immunological, haematological and reproductive systems. A very alarming fact is that the onset of pathologies also occurred in acute respiratory and cardiovascular conditions even for levels of particulate concentration well below the limits set by law by the World Health Organization.

The problem of pollution is also reflected in the economic field as it involves an increase in the cost of public spending in terms of health. It is estimated to be responsible for the premature death of around 800,000 people worldwide. In the analyzes shown below, the main correlation indices between pollution, human health and human health are mainly based on life expectancy, premature deaths, purchase of drugs and hospital admissions.

Examining the data on anthropogenic emissions of particulate matter in the atmosphere in 2000, the CAFE (Clean Air For Europe) program reported a value of 348,000 premature deaths per year due to the exposure of pm25 only in Europe.

Life expectancy also drops by about a year or even two if you are in particularly polluted countries like Belgium, northern Italy, Poland or Hungary. Figure 2.3 shows a map of Europe in which different zone are highlighted based on the reduction in life expectancy.

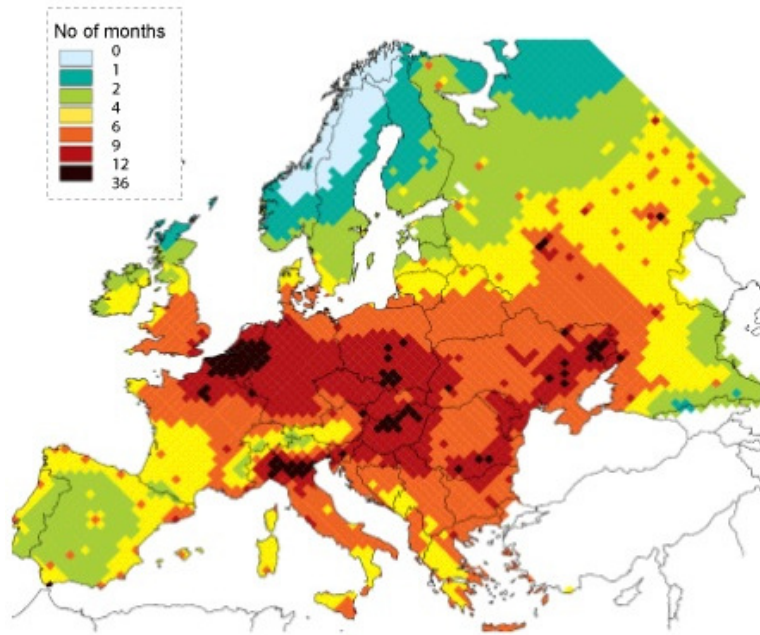


Figure 2.3: PM 2.5: decrease in life expectancy in 2000s. [2]

One of the systems most affected by suspended dust is definitely the respiratory one. The most common disorders among exposed people are:

- decrease in lung function
- irritation of the upper airways
- aggravation of asthmatic disorders
- chronic bronchitis
- tumors

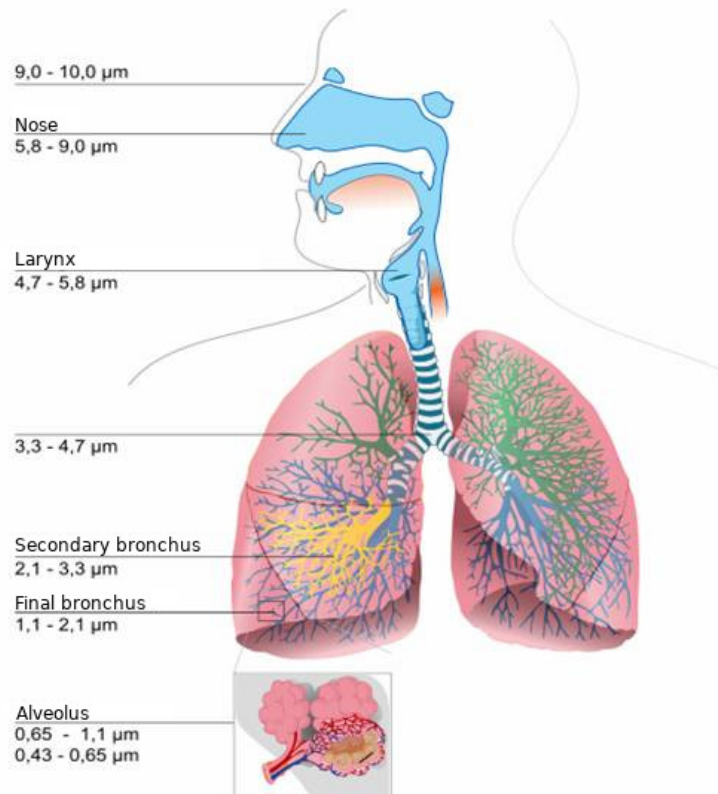


Figure 2.4: PM penetration in the respiratory system. [3]

The polluting particles enter the human body through the main airways that are the nose and the mouth. This process is called incorporation. It is composed in different phases and the modalities depend mainly on the dimensions of the particles under examination. The main phases are inhalation, deposition, retention or possible expulsion.

By analyzing the respiratory system more thoroughly, it is possible to divide the latter into three parts:

- **Extrathoracic region:** it is the upper part of the respiratory system that includes the nose and the mouth up to the larynx
- **Tracheo-bronchial region:** this section includes trachea and bronchi and for this reason it is also called the intermediate region
- **Alveolar region:** the last section consists of the ramifications of the bronchi up to the pulmonary alveoli

Considering Figure 2.4, it can be noted that not all particles of particulate can penetrate the same way within the respiratory system. In fact, the latter constitutes the first natural "filter" against the pollution of the air, blocking the particles from the largest to the smallest, step by step. Depending on the penetration capacity, a new classification of atmospheric particulate can be defined.

It is defined by the UNI ISO 7708 directive which subdivides the particulate into:

- **Inhalable fraction:** it is the total mass of particles that can enter through the nose and/or mouth
- **Extrathoracic fraction:** it consists of particles that can not penetrate beyond the larynx
- **Thoracic fraction:** it consists of particles that can penetrate beyond the larynx
- **Tracheo-bronchial fraction:** it is the total mass of the particles that are able to pass the larynx but which can not reach the alveoli
- **Respirable fraction:** it is the total mass of the particles that can reach the alveoli

Once in the respiratory system, the particles of particulate can be subjected to the phenomenon of adhesion. That is, they can come into contact with any surface of the lungs or respiratory tracts and adhere to them. The probability and the manner in which this phenomenon occurs are mainly dictated by the size and aerodynamic characteristics of the particles.

The phenomenon that opposes the deposition is expulsion. It occurs in different ways and times in different parts of the respiratory system and is effective only for non-soluble particles. For example, in the extrathoracic section the deposition phenomenon occurs principally by inertia but the mucous secretions guarantee a removal in a period of time that lasts a few minutes. The same happens also for the particles that penetrate the tracheo-bronchial section only that in this case the removal times rise in the order of hours.

The behavior of the particles that go all the way to the alveolus is completely different. In these cases the deposition time is expanded reaching the months or even years. In cases where the particles should be soluble they will be absorbed by the tissues making it impossible to remove them. The particulate causes the onset of different symptoms depending on the part in which it is deposited.

The particles that penetrate the upper part of the respiratory system, not crossing the larynx, cause an effect of dryness and inflammation of the oral and nasal cavities. These effects, although annoying, do not always lead to serious consequences. As for the finer particles, the situation is completely different. In fact, if they manage to penetrate to the alveoli they can enter the bloodstream, increasing the viscosity of

the plasma (that is the liquid part of the blood), favoring the formation of clots and increasing the probabilities of thrombosis. The side effects of the particulate also depend on its chemical composition. Among the substances that cause the most damage to the whole organism, there are certainly the metals mainly constituted by lead, arsenic and mercury and the Polycyclic Aromatic Hydrocarbons (PAH) which is the main waste product of the combustion of diesel engines. Other minor substances are acids such as sulfur dioxide and nitrogen oxides.

Several studies still ongoing are trying to identify which of the different properties of the particulate particle has the greatest impact on human health. Among the most important are the concentration (both numerical and in terms of mass), the dimensions, the presence of metals, the acidity and the different chemical properties of the core and the surface.

According to the above, the factors that most characterize the interaction with the human body are the shape and the surface area, the chemical reactivity and the water solubility. To this day, a certain result has not yet been achieved on what is the most dangerous aspect.

2.3.2 Effects on vegetation and ecosystems

The main sources of interaction between atmospheric particulate and ecosystems are the processes of dry or wet deposition on soil or vegetation. The resulting effects depend on the chemical composition of the particles (presence of nitrates, sulphates, metals or nutrients) and the susceptibility of the ecosystems.

The particles consisting of nitrates and sulphates represent the pollutants that produce the greatest consequences following deposition on the ground. They are able to alter the circulation and absorption of nutrients, change the structure of the ecosystem and condition biodiversity. Soil acidification (mainly linked to the presence of H_2SO_4 and HNO_3) and changes in plant growth are the most important environmental effects of the deposition of sulphates and nitrates in the soil.

The deposition of nitrates on the soil has heavy consequences also on the balance of aquatic systems; uncontrolled flows of nitrates can cause a significant reduction in the amount of oxygen in the water and lead to serious eutrophication. Relevant effects on the feeding cycle of plants can also derive from the deposition of heavy metal particles, such as Copper, Nickel and Zinc.

The deposition of the particulate on vegetation can have effects of a physical and chemical nature. The particles that remain on the leaf surface for a long time represent an obstacle to sunlight, interfering with photosynthesis and inhibiting the development of plants. The particulate can also exert an acidic and oxidizing action on the vegetation, causing damage to the plant tissues. The fine particulate also conditions the solar radiation that passes through the atmosphere, directly through

the scattering and solar absorption phenomena and indirectly acting as condensation nuclei in the formation of cloud systems. It is estimated that the mist decreases the solar radiation on the soil by a percentage equal to 8%; in some rural areas the decrease in the harvest was attributed to the increase in the quantity of particles airborne.

The effects that the particulate exerts on vegetation and on ecosystems are difficult to quantify and they vary significantly in time and space. The phenomena of deposition of atmospheric particulate on the receptors depend on numerous factors, including deposition modalities, wind speed, humidity, surface roughness and particle characteristics (size, shape, chemical composition, etc.). In addition, each ecosystem has specificities such that the assessment of effects on the basis of analysis on another ecosystem is inadequate.



Figure 2.5: Leaf damage after exposure to 1.5 ppm of SO_2 for 7 hours. [3]

2.3.3 Effects on climate and micro climate

The atmospheric particulate absorbs and/or reflects the radiations coming from the Sun depending on the size and chemical composition of the particles that compose it, and on the wavelength of the radiation itself. This phenomenon directly affects the terrestrial energy balance to which the climate is consequently linked.

The effects of this effect will depend on the relative amount of light energy reflected back to space (backscattering) compared to that absorbed. In addition, particulate particles can act indirectly in favor of cooling the planet. In fact they can act as condensation nuclei for clouds, increasing their probability of formation. The presence of clouds has a double effect: if on one side they reflect the sunlight (more efficient reflection than that of the oceans and the land emerged) leading to a cooling of the Earth's surface, on the other hand they can also have a role in the

phenomena of absorption of terrestrial infrared radiation, contributing positively to the warming of the Earth.

Particulate matter also affects the urban microclimate. In these areas, air pollution contributes to the formation of the so-called "heat island", due to the high number of buildings, which reduces the emission of long-wave radiation at night. In addition to this, the particulate present in large cities can reduce the amount of solar radiation reaching the ground by more than 15%.

This effect is all the more evident the more the Sun is low on the horizon as the path traveled by light through the polluted air increases as the sun's height decreases. So, at a given amount of particulate, solar energy will be reduced more intensely in cities placed at high latitudes and in colder periods.

Compared to the surrounding areas, the relative humidity of the cities is generally less than 2-8%; this is due to the fact that cities have higher temperatures and that rainwater has dried up in a short time. Nevertheless, in the cities clouds and fog are often formed thanks to human activities which in urban areas produce large quantities of particles that act as a condensate nucleus, favoring the formation of clouds and mists.

When the hygroscopic nuclei are numerous, the water vapor condenses rapidly on them, sometimes even in situations of under-saturation, causing an increase in precipitation on the city of atmospheric particulate matter.

2.3.4 Effects on visibility

Visibility is defined as the maximum distance, calculated in a certain direction, to which a dark object in daylight or a source of light not focused in the night is identified. One of the useful parameters for assessing visibility is the light extinction coefficient, which is defined as the attenuation of light by unit distance. In turn, the extinction coefficient is defined as the sum of the absorption and reflection coefficients of gases and particles and is directly proportional to the mass concentration of the particles.

An increase in atmospheric particulate concentration therefore corresponds to an increase in the light absorption coefficient. The decrease in visibility can have localized effects, attributable to a reduced number of sources or single sources, or it can affect large geographical areas. This phenomenon can be traced back to the reflection of solar radiation by fine particles with dimensions of the same order of magnitude as the wavelength of the visible radiation. The influential parameters are the particle size distribution, the chemical composition of the aerosol and the relative humidity value. Moisture favors the absorption process and contributes to the increase in particle volume.

Not all particulate matter comes into play when analyzing the problem of visibility. In fact, it restricts itself to the responsible solid particulate, which consists mainly

of sulphates, nitrates and organic compounds.

2.3.5 Effects on materials

It is known that building materials (metals, rocks, cement) are affected by external weather conditions. Oxidized metals tend to form a protective film; this layer is subject to natural corrosion but the presence of anthropogenic pollutants, in particular Sulfur Dioxide (SO_2), accelerates these corrosive processes and weakens the film. The dry deposition of SO_2 particles damages above all limestone, marble and cement, favoring the conversion of calcite (calcium carbonate) into gypsum (calcium sulphate dihydrate).

The extent of damage depends on the concentration of sulfur dioxide, on the permeability and on the moisture content of the material concerned. In humid conditions, in fact, the particulate deposited constitutes a reservoir of condensation nuclei for the water droplets, in which many gases are dissolved, increasing the acidity of the depositions.

The deposition of the particulate, in addition to the corrosive processes, also generates the blackening of the building materials and of all the exposed surfaces, causing serious damage to the artistic, architectural and archaeological heritage.



Figure 2.6: Palazzo Rivaldi in Rome: example of black crusts. [4]

2.4 Regulatory framework

The Italian legislation that regulates air pollution is based on various laws issued over a wide period of time and falls within the broader scope of the general air quality legislation. The first provisions defining limit values for some pollutants in the atmosphere date back to the 80s; the D.P.C.M. of 1983 established maximum limits of acceptability, partly modified with the D.P.R. n.203/1988. The measures set limits to be met over a medium to long period (one month or one year) for SO_2 , NO_2 ,

O₃, CO, Pb, Fl and suspended particles and short-term limits for non-methane hydrocarbons and carbon monoxide.

The comparison between the data collected and these limits allowed the analysis of the air quality status and the verification of the long-term trend of air pollution. Ministerial Decree 25/11/1994 introduced the concept of PM₁₀, defined as "respirable fraction of total particulate matter" and constituted by particles with an aerodynamic diameter of less than 10 μ m. The decree also defined the technologies and measurement methods, monitoring period and quality objectives for the PM₁₀ and provided indications for the preparation of permanent monitoring systems for the concentrations of benzene, IPA and PM₁₀ in the urban areas most at risk.

For the first time the concepts of the state of attention ("a situation which, if persistent, determines the risk of reaching the state of alarm") and that of an alarm state ("situation of atmospheric pollution which, if persistent, a potential condition of exceeding the maximum limits of acceptability and health risk for the population").

At European level, Directive 96/62 / EC of 27 September 1997 on the assessment and management of ambient air quality has re-established the reference framework for air quality assessment.

The provision established the basic principles for the definition of a common strategy for the assessment and management of air quality with the following aims:

- define and establish environmental quality objectives in the European Community in order to avoid, prevent or reduce harmful effects on human health and the environment as a whole;
- assess the quality of ambient air in the Member States on the basis of common methods and criteria;
- have adequate information on ambient air quality and ensure that they are made public, inter alia by means of alarm thresholds;
- maintain the quality of the ambient air, where it is good, and improve it in other cases.

The directive has been implemented in Italy with the Legislative Decree n. 351 of August 4th 1999, which defined the principles for the assessment of air quality on the national territory based on common criteria and methods.

The introduction of the limit value concepts was important ("level set on the basis of scientific knowledge in order to avoid, prevent or reduce harmful effects on human health or the environment as a whole, which must be achieved by a predetermined deadline which must not subsequently be exceeded") and the target value ("level set to avoid, in the long term, any further adverse effects on human health or the environment as a whole; as far as possible, during a given period").

Directive 1999/30/EC of April 22th 1999 established the quality limit values for ambient air for sulfur dioxide, nitrogen oxides, lead and particles. PM10 is defined as the fraction of total particles that penetrate through a selective dimensional input with an interruption efficiency of 50% for an aerodynamic diameter of 10 μm . Similarly, PM2.5 is defined for which, although no specific limit values are defined, the same is adopted monitoring methodology indicated for PM10. In 2000 the Directive 2000/69/CE that sets the limits for benzene and carbon monoxide is approved; the limit values for ozone are instead defined by the subsequent provision (Directive 02/03/EC).

In Italy the Directives 1999/30/CE and 2000/69/CE are implemented with the D.M. n. 60 of 2002, which partly simplifies the national regulatory framework on the subject by repealing certain provisions established by previous regulations. The community directive relating to ozone, on the other hand, is implemented with the Legislative Decree n.183/2004. Legislative Decree n.155/2010 entered into force on October 1th 2010 and implemented the most recent Community directive (directive 2008/50/EC on ambient air quality and cleaner air for Europe).

The decree can be considered a sort of single text on air quality because it rationalizes the previously issued legislation, confirming the system of limits and prescriptions already in force (ARPAS, 2011). The standard reiterates the limit concentrations for SO_2 , NO_2 , benzene, CO, Pb and PM10 and, for the first time in Italian law, establishes the limit values for PM2.5, the target values for ozone and the target values of the concentrations of Arsenic, Cadmium, Nickel, and Benzo(a)pyrene in PM10 particulate.

The introduction of limit concentrations for some elements potentially present in PM10 reveals an increasing attention to the chemical composition of the particulate, against the limit values so far established in terms of mass concentration, implicitly assuming that the particles are equally harmful regardless of the chemical composition of the same.

Chapter 3

PM measurement techniques: state of art

The measurement techniques of atmospheric particulate can be divided into two large groups: CT (Collecting Techniques) and IST (In Situ Techniques).

The techniques based on sample collection (CT) are divided into several phases. Initially, the particles are collected on a filter by the use of pumps or others air current generators, then the collected sample is analyzed with the measurement instrumentation. The critical point of these techniques is that during sample collection, and in any case before analysis, it is possible that the particles may be contaminated or may undergo changes in their physical and chemical properties (including number and dimension). A second problem is related to the long delay with which the requested measure will be available.

Differently from the previous case in the in situ techniques the analysis is carried out by flowing the air inside the measurement apparatus, in order to minimize the alterations of the properties of the particles. Theoretically, this would allow the characterization of the particles in terms of number and size as well as the possibility of carrying out continuous analysis. The disadvantage consists in the fact that, generally, these techniques provide an indirect measure, and therefore the expected value is obtained by applying suitable mathematical models. This often leads to a low correlation between results from measuring instruments based on different principles.

The main technologies belonging to both groups are described below. Starting from the gravimetric method, widely used in environmental characterization systems as well as reference techniques at legislative level, we will analyze indirect systems such as opacimeters, fumimeters, condensation particle counters and differential mobility analyzers.

3.1 The gravimetric techniques

The Ministerial Decree M.D. 60/02 provides a reference method for the measurement of particulate matter. That is a tested method with sufficient guarantees of precision and accuracy. Next to this there are the equivalent methods, ie methods capable of provide the concentration of PM₁₀, comparable with the reference method]. The official reference method for measuring PM₁₀ is the gravimetric technique, whose operating principles are described below.

3.1.1 Working principles

A sampler aspirates constant flow air through a sampling head. In it the particulate is separated inertially thanks to its particular geometry. The desired dimensional fraction is then collected on a filter for the established sampling period. The filter, after having been brought into fixed conditions of temperature and humidity (20 °C, 50% R.U.), is weighed before and after the sampling so as to determine by difference the mass of the collected particulate. The total volume of extracted air is calculated based on the sampler flow value and sampling duration, and is reported at normal reference conditions (0 °C, 101325 Pa) by means of ambient temperature and pressure values measured by the sampler. The mass concentration of PM₁₀ in atmospheric air it is calculated by dividing the total mass of the collected particles in the dimensional range of PM₁₀ for the volume of the sampled air and is expressed in micrograms per normal cubic meter ($\mu\text{g}/\text{m}^3$). The sampler is designed in such a way that:

- aspirate the air sample through the inlet system and through the particulate material collection filter with uniform speed
- maintain and fix the filter in a horizontal position so that the sample of air is sucked downward through the filter
- allow the filter to be inserted and removed comfortably
- protect the filter and the sampler from precipitation and prevent it insects and other debris are sampled
- minimize the losses of air that would lead to an incorrect measurement of the volume of air passing through the filter
- discharge the intake air at a distance from the air inlet system sufficient to minimize the sampling of the emitted air
- minimize the collection of dust from the support surface

The sampler must also have an inlet air sample system which, by operating within a certain range of flow rates, is able to discriminate the particulate material in the dimensional range of the PM_{10} conforming to all the specifications of correct functionality provided. The air inlet system of the sampler should not show significant dependence on wind direction. The latter requirement can generally be obtained by using an air inlet with a circular symmetry with respect to a vertical axis. The sampler has a control device able to maintain the working flow rate within the limits specified in the setting phase of the instrument, even in the presence of limited voltage variations on the power line and for pressure drops on the filter. [15]

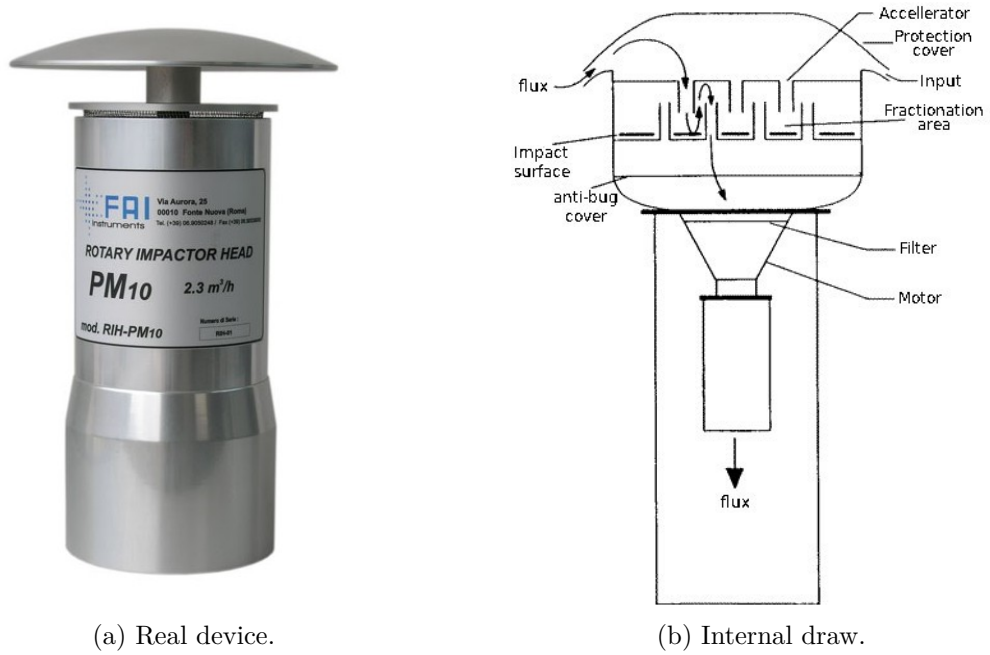


Figure 3.1: Impactor head.[5]

3.1.2 Measurement procedure

The sampling line must be designed in such a way that the air temperature near the filter does not exceed above $5^{\circ}C$ the ambient air temperature and that there are no obstructions or fluid dynamic impediments such as cause quantifiable pressure drops on the PM_{10} particulate sample.

Furthermore, before starting the sampling, it is important to inspect the filter to check that there are no leaks or foreign particles and other imperfections and prepare a data sheet for each filter marked with a number of identification.

Filters must be conditioned immediately before carry out the weighs (pre-sampling and post-sampling) to the following conditions:

- **Conditioning temperature:** $(20 \pm 1)^{\circ}\text{C}$
- **Conditioning time:** 48 h
- **Relative humidity:** $(50 \pm 5) \%$

The new filters must be stored in the conditioning chamber until pre-sampling weighing. The filters must be weighed after the conditioning period. Weighing pre and post-sampling must be performed with the same scale and, possibly, by the same operator, using an effective technique to neutralize the electrostatic charges on the filter.

Filters must be handled with care, using tweezers and gloves to avoid transferring impurities on the filter; then the weighed filter is inserted in the sampler, more precisely in a special container called "filter holder". In the setting phase of the instrument the duration of the sampling, the constant suction flow value of the pump, the value of the minimum suction flow below which the sampler automatically stops the sampling and the normalization temperature must be inserted. Once the sampler has been switched on, it will be necessary to wait until the conditions are reached regime.

Sampling must be of a systematic type, with frequency constant distributed throughout the year. The filter must be removed carefully from the sampler, if possible in the laboratory, using tweezers and gloves to avoid transferring impurities on the filter. Only the outer edges of the filter should be touched. The filter must be stored in a closed filter holder and must be left in a controlled environment for 48 hours with the same conditions of humidity and temperature used for filter conditioning before sampling.

Immediately after conditioning, any electrostatic charges must be eliminated by means of an electrostatic neutralizer and the filter must be weighed taking note of the mass after sampling and the identification number of the filter.

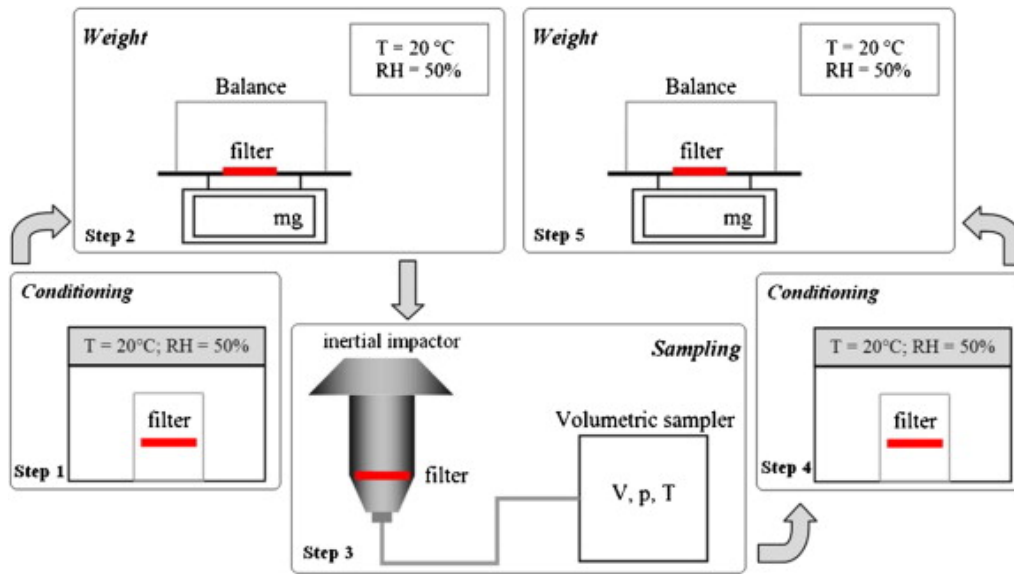


Figure 3.2: Measurement cycle of gravimetric technique. [6]

The following control procedures must be performed on the sampler:

- Checking the tightness of the pneumatic system.
The pneumatic system must not have losses higher than $0.01 \text{ m}^3/\text{h}$ when the sampling filter is replaced with a sealed membrane. This check must be performed at least at the beginning and end of each measurement campaign and anyway whenever maintenance is performed on the instrument.
- Check on the accuracy of the flow measurement.
With this procedure it must be verified that the sampler measure the flow rate with an accuracy of at least 2% of the read value. This check must be performed at least at the beginning and at the end of each measurement campaign and anyway whenever one is executed maintenance. The response of the pressure and temperature sensors must be checked at least at the beginning and at the end of each measurement campaign and however whenever a maintenance is performed on the instrument.

The analytical balance must at least have a resolution of $\pm 1 \mu\text{g}$. The weighing procedures must be performed in an environment where the temperature conditions and the relative humidity correspond to those conditioning the filters.[16]

Quality control on the weighing procedure requires:

- Evaluation of the accuracy during the weighing phases. The weighing of each filter must be repeated at least twice. The standard deviation of the differences between the repeated weighs must not exceed $20 \mu\text{g}$ on at least three weighs.

- Accuracy control. Before each individual weighing group the accuracy of the scale must be checked using samples of referable mass. As a further quality control it is necessary to use at least two laboratory white filters whose weighing must be repeated each time a group of weighs is made. The difference in the weighing values of the laboratory whites provides information on the accuracy of the mass measurement of the material particulate crop.

The measurement of particulate concentration occurs indirectly. That is, measuring other quantities (initial and final mass of the filters, sampled volume, etc. ..) directly determinable. It is indicated in $\mu\text{g}/\text{m}^3$ and is measured under normal conditions of temperature and pressure. The concentration is obtained from the relation:

$$C = \frac{\Delta m}{V_N} = \frac{m_f - m_i}{V_N} \quad (3.1)$$

where V_N is the normalized volume of the sampled air and m_f and m_i are the mass value of the filter after and before the sampling phase respectively.

3.2 OPC

The OPC count and measure the particle size. The operating principle is based on three physical phenomena: light extinction, light scattering and direct image. All types of OPC use a light source, which generates an illuminated region called sampling volume, and a photodetector.

OPCs designed to detect particles through light extinction (see Figure 4.10d) use the continuous light source and a photodetector. The continuous source points directly on the photodetector which returns a constant current signal.

When the particle passes through the sample volume, it obscures the source, preventing the light from hitting the photodetector. The photodetector at the passage of the particle will provide a smaller signal depending on the size of the particle and its ability to absorb light (that is, by the constant optics of the material of which it is composed).

OPCs designed to operate in direct image (Figure 4.10e) use a light source consisting of a very powerful halogen lamp, a lens system and a high resolution CCD. The particles passing through the sampling volume, diffuse the light that is collected by the lens system and conveyed to the CCD which records the shadow of the particle projected onto the detector.

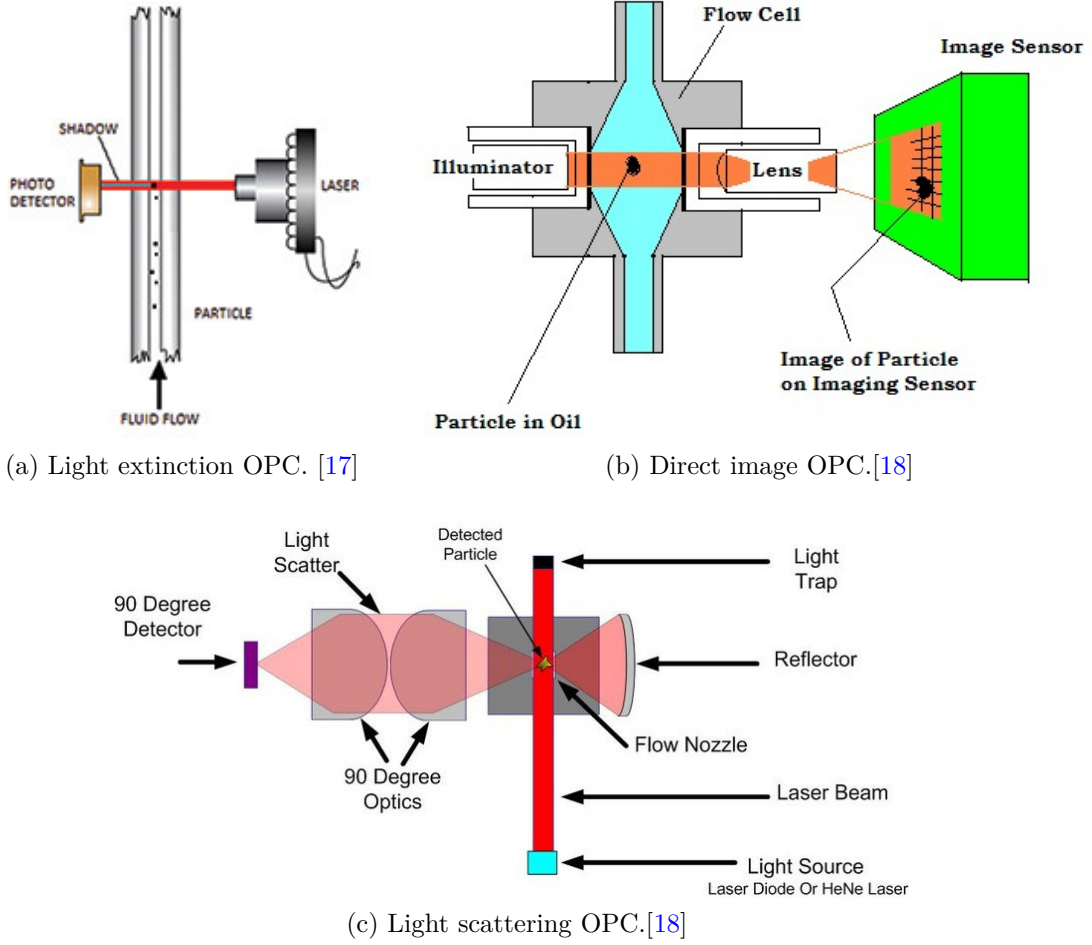


Figure 3.3: OPCs working principles.

Light scattering OPCs (Figure 4.10c) use a light source (typically one monochromatic laser), a lens and a photodetector. The photodetector, unlike the extinguishing system, is positioned at 90° with respect to the direction of the laser beam, measuring the light diffused by the particle with respect to a certain angle. The light is collected by a mirror that focuses it on the photodetector. The signal that returns the photodetector depends on the size of the particle but also on its shape and index of refraction.

The dimensional range of the particles detectable by the OPCs depends on the operating principle on which they are based. For light-extinction OPCs, the smallest detectable particle will have the size of $1\ \mu\text{m}$, while for OPCs based on the direct image it will be possible to detect particles with a dimension $> 0.1\ \mu\text{m}$. OPCs that exploit the scattering principle can detect particles up to $0.05\ \mu\text{m}$

On the market there are various types of OPCs that measure different dimensional ranges, from ultra-fine powders of a few nanometers in diameter, to powders of hundreds of microns in size.

3.3 DMA, CPC and DMPS

The DMA (Differential Mobility Analyzer) is a measurement technique that is based on particle separation according to their electrical mobility.

The device is composed of a grounded cylinder in which an electrically charged bar is inserted. An electric field is created between the walls and the central bar. The air flow to be analyzed is injected from the base of the cylinder. The charged particles move inside the instrument with a speed that depends on their electric mobility. In the upper part of the cylinder there is a slit from which particles having a particular mobility can exit. All the others remain inside the device and will be eliminated with the exhaust flow.

The mobility of the exiting particles depends on the geometry of the cylinder and on the intensity of the electric field. In order for the device to operate properly all flows must be laminar. The range of dimensions that can be analyzed by a single device varies according to the voltage applied to generate the electric field and the precision with which it can be controlled.

The CPC (Condensing Particle Counter) is used to measure very small aerosol particles with optical instruments after particle growth for condensation of a working fluid (butanol or water vapor) on the particles. Measuring the volumetric air flow and counting the particles per unit of time, the instrument provides the concentration of particles in the air.

Modern CPCs provides the following characteristics: measurement of the sampled volume, supersaturation of a working fluid which cools the incoming gas, growth of the particles by condensation of the fluid of work on the particles and optical detectors of the grown particles.

Generally CPCs are distinguished by the method by which they generate supersaturation of the fluid of work. The two most commonly used types are the *expansion-type CPC* and the *conductive cooling-type CPC*. In both cases the sucked gas is first saturated with the vapor of the working fluid and subsequently cooled, which results from the reduction of the vapor pressure to saturation of the working fluid.

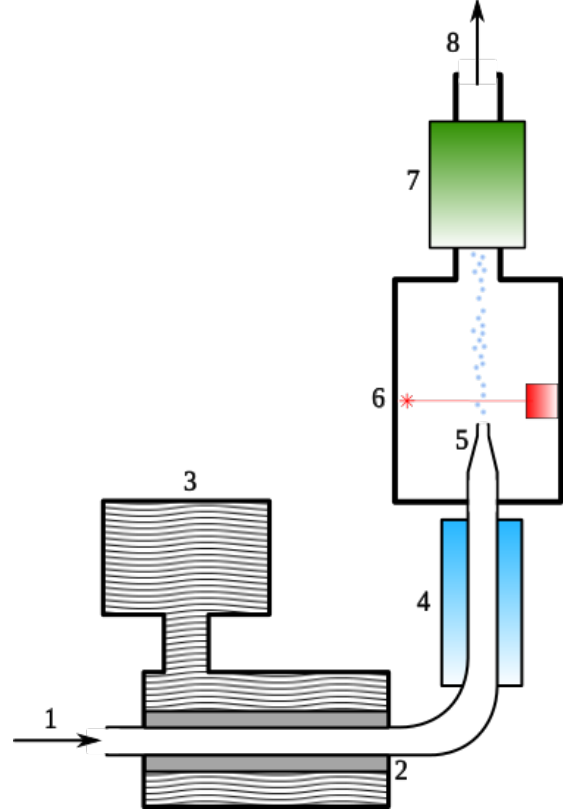
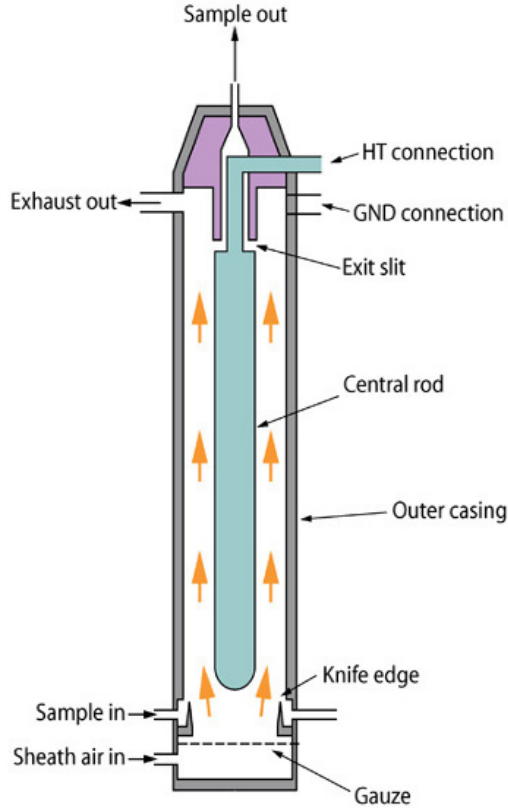


Figure 3.4: DMA schematic diagram. [7] Figure 3.5: CPC schematic diagram. [8]

In the *expansion-type CPC* the working fluid is usually water. The air stream is first humidified to reach the saturation of the water vapor at ambient temperature. Then the aerosol is trapped in the expansion chamber where its volume increases. In some instruments, expansion is caused by a sudden increase in the volume of air using a movable piston; in other models the pressure is reduced inside the expansion chamber by pumping away the air fraction.

The first models were manually operated, while the latter are driven by automatic mechanisms. However, the measurement is not continuous until the CPC operates cyclically. While this type is considered the most direct method to measure and therefore more predictable to become standard, the cyclic method is more disadvantageous, because it is incompatible with the requests for fixed instruments.

The *conductive cooling-type CPC* consists mainly of a saturator, a condenser, and an optical detector. The working fluid is generally an alcohol, for example butanol, methanol or ethanol. Air flows continuously inside the instrument, guided towards the exit by a depressor. It first passes into a heated alcohol tank, where it is saturated with alcohol vapor. Depending on the saturator temperature, residence

time increases until the air saturates with vapor. Then the aerosol enters in the condenser, consisting of a tube maintained at a low temperature by cooling the walls. In the condenser tube the saturated gas vapor cools by conduction and convection, reaching the supersaturation of the alcohol. According to the supersaturation level, the particles of a certain size grow by condensation of the alcohol vapor to a size of a few micrometers.

These "drops" are then counted by the optical detector by means of light impacts. In the condenser, the super-saturation of the alcohol vapor is not homogeneous above the cross-section of the tube. This is because the surface of the pipe effects a dispersion of both heat and vapor: the temperature decreases from the center towards the walls and at the same time the concentration of the vapor decreases towards the walls due to the concentration of the material on them.

As a result of this non-homogeneous distribution of supersaturation, the diameter for activation of the aerosol particles is a function of the position of the particles inside the condenser.[19]

Fig. 3.5 shows the main parts of the instrument:

- | | |
|---|--------------------|
| 1. Air inlet | 5. Focusing nozzle |
| 2. Porous material block, which is heated to saturation temperature | 6. Laser counter |
| 3. Working fluid | 7. Air pump |
| 4. Condenser | 8. Air outlet |

A DMPS consists of a DMA and a CPC. Particles are first size selected with the DMA and then counted with the CPC.

Part II

Case study

Chapter 4

PM monitor:

The device made at the *Politecnico di Torino* laboratories is a hybrid between an optical particle counter and a more classic gravimetric meter. Specifically it adopts an optical system to count and recognize the particles while uses the techniques and the filters of the gravimetric systems to capture them.

4.1 Device overview

The device was made in two different versions: the first portable, manually operated measuring $105 \times 59 \times 98$ cm. The second one is bigger $240 \times 90 \times 190$ cm, it is completely automatic and it complies with the IP55 standard. Both devices use the same components except for the solar panel and the two motors absent in the portable version. For this reason all the considerations made below are to be considered as identical for both objects.

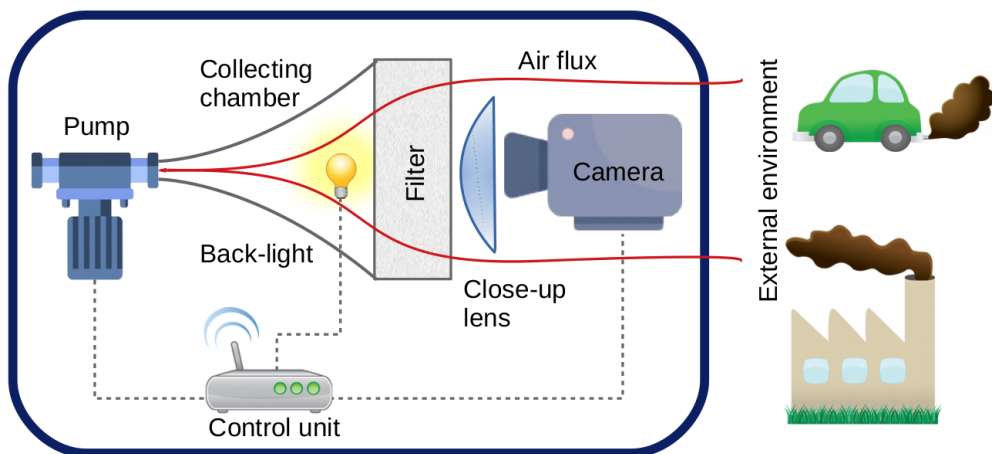


Figure 4.1: Device block diagram.

Referring to Figure 4.1, an air flux is forced from the external environment on a sampling fiberglass filter by a membrane pump. An optical system, composed by a camera and a Macro close-up lens, takes photos of the filter surface at different back-light wavelength. Finally, a dedicated software running on a *RaspberryPi Zero Wireless* analyzes the photos and detects the particles. The collected data can be transmitted on a remote server via *WiFi 802.11* connection, or stored on the internal memory. The detailed working cycle description is at the section 4.1.2.

4.1.1 Part list

According to the figures 4.2c and 4.3c, the device is composed of the following parts:

1. **Camera:** The core of the optical compendium is the camera. It has the task of capturing the images of the filter that will subsequently be analyzed by the software. In this regard, a Sony Pixel IMX219, 8M Pixel sensor was chosen. This is mounted in a module measuring $25\text{ mm} \times 24\text{ mm} \times 9\text{ mm}$. It includes a 15 pin connector which allows interfacing via a CSI (Camera Serial Interface) interface and a lens with a fixed focal length of 3.04 mm. A peculiarity is that this lens does not have an IR filter, this makes it possible to use this light source for filter analysis.
2. **Close-up lens:** The close-up lens is located immediately next to the camera. It is a Macro lens with a magnification of $15\times$ that allows focusing at very close distances. It is essential to increase the measurement resolution of the device and then identify the smaller particles.
3. **Filter:** The filter is the medium through which it is possible to capture the particulate particles suspended in the air flow. For this purpose a Grade GF 10 Glass Filter with organic binder was used. Produced by GE Healthcare's Life Sciences is a borosilicate glass filter with an average porosity of $3\mu\text{m}$. In the two versions of the device the same filter is installed. The only difference is in the form factor: in fact while in the portable version is a 47 mm diameter disk, in the automatic version is a tape with a height of 30 mm and a length of 20 m. Both filters are about $350\mu\text{m}$ thick.
4. **Led panel:** The led panel provides the light source used to illuminate the filters during their analysis. It consists of a custom PCB on which three LEDs are mounted. The first is an infrared Honeywell SEP8705-003 THT (Through-Hole Technology) that emits at 880 nm, the second is the ultraviolet Nichia NSSU100DT SMD (Surface Mounting Device) that emits at 375 nm while the third is a Red Green Blue multi chip led LRTB R98G SMD by Osram that emits at three different wavelengths: 470 nm, 528 nm and 625 nm.

5. **Air pump:** A constant flow of air is essential to contaminate the filter and therefore guarantee a good final measurement. For this task the D200 micropump by RS Pro was used. This is a membrane pump, 5V DC powered with a nominal output air flow of 150 mL/min.
6. **Collecting chamber:** In both devices the filter is located between two collecting chambers. The first communicates with the outside environment while the second is connected to the suction outlet of the pump. The goal of the chambers is make the air flow uniform and to force it onto a 1.5×1.5 cm square portion of the filter. They are made of PLA plastic by 3D printing.
7. **Battery:** The power source of the device consists of a 8800mAh LiPo battery. This can be recharged either by an external power supply or by solar panel (only in the automatic version).
8. **Box:** The device case is different in the two versions. In the portable one it was designed and built ad-hoc to minimize the size of the single components. It is made of PLA plastic and produced by 3D printing. The automatic version, on the other hand, has different characteristics and needs so an external junction box available on the market has been chosen. This choice was preferred for several reasons: the first is purely economic as it would have cost too much both in terms of materials and time to print the entire box, the second reason is that the available 3D printer does not allow prints of such a large size, the third and last is that was possible to insert everything in a container complying with the IP55 standard, thus ensuring correct operation even in unprotected open areas.
9. **Control unit:** This unit consists of two parts. The first is a RaspberryPi Zero Wireless. It runs the software that manages the device and analyzes the captured images. Equipped with Bluetooth and WiFi connectivity for data transmission also has an internal memory in which to store data in case of lack of connection. The second part consists of a PCB developed specifically for this device. Mounted directly above the RaspberryPi thanks to the 40 pin connector is equipped with all the circuitry for power management (solar panel and battery charge controller, step-up, step-down), LED control and motors and pump drive.
10. **Power switch:** The device is equipped with a SPST (Single-Pole, Single-Throw) switch to allow its switching on and off.
11. **Collecting chamber motor:** In the automatic device the filter consists of a tape that is placed between the two collection chambers. In order to guarantee an hermetic closure, necessary for the pump to work properly, an SG90 servo

motor from TowerPro was used. It is connected directly to one of the two collecting chambers and makes it rotate about 50°, opening, with respect to its vertical position when the filter must be moved. In the same way it closes for the time in which the filter must be exposed. The torque of the motor is $2.5 \text{ Kg} \cdot \text{cm}$

12. **Filter motor:** The filter motor performs two important operations: the first is to expose a clean section of the filtering tape when a new measurement is to be started, the second is to move the filter part exposed by the collection chambers to the optical measurement unit. All in an automatic way without intervention of manual operators. The motor used is a McLennan 42M048C-1U unipolar stepper. Among the most interesting plate data there is the torque of $6.6 \text{ N} \cdot \text{cm}$ and the step-angle of 7.5° . This type of motor has been preferred for this last feature, that is the possibility of performing small controlled rotations and obtain a very precise movement of the filter.

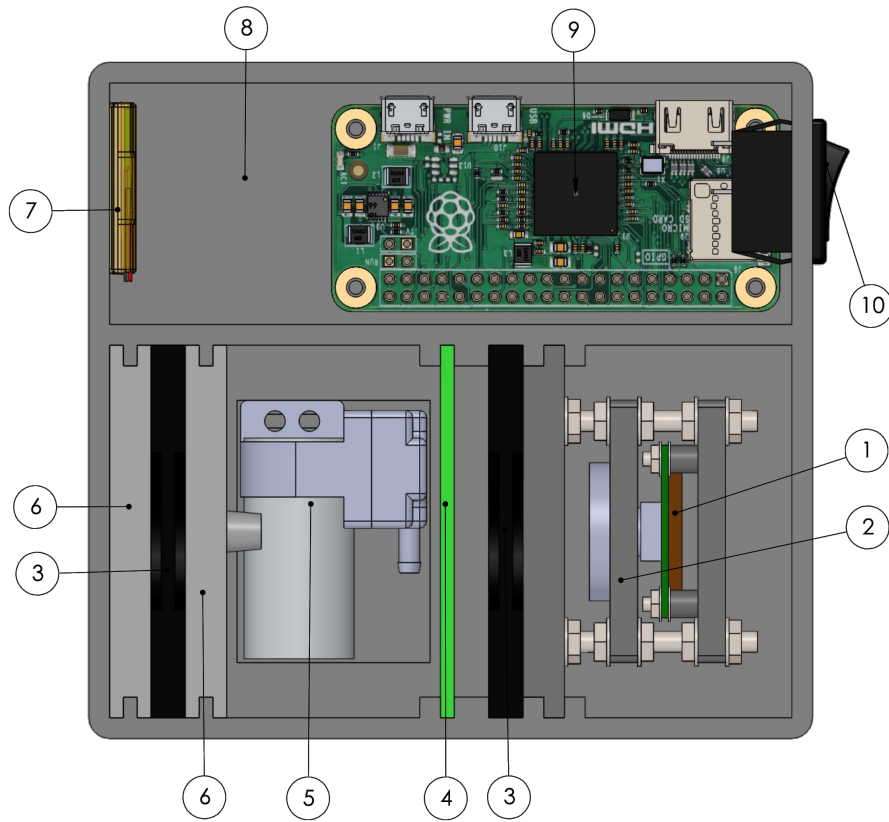
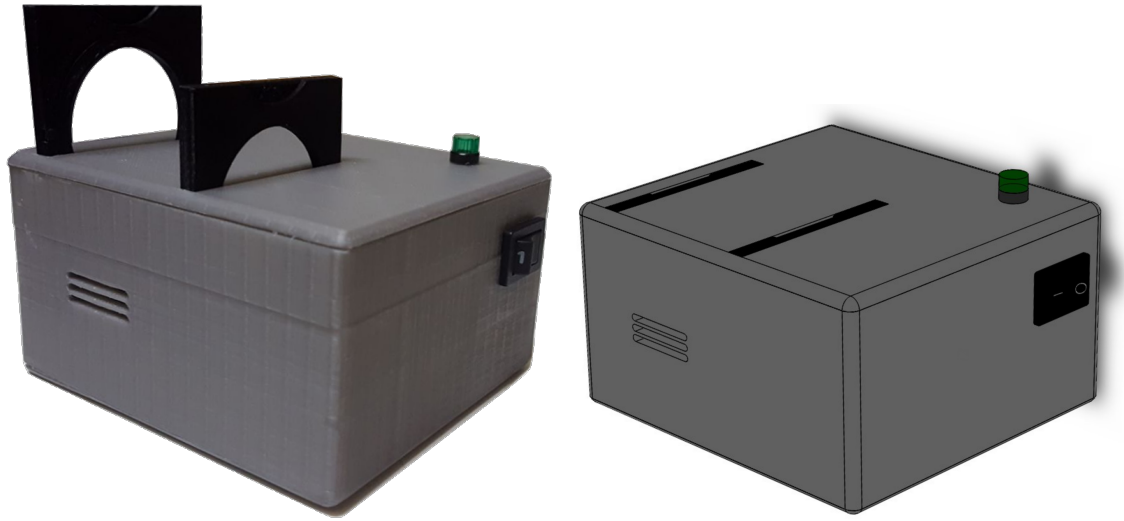
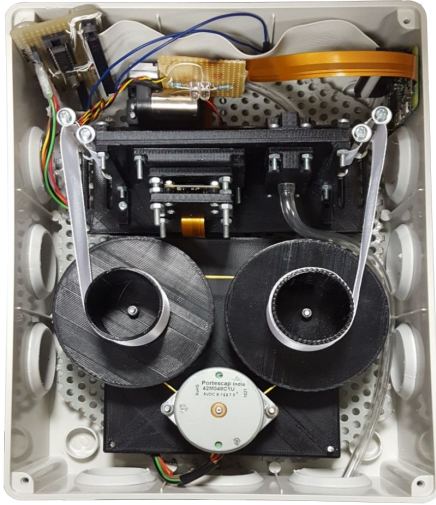


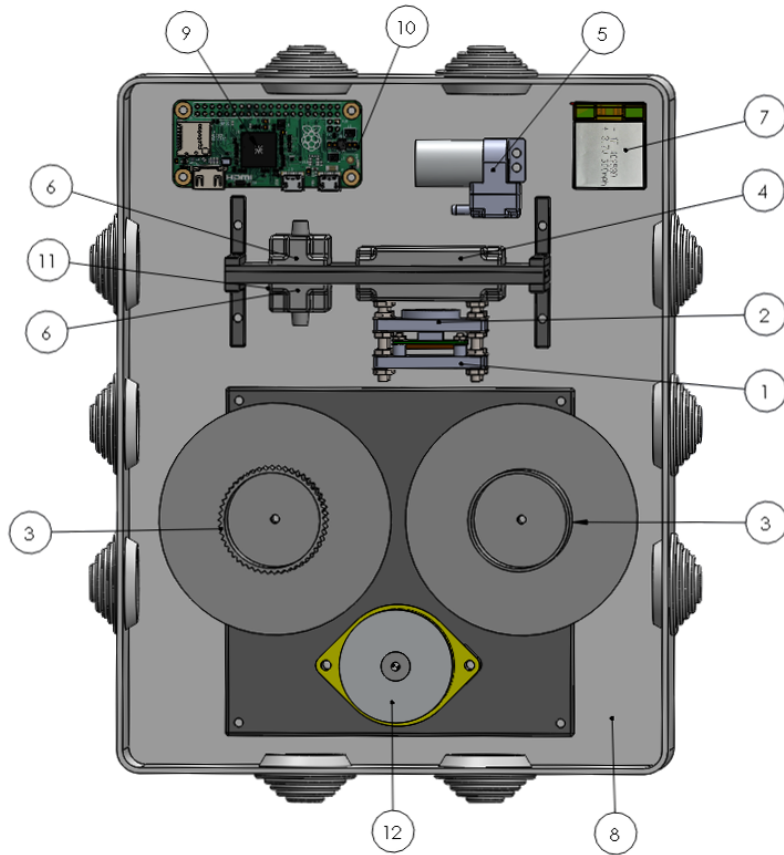
Figure 4.2: Portable device



(a) Prototype photo.



(b) Isometric view.



(c) Upper view: ballon.

Figure 4.3: Automatic device

4.1.2 Working cycle

The working cycle of the instrument consists of the following phases:

1. **Filter replacement:** The first step to start a new measurement is the insertion of a clean filter into the machine. This phase is performed differently based on the two devices. In the portable device it must be done manually. It is necessary to extract the filter holder (3), remove the circular locking ring, insert the new disc filter in the holder and lock it again. In the automatic device the filter replacement is performed by rotating the tape mounted on the motor (12).
2. **Filter exposure:** Once the filter is in position it is possible to proceed with its exposure. To guarantee the correct movement of the tape filter, in the automatic version the collecting chamber connected to the pump is movable. Before this phase it is therefore necessary to reposition it in contact with the filter. This operation is automatically activated by the collecting chamber motor. The air flow is generated by a DC membrane pump. Since this phase is very long (about 20 hours), the control unit goes into standby state to reduce power consumption.
3. **Filter photo:** To analyze the exposed filter is important to move it to the right position. In the manual device it is necessary to extract the filter holder from the collecting chambers and introduce it between the led panel and the optical setup. In the automatic version it is necessary to make a rotation of the filter motor. Once in position, in both cases, the filter is located between a LEDs light source and a measuring optical unit consisting of a macro lens and a camera. The control unit captures six images of the filter, one for each available wavelength plus one in which all the LEDs are turned on and saves them in the internal memory.
4. **Image analysis:** The main phase of the whole process is the optical analysis of the filter. It occurs by capturing six photos illuminating the filter with red, green, blue, infrared, ultraviolet light and with the sum of the previous ones. The captured images are subsequently analyzed by a specifically developed software that provides output data relating to the concentration and dimensional classification of the particulate.
5. **Data sending:** The last step consists in sending the data just obtained to a server or a local device. The control unit is equipped with WiFi and Bluetooth connectivity. With no wireless connection, data can be saved in the internal memory.

4.2 Control unit

The control unit is the computational core of the device. It has the task of running the particle recognition software and coordinating all parts of the measurement system. There are two parts: a RaspberryPi Zero W and a custom interface board.

4.2.1 RaspberryPi Zero W

The Raspberry Pi is a single-board computer. Born in United Kingdom with the aim of developing a low-cost product to promote and disseminate programming sciences in schools, today it is one of the most used products in the research and development sectors, as well as among the makers. The main features that make it so widespread are its economy, high versatility and compactness. In fact it is possible to run a Linux-based operating system on a $6.5\text{ cm} \times 3\text{ cm}$ board for only 5\$.

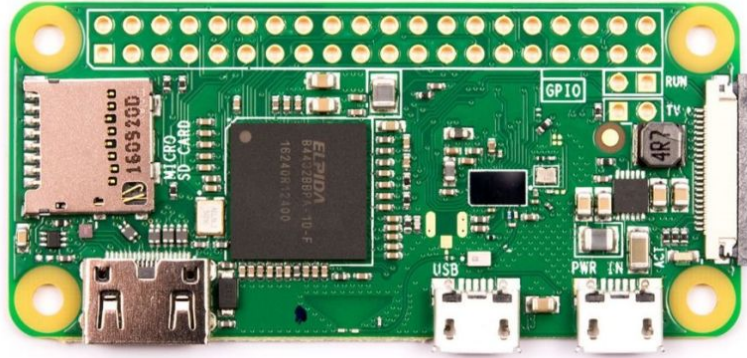


Figure 4.4: RaspberryPi Zero W.[9]

The hardware is based on a 1 GHz Broadcom BCM2835 processor and is equipped with 512 MB of LPDDR2 RAM. There is also a micro SD slot to manually install your internal storage. The video output is a mini-HDMI port with a maximum output of 1080p / 60fps, while power is supplied via a micro USB connector. Particularly important is the presence of the CSI (Camera Serial Interface) connector through which it is possible to directly connect a serial camera module and a 40-pin connector useful for connecting controllers and peripherals. It is also equipped with a wireless communication module according to the *WiFi 802.11* and *Bluetooth 4.0* standards.

Regarding the software, the RaspberryPi Zero W is able to run a complete Linux-based operating system. In this case, Raspbian, a version of the most famous Debian

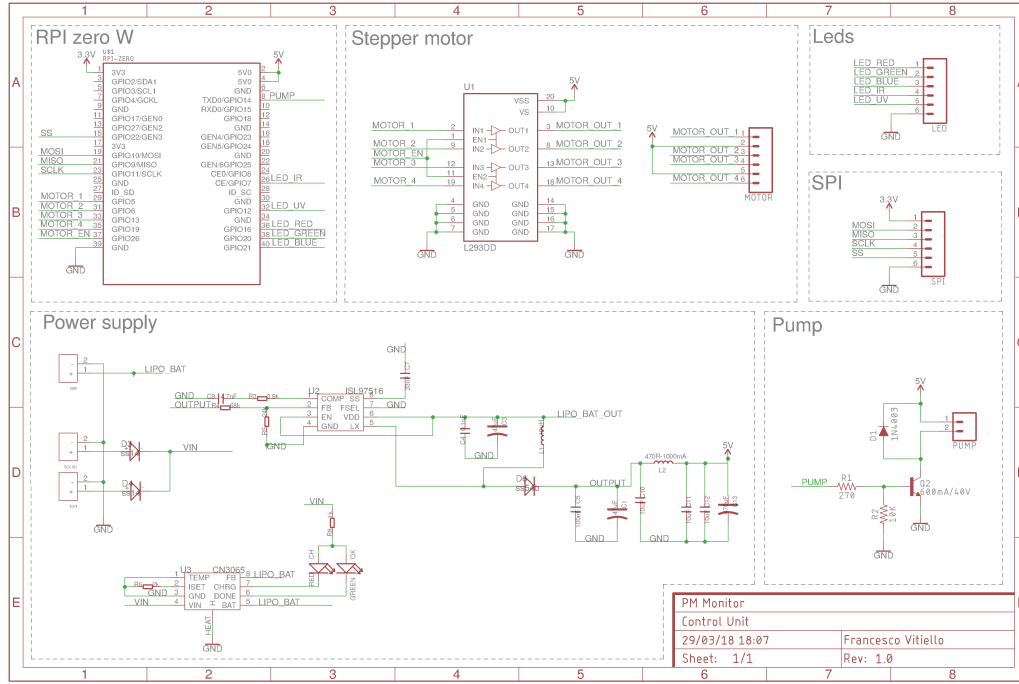
optimized for the hardware of this board, has been chosen. On it is executed the software, written entirely in Python language, that control the device and recognize the particles.

4.2.2 Interface board

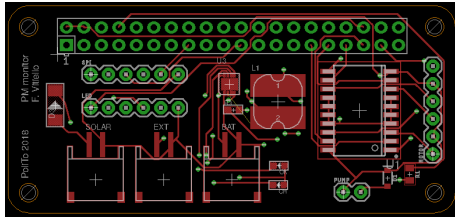
The interface board is a module developed specifically for this device and is connected directly to the RaspberryPi Zero W via a 40-pin connector. It is a two-layer PCB of the same size as the board on which it is mounted (6.5 cm \times 3 cm) and all the components and connectors soldered on it are necessary to perform two important functions:

- **Device operation:** The device receives from the Raspberry Pi all the control signals useful to manage the functionality of all the components of the system and mounts the drivers for their piloting. The devices that need external components are mainly the stepper motor and the pump. They are driven respectively by an LM293D and a circuit based on a bjt 2n222A transistor. In this section we find also the resistors connected in series to the LEDs and an SPI connector that allows the interfacing of external peripherals for future developments.
- **Power supply:** The interface board has the task of supplying power to the entire system. It is equipped with three connectors to which a battery, a solar panel and an auxiliary power source can be connected. The power source is chosen automatically according to the available ones. The main components are the Li-on CN3065 battery charge regulator and the step-up circuit necessary to bring the output level of the battery out to 5V when it is selected as the system power source.

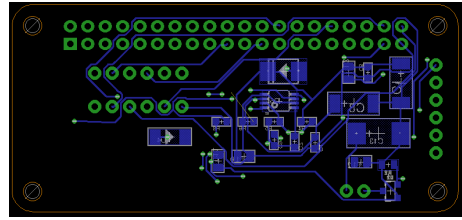
The figure 4.5 shows the schematic (a) of the realized circuit and the top (b) and bottom (c) views of the PCB.



(a) Schematic.



(b) Top view.



(c) Bottom view.

Figure 4.5: Control unit

4.3 Image processing software

The image processing software is a fundamental element of the atmospheric particulate matter's measurement. It has been developed to analyze the images of the filter surface captured by the camera and to provide, as output, data about the concentration and the dimensional analysis of the particles found.

The software is written in Python, a high level programming language, and uses the openCV (Open Source Computer Vision Library) software library. This library is based on an open-source project and are released under a BSD license. Natively it was written only in C++ language, today are available a large number of interfaces

for the most important programming languages such as Python, Java and Matlab. Another advantage is that it is multiplatform, being compatible with Windows, Linux, Mac OS and Android operating systems.



Figure 4.6: OpenCV logo.[\[10\]](#)

The optical analysis algorithm developed for the measurement and characterization of the particulate matter is divided into two phases: preprocessing and detection.

4.3.1 Preprocessing

The first phase of the optical recognition algorithm is the so-called preprocessing phase. At this stage the image is prepared so that it can be processed in the best way by the detecting algorithm. Specifically, a gray-scale conversion, a selective Gaussian blurring and 2D Kernel filter are applied in sequence in order to increase the sharpness of the image and to mark the contrast between the particles and the fibers of the filter where they are inserted. The different sub-phases are explained below.

Gray-scale conversion

The images provided by the camera are based on the RGB color model. It is an additive type model where each color is defined by the sum with different intensity of the three primary colors: Red, Green and Blue. It is therefore possible to split the original image into three different images based on the color component. This is done using the *split* function provided by the library.

Listing 4.1: Gray-scale conversion software implementation.

```
1 input_image = cv2.imread(image)
  b, g, r = cv2.split(input_image)
```

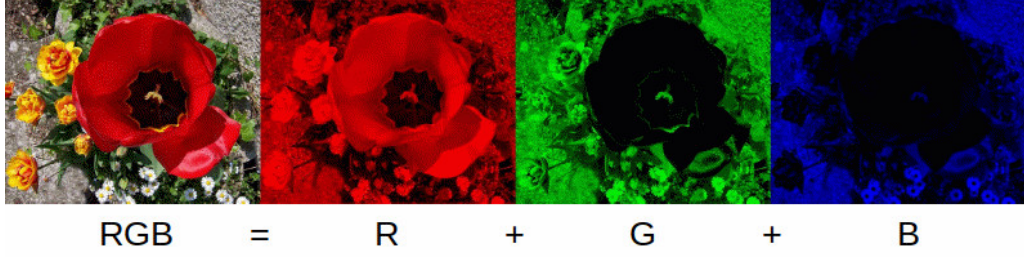


Figure 4.7: Additive RGB trichromy of a real image.[11]

On this principle are based many electronic devices such as monitors. Although the operation is opposite, also the camera sensor used is made up of three arrays of detectors, each one sensitive to a different wavelength. The purpose of this conversion is therefore to select a single channel, that is the image produced by a single sensor array and not the sum of the three combinations.

Since the device has light sources of different wavelengths, the conversion procedure will depend on the active source at that moment. Specifically, the level closest to the wavelength of the light source is selected. While in the case of combined sources, the output image is given by the following formula:

$$G = \sqrt{R^2 + G^2 + B^2} \quad (4.1)$$

The table shows the link between light source and the correspondent channel taken into account.

Table 4.1: Light source and correspondent output channel.

Light source	Selected channel
Red	Red
Green	Green
Blue	Blue
Infrared	Red
Ultraviolet	Blue
Combination	$\sqrt{R^2 + G^2 + B^2}$

Selective Gaussian blurring

Once the image has been converted to gray-scale mode, a selective Gaussian blurring filter is applied. The Gaussian smoothing operator is a 2-D convolution operator that is used to ‘blur’ images and remove detail and noise. It uses a kernel matrix that represents the shape of a Gaussian hump.

Considering a 2D space, the Gaussian function has the form:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4.2)$$

where σ is the standard deviation and x and y are the point coordinates. The distribution is shown in fig. 4.8.

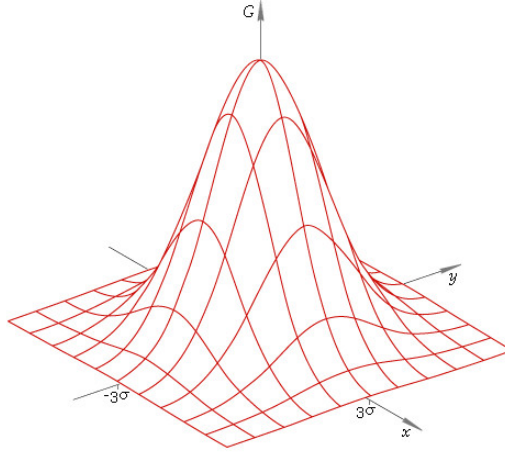


Figure 4.8: 2D Gaussian distribution.[12]

The low-order principle of the Gaussian blur filter consists in carrying out a convolution between the image and the function itself. For computational purposes, the image is represented by a matrix of numerical values, so in order to execute the desired convolution it is necessary to generate a matrix that represents the Gauss curve in the two-dimensional plane. This technique is called "point-spread".

Mathematically , the Gaussian function assumes non-zero values in all points of space. It is therefore impossible to define this function with a matrix as it would require infinite dimensions. What is accomplished is to approximate this function assuming that it becomes null when we move away from the average value of a quantity equal to about three times the standard deviation. In these approximation conditions the matrix has finite dimensions and is therefore feasible.

This is a suitable integer-valued convolution kernel with $\sigma = 1.0$.

$$\frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}. \quad (4.3)$$

After defining the kernel array, standard convolutions techniques can be applied.

By exploiting the mathematical properties of the isotropic Gaussian equation, it is possible to separate the components along the x and y axes, considerably reducing the calculation time. The convolution on the two-dimensional plane can then be calculated by carrying out two monodimensional convolutions in sequence. The first along the direction of the x axis, the second along the direction of the y axis.

Some DSPs (Digital Signal Processors) use pipeline hardware architectures to further speed up this calculation. In this case the convolution between the image and a smaller Gaussian kernel is executed in an iterative way. This technique is used mainly when the function has a high standard deviation.

To apply this filter, can be used the *GaussianBlur* function provided by the library.

Image 4.9 shows the effects of the filter with two different standard deviations values.[12]

Listing 4.2: Gaussian blurring software implementation.

```
1 blur = 3
  imf = cv2.GaussianBlur(im0, (blur, blur), 0)
```

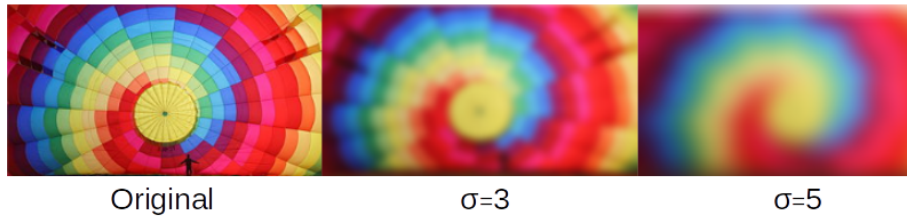


Figure 4.9: Gaussian blurring with different σ values.[13]

2D Kernel filter

The last step of the preprocessing phase is to transform the image using a 2D Kernel filter.

This uses the same techniques described above for Gaussian blurring. A correlation is made between the image and a matrix called Kernel.

The difference with respect to the previous point is that it is possible to modify the matrix both in dimensions and in the values contained therein, obtaining a filter capable of optimally adapting to the image being analyzed.

The formula used to make the transformation is as follows:

$$dst(x, y) = \sum_{\substack{0 < x' < \text{kernel.cols} \\ 0 < y' < \text{kernel.rows}}} \text{Ker}(x', y') * \text{src}(x + x' - \text{anchor.x}, y + y' - \text{anchor.y}) \quad (4.4)$$

where *dst* is the output image from the process, *Ker* is the kernel matrix, *src* is the input image and *anchor.x* and *anchor.y* are the coordinates of the anchor point that indicates the relative position of a filtered point within the kernel.

For the case in question a 5×5 integer matrix was constructed starting from three parameters: $K_0 = -1$, $K_1 = 2$ and $K_2 = 4$. The kernel obtained is:

$$\text{Ker} = \begin{bmatrix} K_0 & K_0 & K_0 & K_0 & K_0 \\ K_0 & K_1 & K_1 & K_1 & K_0 \\ K_0 & K_1 & K_2 & K_1 & K_0 \\ K_0 & K_1 & K_1 & K_1 & K_0 \\ K_0 & K_0 & K_0 & K_0 & K_0 \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & 2 & 2 & 2 & -1 \\ -1 & 2 & 4 & 2 & -1 \\ -1 & 2 & 2 & 2 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix} \quad (4.5)$$

The anchor point should lie within the kernel, in this case the value $(-1, -1)$ was chosen. It means that the anchor is at the kernel center.^[10]

Listing 4.3: 2D Kernel filter software implementation.

```

1   ker0 = -1.0
   ker1 = 2.0
3   ker2 = 4.0
   ker = np.array([[ker0, ker0, ker0, ker0, ker0], [ker0, ker1, ker1,
   ker1, ker0], [
5       ker0, ker1, ker2, ker1, ker0], [ker0, ker1, ker1,
       ker1, ker0], [ker0, ker0, ker0, ker0, ker0]]) /
       ker2
   imt = cv2.filter2D(imf, -1, ker)

```

This filter is used to increase the sharpness of the image.

The correct choice of the Kernel matrix values is a fundamental element to obtain good results in the detecting phase.

4.3.2 Detection

The detection algorithm is the core of the entire measurement process. It deals with identifying the particles deposited on the surface of the filter, counting them and estimating the area.

To do this a function provided by the *OpenCV* library has been used: *SimpleBlob-Detector*.

This function implements an algorithm composed by four steps:

1. Convert the input image to binary images. To do this, the software applies repeated threshold levels starting from a minimum value *minThreshold* up to an excluded maximum *maxThreshold*.
2. Identify the particles present in each image and obtain the coordinates of the respective centers.
3. It groups the particles present in the different images that have the same coordinates of the centers. In this way elements that have very close centers are identified as a single particle.
4. Once the particles are grouped, the center of the group and its dimensions are defined.

The second step can be performed in different modes. This can be set by *filterBy** function. Available filtrations are:

- **By color:** This filter compares the particles in the image with a reference color (*blobColor*) that can be set by the user. If they are different, the particle is discarded. A value of *blobColor* = 0 indicates the dark particles, the value *blobColor* = 255 indicates the light ones.
- **By area:** This filter considers only valid particles whose area is between two user-settable values: *minArea* and *maxArea*.
- **By circularity:** The particles which have a circularity factor, expressed by $\frac{4 \cdot \pi \cdot \text{Area}}{\text{perimeter} \cdot \text{perimeter}}$, between *minCircularity* and *maxCircularity* are extracted.
- **By inertia:** The particles which have the inertia ratio between *minInertiaRatio* and *maxInertiaRatio*.
- **By convexity:** This filter extracts all the particles which have the convexity factor, expressed as the ratio between area and area of blob convex hull, between *minConvexity* and *maxConvexity*.

To obtain a dimensional analysis of the particulate matter, it was decided to use this algorithm by performing a search by area.

The implementation of the algorithm is shown in the listing 4.4.

It is possible to notice that all the filtering methods have been set to false, except for the area that is set to true. It is also important to note two very important parameters that are *maxThreshold* and *minArea*.^[10]

These indicate respectively the upper value of the threshold used to convert the input image into binary images and the minimum area, expressed in square pixels, to consider valid the found particle.

Changing these two parameters changes the behavior of the algorithm obtaining results that are very different from one another.

It is therefore essential to try to choose an optimal value of these parameters in order to obtain the most accurate and reliable estimate of the number and size of the particles present on the filter.

Listing 4.4: Detection algorithm software implementation.

```
1   maxT = 106
   area = 16
3   params = cv2.SimpleBlobDetector_Params()
   params.minThreshold = 0
5   params.maxThreshold = maxT
   params.filterByArea = True
7   params.minArea = area
   params.filterByCircularity = False
9   params.minCircularity = 0.15
   params.filterByConvexity = False
11  params.minConvexity = 0.88
   params.filterByInertia = False
13  params.minInertiaRatio = 0.01
   det = cv2.SimpleBlobDetector(params)
15  keypoints = det.detect(imf)
```

The algorithm outputs data about the particles found in the form of *keypoints* array. The *keypoints* object in turn consists of several elements that are:

- **Pt** : x and y coordinates of the particle.
- **Size**: Diameter of the particle.
- **Angle**: Orientation of the particle.
- **Response**: Detector response on the particle.

- **Octave:** Pyramid octave in which the particle has been detected.
- **Class_id:** Particle id.

What is most interesting is the data related to the size. It is to be considered as the diameter relative to a perfectly circular particle having an equivalent area equal to that of the analyzed particle.

From this it is possible to calculate the real dimensions of the particle necessary for dimensional analysis.

Another very important data is that related to the position: knowing the coordinates of the particle found is useful during the calibration phase to identify which particles have been correctly recognized and which are not allowing to modify the configuration parameters seen before.

This data can also be used to perform a differential analysis between images of the same filter captured with light sources at different wavelengths, thus creating a sort of differential spectroscopy.

Figure 4.10 shows an example of a filter analyzed by the particle recognition software.

The sample is a filter exposed on March 15, 2018 from 00:00 to 24:00 in the metropolitan city of Turin. The sensor has been positioned at a height of about one meter near a fairly busy road. The photograph was obtained by illuminating the filter with a green light at 528 nm.

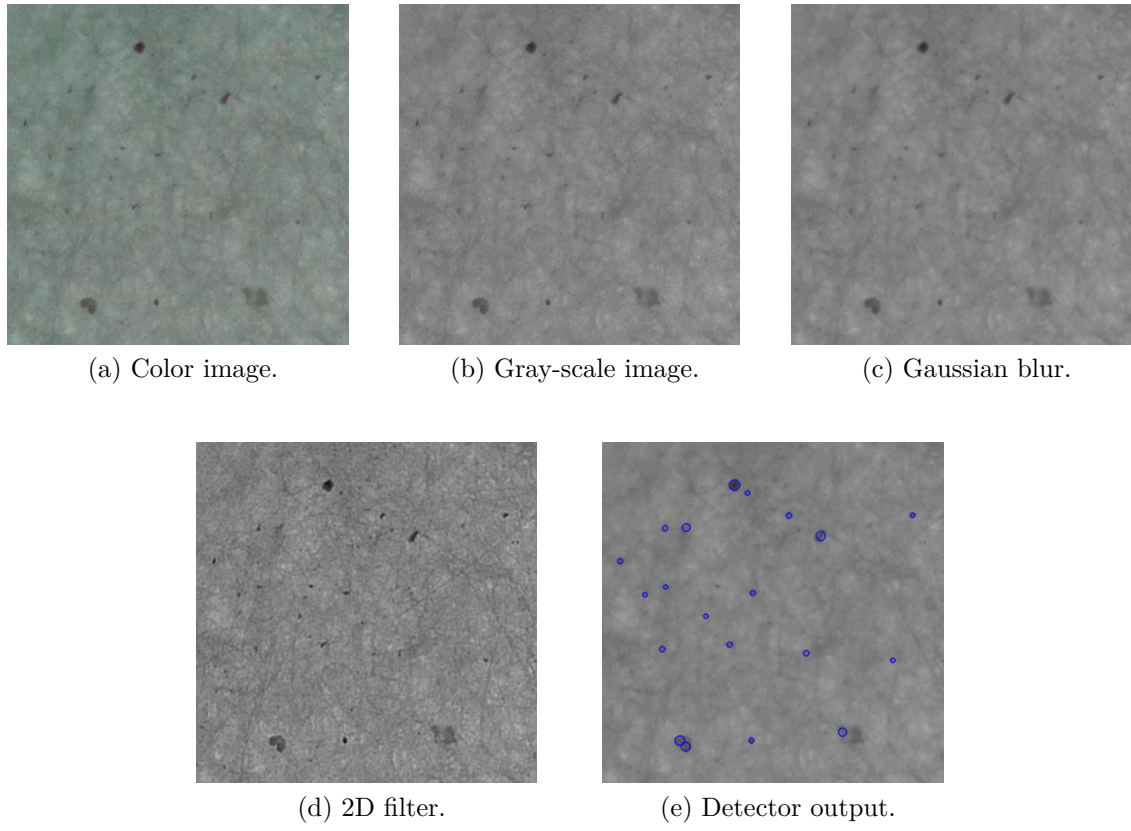


Figure 4.10: Detection steps.

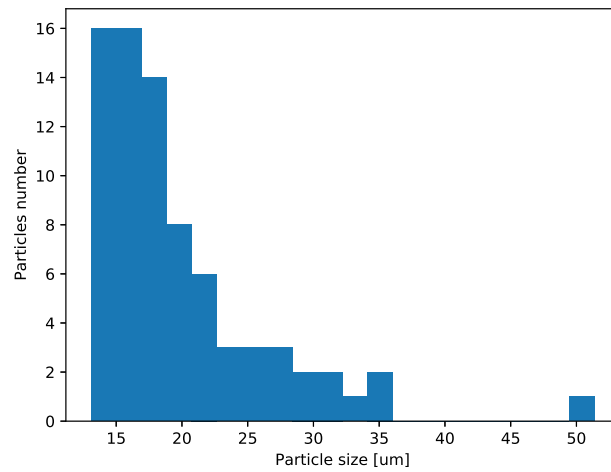


Figure 4.11: Output histogram.

4.4 Grid architecture

To obtain a more reliable and detailed measurement of atmospheric particulate concentration, it is essential to increase the number of measurement nodes and to create a communication infrastructure that allows the transmission of the obtained data.

Nowadays there are many ways of connecting measurement nodes to obtain a sensor network. They differ in terms of cost, technology, protocol and means of communication, maximum reachable distance, bitrate, power consumption, etc.

Analyzing the device in question it is equipped with a RaspberryPI Zero W which provides wireless communication according to the *WiFi 802.11* and *BLE 4.0* (Bluetooth Low Energy) standards.

Comparing the characteristics of these two technologies, it was decided to develop a wireless sensor network based on *WiFi* communication. This solution has been preferred because it allows to cover greater distances than *Bluetooth*, it is easy to implement and above all it is widely used and allows a direct connection to the Internet.

The elements of this network are mainly four:

1. **Server node:** The server node is essentially the measurement node. The wireless sensor network is equipped with a server interface written in Python capable of receiving commands and sending data via the HTTP port 8000. In addition to the HTTP protocol, it is also accessible via the SSH and SFTP protocols on ports 21 and 22, respectively. These two protocols have been implemented to allow full remote management and maintenance of the device.
2. **Internet:** The Internet is a public access network capable of interconnecting devices all over the world. This allows the different elements (server node, client and database) to dialogue even if they are geographically located in different places.
3. **Remote DB:** The data obtained from the sensor node are sent via a POST HTTP request to a PHP script that takes care of storing them in a MySQL database. This step is fundamental both for saving data and also because it is possible to create different types of interfaces (websites, mobile applications, etc.) useful for the consultation of data on a large scale.
4. **Remote client:** The remote client is a software written in Python capable of interfacing with all sensor nodes on the network. It allows to monitor the status of the device, set its operating parameters and download the data and images obtained.

The block diagram of the measuring grid architecture is shown in fig. 4.12.

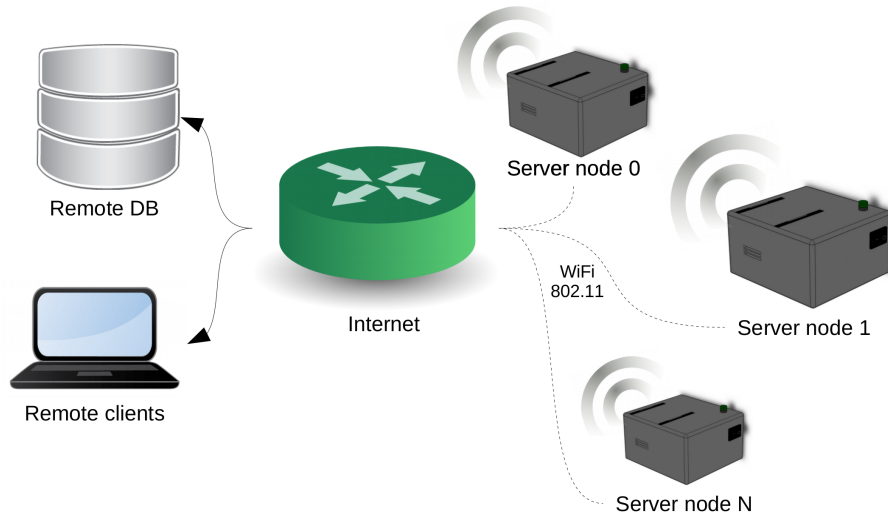


Figure 4.12: Measuring grid architecture.

Chapter 5

Calibration and data analysis

This chapter presents all the steps taken to calibrate the system in each of its components. It will deal with the operational procedures and the mathematical models used to estimate the uncertainty. The last part shows the data relating to the measurement campaign carried out during the development of the device.

5.1 Calibration

The calibration phase should not be confused with the adjustment phase ("Calibrazione" in Italian). In fact, while the first is a procedure by which the metrological characteristics of a measuring system are to be defined and is performed annually or more, the second one aims to increase the accuracy of measurement and is performed every time that it is intended to carry out a new measurement.

The calibration of a measurement system is carried out by comparing the values provided by the instrument under examination and by a reference instrument called the sample instrument. The characteristics that are to be defined during the calibration phase are:

- **Metrological characteristics:** in this phase, important plate data of the instrument are defined, such as the accuracy, linearity and reproducibility of the measurements.
- **Instrument precision:** the concept of precision in the metrological field is no longer used. On the other hand, it is an index of repeatability of the measure and represents the degree of convergence of the data collected individually with respect to the average value of the series to which they belong. In other words, it represents the variance of the data obtained with respect to the sample average.

- **Instrument trans-characteristic:** the trans-characteristic or transfer function is the mathematical relation between the quantity (or the quantities) entering an instrument and the output one. In fact, many types of measuring instruments transform physical quantities into signals, mostly electrical, that are easy to analyze and memorize.

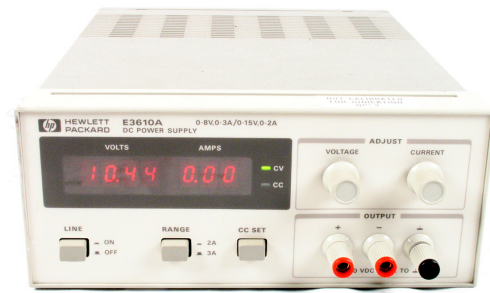
5.1.1 Pump calibration

The first source of error in the measurement and characterization of the particulate matter is the pump. In fact, it forces the air flow through the filter causing the suspended particles in it to be captured. Knowing precisely the volume of air analyzed is a fundamental step to obtain a correct concentration value.

The tools used for the calibration are the *BROOKS INSTRUMENT 3000 series* flow-meter and a *HP E3610A* bench power supply unit.



(a) Flow-meter (*BROOKS INSTRUMENT 3000 series*).



(b) Power supply unit (*HP E3610A*).

Figure 5.1: Calibration instrumentation.

Calibration procedure

The calibration of the pump was performed by connecting the outlet pipe of the flow-meter to the suction outlet of the pump, then the valve on the flow-meter inlet pipe was completely opened so as not to cause obstructions to the air flow. Once the movement of the indicator inside the flow-meter was checked and the connections were tightly closed, the latter was positioned vertically with the aid of a special support. This is of fundamental importance to avoid that the moving indicator inside the flow-meter can be moved or undergo friction due to the inclination of the measuring instrument. Last step was to connect the pump power terminals to the adjustable power supply and verify that everything worked.

The calibration process was carried out by supplying the pump with a DC voltage from 1 V to 5 V with steps of 0.5 V. For each voltage level the flow value indicated by the flow-meter was read. The data regarding the flux are expressed in [l/min].

Data and uncertainties estimation

The data obtained during the calibration phase are listed in the Table 5.1. From a first analysis it is possible to see that for voltages close to that of use equal to 5V, the behavior of the pump is characterized by a strong linearity. This is visible by comparing the data obtained (blue line) with the respective linear regression line (green line) shown in Figure 5.3.

The linear regression line equation is given by:

$$y = 0.095 \cdot x + 0.076667 \quad (5.1)$$

The value obtained by supplying the pump at 5 V and leaving the air inlet outlet free, is 0.41 l/min.

In real conditions of use, however, the air inlet of the pump is not free but is connected to the filter which is an obstruction to the air flow. For this reason, after having correctly connected the pump to the collection chambers where the filter is present, a new measurement was carried out which reported a value of 0.375 l/min.

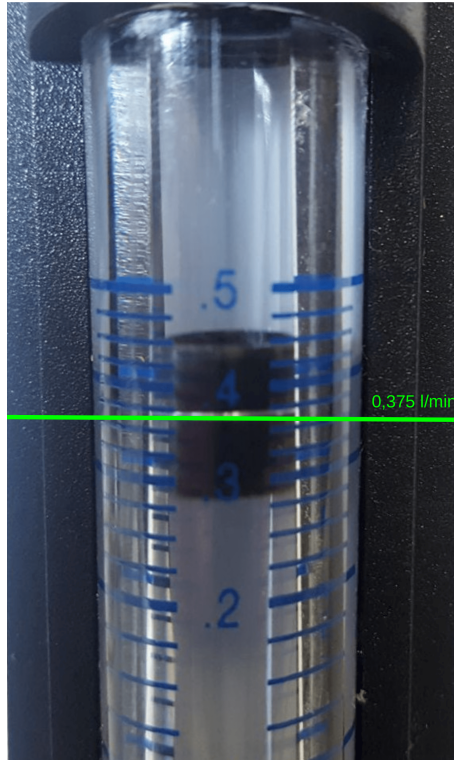


Figure 5.2: Measurement with the pump connected to the collecting chambers.

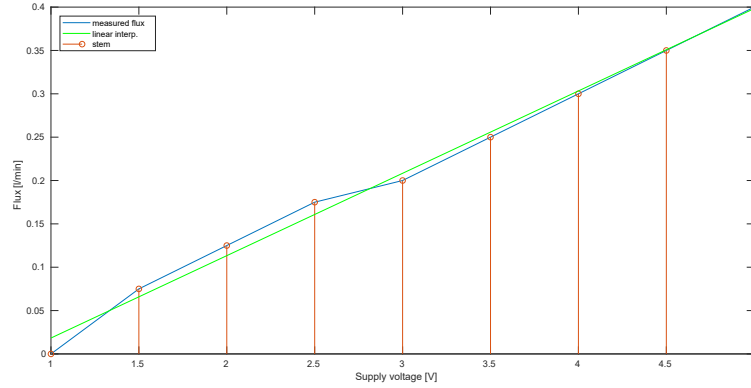


Figure 5.3: Pump vacuum profile.

Table 5.1: Vacuum profile data.

Supply voltage [V]	Flux [l/min]
1	0
1.5	0.075
2	0.125
2.5	0.175
3	0.2
3.5	0.25
4	0.3
4.5	0.35
5	0.4

The uncertainties of the values obtained are mainly due to the uncertainty of the instruments used. It is necessary to correctly evaluate these uncertainties in order to provide a range of variability of the measured value.

The first instrument to be analyzed is the variable power supply unit. From the datasheet, supplied by the manufacturer, it has an uncertainty equal to the 0.05% of the value read. Besides it must also be considered the error due to the fact that the device shows a value with only two decimal digits.

The same reasoning was also made for the flow-meter. The datasheet shows an uncertainty value equal to the 4% of the full-scale. To this the reading error, which is equal to half division, must be added. Being the full scale of the instrument 0.5 l/min and each division 0.025 l/min, the total uncertainty is equal to:

$$u_f = (0.04 \cdot 0.5) \text{ l/min} + \frac{1}{2} 0.025 \text{ l/min} = 0.0325 \text{ l/min} \quad (5.2)$$

Considering the above, and supplying the pump with a voltage of 5 V, the flow measurements, with and without the pump connected to the collection chambers, become:

$$\begin{aligned}\Delta_v &= (0.05 \cdot 5) \text{ V} = \pm 0.0025 \text{ V} \\ \Delta_f &= 0.095 \cdot 0.0025 + 0.076667 = \pm 0.0771 \text{ l/min} \\ F_{free} &= (0.4 \pm \pm 0.11) \text{ l/min}\end{aligned}\tag{5.3}$$

$$F_{filter} = (0.37 \pm \pm 0.11) \text{ l/min}\tag{5.4}$$

5.1.2 Software calibration

Another source of error in particulate measurement is particle recognition software. Testing a complex algorithm is not simple and is impossible to completely test a software considering all the possible cases. It is therefore necessary an empirical approach that is able to obtain acceptable results with an acceptable level of complexity.

Calibration procedure

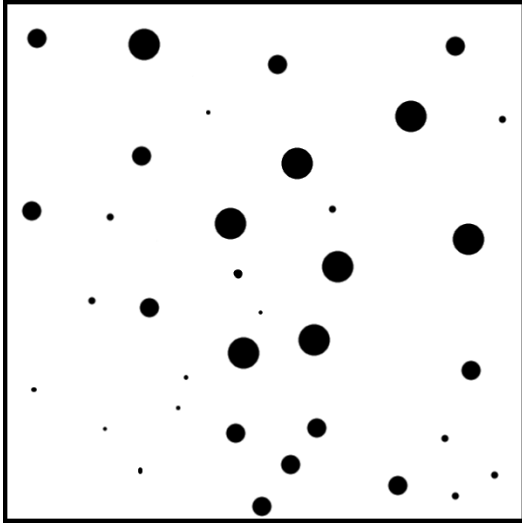
The system calibration is aimed at identifying the error regarding the number of identified particles and their dimensions. To do this, a three-step process was carried out.

In the first phase a test image was generated composed of a known number of particles having different diameters, also known. The image was created using the *Gimp* graphics software. The particles are black on a white background. This color choice was not random, in fact what we want to do at this stage is to test the software's capabilities in the best possible conditions, ie with the maximum contrast between the background and the particle to be analyzed.

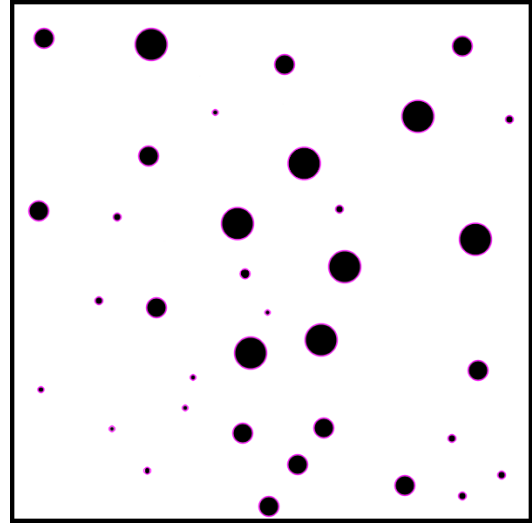
The test image consists of 35 spherical particles with a diameter between 6 px and 37 px. Figure 5.4 shows the image used as a test (a) and the relative image produced by the recognition software (b). The purple circumferences indicate the particle found: the center of the circumference corresponds to the center of mass of the particle while the diameter of the circumference corresponds to the diameter of a circumference with an equivalent area to that of the identified particle. The first test image dimensions are 1600 px × 1600 px.

In the second phase of the test, instead, the photo of a real filter was used as a test image. All the particles present were counted manually in order to compare the data with the software algorithm. Since the real particles are not perfectly spherical,

it is difficult to calculate the area manually. For this reason, in this phase, only the correspondence between the number of particles present in the test image and those actually identified was analyzed. Given the large number of particles present on the filter, it was decided to analyze only a small part of the image in order to facilitate manual counting operations. The second test image dimensions are $600 \text{ px} \times 600 \text{ px}$.

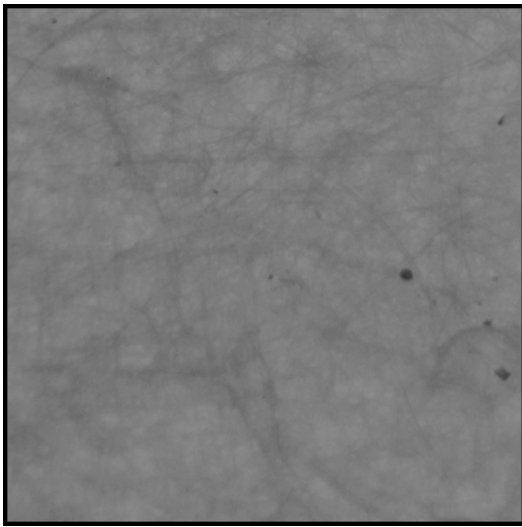


(a) Input image.

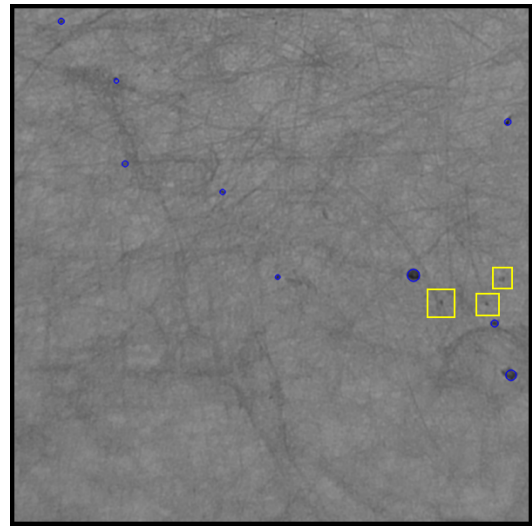


(b) Output image.

Figure 5.4: First test images.



(a) Input image.



(b) Output image.

Figure 5.5: Second test images.

The last phase was carried out to obtain a conversion factor useful for moving from pixels to micrometers. To obtain this value a millimeter calibration mask was photographed. The image was captured with the same parameters with which the filters are analyzed.

Figure 5.6 shows the result obtained.

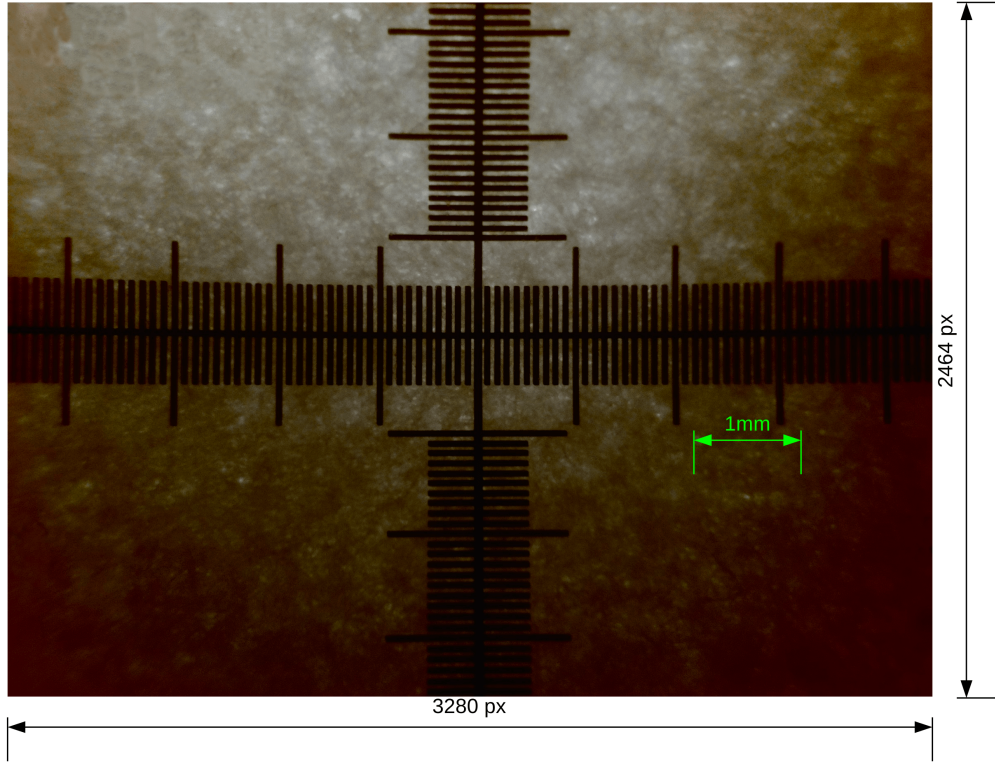


Figure 5.6: Size calibration test.

The captured image has a width of 7.8 mm and a height of 6.6 mm. Since the resolution is 3280 px \times 2464 px, and considering the distortion error due to the lens, each pixel has a size of 2.7 μ m.

Thanks to the value just found, the software has been modified in order to provide information about the size of the particles found no longer in pixels but in micrometers.

Data and uncertainties estimation

The data obtained in the first test phase are listed in the Table 5.3. In all the tests performed, the software has always recognized all the particles in the image. As for the dimensions they are slightly different from those of entry. The difference is so small that the error can also be due to the software by which these

were generated. In any case, variations so small in size do not involve any type of error in the evaluation of atmospheric particulate matter.

The second test is different. As mentioned above, it was impossible to verify the exact correspondence of the all particle size. For this reason only the number of found particles was analyzed.

Referring to Figure 5.5a, there are 12 particles in it but only 9 are identified by the software. Figure 5.5b shows in blue the particles correctly identified, with their dimensions, and in yellow the unidentified particles.

This test phase was repeated several times in order to obtain a mean value of the error percentage.

According to the obtained data , the average error rate is the 12.36%.

Table 5.2: Second test: data

N.	Input particles	Detected particles	Error [%]
1	15	13	13.33
2	12	9	25.00
3	22	21	4.54
4	18	15	16.60
5	8	8	0.00
6	12	8	33.33
7	24	22	8.33
8	20	19	5.00
9	11	11	0.00
10	29	25	13.79
11	24	20	16.66
12	17	17	0.00
13	23	20	13.04
14	26	21	19.23
15	18	15	16.60

Table 5.3: First test: data

N.	Input particles diameter [px]	Detected particles diameter [px]
1	23.00	23.0217285156
2	9.00	9.0553855896
3	23.00	23.0217285156
4	9.00	9.0553855896
5	7.00	7.81024980545
6	23.00	23.0217285156
7	9.00	9.0553855896
8	23.00	23.0217285156
9	23.00	23.0217285156
10	7.00	7.0
11	23.00	23.0217285156
12	37.00	37.0135116577
13	37.00	37.0135116577
14	23.00	23.0217285156
15	9.00	9.0553855896
16	11.00	11.4978895187
17	37.00	37.0135116577
18	37.00	37.0
19	9.00	9.0553855896
20	37.00	37.0
21	9.00	9.0553855896
22	23.00	23.0217285156
23	37.00	37.2826156616
24	23.00	23.0217285156
25	9.00	9.0553855896
26	37.00	37.0
27	23.00	23.0217285156
28	23.00	23.0217285156
29	37.00	37.0135116577
30	23.00	23.0217285156
31	6.00	6.16227769852
32	6.00	6.45100975037
33	6.00	6.16227769852
34	6.00	6.16227769852
35	6.00	6.45100975037

5.1.3 System trans-characteristic

The software provides the data in $[\mu\text{m}^2/\text{m}^3]$, it is therefore necessary to find a conversion factor that reports the data in the most common form $[\mu\text{g}/\text{m}^3]$. This coefficient was obtained by calculating, for each measure, the ratio between the reference value supplied by the Region and that obtained by the device. Then the arithmetic mean of the results was performed.

This coefficient, not being a density, is not to be understood as a conversion factor that allows to convert values expressed in $[\mu\text{m}^2/\text{m}^3]$ to values in $[\mu\text{g}/\text{m}^3]$. It aims to verify if the trend of the data obtained by the device reflects those provided by the Region.

The trans-characteristic of the system is:

$$C = \left[\frac{Q \cdot A_p}{A_e} \cdot \frac{t}{1000} \right]^{-1} \cdot P_x \cdot \alpha \quad (5.5)$$

Where C is the particulate matter concentration, Q is the pump flow-rate expressed in $[\text{l}/\text{min}]$, A_p and A_e the photographed and the exposed area in centimeters respectively, t the time exposure in minutes, P_x the total area in square micrometers of all detected particles and α the conversion factor.

For the calculation of the uncertainties, the deterministic model is applied to the Equation 5.5.

$$\Delta C = \left| \frac{\partial C}{\partial Q} \right| \Delta Q + \left| \frac{\partial C}{\partial A_p} \right| \Delta A_p + \left| \frac{\partial C}{\partial A_e} \right| \Delta A_e + \left| \frac{\partial C}{\partial t} \right| \Delta t + \left| \frac{\partial C}{\partial P_x} \right| \Delta P_x \quad (5.6)$$

Analyzing the formula one can make the measurement uncertainties due to the exposed and photographed areas and that related to the exposure time negligible. In fact, they show variations with respect to the expected value that are very small so as not to change the concentration value obtained.

$$\Delta C = \left| \frac{\partial C}{\partial Q} \right| \Delta Q + \left| \frac{\partial C}{\partial P_x} \right| \Delta P_x \rightarrow \Delta C = \frac{\Delta P_x}{\bar{P}_x} + \frac{\Delta Q}{\bar{Q}} \quad (5.7)$$

5.2 Data analysis

During the development phase a measurement campaign was conducted in two different cities: Turin and Caserta.

For each of them 11 samples were collected in a period starting from March 7, 2018 to March 19, 2018 with the exception of two days (8 and 10) in which the measure

could not be performed due to malfunctions of the device. Starting from these data, the system was calibrated and set up.

The raw data provided by the instrument show the diameter in pixels of the particles found. Starting from this value, and assuming that the particles were perfectly spherical, the total area of the identified particles was calculated.

Further step was to normalize the value found. In fact, it is standard practice for measurements of atmospheric particulate to report the concentration per cubic meter of air. For compliance, the value obtained was multiplied by a factor that takes into account the flow in [l/ min] of the pump and the exposure time in minutes [min]. In this way, values were obtained with units of $[\mu\text{m}^2/\text{m}^3]$.

At this value an additional corrective factor has been multiplied taking into account the difference in area between the filter section exposed to the air flow and that effectively analyzed by the camera.

Figure 5.7 and Figure 5.8 show the reference data provided by the Regions (a) and those measured by the instrument (b), during the sampling period.

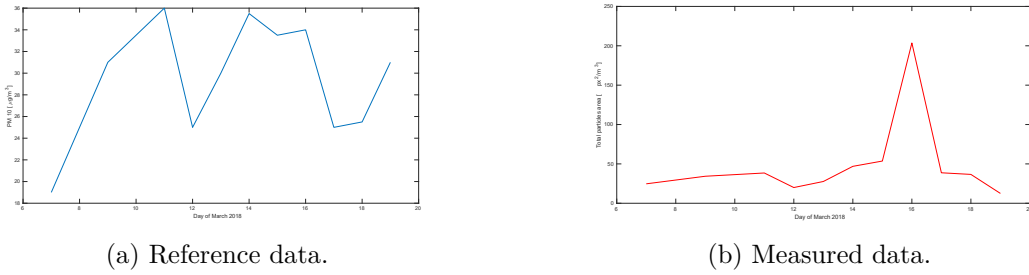


Figure 5.7: Data collected in Caserta.

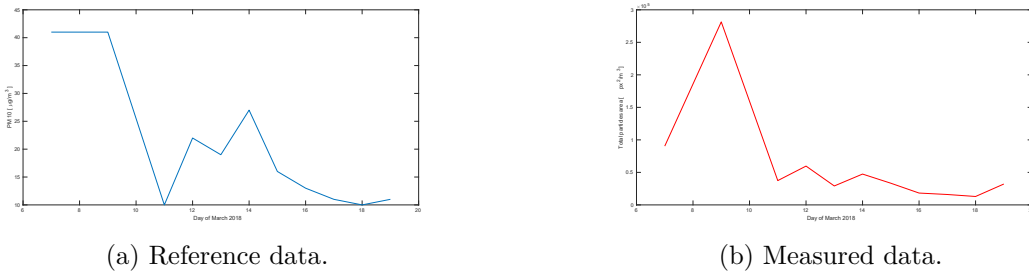


Figure 5.8: Data collected in Turin.

Since the two data series (those provided by the Regions and those measured by the instrument) are characterized by different physical quantities, it is not possible

to make a direct comparison between them. Nevertheless this phase is very important in order to evaluate the goodness of the measurement system. To do this a conversion factor α has been calculated, which allows to compare the data at our disposal.

This coefficient is referred only to the data collected in Caserta. It was obtained by calculating, for each measure, the ratio between the reference value supplied by the Region and that obtained by the device. Then the arithmetic mean and the standard deviation of the collected results was performed.

The mean value is $\bar{\alpha} = 2.72 \cdot 10^{-4}$ and its standard deviation is $\sigma_{\alpha} = 1.67 \cdot 10^{-4}$.

This factor must be used only to compare the trend of the data obtained by the instrument with those provided by the Regions. It is not a direct conversion factor.

For the sake of completeness, the calculation of α was also performed for the data collected in Turin. All the values are listed in Table 5.4 and in Table 5.5.

Table 5.4: α evaluation from data collected in Caserta.

Day	Reference value [$\mu\text{g}/\text{m}^3$]	Measured area [$\mu\text{m}^2/\text{m}^3$]	α [$\mu\text{g}/\mu\text{m}^2$]
7	19	85.54E+03	222.12E-06
9	31	118.86E+03	260.80E-06
11	36	133.21E+03	270.25E-06
12	25	69.19E+03	361.30E-06
13	30	95.79E+03	313.19E-06
14	35.5	162.48E+03	218.49E-06
15	35.5	185.74E+03	191.13E-06
16	34	705.34E+03	48.20E-06
17	25	134.10E+03	186.42E-06
18	25.5	126.98E+03	200.83E-06
19	31	43.29E+03	716.03E-06
			$\bar{\alpha} =$ 2.72E-04
			$\sigma_{\alpha} =$ 1.67E-04

Table 5.5: α evaluation from data collected in Turin.

Day	Reference value [$\mu\text{g}/\text{m}^3$]	Measured area [$\mu\text{m}^2/\text{m}^3$]	α [$\mu\text{g}/\mu\text{m}^2$]
7	41	90.38E+03	453.63E-06
9	41	281.44E+03	145.68E-06
11	10	37.39E+03	267.46E-06
12	22	59.69E+03	368.56E-06
13	19	29.13E+03	652.16E-06
14	27	47.41E+03	569.45E-06
15	16	33.36E+03	479.66E-06
16	13	18.22E+03	713.35E-06
17	11	16.00E+03	687.46E-06
18	10	12.95E+03	772.49E-06
19	11	32.12E+03	342.52E-06
			$\bar{\alpha} = 4.96\text{E-}04$
			$\sigma_{\alpha} = 2.01\text{E-}04$

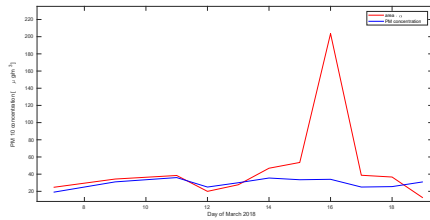
After obtaining an average value of alpha, this value was multiplied by the data relating to the total area of the identified particles.

The trend of the results obtained was compared with that of the concentration data of the atmospheric particulate supplied by the Regions.

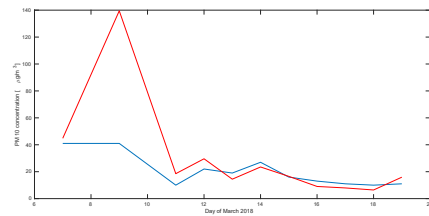
The Figure 5.9 shows the results obtained both on the data collected in Caserta (a) and on those collected in Turin (b). The red line represents the data provided by the device while the blue one are those provided by the Regions.

Taking into account only the data collected in Caserta, it is possible to notice that there is a quite significant correlation between the data obtained and the actual levels of pollution except for a value probably due to an analysis error.

The same happens when analyzing the data collected in Turin.



(a) Data collected in Caserta.



(b) Data collected in Turin.

Figure 5.9: Analysis of data trends.

As said before, the factor α is not suitable to be used as a conversion factor from square pixels to micrograms. To get out of the data instrument in the canonical form $[\mu\text{g}/\text{m}^3]$ it is necessary first of all to calculate the total volume of the particles identified by the software.

Recall that the algorithm provides only the pixel size of the particle diameter. Assuming that the particle is perfectly spherical it is possible to calculate the volume. This operation is performed for all the identified particles and the obtained values summed. What is achieved at the end is the total volume of particles identified in the analyzed airflow.

As seen in the case of the area, the value is normalized to the single cubic meter of air and converted in μm . The final output data will be in $[\mu\text{m}^3/\text{m}^3]$.

Knowing the volume of the particles, to get their weight it is sufficient to know their density. The problem of atmospheric particulate matter is that it is a mixture of heterogeneous substances. So there is no single usable density value.

This phase of the data analysis was aimed at identifying a mean density value for the urban particulate of the cities of Turin and Caserta.

It was obtained by calculating, for each measure, the ratio between the reference concentration value supplied by the Region and the total volume calculated before. Then the arithmetic mean and the standard deviation of the collected results was performed.

Table 5.6: Density ρ evaluation from data collected in Caserta.

Day	Reference value $[\mu\text{g}/\text{m}^3]$	Measured volume $[\mu\text{m}^3/\text{m}^3]$	ρ $[\text{kg}/\text{m}^3]$
7	19	1.96E+06	9,69E+03
9	31	1.76E+06	17,65E+03
11	36	2.11E+06	17,09E+03
12	25	1.25E+06	20,07E+03
13	30	1.77E+06	16,95E+03
14	35,5	2.45E+06	14,50E+03
15	35,5	2.89E+06	12,28E+03
16	34	20.65E+06	1,65E+03
17	25	2.34E+06	10,68E+03
18	25,5	2.16E+06	11,81E+03
19	31	642.68E+03	48,24E+03
			$\bar{\rho} = 1.64\text{E}+04$
			$\sigma_{\rho} = 1.16\text{E}+04$

Table 5.7: Density ρ evaluation from data collected in Turin.

Day	Reference value [$\mu\text{g}/\text{m}^3$]	Measured volume [$\mu\text{m}^3/\text{m}^3$]	ρ [kg/m^3]
7 ^ã	41	20.44E+06 ^ã	2,01E+03 ^{ãã}
9 ^ã	41	112.32E+06	365,04E+00
11	10	5.44E+06 ^{ãã}	1,84E+03 ^{ãã}
12	22	10.97E+06 ^ã	2,01E+03 ^{ãã}
13	19	3.74E+06 ^{ãã}	5,08E+03 ^{ãã}
14	27	7.77E+06 ^{ãã}	3,48E+03 ^{ãã}
15	16	4.58E+06 ^{ãã}	3,49E+03 ^{ãã}
16	17	1.85E+06 ^{ãã}	9,19E+03 ^{ãã}
17	18	1.52E+06 ^{ãã}	11,82E+03 ^ã
18	19	1.11E+06 ^{ãã}	17,15E+03 ^ã
19	20	4.33E+06 ^{ãã}	4,62E+03 ^{ãã}
			$\bar{\rho} = 5.55\text{E}+03$
			$\sigma_{\rho} = 5.12\text{E}+03$

Unlike the previous case, the data obtained from the calculation of the average density are not to be considered acceptable.

In fact, they present very high values, incompatible with the densities related to the individual components that compose the atmospheric particulate.

For reasons of time, no further measurements or checks were carried out. This factor is still an element of study in order to find a solution that allows a more complete characterization of the system.

One of the characteristics that make this type of device unique is the ability to perform a spectrographic analysis of the individual particles. It is accomplished by analyzing the different properties of the single particle in the different images captured at different wavelengths.

During the development phase it was not possible to complete this characterization but we analyzed some filters exposed with a scanning electron microscope. It is also equipped with an X-ray module that has allowed a precise characterization of the chemical composition of the analyzed particle.

Figure 5.10 shows two examples of particles photographed with the electronic microscope. Table 5.8 and Figure 5.11 shows data about the chemical analysis.

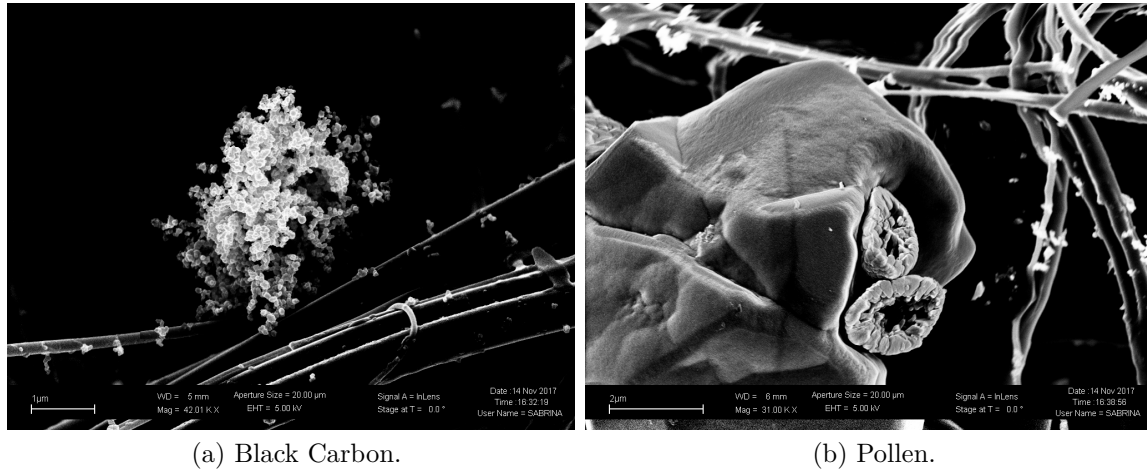


Figure 5.10: Electronic microscope photos.

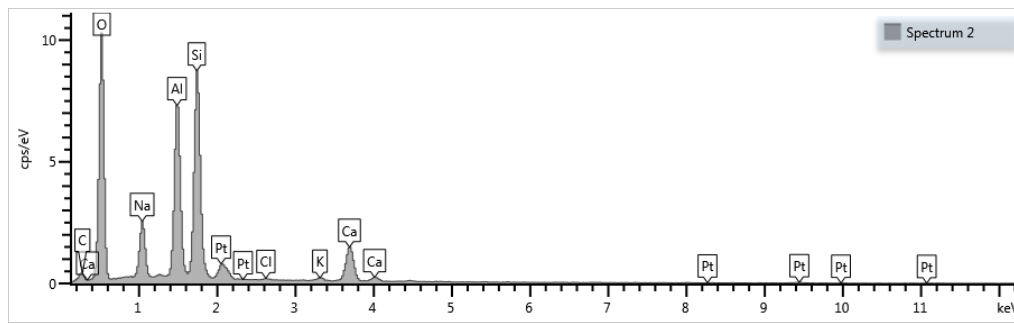


Figure 5.11: Sample chemical composition.

Table 5.8: Chemical analysis data.

Element	Atomic %
C	08.35.00
O	65.12.00
Na	04.58.00
Al	09.11.00
Si	10.43.00
Cl	00.07.00
K	00.20.00
Ca	02.53.00
Total:	100.00.00

Chapter 6

Conclusions

The aim of this thesis work was to develop an innovative solution for the measurement and characterization of the atmospheric particulate matter. This task has been completed even if, as it is possible to imagine, a final level of quality has not been reached to consider the development process completed.

The data obtained from the measurement campaign, described in detail in Chapter 5, show satisfactory results considering that the device used is a prototype still under development. This bodes well for future developments of this device.

The system features make this solution a valid low-cost alternative to the commercial air sampling devices.

During the whole research and development phase the biggest issue was the lack of a reference literature regarding this type of analysis. In fact, although today there are many techniques for measuring the concentration of atmospheric particulate, none of those is based on the same principle as that developed by us.

Working on such an innovative project was exciting because it allowed me to combine the theoretical phase, where we have studied the space of possible solutions, with an experimental phase in which, with very empirical processes, we have tested new technologies and solutions.

Another very important aspect was to have followed the development process in all its parts, starting from the feasibility study, passing through the design and construction of the various prototypes up to the calibration and testing phase. This allowed me to fully understand the critical aspects of each phase and develop different solutions and approaches for each context.

Future work

There is certainly plenty of room for improvement in the device characteristics. To develop them in the best way, it is required a work on two main fronts: the optical measurement unit and the particle recognition algorithm.

As regards the first development front, it is necessary to increase the measurement resolution and the quality of the images captured.

A first solution to this problem may be to replace the lens with an optical assembly which has superior characteristics in terms of magnification and sharpness of the images. In order to be able to perform the spectrographic analysis, the device should be equipped with an automatic focusing system so as to be able to adapt to the various backlight sources installed.

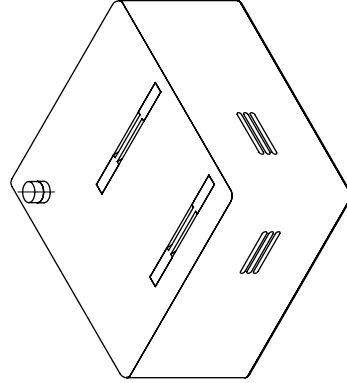
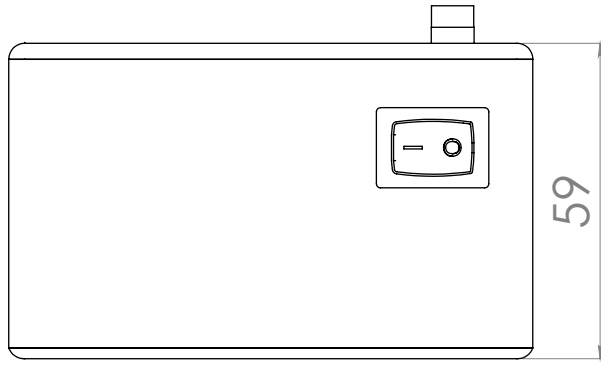
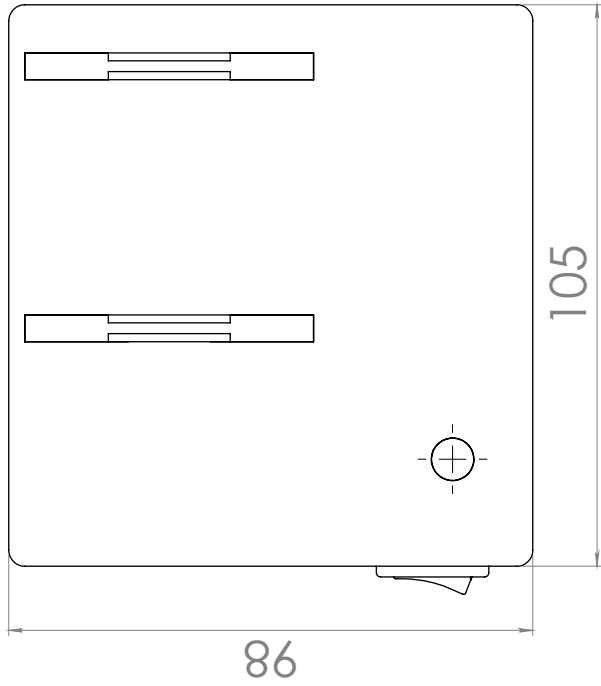
The second working front concerns the software. Improving the identification and particle analysis algorithm is the first step in reducing measurement errors. To do this, it is necessary to perform a large number of measurements in order to identify the optimal average parameters with which to calibrate the software.


From the analysis of the data obtained during the measurement campaign, a link between the concentration of atmospheric particulate and climatic conditions has been noticed. Parallel to the above, could be developed an adaptive software, capable of modifying the parameters defined above on the basis of information external to the measurement, such as temperature, humidity, rain levels, etc.

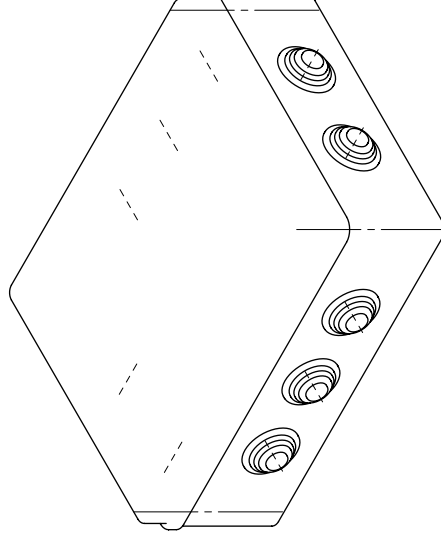
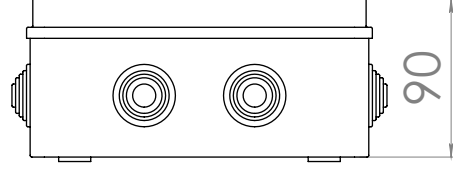
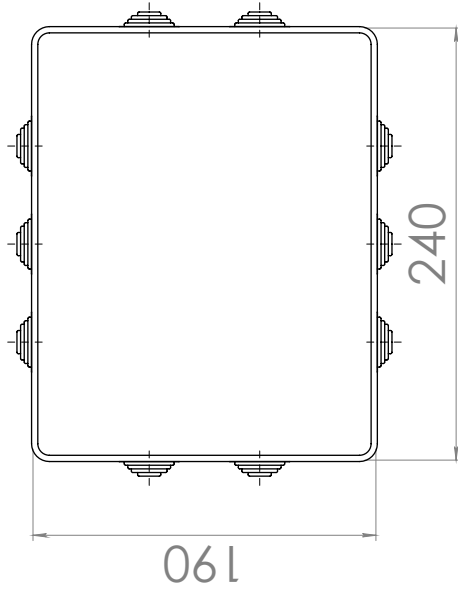
Further improvements may be those related to the communication system. Since the amount of data transmitted is very small, it is possible to equip the device with *LoRa* transceiver. It allows the transmission of data over a large distance with very low power consumption, thus providing the basis for creating a monitoring network much more widespread than those installed today, in order to obtain more detailed information on the air pollution level.

Appendix A


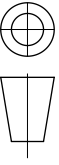
Drawings



UNI EN 22768-mK		0.5x45°	
 POLITECNICO DI TORINO	CANDIDATO	DATA	
	Francesco VITIELLO	20/03/2018	
	CORSO DI LAUREA	SCALA	
	Ingegneria elettronica	1:1	
OGGETTO	PM monitor	PESO (Kg)	
		0,280	
DESCRIZIONE	Portable version	FOGLIO	DISEGNO N.
		A3 1/1	1
Politecnico di Torino - Corso Duca degli Abruzzi, 24 - 10129 TORINO			



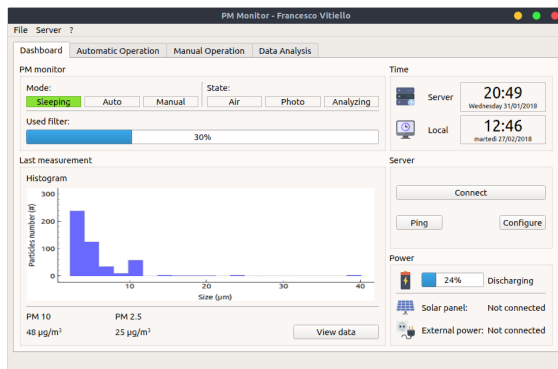
UNI EN 22768-mK 0.5x45°

	POLITECNICO DI TORINO		CANDIDATO Francesco VITIELLO CORSO DI LAUREA Ingegneria elettronica	DATA 20/03/2018
				SCALA 1:3
OGGETTO	PM monitor		PESO (Kg) 1.92	
DESCRIZIONE	Automatic version		FOGLIO A3 1/1	
Politecnico di Torino - Corso Duca degli Abruzzi, 24 - 10129 TORINO				

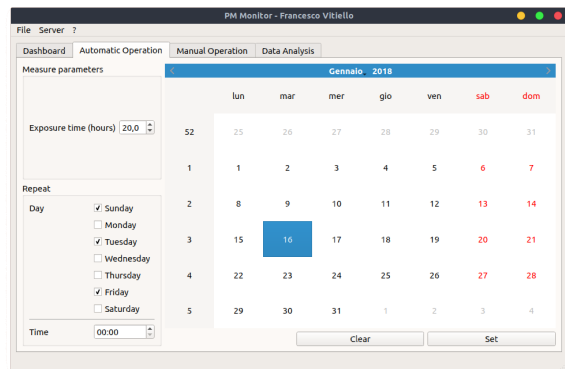
Appendix B

Software: client application

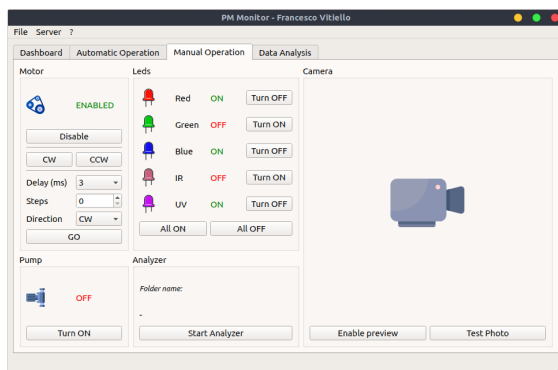
B.1 Screenshots



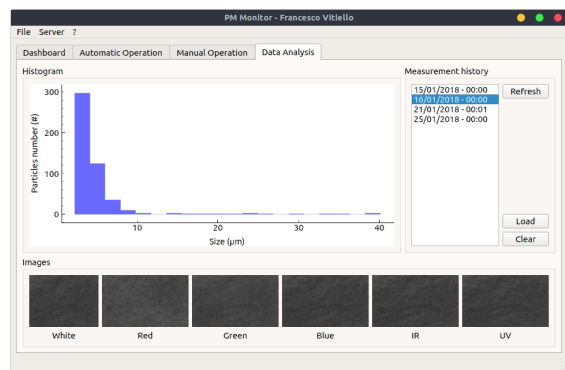
(a) Dashboard.



(b) Automatic operation.



(c) Manual operation.



(d) Data analysis.

Figure B.1: Client software screenshots.

B.2 Main.py

```
#!/usr/bin/python
2 # -*- coding: utf-8 -*-
import Pyro4
4 from PyQt5 import QtWidgets
import sys
6 import os
import cv2
8 import time
import pickle
10 import threading
import numpy as np
12 from design import Ui_MainWindow
import pyqtgraph as pg
14 from pyqtgraph.Qt import QtGui
from analyzer import analyzer
16 import shutil
import paramiko
18 from scp import SCPClient

20 pg.setConfigOption('background', 'w')
pg.setConfigOption('foreground', 'k')
22 null_image = 'Img/white.jpg'
server_home = '/home/pi/Desktop/sw/'
24 firstDataUpdate = True
state = 0
26 pm_server = 0
ssh = paramiko.SSHClient()
28 ssh.load_system_host_keys()
ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
30 previewEnabled = False
serverConnected = False
32 firstConnected = False
server_mode = -1
34

36 class Video:

38     def __init__(self, capture):
        self.capture = capture
40         self.currentFrame = np.array([]) # @UndefinedVariable

42     def captureNextFrame(self):
        (ret, readFrame) = self.capture.read()
44         if ret:
            self.currentFrame = cv2.cvtColor(readFrame,
46                                             cv2.COLOR_BGR2RGB) #
                                                    @UndefinedVariable
```

```
48     def convertFrame( self ):
49         try:
50             (height , width) = self.currentFrame.shape[:2]
51             img = QtGui.QImage( self.currentFrame, width, height ,
52                                 QtGui.QImage.Format_RGB888)
53             img = QtGui.QPixmap.fromImage(img)
54             self.previousFrame = self.currentFrame
55             return img
56         except:
57             return None
58
59
60     class MyFirstGuiProgram( Ui_MainWindow ):
61
62         def __init__( self , mainwindow ):
63             Ui_MainWindow.__init__( self )
64             self.setupUi( mainwindow )
65             self.setStateGui()
66
67             # hidden elements
68
69             self.prog_download = QtGui.QProgressBar() # @UndefinedVariable
70
71             # graph init
72
73             global graph_data
74             graph_data = self.graph_data
75             graph_data.setLabels( left=( 'Probability' , '%' ), bottom=( 'Size' ,
76                                     'um' ) )
77
78             # menu actions
79
80             self.action_sleeping.triggered.connect( self.actionSleeping )
81             self.action_auto.triggered.connect( self.actionAuto )
82             self.action_manual.triggered.connect( self.actionManual )
83
84             # Connect ui elements
85
86             self.btn_refresh.clicked.connect( self.button_refresh )
87             self.btn_load.clicked.connect( self.button_load )
88             self.btn_clear.clicked.connect( self.button_clear )
89
90             # manual operation
91
92             self.btn_pump.clicked.connect( self.button_pump )
93             self.btn_led_red.clicked.connect( self.button_led_red_toggle )
94             self.btn_led_green.clicked.connect( self.button_led_green_toggle )
95             )
96             self.btn_led_blue.clicked.connect( self.button_led_blue_toggle )
```

```
96     self.btn_led_ir.clicked.connect(self.button_led_ir_toggle)
97     self.btn_led_uv.clicked.connect(self.button_led_uv_toggle)
98     self.btn_led_allon.clicked.connect(self.button_led_allon)
99     self.btn_led_alloff.clicked.connect(self.button_led_alloff)
100    self.btn_motor_cw.clicked.connect(self.button_motor_cw)
101    self.btn_motor_ccw.clicked.connect(self.button_motor_ccw)
102    self.btn_motor_go.clicked.connect(self.button_motor_go)
103    self.btn_motor_enable.clicked.connect(self.button_motor_enable)
104    self.btn_preview.clicked.connect(self.button_preview_toggle)

106    # dashboard

108    self.btn_connect.clicked.connect(self.button_connect)

110    def actionSleeping(self):
111        msg = QtGui.QMessageBox() # @UndefinedVariable
112        msg.setIcon(QtGui.QMessageBox.Warning) # @UndefinedVariable
113        msg.setText('Do you want to set the SLEEP mode?')
114        msg.setWindowTitle('Change mode')
115        msg.setStandardButtons(QtGui.QMessageBox.Yes
116                               | QtGui.QMessageBox.No) #
117                               @UndefinedVariable

118        retval = msg.exec_()
119        if retval == QtGui.QMessageBox.Yes: # @UndefinedVariable
120            print 'Yes.'
121            pm_server.setCommand('F', 'MODE', 'SLEEP')
122        else:
123            pass

124    def actionAuto(self):
125        msg = QtGui.QMessageBox() # @UndefinedVariable
126        msg.setIcon(QtGui.QMessageBox.Warning) # @UndefinedVariable
127        msg.setText('Do you want to set the AUTO mode?')
128        msg.setWindowTitle('Change mode')
129        msg.setStandardButtons(QtGui.QMessageBox.Yes
130                               | QtGui.QMessageBox.No) #
131                               @UndefinedVariable

132        retval = msg.exec_()
133        if retval == QtGui.QMessageBox.Yes: # @UndefinedVariable
134            print 'Yes.'
135            pm_server.setCommand('F', 'MODE', 'AUTO')
136        else:
137            pass

138    def actionManual(self):
139        msg = QtGui.QMessageBox() # @UndefinedVariable
140        msg.setIcon(QtGui.QMessageBox.Warning) # @UndefinedVariable
141        msg.setText('Do you want to set the MANUAL mode?')
142        msg.setWindowTitle('Change mode')
143        msg.setStandardButtons(QtGui.QMessageBox.Yes
```

```
144         | QtGui.QMessageBox.No) #
        @UndefinedVariable
    retval = msg.exec_()
146    if retval == QtGui.QMessageBox.Yes: # @UndefinedVariable
        print 'Yes.'
148        pm_server.setCommand('F', 'MODE', 'MANUAL')
    else:
150
        pass
152
def play(self):
154    global previewEnabled
    global pm_server
156    pm_server.setCommand('C', 'PREVIEW', 1)
    self.video = Video(cv2.VideoCapture('http://192.168.0.7:8090/'))
    # @UndefinedVariable
158    try:
        self.video.captureNextFrame()
160        self.lbl_video.setPixmap(self.video.convertFrame())
        self.lbl_video.setScaledContents(True)
162    except TypeError:
        print 'No frame'
164    if previewEnabled:
        t = threading.Timer(0.1, self.play)
166        t.start()

168    def button_preview_toggle(self):
        global pm_server
170        global previewEnabled
        previewEnabled = not previewEnabled
172
        if previewEnabled:
174            t = threading.Thread(name='play', target=self.play)
            t.start()
176        else:
            pm_server.setCommand('C', 'PREVIEW', 0)
178
def button_connect(self):
180    global pm_server
    global ssh
182    global serverConnected
    global firstConnected
184    firstConnected = True
    pm_server = Pyro4.Proxy('PYRONAME:example.greeting')
186    try:
        pm_server._pyroBind()
188        serverConnected = True
    except Pyro4.errors.CommunicationError:
190        print 'NOPE IT IS NOT REACHABLE!'
    ssh.connect('192.168.0.7', username='pi', password='raspberry')
```



```
192     def button_refresh(self):
194         self.updateData()

196     def button_clear(self):

198         # clear all images

200         self.lbl_white.setPixmap(QtGui.QPixmap(null_image).scaled(self.
            lbl_white.size()))
        self.lbl_red.setPixmap(QtGui.QPixmap(null_image).scaled(self.
            lbl_red.size()))
202         self.lbl_green.setPixmap(QtGui.QPixmap(null_image).scaled(self.
            lbl_green.size()))
        self.lbl_blue.setPixmap(QtGui.QPixmap(null_image).scaled(self.
            lbl_blue.size()))
204         self.lbl_ir.setPixmap(QtGui.QPixmap(null_image).scaled(self.
            lbl_ir.size()))
        self.lbl_uv.setPixmap(QtGui.QPixmap(null_image).scaled(self.
            lbl_uv.size()))

206         # clear graph

208         graph_data.clear()

210     def button_load(self):
212         for the_file in os.listdir('tmp'):
            file_path = os.path.join('tmp', the_file)
214             if os.path.isfile(file_path):
                os.unlink(file_path)
216             elif os.path.isdir(file_path):
                shutil.rmtree(file_path)

218         # add progressBar

220         self.statusBar.addWidget(self.prog_download)
222         self.prog_download.setValue(0)

224         # startDownload

226         self.downloadData()
        self.statusBar.removeWidget(self.prog_download)
228         self.statusBar.showMessage('Download completed.', 5000)

230     def downloadData(self):
        value = 0
232         global server_home
        data_filenames = [
234             '/white.bmp',
            '/red.bmp',
```

```
236         '/green.bmp',
237         '/blue.bmp',
238         '/ir.bmp',
239         '/uv.bmp',
240         '/keypoints.txt',
241     ]
242
243     index = self.list_history.row(self.list_history.currentItem())
244     if index >= 0:
245         os.chdir('tmp')
246
247         # download images
248
249         scp = SCPClient(ssh.get_transport())
250         path = 'Data/' + state[3][index]
251         for name in data_filenames:
252             scp.get(server_home + path + name)
253             value += 10
254             self.prog_download.setValue(value)
255
256         # load images
257
258         self.lbl_white.setPixmap(QtGui.QPixmap('white.bmp'
259                                                ).scaled(self.
260                                                           lbl_white.size()
261                                                           ))
262         self.lbl_red.setPixmap(QtGui.QPixmap('red.bmp'
263                                               ).scaled(self.lbl_red.
264                                                         size()))
265         self.lbl_green.setPixmap(QtGui.QPixmap('green.bmp'
266                                                 ).scaled(self.
267                                                           lbl_green.size()
268                                                           ))
269         self.lbl_blue.setPixmap(QtGui.QPixmap('blue.bmp'
270                                                ).scaled(self.
271                                                           lbl_blue.size()))
272         self.lbl_ir.setPixmap(QtGui.QPixmap('ir.bmp'
273                                              ).scaled(self.lbl_ir.
274                                                        size()))
275         self.lbl_uv.setPixmap(QtGui.QPixmap('uv.bmp'
276                                              ).scaled(self.lbl_uv.
277                                                        size()))
278
279         self.prog_download.setValue(90)
280
281         # load keypoints data
282
283         index = pickle.loads(open('keypoints.txt', 'rb').read())
284         keypoints = []
285         for point in index:
```

```
278         temp = cv2.KeyPoint( # @UndefinedVariable
279             x=point[0][0],
280             y=point[0][1],
281             _size=point[1],
282             _angle=point[2],
283             _response=point[3],
284             _octave=point[4],
285             _class_id=point[5],
286         )
287         keypoints.append(temp)
288
289         # plot data
290
291         size = []
292         for point in keypoints:
293             size.append(point.size)
294         (y, x) = np.histogram(size, bins=20) # @UndefinedVariable
295         graph_data.plot(x, y, stepMode=True, fillLevel=0, brush=(0,
296             0, 255, 150))
297
298         os.chdir('..')
299         scp.close()
300         self.prog_download.setValue(100)
301
302         # manual operation
303
304         def button_pump(self):
305             global pm_server
306             pm_server.setCommand('P', 'TOGGLE')
307
308         def button_led_red_toggle(self):
309             global pm_server
310             pm_server.setCommand('L', 'TOGGLE', 'RED')
311
312         def button_led_green_toggle(self):
313             global pm_server
314             pm_server.setCommand('L', 'TOGGLE', 'GREEN')
315
316         def button_led_blue_toggle(self):
317             global pm_server
318             pm_server.setCommand('L', 'TOGGLE', 'BLUE')
319
320         def button_led_ir_toggle(self):
321             global pm_server
322             pm_server.setCommand('L', 'TOGGLE', 'IR')
323
324         def button_led_uv_toggle(self):
325             global pm_server
326             pm_server.setCommand('L', 'TOGGLE', 'UV')
327
328         def button_led_allon(self):
```

```
328     global pm_server
pm_server.setCommand('L', 'SET', 'RED', 1)
330 pm_server.setCommand('L', 'SET', 'GREEN', 1)
pm_server.setCommand('L', 'SET', 'BLUE', 1)
332 pm_server.setCommand('L', 'SET', 'IR', 1)
pm_server.setCommand('L', 'SET', 'UV', 1)
334
def button_led_alloff(self):
336     global pm_server
pm_server.setCommand('L', 'SET', 'RED', 0)
338 pm_server.setCommand('L', 'SET', 'GREEN', 0)
pm_server.setCommand('L', 'SET', 'BLUE', 0)
340 pm_server.setCommand('L', 'SET', 'IR', 0)
pm_server.setCommand('L', 'SET', 'UV', 0)
342
def button_motor_cw(self):
344     global pm_server
pm_server.setCommand('M', 'STEP', 5, 1, 'CW')
346
def button_motor_ccw(self):
348     global pm_server
pm_server.setCommand('M', 'STEP', 5, 1, 'CCW')
350
def button_motor_go(self):
352     global pm_server
delay = int(self.cmb_delay.currentText())
354     direction = self.cmb_direction.currentText()
steps = int(self.spin_steps.value())
356     pm_server.setCommand('M', 'STEP', delay, steps, direction)

358 def button_motor_enable(self):
    global pm_server
360     pm_server.setCommand('M', 'TOGGLE')

362 def setStateGui(self):
    global pm_server
364     global state
    global firstDataUpdate
366     global previewEnabled
    global serverConnected
368     global firstConnected
    global server_mode
370
    # check connection
372
    if firstConnected:
374         try:
            pm_server.__pyroBind()
376
            pm_server.isConnected()
```

```
378         serverConnected = True
379     except Pyro4.errors.CommunicationError:
380
381         serverConnected = False
382
383     # update
384
385     date = '<html><head></head><body><p>' + time.strftime('%H:%M') \
386           + '<br/><span style=" font-size:8pt;">' \
387           + time.strftime('%A %d/%m/%Y') + '</span></p></body></html>'
388
389     self.lbl_time_local.setText(date)
390
391     if serverConnected:
392         try:
393             state = pm_server.getState()
394         except Pyro4.errors.ConnectionClosedError:
395             print 'error'
396             serverConnected = False
397
398     # hour
399
400     date = '<html><head></head><body><p>' + state[4][0] \
401           + '<br/><span style=" font-size:8pt;">' + state[4][1] \
402           + '</span></p></body></html>'
403     self.lbl_time_server.setText(date)
404
405     # mode
406
407     if state[4][2] != server_mode:
408         server_mode = state[4][2]
409         if state[4][2] == 0:
410             self.lbl_state_sleeping.setStyleSheet('background-
411                                                    color: rgb(138, 226, 52)')
412             self.lbl_state_auto.setStyleSheet('background-color
413                                                : rgb()')
414             self.lbl_state_manual.setStyleSheet('background-
415                                                  color: rgb()')
416         elif state[4][2] == 1:
417             self.lbl_state_sleeping.setStyleSheet('background-
418                                                    color: rgb()')
419             self.lbl_state_auto.setStyleSheet('background-color
420                                                : rgb(138, 226, 52)')
421             self.lbl_state_manual.setStyleSheet('background-
422                                                  color: rgb()')
```

```

422         )
423     else:
424         self.lbl_state_sleeping.setStyleSheet( 'background-
425             color: rgb()' )
426         self.lbl_state_auto.setStyleSheet( 'background-color
427             : rgb()' )
428         self.lbl_state_manual.setStyleSheet( 'background-
429             color: rgb(138, 226, 52)' )
430
431     if state[4][2] == 2:
432         self.setManualGui( 'ENABLED' )
433     else:
434         self.setManualGui( 'DISABLED' )
435
436     # pump
437
438     if state[0] == 0:
439         self.lbl_pump_state.setText( 'OFF' )
440         self.lbl_pump_state.setStyleSheet( 'color: red' )
441         self.btn_pump.setText( 'Turn ON' )
442     else:
443         self.lbl_pump_state.setText( 'ON' )
444         self.lbl_pump_state.setStyleSheet( 'color: green' )
445         self.btn_pump.setText( 'Turn OFF' )
446
447     # motor
448
449     if state[1] == 0:
450         self.lbl_motor_state.setText( 'DISABLED' )
451         self.lbl_motor_state.setStyleSheet( 'color: red' )
452         self.btn_motor_enable.setText( 'Enable' )
453     else:
454         self.lbl_motor_state.setText( 'ENABLED' )
455         self.lbl_motor_state.setStyleSheet( 'color: green' )
456         self.btn_motor_enable.setText( 'Disable' )
457
458     # led red
459
460     if state[2][0] == 0:
461         self.lbl_led_red_state.setText( 'OFF' )
462         self.lbl_led_red_state.setStyleSheet( 'color: red' )
463         self.btn_led_red.setText( 'Turn ON' )
464     else:
465         self.lbl_led_red_state.setText( 'ON' )
466         self.lbl_led_red_state.setStyleSheet( 'color: green' )
467         self.btn_led_red.setText( 'Turn OFF' )
```

```
468         # led green
470     if state[2][1] == 0:
472         self.lbl_led_green_state.setText('OFF')
474         self.lbl_led_green_state.setStyleSheet('color: red')
476         self.btn_led_green.setText('Turn ON')
478     else:
480         self.lbl_led_green_state.setText('ON')
482         self.lbl_led_green_state.setStyleSheet('color: green')
484         self.btn_led_green.setText('Turn OFF')
486
488     # led blue
490
492     if state[2][2] == 0:
494         self.lbl_led_blue_state.setText('OFF')
496         self.lbl_led_blue_state.setStyleSheet('color: red')
498         self.btn_led_blue.setText('Turn ON')
500     else:
502         self.lbl_led_blue_state.setText('ON')
504         self.lbl_led_blue_state.setStyleSheet('color: green')
506         self.btn_led_blue.setText('Turn OFF')
508
510     # led ir
512
514     if state[2][3] == 0:
516         self.lbl_led_ir_state.setText('OFF')
518         self.lbl_led_ir_state.setStyleSheet('color: red')
520         self.btn_led_ir.setText('Turn ON')
522     else:
524         self.lbl_led_ir_state.setText('ON')
526         self.lbl_led_ir_state.setStyleSheet('color: green')
528         self.btn_led_ir.setText('Turn OFF')
530
532     # led uv
534
536     if state[2][4] == 0:
538         self.lbl_led_uv_state.setText('OFF')
540         self.lbl_led_uv_state.setStyleSheet('color: red')
542         self.btn_led_uv.setText('Turn ON')
544     else:
546         self.lbl_led_uv_state.setText('ON')
548         self.lbl_led_uv_state.setStyleSheet('color: green')
550         self.btn_led_uv.setText('Turn OFF')
552
554     # preview
556
558     if previewEnabled:
560         self.btn_preview.setText('Disable preview')
562     else:
564         self.btn_preview.setText('Enable preview')
```

```
518         # data list
520
521         if firstDataUpdate:
522             self.updateData()
523             firstDataUpdate = False
524
525         t = threading.Timer(1, self.setStateGui)
526         t.start()
527
528     def updateData(self):
529         global pm_server
530         global state
531         state = pm_server.getState()
532         data_list = state[3]
533         self.list_history.clear()
534         for data in data_list:
535             self.list_history.addItem(self.processName(data))
536
537     def processName(self, data):
538         return data[6:8] + '/' + data[4:6] + '/' + data[0:4] + ' - ' \
539             + data[9:11] + ':' + data[11:13]
540
541     def setManualGui(self, state):
542         if state == 'ENABLED':
543
544             # leds
545
546             self.btn_pump.setEnabled(True)
547             self.btn_motor_enable.setEnabled(True)
548             self.btn_led_red.setEnabled(True)
549             self.btn_led_green.setEnabled(True)
550             self.btn_led_blue.setEnabled(True)
551             self.btn_led_ir.setEnabled(True)
552             self.btn_led_uv.setEnabled(True)
553             self.btn_led_alloff.setEnabled(True)
554             self.btn_led_allon.setEnabled(True)
555
556             # motor
557
558             self.btn_motor_enable.setEnabled(True)
559             self.btn_motor_cw.setEnabled(True)
560             self.btn_motor_ccw.setEnabled(True)
561             self.btn_motor_go.setEnabled(True)
562             self.cmb_delay.setEnabled(True)
563             self.spin_steps.setEnabled(True)
564             self.cmb_direction.setEnabled(True)
565
566             # pump
```



```
568         self.btn_pump.setEnabled(True)
570         # camera
572         self.btn_preview.setEnabled(True)
573         self.btn_photo.setEnabled(True)
574     else:
576         # leds
578         self.btn_pump.setEnabled(False)
579         self.btn_motor_enable.setEnabled(False)
580         self.btn_led_red.setEnabled(False)
581         self.btn_led_green.setEnabled(False)
582         self.btn_led_blue.setEnabled(False)
583         self.btn_led_ir.setEnabled(False)
584         self.btn_led_uv.setEnabled(False)
585         self.btn_led_alloff.setEnabled(False)
586         self.btn_led_allon.setEnabled(False)
588         # motor
590         self.btn_motor_enable.setEnabled(False)
591         self.btn_motor_cw.setEnabled(False)
592         self.btn_motor_ccw.setEnabled(False)
593         self.btn_motor_go.setEnabled(False)
594         self.cmb_delay.setEnabled(False)
595         self.spin_steps.setEnabled(False)
596         self.cmb_direction.setEnabled(False)
598         # pump
600         self.btn_pump.setEnabled(False)
602         # camera
604         self.btn_preview.setEnabled(False)
605         self.btn_photo.setEnabled(False)
606
608 if __name__ == '__main__':
609     app = QtWidgets.QApplication(sys.argv)
610     mainwindow = QtWidgets.QMainWindow()
611     prog = MyFirstGuiProgram(mainwindow)
612     mainwindow.show()
613     sys.exit(app.exec_())
```

Appendix C

Software: server application

C.1 Main.py

```
1 #!/usr/bin/python3

3 import Pyro4
  import sys
5 import os
  import cv2
7 import time
  import cPickle
9 import numpy as np
  from analyzer import *
11 from pump import *
  from leds import *
13 from motor import *
  from camera import *
15 from utils import *
  import RPi.GPIO as GPIO
17 GPIO.setmode(GPIO.BCM)
  GPIO.setwarnings(False)
19
  GPIO.setup(14, GPIO.OUT)
21

23 @Pyro4.expose
  class PMmonitor(object):
25
      def setCommand(self, obj, param1=-1, param2=-1, param3=-1, param4
        =-1):
27          if obj == 'D':
              print dataLst()
29          if obj == 'C':
              if param1 == "PREVIEW":
```

```
31         cameraPreview(param2)

33     if obj == 'P':
34         if param1 == "SET":
35             pumpSet(param2)
36         if param1 == "TIME":
37             pumpTime(param2)
38         if param1 == "TOGGLE":
39             pumpToogle()
40     if obj == 'L':
41         if param1 == "SET":
42             ledSet(param2, param3)
43         if param1 == "TOGGLE":
44             ledToogle(param2)
45
46     if obj == 'M':
47         if param1 == "STEP":
48             motorStep(param2, param3, param4)
49         if param1 == "TOGGLE":
50             motorToogle()
51         if param1 == "SET":
52             motorSet(param2)
53     if obj == 'F':
54         if param1 == "MODE":
55             if param2 == "SLEEP":
56                 setServerMode(0)
57             if param2 == "AUTO":
58                 setServerMode(1)
59             if param2 == "MANUAL":
60                 setServerMode(2)
61
62     def getState(self):
63         state = []
64         state.append(pumpState())
65         state.append(motorState())
66         state.append(ledState())
67         state.append(dataLst())
68         state.append(serverInfo())
69         return state

71     def isConnected(self):
72         return 1
73 daemon = Pyro4.Daemon("192.168.0.7")
74 ns = Pyro4.locateNS()
75 uri = daemon.register(PMmonitor)
76 ns.register("example.greeting", uri)
77 print("Ready.")
daemon.requestLoop()
```

C.2 Analyzer.py

```
#!/usr/bin/python3
2

4 import sys
  import os
6 import cv2
  import time
8 import cPickle
  import numpy as np
10

12 flag = False
  thrT = 2.0
14 kpnt = None

16 def Analyzer(image):
18     input_image = cv2.imread(image)
      b, g, r = cv2.split(input_image)
20     cv2.imwrite("red.bmp", r)
      cv2.imwrite("green.bmp", g)
22     cv2.imwrite("blue.bmp", b)
      imo = b
24     blur = 3
      imf = cv2.GaussianBlur(imo, (blur, blur), 0)
26     ker0 = -1.0
      ker1 = 2.0
28     ker2 = 4.0
      ker = np.array([[ker0, ker0, ker0, ker0, ker0], [ker0, ker1, ker1,
30                      ker1, ker0], [ker0, ker1, ker2, ker1, ker0], [ker0, ker1, ker1,
                          ker1, ker0], [ker0, ker0, ker0, ker0, ker0]]) /
                          ker2
      imt = cv2.filter2D(imf, -1, ker)
32     maxT = 106
      area = 16
34     params = cv2.SimpleBlobDetector_Params()
      params.minThreshold = 0
36     params.maxThreshold = maxT
      params.filterByArea = True
38     params.minArea = area
      params.filterByCircularity = False
40     params.minCircularity = 0.15
      params.filterByConvexity = False
42     params.minConvexity = 0.88
      params.filterByInertia = False
44     params.minInertiaRatio = 0.01
```

```
det = cv2.SimpleBlobDetector(params)
46 keypoints = det.detect(imf)
out = cv2.drawKeypoints(
48     imf,
    keypoints,
50     np.array([]),
    (255,
52     100,
    255),
54     cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
cv2.imwrite('output.bmp', out)
56 to = time.time() - ti
print "Found {} blobs in {} s".format(len(keypoints), to)
58 ts = thrT - to
if (ts > 0):
60     time.sleep(thrT)

62 print "Creating the blob hystogram..."

64 index = []
for point in keypoints:
66     temp = (point.pt, point.size, point.angle, point.response,
    point.octave, point.class_id)
    index.append(temp)

68
keypoint_file = open('keypoints.txt', 'w')
70 keypoint_file.write(cPickle.dumps(index))
keypoint_file.close()
72

74 def dataLst():
    directory_list = []
76     for root, dirs, files in os.walk("Data", topdown=False):
        for name in dirs:
78             directory_list.append(os.path.join(name))
    directory_list.sort()
80     return directory_list
print hello
```

C.3 Camera.py

```
1 import os
import sys
3 import subprocess

5 pid = 0

7
def cameraPreview(state):
```

```
9     global pid
10    if state:
11        pid = subprocess.Popen(
12            "raspivid -o - -t 0 -hf -w 1024 -h 768 -fps 50|cvlc -vvv
13            stream:///dev/stdin --sout '#standard{access=http,mux=
14            ts,dst=:8090}' :demux=h264",
15            shell=True)
16    else:
17        pid.kill()
```

C.4 Motor.py

```
1 import RPi.GPIO as GPIO
2 import time
3 from OpenSSL.rand import status

5 GPIO.setmode(GPIO.BCM)
6 GPIO.setwarnings(False)
7 coil_A_1_pin = 5
8 coil_A_2_pin = 6
9 coil_B_1_pin = 13
10 coil_B_2_pin = 19
11 enable = 26

13 StepCount = 8
14 Seq = range(0, StepCount)
15 Seq[0] = [0, 1, 0, 0]
16 Seq[1] = [0, 1, 0, 1]
17 Seq[2] = [0, 0, 0, 1]
18 Seq[3] = [1, 0, 0, 1]
19 Seq[4] = [1, 0, 0, 0]
20 Seq[5] = [1, 0, 1, 0]
21 Seq[6] = [0, 0, 1, 0]
22 Seq[7] = [0, 1, 1, 0]

25 GPIO.setup(coil_A_1_pin, GPIO.OUT)
26 GPIO.setup(coil_A_2_pin, GPIO.OUT)
27 GPIO.setup(coil_B_1_pin, GPIO.OUT)
28 GPIO.setup(coil_B_2_pin, GPIO.OUT)
29 GPIO.setup(enable, GPIO.OUT)

31

33 def motorSet(status):
34     GPIO.output(enable, status)

35

37 def motorToggle():
```

```
state = not GPIO.input(enable)
39 GPIO.output(enable, state)

41
def setStep(w1, w2, w3, w4):
43     GPIO.output(coil_A_1_pin, w1)
        GPIO.output(coil_A_2_pin, w2)
45     GPIO.output(coil_B_1_pin, w3)
        GPIO.output(coil_B_2_pin, w4)
47

49 def motorStep(delay, steps, direction):
    if direction == "CW":
51         for i in range(steps):
            for j in range(StepCount):
53                 setStep(Seq[j][0], Seq[j][1], Seq[j][2], Seq[j][3])
                    time.sleep(delay / 1000.0)
55     else:
        for i in range(steps):
57             for j in reversed(range(StepCount)):
                setStep(Seq[j][0], Seq[j][1], Seq[j][2], Seq[j][3])
59                 time.sleep(delay / 1000.0)

61
def motorState():
63     return GPIO.input(enable)
```

C.5 Pump.py

```
1 import RPi.GPIO as GPIO
import time
3 import threading
PUMP_pin = 14
5
GPIO.setmode(GPIO.BCM)
7 GPIO.setwarnings(False)

9 GPIO.setup(PUMP_pin, GPIO.OUT)

11
def pumpState():
13     return GPIO.input(PUMP_pin)

15
def pumpSet(state):
17     GPIO.output(PUMP_pin, state)

19
def pumpTime(seconds):
```

```
21     pumpSet(1)
    t = threading.Timer(seconds, pumpOff)
23     t.start()  # after 30 seconds, "hello, world" will be printed

25
def pumpOff():
27     pumpSet(0)

29
def pumpToggle():
31     state = not GPIO.input(PUMP_pin)
    GPIO.output(PUMP_pin, state)
```

C.6 Leds.py

```
import RPi.GPIO as GPIO
2
GPIO.setmode(GPIO.BCM)
4 GPIO.setwarnings(False)

6 RED_gpio = 16
  GREEN_gpio = 20
8 BLUE_gpio = 21
  IR_gpio = 7
10 UV_gpio = 12

12 GPIO.setup(RED_gpio, GPIO.OUT)
  GPIO.setup(GREEN_gpio, GPIO.OUT)
14 GPIO.setup(BLUE_gpio, GPIO.OUT)
  GPIO.setup(IR_gpio, GPIO.OUT)
16 GPIO.setup(UV_gpio, GPIO.OUT)

18
def ledState():
20     led = []
    led.append(GPIO.input(RED_gpio))
22     led.append(GPIO.input(GREEN_gpio))
    led.append(GPIO.input(BLUE_gpio))
24     led.append(GPIO.input(IR_gpio))
    led.append(GPIO.input(UV_gpio))
26     return led

28
def ledSet(color, state):
30     if color == "RED":
        pin = RED_gpio
32     if color == "GREEN":
        pin = GREEN_gpio
34     if color == "BLUE":
```



```
        pin = BLUE_gpio
36     if color == "IR":
        pin = IR_gpio
38     if color == "UV":
        pin = UV_gpio
40     GPIO.output(pin, state)

42
def ledToggle(color):
44     if color == "RED":
        pin = RED_gpio
46     if color == "GREEN":
        pin = GREEN_gpio
48     if color == "BLUE":
        pin = BLUE_gpio
50     if color == "IR":
        pin = IR_gpio
52     if color == "UV":
        pin = UV_gpio
54     state = not GPIO.input(pin)
        GPIO.output(pin, state)
```

C.7 Client.py

```
1 # saved as greeting-client.py
import Pyro4
3 # name = input("What is your name? ").strip()
greeting_maker = Pyro4.Proxy("PYRONAME:example.greeting")
5                                     # use name server object lookup uri
                                     shortcut
print(greeting_maker.get_fortune(0))
```

C.8 Utils.py

```
import time
2 server_mode = 0
  server_state = 0
4

6 def serverInfo():
    global server_mode
8    global server_state
    info = []
10    info.append(time.strftime("%H:%M"))
    info.append(time.strftime("%A %d/%m/%Y"))
12    info.append(server_mode)
    info.append(server_state)
14    return info
```

```
16 def setServerMode(mode):
18     global server_mode
19     server_mode = mode
20
22 def setServerState(state):
23     global server_state
24     server_state = state
25
26 def getServerMode():
27     global server_mode
28     return server_mode
29
30
32 def getServerState():
33     global server_state
34     return server_state
```

Bibliography

- [1] Epa, united states environmental protection agency. Online, accessed on January 2018.
- [2] Clean air for europe program. Online, accessed on March 2018.
- [3] Legambiente. Online, accessed on January 2018.
- [4] P Bonanni, R Daffinà, R Gaddi, A Giovagnoli, V Silli, and M Cirillo. Limpatto dell'inquinamento atmosferico sui beni di interesse storico-artistico esposti all'aperto. *Editoria: Roma, Italy*, 1, 2006.
- [5] Fai instruments. Online, accessed on January 2018.
- [6] G. Buonanno, M. Dell'Isola, L. Stabile, and A. Viola. Critical aspects of the uncertainty budget in the gravimetric pm measurements. *Measurement*, 44(1):139 – 147, 2011.
- [7] Centre for atmospheric science. Online, accessed on January 2018.
- [8] Cpc, definition. Online, accessed on January 2018.
- [9] Robotstore. Online, accessed on March 2018.
- [10] Open cv official reference. Online, accessed on March 2018.
- [11] Rgb color profile. Online, accessed on January 2018.
- [12] Gaussian blurring theory. Online, accessed on March 2018.
- [13] Gaussian blur filter. Online, accessed on March 2018.
- [14] Aerosol definition. Online, accessed on January 2018.
- [15] Richard B Schlesinger. Properties of ambient pm responsible for human health effects: coherence between epidemiology and toxicology. *Inhalation toxicology*, 12(sup1):23–25, 2000.
- [16] Ministerial decree 60/02. Online, accessed on March 2018.
- [17] Liquid particle counting applications in pharmaceutical manufacturing. Online, accessed on January 2018.
- [18] Particle counter. Online, accessed on January 2018.
- [19] Antonio Briganti. *Filtrazione e disinquinamento dell'aria*.
- [20] Jonathan M Samet. What properties of particulate matter are responsible for health effects? *Inhalation Toxicology*, 12(sup1):19–21, 2000.