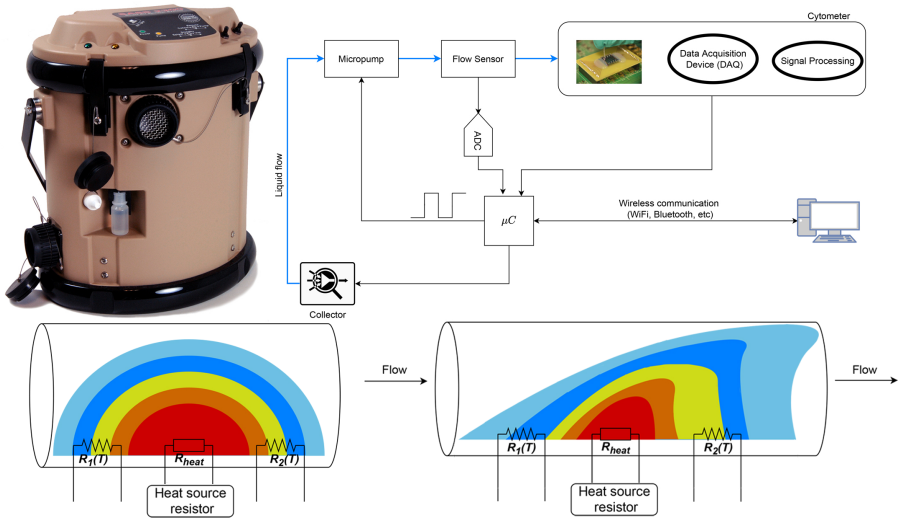
TÉCNICO LISBOA



Flow Control for a Portable Airborne Biocontamination Detection System

Sara Margarida Gonçalves Soares

Thesis to obtain the Master of Science Degree in

Electronics Engineering

Supervisors: Prof. Luís Filipe Soldado Granadeiro Rosado
Co-Supervisor: Cap. Wilson David Talhão Antunes

Examination Committee

Chairperson: Prof. Paulo Ferreira Godinho Flores
Supervisor: Prof. Luís Filipe Soldado Granadeiro Rosado
Member of the Committee: Prof. Gonçalo Nuno Gomes Tavares

October 2021

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa

Declaração

Declaro que o presente documento é um trabalho original da minha autoria e que cumpre todos os requisitos do Código de Conduta e Boas Práticas da Universidade de Lisboa.

Acknowledgments

First of all, I would like to thank my supervisor, Professor Luís Rosado, for accepting to have me as a student and finding the time to put up with me, always supporting me and helping whenever it was needed. Thank you also to the Laboratorio de Bromatologia e Defesa Biológica do Exército and INESC-MN for the equipment provided.

Thank you also to all my colleagues and friends who have helped me while working at Instituto Superior Técnico. Thank you to my family who supported me during these past five years.

I also want to thank the Instituto de Telecomunicações for the conditions and materials provided for the completion of the dissertation.

Abstract

A quick, real-time detection of pathogenic agents is an important aspect of ensuring public health, and a quick response by authorities to concerning dangers. It is also an important aspect for industries that require a clean, sterilized environment. As such, a system capable of sampling, analyzing and producing real-time results is important to help in these sectors.

To be able to perform such tasks, a system using a commercially available air sampler, microfluidics and cytometry is being developed, in order to be able to get results on the field without the need to collect and transport samples to a laboratory for further analysis. The system gathers samples from the air into a liquid, and a micropump controls the flow rate of the stream that enters the previously developed microfluidics chip. There, the samples are mixed with Superparamagnetic Nanoparticles (MNPs) that bind to the targeted pathogenic agents, and then enter the cytometer, which contains Magnetoresistive (MR) sensors. The MR sensors are affected by the magnetic field of the MNPs and detect the targeted agents.

In this thesis, a flow control system was designed and tested. A flow rate sensor was developed in order to monitor the flow rate in the channels, and ensure a stable fixed speed. It explores heat dispersion modifications caused by a flowing stream which allow determining the stream velocity. Pump control via Pulse-Width Modulation (PWM) is achieved by converting that signal to a Direct Current (DC) voltage. Since portability is also a very important aspect of any modern device, wireless control of the micropump is also obtained, with the development of a Graphical User Interface (GUI) for ease of use. Wireless control of a few basic features of the air sampler is also achieved.

Keywords

Flow rate sensor, Closed loop control, Portable Biocontamination Detection, Microfluidics, Cytometry

Resumo

A detecção rápida e em tempo real de agentes patogénicos é um importante aspecto para garantir a saúde pública, com uma rápida resposta por parte das autoridades a perigos deles resultantes. É também um aspecto importante de indústrias que requerem um ambiente limpo e esterilizado. Assim, um sistema capaz de recolher amostras, analisar e produzir resultados em tempo real é uma importante ajuda nestes sectores.

De modo a ser possível realizar tais tarefas, um sistema que usa um colector de amostras aerosolizadas disponível comercialmente, microfluidica e citometria está a ser desenvolvido, de modo a obter resultados no terreno sem a necessidade de recolher e transportar amostras para analisar num laboratório. O sistema usa o colector para recolher amostras do ar para um líquido e uma microbomba controla o fluxo da corrente que entra no *chip* de microfluídica a ser desenvolvido. As amostras são misturadas com partículas superparamagnéticas que se associam aos agentes que se desejam detectar, entrando depois a corrente que saí do *chip* num citometro que usa sensores magneto-resistivos. Os sensores são afectados pelo campo magnético das particulas superparamagnéticas e detectam os agentes patogénicos.

Nesta tese, um sistema de controlo de fluxo foi desenhado e testado. Foi desenvolvido um sensor de fluxo de modo a monitorizar a velocidade da corrente nos canais e garantir uma velocidade estável e fixa. O sensor desenvolvido explora modificações na dispersão de temperatura causadas pelo fluído circulante que possibilitam a determinação da velocidade do fluxo. O controlo da microbomba é realizado através da conversão de um sinal Pulse-Width Modulation (PWM) em tensão Direct Current (DC). Sendo a portabilidade um factor importante de qualquer aparelho moderno, também é obtido controlo sem fios da microbomba, com o desenvolvimento de um Graphical User Interface (GUI) para facilitar o uso do sistema. Controlo sem fios de algumas componentes do colector também é obtido.

Palavras Chave

Sensor de fluxo, Controlo em malha fechada, Detecção portátil de biocontaminação, Microfluídica, Citometria

Contents

List of Figures

List of Tables

Listings

Acronyms

ADC	Analog to Digital Converter
AMR	Anisotropic Magnetoresistive
CCA	Constant Current Anemometer
CMRR	Common-Mode Rejection Ratio
CTA	Constant Temperature Anemometer
DAC	Digital to Analog Converter
DC	Direct Current
EMI	Electromagnetic Interference
GUI	Graphical User Interface
INESC-MN	Instituto de Engenharia de Sistemas e Computadores – Microsistemas e Nanotecnologias
MNPs	Superparamagnetic Nanoparticles
MP	Magnetic Particles
MR	Magnetoresistive
MRR	Magnetoresistance Ratio
NTC	Negative Temperature Coefficient
PCB	Printed Circuit Board
PID	Proportional Integral Derivative
PWM	Pulse-Width Modulation
RMS	Root Mean Square
SPI	Serial Peripheral Interface
SV	Spin-Valve
UART	Universal Asynchronous Receiver/Transmitter

1

Introduction

Contents

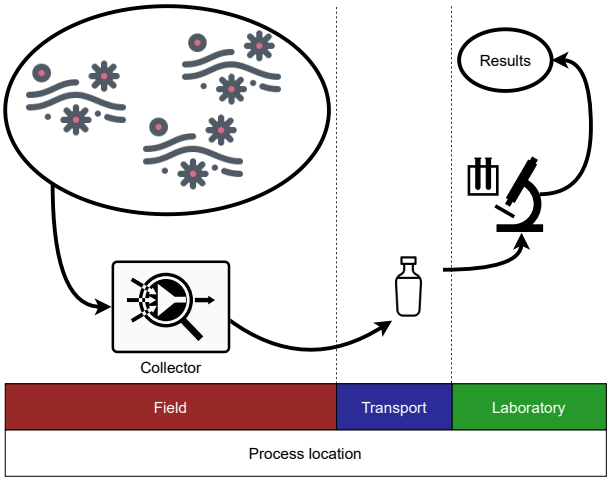
1.1 Motivation

The detection of pathogenic agents is an important tool to ensure public health, helping the authorities to answer quicker and more effectively, while also avoiding mass panicking. In addition, the detection of other type of agents is also very important, specially in industries where a clean environment is not only needed, but required, like in the medical and food industries. Since air spreading is the most effective, and maybe most undetectable, pathway for biological agents' dissemination, its detection is of paramount importance, not only to prevent its spreading, like in the case of easily transmitted airborne diseases, but to prevent damage done by any possible attacks using air spreading, dangerous bio-agents that may happen in hostile environments.

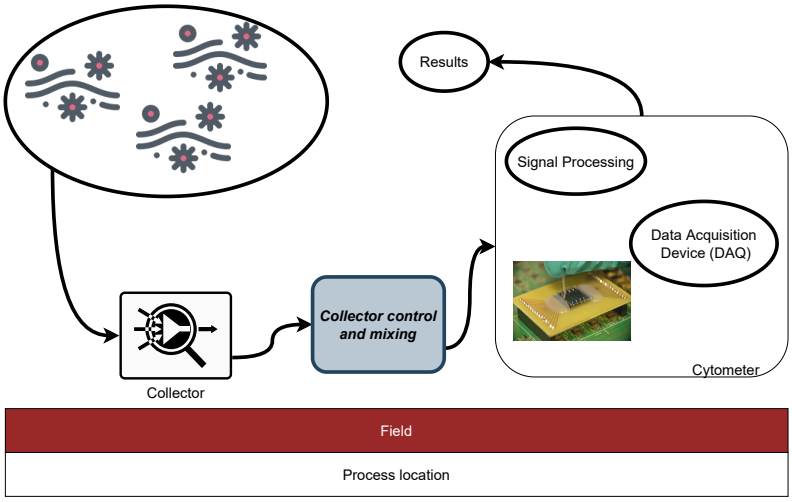
If a device that could give real time results regarding the presence, or not, of pathogenic or non-pathogenic agents existed, industries where sterilization is a must would greatly benefit from it. With a device installed in a space that must be kept relatively sterilized at all times, for example a surgery room or a food factory, any introduction of an agent that could be harmful would be detected in a relatively quick and non-laborious way, ensuring a quick response from the responsible entities. The same could be said about public health issues. To avoid the dissemination of contagious diseases that could be spread by air, a device that could quickly detect such disease would greatly help in the containment of mass spreading, using the device to assure that closed spaces are safe and that the safe spaces would not be breached.

As of now, any commercial solutions for the detection of aerosolized biological agents are separated into the use of a collector unit, which gathers samples in whichever environment is needed, but is not completely portable and wireless, and a laboratory analysis of the collected samples. Since this is a process that can be lengthy and cumbersome, the goal of this thesis is to develop enabling technologies to upgrade a commercial aerosol collector into a single device capable of both collecting and analyzing samples, providing a quick and real time detection of any existing harmful bio-agents in the environment being tested.

Fig. 1.1(a) describes the current process for the detection of aerosolized biological agents. A collector gathers samples from the air and stores them in a vial (or other containers) for later examination at a laboratory, which then gives the results. Fig. 1.1(b) describes the target process. The same collector gathers samples from the air, however, instead of storing them in a container, the samples pass through a microfluidics chip where they are mixed with a liquid containing Superparamagnetic Nanoparticles (MNPs) that bind to the biological targets. Following the mixing of the fluids, the mixture flows through a cytometer that uses Magnetoresistive (MR) sensors and detects the MNPs, and since those are bound to the biological targets, it consequently detects them. The results are then processed for later visualization by the users. The objective is for most, if not all, of this process to be done by a single device.



(a) Current detection process



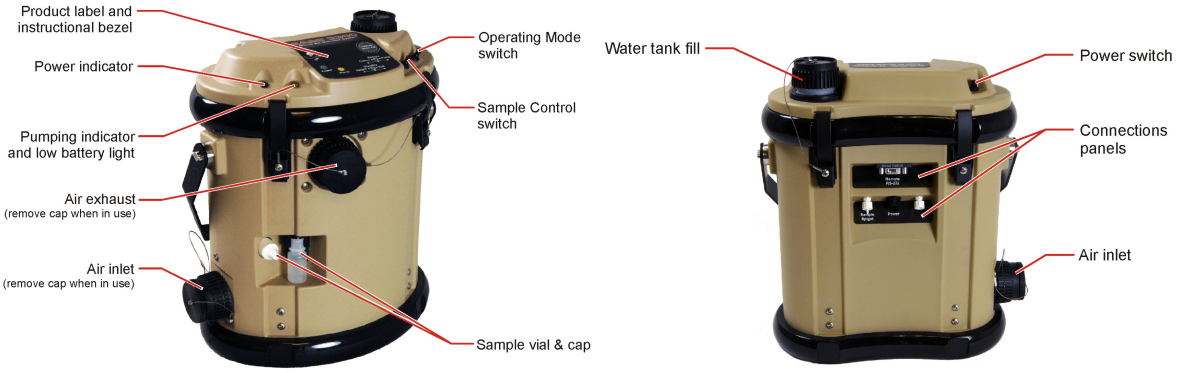
(b) Target process with in-situ detection and a portable device

Figure 1.1: Process of collecting and detecting aerosolized biological agents.

1.2 Goals

The main goal of this thesis is to create a system capable of extracting and mixing the fluids gathered in a commercial aerosol collector provided by the Laboratório de Bromatologia e Defesa Biológica do Exército and prepare them to be processed by the cytometer being developed by Instituto de Engenharia de Sistemas e Computadores – Microsistemas e Nanotecnologias (INESC-MN)'s Spintronics and Magnetic Biosensors group. The collector used is a SASS 2300 wetted-wall air sampler (Fig. 1.2). It is a portable multi-stage wetted-wall cyclone sampler that is able to extract particulates and water-soluble

chemical vapors from air and transfer them to a liquid for later examination. It has a 325 liter per minute air collection rate and a 0.5-10 μm particle size range. The obtained liquid can be either stored in a vial or it can be transferred through a tube to another suitable container. The air sampler can be controlled by a Windows computer with a proprietary software, being connected to the computer through a RS-232 comm port. The cytometer being developed uses MR sensors, more specifically, Spin-Valve (SV) sensors, and it will have an input flow rate of about 1-20 μL/min. There are about 10 channels in total on the cytometer, which means the pump needs to be able to output a flow of about 10-200 μL/min. The sensor also needs to be able to read values inside that range, so proper control can be executed.



(a) Front view of SASS 2300 **(b)** Back view of SASS 2300

Figure 1.2: SASS 2300 wetted-wall air sampler. Taken from [1].

Fig. 1.3 shows a screenshot of the the proprietary software that controls the air sampler when in automatic mode. It is able to customize the collection protocol, from the state of the fan when the pump in on, to the run time of each component and volume of air sampled. It is also possible to control the cleaning routine of the air sampler.

Besides these controls, the operation manual of the air sampler also provides ASCII commands that can be sent over the serial data link to control or interrogate the sampler [1], allowing the use of, for example, a microcontroller to control the air sampler, without the need of the proprietary software.

The key objectives of this project are:

- Develop a wireless control system for the collector - Since the objective is to have a portable device, it is also important that it is able to receive commands and transmit the gathered information about the environment wirelessly, to be able to avoid putting in danger the users by having them enter a possibly contaminated environment. Alternatively, if it is able to transmit the gathered information wirelessly, it can also issue warnings if a room that is under constant testing gets contaminated, for example a breach in a room that needs constant sterilization;

- Develop a pump-control system with feedback capable of regulating the flow rate - To control the

flow of the fluids that enter the microfluidics chip, a capable pump and precise control is very important. To avoid possible damages to the microfluidics chip or even the pump, having information regarding the flow in the chip (any blockages that may result, either total blockages or partial ones) is necessary, hence the need for a system with feedback from a flow rate sensor. If a total blockage occurs, it's important to stop the flow to avoid breaking any materials, and if a partial one occurs, increasing the flow rate, and therefore the pressure, to try and clear the channel is a possible solution;

- Design a flow rate sensor that is able to measure accurately enough very small values (about 10-200 μL/min), and even smaller variations, with very little error. A sensor based on the heat dispersion modification cause by a flowing fluid and its effect on the value of thermal dependent resistors is considered for this application.

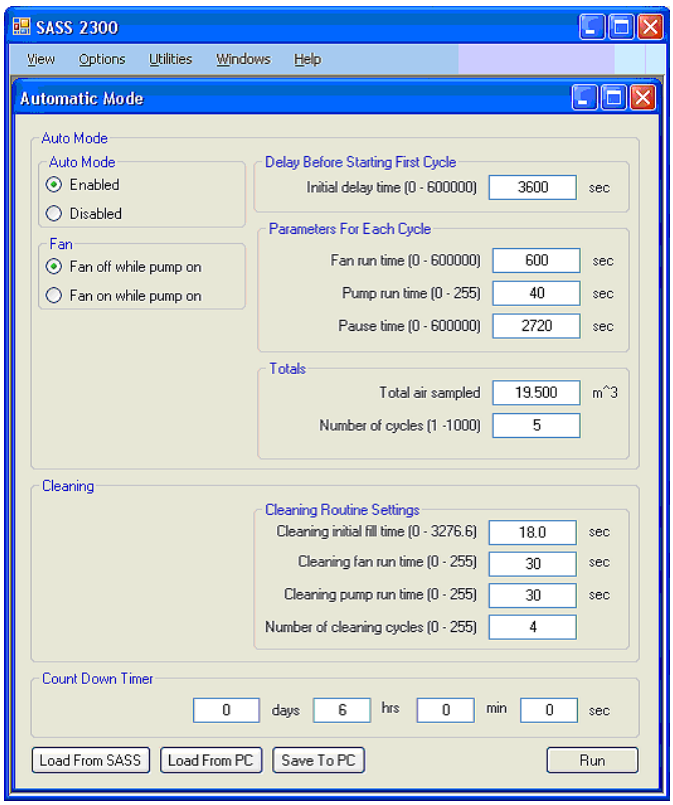


Figure 1.3: Screenshot of the software and the available controls. Taken from [1].

The main difficulty of this project was the flow rate sensing and regulating of the pump to act accordingly, since, in these cases, the flow rate is of very small values, about 10-200 μL/min. As such, it is of the utmost importance that the sensor has, and maintains, a high accuracy. Besides that, there is also a need to adapt the system to the specifications of a commercial collector and the specifications of a cytometer that is in development.

1.3 Document organization

This thesis is organized as follows: Chapter 1 gives a brief introduction about the problem that motivated this thesis and the goals expected to achieve by the end of this project.

In Chapter 2, the important concepts for the understanding of this project as a whole are given, as well as a brief overview and explanation of the existent technology involved in the whole project and its shortcomings and advantages. A more detailed discussion on flow sensors is presented, from a brief explanation of different types of flow sensors and the choice to implement a temperature based type, to a explanation of three different methods of obtaining the flow rate from temperature based sensors.

Chapter 3 presents the developments made in this project. It describes and explains the work done developing the sensor, including its modelling, designing and prototyping. It also describes and explains how the pump is controlled and the circuit used to achieve the control. Besides the development of the hardware components, it also describes the development and operation of the software components, such as the software control of the pump, the wireless communication, and the Graphical User Interface (GUI) developed.

Chapter 4 is the chapter where the experimental results are documented. It describes the experiments done in order to obtain the results that show the operation of the multiple components developed in this project, as well as the complete developed system working together. It describes how the sensor was characterized in order to obtain its range of functioning, how the Pulse-Width Modulation (PWM) control performs, and how much noise is in the circuit.

The final chapter, Chapter 5, regards the conclusions about the work done in this thesis, as well as future work that can be developed in order to upgrade this project to a higher level and optimize it.

2

Chapter 2 - State of the art

Contents

2.1 Air sampling - A brief overview

As explained in the previous chapter, the available air sampler uses a wetted-wall cyclone separation principle to perform sampling of the air. However, this isn't the only principle to perform such task. There are various other methods, such as filtering using a gelatin or electret filter, liquid impingment, electrostatic precipitation or impaction [5].

A cyclone separator works by having air entering the tangential inlet at the top at high speeds, then, when inside the cyclonic cup, it starts to flow in a circular (due to the tangential entry) downwards motion, creating a vortex. As the diameter of the cone starts decreasing, the speed of the air flow starts to increase and the bigger particles, which have a higher inertia, are separated and impact the side of the separator and leave the cyclone separator by the bottom, while the smaller, low inertia particles remain in the flow, traveling upwards in the inner vortex that was created by the outer vortex due to its high velocity, and leaving the separator by the top. Fig. 2.1 shows a diagram of a cyclone separator and the created vortexes.

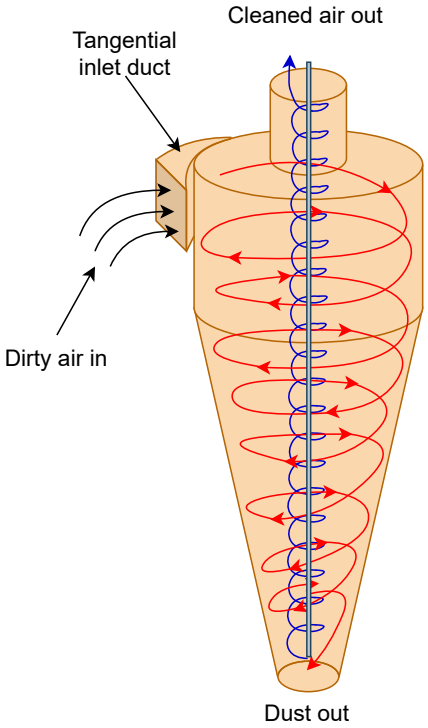


Figure 2.1: Diagram of a cyclone separator and its internal airflow. Inner vortex is represented in blue while the outer vortex is represented in red.

The schematic of the SASS 2300 air sampler's cyclonic cup and adjacent system can be observed on Fig. 2.2. The cleaned air is mixed with a liquid, typically distilled water, after leaving the cyclonic cup, for further storage, in either a vial or another suitable container, as described in chapter 1.

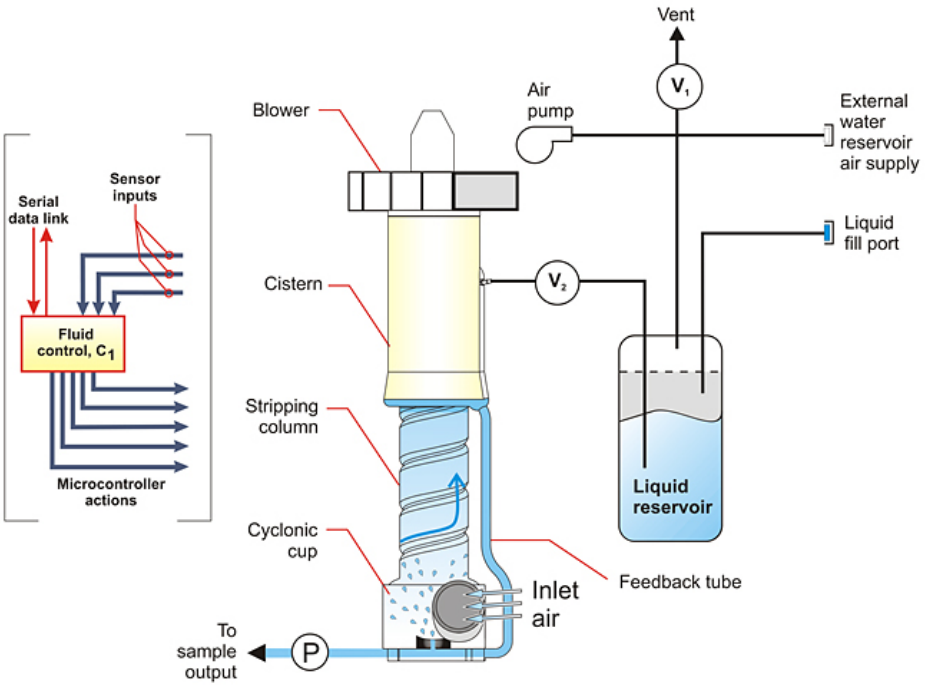


Figure 2.2: Schematic of the water flow and cyclonic cup of the SASS 2300. Taken from [1].

From [6], and from Fig. 2.2, it can be seen that the way the SASS 2300 air sampler works is not like a conventional cyclone separator. While it has a cyclone cup integrated in the whole system, the cyclone cup is upside down, and the air inlet is on the bottom of the device. The device has a fan at the top which pulls the air through the inlet, and towards the cyclonic cup. It also has a demister, that is able to separate the water for reuse and dispose of the undesired particles. It uses water (or other suitable liquid) to aid in the separation of the particles from the air that comes through the inlet, as well as to store the resulting sample. It is able to control the concentration of the samples by controlling the volume of water used and/or the volume of air sampled.

2.2 Biocontamination detection - Cytometer

Cytometry is the technique that allows for the measurement of the characteristics of microscopic particles, like for example, cells. Flow cytometry is the technique that allows the measurement of microscopic particles suspended in a flowing fluid. The analysis of the particles can be performed by using various different methods, however, in this case, the cytometer being developed by INESC-MN's Spintronics and Magnetic Biosensors group employs MR cytometry. There are various types of MR sensors, this technology being used thanks to its versatility and the fact they can be very compact. Some of the types of sensors using the MR effect are the Anisotropic Magnetoresistive (AMR), the SV, and the Magnetic Tunnel Junctions sensors, the latter two having a higher performance than the first one [7].

The fluid to test is mixed with Magnetic Particles (MP) in a microfluidics chip. Those MP are labeled with specific antibodies that bind with the targeted particles, and when they pass through a MR sensor, it is possible to sense the presence of the targeted particles and what their concentration is. Fig. 2.3 describes an illustration of the process. The test fluid with all the particles collected passes above a MR sensor, and the targeted particles, bound with the MP, are detected by the sensor due to the magnetic field.

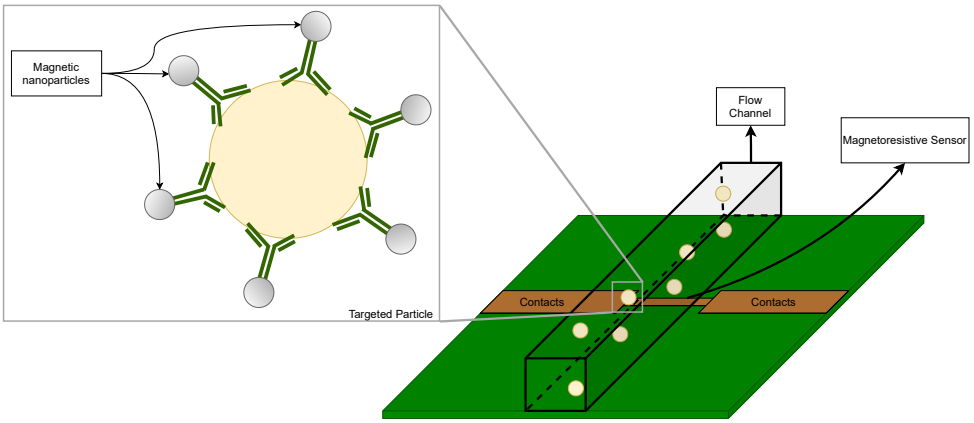


Figure 2.3: Diagram of particles passing by a MR sensor. Adapted from [2].

A microfluidic lab on chip is a system with the capability to perform multiple tasks required of any analytical laboratory, such as mixing, extraction, separation, and detection. These systems offer performance advantages, such as higher sample processing speed and throughput, over large-scale laboratory instrumentation [8], these advantages coming from the increased surface to volume ratio, due to the diminutive scale of flow channels in the systems [9]. Typically, these chips have channels of sub-millimiter dimension. At such reduced dimensions, these chips allow not only for a high mobility, but also for the use of small volume of samples.

While INESC-MN has the ability to develop both types of MR sensors, the cytometer being developed uses SV sensors. A SV is a device which electrical resistance changes depending on the relative magnetization of the layers. In Fig. 2.4, two states of a spin-valve device can be seen, the high resistance and the low resistance state. The state of the device depends if it is subjected to an external magnetic field or not. If a magnetized particle passes through the spin-valve device, its state changes and the particles are detected.

These sensors are linear magnetic field transducers based on the giant magnetoresistance effect, and by patterning the SV with a large aspect ratio, with the long direction perpendicular to the pinned layer magnetization easy axis, the demagnetizing field favour the alignment of the free layer magnetization at 90° with respect to the pinned layer easy axis [3]. If there are small variations of the external magnetic field, the angle between the pinned and free layers will change, consequently changing the

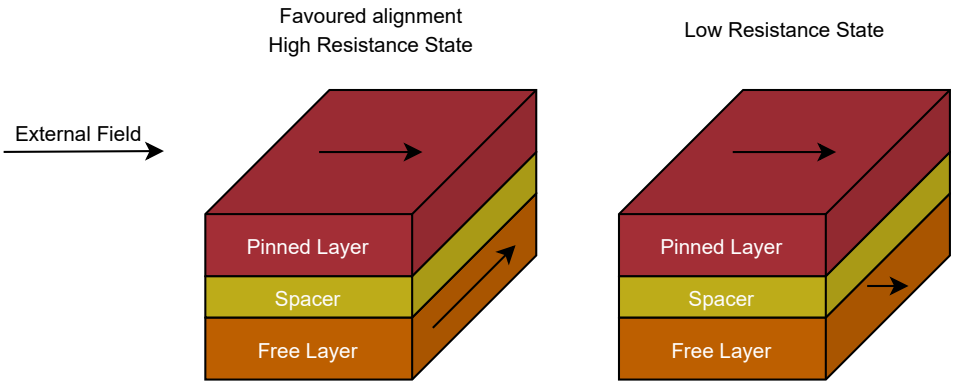


Figure 2.4: SV device resistance states.

sensor resistance. With this configuration, the SV resistance changes linearly with respect to the external field [3]. To facilitate the calculations, the dynamic response of the sensor is calculated with a few assumptions in mind: the targeted particles are spherical; the magnetic moment of a particle is the combinations of the magnetic moment of all the particles; and the magnetic nanoparticles are assumed to be uniformly distributed over the targeted particle's spherical surface. The presence of the magnetic fields of the MP is detected by a change in the sensor resistance when biased with a fixed sense current. The response of the SV sensor is calculated with

$$\Delta V = -\left(\frac{R_{AP} - R_P}{R_P}\right) \times R_{sq} \times \frac{L}{W} \times I \times \frac{B_y}{2B_k^{eff}}, \tag{2.1}$$

where $((R_{AP} - R_P)/R_P)$ is the sensor's Magnetoresistance Ratio (MRR), R_{AP} being the resistance at the antiparallel state and R_P the resistance at the parallel state, R_{sq} is the square resistance in the parallel state, L/W is the aspect ratio of the sensor, I the sensing current, B_y the component of the average fringe field and B_k^{eff} the effective free layer anisotropy field. Fig. 2.5 shows a diagram of a SV sensor in a chip (not to scale), with a particle moving with speed and direction $\vec{v}$.

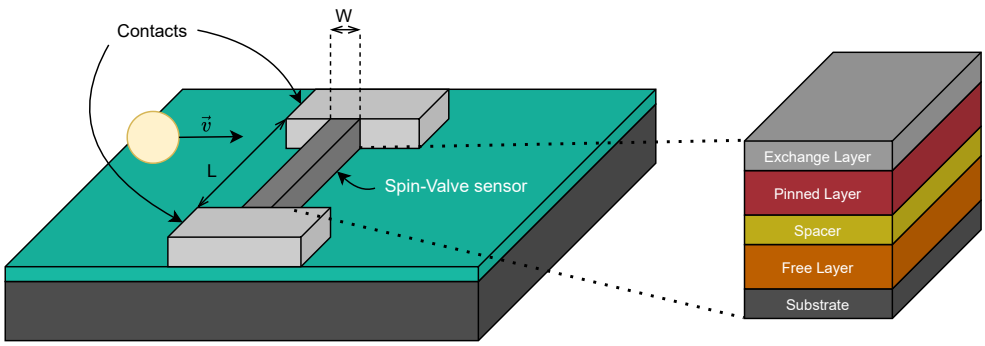


Figure 2.5: Chip with a SV sensor. Adapted from [2].

Fig. 2.6 shows the transfer curve of a SV sensor using (2.1), where the sensing current is 1 mA, L = 10 μm, and W = 3 μm. This figure shows that between around ±2.5 mT, the sensor has a approximately linear operation, while below and above that threshold, the results start to vary differently. Therefore, those can be seen as the saturation zones. Since the resistance and voltage are proportional, the R/B transfer curve is similar.

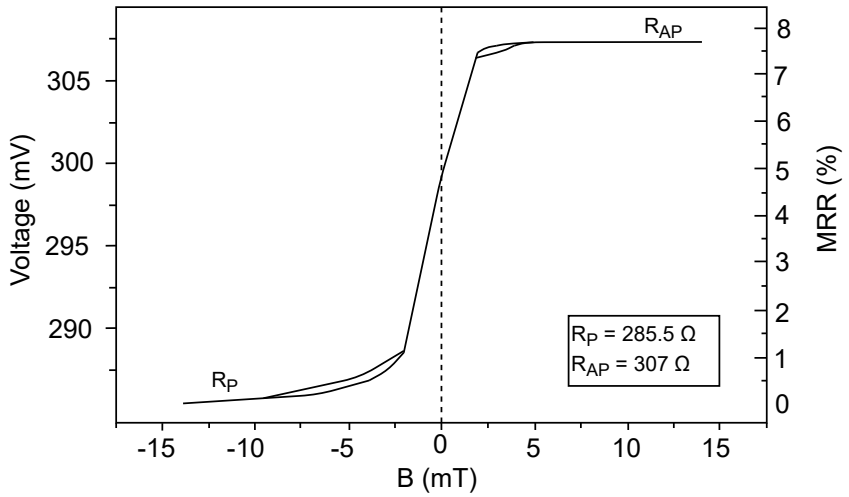


Figure 2.6: Transfer curve of a SV sensor, with MR = 7.69% and a sensitivity of 4.8 V/T for a 1 mA sensing current. Adapted from [3].

The cytometer being developed by the INESC-MN group has around 10 channels in parallel, with an input flow rate in the ranges of 1-20 μL/min. The channel's size varies depending on the type of particle to analyze. For example, for bacterias, the channel size is of about 10 μmX100 μm, while for cells it is 50 μmX100 μm, and 1 μX100 μm for anthrax. The cytometer's sensor has a length of about 2 μm (in the flow's direction), a sampling speed of 200 KS/s, and a bandwidth of 50 KHz. The necessary bandwidth can be approximated by

$$\begin{cases} MaxSpeed = \left(\frac{\frac{FlowRate}{60}}{ChannelArea}\right)\ m/s \\ BW = \left(\frac{MaxSpeed}{SensorLength}\right)\ Hz. \end{cases} . \tag{2.2}$$

Using the worst case, that is, the maximum flow rate (20 μL/min) in the smallest channel (1 μX100 μm), the bandwidth would need to be

$$\begin{cases} MaxSpeed = \left(\frac{\frac{20\ \mu}{60}}{1\ \mu X100\ \mu}\right) = 3.333\ m/s \\ BW = \left(\frac{3.333}{2\ \mu}\right) = 1.6665\ MHz. \end{cases} . \tag{2.3}$$

Using the best case, that is, the minimum flow rate (1 μL/min) in the largest channel (50 μX100 μm), the bandwidth would need to be

$$\begin{cases} MaxSpeed = \left(\frac{\frac{1\ \mu}{60}}{50\ \mu X 100\ \mu}\right) = 3.334\ mm/s \\ BW = \left(\frac{3.334\ m}{2\ \mu}\right) = 1.667\ kHz. \end{cases} . \tag{2.4}$$

As (2.3) shows, in the worst case scenario, the 50 kHz bandwidth of the cytometer is a potential limitation. Particularly when the smallest section channels are used, the flow rate must be constrained to ensure that the sensor signal is properly acquired bearing in mind the cytometer's bandwidth and maximum sampling rate. In the best case scenario (2.4), however, the bandwidth of the cytometer is well adequated. The bandwidth necessary for cytometry is much higher than the one needed for the system developed in this project. Since this project deals with flow changes, the necessary bandwidth will be small, possibly less than 10 Hz, since flow changes cannot occur suddenly and take time, due to many factors, such as component time constants, and the liquid itself.

2.3 Flow rate sensing and control

There are multiple methods to measure the velocity of a fluid, be it of a mechanical or temperature nature, both of these methods usually being indirect, using other factors to later determine the flow rate. The sensors of mechanical nature usually involve moving parts and are of big dimensions, and therefore not indicated to be used in microfluidics. There are deflection based sensors, consisting of structures oriented perpendicular or parallel to the flow direction which deflect when fluid flows through the channel [10]. Since these sensors have total dimensions exceeding 1 mm, their integration in microfluidic channels is not easy or perhaps not even possible. Besides the dimensions, there is also the fact that the sensors that operate based on deflection through lift require operation at Reynolds numbers above those found in microfluidic channels [10]. The Reynolds number is the ratio of inertial forces to viscous forces. This number is what helps predict the flow patterns in different fluid flow situations. Usually, in microfluidics, the Reynolds number is very low, which means that the flow is strictly laminar.

Of the temperature nature, one of the most used devices is the anemometer, more specifically, the hot-wire anemometer. This type of device is usually reserved for measures regarding air flow or gas streams, and as such not used in microfluidics for measures regarding fluids. However, it is a good example of how flow rate is measured based on the temperature of a substance and what happens to the heat dispersion when said substance is flowing, be it a fluid or gas stream. A hot-wire anemometer is typically comprised by two probes with a wire stretched between them. There are two types of anemometers, Constant Temperature Anemometer (CTA) and Constant Current Anemometer (CCA). With CCA, there is a risk of burning the wire, if the cooling effect of the flow is too low. Since the current stays constant, if the flow is too slow, and thus the cooling effect too low, the wire can overheat due to a

high value of dissipated power, since

$$P = I^2 R\ [W]. \tag{2.5}$$

Therefore, a CTA is safer to use, to avoid damaging the probe. For this type of anemometer, a Wheatstone bridge configuration is typically used, where the wire is the resistor under measurement. As shown in Fig. 2.7, there are two variable resistors in series, parallel to two known resistors. The objective is to keep the temperature of the wire constant, that is to keep the circuit in equilibrium, where

$$\frac{R_1}{R_W(T)} = \frac{R_2}{R_3}, \tag{2.6}$$

with $R_W(T)$ being the resistance to measure, which varies depending on the temperature, T. Since CTA maintains the probe's temperature, that is, it's dissipated power, the change in temperature depends on the speed of the flow passing through the probe.

To do this, first the resistor R_3 is used to calibrate the device initially, and put the circuit in equilibrium. Then the device is put in the flow to measure and, when there is a variation of the air flow, the resistor $R_W(T)$ changes, which creates the voltage difference between nodes 1 and 2, and consequently a change in temperature, this change being measured by the amplifier and corrected by the feedback current. When the circuit is again at equilibrium, the changes in current to regain equilibrium can be measured and used to calculate the flow rate.

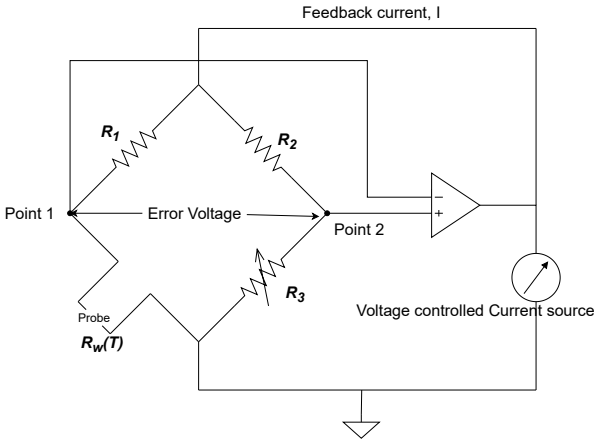


Figure 2.7: Wheatstone bridge configuration of a constant temperature hot-wire anemometer.

A problem this device has, however, is the fact that it can be affected by the room temperature. When the room temperature rises, the wire is susceptible to resistance variations, which can ultimately affect the overall reading of the flow rate and the accuracy of the values obtained.

To solve this problem, a similar configuration could be used, as shown in Fig. 2.8. In this configuration, two thermistors are used, one that is subjected to the fluid which flow rate is to be measured (resistor $R_W(T)$, the probe resistor, where T is the temperature to which the probe is subjected to), and

the other outside the fluid (resistor $R_3(T_{amb})$. The R_3 thermistor serves as a control variable to diminish the effect the room temperature has on the measurements, since its value is dependent on T_{amb}, the room temperature. The flow rate is calculated by measuring the voltage drop difference between R_3 and R_W.

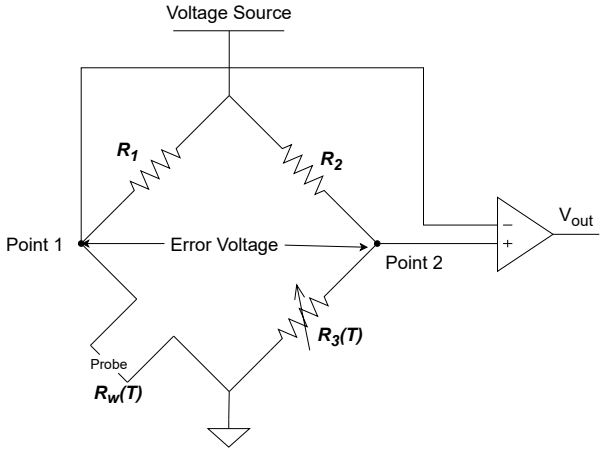
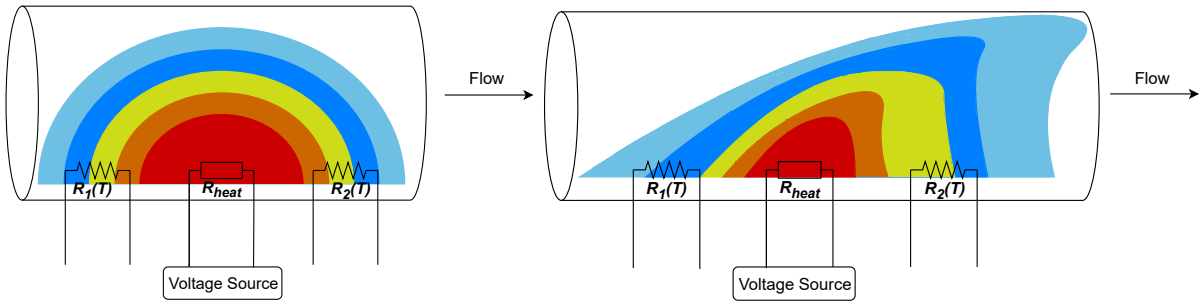


Figure 2.8: Wheatstone bridge configuration with two thermistors.

While this method guarantees no error voltage when there is no flow, that is, eliminates a possible offset that may exist due to variations in the ambient temperature, this method can still suffer an impact in readings from room temperature variations, albeit less than in the hot-wire anemometer method, since the room temperature will affect the two thermistors differently, the one subjected to the fluid will probably not be as affected as the other one, which still results in sensitivity variations.

Since the flow rate values to be measured in this project are very small and require a very high accuracy, it's important to minimize as much as possible the error in the readings. As such, a new method which presents some advantages over the previous ones is studied. In this method, the objective is to measure the difference in voltages between two thermistors in different points. In the middle point between these two thermistors, a resistor working as a heat source is present, slightly heating the fluid, and the temperature field this resistor creates will affect the voltage in the thermistors.

Fig. 2.9 shows a diagram of how this method works. The resistor R_{heat} heats the fluid and creates a temperature field. If there is no flow (Fig. 2.9(a)), R_{heat} will heat the fluid equally, and the temperature field is symmetric around it. This means that $R_1(T)$ and $R_2(T)$ will sense the same temperature. However, if the fluid is moving (Fig. 2.9(b)), the heat will travel with the fluid, that is, the heat lines will follow the flow of the fluid and the thermistors will have different temperatures, resulting in a voltage difference across them.



(a) Heat dispersion when the fluid is still

(b) Heat dispersion when the fluid is flowing

Figure 2.9: Heat dispersion in the fluid when in the presence of a heat source. Blue represents the lower temperatures while red represents the higher ones.

Fig. 2.10 shows the circuit for this sensor, where R_{heat} is maintained at a constant temperature, and $R_1(T)$ and $R_2(T)$ are connected to an amplifier in order to measure the voltage difference. All the resistors are kept inside the microfluidics flow channel.

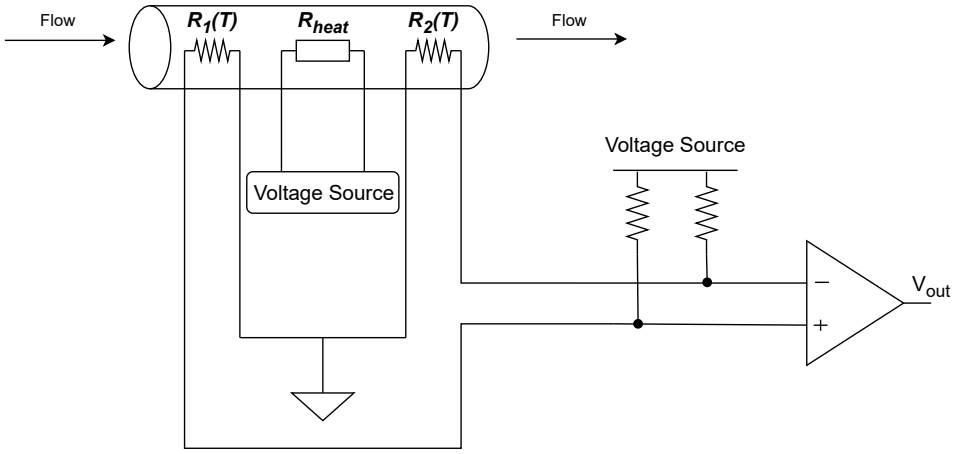


Figure 2.10: Configuration using two thermistors and a third resistor as a heat source. All three resistors are subjected to the fluid.

3

Chapter 3 - System Development

Contents

3.1 System architecture

Fig. 3.1 shows the original proposed architecture and the final developed system. Originally, the goal was to integrate the system with the cytometer module and receive information from it. However, that was not possible, due to time constraints, and the developed system can be observed inside the red lines. It consists of a controller module, which controls both the micropump and the collector, and receives information from the flow sensor, via an Analog to Digital Converter (ADC), in order to adjust the flow rate to the desired value, using a Proportional Integral Derivative (PID) controller. It also receives and sends wireless messages from and to a computer.

The controller module is made up of two different microcontrollers with distinct tasks: one controls the micropump and receives information from the flow sensor, and the second is the master, that communicates with all the components of the system, like the first microcontroller, the air collector, and the computer. A more detailed explanation of the function of both microcontrollers will be done in the next sections of this document.

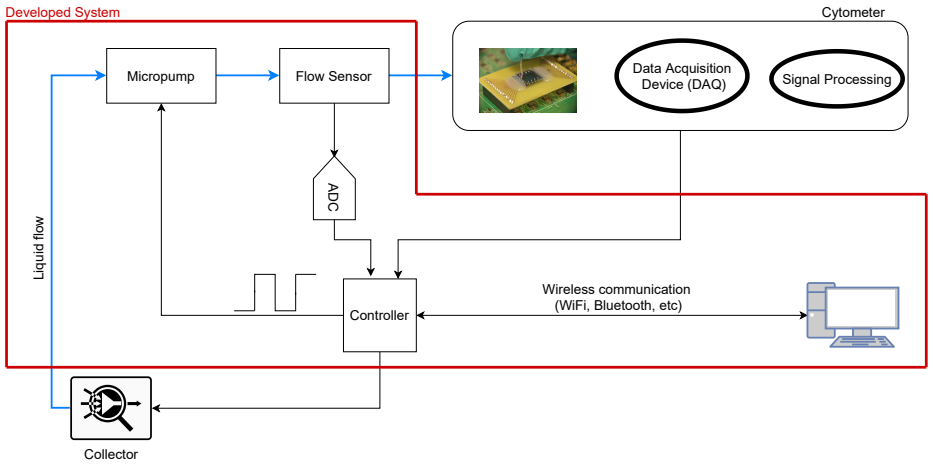


Figure 3.1: Proposed architecture and final developed system. Blue lines represent microfluidics/flow connections and black lines represent electric connections. Developed system is represented inside the Red lines.

The micropump used was the M200S peristaltic micropump from TCS Micropumps (Fig. 3.2). This pump was used due to its availability on the laboratory. This micropump only offers flow rates in the range of 330 mL/min to 700 mL/min, which is much bigger than the required 10-200 μL/min for the microfluidics channels and the cytometer. In order to overcome this setback, a system that reduces flow was developed. This system involves using needles and less wide tubing at the ending point to restrict the stream.

The flow sensor configuration adopted in this project was the one shown in Fig. 2.10 due to the fact that it offers no offset and the best accuracy from the three methods studied.

Figure 3.2: M200S peristaltic micropump from TCS Micropumps. Taken from [4].

3.2 Sensor Circuit

3.2.1 Sensor Modelling

Fig. 3.3 shows a simplified diagram of the circuit printed on a Printed Circuit Board (PCB), with the resistors as shown in Fig. 2.10 and Fig. 2.9. The circuit needs to be as symmetrical as possible to avoid any problems that may arise, and to narrow the amount of factors that may create imbalances in the circuit. Asymmetries in the distance to the middle point are such a factor, as well as asymmetries between the thermistors. To assure the best performance, the distance d_{sym} between the thermistors $R_1(T)/R_2(T)$ and R_{heat} needs to be optimized. To do so, a study of the the thermal circuit and the way the heat transfers in the fluid depending on the distance between the resistors was made, in order to understand how the sensor will function and what is the optimal distance and conditions.

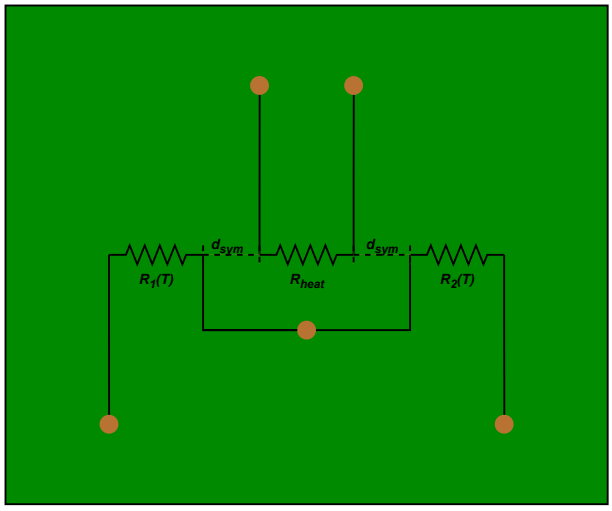


Figure 3.3: Simplified diagram of the PCB module for the sensor.

The study of the thermal system begins by studying Fourier's law for heat conduction. According to Fourier's law for heat conduction in a one-dimensional form,

$$q' = -k\frac{dT}{dx}, \tag{3.1}$$

where q' (W/m^2) is the heat flux, k (W/(m×K)) is the thermal conductivity, T is the temperature difference between the points at which a thermal transfer occurs, with x being the position of those points, which means dx is the distance L between them, that is,

$$\frac{dT}{dx} = \frac{T_2 - T_1}{L}. \tag{3.2}$$

As such, (3.1) can be written as

$$q' = -k\frac{T_2 - T_1}{L} = k\frac{\Delta T}{L}. \tag{3.3}$$

To obtain the heat rate by conduction, the heat flux needs to be multiplied by the area of the plane through which the heat transfer occurs, and as such, the heat rate q (W) is given by

$$q = q' \times A, \tag{3.4}$$

where A is the area of the plane. These equations can be further simplified by the fact that

$$R_t = \frac{L}{kA}, \tag{3.5}$$

R_t being the thermal resistance of the transfer plane.

As such, combining (3.3), (3.4) and (3.5),

$$q = \frac{\Delta T}{R_t} \tag{3.6}$$

is obtained. All the above equations and explanations can be found in [11].

From (3.6), a similarity to Ohm's law can be found,

$$i = \frac{\Delta V}{R},$$

where q is equal to the current, T to the voltage and the thermal resistance R_T to the electrical resistance. As such, an analogous electrical circuit can be found, and the heat transfer study can be more easily performed. The thermal system when there is no flow is as Fig. 3.4 shows, where T_1 and T_2 are the temperature of the thermistors $R_1(T)$ and $R_2(T)$, respectively, T_0 is the constant temperature

of the heat source resistor R_{heat}, and R_1 and R_2 are the thermal resistances of the fluid between the thermistors and the heat source resistor. Since the circuit will be symmetrical and the distance between the thermistors and the resistor will be the same, as said before, from (3.5) it can be deduced that the thermal resistances will have the same value.

Figure 3.4: Thermal system of the flow channel where the sensor will be placed when there is no flow.

To perform simulations, an electric circuit that performs like the sensor would in reality was designed, and can be analyzed in Fig. 3.5. Since T_0 needs to be constant, it can be represented by a constant voltage source. The thermal resistances between the two thermistors, T_1 and T_2, and the middle point are represented by the electrical resistances R_1 and R_2. The flow rate of the stream and the thermal capacitance of the thermistors are modeled by the variable resistances R_{f1} and R_{f2}. The value of these resistances changes according to

$$\begin{cases} R_{f1} = c \times (f(flux)) \\ R_{f2} = c \times (f(-flux)) \end{cases}, \tag{3.7}$$

where c is a constant that represents the thermal capacity, and $f(flux)$ a function varies depending on the flow rate. This function is unknown, but it represents how the resistor would behave depending on the flow rate. Since the two thermistors would behave symmetrically, one resistance varies with a positive flux, and the other with the negative value. The resistances R_{1amb} and R_{2amb} represent the thermal resistances between the thermistors and the ambient temperature, T_{amb}.

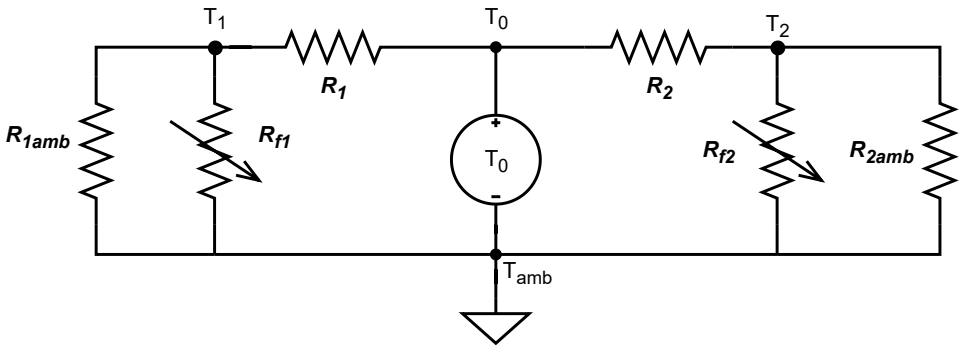


Figure 3.5: Electrical equivalent for the thermal system in Fig. 2.9

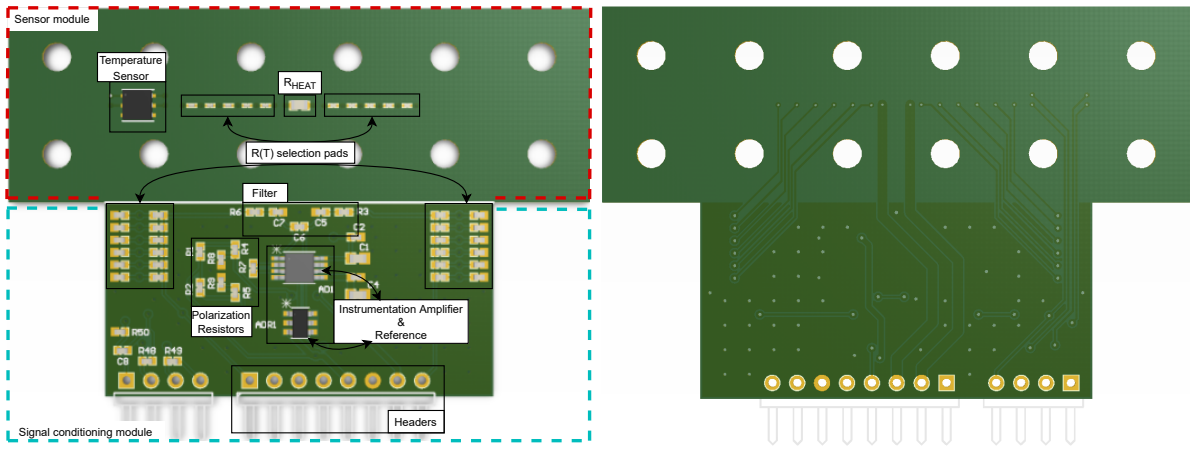
With this circuit, it is possible to run simulations having the distance between the resistors varying, and with different flow rates. This can be achieved by changing the value of the thermal resistances, R_1 and R_2, and the value of the constant $flux$ in 3.7, respectively. The simulations will show a range of val-

ues where the sensor is more sensitive and optimal. This simulation, however, was not performed, and this study is merely a simple explanation of how simulations regarding this system could be performed.

3.2.2 Sensor Design

Following the sensor modelling, it is necessary to design the sensor. Fig. 3.6 shows the PCB design for both the sensor and the signal processing modules.

The sensor module consists of a temperature sensor, and the thermistors plus heat source configuration of Fig. 2.10. Since a trial-and-error approach was preferred over simulations, instead of placing only one pair of thermistors at a certain fixed distance from the heat source, five equally spaced thermistors were placed on each side of the heat source. This allows for testing of the performance of the sensor with different distances to the heat source without having to design multiple PCBs. Along with this, a selection matrix was designed to be able to choose the thermistors and also allow for connecting them in series to test different configurations. Fig. 3.7 shows the circuit for this matrix. The red lines on the image show the chosen configuration for this project, and how the current would flow in the circuit. The matrix works by mounting and connecting only the necessary pads. If R_{27} were to be chosen instead, then R_{16} and R_{39} would be connected instead. This matrix also provides the choice of measuring the signal from resistors in series. For example, if it was needed to measure the series of resistors R_{27} and R_{28}, resistor R_{39} would be mounted instead of resistor R_{40}. On Fig. 3.6(a) the pads to choose the configuration can be seen, on the signal processing module.



(a) Top view of the sensor PCB

(b) Bottom view of the sensor PCB

Figure 3.6: Altium Designer's 3D view of the Sensor's PCB.

Since the voltage difference is very low, it is important to minimize any noise and interferences that may exist. As such, it was decided to make one only PCB with both modules to avoid any interferences

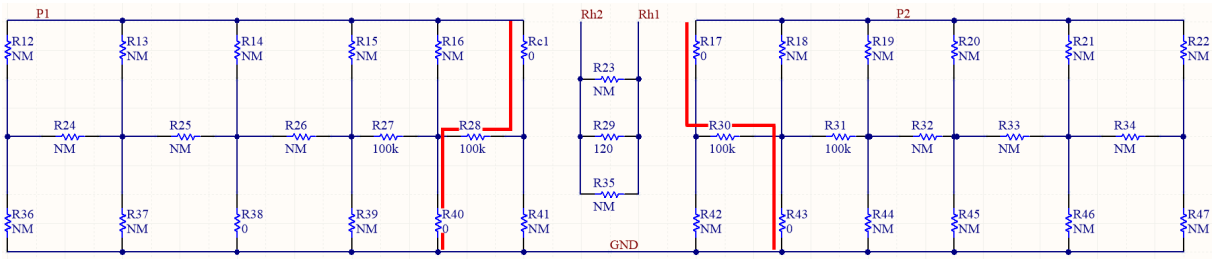


Figure 3.7: Thermistor selection pads. NM means Not Mounted and represents components not assembled in the prototype.

that may be created by long connections between the modules. The sensor module was made to contain only the minimal components necessary. The top part of the PCB, the side through which the flow channel passes by, was made so that it had the fewer possible copper tracks. The connections to the signal conditioning module were made through the bottom layer only, having vias from the top layer to the bottom one. This is to avoid any damaging of the tracks due to the passing liquid. This part of the PCB also doesn't have a ground plane, again to avoid any possible damage to the circuit due to liquids.

The thought of design to integrate the PCB with the tubing containing the fluid is by inserting the PCB between two acrylic pieces. One of the pieces would be a simple acrylic block with screw holes in it to attach the pieces together. The top piece would be designed to attach to the tubing and circulate the fluid through the PCB components. Its design can be observed in Fig. 3.8. The side holes attach to the tubing and then a tunnel connects those to the main channel.

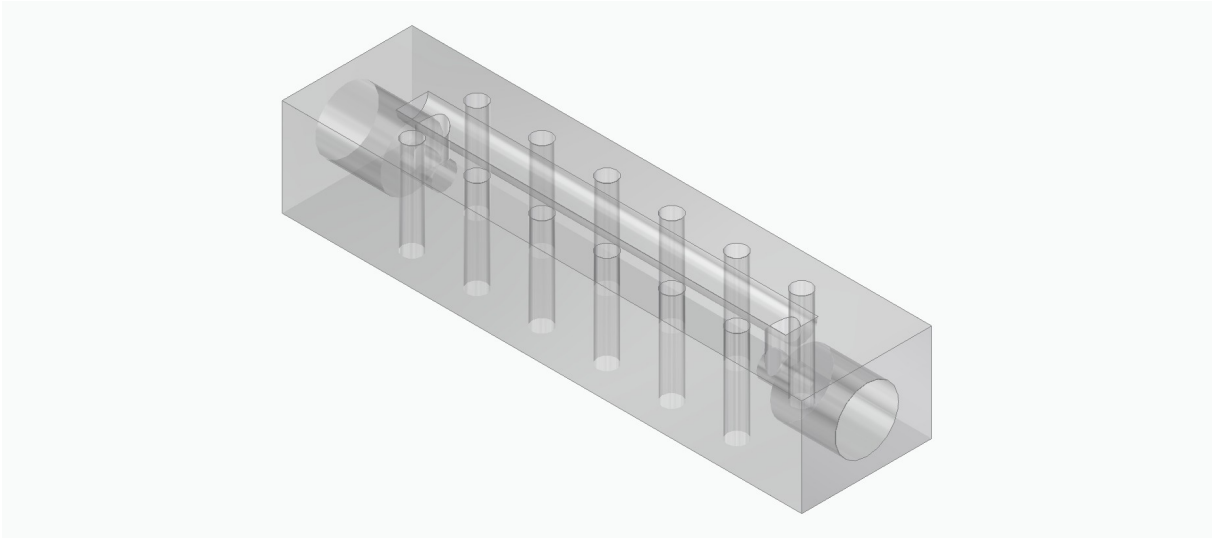
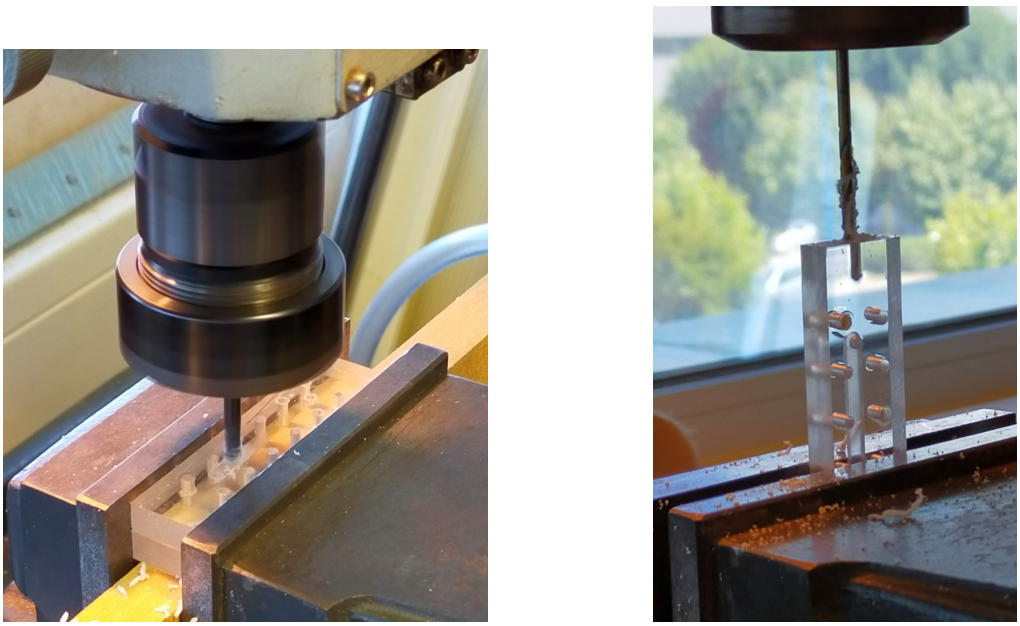


Figure 3.8: 3D design of the acrylic piece.

This piece was manufactured from a block of acrylic, using a milling machine to drill the holes and make the channel. Fig. 3.9 shows sections of the manufacturing process.

To seal the components and make sure the liquid doesn't oxidize them, and also to avoid short-



(a) Milling of the fluid channel.

(b) Drilling of the tubing connection holes.

Figure 3.9: Photograph of the top acrylic piece's manufacturing process.

circuits within the board, a thin layer of *plastik70* was applied.

Since the complete piece has enough screw holes all around the channel, it creates enough pressure against the PCB that it creates a seal around the channel, and no leakage occurs. This seal effect is also the reason why the top layer of the sensor module was designed to have a minimal amount of tracks. Without the tracks, the surface is smooth enough that it creates equal pressure all throughout the PCB, ensuring an equal seal on all sides.

3.2.3 Sensor Signal conditioning

As seen in Fig. 3.6, the sensor conditioning module consists of an amplification circuit for the sensor. The circuit can be seen in Fig. 3.10. The thermistors ($R_1(T)$ and $R_2(T)$) are connected to a voltage divider in order to allow them to be connected to the amplifier. Since it is important to have a high input impedance, an instrumentation amplifier was chosen. A high input impedance is necessary so that it ensures that no current is sinked from the polarization and sensing resistors. If there was no high input impedance, the amplifier could substantially load the sensor circuit causing an unwanted voltage drop. The output signal is then directly sent to the outside. To help reduce any noise that may exist, a Electromagnetic Interference (EMI) filter is put between the signals and the inputs of the amplifier.

The PCB also has via stitching. This is a technique in which vias are placed along the PCB in order

to reduce noise in the board. This increases the impedance of the current return loops and also acts as additional shielding. The via stitching is done in a grid with 2.5 mm of spacing between vias.

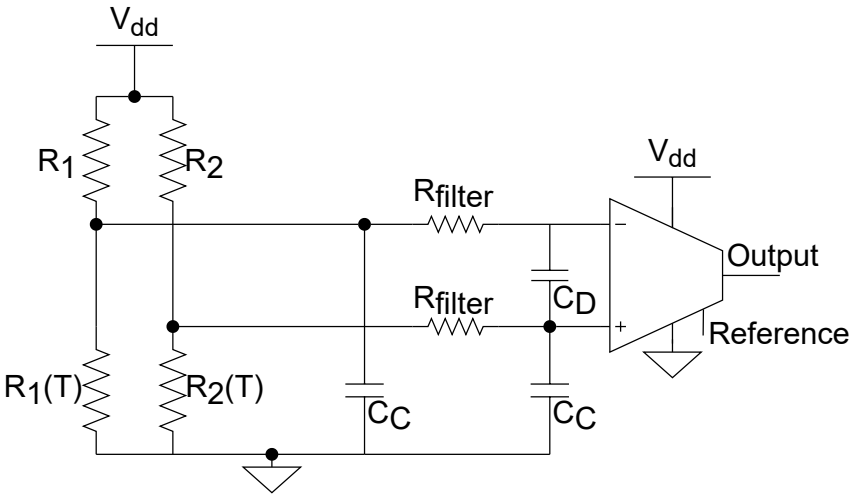
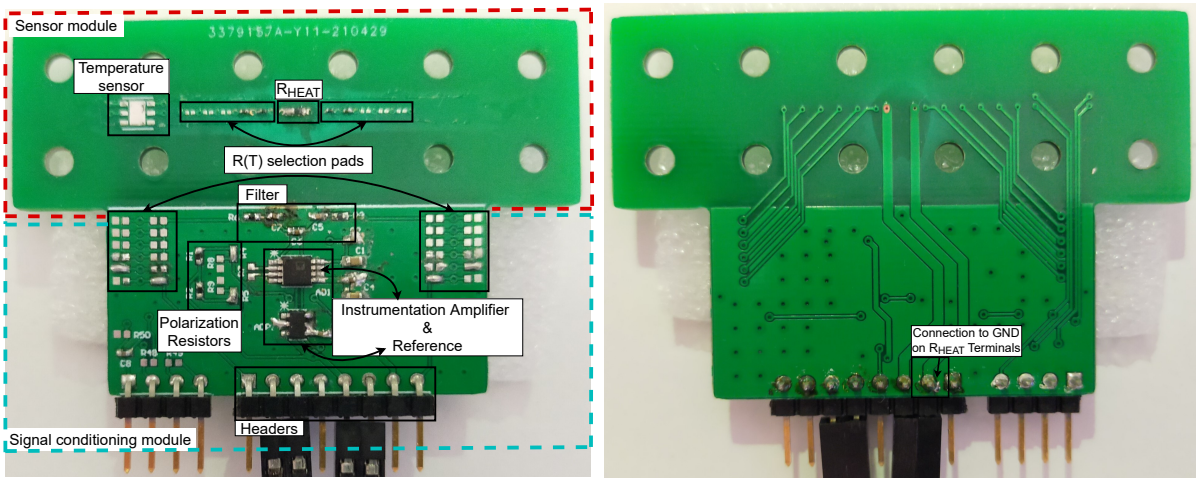


Figure 3.10: Signal conditioning circuit.

3.2.4 Sensor prototyping

The final step is to create a prototype of the sensor. In order to do so, it's important to choose the correct components for the system. As said before, an instrumentation amplifier was used. The most important characteristics of this component is that it needs to work with a 5 V single supply voltage, good Common-Mode Rejection Ratio (CMRR), and a low input noise value. The amplifier chosen was the Analog Devices' AD8227. It checks all the important characteristics, and it is also rail-to-rail output, which means the system is able to utilize the entire voltage range. Since this is an instrumentation amplifier, it requires a reference. To obtain a balanced range of values for the ADC, the optimal value would be half of the maximum value the ADC can read. Since the used ADC has a maximum value of 1.5 V (and minimum of 0 V), the reference would optimally be 750 mV. However, a reference with that value could not be found, and the closest was Analog Device's ADR130, a reference capable of outputting 500 mV. As such, the output is slightly unbalanced, and there is a wider positive voltage range than there is negative range. This imbalance would be more problematic if the system was capable of negative flow, however, since the flow should only move in one direction, this is not as problematic. Fig. 3.11 shows the produced PCB with the soldered components.

As seen in the figure, there is a spot for a temperature sensor. This sensor would improve the sensor by measuring the ambient temperature of the liquid. This could help analyse the standby point of the sensor, since this could be affected by the ambient temperature due to differences between the thermistors. However, it was not possible to implement this sensor due to a shortage in supply of the sensor.



(a) Top view of the sensor PCB

(b) Bottom view of the sensor PCB

Figure 3.11: Photograph of the Sensor's PCB.

The thermistors for the sensor module were chosen based on its resistance tolerance. It's important that the thermistors have very little error in it's value, so a low resistance tolerance is important. The used thermistors have a value of 100 kΩ. The heating source resistor was chosen in a different way. At first, a low impedance resistor was chosen (around 22 Ω), however, when testing the operation of the sensor with a 5 V supply, the current was too high and it boiled the liquid in the channel, which caused bubbles to form, and the movement of the bubbles, as well as the air pockets they formed, created disturbances in the measured signal. Since it is important to maintain the 5 V supply, a higher impedance resistor was tested, and the final prototype contains a 120 Ω resistor. With this value, there are still some bubbles present, but this was the best case tested, and so it was maintained, since a lower temperature would cause the sensor to not work, or to be less sensible.

As mentioned in previous sections, a trial-and-error approach was preferred. To test the distance to the heat source, the first two spots were soldered and tested. It was confirmed that the closest spot to the heat source was able to output good signals and a good read of the flow variation.

For the EMI filter, the formulas found in the amplifier's datasheet [12] were used, where

$$\begin{cases} FilterFrequency_{DIFF} = \left(\frac{1}{2\pi R_{filter}(2C_D+C_C)}\right) \\ FilterFrequency_{CM} = \left(\frac{1}{2\pi R_{filter}C_C}\right) \end{cases} . \tag{3.8}$$

$DIFF$ and CM refer to the differential and commom mode, respectively. The placement of the capacitors C_D and C_C are as shown in Fig. 3.10. It is also mentioned in the datasheet that $C_D \geq 10C_C$. With this in consideration, and with the available components,

$$\begin{cases} C_C = 1\ nF \\ C_D = 10\ nF \\ R_{filter} = 7.5\ k\Omega \\ FilterFrequency_{DIFF} \approx 1011\ kHz \approx 1,011\ MHz \\ FilterFrequency_{CM} \approx 21221\ kHz \approx 2.1221\ MHz \end{cases} . \tag{3.9}$$

For the polarization resistors, a value of 180 kΩ was chosen for the resistors. For a linear read of the thermistors' values, the polarization resistors need to be of a higher value than the thermistors, so that they can serve as a sort of current source. Since the polarization resistors have such a higher value, they are the ones that dictate the current value in the voltage divider configuration, which means there is less of a variation of current with the change of the thermistors' values.

Finally the decoupling capacitors used for each component were chosen based on the datasheet recommendations. For the amplifier, 10μF and 100 nF were placed on both the positive and negative voltage supply pins. For the reference, a 100 nF capacitor was added to the output. There is no need to add one to the voltage suppy pin, since that value is share with the rest of the PCB, and a decoupling capacitor was already added for the amplifier.

When testing the PCB, it was noted that the output signal was not expected, and after analysing the circuit, it was noted that the heating source did not share the ground plane with the rest of the circuit. Since the ground wasn't shared, this caused the existence of a common mode component that electrically coupled from the heating source to the thermistors, which caused a disturbance in the values read by the thermistors. To correct this, a connection to the ground plane was made in one of the heating source terminals, as shown in Fig. 3.11(b). The top layer of the PCB was scratched off to show the ground plane and that point was connected to the pin itself.

Fig. 3.12 shows a photograph of the complete sensor, together with the acrylic piece.

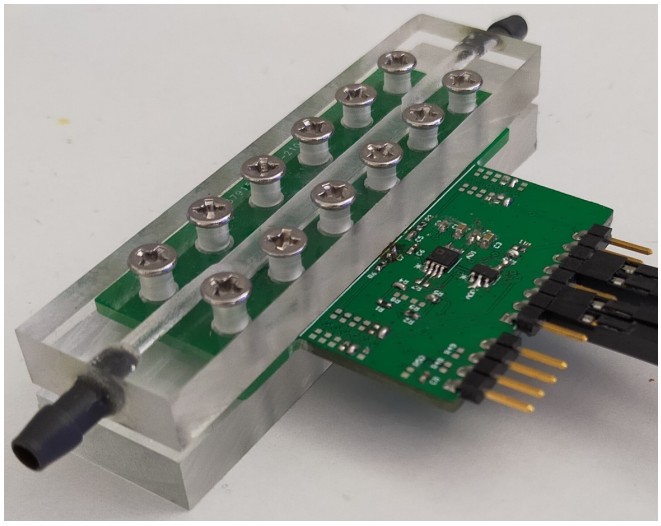
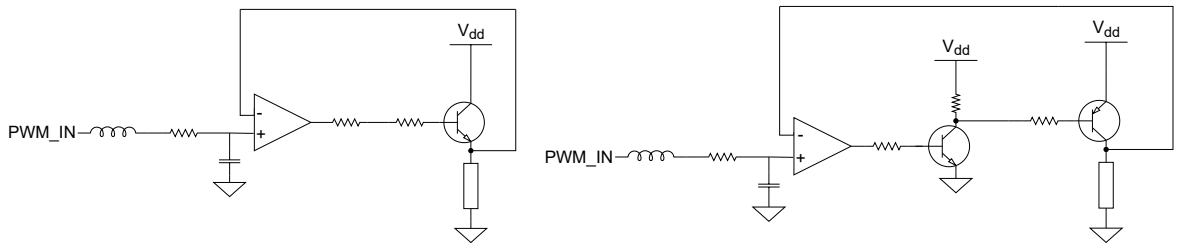


Figure 3.12: Photograph of the sensor PCB placed in the acrylic set.

3.3 Pump Driver Circuit

Initially, there were two methods considered to control the micropump. Linear control using a Digital to Analog Converter (DAC) was one of the possible methods, due to its easy linear control. However, integrated, high resolution DACs are not commonly found in microcontrollers, especially if multiple DACs are required. As such, another posible method was found: using a PWM signal (which most microcontrollers can easily generate) converted into Direct Current (DC) voltage. This method offers an integrated, high resolution solution. Converting the PWM signal into a DC voltage, instead of directly applying the PWM signal to the pump with a transistor driver, also avoids the creation of EMI, since there is no commutation of the current applied to the pump. This avoids possibly damaging other eletronics in the system, such as the cytometer, due to the EMI.

PWM works by using a square wave with varying duty-cycles to change the average voltage value, and thus the average power supplied to the load. If the frequency is fast enough, it simulates a constant voltage that can be easily varied by changing the duty-cycle. To convert the PWM signal into a DC voltage that controls the pump, multiple circuits were designed for testing. A PCB with two different circuits was fabricated, so that both circuits could be tested without fabricating multiple different PCBs. Fig. 3.13 has the two possible circuits that can be formed depending on the connections made, and Fig. 3.14 shows the 3D view of the design PCB.

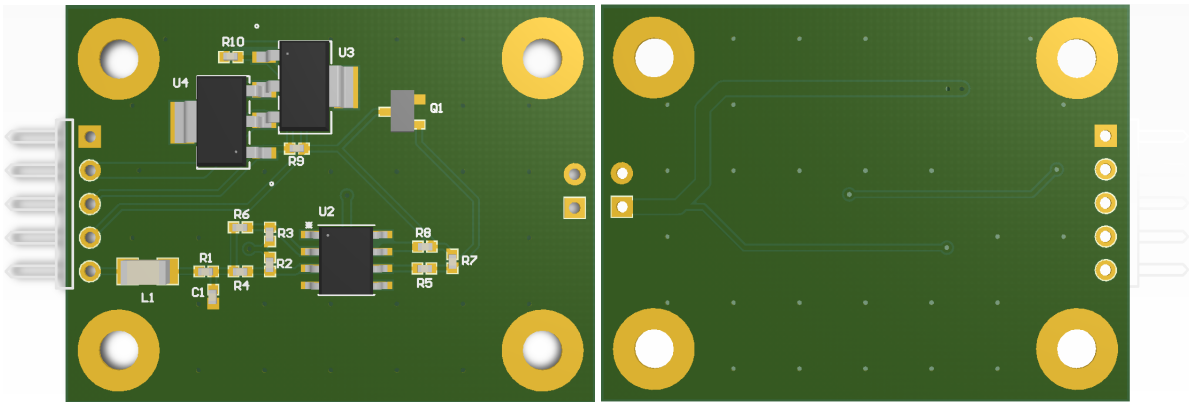


(a) NPN transistor circuit **(b)** NPN driver transistor and PNP transistor circuit

Figure 3.13: PWM driver circuits.

As seen in the above figures, to reduce the mentioned EMI, the PWM signal from the microcontroller is filtered. The PCB has the possibility to implement a second order filter, if necessary, or a simple first order filter. In this case, a first order filter attenuated the signal sufficiently.

Both circuits in Fig. 3.13 have a operational amplifier as a buffer, with a follower circuit configuration. This means that the point connected to the negative input follows the voltage at the positive input. This means that the voltage to the load replicates that of the PWM (after the filter). The only difference between Fig. 3.13(a) and Fig. 3.13(b) is that the second one has a common emitter used to drive the output transistor with extended voltage swing. The pump driver transistor also changes to a PNP



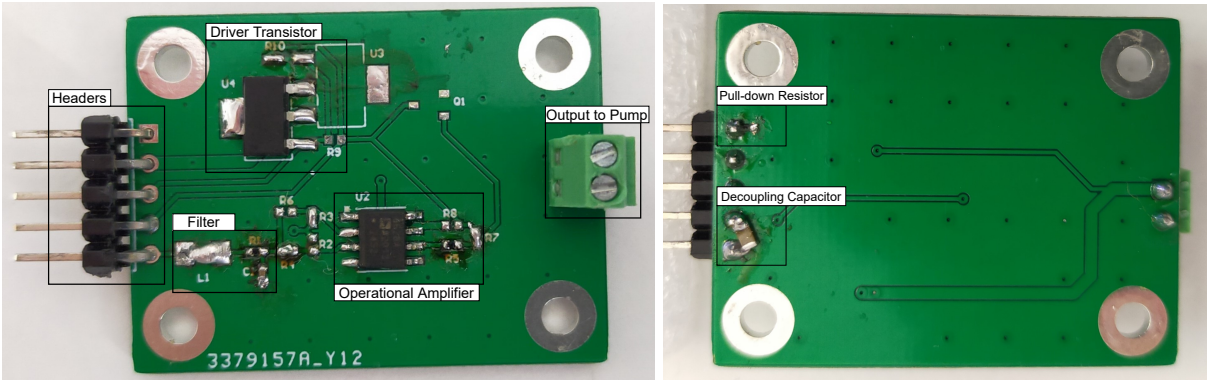
(a) Top view of the pwm PCB **(b)** Bottom view of the pwm PCB

Figure 3.14: PWM driver PCB.

transistor, since the common emitter amplifier inverts its input, that is, as the input increases, the output of the amplifier decreases. As such, to ensure the negative feedback of the overall loop, a PNP transistor was needed.

Besides possible current issues Fig. 3.13(a) has, there could also be a problem with the voltage drop. Since it is a NPN transistor, the voltage at the emitter comes from the base, with a voltage drop at the base-emitter junction. The value of this drop depends of the transistor, but is around 0.7 V. With this, there could be a problem obtaining the full value range with this circuit.

The first circuit tested was the one in Fig. 3.13(a). Since this circuit has less components, it would be more advantageous in the future, be it cost-wise or space-wise, allowing for smaller PCBs. Fig. 3.15 shows a photograph of the PCB with the components and connections for this circuit. After testing the circuit, it was verified that it worked correctly if the circuit was properly dimensioned.



(a) Top view of the pwm PCB **(b)** Bottom view of the pwm PCB

Figure 3.15: PWM driver PCB.

The most important components in this circuit would be the filter, the transistor, and the resistance at the amplifier's output. The filter was dimensioned to filter anything above 10 Hz, since that is a close enough to DC value, and sufficient to attenuate the signal to a reasonable value. Another reason for this value is that the flow rate itself would not change at a speed higher than 10 Hz, that is, it would take more than 100 ms for the flow to change, and for the sensor itself to pick it up, due to the time constants of the thermistors. This means that even if there is are slight peaks and the signal isn't completely analog, it would not affect the overall operation of the system because the peaks would be too fast to actuate on. The resistor used in the filter has a value of 33 kΩ and the capacitor a value of 470 nF. As will be seen in chapter 4, this creates a nearly linear signal. The transistor chosen was the BCP68 from Nexperia. It was chosen due to its high maximum current. In this application, so that the circuit can function without a current amplifier to drive the second transistor, it's important that the transistor used is able to source enough current to the pump. Finally, the resistance at the output of the amplifier needs to be small enough so that the current provided to the transistor is also sufficient for it's correct functioning. As such, a 500 Ω resistor was used.

The criteria for choosing the operational amplifier was its supply voltage. It needed to properly work with the 5 V that would supply the PCB. As such, the AD8628 from Analog Devices was chosen. This amplifier, besides being operating on a 5 V single-supply mode, also has both rail-to-rail input and output. This means there is no need to worry about the input and output values maximum values, since they should always be less than 5 V (since the maximum PWM value is 3.3 V), and the minimum value should at most be equal to zero.

As seen on Fig 3.15(b), a decoupling capacitor and a pull-down resistor were also added after the PCB was produced. The decoupling capacitor has a value of 10 μF and the pull-down resistor a value of 11 kΩ. The pull-down was added to assure that the signal is zero when the pin is left open.

As in the sensor's PCB, this one also employs via stitching. In this PCB, the vias are applied on a 5X5 mm grid.

3.4 Software/Firmware

To develop this project, the system was divided into two different blocks based on functionality. There's the low level digital signal processing block, which receives the signals from the sensor and processes them, using those signals to control the micropump. It also receives and sends information from and to the High level controller block. This is the block that serves as a sort of master in the system. It performs all the high level functionalities, such as communicating with the computer (via Bluetooth) and sending commands to both the low level block and the air collector. Fig. 3.16 shows a block diagram of the relations between the different blocks and components of the system.

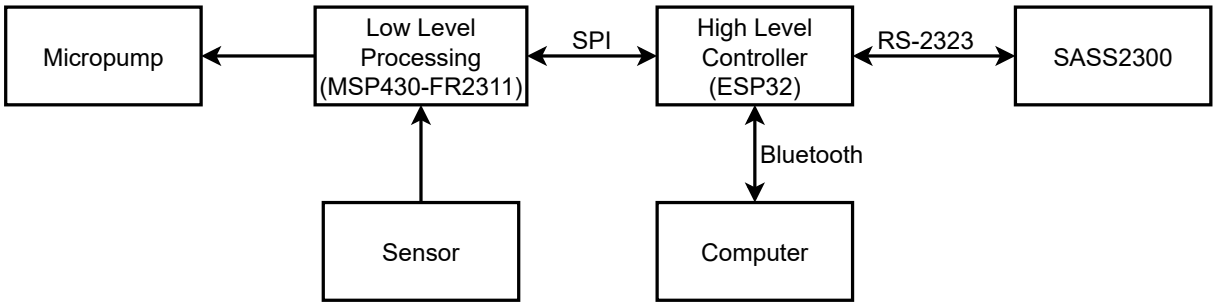


Figure 3.16: Block diagram of the system.

As seen in the figure, two different microcontrollers are used for the different functionalities. For the low level processing, a MSP430-FR2311 is used, while for the high level controller, an ESP32 is used.

Fig. 3.17 Shows a diagram with the connections between all components.

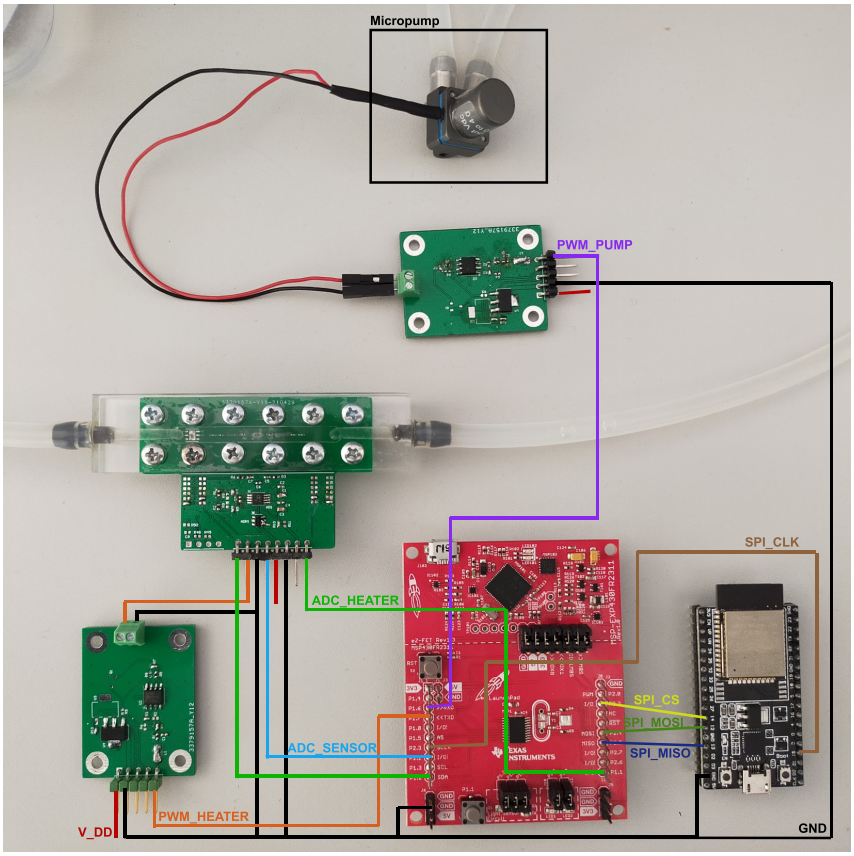


Figure 3.17: Diagram of all the connections between components.

3.4.1 Low Level Digital Signal Processing

3.4.1.A PID controller

A vital part of this project is the pump control. The objective is to be able to maintain a stable flow rate throughout the process, and to do that, the pump needs to be controlled accordingly. If the flow is restricted and drops in value, the pump needs to make up for that restriction by increasing its value. In the same way, if the flow suddenly increases, the pump needs to be able to decrease its value so that the flow stabilizes in the desired value.

To perform this control, the PID controller was studied. This controller actuates by virtue of 3 different terms. As the name indicates, there is one term that is proportional to the error, another proportional to the integral of the error, and the last one is proportional to the derivative of the error. The PID control equation in the time domain, as shown in [13], is

$$u(t) = k_P e(t) + k_I \int_{t0}^{t} e(\tau)\, d\tau + k_D \dot{e}(t). \tag{3.10}$$

To apply this type of controller, it is first needed to understand what type of system there is, and what type of control it needs. Some systems only require a proportional type of control, while some require only proportional and derivative control in order to function well.

To understand each term's importance, one can think of proportional control as taking into account only the present. It takes only the current error into account for the control of the system. Integral control is memory, or the past. It remembers what happened before and takes that into account for how it should control the system. Derivative control is the future, where the controller takes into account what can happen and prepares for it when controlling the system. However, each term has an associated problem, which is usually solved by one of the other two terms. For example, depending on the system, the proportional term has a steady state error, that is, the error never reaches zero, and instead hovers around the desired goal. This problem is solved by the integral term. This term is a sum of all past errors, and since the error is still non-zero, it continues controlling the system towards the desired goal. However, this term usually has oscillation problems, and makes the transient response longer. The derivative term solves both of those problems. The oscillation problems usually come from overshooting the goal, that is, passing the desired position, and then correcting it. Since the derivative term takes into account what may happen in the future, it prevents the overshooting, and consequently, the oscillations in the output of the system.

As previously stated, for this project, it is only necessary to reach the desired flow rate and then maintain that value. As such, the type of controller needed is a PI controller. The proportional term serves to reach the desired value, and the integral term serves to maintain the value, so that the pump doesn't simply turn off when the proportional term reads zero error. The derivative term could be imple-

mented to provide a smoother transition to the desired flow rate, however, it is not necessarily needed. If the system presented noticeable oscillations as it neared the desired value, the derivative term would be important to implement, however, that was not the case. As such, the control equation will be

$$u(t) = k_P e(t) + k_I \int_{t0}^{t} e(\tau)\, d\tau. \tag{3.11}$$

However, as described in [13], *whenever an actuator is used that can saturate (which is almost always the case), extra care is required in implementing integral control. The controller must be augmented with an anti-windup feature to deal with the actuator saturation*. The problem with integral windup is that the integrator keeps summing, even when the actuator cannot physically comply to the commands. This means that when the system needs the actuator to reverse the action, it takes a higher amount of time, since the system needs to overcome the integrator error.

In this case, since the motor can saturate, that is, it has a maximum (and minimum) value of operation, it is necessary to limit the algorithm so that it does not go over the possible values. As seen in listing 3.1, the anti-windup feature basically consists in stopping the algorithm from integrating when the result value goes over a certain upper or lower limit. In this case, the lower limit is zero, since the pump can only move in one direction, while the upper limit is the maximum value the PWM can take, which is 2^{16}. The anti-windup algorithm also needs for the errors to both be either negative or positive, since if they are of different sign, that means the integrator is trying to reduce the proportional error, instead of worsening it.

Listing 3.1 shows the complete software implementation of the PI controller. It calculates the error by subtracting the ideal value and the sensor's value, and then multiplies that error by the proportional gain. Afterwards, it calculates the integral error. In the first instance of the controller, this value is 0, since there is no past error. This error is calculated by summing the errors, and multiplying the new added value by the time constant *DT*, as seen on line 14 of the listing.

Listing 3.1: PID function for the firmware of the MSP430 microcontroller.

```
1 void PID_function(uint16_t sensor){
2     error = ideal;
3     error = error - sensor;
4     u = KP*error;                 // Q0 = Q0*Q0
5     u = (u<<16) + KI*error_i;           // Q16 = Q16 + Q16
6     u = u>>16;
7
8     // Anti-windup
9     if((u >= UMAX || u < 0) && (((error >= 0) && (error_i >= 0))
```

```
10                                || ((error < 0) && (error_i < 0))){
11          error_i = error_i;}
12      else{
13          error_i = error_i + DT*error;}}
```

Listing 3.2 shows the values and types of the variables and constants used. The terms gains (constants *KP* and *KI*) for a PID controller are hard to tune, and the ones used in this project were obtained virtue of trial and error. They are not optimized for the system. The time constant *DT* refers to the actuation time of the controller, and should be of a lower frequency value than the acquisition frequency. In this case, its value is of 100 Hz (0.01 ms), which means the acquisition frequency needs to be equal or higher in value.

Listing 3.2: Values for the variables and constants used

```
1  #define DT       655     // Q16 - 100 Hz
2  #define KP       100
3  #define KI       25
4
5  uint16_t ideal;
6  int32_t error = 0;
7  int64_t u = 0;
8  int64_t error_i = 0;
```

In order to optimize the code and ensure that the microcontroller performance is enough for the needed processing, fixed point arithmetic was used. This choice allows for faster operations, since all the variables are integers, and the used microcontroller has no hardware support for floating point operations. This representation works by establishing how many fractional digits are needed in order to maintain the needed resolution. The representation is noted by Qx, where x is the number of bits used to represent the fractional digits. If a completely fractional unsigned number is represented by a 16 bit variable, all of the bits can be used to represent the modulus. However, if it is a signed number, one of the bits needs to be used for the sign. The way the number is converted is by multiplying it by 2^x, where x is, as said before, the number of bits used on the fractional part representation. This is usually performed by shifting the number, since shifting is the easiest operation for the microcontroller to perform. To obtain the original value again, after the arithmetic is done, the variable needs to be divided by the same number it was multiplied, 2^x, and the result is obtained in its original fractional form.

The portion of the code where most, if not all, arithmetic is performed is present in Listing 3.1. In this case, some of the variables are signed, and others are unsigned, so the arithmetic needs to be performed carefully. The *error* variable is always an integer, since it comes from reading the ADC, which

only gives integer values, and as such does not need to be changed. The *error_i* variable, however, comes from a multiplication with the constant *DT*. This constant represents the time constant in the integration, and is usually fractional. In this case, it is 0.01 s, and needs to be transformed into fixed point representation. To do so, and since there are no integer digits originally, all sixteen bits of the variable can be used to represent the constant fractional part. To find its representation, the fractional number is multiplied by 2^{16}, and its value becomes 655. To perform a multiplication with fixed point representation, the result needs to be represented by a number of bits equal to the sum of the bit-width of the multiplied variables. That is, if the values to be multiplied are represented by Qx and Qy, the resulting value would need to be represented in Q(x+y). This means that the final value of the multiplication of *error* and *DT* after the operation on line 15 would need to be Q16, since *error* is Q0, and *DT* is Q16. As such, *error_i* would need to also be Q16, since to perform a sum, all variables need to be of the same representation. This situation is analogous to a regular sum, in which the comma needs to be aligned so the operation can be performed. In order for the comma to be aligned for the sum, all the numbers need to be represented in the same format. As such, on line 6, to perform the sum of *u* and *KI*error_i*, the variable *u* needs to be converted into Q16. Afterwards, and since *u* needs to fit in a 16-bit variable (since that is the maximum resolution for the PWM), the number is once again divided by 2^{16}, to return to its correct value.

3.4.1.B PWM signal

As mentioned before, PWM offers a high resolution solution for the pump control. In this project, the PWM signal was generated using the Timer module of the microcontroller, and as such, the resolution is dependent on the bits of the register the module uses. In this case, the entire register was used to count for the timer, and as such, the highest resolution of the controller is of 16 bits. This value can be changed in case a slower operation frequency is needed (in order to more easily filter the signal, for example).

3.4.2 High level controller

One of the goals for this project is to have remote control over the system. There were multiple ways to achieve this goal, however, due to the ease of use, Bluetooth was chosen as the wireless protocol to use. To achieve communication between a computer and the system, a microcontroller capable of wireless communication is required. This is why, as said before, the chosen microcontroller was an ESP32 with a Bluetooth module.

Due to the processing and communication capabilities of this microcontroller, it will be used as the central component of the system. This is the component that will receive and process any requests by the user and then communicate with the other components.

It communicates via Serial Peripheral Interface (SPI) with the MSP430, and via RS-232, using the Universal Asynchronous Receiver/Transmitter (UART) controller, with the air collector. It acts as the master in the protocol while the other two are the slaves. While in the SASS 2300 case, it does not receive any messages back, in the other case, it will receive updates on the system's status if requested by the user.

The ESP32 is programmed to initiate the Bluetooth protocol and remain listening for any devices trying to connect to them. After connecting to a device, it stays in a while cycle listening for any messages. At the same time, the ESP32 is also constantly sending status update requests to the MSP430. This message only changes if the user decides to modify the flow rate of the system. In that case, the ESP32 sends a message with the value change to the MSP430 and then returns to sending update request messages. This way, if the user requests an update, the ESP32 already has the necessary values, and only needs to send them to the user via Bluetooth. This is all done concurrently. There are two execution threads, one for the SPI communication and the other for the Bluetooth communication.

In order to maintain the integrity of the values received from the MSP430, a mutex is used to block the variables until the SPI communication finishes, as can be seen in listings 3.3 and 3.4. If the code is in the middle of a SPI communication, the Bluetooth cycle cannot write to the computer, and needs to wait until the communication is ending, and if the Bluetooth cycle is writing to the computer, the SPI communication cannot start.

Listing 3.3: Section of the code for the SPI cycle where a mutex is used.

```
while (1){
    vTaskDelay(100 / portTICK_PERIOD_MS);
    xSemaphoreTake(xMutex, 10 / portTICK_PERIOD_MS);
    if (REQ){
        memcpy(TX_buffer, sendreq_buffer, sizeof(sendreq_buffer));
    }
    else{
        memcpy(TX_buffer, sendpwm_buffer, sizeof(sendpwm_buffer));
        REQ = 1;
    }
    while (n < 8){
        t.length = 8;
        t.tx_buffer = &TX_buffer[n];
        t.rx_buffer = &RX_var;

        ret = spi_device_polling_transmit(handle, &t);
```

```
17          memcpy(&recv_buffer[n], &RX_var, sizeof(RX_var));
18          n++;
19      }
20      xSemaphoreGive(xMutex);
21      n = 0;
22  }
```

The *SERIALFLAG* variable in listing 3.4 only becomes active when a Bluetooth message comes, which means that this code only runs if there is a command from the user side. The *memset* function is there to guarantee that the buffer with the Bluetooth message is always emptied of the previous message, which guarantees that the new message does not become corrupted by already existing values in the buffer.

The ESP32 can respond to three different requests, depending on the value it receives on the first byte of the Bluetooth communication. As seen in listing 3.4, it verifies if the byte is 0xAA, 0xFF, or another. The 0xAA byte represents a request to the air collector. If that byte is received, it will copy the remaining message and send it to the air collector. The 0xFF byte represents a status request. As previously stated, the ESP32 is always updated on these values, and as such, only needs to send back to the user the corresponding buffer. If the byte is anything else (usually it is 0x00), it represents a flow rate value change, and the ESP32 sends a command to the MSP430.

Listing 3.4: Section of the code for the Bluetooth cycle where a mutex is used.

```
1  while (1)
2      {
3          vTaskDelay(100 / portTICK_PERIOD_MS);
4          if (SERIALFLAG)
5          {
6              if (blt_buffer[0] == 0xAA){
7                  uint8_t *data = (uint8_t*) malloc(3);
8                  char y[3] = {blt_buffer[1], blt_buffer[2], 0};
9                  memcpy(data, y, 3);
10                 uart_write_bytes(UART_NUM_1, (const char *) data, 3);
11             }
12
13             else if (blt_buffer[0] == 0xFF)
14             {
15                 REQ = 1;
16                 xSemaphoreTake(xMutex, 10 / portTICK_PERIOD_MS);
```

```
17                  esp_spp_write(blt_handle, sizeof(recv_buffer), recv_buffer);
18                  xSemaphoreGive(xMutex);
19              }
20              else
21              {
22                  REQ = 0;
23                  memcpy(&sendpwm_buffer[1], blt_buffer, sizeof(blt_buffer));
24              }
25              memset(blt_buffer, 0, blt_len);
26              SERIALFLAG = 0;
27          }
28      }
```

The message protocol agreed on for the communication between the ESP32 and the MSP430 is as follows: Each message will consist of 8 distinct fields of 8 bytes each. The first byte indicates the nature of the message, if it is a request for updates or a modification of the desired flow rate. As seen in Fig. 3.18, if the message is a request, the first byte will have a value of 0xFF, the next one will be a dummy byte, to allow for the MSP430 to sync up, and the 3 remaining pairs of bytes will have the information for the Temperature, PWM, and ADC values, respectively. The protocol needs a dummy byte to sync up because the MSP430 response will always be a byte afterwards. That means, that in order to not lose any information, a dummy byte needs to be sent. The dummy byte is also important in order to give time to the MSP430 to read and respond correctly to the given command. This way, the information will be in the correct field and can be easily read by the ESP32.

In this project, the temperature sensor was not implemented and thus not integrated in the code, but the message protocol keeps on taking it into account, for possible future integration of a temperature sensor in the project. For a change of the desired flow rate, the first byte is of value 0x00, and then the value is sent on the third and fourth bytes, with the remaining bytes being dummies to maintain the 8 fields format in both the request and the value change messages.

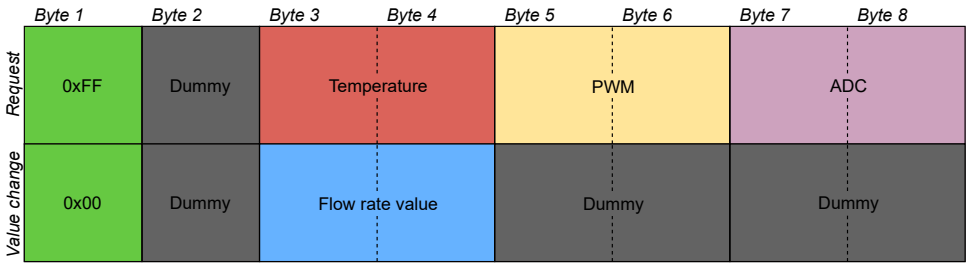


Figure 3.18: Diagram of the agreed on message protocol.

3.4.3 Graphical User Interface

As previously stated, to be able to remotely control the system, Bluetooth was used. It is used to communicate between the computer and the ESP32 microcontroller with a Bluetooth module. For the user to be able to perform various tasks within the system, a GUI was created using python and various modules.

For the graphical component of the interface, a built-in python module called TKinter was used, and to perform the Bluetooth communication, the PySerial module. The use of the PySerial module comes from the fact that on a computer, the Bluetooth functionalities work as an emulated serial port. As such, the serial communication functions of this module can be used.

Since this is a very simple GUI, it was divided into two separate parts: The SASS 2300 related commands, and the pump control commands. As seen in Fig. 3.19, the SASS 2300 commands contain only two on/off buttons that control the operation of the air collector's internal pump and fan. When pressed, the buttons activate a function that sends a command to the ESP32, which in turn sends the corresponding message to the air collector and performs the desired operation.

For the pump control commands, there's again two different modules: the command sending module, and the update receiving module. The command sending module consists of an entry bar, where the user can input the desired flow rate value, which is then sent by Bluetooth to the microcontroller, and then by SPI to the MSP430. It also has a request button, which when pressed, sends a request for updates to the system, and then receives and displays said updates on the receiving module.

There's also a drop-down menu, common to both modules, with the multiple serial port options, in which the user must select the correct one for each system.

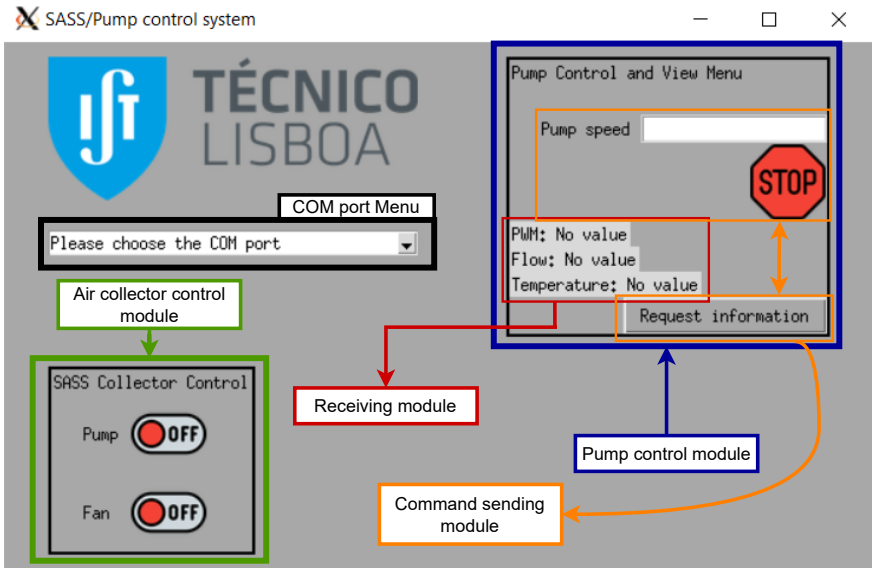


Figure 3.19: Developed GUI for system control.

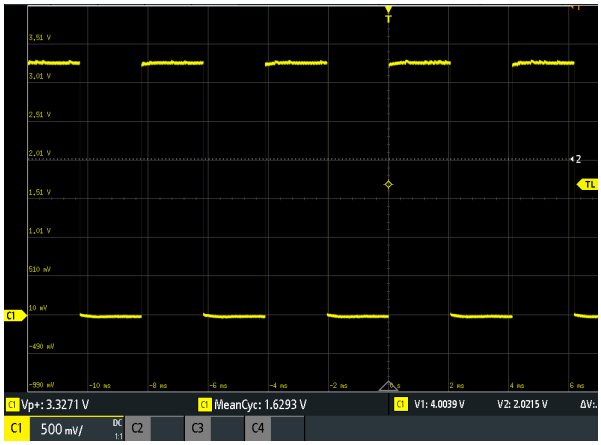
4

Chapter 4 - Experimental Results

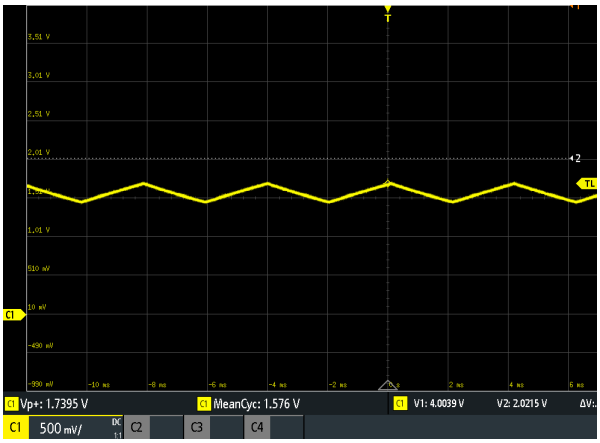
Contents

4.1 Microfluidics Pump Driver Electrical Characterization

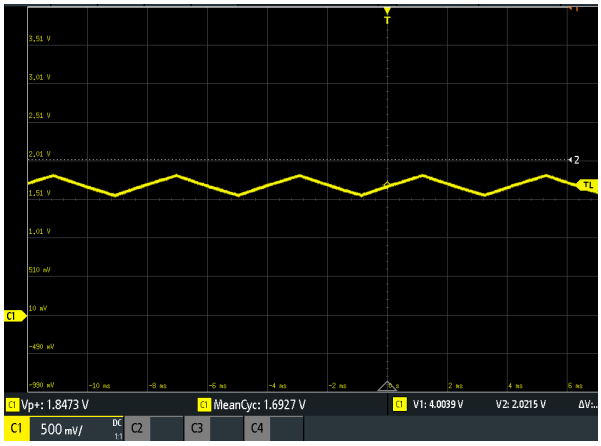
To test the correct operation of the PWM driver circuit, the waveform of the PWM signal was analyzed at different points of the PCB. Fig. 4.1 shows the waveform of the signal at four different points: The original PWM signal, after the filter attenuation, after the amplifier, and finally at the output to the pump.



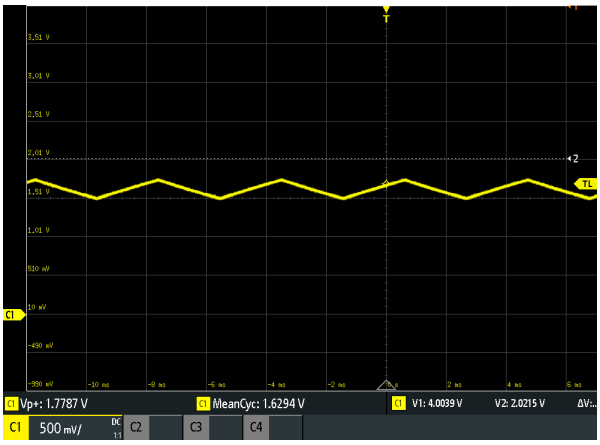
(a) PWM input



(b) Filter output



(c) Operational Amplifier output



(d) Output to pump

Figure 4.1: Oscilloscope image of the waveform of the PWM input at various phases in the PCB

At the bottom of the waveform, the mean voltage value can also be seen. Analyzing the waveform at Figs. 4.1(a) and 4.1(d), and the mean value, it can be seen that it is roughly the same, which means that the filtering was successful. This test was made with a duty cycle of about 50%, which means that the expected value should be about half the maximum value. As shown in Fig. 4.1(a), the maximum value of the PWM signal is about 3.32 V, which means that the mean value of 1.629 V is about half, as expected.

Another important test to perform in this type of circuit, is the power consumption. To determine how much current it consumes, the current at 4 different conditions was measured. The conditions were: no load, load of 1 kΩ, maximum value ($\approx$3.3 V) of PWM signal, and minimum value ($\approx$0 V) of PWM signal. Table 4.1 shows the measured values.

Table 4.1: PWM circuit's power consumption.

			Current (μA)
No load	PWM	MIN	599.8
	PWM	MAX	748.1
Load = 1 kΩ	PWM	MIN	753.5
	PWM	MAX	3350.7

As the table shows, the quiescent current (current drawn by a circuit in standby mode with small or no load) of the circuit is very low, and most of the current is drawn by the load, and not the circuit itself.

The oscilloscope used in these results was a Rohde & Schwarz RTB2004, and the multimeter used was a UNI-T UT71C.

4.2 Sensor Circuit/Characterization

4.2.1 Sensor Characterization

In order to characterize the sensor and be able to find its operation range, it is necessary to test the sensor at various different known flow rates. To perform these different measurements, a device able to output a constant known flow rate is necessary. This device also needs to be able to output low enough flow rates. As such the device shown in Fig. 4.2 was assembled. Optimally, a syringe pump should be used, however, due to the lack of availability, a self-made one was assembled. It uses a syringe, as a regular syringe pump does, and then uses a stepper motor to move a cylinder that pushes the syringe. The stepper motor is controlled using a CNC shield V3 mounted on an arduino. The arduino is programmed with the CNC milling controller and g-code parser GRBL.

The stepper motor can be programmed to go as slow as 1 mm/min. A millimeter on the used syringe corresponds to about 1/6 of a milliliter. As such, the slowest flow rate achievable with this configuration is 1/6 mL/min, which is within the 10-200 μL/min range total of the cytometer.

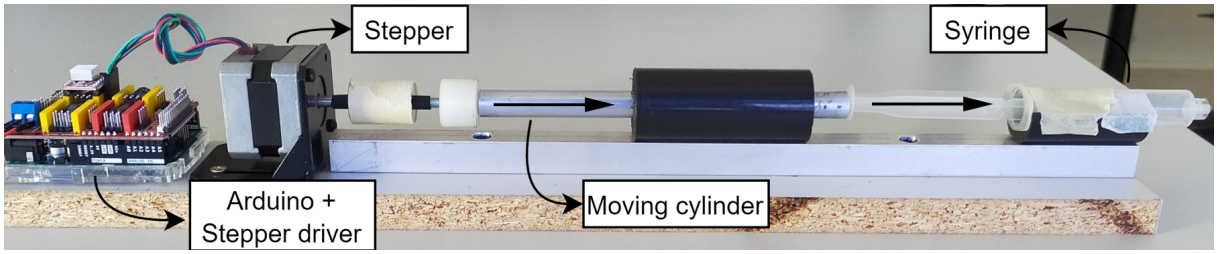
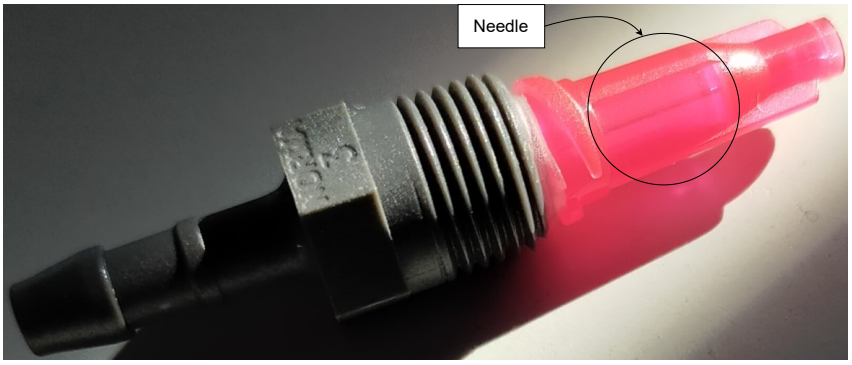


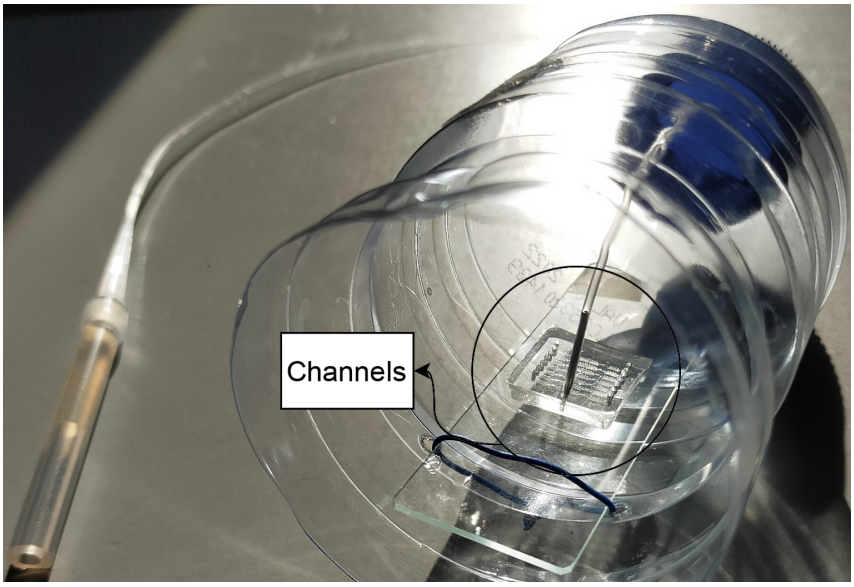
Figure 4.2: Photograph of the finished Syringe pump setup.

After using the syringe pump to obtain the sensor's operation range, a test using the micropump would be conducted, with a flow restrictor, like the ones in Fig. 4.3, to be able to achieve smaller values of flow rate with the micropump available. The flow rate of the liquid would have to be measured with other means to compare to the values the sensor would output. The measurement could be performed by making timed tests and then measuring the weight of the resulting liquid, using it to calculate its volume, and then, with the volume and time, calculate the flow rate. Comparing the sensor output from these tests with the results from the characterization with the syringe pump, it would be possible to analyse the error of the sensor.

Unfortunately, the sensor was not able to be characterized due to its bad operation. The operating point could not be found. This could be due to multiple factors. The rising of the ambient temperature of the liquid could cause the operating point to shift throughout the operation of the sensor, and without a way to measure that factor, it cannot be controlled and accounted for in the system. The bubbles formed by the heating source also affect the operation of the sensor, however, reducing the temperature means the temperature field is not big enough to reach the thermistors. This causes the thermistors to not have a relevant value difference between them. Another problem should be the fact that the heating source is not controlled. That is, the resistor acting as the heater is fed a fixed voltage, and as such, it's temperature rises. However, this rise in temperature is not controlled, which means the temperature it is sourcing the system could (and does) change with time. This, along with the rising of the ambient temperature, should be the most relevant problem for the sensor's operation, since the sensor has an expected behaviour with a flux of air. Before testing the circuit with liquids, the circuit was first tested with a varying flux of air, and its behaviour was as expected. This means that when the measuring target has a lower thermal capacitance, the sensor performs better. Another possible problem related to the use of liquids could also be current being conducted through the liquid, which would cause undesired interferences.



(a) Photograph of the flow restrictor used.



(b) Photograph of the channels used to restrict flow

Figure 4.3: Photographs of the flow restrictors used to reduce the micropump flow rate to values closer to a real situation.

A way to solve one of the possible problems in the circuit is improving the circuit and adding a control system for the heating source. Fig. 4.4 shows a possible implementation of an improved circuit. This implementation takes the signal directly from the thermistors into the microcontroller, where the values are read and an average is done, with the result being applied to a PID controller. The average of the two thermistors could be used to estimate the bulk temperature in the heater and sensing resistors region. The output of the controller is then used to control the voltage to the heating source resistor. PWM would be used for the output of the microcontroller, with the pump driver circuit being reused to achieve analog voltage values. The goal for this additional control loop is to ensure that regardless of the flow rate, the temperature operation point remains known and as constant as possible. With this aspect sorted, it is

expected that the differential measurement of the sensing resistors allows for an accurate and stable estimation of the flow rate.

This circuit is being tested and implemented. Its implementation is relatively simple, since it reutilizes components used in the other sections of the project. The microcontroller has more ADC channels that can be used, so no new components are required to implement this control system. With this, the value of the heating source should be somewhat stable in regards to dissipated power, and should fix some of the heating problems the sensor has.

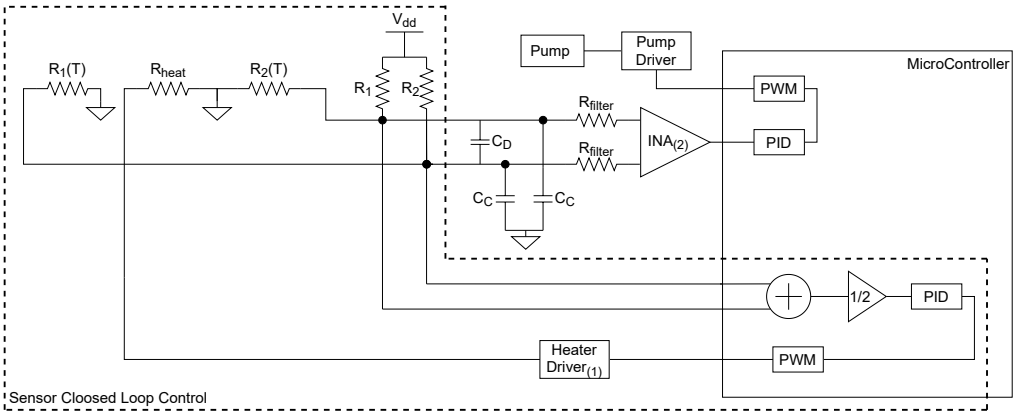


Figure 4.4: Diagram of the improved sensor circuit. (1) Heater driver implemented with same circuit as Pump driver. (2) Instrumentation Amplifier.

4.2.2 Circuit Noise

Since this circuit deals with very low voltage differences, it is important to do an analysis of the noise the components produce. The most important noise source is the amplifier and the voltage divider at its input. To calculate the noise it generates, the datasheet was consulted. At 25℃ and 1 kHz, for a gain of 5, the amplifier has an input voltage noise of 24 nV/√Hz, an output noise of 310 nV/√Hz, and a current noise of 100 fA/√Hz [12]. The thermal noise of the resistors at the input can be calculated with

$$e_{Nthermal} = 2\sqrt{(RkT)}, \tag{4.1}$$

where *R* is the resistor's value in Ohms, *k* is the boltzmann constant in J/K, and *T* is the temperature in Kelvin. The resistor's value in this case, will be the sum of the filter's resistor (7.5 kΩ) with the parallel of the voltage divider resistors. At 25℃, the thermistors used have a value of 100 kΩ, and the resistor used in the divider is of 180 kΩ. As such,

$$R = \frac{180 \times 100}{180 + 100} + 7.5 = 71.7857\ k\Omega. \tag{4.2}$$

With this value, the thermal noise can be calculated

$$e_{Nthermal} = 34.380\ nV/\sqrt{Hz}. \tag{4.3}$$

To calculate the current noise, the datasheet value is multiplied by the resistor's value,

$$e_{Nc} = 100 \times 10^{-15} \times 71785.7 = 7.17857\ nV/\sqrt{Hz}. \tag{4.4}$$

To calculate the total noise, the sum of the squares of the values needs to be done,

$$e_{Ntotal} = \sqrt{2 \times 34.380^2 + 2 \times 7.17857^2 + 24^2 + \left(\frac{310}{G}\right)^2}\ nV/\sqrt{Hz}. \tag{4.5}$$

The current and resistor thermal noise are multiplied by two due to the fact that they are repeated in each input of the amplifier. The output voltage noise is divided by the gain because its value decreases with gain. With this, depending on the gain, the total value of the noise is

$$\begin{cases} e_{Ntotal} = 82.988\ nV/\sqrt{Hz}, & with\ G = 5 \\ e_{Ntotal} = 63.277\ nV/\sqrt{Hz}, & with\ G = 10 \end{cases}. \tag{4.6}$$

This result, however, does not account for the noise generated by the reference. This is because the reference's datasheet does not offer a spectral noise value. The value it offers is a 3 μVp-p voltage noise, for frequency values between 0.1 and 10 Hz. As such, to account for this value, the noise value will again be calculated; now with the reference's noise value. To be able to sum the multiple noise values, it is necessary to convert all of them into Root Mean Square (RMS). Using a 99.9% conversion factor [14], the peak to peak value of the reference (3 μVp-p) is

$$e_{Reference} = \frac{3\ \mu}{6.6} = 454.5\ nV_{RMS}. \tag{4.7}$$

To obtain the RMS noise value of the amplifier, (4.5) can be used. By analysing the datasheet [12], it can be noted that the 1/f corner of the amplifier's noise spectral density is at a lower frequency than 10 Hz. As such, the calculated noise value is still valid at these frequency values. To obtain its RMS noise value, at a gain of 5,

$$e_{Ntotal} = 82.988 \times \sqrt{10} = 262.431\ nV_{RMS}. \tag{4.8}$$

To obtain the total noise, including the reference noise value,

$$e_{NtotalRMS} = \sqrt{(262.431 \times G)^2 + (454.5)^2} = \sqrt{(262.431 \times 5)^2 + (454.5)^2} = 1.38864\ \mu V_{RMS}. \tag{4.9}$$

This study of the circuit noise is an important factor when the measurements are of very small magnitude, and require great gain. This is not the case, since a gain of 5 proved to be sufficient, however, since a correct characterization of the sensor was not possible, this study is nonetheless important if it is proven that the measurements need a higher gain. However, it is possible to make a comparison using the step of the used ADC. The step would be of

$$\frac{1.5\ V}{1024} = 1.465\ mV.$$

This means that the calculated noise is much smaller than the step, and would not cause a major interference.

4.3 PID control - Step Response

To evaluate the performance of the PID controller, its step response was evaluated. Using a voltage divider formed by two 1 kΩ resistors and with the ADC reading the middle point between the two sections, and with the output from the PID controller being the voltage source of the divider, it's possible to simulate the response of the sensor. This simulation, however, assumes a linear operation, independent of all external factors, however, it can determine if the PID controller is working correctly or not. With the divider, and by changing the ideal value in the PID controller, it's possible to simulate the response of a sensor to multiple situations.

First the step responses to a change from minimum to maximum, and vice versa, are obtained. The responses can be seen in Figs. 4.6(a), 4.6(b), and 4.6(e).The first two were measured by having the ADC at the middle point of the voltage divider and changing the ideal value in the controller from zero to 1023 (maximum value for the 10 bit ADC, corresponds to about 3.3 V, since the voltage divider halves the voltage source value), and vice versa. The third one was measured by connecting the ADC ground and changing the ideal value to 1023. Fig 4.6(a) shows a step response of about 52 ms, Fig. 4.6(e) a step response of about 78 ms, and Fig. 4.6(b) a step response of about 69 ms. Figs 4.6(c) and 4.6(d) show the step response when changing from half the maximum value to the maximum value, and vice versa. The response has a peak of about 10-20 ms and then takes about 1 s to settle at the desired value.

Fig. 4.6(f) shows an outlier in the signal. It shows the usual small peaks due to the signal not being 100% filtered and then a bigger and smaller peak, which cause a certain noise in the signal. The bigger peak is the one that causes the smaller peak, since the voltage rises due to the bigger peak and is then compensated by the smaller peak. This abnormality could be caused by a bad value in the time constant, DT of the PID controller. Due to the time constant and the ADC frequency not being correctly chosen, it could cause the controller to be faster than the ADC, which would cause it to use an old value,

and act on it instead of the real value, causing a higher peak than necessary.

Since the system should not have abrupt changes, with the exception of a sudden total obstruction of a channel, the time response of the controller is acceptable. If a faster response was needed, the controller's constants could be changed accordingly. For example, an increase in the proportional and integral constants. The most important constant, however, would be the *DT* constant, which represents the amount of times the controller acts in a second, specially in regards to possibly correcting the peaks in the signal.

Attempting to implement a full PID controller instead of a simple PI controller could also be an improvement in the stability, and consequently the speed, of the controller.

The oscilloscope used in these results was a Tektronix TDS2012.

Fig. 4.5 shows the GUI displaying the information it receives from the microcontroller regarding the read values from the ADC and the PWM values being output to the system.

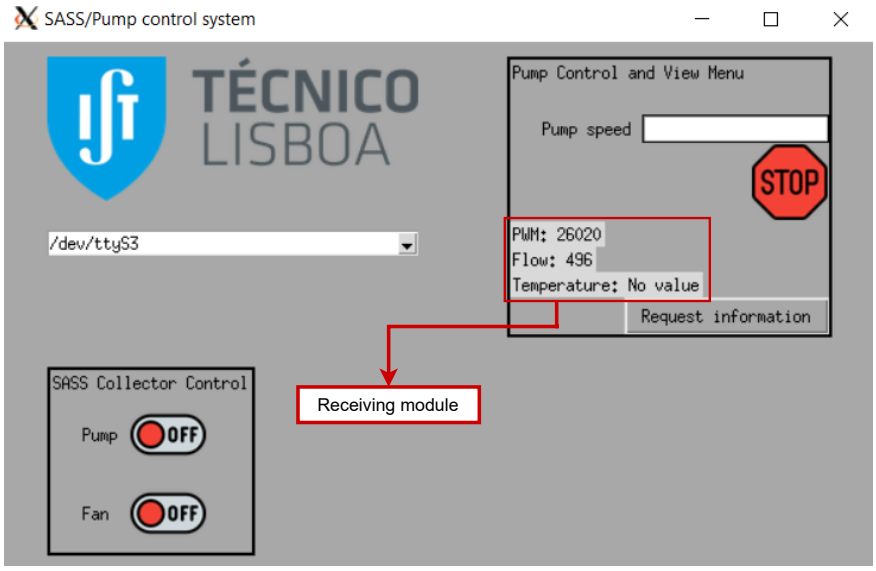
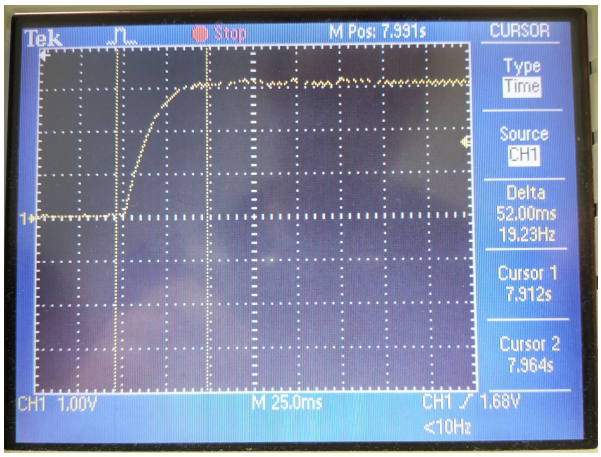
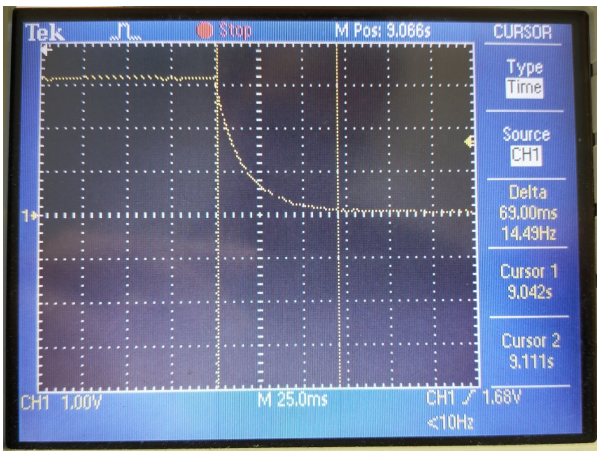


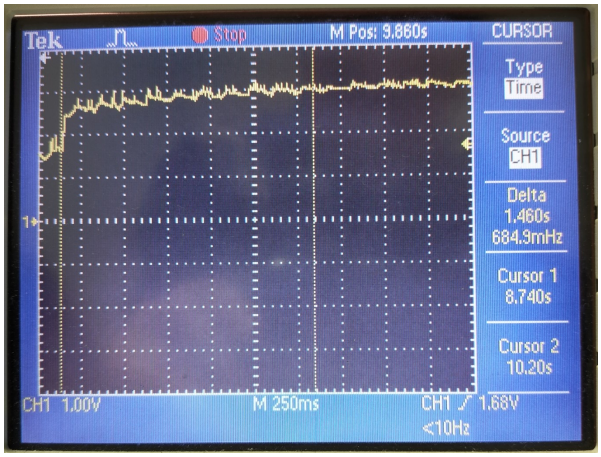
Figure 4.5: GUI program receiving the read data from the microcontroller.



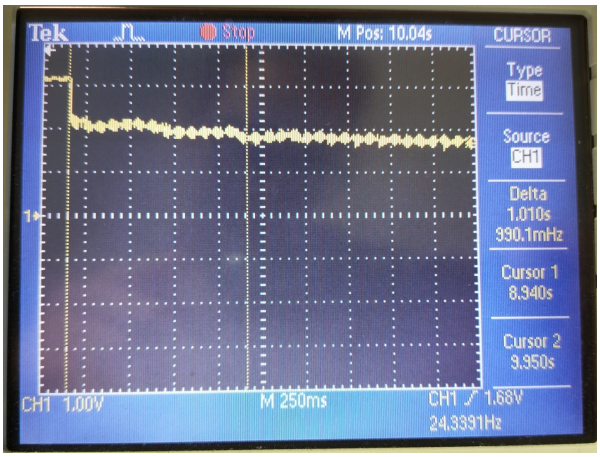
(a) From zero to max



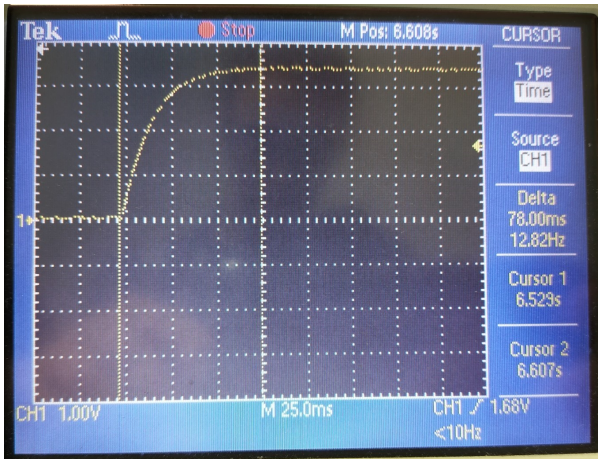
(b) From max to zero



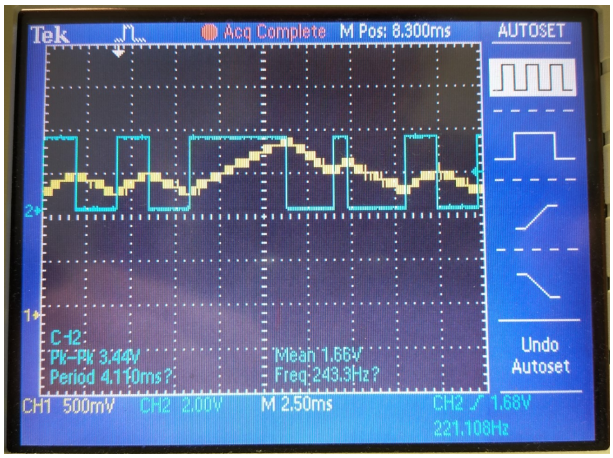
(c) From half to max value



(d) From max value to half



(e) ADC input to ground, zero to max



(f) PID unfiltered peaks and unstable peaks

Figure 4.6: Oscilloscope images of multiple step responses of the PID controller. Blue represents the unfiltered PWM signal and yellow represents the driver circuit's output signal.

5

Conclusions and Future Work

Contents

5.1 Conclusions

A real-time device capable of giving accurate results regarding the presence of pathogenic agents is a tool that could help society in various different sectors, such as industries where a clean environment is important, or public health safety. Efforts have been carried with the goal of developing a device capable of giving those real-time, on the field results, replacing the currently lengthy and troublesome process of collecting samples and transporting them to a lab for further analysis. To develop such a device, it's important to analyze methods to collect air samples and detect the pathogenic agents in the samples. This project focuses on developing a control system of the microfluidics pumps, regulating the flow rate of the stream that enters the cytometer and microfluidics components, as well as obtain feedback on the flow rate to verify that a continuous and controlled stream is maintained. To do so, methods for controlling the pump were analyzed, using PWM to create the voltage that controls the pump due to its ease to implement, low power consumption, and high resolution. Methods to sense the flow rate were also analyzed, the chosen method using the heat dispersion caused by a flowing fluid and thermistors to deduce the flow rate of a stream.

To further the development of this project, and to better integrate it in a real world scenario, some future work is necessary. Some problems found while working would need to be solved. The sensor's operation would need improvements to assure its correct behaviour. Some of the problems found were the rising of the ambient temperature of the liquid in the channels. This causes the resting operating point to shift over time, and no way to measure this shift, causing different values for the same flow rate, and even saturation in values previously inside the operating range. Another problem found is the uncontrolled heating source. Since the heating source does not have a control system to make sure it maintains the same temperature over time, it cause shifts in the values read. A fix to this problem is to use the average of the absolute value of the Negative Temperature Coefficient (NTC) thermistors and a PID controller to control the voltage going to the heating source resistor, and as such maintain a stable temperature throughout the measurements of the flow rate.

Some other suggestions to improve upon the sensor circuit would be to make the sensor integrated so it can be smaller and better suited for even smaller flows. Another suggestion is to use another microcontroller with higher definition ADCs in order to achieve a better resolution in the sensor.

The pump driver circuit's filtering could also be improved in order to achieve a better attenuation of the PWM square wavelenghts.

The software also requires improvements, such as optimization of the firmware for both microcontrollers. The implementation of a full PID controller, and optimization of all the components on the MSP430, as well as optimization of the code, specially the execution threads, of the ESP32, would ensure a smoother and quicker overall system. The GUI could also be upgraded to look more modern and accessible by virtue of better modules and libraries.

Furthermore, wireless functionalities regarding the air sampler should also be added, in order to fully convert to a completely wireless system.

Bibliography

[1] SASS 2300 wetted-wall air sampler. Accessed 23-Dec-2020. [Online]. Available: http://www.resrchintl.com/SASS_2300_air_sampler.html

[2] E. M. E. da Câmara Cunha, "Design of a compact aerosol system for controlled biocontaminations," Master's thesis, Instituto Superior Técnico, 2017.

[3] J. Loureiro, P. Z. Andrade, S. Cardoso, C. L. da Silva, J. M. Cabral, and P. P. Freitas, "Magnetoresistive chip cytometer," *Lab Chip*, vol. 11, pp. 2255–2261, 2011. [Online]. Available: http://dx.doi.org/10.1039/C0LC00324G

[4] TCS Micropumps. Accessed 10-Jan-2021. [Online]. Available: https://micropumps.co.uk/TCSM200range.htm

[5] M. Dybwad, G. Skogan, and J. M. Blatny, "Comparative testing and evaluation of nine different air samplers: End-to-end sampling efficiencies as specific performance measurements for bioaerosol applications," *Aerosol Science and Technology*, vol. 48, no. 3, pp. 282–295, 2014. [Online]. Available: https://doi.org/10.1080/02786826.2013.871501

[6] E. W. Saaski, C. C. Jung, and D. A. McCrae, "High efficiency wetted surface cyclonic air sampler," US Patent 6 532 835B1, Mar.18 2003.

[7] P. P. Freitas, R. Ferreira, and S. Cardoso, "Spintronic sensors," *Proceedings of the IEEE*, vol. 104, no. 10, pp. 1894–1918, Oct. 2016. [Online]. Available: https://doi.org/10.1109/jproc.2016.2578303

[8] L. E. Locascio, "Microfluidic mixing," *Analytical and Bioanalytical Chemistry*, vol. 379, no. 3, pp. 325–327, Jun 2004. [Online]. Available: https://doi.org/10.1007/s00216-004-2630-1

[9] C.-Y. Lee, C.-L. Chang, Y.-N. Wang, and L.-M. Fu, "Microfluidic mixing: A review," *International Journal of Molecular Sciences*, vol. 12, no. 5, pp. 3263–3287, 2011. [Online]. Available: https://www.mdpi.com/1422-0067/12/5/3263

[10] D. A. Czaplewski, B. R. Ilic, M. Zalalutdinov, W. L. Olbricht, A. T. Zehnder, H. G. Craighead, and T. A. Michalske, “A micromechanical flow sensor for microfluidic applications,” *Journal of Microelectromechanical Systems*, vol. 13, no. 4, pp. 576–585, 2004.

[11] T. L. Bergman, F. P. Incropera, D. P. DeWitt, and A. S. Lavine, *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2011.

[12] *Wide Supply Range, Rail-to-Rail Output Instrumentation Amplifier*, Analog Devices, 2009, rev. 0. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/AD8227.pdf

[13] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback control of dynamic systems*. Pearson, 2015.

[14] *Op Amp Noise Relationships: 1/f Noise, RMS Noise, and Equivalent Noise Bandwidth*, Analog Devices, 2009, mT-048. [Online]. Available: https://www.analog.com/media/en/training-seminars/tutorials/MT-048.pdf