Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

School of Engineering and Sciences

Optical flow sensor for droplet-based Lab-on-PCB devices.

A thesis presented by

Daniel Hugo Solano Teran.

Submitted to the
School of Engineering and Sciences
in partial fulfillment of the requirements for the degree of

Master of Science

In

Nanotechnology

Monterrey Nuevo León, December 1st, 2020

Dedication

I dedicate this work to my mother Roxana and my father Oscar, for all their patience, support, and love throughout all my life, I can never be thankful enough. Further to all my family members that were by my side in every little step in this journey, and for cheering me up during the lockdown. And last but not least, this is for my grandparents that are no longer with me Salome, Hugo, and Isaac.

Acknowledgements

First of all, I would like to express my sincere gratitude to the CONACYT scholarship (CVU: 970973), the CONACYT-OEA-AMEXCID 2019 scholarship programme, and Tecnologico the Monterrey university for all the support.

Further, I would like to thank my supervisor Ph.D. Sergio Camacho, for all the guidance, contribution, and support throughout the project. I am also thankful to the committee members Ph.D. Matias Vázquez and Ph.D. Antonio Luque for their thoughtful comments and recommendations on this dissertation.

To conclude, I cannot forget to thank my family and friends specially MSc. Kendra Corral, and MSc. Cinthia Ramirez for all the unconditional support in these very intense academic years.

Optical flow sensor for droplet-based Lab-on-PCB devices

by

Daniel Hugo Solano Teran

Abstract

Advancements on Lab-on-a-PCB devices nowadays focus on design goals such as **A**ffordable, **S**ensitive, **S**pecific, **U**ser-friendly, **R**apid and robust, **E**quipment-free, **D**eliverable to end-users (ASSURED) devices. However, most of these new systems present external equipment dependencies, complex set-up processes, low reproducibility factors, and intricate manufacturing processes. For many industries (medical, pharmaceutical, cosmetics), Lab-on-a-PCB devices are capable of characterizing multiphase systems such as cell-in-droplets identification, flow-phase characterization, and micromixing detection. Thus, this work presents a new optical droplet detector, employing common and cost-effective electronics components. The device consists of a fluid channel between a light-emitting diode (LED) and a photo-resistor (LDR), whose voltage variation is measured and then processed with an ARDUINO microcontroller. This new sensor can determine different multiphase flow properties such as velocity, flow, droplet lengths, and volume with high-speed throughput up to 1000 droplets per second. Furthermore, this sensor presents a modular electronic design that provides a simple calibration, high adaptability, and a standardized fabrication process. Therefore, it creates a cost-effective, portable, easy-to-fabricate, and plug-and-play environment for the alignment with the ASSURED criteria.

Droplet detection and characterization showed MRE values ranging from 2.4% up to 17%. The lowest MRE value was obtained using a two-phase flow system with water-in-air droplets at a sampling rate of 2.3 kHz for flow rates starting at 20 up to 425 µL/min. In contrast, the highest MRE value reported was under a three-phase flow system for dyed and pure water-in-air droplets at a 5 kHz sampling rate at a 250 µL/min flow rate.

List of Figures

Figure 1. Optical droplet detector operation principle. 7
Figure 2. Optical droplet detector ideal expected signals and timing diagrams. 8
Figure 3. Optical droplet detector final prototype. 9
Figure 4. Optical droplet sensor electrical schematic. 10
Figure 5. PCB Arduino shield. 12
Figure 6. Arduino shield PCB final design. 12
Figure 7. Sensor PCB (LED + LDR). 13
Figure 8. Sensor PCB final design. 13
Figure 9. Optical droplet detector assembly. 14
Figure 10. Optical droplet sensor assembly. 15
Figure 11. Calibration scale for droplets measurements. Each division represents 0.01 mm for a total length of 1 mm. 16
Figure 12. ImageJ calibration results. A) Summarized measurements and mean calculation in pixels. B) Pixel selection from the 1 mm calibration scale. 16
Figure 13. ImageJ droplet measurement. A) Summarized data and results in millimeters. B) Dyed-water droplet measured. 17
Figure 14. Arduino inputs and outputs connections. 18
Figure 15. Arduino IDE algorithm flow diagram. 19
Figure 16. MATLAB algorithm flow diagram. 21
Figure 17. Raw signal for a water in air droplet at 370 µl/min. 23
Figure 18. Savitzky-Golay filtering results (). 24
Figure 19. Shifted signal for Zero-crossing detection. 25
Figure 20. Local maxima and minima for a filtered signal. 26
Figure 21. Experimental set-up. 30
Figure 22. Three experimental water-in-air droplets and flow displacement. 31
Figure 23. Water-in-air raw signal recorded at 200 µL/min constant flow rate. 32
Figure 24. Filtered signal for a 200 µL/min constant flow rate. 33
Figure 25. Sensor's velocities and flow rates measurements vs. theoretical references 35
Figure 26. MRE in % per experiment for velocities and flow rate measurements. 36
Figure 27. Sensor's length and volume measurements vs. theoretical references. 37
Figure 28. MRE in % per experiment for length and volume measurements. 37
Figure 29. High-speed sampling rate droplets configuration. 38
Figure 30. Raw signal extracted at 670 µL/min constant flow rate. 38
Figure 31. Filtered signa for a 670 µL/min flow rate. 39
Figure 32. Sensor's velocities and flow rates measurements vs. theoretical

references for high-speed sampling rate. Red stars represent measurements for 70, 270, 470, 670 and 870 µL/min. Orange stars are measurements for 170, 370, 570 and 770 µL/min. 41
Figure 33. High-speed sampling measurements MREs. 42
Figure 34.Sensor's length and volume measurements vs. theoretical references for high-speeds sampling rate. 43
Figure 35. MRE values for measured lengths and volume using high-speed sampling rate. 44
Figure 36. Enhanced calibration process. 45
Figure 37. Dyed-water droplets for enhanced calibration testing. 46
Figure 38. Raw signal for dyed-water droplets recorded at 100 µL/min. 46
Figure 39. Filtered signal for dyed-water droplets recorded at 100 µL/min. 47
Figure 40. λ boxplot, IQR selection, and final fitted distribution. 48
Figure 41. Flow rate and velocities measurements for dyed-water droplets. 48
Figure 42. MRE values for velocities and flow rates measurements using dyed-water droplets. 49
Figure 43. Dyed-water droplets lengths and volume measurements. 49
Figure 44. MRE values for lengths and volume measurements using dyed-water droplets. 50
Figure 45. Random dyed-water droplets for calibration testing. 50
Figure 46. Three-phase flow experimental droplets. 51
Figure 47. Signal recording for three-phase flow experiment at 100µL/min constant flow rate. 52
Figure 48. Three-phase flow filtered signal at 100 µL/min constant flow rate. 53
Figure 49. λ boxplot, IQR selection, and final fitted distribution for three-phase flow experiments. 54
Figure 50. Three-phase flow velocities and flow rate measurements. A) Expected vs measured values. B) MRE values in % for each experiment. .54
Figure 51. Three-phase lengths and volume measurements. A) Expected vs measured values. B) MRE values in % for each experiment. 55
Figure 52. Pure-water and dyed-water droplets for calibration testing. 56

List of Tables

Table 1. State-of-the-art review. 4
Table 2. ATmega328P ADC Prescaler Selections [29]. 20
Table 3. ATmega328P ADC register A bits [29]. 20
Table 4. Bootstrap sampling technique. 29
Table 5. Calibration experiments for standard ADC configuration. 32
Table 6. λ results for the calibration process signals. 33
Table 7. Calibration experiments for standard high-speed sampling rate. 39
Table 8. λ results for high-speed sampling configurations 40
Table 9. Dyed-water droplets λ calculated values. 47
Table 10. Calibration process testing results for dyed-water droplets. 51
Table 11.λ values calculated for three-phase flow experiments. 53
Table 12. Three-phase flow calibration testing results. 56

Contents

Abstract vi

List of Figures vii

List of Tables ix

Introduction 1
1.1. Problem definition 2
1.2. Hypothesis and research questions 2
1.3. Objectives 3
1.4. State-of-the-art 3
1.5. Solution overview 6

Methodology 10
2.1 Electronic design. 10
2.2 PCB fabrication. 14
2.3 Image processing for droplet measurement. 15
2.4 Programming. 17
2.4.1 Arduino IDE 17
2.4.2 MATLAB 21
2.5 Statistical methods. 28
2.6 Assembly. 29

Results 31
3.1 Standard ADC sampling rate. 31
3.1.1 Calibration process. 32
3.1.2 Tests. 34
3.2 High speed ADC sampling rate. 38
3.2.1 Calibration process. 39
3.2.2 Tests. 41

Calibration process and experimental improvements. 45
4.1 Experimental tests with dyed-water droplets 45
4.2 Three-phase flow experimental tests. 51

Conclusions and future work. 58
5.1 Future work. 60

Appendix A 61

Appendix B 63

Appendix C 71

References..72

Curriculum Vitae 75

Chapter 1

Introduction

Microfluidics has been defined as the science and technology for the analysis of very small quantities of samples and reagents (10^{-9} to 10^{-18} Liters), carrying out separations and detections with high sensitivity and resolutions at low cost and short times [1], [2]. Thus, research and work in this area it has uncovered applications in numerous fields, such as biology, physics, biotechnology and pharmacy; and has contributed as key component for the development of integrated and miniaturized systems, called micro total analytical systems (µTAS) or lab-on-a-chip devices (LOC) [3]. However, most of these devices have faced difficulties achieving real-life adoption, standardized fabrication processes, and mass production [4].

The use of printed circuit boards (PCB) for lab-on-a-PCB (LOP)devices as a fabrication technique for electronic and microfluidic integration is one promising approach for the overcome of these problematics with high impact in system integration and mass fabrication with low costs, due to its well standardized and mature manufacturing process [3], [4].

Furthermore, the fabrication of these systems has widely adopted a new criterion, in order to match **A**ffordable, **S**ensitive, **S**pecific, **U**ser-friendly, **R**apid and robust, **E**quipment-free, **D**eliverable to end-users, called ASSURED criteria as presented in [5]–[9]. This criterion was developed by The World Health Organization Sexually Transmitted Diseases Diagnostics Initiative (SDI) as a benchmark for the improvement of sexually transmitted infections (STIs) diagnosis [10].

Most of these devices' fundamental principle relies on the detection and characterization of two-phase flows (gas-liquid, liquid-liquid, or nanoparticle-liquid), known as droplet microfluidics, with excellent characteristics for chemical, biomedical, biological, and industrial applications [11]. This technique reduces complicated experiments in laboratories by reducing its size, minimizing reagent consumptions, and increasing the repeatability of experiments conducted with a single device. Nonetheless, challenges still in portability due to dependencies of external devices [12]–[14], cost-effectiveness, and complex fabrication processes in which controlled environment is required [14], [15].

Hence this research aims to develop a fully integrated cost-effective and portable droplet sensor for a Lab-on-a-PCB device with decentralized settings. Reducing the requirement of external equipment and human efforts into measuring processes, generating an automated Plug-and-Play environment pursuing the ASSURED criteria. This work is composed of five chapters, an introduction, methodology, results, experimental improvements, and conclusions.

1.1. Problem definition

This work aims to develop an optical sensor for droplet characterization and counting, to overcome the difficulties of portability, cost-effectiveness, user-friendliness, and autonomy on lab-on-PCB devices. Its main characteristic is a high-speed throughput, up to 1000 droplets per second for multiple multiphase flow fluids capable of detecting and characterizing dispersed droplets using standard electronic components. In addition, to achieve the ASSURED criteria, the device design was thought towards its energetic consumption, and portability with a working area of 38 by 55 mm, including peripherals for the Arduino connection using its USB source for powering the system or an external battery if it is required. The use of standard electronic components reduces costs and ease the fabrication process by integrating standard PCB fabrication techniques without the need for a sophisticated laboratory offering an overall fabrication cost under 20 USD. The resulting device has a full autonomous functioning that does not require any human intervention during its measurements or detection processes; therefore, it does not require a complex training process delivering an easy to use environment.

1.2. Hypothesis and research questions

This work hypothesizes that using optoelectronics combined with an Arduino UNO microcontroller as the fundamental principle for droplet detection and characterization can meet the requirements for a precise and accurate droplet-based system, besides, to overcome the existing gaps in achieving the ASSURED criteria. Therefore the following research questions arise:

- What are the key factors that limit the sensor measurements and throughput?
- Is the Arduino UNO capable of high-speed processing sensor signals?
- How often does the sensor need a calibration process?
- Is it possible for the sensor to detect and characterize optically similar fluid-phases in one experiment?
- What are the advantages and disadvantages of choosing a standard PCB fabrication process for the sensor?
- What is the impact of using optoelectronics and standard PCB fabrication techniques on the achievement of the ASSURED criteria?

Therefore this work proposes a portable, programmable, easy-to-fabricate, user-friendly, autonomous, and affordable optical droplet sensor for lab-on-P.C.B. devices to enhance the ASSURED criteria completion. The use of standard electronics components reduces its fabrication costs and complexity alongside the well-known Arduino-UNO microcontroller to develop a user-friendly interface for programmable applications using an in-situ powering source.

The sensor's fundamental principle is the voltage variation of a photoresistor aligned to the microchannel and the light source (LED). It is then processed using the microcontroller to detect and calculate the droplet's length, velocity, flow rate, and volume. Furthermore, this work aims to demonstrate that with the use of standard components, the sensor output is similar to more elaborated sensors in precision and accuracy, also in detection rates of droplets per second.

1.3. Objectives

This work's main objective is to develop a portable, programmable, easy-to-fabricate, user-friendly, autonomous, and affordable optical droplet sensor for characterization and quantification of multi-phase fluids in lab-on-a-PCB devices, enhancing the ASSURED criteria completion. (High-speed)

The following specific objectives are the guidelines for the proposed optical sensor realization:

- To design the electronic system and peripherals using an LED and a photoresistor as the primary sensor's functioning principle.
- To fabricate the sensor using standard electronic components and PCB fabrication process.
- To program the Arduino UNO microcontroller to capture the device's signal response and user inputs using the Arduino IDE software.
- To develop a system for data reading and signal processing using MATLAB software.
- To integrate the hardware and software into a final device.

1.4. State-of-the-art

Throughout recent years droplet-based microfluidics for LOC devices has gained attraction in the research field due to their promising results in numerous fields, such as biology, physics, biotechnology, and pharmacy [12]. Therefore many state-of-the-art devices have arisen, such as the U-Chip illustrating a fabrication methodology for a cost-effective PCB microfluidic device with USB source as the driving unit [16]. Another approach is a PNA-based Lab-on-PCB diagnostic platform for rapid and high sensitivity DNA quantification for biosensing applications [7]. Although these few devices are excellent choices for Lab-on-PCB devices, there is still a gap in achieving ASSURED criteria. The U-chip has not been tested on disease detection, and the lifetime of the electrodes is ten minutes as a result of degradation. In contrast, the device purposed in [7] has portability issues due to external systems' requirements as syringe pumps and source power. However, a more in-depth study for state-of-the-art devices in droplet-based systems is shown in the following Table 1, in which various measuring techniques for droplet detection are compared by their features, flow phases, and final specifications.

Table 1. State-of-the-art review.

Reference	Technique	Measurements	Flow phases	Parameters
[14]	Capacitive sensor employing interdigital electrodes (IDEs) and a dielectric film.	Droplet velocity. Constant that allows to distinguish different flow patterns (Do). Droplet diameter distribution.	Continuous phase: deionized water. Dispersed phase: olive oil.	Microchannel: 150 μm height and 800 μm width. Droplet flow: Q_O=0.001 mL/ min, Q_W =0.005 mL/ min. Short slug flow: Q_O=0.003 mL/ min, Q_W=0.03 mL/ min. Long slug flow: Q_O=0.004 mL/ min, Q_W=0.06 mL/min No marginal values for size nor velocity. Sampling frequency: 32.2 Hz.
[17]	Optical diagnosis through back-light imaging. Diffused back-illumination DBI. Camera: Photron SA5. Global field lens: Nikon Sigma 24-70 mm 31 kfps. Near field lens: DistaMax lens 150 kfps.	Droplet diameter distribution. Droplet velocity distribution.	Urea-water solution spray.	Minimum diameter detected: 21.7 μm. Droplets over 2000 μm were neglected.
[15]	Microfluidic impedance cytometry (MIC). In contact Field's Metal (icFM) microelectrodes Microchannels for electrode deposition with melted Fermium (FM).	Cellular electrical impedance Droplet detection. Droplet length Cell-in-droplet quantification.	Continuous phase: fluorinated oil Dispersed phase: erythrocyte cell suspension in 1×Phosphate-buffered saline (PBS).	Microchannel: 20 μm height and 20 μm width. Droplet volume: ≈ 150 pL. Sampling rate: 7200 Hz.
[18]	Micro-particle image velocimetry (μPIV, optical method to characterize microscale flows by tracking the motion of particles seeded in the flow).	Droplet velocity for different droplet sizes in the channel Droplet velocity fields. Effect of viscosity ratio on flow topology.	Continuous phases: Silicon oil, HFE7500. Dispersed phases: Water/glycerol, pure water. Four different fluid mixtures.	Microchannel: 150 μm height and 200 μm width. Flowrates: 10 μL/h to 5000 μL/h. Variation coefficient of velocities: 2% ± 4%.

		Interfacial tension. Capillary number.		
[13]	Optical detection: Light source and photo diode. Laser source (Laser diode, $\lambda = 635\ \text{nm}$). CCD camera and microscope system.	Droplet detection. Droplet length Droplet shape.	Continuous phase: oil. Dispersed phase: pure DI- water.	Microchannel: 340 μm height and 340 μm width. Flowrates: 10 μL/h to 100 μL/h.
[19]	Droplet-based microfluidic chip for 6-thioguanine detection in human serum using surface-enhanced Raman scattering (SERS).	SERS scatterings plots, from 60 mono-dispersed droplets within 10s (8 different sampling positions).	Continuous phase: silicone oil (Span 80). Dispersed phase: AuNPs sol (gold colloid), B-R buffer and sample solution.	Microchannel: 90 μm height and 140 μm width Flowrates: silicon oil, gold colloid, B-R buffer and analyte 4.0 μL⁄ min, 3.0 μL⁄min, 2.5 μL⁄min and 4.0 μL⁄ min respectively. Droplet length: 770 μm. Droplet volume: 9.7 nL.
[12]	CMOS camera attached to a microfluidic device. Automatic image processing algorithm based on differencing-based motion detection.	Droplet detection. Droplet counting.	Continuous phase: high purity mineral oil. Dispersed phase: phosphate-buffered saline and 8% polyethylene glycol diacrylate pre-polymer.	Microchannel: 100 μm height and 200 μm width Flowrates: from 0.3 μL⁄ min and 1.3 μL⁄ min max. CMOS rate: 64 images per 4 sec. Max throughput: 12 droplets⁄ sec with 8% error rate.

A capacitive sensor is presented in [14], where interdigital electrodes called IDEs, and a dielectric film are its operating principle for droplet detection. This sensor can measure droplet velocity, a constant to distinguish differences in flow patterns called Do, and finally, it calculates a distribution for droplet diameters. Although this work describes promising results in droplet detection and characterization, its dependency on an external power source arises an obstacle towards portability.

Another approach is presented in [17], using optical diagnosis through back-light imaging with a high-speed capture camera setup. This technique characterizes droplets expelled from a urea-water solution in engine exhaust-like conditions focusing on their diameter and velocity. The minimum diameter detected was 21.7 μm, with the exception that measurements over 2000 μm were neglected. Even though this system offers a different approach, it is important to mention that similar setups with high-speed cameras are used to complement and validate droplet-based devices such as described in [14].

Microfluidic impedance cytometry MIC provides promising results in cell detection and classification, as described in (Panwar 2018). The device uses in-contact field's Fermium microelectrodes called icFM, which allows the system to detect and measure droplets using electrical impedance. This method's advantage is that it can detect cells or particles trapped inside of droplets within volumes of 150 pL at 10 Hz of frequency. However, this device's main obstacle is its portability due to external dependencies such as the driving unit and a power source.

Surface-enhanced Raman scattering (SERS), combined with a droplet-based microfluidic chip for 6-thioguanine (6-TG, a cancer drug) detection in human serum, is presented in [19]. This device takes the advantages of a PDMS microfluidic device for micro mixing a solution of 6-TG, a Britton-Robinson buffer, and gold nanoparticles, which are the substrate for the SERS tests. The chemical adsorption between 6-TG and gold nanoparticles takes place inside the droplet, and it is measured with SERS in six different places throughout a serpentine microchannel. The setup allows analyzing and comparing 6-TG reactions at the beginning and the end of the channel. Therefore this system describes excellent results towards specific droplet characterization. Nonetheless, it becomes a system that requires a laboratory with a SERS machine to replicate experiments.

The last device presented in Table 1, is the integration of a CMOS camera with a microfluidic device that detects and counts droplets using an automatic image processing algorithm based on motion detection [12]. This system was tested using flowrates from 0.3 µL/min up to 1.3 µL/min with a maximum throughput of 12 droplets per second, and an error rate of 8.9%. Although this setup offers a more portable environment with just the dependency of a syringe pump, the rate detection is low compared to robust systems that can detect droplets at a higher flow rate [18].

As a result of the study of these state-of-the-art devices, the most common difficulty found is portability due to dependencies to powering sources, microscopes, cameras, and syringe pumps, which hinders their functionality outside of controlled laboratories or environments. Furthermore, the electrodes fabrication processes for sensing droplets such as in [14], [15] need controlled laboratories. However, as they are built within the microfluidic device, this limits their reproducibility. Therefore, as many of these systems have successfully achieved a specific application, it is essential to develop new design and fabrication methodologies on behalf of achieving the desired ASSURED criteria for LOC systems.

1.5. Solution overview

This work describes an optical detector for counting and characterizing droplets inside microchannels. Its operation principle relies on the alignment of a light source, the microchannel, and a light sensor. The light source is a light-emitting-diode (LED), and the light sensor is a photoresistor or light-dependent resistor (LDR). This arrangement develops an environment where whenever a droplet crosses through this "sensing area," it varies the read signal from the LDR, indicating a change of phase in the channel. Furthermore, the

LDR needs to be insulated from the surroundings to minimize external noise appearance into the reading signal. The setup for this sensor is described as shown in Figure 1.

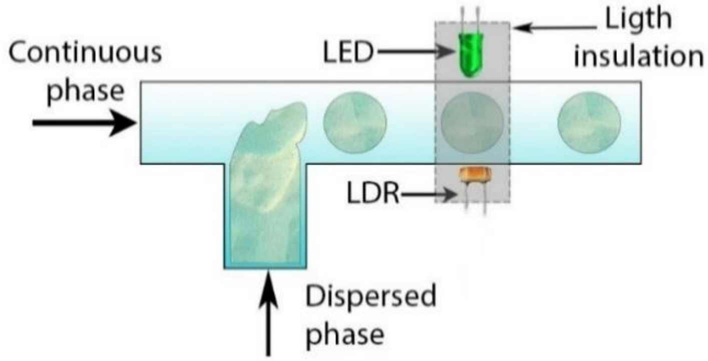


Figure 1. Optical droplet detector operation principle.

LDR signals are recorded using the Arduino UNO microcontroller to calculate the system features: droplet detection, counting, velocity, length, flow rate, and volume. The signal processing and algorithms for calculations are treated using MATLAB [20] for signal and results visualization.

The timing diagram for expected signals and behavior of the sensor's operation principle is described in Figure 2, in which the voltages for the continuous and dispersed phase are expressed by V_c and V_d, respectively. Besides, the phase changing, and the detection regions represent the time required for a droplet to enter and exit the sensing area.

As shown in Figure 2, the signal response changes whenever the dispersed phase (droplet) is clearer or darker than the continuous phase. Figure 2-A describes a case in which droplets are more transparent than the continuous-phase allowing more light to be sensed by the photoresistor; thus, V_d decreases compared to V_c, resulting in a positive phase transition voltage (ΔV). On the other hand, the signal of Figure 2-B shows a darker dispersed phase, in which V_d is greater than V_c; therefore, the phase transition voltage is negative.

Therefore, for developing this system the first step is to design the electrical circuit using the USB connection from the Arduino microcontroller for powering the device. Furthermore, the fabrication of the printed circuit boards was done using the ferric chloride etching technique.

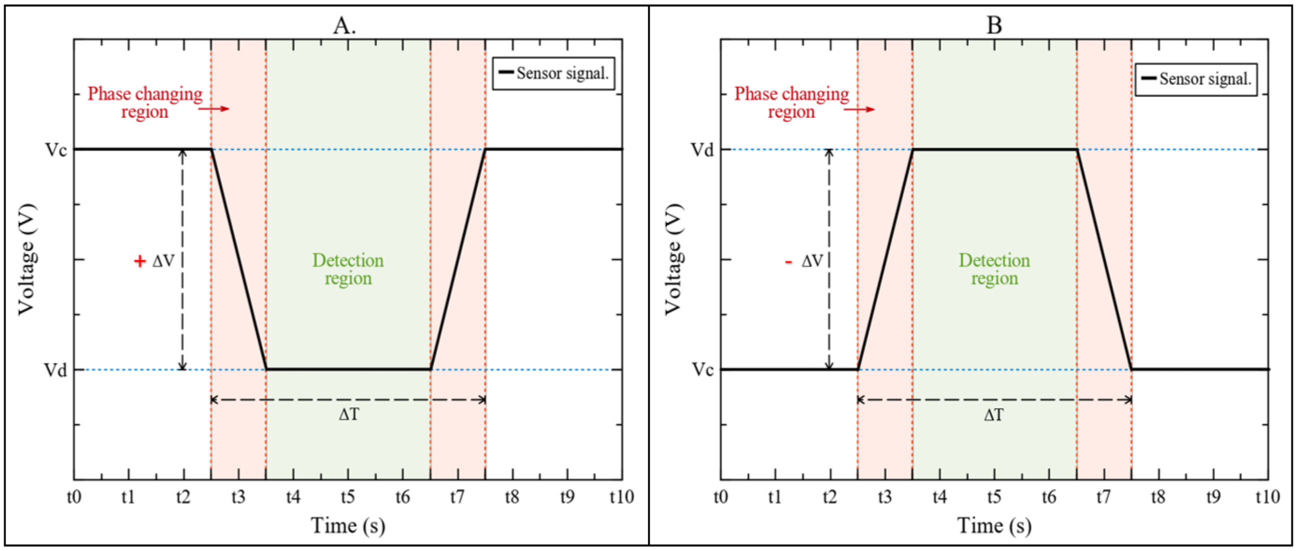


Figure 2. Optical droplet detector ideal expected signals and timing diagrams.

After the fabrication process, the programming stages take place for developing the Arduino UNO and MATLAB algorithms to read, record, store, process, and calculate the data. The microcontroller was programmed using Arduino IDE [21], focusing on its analog to digital converter transferring the recorded data from the LDR to the MATLAB environment using the USB connection. Inside the MATLAB environment, the data is reconstructed in a voltage vs. time signal using a filter to reduce possible noise extracted from the raw data, afterward using an algorithm to identify local maxima and minima, the calibration or calculation processes can begin accordingly.

The latest step is to combine the software and hardware into one system in conjunction with the microcontroller. The final prototype is presented in Figure 3, in which the electronic design, the microcontroller, and a microchannel are unified into one device. The figure also describes the alignment of the sensor's insulated components (LED and LDR) with a microchannel. The final device shows two fundamental circuit boards, one for the main sensor components and another for the integration and compatibility with Arduino UNO. This arrangement allows an enhanced maintenance environment and the opportunity to develop new designs for the principal sensor with reduced microfluidic channels.

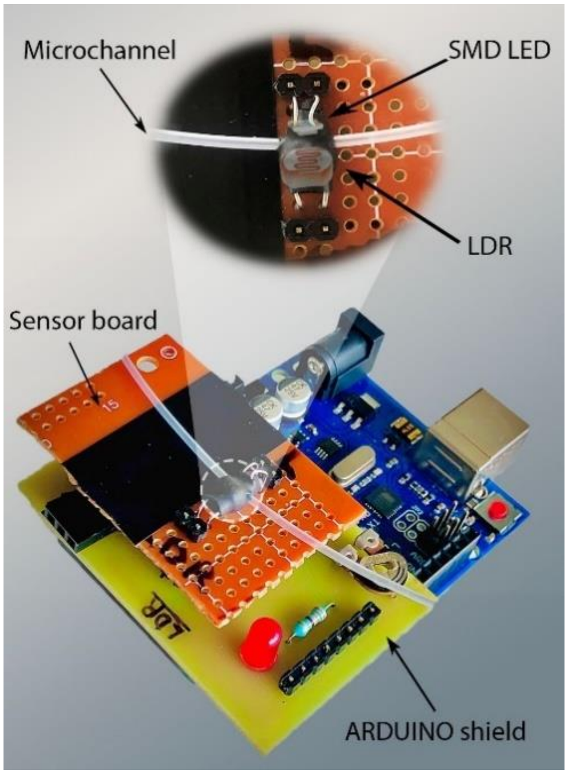


Figure 3. Optical droplet detector final prototype.

Chapter 2

Methodology

Throughout this section, we describe and analyze the process and techniques used to develop the optical droplet senor for lab-on-a-PCB devices. These methods involve electronic design and fabrication, Arduino microcontroller [22] and MATLAB [20] programming, statistical methods, and a final assembly.

The microcontroller programming section focuses on the ports and user interface needed to read, save, and transfer the photoresistor signals through the USB connection. On the other hand, MATLAB algorithms receive the microcontroller data and calculate the desired droplet features such as length, velocity, volume, and flow rate.

The statistical methods implemented was designed to study a variable behavior which its distribution is unknow and it is the key component for the sensor's calibration process.

2.1 Electronic design.

This sensor's operation principle is the variation of the light intensity in multi-phase microfluidic flow, generated from a light source and a detector, which in this case is an LED and LDR, respectively. Therefore, an electronic PCB design was developed using Eagle software [23] in order to connect and adjust the required components and peripherals to a final PCB. The circuit's power supply is the five volts delivered by the Arduino USB connection. Thus the electronic schematic is classified into four linked circuits, as shown in Figure 4.

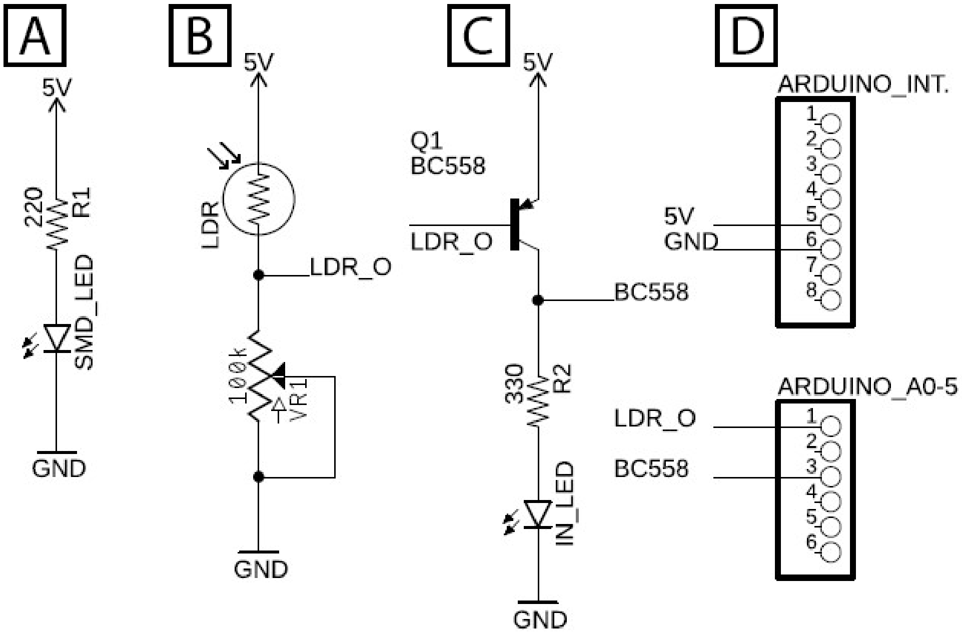


Figure 4. Optical droplet sensor electrical schematic.

Figure 4-A describes a voltage divider for turning on the sensor's light source. To calculate the resistor value R1 the Ohm’s law was used in conjunction with the forward voltage (V_F), and current (I_F) values extracted from the LED datasheet as shown in (1) in which (V_S) stands for the source voltage in this case 5 V, and R is the resistance value, (R1) in this case. The LED used in this sensor is a Surface Mount Device (SMD) 1206 green LED with a V_F value of 2 V and an I_F of 20 mA [24].

$$R=\frac{V_S-V_F}{I_F} \tag{1}$$

Replacing the values in (1) the resistor R1 final value is 150 Ω. However, to ensure that the LED does not work on its limit region, we choose a 220 Ω 0.25-Watt resistor [25], as shown in Figure 4-A.

Figure 4-B describes the LDR circuit. Similar to Figure 4-A, this circuit uses a voltage divider configuration with a 10 MΩ photoresistor [26] to sense light variations and a 100 kΩ variable resistor for calibration processes. The potentiometer adjusts the voltage of the LDR delivered to the next stage. The values used in the experiments for the best signal reading and interaction with the next stage are in the region of 75 to 90 kΩ.

The electronic schematic's in Figure 4-C shows a transistor switching circuit, which allows a BC558 bipolar junction transistor (BJT) with a PNP configuration [27] to behave as a switch, turning on a standard 5 mm LED [28] whenever the voltage between the Emitter and Base pins is above 0.7 volts called the saturation region. Furthermore, the transistor's voltage output from the Collector pin is connected to peripherals pins that allow compatibility with the Arduino microcontroller.

Similarly to R1, (1) was used to calculate the R2 resistance for the IN_LED. Whenever the transistor is in the saturation region, 5 V is generated in its Collector. The standard 5mm red LED used in this circuit has similar characteristics as the SMD LED from part A: 10 mA for its I_F, and 2 V for V_F. Therefore the R2 resistance calculated value using (1) with the values mentioned before is 300 Ω. Nonetheless, to protect the IN_LED for high current peaks, we decided to use a 330 Ω 0.25-Watt resistance [25] for R2, as shown in Figure 4-C.

The variation of VR1 (Figure 4-C) affects the voltages of the flow phases and controls the transistor on stage C, which at low voltages, the LED will not turn ON. It is essential to check the potentiometer position whenever repeated experiments are required to reduce errors in the calibration process and the final measurements.

After the electronic design, the printed circuit boards (PCBs) were designed using the same software as the electronic schematic (Eagle), considering the dimensions that the use of the ferric chloride etching fabrication technique can achieve. The final PCB design divided the electronic schematic into two boards to separate the sensing area (microfluidic channel, LED, and LDR) with the rest of the components.

First, the Arduino interface PCB connection was designed, called Arduino Shield. This circuit board contains all the device's electronic components except the LED and LDR from the primary sensor. This design used Arduino datasheet [22] specifications to locate pins for the PCB and microcontroller interaction, in addition to specific connectors for the main sensor PCB conjunction, as shown in Figure 5. The final design dimensions are 57.15 mm in length and 38.10 mm in width.

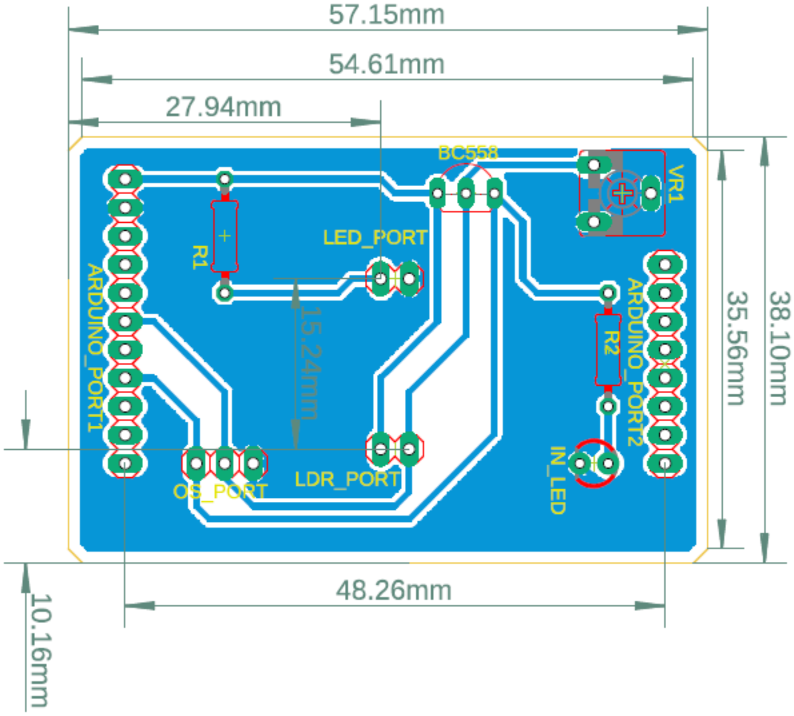


Figure 5. PCB Arduino shield.

After the design process was finished, the Arduino interface PCB 3D-design was implemented using Fusion 360 software [29], described in Figure 6 to visualize the required results after the fabrication process.

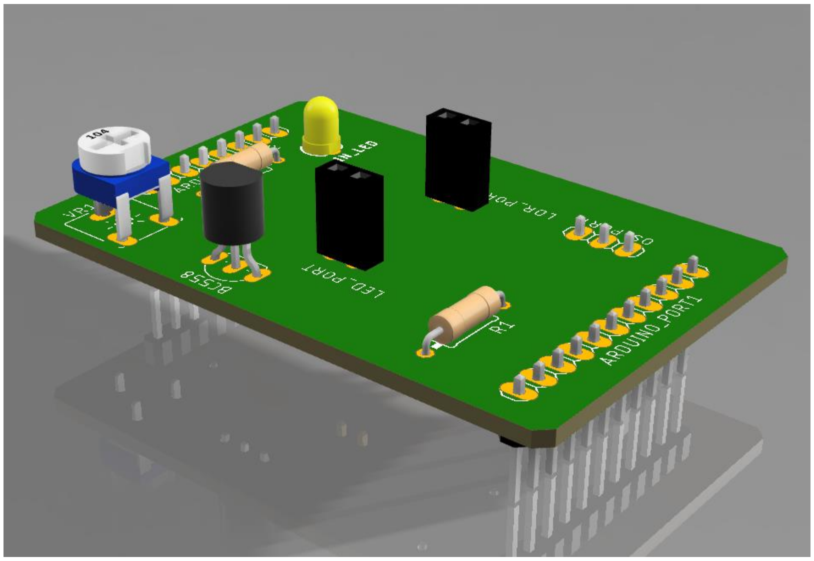


Figure 6. Arduino shield PCB final design.

Figure 7 describes the PCB designed for the sensing area (SMD LED and LDR). This design allows the circuit board to detach from the Arduino shield connection to ease maintenance and work with future strategies using the same pins to include a microchannel in the PCB. This board's size is 57.15 mm length and 25.40 mm width, as shown in the figure.

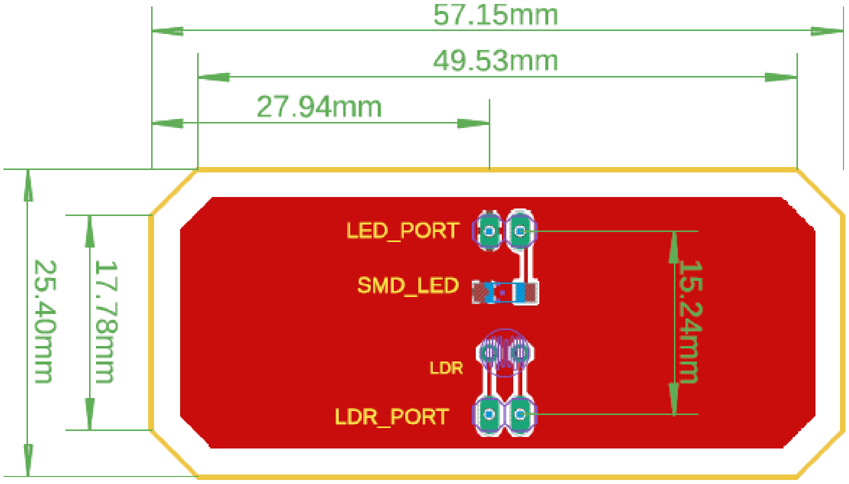


Figure 7. Sensor PCB (LED + LDR).

Similarly to the Arduino shield, this PCB was 3D implemented using Fusion 360, as shown in Figure 8 This figure describes a desired detachable board in which future work can develop microchannel devices and integrate them into the sensor without difficulties. Although this PCB shows a desire position for the LDR and SMD LED, the final application must have an additional microchannel and the insulation coating for reducing noise readings.

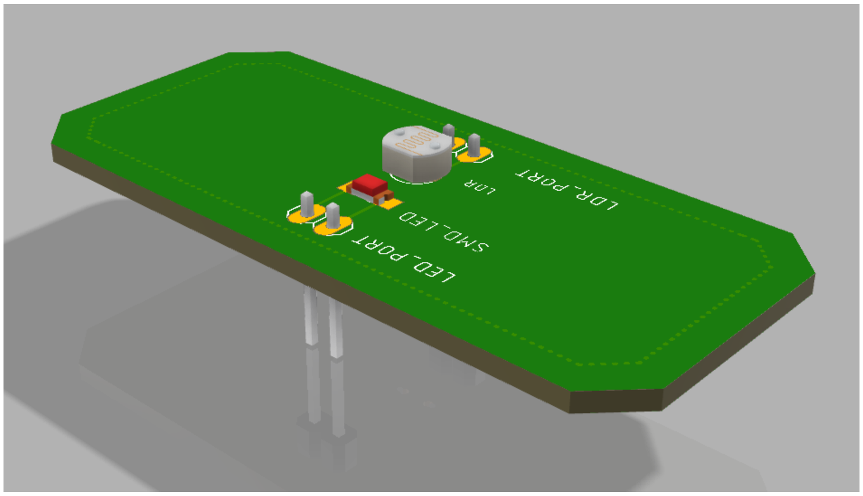


Figure 8. Sensor PCB final design.

Figure 9 describes a final assembly of the PCBs designed and the Arduino microcontroller. This figure illustrates the form and shape that the device will have after the fabrication process and the interactions between the designed circuit boards and the Arduino.

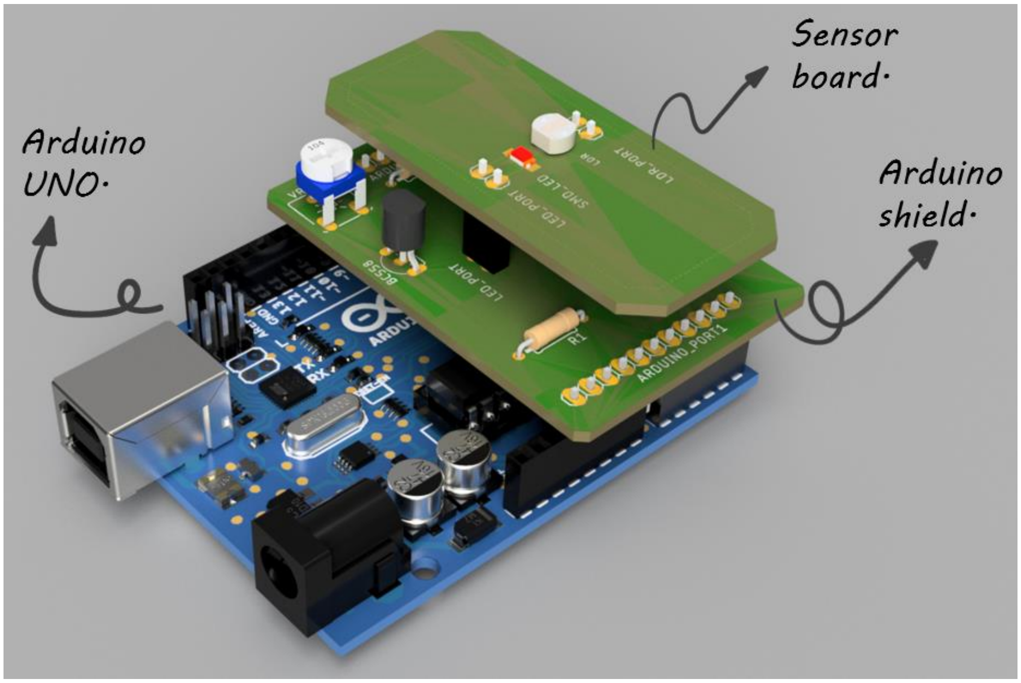


Figure 9. Optical droplet detector assembly.

2.2 PCB fabrication.

To fabricate the printed circuit boards designed in the previous section, we used the ferric chloride etching technique, which does not require controlled environments or sophisticated devices to manufacture PCBs. This technique is the most effective in terms of cost-effectiveness, requirements, and time-consuming for simple one layered PCBs. Nevertheless, it is possible to fabricate multiple layers and complex boards, yet the procedures can increase its difficulty and depends on the technician's experience.

The materials required for the Arduino shield's fabrication from Figure 6 and the sensor board from Figure 7 are the following:

- Single-sided copper-clad laminated PCB board.
- Electrical components from de designs.
- Soldering iron kit.
- Driller.
- Ferric chloride.
- PCB toner transfer paper.
- Iron.

The first step in the fabrication is to print the designs into the transfer paper using a toner printer and cut the PCB board to the designed sizes using a handsaw. The PCBs were then cleaned using a polisher adapter for the drill and washed with isopropyl alcohol before transferring the tracks into the copper layer. The PCB layout is then aligned to the copper side of the cleaned boards to apply pressure and controlled heat with the iron into the paper until the toner adheres in the copper. Once the paper residuals were removed from the PCBs using water, the boards were immersed into ferric chloride for chemical etching of the copper that is not covered with the transferred toner. The final step is to remove the toner from the final PCB using acetone.

Finally, the circuit boards were drilled using electronic design specifications, and then the components were soldered into the board with the soldering iron kit. The final assembled device is shown in Figure 10 with the integration of the Arduino UNO.

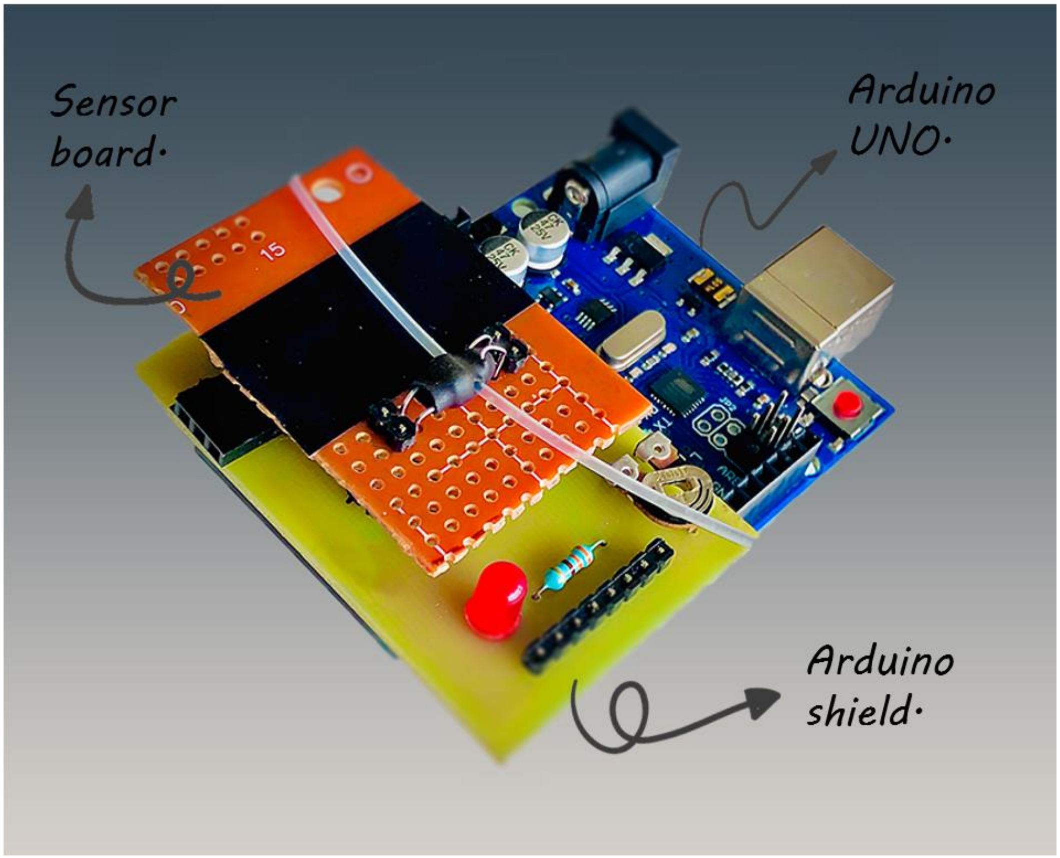


Figure 10. Optical droplet sensor assembly.

2.3 Image processing for droplet measurement.

In order to characterize the droplets before each experiment to compare results obtained with the proposed device, we used images from a microscope along with a calibration scale of 1 mm with 0.01 mm divisions, as shown in Figure 11.

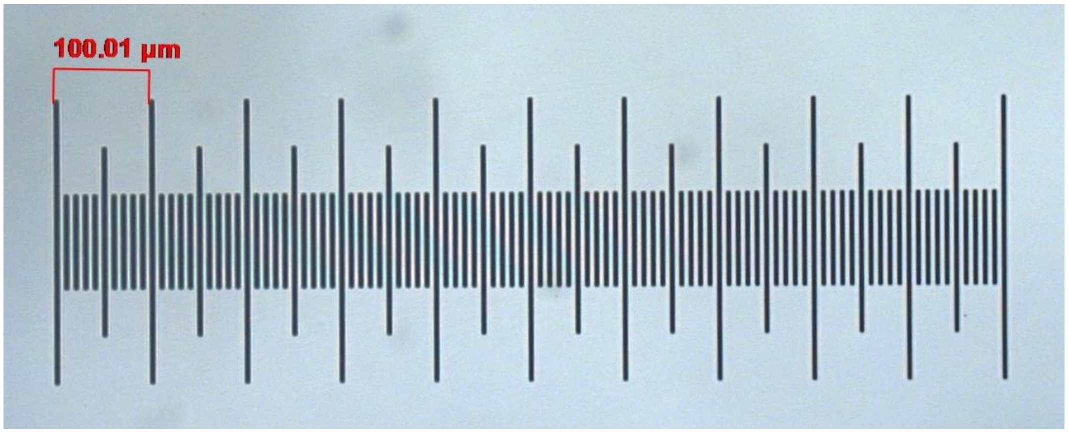
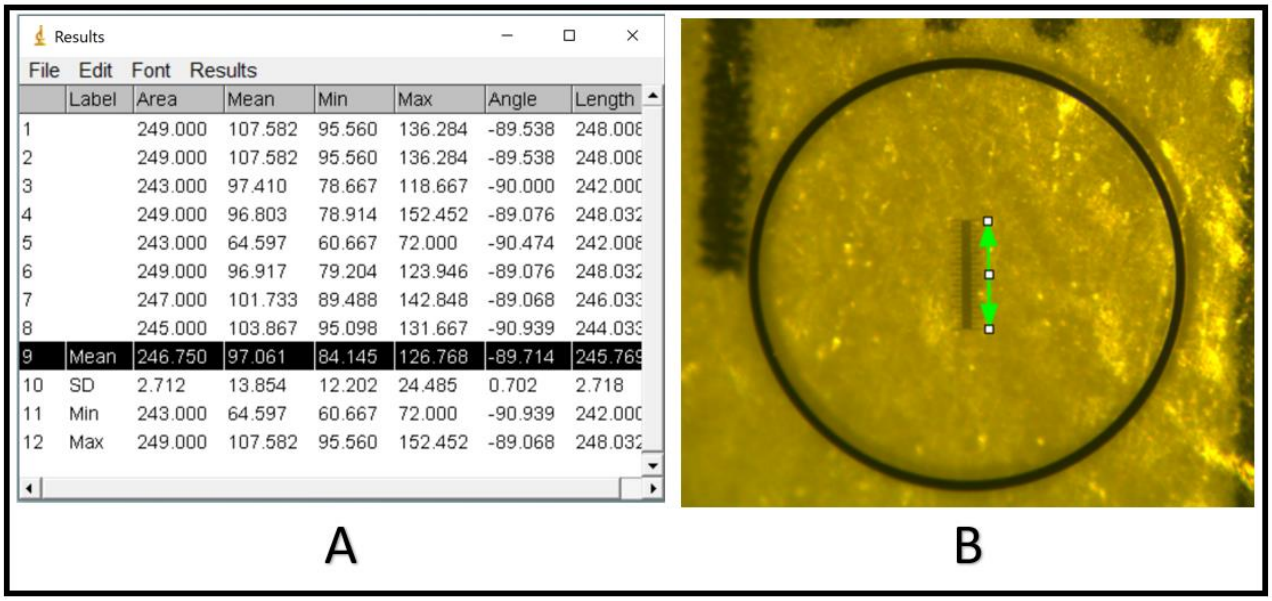


Figure 11. Calibration scale for droplets measurements. Each division represents 0.01 mm for a total length of 1 mm.

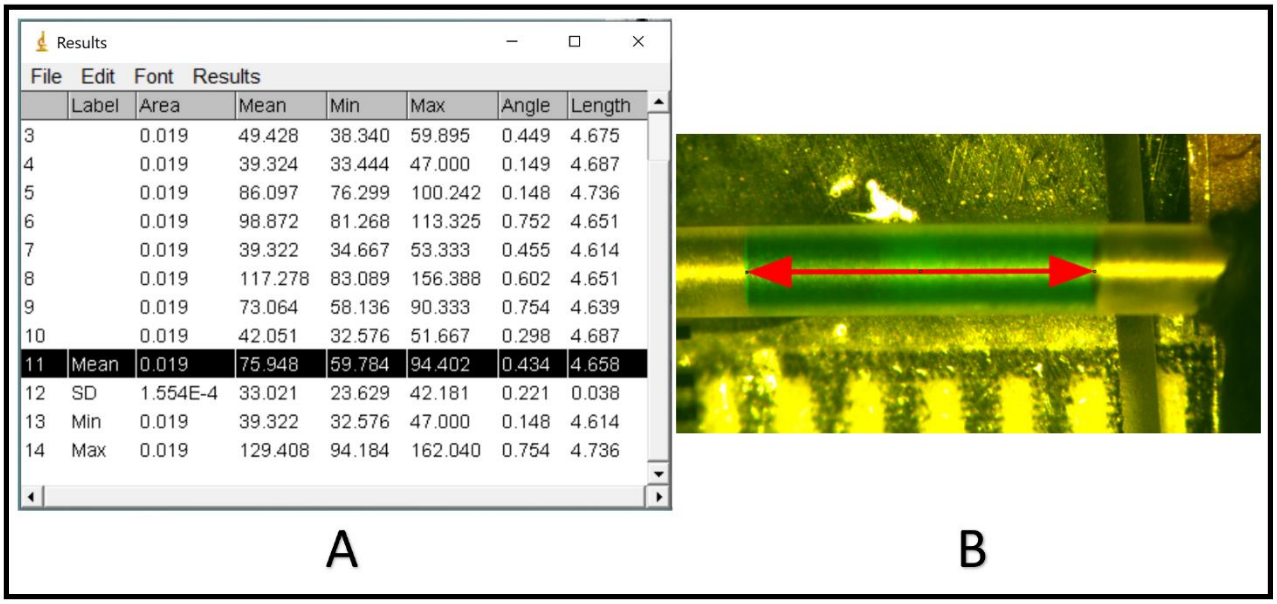
Once the images were extracted, we used ImageJ software (ref) for droplet length measurements. The first step is to calibrate the number of pixels for the known scale. Several measures, 20 approximately, were taken to obtain a mean value for the pixels to reduce the errors. This process is described in Figure 12, in which section A describes eight measures and the final mean value for the number of pixels (highlighted in black) , taken from the scale presented in section B.



	Label	Area	Mean	Min	Max	Angle	Length
1		249.000	107.582	95.560	136.284	-89.538	248.008
2		249.000	107.582	95.560	136.284	-89.538	248.008
3		243.000	97.410	78.667	118.667	-90.000	242.000
4		249.000	96.803	78.914	152.452	-89.076	248.032
5		243.000	64.597	60.667	72.000	-90.474	242.008
6		249.000	96.917	79.204	123.946	-89.076	248.032
7		247.000	101.733	89.488	142.848	-89.068	246.033
8		245.000	103.867	95.098	131.667	-90.939	244.033
9	Mean	246.750	97.061	84.145	126.768	-89.714	245.769
10	SD	2.712	13.854	12.202	24.485	0.702	2.718
11	Min	243.000	64.597	60.667	72.000	-90.939	242.000
12	Max	249.000	107.582	95.560	152.452	-89.068	248.032

Figure 12. ImageJ calibration results. A) Summarized measurements and mean calculation in pixels. B) Pixel selection from the 1 mm calibration scale.

After the ratio of pixels per distance is set in the program, the process for measuring the droplet length follows a similar approach, taking several measurements with ImageJ in order to obtain a final mean value in millimeters with its standard deviation. Figure 13, presents the measurement procedure for a green droplet in the sensor. The mean value obtained after ten length samples is 4.658 ± 0.038 mm, as described in section A, highlighted in black. Section B shows the dyed droplet used for measurements.



	Label	Area	Mean	Min	Max	Angle	Length
3		0.019	49.428	38.340	59.895	0.449	4.675
4		0.019	39.324	33.444	47.000	0.149	4.687
5		0.019	86.097	76.299	100.242	0.148	4.736
6		0.019	98.872	81.268	113.325	0.752	4.651
7		0.019	39.322	34.667	53.333	0.455	4.614
8		0.019	117.278	83.089	156.388	0.602	4.651
9		0.019	73.064	58.136	90.333	0.754	4.639
10		0.019	42.051	32.576	51.667	0.298	4.687
11	Mean	0.019	75.948	59.784	94.402	0.434	4.658
12	SD	1.554E-4	33.021	23.629	42.181	0.221	0.038
13	Min	0.019	39.322	32.576	47.000	0.148	4.614
14	Max	0.019	129.408	94.184	162.040	0.754	4.736

Figure 13. ImageJ droplet measurement. A) Summarized data and results in millimeters. B) Dyed-water droplet measured.

2.4 Programming.

This section is divided into two main parts, Arduino IDE [21] and MATLAB [20] programming, due to their algorithms objective and platforms are different. However, they complement each other to produce the final system. Arduino programming essentially focuses on the photoresistor voltage reading and recording controlled by the user. On the other hand, MATLAB collects the Arduino data and concentrates on signal processing and calculations for the desired droplet characteristics.

2.4.1 Arduino IDE.

This section describes the algorithm used to read the data from the LDR in addition to enable an environment in which the user can control when to start/stop the signal recording. These features have been programmed into the Arduino microcontroller ATmega328P [30] using the Arduino IDE software [21].

This algorithm's workflow is first to configure the microcontroller using its 10-bit analog to digital converter (ADC) pin A2 to read the BC558 collector output [27]. After the configuration, the user can use push buttons connected to the digital pins 1 and 6 to begin or stop the signal recording turning ON a recording indicator red LED [28] connected to pin 10. The structure for all the inputs and outputs of the microcontroller is shown in Figure 14.

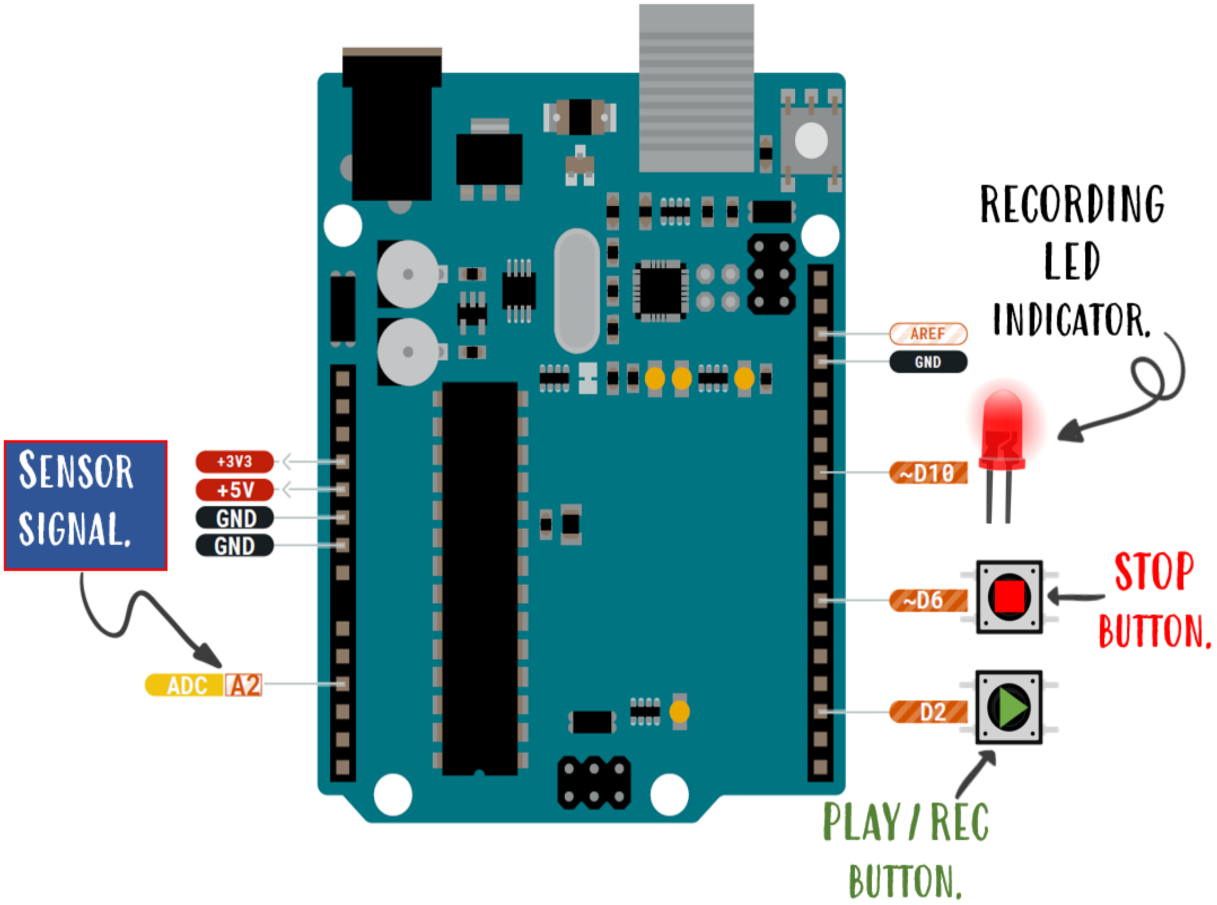


Figure 14. Arduino inputs and outputs connections.

Therefore, to illustrate the Arduino IDE program, we designed a flow diagram presented in Figure 15. The first step is to set up the input/output (I/O) pins, which are the indicator LED and the STOP and REC buttons connected to the pins 10, 2, and 6. Then the ADC set up for the ATmega328P takes place in order to achieve a high-speed throughput. Finally, the program starts a loop process. When the user presses the REC button, the LED recording indicator turns on, and the sensor starts recording and printing the signal on the Arduino's IDE serial monitor until the Stop button is pressed. When the recording process is stopped, the program prints the number of samples taken from the recording and calculates its frequency, then waits until the user starts a new recording.

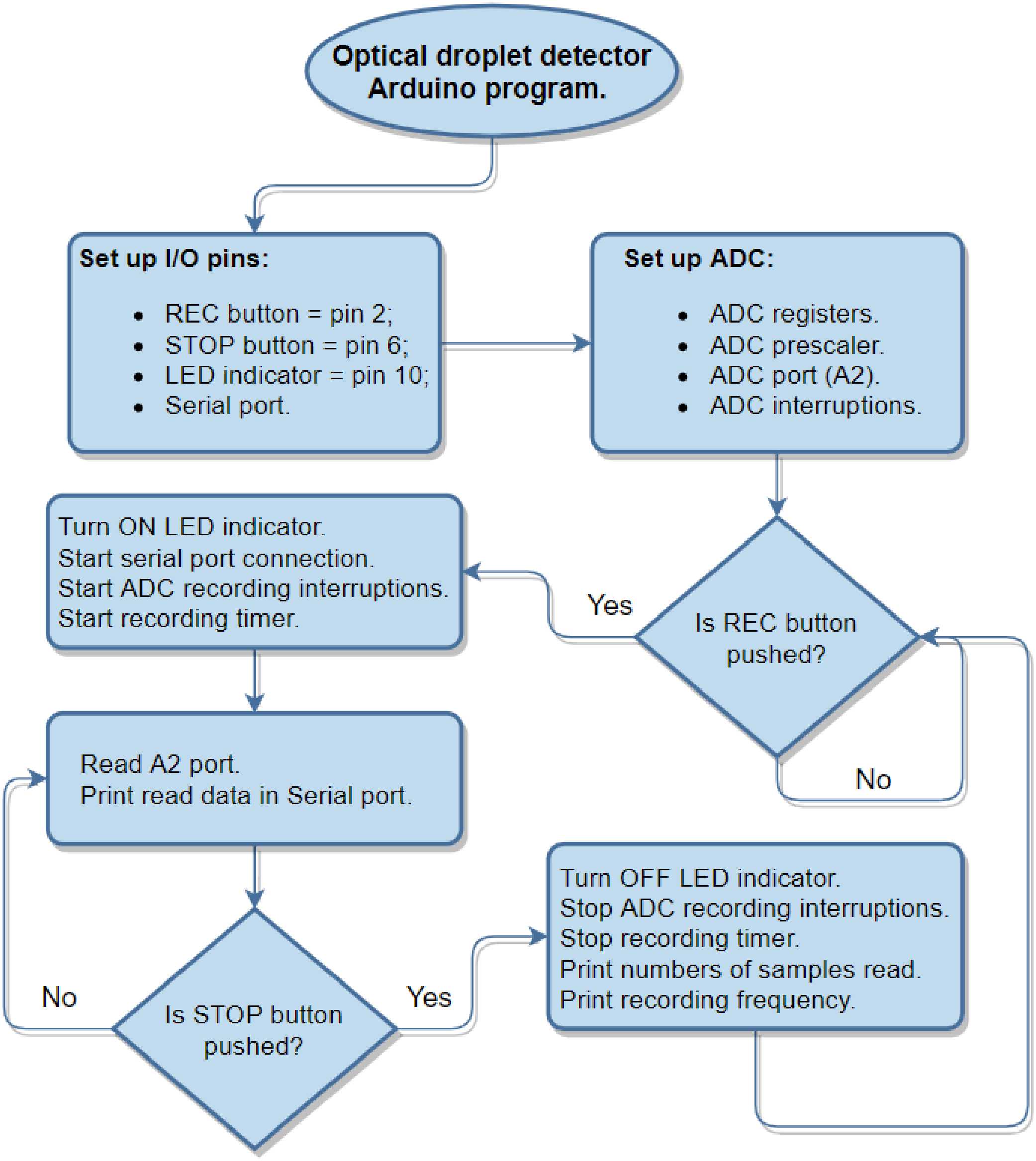


Figure 15. Arduino IDE algorithm flow diagram.

The ATmega328P ADC has a vital function in the sensor's throughput due to its sampling rate frequency. The ADC standard setup uses a 128-division factor (prescaler) between the system clock frequency and the ADC input clock, and it takes 13 ADC clock cycles to finish a conversion [30]. Therefore, we used (2) to calculate the maximum microcontroller sampling rate (F_S).

$$F_S = \left(\frac{\text{ADC clock}}{\text{prescaler}}\right) \div \text{Conversion cycles} \tag{2}$$

Thus, replacing the previously mentioned values, the maximum F_S is 9615.38 Hz. However, since the microcontroller is processing additional I/O data (buttons and digital pins), the sampling rate using standard configurations was 2300 Hz, approximately tested utilizing the algorithm.

Although we thought this sampling frequency was enough to achieve a high-speed throughput, it was revealed (discussed in the Results section) that the final rate was around 400 droplets per second.

To increase these results, we modified the ADC registers to read the analog port with Auto-triggered interruptions and change the prescaler to 8 for a higher sampling frequency speed. To change the Prescaler, we used the ATmega328P datasheet for changing the ADC register bits (APDS0, APDS1, and APDS2), as shown in Table 2.

Table 2. ATmega328P ADC Prescaler Selections [30].

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

To enable the Auto-triggered mode, interruptions, and the ADC itself, some bits of the register A of the ADC needs configuration, as shown in Table 3. The Auto Trigger mode is enabled by changing the bit 5 (ADATE) of the register A to 1. Similarly, the interruptions are enabled by the bit 3 (ADIE), and the ADC itself is controlled by the bit 7 (ADEN).

Table 3. ATmega328P ADC register A bits [30].

Bit	7	6	5	4	3	2	1	0	
(0x7A)	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/ Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Therefore, using (2) with an 8 Prescaler and bits 5, 3, and 7 of the ADC register A changed to 1, the Arduino's maximum sampling frequency is 153.8 kHz. However, testing the results with the algorithm, the final sampling frequency obtained is approximately 5000 Hz increasing the sensor's throughput up to 1000 droplets per second. The final Arduino IDE code is described in Appendix A.

2.4.2 MATLAB.

The algorithms programmed in this section read, store, and filter the data obtained from the Arduino to calculate in-flow-droplet properties such as velocity, flow rate, volume, and length using MATLAB software [20]. Therefore, Figure 16 describes the workflow for this section. The first step is to read, store, and convert the data using the serial monitor connection between Arduino and the computer. Then the data is reconstructed, filtered, and plotted to examine its response. Afterward, the program detects the signal's local maxima and minima to begin a calibration process, calculating a constant called "interfacial distance" represented by “λ”. Once the calibration process is done, λ is used to calculate the in-flow-droplet features.

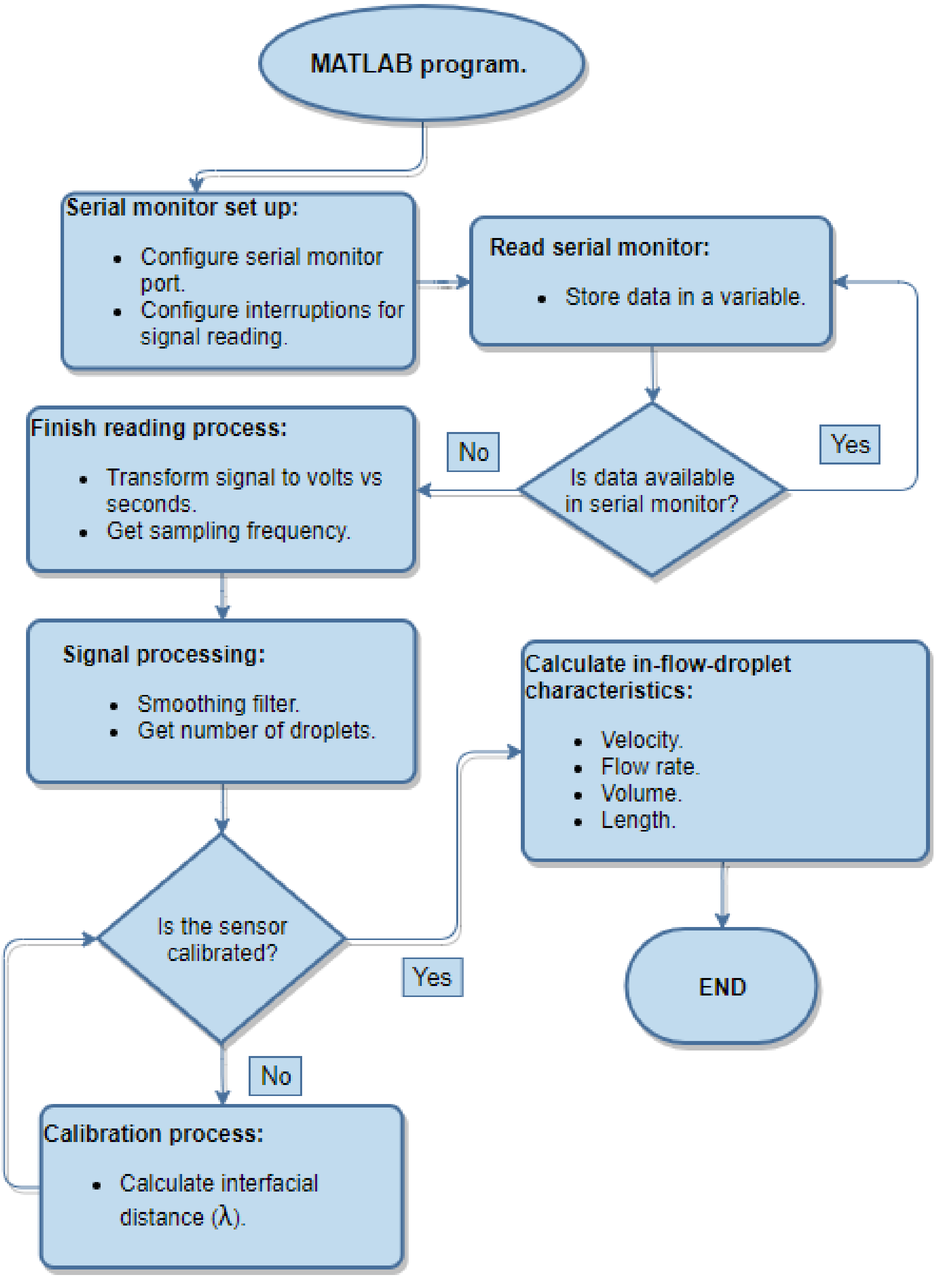

MATLAB program.
Serial monitor set up:
Configure serial monitor port.
Configure interruptions for signal reading.
Read serial monitor:
Store data in a variable.
Is data available in serial monitor?
Yes
No
Finish reading process:
Transform signal to volts vs seconds.
Get sampling frequency.
Signal processing:
Smoothing filter.
Get number of droplets.
Is the sensor calibrated?
Yes
No
Calibration process:
Calculate interfacial distance (λ).
Calculate in-flow-droplet characteristics:
Velocity.
Flow rate.
Volume.
Length.
END

Figure 16. MATLAB algorithm flow diagram.

The workflow presented in Figure 16 describes complex algorithms for signal processing, analysis, and statistical methods, which could overload the ARDUINO microcontroller if integrated into the same processor and impact the sensor's high-speed throughput directly. Therefore, we selected the MATLAB environment to read, analyze, and process the signal obtained due to its high adaptability to build a stand-alone program that could run on any computer without any MATLAB license. The alternative of using a high-speed microcontroller, such as the Intel Edison, Raspberry PI, among others, to embed the entire system in one platform could be a possible solution; however, it will increase fabrication costs and maintenance complexity.

2.4.2.1 Serial monitor set up.

This algorithm's serial monitor configuration was designed using the "serialport ()" MATLAB function, using the COM4 port and 250,000 baud-rate, since the maximum sampling frequency is 153,800 Hz for the Arduino's algorithm as discussed in the previous section. Whenever the recording process is finished, since the Arduino uses a 10-bit ADC converter, each sample value varies from 0 to 1024, describing 0 and 5 volts, respectively. Therefore, we used equation (3) to transform the acquired data from bits to volts following the ATmega328P datasheet [30]. D_B represents the value in bits extracted from the Arduino, V_{REF} is the ADC voltage reference, which is 5 volts, and V_{IN} describes the result expressed in volts.

$$V_{IN}=\frac{D_B * V_{REF}}{1024} \tag{3}$$

To calculate the samples' time vector, we used the inverse of the sampling frequency extracted from the Arduino algorithm obtaining the rate in seconds, thus generate an equally separated vector starting at zero seconds until the last sample. Figure 17 describes a raw signal in volts vs. seconds extracted from a random water-in-air droplet at 370 µL/min.

As shown in Figure 17, the raw signal extracted from the Arduino denotes noise compared with the ideal signal presented in Figure 2, due to the Arduino's sampling rate, resolution, and external factors. Therefore, we used a Savitzky-Golay smoothing filter [31] to reduce distortions.

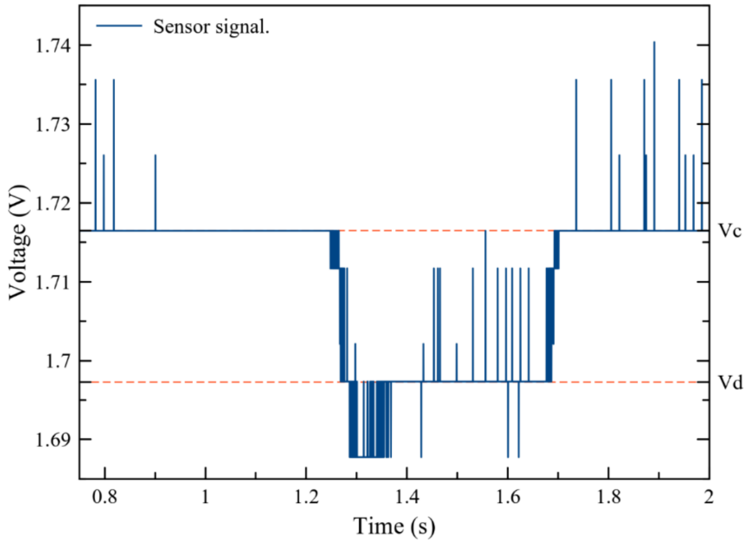


Figure 17. Raw signal for a water in air droplet at 370 µl/min.

2.4.2.2 Signal filtering.

The Savitzky-Golay filter change each point of the signal using a combination of the nearby values contained in a specific sized frame called "window" centered at the point. The filter's basis uses the moving average algorithm, as shown in equation (4), in which a (k) represents the number of data points to the left and right of the given point (x_s), and (L) represents the window length, which is: L=2k +1 [32].

$$x_s \rightarrow \hat{x}_s = \frac{1}{L}\sum_{r=-k}^{k} x_{s+r} \tag{4}$$

Therefore, to generalize this idea, Savitzky-Golay filters in MATLAB use a least-squares fitting with an *n*th-order polynomial through the window's signal values. The fitted polynomial curve's central point is taken as the new smoothed data point, for a given point x_s as shown in (5).

$$\begin{bmatrix} x_{s-k} \\ \vdots \\ x_{s-1} \\ x_s \\ x_{s+1} \\ \vdots \\ x_{s+k} \end{bmatrix} = \begin{bmatrix} b_0 + b_1(t_s - k\Delta t) + b_2(t_s - k\Delta t)^2 + \cdots + b_n(t_s - k\Delta t)^n \\ \vdots \\ b_0 + b_1(t_s - 1\Delta t) + b_2(t_s - 1\Delta t)^2 + \cdots + b_n(t_s - 1\Delta t)^n \\ b_0 + b_1(t_s - 0\Delta t) + b_2(t_s - 0\Delta t)^2 + \cdots + b_n(t_s - 0\Delta t)^n \\ b_0 + b_1(t_s + 1\Delta t) + b_2(t_s + 1\Delta t)^2 + \cdots + b_n(t_s + 1\Delta t)^n \\ \vdots \\ b_0 + b_1(t_s + k\Delta t) + b_2(t_s + k\Delta t)^2 + \cdots + b_n(t_s + k\Delta t)^n \end{bmatrix}$$
$$= \begin{bmatrix} a_0 + a_1(-k) + a_2(-k)^2 + \cdots + a_n(-k)^n \\ \vdots \\ a_0 + a_1(-1) + a_2(-1)^2 + \cdots + a_n(-1)^n \\ a_0 + a_1(0) + a_2(0)^2 + \cdots + a_n(0)^n \\ a_0 + a_1(1) + a_2(1)^2 + \cdots + a_n(1)^n \\ \vdots \\ a_0 + a_1(+k) + a_2(+k)^2 + \cdots + a_n(+k)^n \end{bmatrix} \tag{5}$$

Consequently, we designed a Savitzky-Golay smoothing filter applying a 0.15 smoothing factor [33], a scalar ranging from 0 to 1, adjusting the filter smoothing level implemented in MATLAB software. The smoothing results increase as the value gets close to 1, and values near 0 produce a less smoothed output. Additionally, this factor adjusts the fixed size in which the signal data is divided for each iteration “window”. Consequently, the best smoothing factor tested using flow rates from 70 to 670 µl/min was 0.15 which was implemented in the algorithm. The results from the designed filter are shown in Figure 18 which describes the filtered signal from Figure 17, validating a greater similarity compared with the ideal expected signals from Figure 2. However the smoothing factor remains as a constant that the user can modify in order to achieve the best fitting results, which will be the minimum value required to obtain a noise-free signal, in order to reduce loss of signal information.

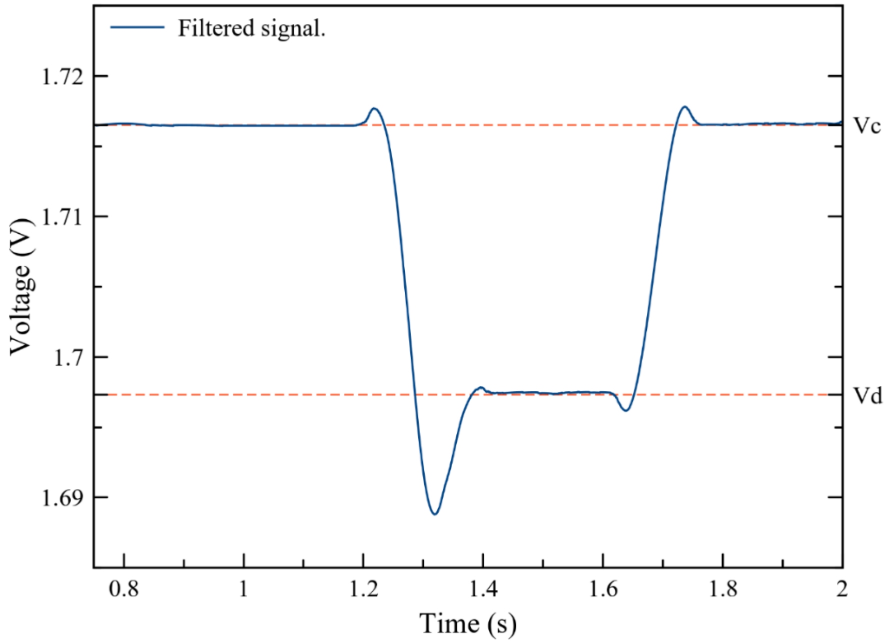


Figure 18. Savitzky-Golay filtering results ().

2.4.2.3 Number of droplets calculation.

The next step is to count the number of droplets after the signal is filtered. Therefore, the algorithm shifts the y-axis (Voltage) subtracting the mean value of the signal to have the zero volts in approximately in the middle of the peaks, as shown in Figure 19. Thus, it counts the number of zero-crossing and divides the result by two, due to the signal for a single droplet must cross zero twice.

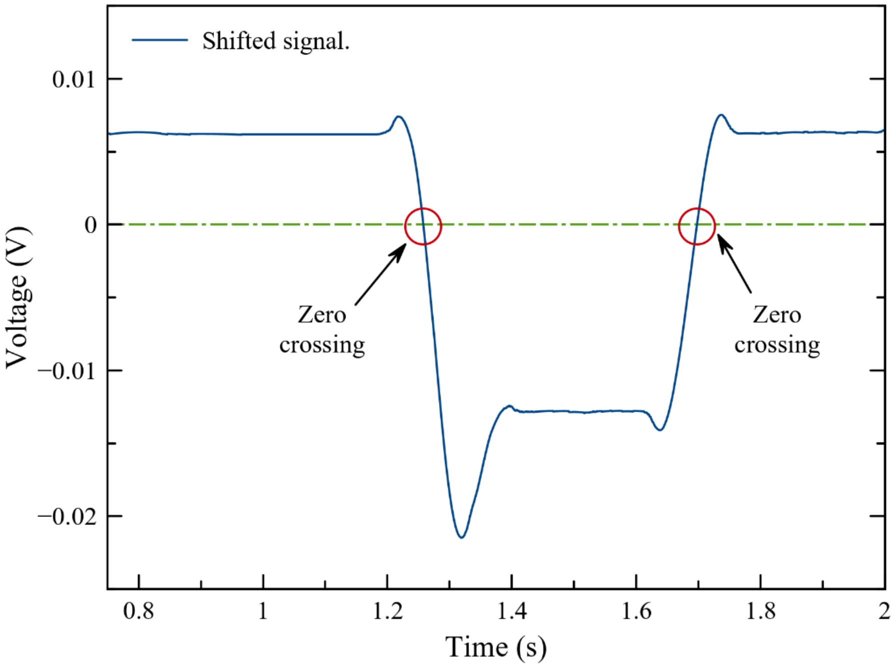


Figure 19. Shifted signal for Zero-crossing detection.

Thus, using equation (6) replacing the number of zero-crossing, which in Figure 19 is two for a single droplet, the final result is one.

$$\text{Number of droplets } = \frac{\text{Number of "Zero} - \text{Crossings"}}{2} \tag{6}$$

2.4.2.4 Local maxima and minima.

After the signal is filtered and the number of droplets extracted, the algorithm extracts the local maxima and minima, due to its vital information for calibration processes or in-flow-droplet characteristics calculations. To find local maxima and minima, we used the "findpeaks" function from MATLAB [34]. This function was programmed with five customizable characteristics to optimize peaks finding in signal shapes similar to Figure 18 and Figure 2, which are: peaks number, minimum distance, prominence, minimum height, and threshold.

As the name states, the number of peaks is the maximum number of peaks retrieved by the algorithm, which set up to double the number of droplets expected. The minimum distance between each peak was established to twice the sampling rate. Minimum peak prominence is a constant that represents how much the peak stands out due to its intrinsic height and its location relative to other peaks, fixed at 10e-4 for the algorithm. The minimum height describes the minimum value required for a peak to be considered local maxima or minima; this is fixed as the mean Y-Axis value. The threshold is a nonnegative real scalar that establishes the minimum height difference between a peak and its neighbors, set up with a 1e-11 value.

The function applied to the signal from Figure 18 to find local maxima and minima is described in Figure 20, where the red triangles indicate the peaks localized automatically.

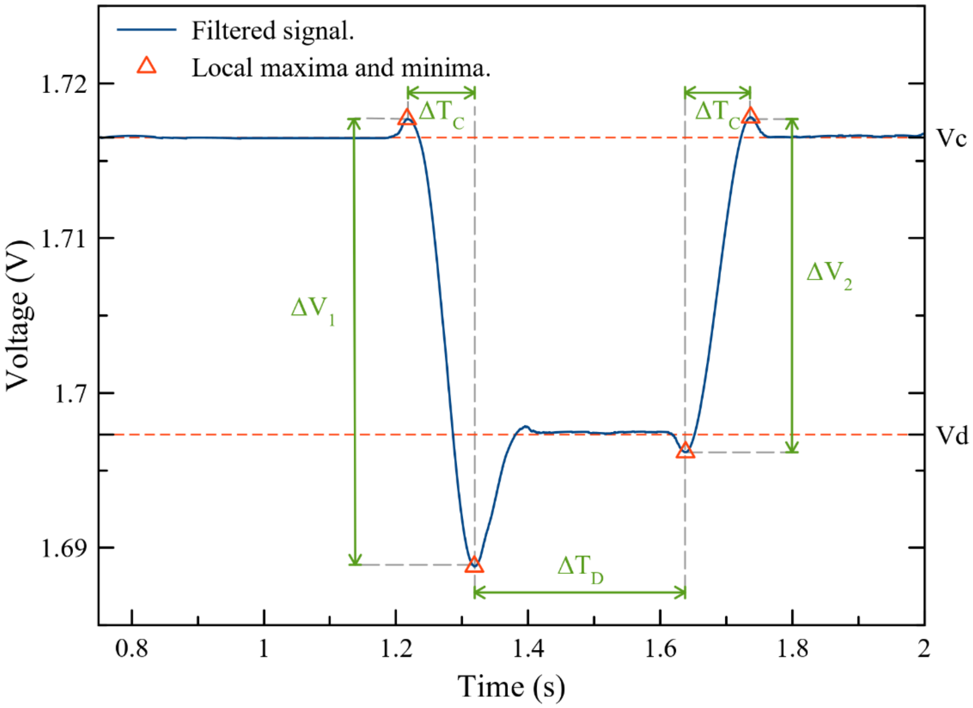


Figure 20. Local maxima and minima for a filtered signal.

2.4.2.5 Calibration process.

Before the in-flow-droplet characteristics calculations, the last step is to begin a calibration process to determine the interfacial distance (λ). For this objective, it is required that the user runs a controlled experiment for adjusting the interfacial distance using a constant flow rate. The calibration process utilizes specific signal characteristics as shown in Figure 20. The first important variable is the voltage difference between a pair of maxima and minima peaks denoted by (ΔV), followed by its time difference expressed as (ΔT_C). Therefore, to calculate λ in millimeters, we first calculated the slope (m) in volts per second using (7) between these peaks due to its linear behavior.

$$m = \frac{\Delta V}{\Delta T_C} \tag{7}$$

Afterward, the experiment velocity (V_E) in millimeters per second is calculated dividing the known flow rate for the controlled experiment (Q) by the cross-sectional area of the microfluidic channel (A_0), which diameter is 762 μm, as shown in (8).

$$V_E = \frac{Q}{A_0} = \frac{Q}{\frac{\pi * 0.762^2}{4}} = \frac{Q}{0.4560} \tag{8}$$

Consequently, the interfacial distance λ in millimeters is calculated by multiplying the inverse of (m) by V_E and ΔV as shown in (9).

$$\lambda = \frac{V_E}{m} * \Delta V = \frac{V_E}{\frac{\Delta V}{\Delta T_C}} * \Delta V = V_E * \Delta T_C \tag{9}$$

Notice that there will be two approximately equal λ values during the calibration process due to the two pairs of peaks for maxima and minima for a droplet in a signal. This number increases proportionally to the number of droplets used as well as the flow rates. Therefore, to predict a final value, we used the statistical Bootstrapping simulation technique, simulating 1000 times the calibration results obtain a final mean λ value and a 95% confidence interval, discussed in-depth the next section.

The interfacial distance "λ" represents the distance that the droplet needs to travel in the microchannel to enter the detection region, or in other words, the fixed distance related to the ΔT_C in the phase changing region described in Figure 2. As described in the figure, this interfacial distance will be approximately equal for both the droplet's entrance and exit from the detection region when there is a good alignment between the photoresistor and the LED. Furthermore, this constant is related to the microchannel's light dispersion, capable of enlarging or shortening the sensing area, conditioned by the photoresistor's 5 millimeters length.

Therefore this interfacial distance "λ" is a suitable parameter for establishing the minimum length required to detect a droplet. The results for testing this parameter demonstrate values ranging from 1.25 up to 3.00 mm. Thus, to assure the correct droplet's measurement, their minimum length was set at 3 mm for this prototype.

After the calibration process is done, the device is ready to record new data and calculate the in-flow-droplet characteristics: velocity, flow rate, length, and volume. Therefore, we use equation (10) to calculate the droplet velocity (V_S) in which ΔT_C is extracted as equation (7) for the new signals.

$$V_S = \frac{\lambda}{\Delta T_C} \tag{10}$$

The resolution for the velocities measurements depends on the minimum change in time that the sensor could detect, which is the sampling rate. As shown from the different ADC configurations, the device's sampling rates are 2.3 and 5 kHz, which extracts a sample for the signal every $4.34*10^{-4}$ and $2.00*10^{-4}$ seconds, respectively. Therefore, the resolution for velocities measurements is $1.08*10^{-3}$ mm/s for slow sampling rates and $5.00*10^{-4}$ mm/s for high-speed sampling rates.

Equation (11) calculates the droplet length (ℓ) by using the droplet velocity extracted from (10), and the total detection time ($\Delta T_C + \Delta T_D$), in which ΔT_D is extracted as shown in Figure 20.

$$\begin{aligned} \ell &= V_S*(\Delta T_C+\Delta T_D) \\ &= V_S*\Delta T_C + V_S*\Delta T_D \\ \ell &= \lambda+(V_S*\Delta T_D) \end{aligned} \tag{11}$$

The resolution for the lengths' measurements follows the same pattern as the velocities, which is entirely related to the sampling frequency. Therefore, the calculated resolution for slow sampling rates is 1.08*10^{-3} mm, and for high-speed sampling rates is 5.00*10^{-4} mm.

The flow rate (Q_S) is calculated by multiplying the measured velocity V_S by the cross-sectional area A_0, as shown in (12).

$$Q_S = V_S{*}A_0 \tag{12}$$

The flow rate resolution is directly proportional to the velocities resolutions by A0. Therefore, the resolution for slow sampling rates is 4.93*10^{-4} mm^3/s, and 2.28e-4 mm^3/s for high-speed sampling rates.

The last in-flow-droplet parameter calculated is the volume (υ), as it is described in equation (13), extracted by multiplying the droplet length ℓ by the cross-sectional area A_0.

$$\upsilon = \ell * A_0 \tag{13}$$

The volume resolution is directly proportional to the length's resolutions by A_0. Therefore, the resolution for slow sampling rates is 4.93*10^{-4} mm^3, and 2.28e-4 mm^3 for high-speed sampling rates.

The resolutions presented in this section are the minimum and theoretically expected by considering only the ADC sampling frequency. It will require additional errors expected such as in the calibration process, microchannel, LED and LDR alignment, and peak selections for the maxima and minima detection.

For measurements comparison we used the mean relative error (MRE) between measured and expected values as described in (14). For more than two MREs the final value will be the average of the results.

$$\text{MRE} = \frac{|expected\ -\ measured|}{expected}*100\% \tag{14}$$

2.5 Statistical methods.

This section's statistical methods are based on confidence intervals using Bootstrapping simulation to estimate a mean value for the interfacial distance. This technique is useful whenever the population's distribution is unknown, which is characteristic of λ values.

The bootstrap simulation methodology builds the confidence interval by drawing random bootstrap samples from the data with replacement (λ_1, λ_2, …,λ_n). Each bootstrap sample is the same size as the data, and it may contain duplicated values due to the replacement values of the sampling. This procedure is illustrated in Table 4; the first column represents the number of the bootstrap sample, followed by the second containing the samples (λ^*_1, λ^*_2, …,λ^*_n). The third and last column the mean for each bootstrapping sample.

The minimum bootstrap samples suggested by the literature is 1000, to have a more accurate result [35].

Table 4. Bootstrap sampling technique.

Sample	Sample Values				Sample Mean
1	λ^*_1	λ^*_2	…	λ^*_n	$\overline{\lambda_1}$
2	λ^*_1	λ^*_2	…	λ^*_n	$\overline{\lambda_2}$
⋮	⋮	⋮	⋮	⋮	⋮
1000	λ^*_1	λ^*_2	…	λ^*_n	$\overline{\lambda_{1000}}$
					$\bar{\lambda}$

Therefore, the 95% confidence interval is calculated using the 97.5 and the 2.5 percentiles for the upper and lower limit, respectively. The 97.5 percentile is the average of the 975th and 976th values from the ordered Table 4. Similarly, the 2.5 percentile is the average of the 25th and 26th values.

In addition to the confidence intervals, the final λ value is calculated using the average of the entire sample mean column of Table 4, which represents the end of the calibration process. The final MATLAB code is described in Appendix B.

2.6 Assembly.

This section describes the prototype's final assembly, converging a single device the designed hardware and software. Furthermore, it presents the requirements and configurations for the sensor to run experiments.

Using the final hardware assembly described in Figure 10, we uploaded Arduino's software into the ATmega38P microcontroller and started the MATLAB program to read and process the sensor's signals. Once this procedure is done, the device is ready for measuring.

Figure 21 describes the experimental set up for the sensor. It uses a New Era Pump Systems syringe pump [36] connected to the microchannel in order to control the microfluidic flow. Additionally, it illustrates the connection required between a computer and the microcontroller to read, process, and record the signal using MATLAB.

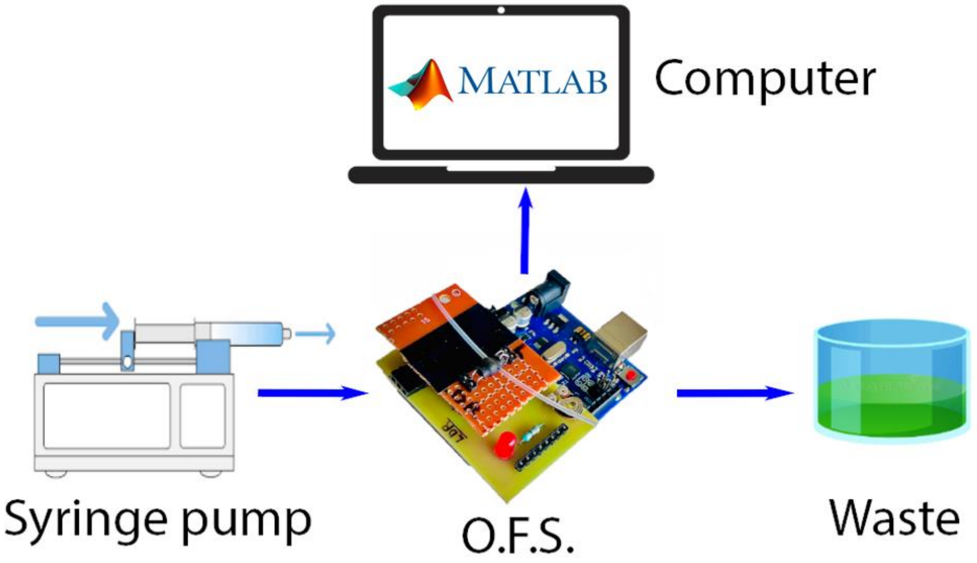


Figure 21. Experimental set-up.

The drops were prepared by suctioning the dispersed phase's solution into the microchannel using a syringe until approximating the desired size of the droplet. This procedure was repeated until forming a system of various drops separated by the continuous phase, which for the experiments reported in this work is air.

The final assembly allows the user to record the reading signal into the MATLAB environment for processing and calculations, delivering the number of droplets detected, velocity, flow rate, length, and volume as outcomes. The expected signals will depend on the dispersed phase's light absorption characteristics, showing higher voltages for darker fluids than transparent, following the behavior described in Figure 2.

Chapter 3

Results

Throughout this section, we will discuss two sets of experiments. The first group was obtained using the standard functions from the Arduino to read the sensor's signal for pure-water tests. The second group was extracted by modifying the ATmega328P microcontroller registers for an enhanced sampling frequency. The experimental set-up used for the experiments follows the configuration described in Figure 21.

3.1 Standard ADC sampling rate.

In order to test the device, water-in-air droplets driven by the syringe pump were used inside the sensor's microchannel. The dispersed phase was arranged in three different slugs or droplets of 6.5054, 4.8882, and 5.1848 mm, described in Figure 22-A. Additionally, Figure 22-B represents the flow direction and displacement of the droplets inside the microchannel. The droplet’s lengths were measured using ImageJ [37].

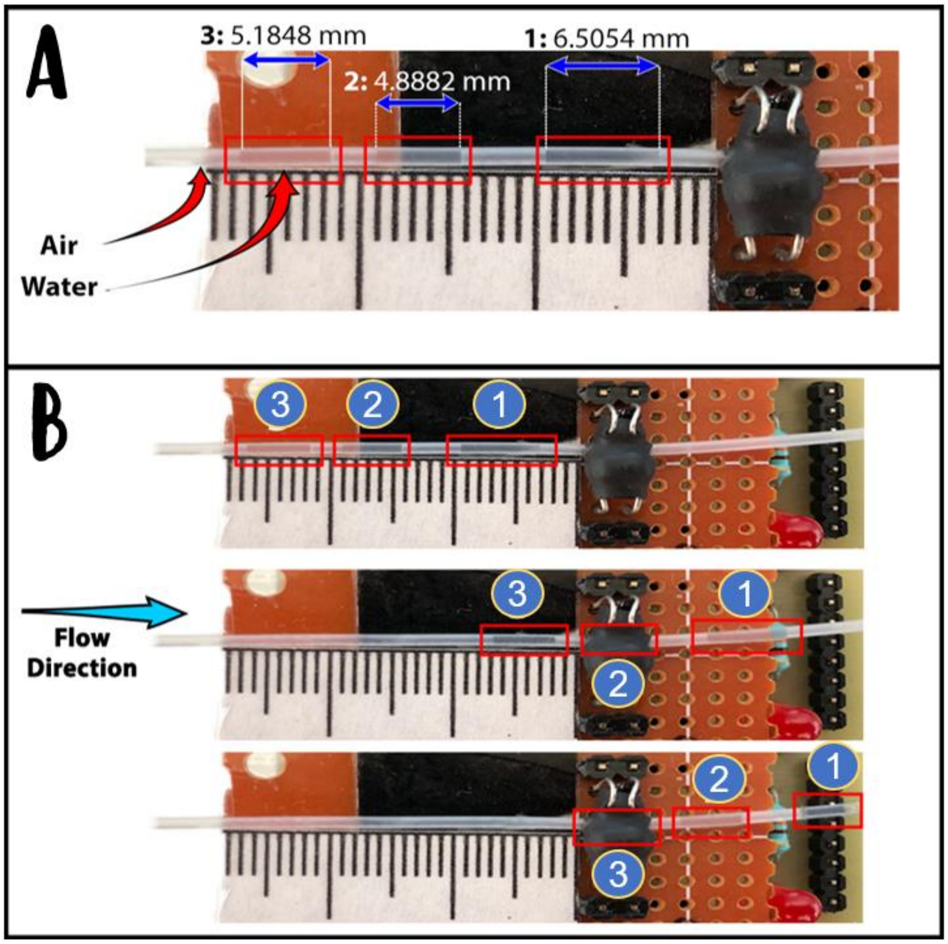


Figure 22. Three experimental water-in-air droplets and flow displacement.

Figure 23 describes the response signal for the experiment at 200 µL/min constant flow rate, this signal shows how the voltage drops whenever the dispersed phase (water) passes through the sensing area. The continuous phase voltage (V_c) for this signal is 1.7725 V and the dispersed phase voltage (V_d) is 1.7529 V, at a 2.3 kHz sampling rate.

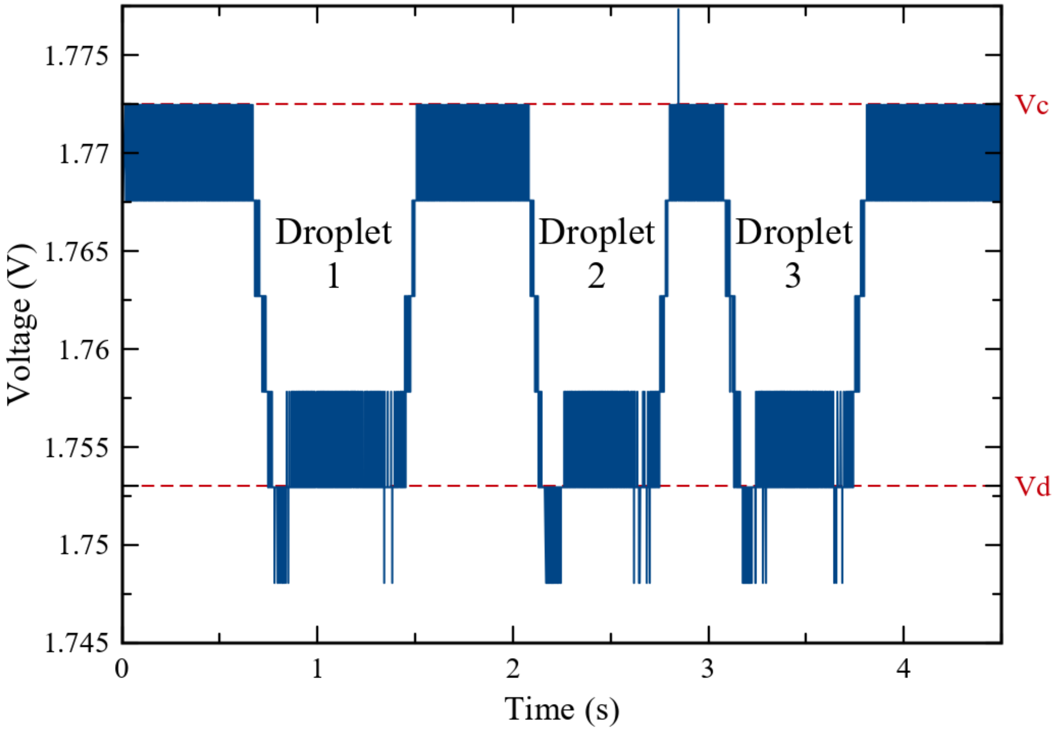


Figure 23. Water-in-air raw signal recorded at 200 µL/min constant flow rate.

3.1.1 Calibration process.

In order to calibrate the sensor, ten experiments were carried with the three droplets mentioned above, starting at 20 µL/min flow rate up to 425 µL/min with intervals of 45 µL/min as described in Table 5.

Table 5. Calibration experiments for standard ADC configuration.

Experiment	Flow rate (µL/min)
1	20
2	65
3	110
4	155
5	200
6	245
7	290
8	335
9	380
10	425

The filtered signal for this microcontroller configuration is shown in Figure 24. Due to the reduced sampling rate, the best smoothing factor for the Savitzky-Golay filter was increased to 0.35 compared to the one described in the previous section. Additionally, the figure describes all the signal characteristics taken to calculate the interfacial distance (λ) using equations (7), (8), and (9).

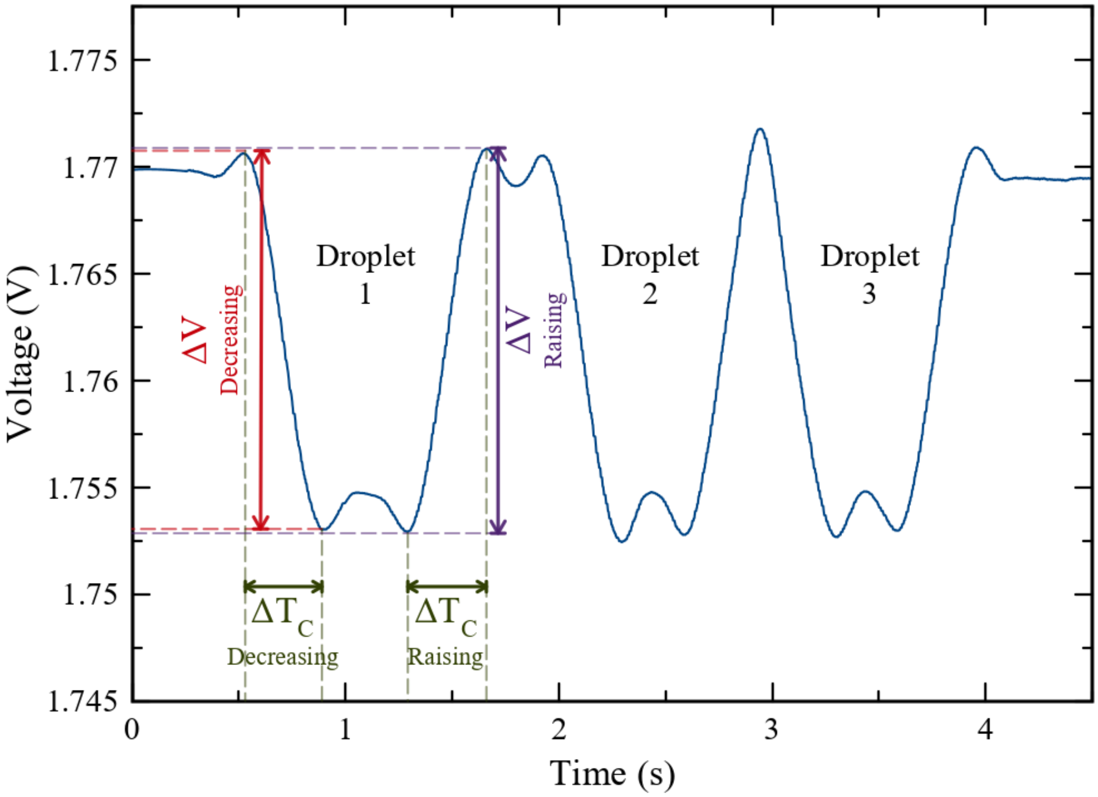


Figure 24. Filtered signal for a 200 μL/min constant flow rate.

Therefore, applying the algorithm for local maxima and minima to the extracted signals and replacing the desired points in the equations mentioned above, we obtained six approximately equal λ values, three using decreasing voltages and three for the raising. These calculations were done automatically using the MATLAB algorithm. The results are described in Table 6.

Table 6. λ results for the calibration process signals.

Experiment	Q (μL/ min)	Droplet	λ Decreasing (mm)	λ Raising (mm)
1	20	1	2.8402	2.9127
		2	2.8616	2.6392
		3	2.638	2.884
2	65	1	2.839	2.8458
		2	2.8694	2.5882
		3	2.6373	2.8544

3	110	1	2.7192	2.7087
		2	2.7203	2.6281
		3	2.5719	2.6879
4	155	1	2.8009	2.8409
		2	2.9359	2.6945
		3	2.6465	2.9488
5	200	1	2.7374	2.7085
		2	2.6821	2.6478
		3	2.6406	2.6673
6	245	1	2.8666	2.7896
		2	2.7695	2.6116
		3	2.7582	2.8652
7	290	1	3.0328	2.9234
		2	2.9595	2.7156
		3	2.8552	3.0016
8	335	1	2.6571	2.6461
		2	2.6604	2.6387
		3	2.6939	2.6925
9	380	1	2.7244	2.7091
		2	2.7954	2.6252
		3	2.8868	2.6435
10	425	1	2.9387	2.728
		2	2.7761	2.7234
		3	2.7261	2.715

Once the λ's values were extracted, we used the Bootstrapping simulation as described in the previous section to simulate 1000 times these experiments to construct a 95% confidence interval and find the mean λ value. Similar to the previous calculations, MATLAB automatically delivers the bootstrapping simulation results.

Hence, the calculated 95% confidence interval for the mean λ value has a lower bound of 2.7323 mm and an upper bound of 2.7869 mm. Consequently, the final λ value for this calibration process is **2.7582 mm**.

3.1.2 Tests.

The velocities extracted from the sensor were calculated using a MATLAB algorithm with equations (9) and (10). These were compared with the experimental velocities from Table 5, calculated separately using (8). Furthermore, from equation (10) for velocities and (12) used to calculate flow rate, we can conclude that they are proportionally related to the microchannel's cross-sectional area (A_0), describing the same pattern shifted by A_0. Therefore Figure 25 describes these results and includes a comparison with the real reference values, in which the left Y-axis represents velocities and the right Y-axis flow rates. The X-axis represents the number of the droplet measured.

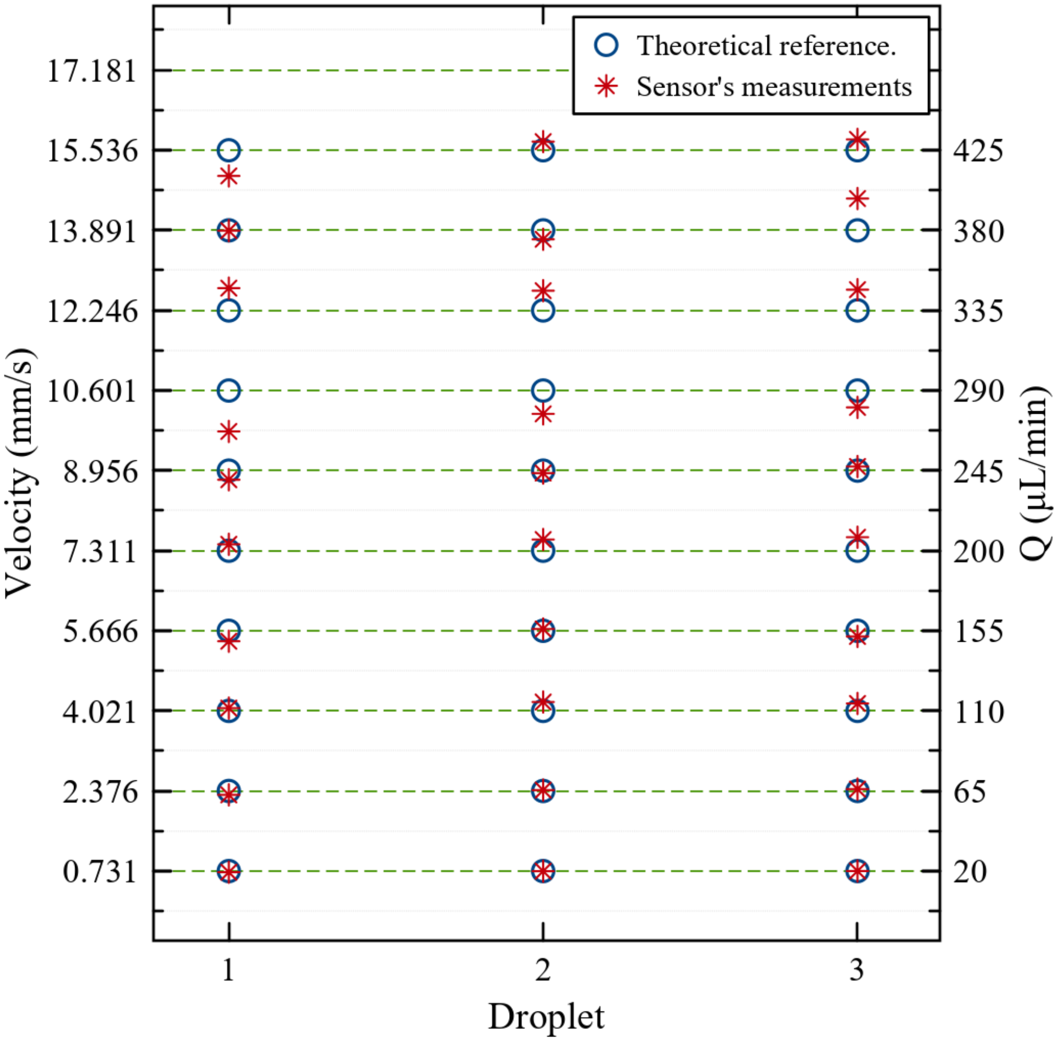


Figure 25. Sensor's velocities and flow rates measurements vs. theoretical references

As shown in Figure 25, the difference between the reference values versus the sensor's measurements are not significant. Therefore, we calculated the mean relative error (MRE), resulting in a 2.4% error between the theoretical and measured values.

However a more in-depth study of the errors is shown in Figure 26, which describes the mean relative error per experiment for both sensor's velocity and flow rate measurements since they are proportional to each other. The X-axis represents the flow rate for each independent experiment, and their mean relative error percentage values are described in the Y-axis. From the figure, we can observe that measurements are below 5.5% (Q = 290 µL/min) error compared with their real reference values.

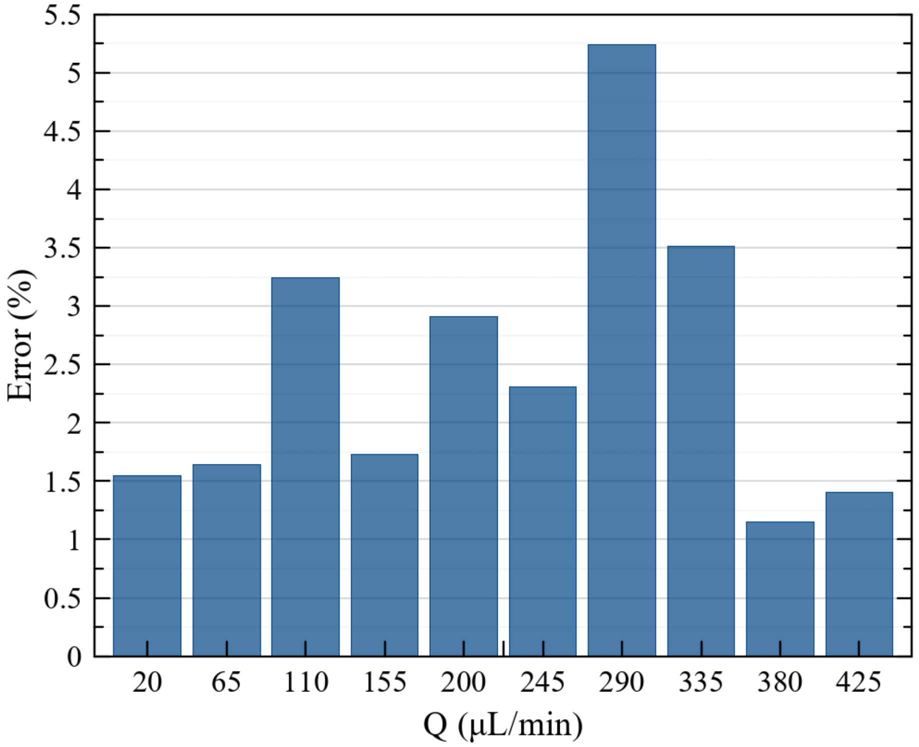


Figure 26. MRE in % per experiment for velocities and flow rate measurements.

Furthermore, droplet lengths and volume were calculated using equations (11) and (13), respectively, similarly to the velocities and flow rates, droplet lengths and volume are proportionally related by A_0. Figure 27 describes the sensor results compared with their theoretical reference values. The left Y-axis denotes droplet length in millimeters and the right Y-axis volume in microliters for a total of the three droplets measured ten times at different flow rates in the ten experiments. The mean relative error for both length and volume resulted in 8.04%.

Figure 28 describes the mean relative errors for the length and volume measurements. Similarly to Figure 25, the X-axis describes the experiment flow rate and the Y-axis the percentage error. We can observe that according to the MRE for these measurements, half of the experiments have less than 7% relative error; however, the experiment under the flow rate of 290 (µL/min) has a 14% error increasing the total MRE error. Although MRE for length and volume is higher than velocity and flow rate, it remains under 10%, which gives promising results compared with state-of-the-art systems.

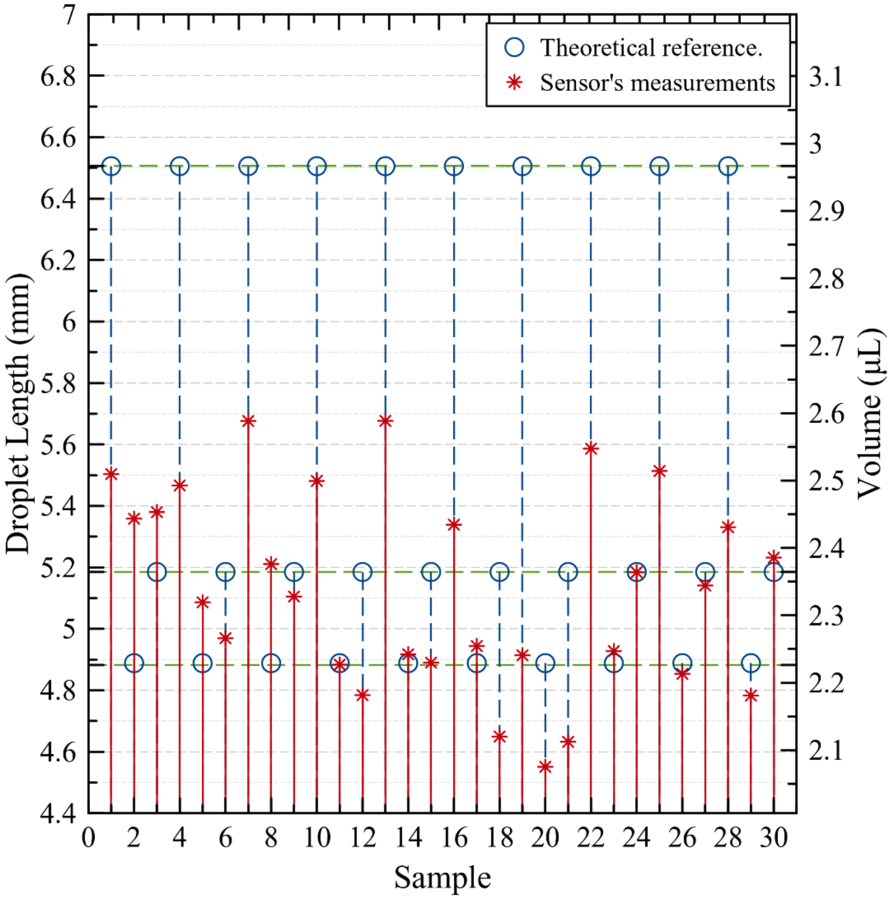


Figure 27. Sensor's length and volume measurements vs. theoretical references.

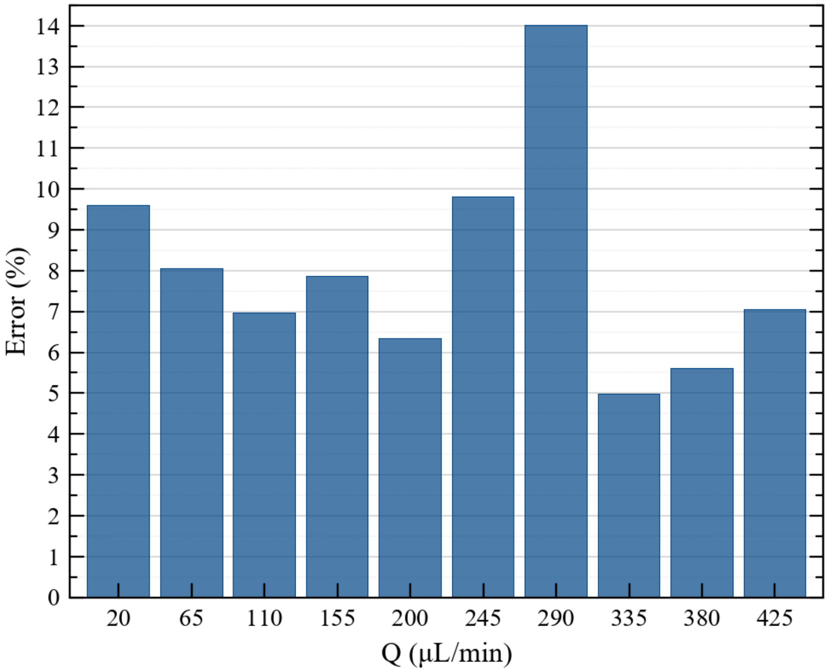


Figure 28. MRE in % per experiment for length and volume measurements.

3.2 High speed ADC sampling rate.

To test the device using a high-speed ADC sampling rate (5kHz) configuration described in the Programming section, we designed nine experiments using three water-in-air droplets at different flow rates, similar to previous tests. The droplets' lengths measured using ImageJ are 3.877, 5.275, and 5.914 mm, as shown in Figure 29.

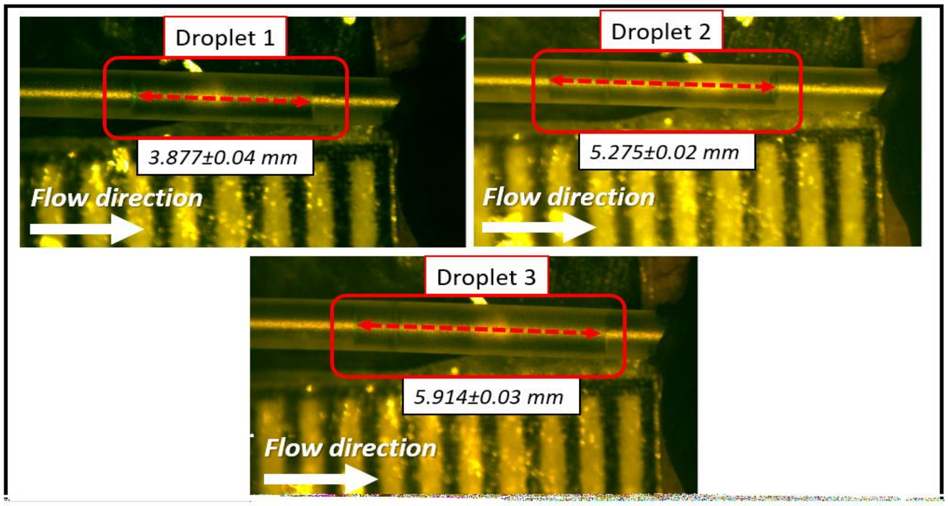


Figure 29. High-speed sampling rate droplets configuration.

The raw signal response for an experiment using a 670 µL/min flow rate is described in Figure 30. This signal describes the same behavior as previous results. The voltage drops whenever the dispersed phase passes through the sensing area. For this signal, the continuous phase voltage is 1.735 V, and the dispersed phase voltage is 1.686 V, extracted from the figure.

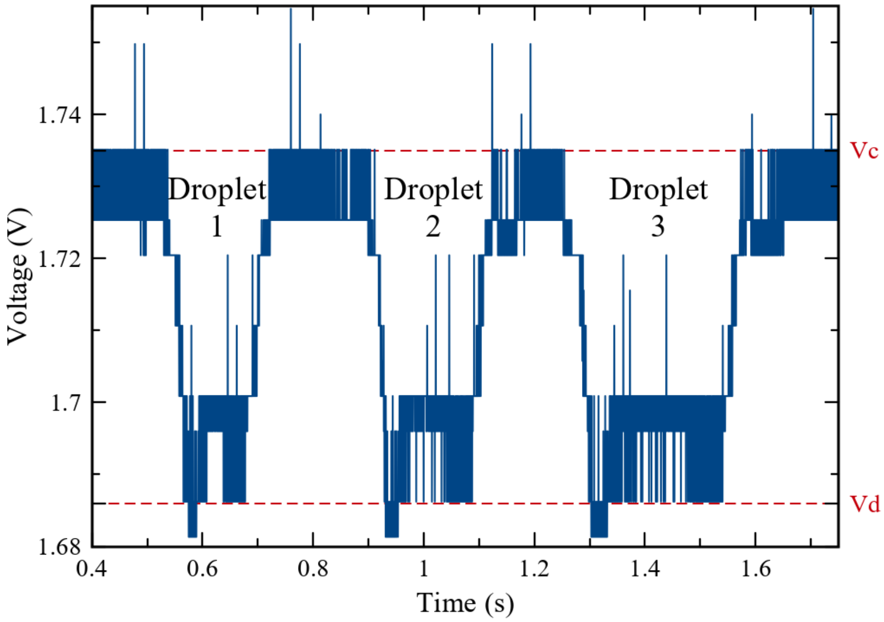


Figure 30. Raw signal extracted at 670 µL/min constant flow rate.

3.2.1 Calibration process.

The new sampling rate requires to calculate a new interfacial distance for these droplets and high-speed acquisition data. Therefore nine experiments were carried with the droplets above mentioned, starting at 70 µL/min flow rate up to 870 µL/min with intervals of 100 µL/min as described in Table 7.

Table 7. Calibration experiments for standard high-speed sampling rate.

Experiment	Flow rate (µL/min)
1	70
2	170
3	270
4	370
5	470
6	570
7	670
8	770
9	870

Figure 31 describes the filtered signal used for calculations. The high-speed sampling configuration reduced the Savitzky-Golay smoothing factor to 0.2 due to the additional samples taken in the same period. Similarly to Figure 24, this figure shows the characteristics taken to calculate λ for equations (7), (8), and (9).

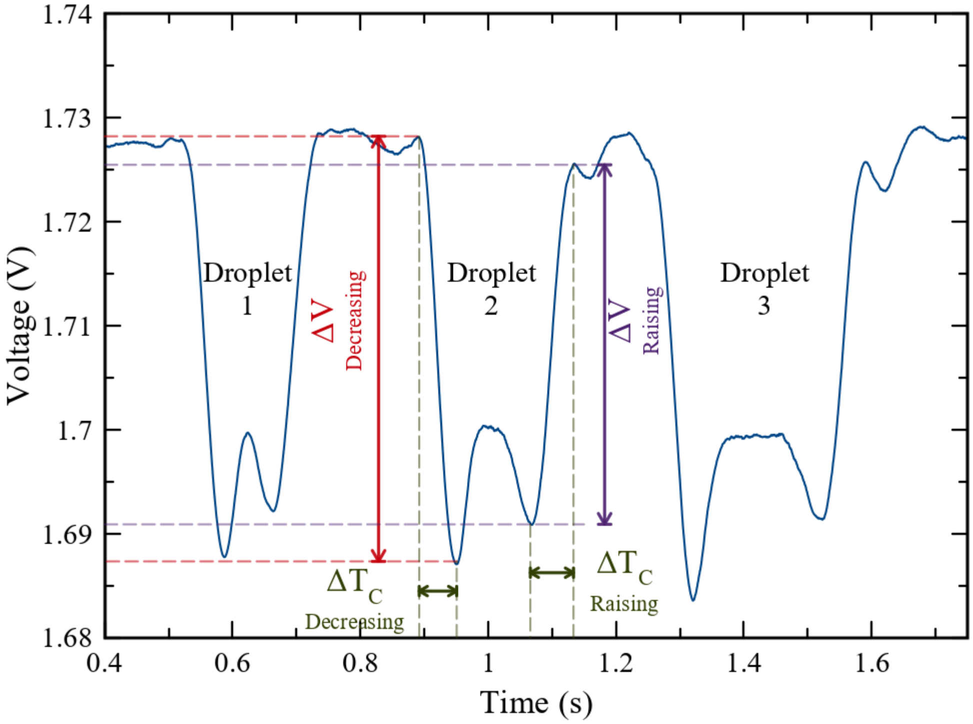


Figure 31. Filtered signa for a 670 µL/min flow rate.

Therefore, after extracting the local maxima and minima points from the signals, we calculated λ for the experiments. Table 8 describes the 54 λ values calculated before the bootstrapping simulation.

Table 8. λ results for high-speed sampling configurations

Experiment	Q (μL/ min)	Droplet	λ Decreasing (mm)	λ Raising (mm)
1	70	1	1.8092	1.5385
		2	1.8767	1.4398
		3	2.0256	1.6271
2	170	1	2.0167	1.7160
		2	1.2500	1.4600
		3	1.5023	1.9471
3	270	1	1.5512	1.6953
		2	1.3440	1.4861
		3	1.6025	2.1946
4	370	1	1.7904	1.8931
		2	1.4253	1.4307
		3	1.5307	1.5470
5	470	1	1.4738	1.5391
		2	1.3329	1.6559
		3	1.4188	1.9582
6	570	1	1.4457	1.8124
		2	1.2582	1.4540
		3	1.9207	1.5707
7	670	1	1.6510	1.7735
		2	1.4711	1.6324
		3	1.7032	1.6373
8	770	1	1.8292	1.8292
		2	1.6885	1.8854
		3	1.8292	1.8292
9	870	1	1.4658	1.5993
		2	1.5357	1.5612
		3	1.5675	1.6534

Using the data presented in Table 8, we generated the bootstrap simulation process for a 95% confidence interval. The lower bound for this simulation appeared in 1.5873 mm and the upper bound at 1.6972 mm, demonstrating that λ does not vary much from one experiment to another. The mean λ value calculated from the bootstrapping simulation is **1.6426 mm**.

3.2.2 Tests.

The velocities were measured using equations (9) and (10) inside the MATLAB algorithm and then compared with the experimental velocities extracted from Table 7 using equation (8). Additionally, the flow rate was measured using equation 12. Both results are presented in Figure 32 due to the velocities are proportionally related by (A_0) to the flow rates. The left Y-axis represents velocities and the right Y-axis flow rates. The X-axis represents the number of the droplet measured.

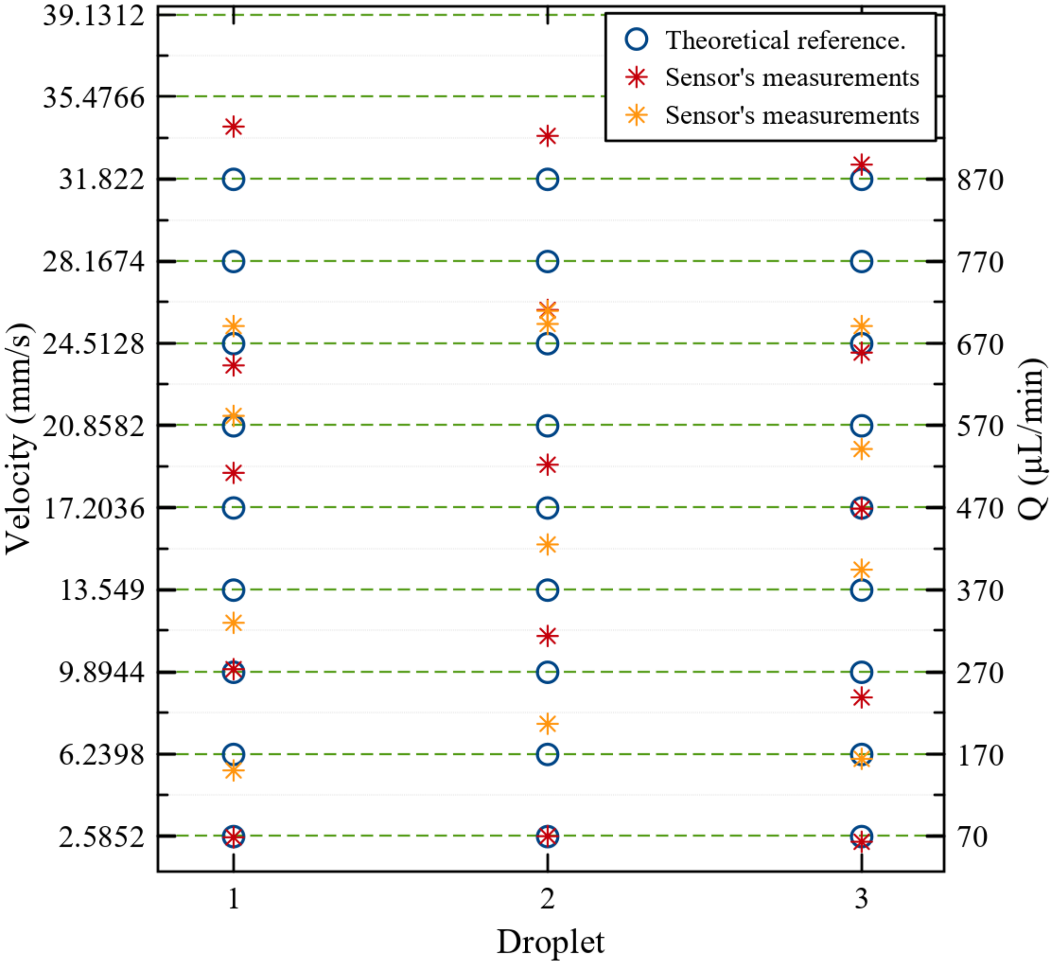


Figure 32. Sensor's velocities and flow rates measurements vs. theoretical references for high-speed sampling rate. Red stars represent measurements for 70, 270, 470, 670 and 870 µL/min. Orange stars are measurements for 170, 370, 570 and 770 µL/min.

As shown in Figure 32, the differences between the reference values versus the sensor's measurements are not significant, except at 570 µL/m, where the sensor's results vary the most related to droplet number 2. However, the mean relative error (MRE) for these measurements resulted in a 7.92% error between the theoretical and measured values.

Opposed to Figure 25, where no overlapping is observed, the results described in Figure 32 show overlapped measurements in the regions from 570 to 770 µL/min. For measurements at 570 µL/min from the right Y-axis, an overlapped measurement for droplet 2 shifted up to the region of the 670 µL/min responses or its analogous velocity in the left Y-axis. Similarly, measurements at 770 µL/min describe the overlapping in a similar way, to the 670 µL/min region for all the droplets tested. Therefore, from this particular section (570 to 770 µL/min with their corresponding velocities), we can conclude that the minimum flow rate between detections to avoid overlapped results would be 200 µL/min, or 7.3092 mm/s for velocities (right and left Y-axis, respectively). This will assure for a future group of measurements to be closer to the target and to reduce their dispersion.

Figure 33 presents the MRE per experiment for the sensor's measurements for velocities and flow rates versus the theoretical values. This figure shows that the highest error percentage was obtained in experiment number two at 170 µL/min with an MRE of 12.17%, followed by experiments at 270, 370, 570, and 770 µL/min, which MREs are close to 10%. However, experiments 70, 470, 670, and 870 µL/min presented error values below 7%.

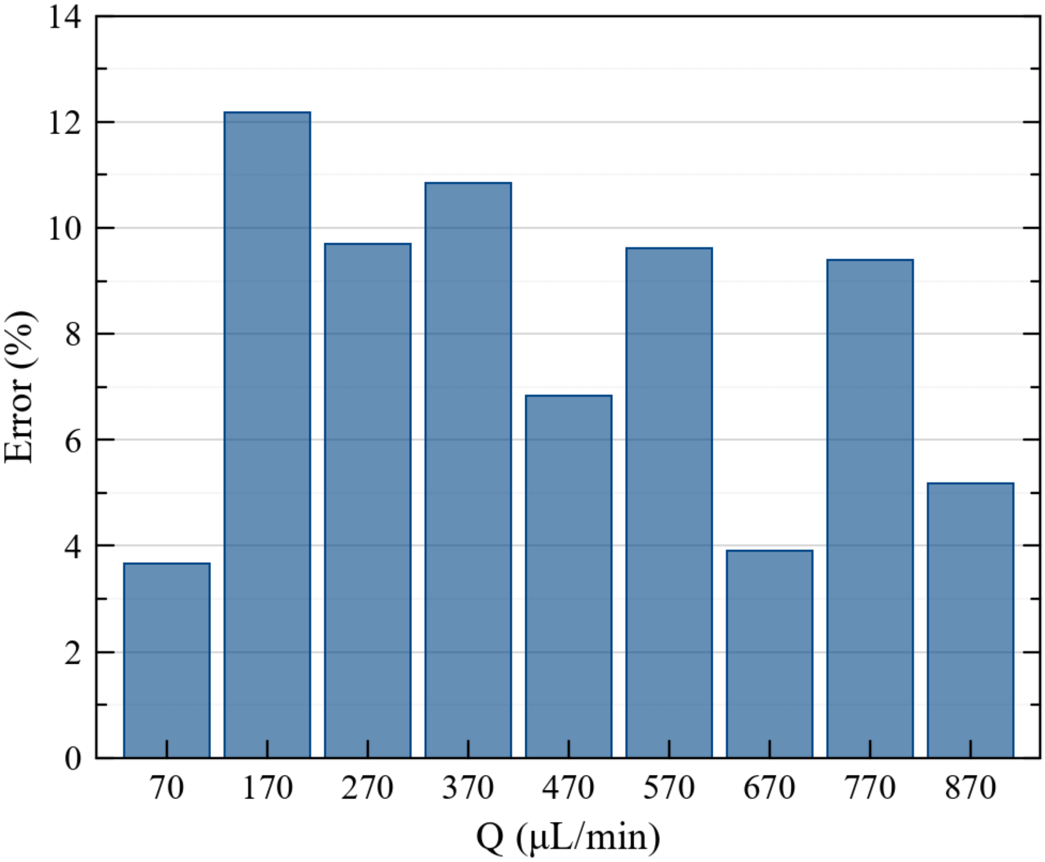


Figure 33. High-speed sampling measurements MREs.

Furthermore, droplet lengths and volume were calculated using equations (11) and (13), respectively. Figure 34 describes the sensor results compared with their theoretical reference values. The left Y-axis denotes droplet length in millimeters with a green dotted line at 3.877, 5.275, and 5.914 mm, showing the experiments' expected values. Additionally, the right Y-axis indicates the volume in microliters for the experiments in which the green line indicates the expected values at 1.768, 2.406, and 2.697 µL, respectively. The mean relative error for both length and volume resulted in 10.85%.

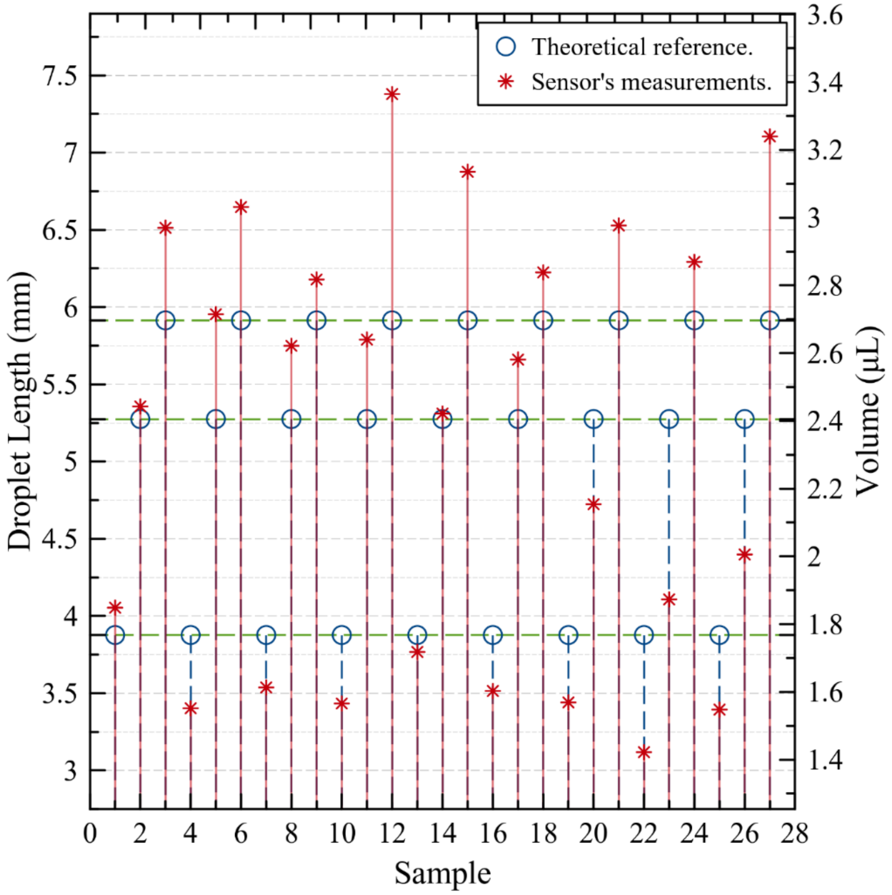


Figure 34.Sensor's length and volume measurements vs. theoretical references for high-speeds sampling rate.

Finally, describes the MRE values for the measured lengths and volume at high-speed sampling rates. The results show two main groups characterized by errors above and below 10%, which their averages results in the final 10.85% MRE value. Even though the measurements have presented increased errors, they show promising results for multiple droplet characterization using the presented sensor.

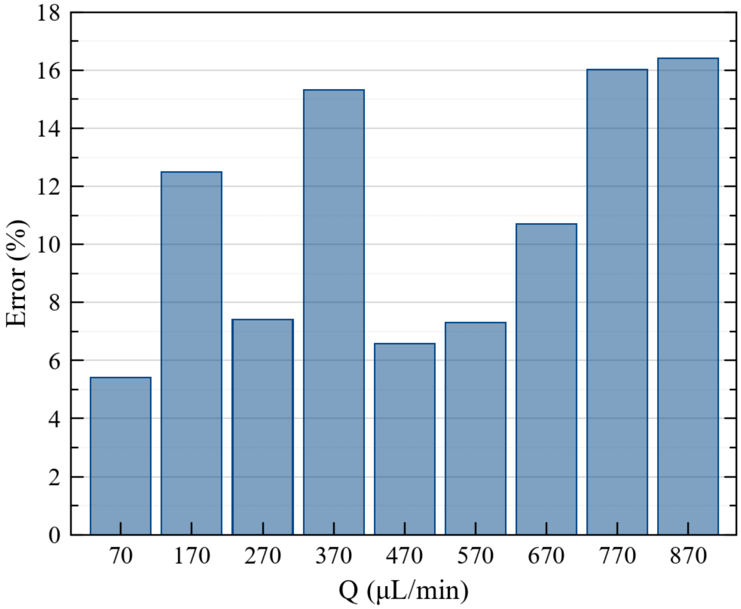


Figure 35. MRE values for measured lengths and volume using high-speed sampling rate.

Chapter 4

Calibration process and experimental improvements.

We described a robust calibration process that used a wide range of flow rates and experiments to obtain λ's right estimate throughout the presented results. However, when a reduced number of experiments is used during this process, the sensor's accuracy may vary, increasing the MRE of the characteristics measured.

Therefore, to tackle this problem, we include a distribution fitting in the calibration process for λ results before the bootstrapping process. This method uses the interquartile range (IQR) for the λ values obtained for a distribution fitting using MATLAB [20], which is further randomly sampled in the bootstrapping simulation, as described in previous chapters. Figure 36 describes the enhanced calibration process.

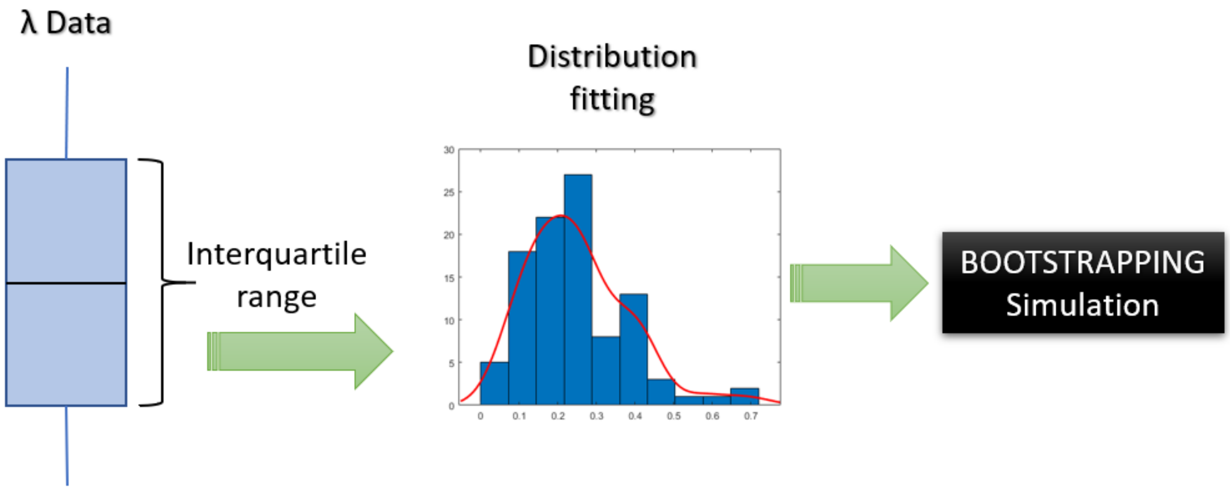


Figure 36. Enhanced calibration process.

Kernel distribution was selected as the fitting target distribution because it represents a nonparametric estimation of a random variable's probability density function [38], reflecting the characteristics of any possible λ value in the experiments.

4.1 Experimental tests with dyed-water droplets

The enhanced calibration process was tested using two sets of experiments with green dyed water. Assessing these improvements and demonstrating the presented sensor's potential to detect and measure similar and different phases.

The droplets' solution was prepared using 30 µL of green food dye per 1 mL of deionized water. As shown in Figure 37, three random droplets or slugs were extracted from the solution and measured using ImageJ software. Results show three droplets with 5.792±0.06, 4.979±0.04, and 5.523±0.03 mm length, respectively.

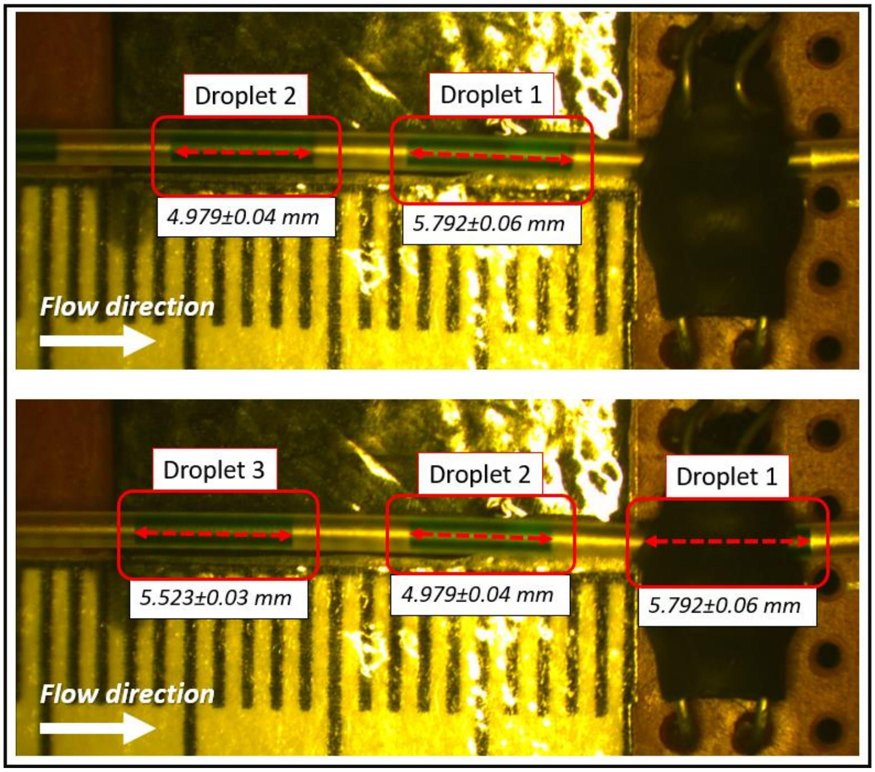


Figure 37. Dyed-water droplets for enhanced calibration testing.

These droplets' calibration process was under two constant flow rates, 100 and 500 µL/min, respectively. Figure 38 describes the raw signal response at 100 µL/min; higher voltages represent the presence of dark droplets with a V_d value of 1.798 volts compared to the continuous phase (air) voltage value of 1.755 volts, as hypothesized in Figure 2, where V_d is above V_c in signal readings.

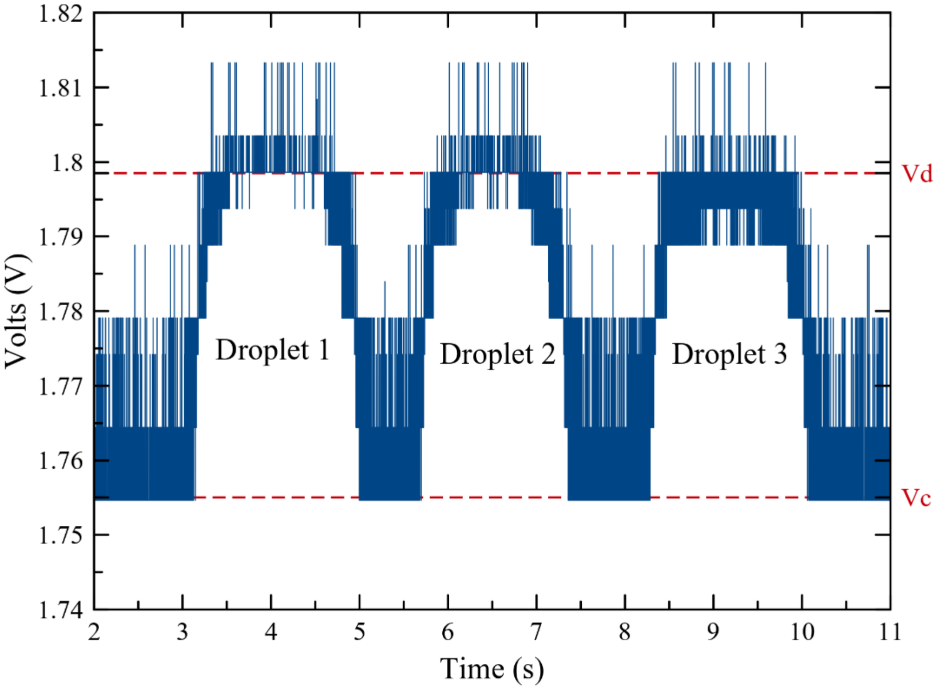


Figure 38. Raw signal for dyed-water droplets recorded at 100 µL/min.

The process for λ calculations previous bootstrapping follows the pattern described previously with a small variation of the smoothing factor decreased to 0.12 for preserving most of the signal information. The filtered signal for a 100 µL/min flow rate is described in Figure 39, with its parameters for calculation.

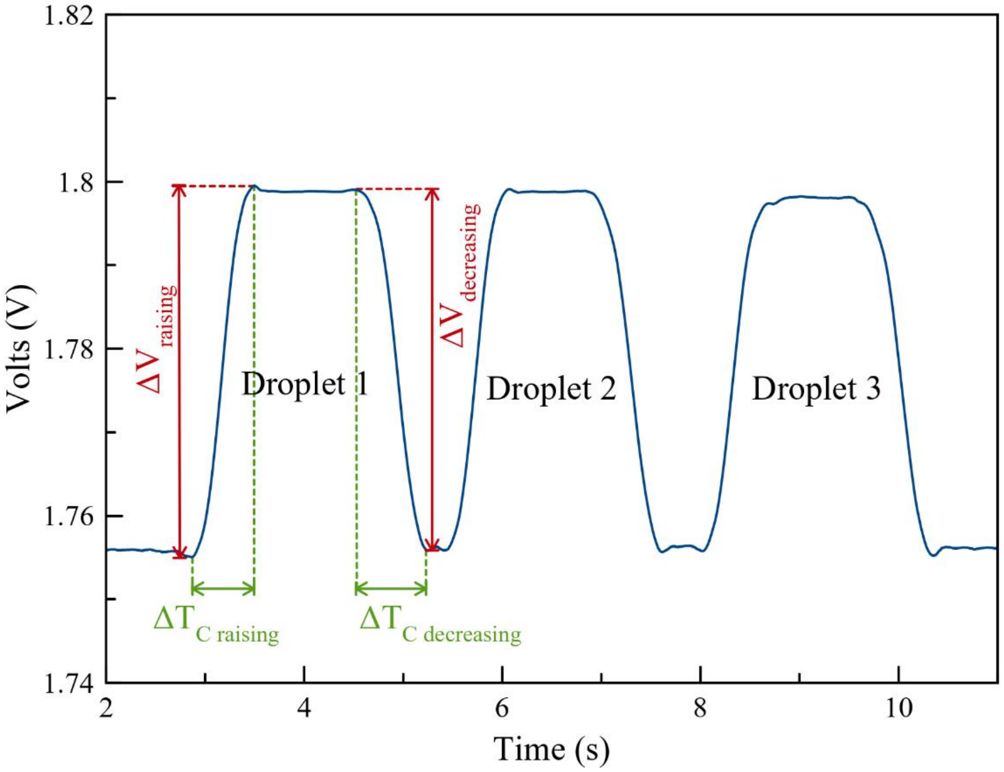


Figure 39. Filtered signal for dyed-water droplets recorded at 100 µL/min.

Table 9 presents the three droplets' λ calculated values at the two different flow rates, leading to twelve responses, that presents variations for the decreasing values at 500 µL/min.

Table 9. Dyed-water droplets λ calculated values.

Experiment	Q (µL/ min)	Droplet	λ Decreasing (mm)	λ Raising (mm)
1	100	1	2.6431	2.3098
		2	2.7732	2.3938
		3	2.8134	2.4691
2	500	1	3.2782	2.2988
		2	3.8118	2.2220
		3	3.6912	2.8580

Therefore, we obtained the results presented in Figure 40 extracting the IQR for Kernel distribution fitting. The figure shows that the distribution is well fitted to the IQR histogram, demonstrating that the distribution describes a very similar behavior for the calculations of the calibration experiments, allowing us to simulate 1000 times random experiments using bootstrapping.

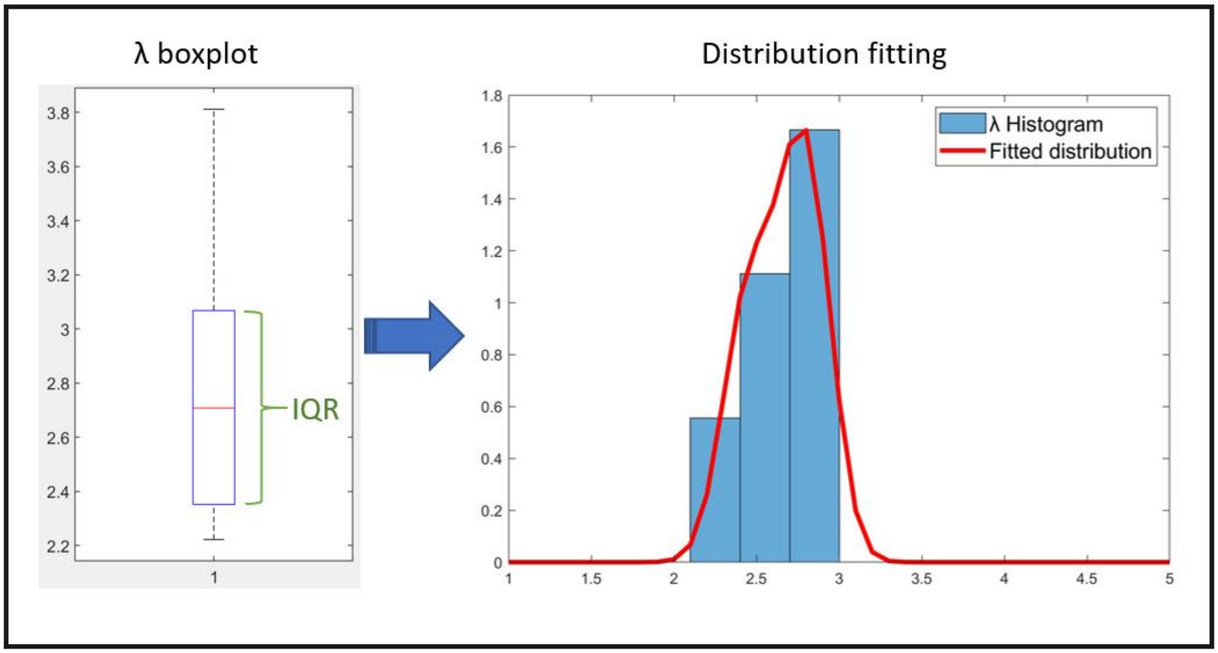


Figure 40. λ boxplot, IQR selection, and final fitted distribution.

After the simulation process for this calibration process, the final calculated **λ value is 2.6570 mm**, with 95% confidence intervals of 2.5229 and 2.7840 mm for the lower and the upper bounds.

Thus, we calculated the velocities and flow rate for the calibration experiments shown in Figure 41. The expected velocities are 3.65 and 18.27 mm/s for the 100 and 500 µL/min flow rates, respectively. As shown in the figure, only the last measurement varies the most compared to the rest of the values, leading to an MRE of 6.13% for the global calculations.

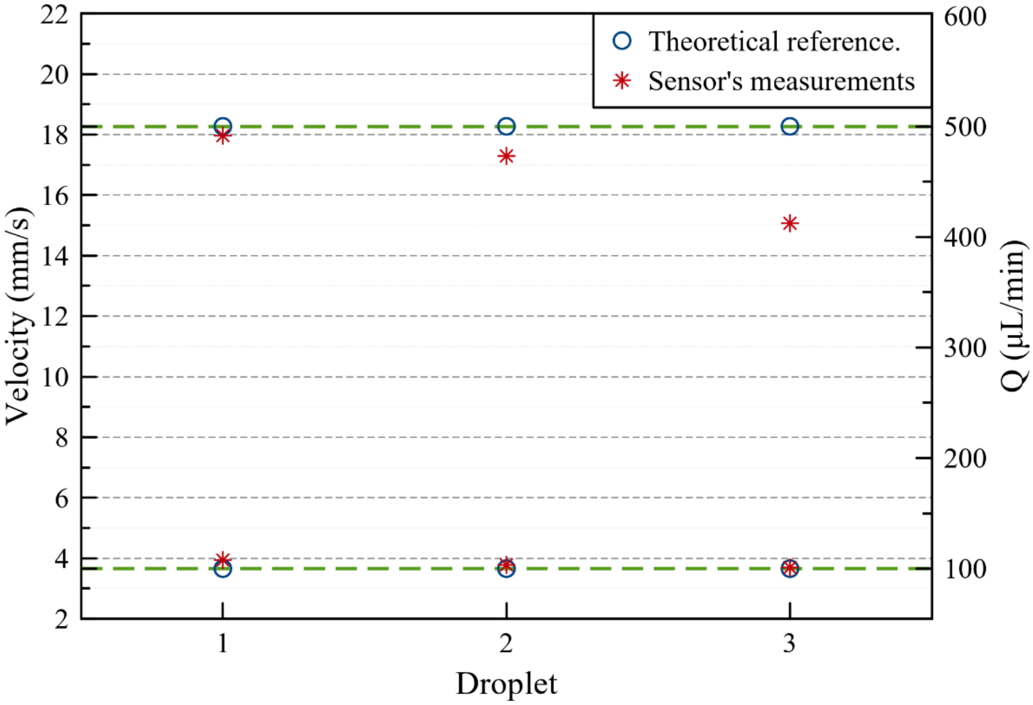


Figure 41. Flow rate and velocities measurements for dyed-water droplets.

Figure 42 describes the MRE values for the velocity and flow rate measurements, showing values around 4% for the first experiment (100 µL/min) and an increase in the error up to 8.2% for the second experiment. These results indicate that higher flow rates will increase the MRE of the expected results, as expected from the hypothesis.

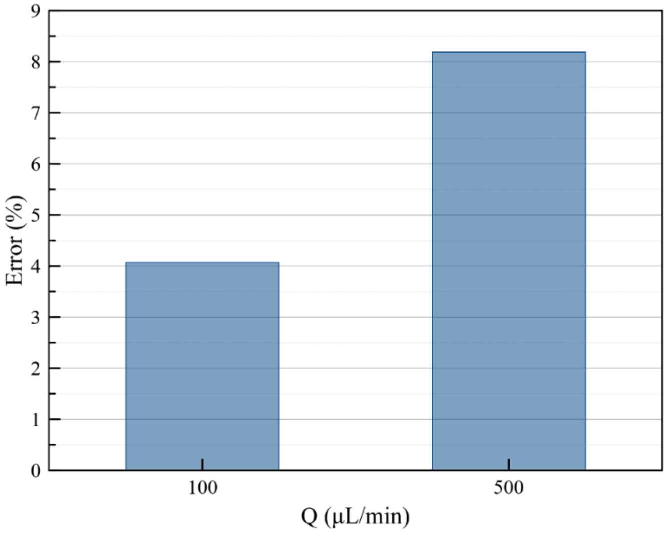


Figure 42. MRE values for velocities and flow rates measurements using dyed-water droplets.

Measured droplet lengths and volume are compared with their theoretical values of the calibration experiments in Figure 43. As shown from the figure, final measured values are similar to the theoretical values obtained using different techniques, as ImageJ for droplet lengths. The MRE calculated for these measurements exhibits an 8.65% error for both properties since they are proportionally related.

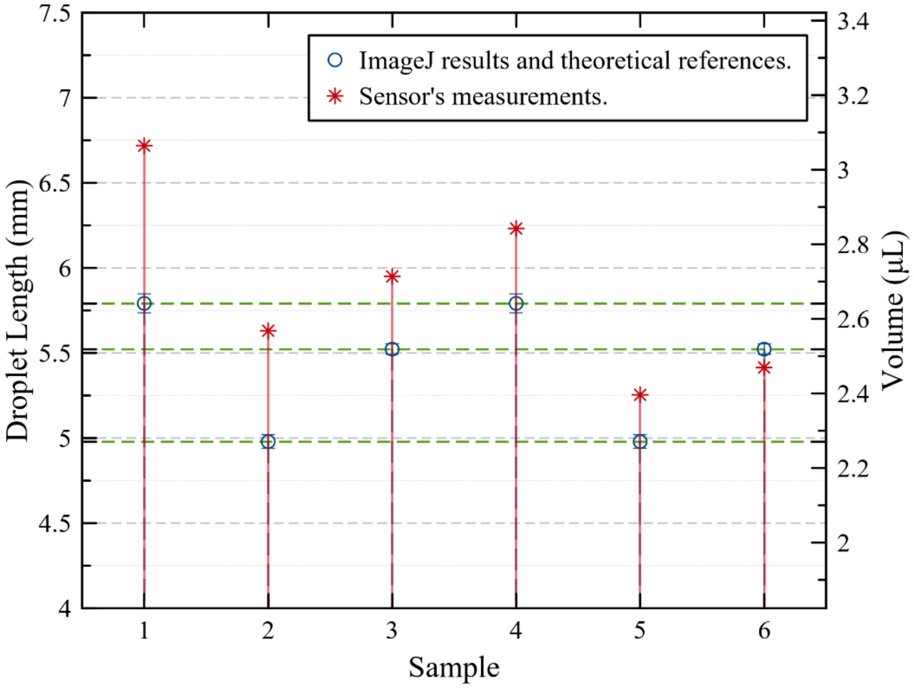


Figure 43. Dyed-water droplets lengths and volume measurements.

The MRE values per experiment for the measured lengths and volume are presented in Figure 44. The results confirm higher error values for the first experiment with 12.28%, as expected from Figure 43. However, the MRE values are reduced to 5.03% for the second experiment. Although the measurements describe higher MRE for low flow rates, 100 µL/min in this specific example, the overall results for this calibration process and tests exhibit a good sensor's performance with a total MRE under 10%.

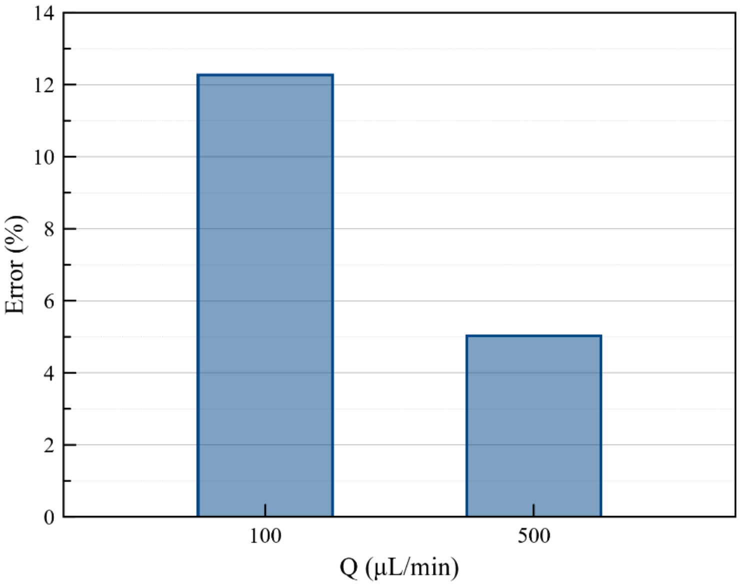


Figure 44. MRE values for lengths and volume measurements using dyed-water droplets.

We performed further tests with this calibration result (λ=2.6570 mm) for two randomly generated dyed droplets at a random flow rate. The resulting droplets are shown in Figure 45, which lengths are 6.457±0.052 and 5.652±0.038 mm (calculated with ImageJ), tested at 226 µL/min.

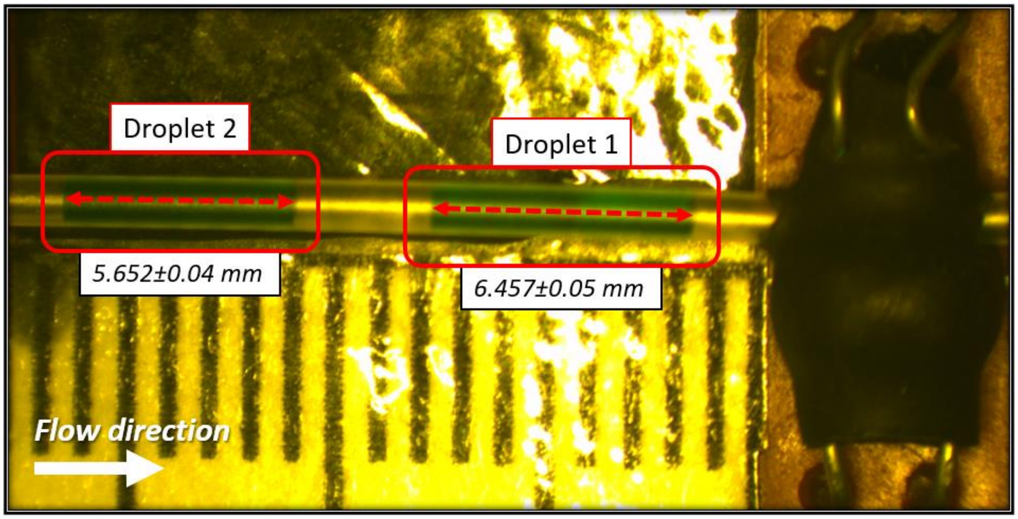


Figure 45. Random dyed-water droplets for calibration testing.

Table 10 describes the results for this test in which all parameters are measured velocity, flow rate, length, and volume with its respective relative error. The velocity and flow rate results exhibit a non-significant MRE increase of 7.6% from 6.13% obtained in the calibration process. Nevertheless, the measured lengths and volume errors are increased to 15.23%; this may suggest that the sensor outcomes deliver more accurate results for velocity and flow rate measurements, as shown from previous experiments this increased error values in length and volume measurements.

Table 10. Calibration process testing results for dyed-water droplets.

	Droplet 1			**Droplet 2**			**MRE**
	Sensor's results	**Expected**	**Relative Error**	**Sensor's results**	**Expected**	**Relative error**	
Velocity (mm/s)	8.9031	8.2596	7.79%	8.8711	8.2596	7.40%	7.60%
Flow rate (µL/min)	243.6089	226.0000	7.79%	242.7325	226.0000	7.40%	7.60%
Length (mm)	7.4202	6.4570	14.92%	6.5301	5.6520	15.54%	15.23%
Volume (µL)	3.3839	2.9446	14.92%	2.9780	2.5775	15.54%	15.23%

4.2 Three-phase flow experimental tests.

To test the device for multiple phases, we designed an experiment utilizing pure and dyed water as dispersed phases and air as the continuous phase. The droplets are described in Figure 46, which lengths are 5.198 mm for the pure-water droplet and 4.883mm for the dyed-water droplet. The measurements were extracted using ImageJ software. We used 100 and 500 µL/min constant flow rates for this test's calibration process and their theoretical data as references for comparison.

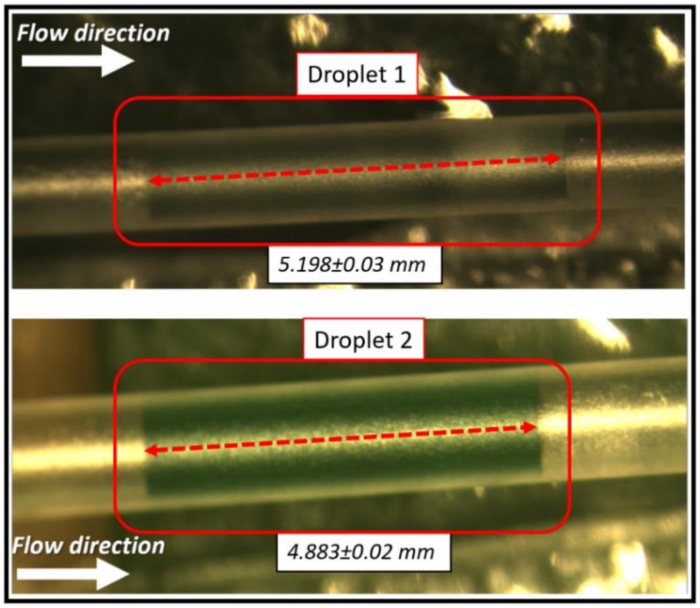


Figure 46. Three-phase flow experimental droplets.

The recorded signal at 100 µL/min is described in Figure 47, showing lower voltages for the pure-water (PW) phase and higher values for dyed-water (DW), as expected from the previous experiments and the Figure 2. The dispersed phases voltages are 1.7205 V for PW denoted by Vd_{PW} and 1.789 V for DW phase denoted by Vd_{DW}. Additionally, the continuous phase voltage (air) is 1.754 V, represented by Vc_{Air}.

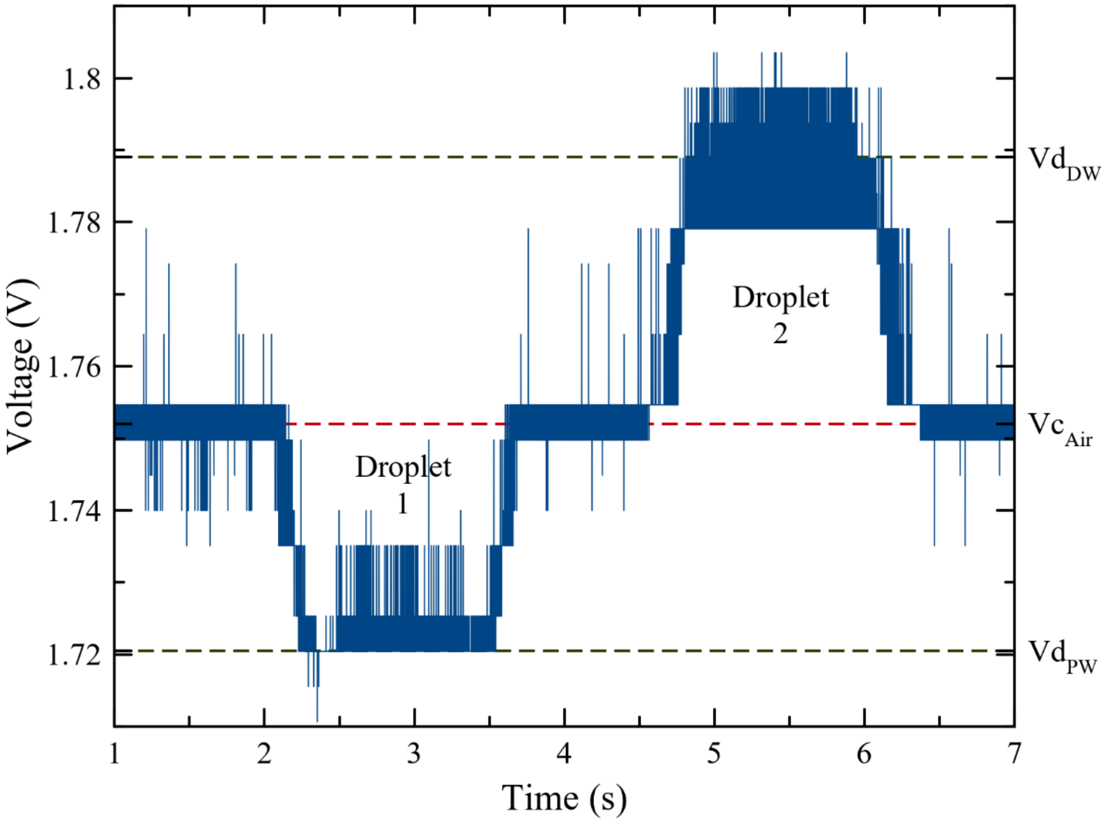


Figure 47. Signal recording for three-phase flow experiment at 100µL/min constant flow rate.

The signal filtering process follows the same procedure as two-phase flow experiments, with a slight increase of the smoothing factor than previous experiments, using a 0.15 value for optimal results. describes the filtered signal from Figure 48 at 100uL/min, indicating the measurements' relevant regions.

In contrast to the two-phases experiments, the signal local maxima and minima extraction has presented difficulties in finding all the required points for the algorithm calculations. However, by using the MATLAB user's interface, we could extract these points manually and import them to our workspace to assess the experiment.

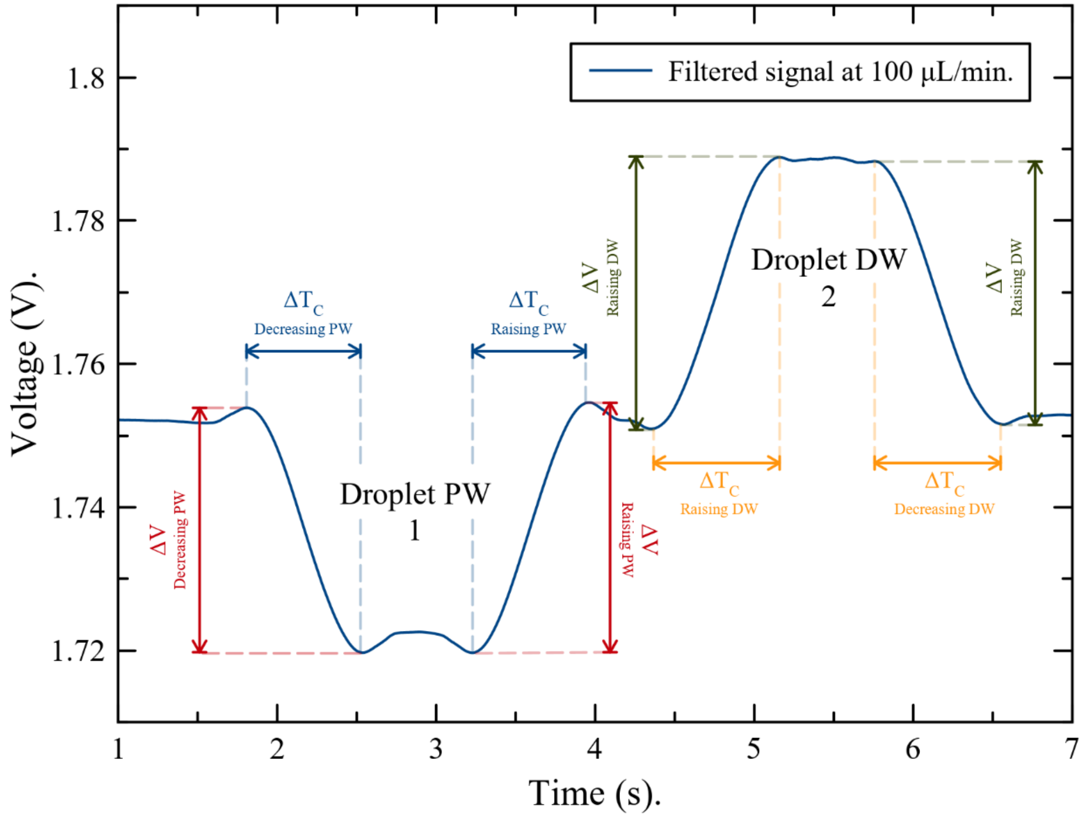


Figure 48. Three-phase flow filtered signal at 100 µL/min constant flow rate.

The interfacial distance (λ) was calculated using the same procedures as previous tests after importing the local maxima and minima. Thus, Table 11 describes the λ experiments' computed values at 100 and 500 µL/min flow rates extracted for a decreasing and raising signal. These results show values gathering around 2.5 mm approximately, implying a non-significant difference between pure-water λ and dyed-water λ.

Table 11.λ values calculated for three-phase flow experiments.

Experiment	Q (µL/ min)	Droplet	λ Decreasing (mm)	λ Raising (mm)
1	100	1	2.5605	2.6292
		2	2.9252	2.8514
2	500	1	2.0685	2.0028
		2	2.2988	2.3061

Therefore, we calculated the final λ value by using the improved procedure previously explained. Figure 49 describes the boxplot for the λ's obtained in Table 11, showing symmetric results without outliers. Furthermore, we extracted the IQR for the final distribution fitting presented in the right panel of figure. After the bootstrapping simulation, the λ value obtained for these three-phase flow experiments is **2.4485 mm**.

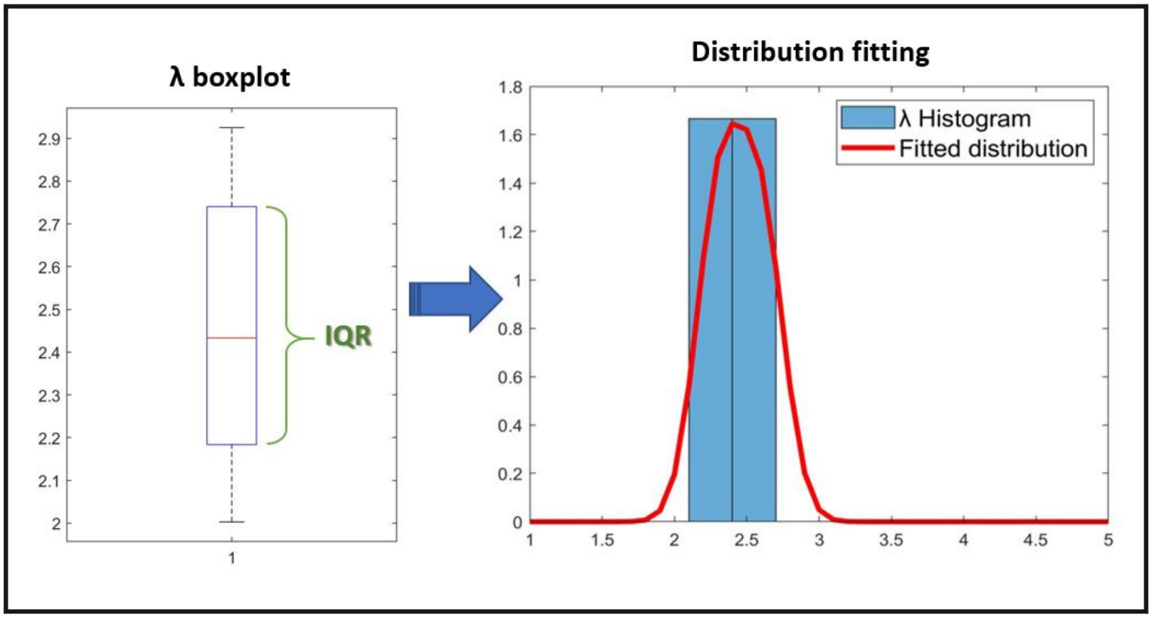


Figure 49. λ boxplot, IQR selection, and final fitted distribution for three-phase flow experiments.

To assess this calibration process, we compare the measurements obtained from the sensor with theoretical reference values. Figure 50-A describes the velocities and flow rate results comparison, showing an increased error for the experiment at 500 μL/min. Thus, Figure 50-B shows the MRE values for each experiment at 100 and 500 μL/min, respectively, which confirms a higher error at 500 μL/min with 13.33%, compared with 10.42% at 100 μL/min. The total MRE for velocities and flow rate measurements for these tests is 11.87%.

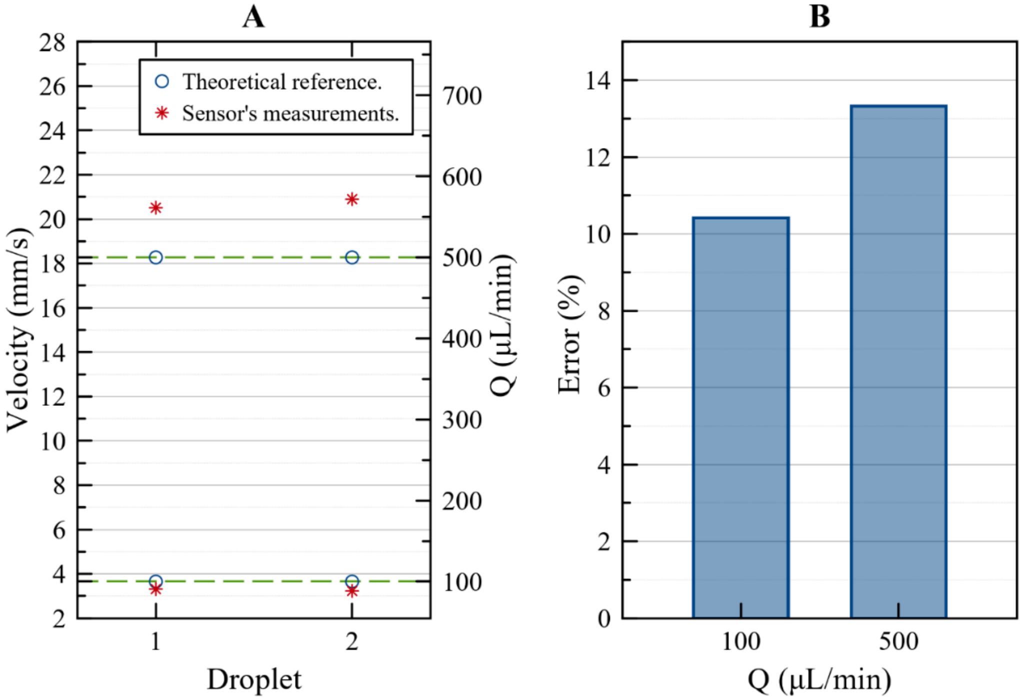


Figure 50. Three-phase flow velocities and flow rate measurements. A) Expected vs measured values. B) MRE values in % for each experiment.

A similar approach was used for droplet lengths and volume measurements. Figure 51-A describes the expected vs. measured results, which follows a similar behavior from previous results showing higher errors at the highest flow rate tested. Therefore, each experiment's MRE values are shown in Figure 51-B, revealing an 18.40% error at 500 µL/min and an 8.34% error at 100 µL/min. The final MRE value for these measurements is 13.37%.

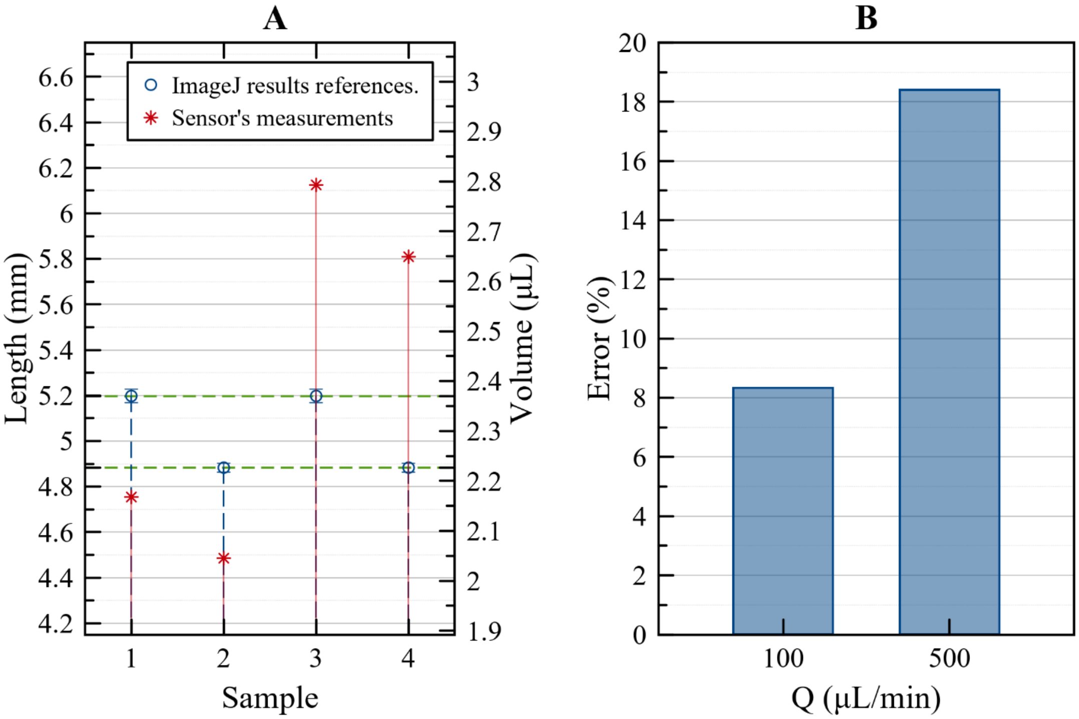


Figure 51. Three-phase lengths and volume measurements. A) Expected vs measured values. B) MRE values in % for each experiment.

We performed a further experiment using two random droplets using pure and dyed water at a 250 µL/min flow rate to test this calibration result. Figure 52 describes the 6.151 mm pure-water and 5.481 mm dyed-water droplets used for this test. The lengths were calculated using ImageJ software.

The calibration assessment results are shown in Table 12, which describes all the measurements performed with their respective MRE compared with expected values. Velocities and flow rate results exhibit increased MRE values of approximately 17%, with a final 17.37% MRE for both droplets. Similar results are shown for lengths and volume measurements, with a 16.75% MRE final value, due to a 19.04% error for pure-water and a 14.46% for the dyed-water droplet. Additionally, the total number of droplets detected by the sensor confirms the presence of the two droplets tested.

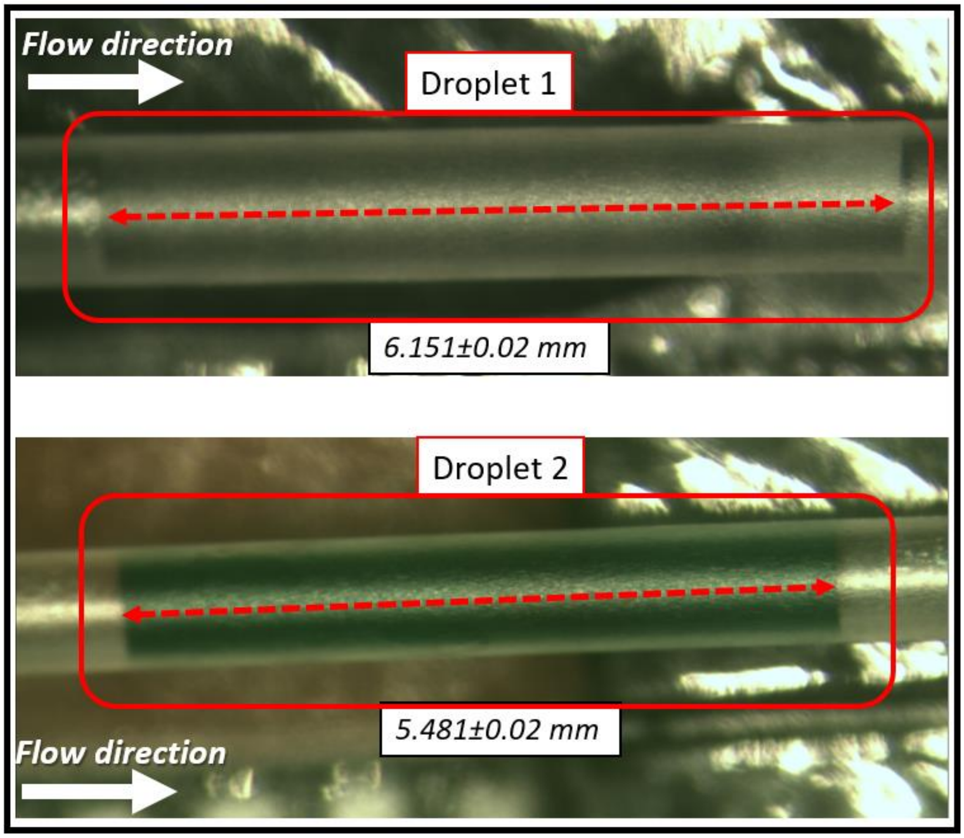


Figure 52. Pure-water and dyed-water droplets for calibration testing.

Although these outcomes show higher MRE values than two-phase flow measurements, they reveal promising results for multiple-phase flow characterization. With minor improvements, these MRE values can be reduced to obtain more accurate and precise measurements.

Table 12. Three-phase flow calibration testing results.

	Droplet 1			**Droplet 2**			**MRE**
	Sensor's results	**Expected**	**Relative Error**	**Sensor's results**	**Expected**	**Relative error**	
Velocity (mm/s)	7.5157	9.1367	17.74%	7.5833	9.1367	17.00%	17.37%
Flow rate (µL/min)	205.6450	250.0000	17.74%	207.4957	250.0000	17.00%	17.37%
Length (mm)	4.9798	6.1510	19.04%	4.6886	5.4810	14.46%	16.75%
Volume (µL)	2.2710	2.8051	19.04%	2.1382	2.4995	14.46%	16.75%
Number of droplets detected	2						

Finally, the mean relative errors reported for individual experiments throughout this chapter, and chapter 3 describes a random behavior compared to the hypothesis that the higher the flow rate, the higher the error. This response's analysis was related to the bootstrapping simulation due to its randomness when sampling from the data loaded. Therefore, when the interfacial distances calculated are loaded into the simulation, the process could randomly select more data from either slow or high rates. Although this procedure could increase the error in some individual experiments, the global MRE values after the calibration process for many experiments will not vary significantly due to the 1000 random samples taken from λ data loaded, which values are approximately equal. On the other hand, the local maxima and minima algorithm could introduce errors in some individual experiments due to minor mismatches within the signal peaks (maxima or minima).

Chapter 5

Conclusions and future work.

In this work, we developed a new optical flow sensor for lab-on-a-PCB devices, integrating microfluidics and electronics into a single portable, affordable, user-friendly, and autonomous system capable of detecting and characterizing droplets velocities, flow rate, lengths, and volume.

The use of printed circuit board PCB techniques reduced the final prototype's fabrication costs using common standard components, such as LED and LDR, as the operation principle and peripherals that enabled the compatibility with the ARDUINO microcontroller. Additionally, these elements ease the fabrication process in a way that there is no requirement for sophisticated laboratories or equipment. Therefore, the final prototype follows a standardized fabrication process developing an affordable, portable, and easy-to-fabricate device with a 38 by 55 mm working area and a 20 USD fabrication costs, including the microcontroller.

The ARDUINO UNO microcontroller demonstrated excellent performance in terms of high-speed data processing and easy-to-use platform due to its wide range of libraries and its compatibility with MATLAB, promoting a software blending. This software combination of ARDUINO IDE and MATLAB enhanced the system's communication with the macroworld and developed a tunable plug-and-play environment.

The algorithm's improvements showed an increased sampling rate, from 2.3 kHz using a standard configuration up to 5 kHz configuring the ATmega328P ADC registers. Thus, the sensor's throughput can detect and characterize 1000 droplets per second with a detection time of 0.2 milliseconds per droplet. These results confirm superior or similar detection speeds compared with state-of-the-art devices: [14] describes a sampling rate of 32.2 Hz, [15] presents a 7200 Hz device, and [12] which system throughput is 12 droplets/second.

Therefore, integrating the fabrication process and algorithms into a single device developed the final portable, affordable, easy-to-use, autonomous device with a plug-and-play environment for the alignment with the ASSURED criteria. The experimental tests present promising results for detecting and characterizing multiple dispersed phases in one microchannel using a single device, establishing a remarkable difference between state-of-the-art systems.

Pure-water-in-air droplets detection and characterization revealed MRE values of 2.4% for velocities and flow rates at 2.3 kHz sampling rate, increasing up to 7.92% for higher sampling rate (5 kHz) and flow rates (70 to 870 µL/min). Droplet length and volume measurements started with an MRE value of 8.04%, with a slight increase of 10.85 % for the fastest sampling rate (5 kHz) and higher flow rates (70 to 870 µL/min). These results demonstrate a reasonable accuracy for multi-parameter measurements delivered by a single device, with similar MRE values compared with state-of-the-art systems, as presented in [18] with approximately 6%, and [12] that presented errors of 8%.

Dyed water was used to change the dispersed phase color to assess the sensor's performance for a darker fluid. The results presented an MRE of approximately 6% for velocities and flow rate (100 and 500 µL/min) measurements and 8.65% for lengths and volume at a sampling rate of 5 kHz. These results were tested by measuring new random dyed-water droplets at a new constant flow rate at 226 µL/min showing a similar 7.6% MRE value for velocities and flow rates and an increased 15.23% for lengths and volume also at a sampling rate of 5 kHz. Although the velocities and flow rate errors follow the accuracy found with pure-water droplets, the increased error in lengths and volume measurements could be due to a certain degree of hydrophilicity of the microchannel walls reshaping the droplets while flowing.

Finally, to assess the most complicated scenario, a three-phase flow was tested combining pure and dyed water as dispersed phases while keeping the air as the continuous phase. Although the final MRE values found were around 17% for velocities, flow rate (250 µL/min), lengths, and volume measurements at a sampling rate of 5kHz, these errors present promising results for measuring multiple characteristics of multiple phases in a single sensor, which significantly stands out from any other device shown in the state-of-the-art.

Droplet detection and characterization showed MRE values ranging from 2.4% up to 17%. The lowest MRE value was obtained using a two-phase flow system with water-in-air droplets at a sampling rate of 2.3 kHz for flow rates starting at 20 up to 425 µL/min. In contrast, the highest MRE value reported was under a three-phase flow system for dyed and pure water-in-air droplets at a 5 kHz sampling rate at a 250 µL/min flow rate.

In comparison with state-of-the-art devices, the sampling rates achieved in this work are approximately 156 times faster than [14], and up to 70% of the 7.2 kHz reported in [15]. Consequently, the flow rates measured in our experiments were approximately 156 faster compared to [14], ten times faster than [18], and 500 times faster than [13]. Moreover, the throughput of our device (1000 droplets per second) is 83 times faster than [12]. Likewise, the detection, counting, and characterization demonstrated in this work are for multi-phase flow systems, compared with [12]–[15], [17]–[19]; in which results are only reported for two-phase systems. Finally, the errors reported in this work are comparable with [18] and [12].

Considering all the sensor's parameters and characteristics, we can conclude that it has promising results in many fields such as drug delivery, micromixing detection, considering some improvements in the microchannels, and reducing the sensing area. However, the current state of the prototype with minor adjustments in the microchannels could be implemented in the application of detecting hydraulic oil contamination as described in [39], in which the flow rates used (40 µL/min up to 640 µL/min) are comparable to the reported throughout this work, for droplets volume of 0.215 µL to 0.305 µL.

Therefore, this new optical flow sensor for Lab-on-PCB devices demonstrates a high-speed throughput for the detection, counting, and multi-feature characterization of droplets in multiple-phase flow systems. In addition, the device fulfills the ASSURED criteria and is compatible with printed circuit boards' fabrication processes and use standard electronic components. The device’s modularity enables to incorporate new microfluidic components based on soft lithography, with further reduction of the device dimensions.

5.1 Future work.

This device's development opens a new atmosphere for improvements, from algorithms to hardware, in benefit of its accuracy, precision, and miniaturization. The use of an SMD LDR could drastically reduce the sensors' sensing area, increasing its accuracy for smaller droplets detection and characterization. Additionally, the device's modularity enables the design of a smaller microchannel embedded into the sensor's PCB for a specific application; an example of this new design is presented in Appendix C.

As described from the three-phase flow assessment, the algorithm will require a revision to detect local maxima and minima for keeping a fully automated process. In addition, an improved algorithm for λ calculations with specific functions for darker and more transparent phases could potentially decrease the MRE values.

Further improvements on the device could replace the SMD LED with different light sources with different wavelengths up to the range of UV or IR. The implications of changing the light source of the device's operation principle will depend on the wavelength region used; whenever this falls in the visible spectrum, this will not represent significant changes in the sensor and could enabled colorimetric analysis. However, for using UV or IR sources it will be necessary to replace the LDR with their specific receptors. Therefore, this generates further implications in the device's interfacial distance and sensing area; nevertheless, it could add some degree of material characterization features embedded in the system based on fluorescence and spectroscopic methods, respectively.

Appendix A

Arduino IDE code.

```
Aint button1=2;
int button2=6;

int button1State=HIGH;
int button2State=LOW;

int ledRec=10;
long numSamples=0;
long t, t0;
char buf[16];
void setup()
{
 Serial.begin(250000);
 //SET ADC PARAMETERS..................
 ADCSRA = 0;            // clear ADCSRA register
 ADCSRB = 0;            // clear ADCSRB register
 ADMUX |= (2 & 0x07);    // set A2 analog input pin
 ADMUX |= (1 << REFS0);   // set reference voltage (REFS0=0 & REFS1=0 ->
AREF=EXTERNAL)
 ADMUX |= (0 << REFS1);  // set reference voltage
 ADMUX |= (0 << ADLAR);  // left align ADC value to 8 bits from ADCH register

 // sampling rate is [ADC clock] / [prescaler] / [conversion clock cycles]
 // for Arduino Uno ADC clock is 16 MHz and a conversion takes 13 clock cycles
 //ADCSRA |= (1 << ADPS2) | (1 << ADPS0);    // 32 prescaler for 38.5 KHz
 //ADCSRA |= (1 << ADPS2);                   // 16 prescaler for 76.9 KHz
 ADCSRA |= (1 << ADPS1) | (1 << ADPS0);    // 8 prescaler for 153.8 KHz
 //ADCSRA |= (1 << ADPS1);    // 4 prescaler for - KHz

 ADCSRA |= (0 << ADATE); // disable auto trigger
 ADCSRA |= (0 << ADIE);  // disable interrupts when measurement complete
 ADCSRA |= (0 << ADEN);  // disable ADC
 ADCSRA |= (0 << ADSC);  // stop ADC measurements

 //SET I/O PARAMETERS........................
 pinMode(button1, INPUT_PULLUP);
 pinMode(button2,INPUT);
 pinMode(ledRec,OUTPUT);
 digitalWrite(ledRec,LOW);
 numSamples=0;
}

ISR(ADC_vect)
{
 //byte x = ADCH;  // read 8 bit value from ADC
 uint16_t x = ADC;
 //Serial.println(x);
 Serial.println(itoa(x, buf, 10));
 numSamples++;

}

void loop()
{
 button1State = digitalRead(button1);
 button2State = digitalRead(button2);
 //Serial.println(numSamples);

 if(button1State==LOW && digitalRead(ledRec)==LOW)
 {
   digitalWrite(ledRec,HIGH);
   ADCSRA |= (1 << ADATE); // enable auto trigger
   ADCSRA |= (1 << ADIE);  // enable interrupts when measurement complete
   ADCSRA |= (1 << ADEN);  // enable ADC
   ADCSRA |= (1 << ADSC);  // start ADC measurements
   t0=micros();
   numSamples=0;
 }
 if (button2State==HIGH && digitalRead(ledRec)==HIGH)
 {
   digitalWrite(ledRec,LOW);
   ADCSRA = 0;             // clear ADCSRA register
   ADCSRB = 0;             // clear ADCSRB register
   ADCSRA |= (0 << ADATE);
   ADCSRA |= (0 << ADIE);
   ADCSRA |= (0 << ADEN);  // disable ADC
   ADCSRA |= (0 << ADSC);  // stop ADC measurements

   t = micros()-t0;  // calculate elapsed time

   Serial.println(numSamples);
   Serial.println((float)numSamples*1000/t);
   Serial.println("stop");
   //Serial.println(" KHz");
   //delay(2000);
   // restart
   t0 = micros();
 }

}
```

Appendix B

MATLAB code

```
% Signal post processing
% Load Data
file = '100.mat';
load (file);
[filepath,name,ext] = fileparts(file);
q=str2num(name);
figure
plot(sT,sVolts);
title(name);
% Get sampling frequency.
dt = mean(diff(sT));
Fs = 1 / dt % Hz
Fs=ceil(Fs);
newFs=Fs;

%% SMOOTHING
z2 = smoothdata(sVolts,'sgolay','SmoothingFactor',0.15);
figure
% Get # of Droplets
nDrpts=getDropletsNumber(z2)
plot(sT,z2)
title(name);
hold on
%% EXTRACT PEAKS MANUALY
pts=vertcat(points.Position);
res=orderPeaksManually(pts,q);
%% Find peaks
mx=max(z2);
mn=min(z2);
center=mx-((mx-mn)/2);
% Top line
dp=mean(z2(50));
pk=mx-dp;
MpkDis=(sT(2)-sT(1))*2;
nPeaks=nDrpts*2;
[pks,locs]=findpeaks(z2,sT,'NPeaks',nPeaks+2,'MinPeakProminence',1
0e-10,'Threshold',9e-11,...
    'MinPeakDistance',MpkDis,'MinPeakHeight',center);
[pksn,locsn]=findpeaks(-
z2,sT,'NPeaks',nPeaks,'MinPeakProminence',10e-4,'MinPeakHeight',-
center);
plot(locs,pks,'*',locsn,abs(pksn),'*')
%plot(locsn,abs(pksn),'bo')
```

```
%% ORDER PEAKS CW
% Extract maximun and minimum points
% 1= High peak
% 0= Low peak
Max=[locs,pks];
Min=[locsn,abs(pksn)];
tagM=ones(length(Max),1);
tagm=zeros(length(Min),1);
Max=[Max,tagM];
Min=[Min,tagm];
%
res2=orderPeaksCW(Max,Min);
%% Add q
qs=[100 500];
qs=repelem(qs,6)';
sVec3=[sVec qs];
%%
sVec2=res;
%% Adding Data
sVec=[sVec;res];
%% ORDER PEAKS!
Max=[locs,pks];
Min=[locsn,abs(pksn)];

tagM=ones(length(Max),1);
tagm=zeros(length(Min),1);

Max=[Max,tagM];
Min=[Min,tagm];

res=orderPeaks(Min,Max);
% EXTRACTING PAIRS OF POINTS (DRECREASE RAISE of SIgnal)
Des=[];
Rais=[];
for i=1:length(res)-1
    if res(i,3)==1 && res(i+1,3)==0
        Des=[Des;res(i,1:2) res(i+1,1:2) q];
    elseif res(i,3)==0 && res(i+1,3)==1
        Rais=[Rais;res(i,1:2) res(i+1,1:2) q];
    end
end
SignalVec=[Des;Rais];
%% First Loop
sVec=SignalVec;
%% Adding Data
sVec=[sVec;SignalVec];
%% Lambda Estimatimation
%lambda2=estimateLambdaFunctn(res);
```

```
%lambda3=estimateLambdaFunctn(res);
lambda=estimateLambdaFunctn(sVec);
%lambda=aux;

%% Bootstrapping
bootLambda=bootstrp(2000,@mean,lambda);
uLambda=mean(bootLambda)
sLambda=std(bootLambda)
%figure;histogram(bootLambda);
%% CI
conf=0.95
Lb=quantile(bootLambda,(1-conf)/2)
Ub=quantile(bootLambda,1-((1-conf)/2))

%% Load Signal Parameters
load 'sVec.mat'
%% PARAMETERS
ndrops=2;
nExperiments=1;
Qs=250;
RealLengths=[6.151  5.481];
sVec=sVec2;
%% VELOCITIES CALCULATION
%Qs=100:400:500;
Qs = transpose(repelem(Qs,ndrops));
%
%Qs=[20;20;20]
Ao=pi*(0.762^2)/4;
Vt=(Qs./60)./Ao;
Vm=getVelocities(sVec,uLambda);
size=length(Vm)/2;
Vres=zeros(size,1);


nAns=ndrops*2;
index=0;

for i=1:nAns:length(Vm)
    for j=1:ndrops
        aux=index+j;
        Vres(aux)=mean([Vm(j+i-1) Vm(j+i-1+ndrops)]);
    end
    index=aux;
end
figure
plot(Vt,'*')
hold on
plot(Vres,'o')
```

```
hold off
legend('Vt', 'Vm')
grid

% Error
VelErr=abs(Vt-Vres)./Vt;
VelMRE=mean(VelErr)
figure;bar(VelErr)

% Calculate Mean Error per Experiment
c=length(VelErr);
ExpErr=[];
VelExpErr=[];
for i=1:ndrops:c
    for j=1:ndrops
        ExpErr=[ExpErr;VelErr(i+j-1)];
    end
    VelExpErr=[VelExpErr; mean(ExpErr)];
    ExpErr=[];
end
figure;bar(VelExpErr)
%% FLOW RATE
% Qm = Vm * Ao
Qm=(Vres.*Ao).*60;
% Errors
QErr= abs((Qm-Qs)./Qs);
QMRE= mean(QErr);
%
figure
plot(Qs,'*')
hold on
plot(Qm,'o')
hold off
legend('Qt', 'Qm')
grid
%% NEW METHOD DROPLET LENGTH
DTd=[];
for i=1:2:length(sVec)-1
    aux=abs(sVec(i+1,1)-sVec(i,3));
    DTd=[DTd;aux];
end

DlengthWT=DTd.*Vres;
DLenght=DlengthWT+uLambda;

% Comparing lenghts
vRealL=repmat(RealLengths,1,nExperiments)';
LengthErr=abs(vRealL-DLenght)./vRealL;
```

```
LengthMRE=mean(LengthErr)
% PLOT DROPLET LENGTH COMPARISSON
figure
stem(vRealL)
xlabel('Sample number')
ylabel('Length (mm)')
xticks(0:1:27);
xlim([0 29])
ylim([3 8])
hold on
stem(DLenght,'*')
hold off
legend('DLt', 'DLm')
grid
% Calculate Mean Error per Experiment
c=length(LengthErr);
ExpErr=[];
ResErr=[];
for i=1:ndrops:c
    for j=1:ndrops
        ExpErr=[ExpErr;LengthErr(i+j-1)];
    end
    ResErr=[ResErr; mean(ExpErr)];
    ExpErr=[];
end
figure;bar(ResErr)
%% CALCULATE DROPLET LENGTH
% Vm * DTc
DTd=[];
DTcD=[];
DTcR=[];
nAns=ndrops*2;
for i=1:nAns:length(sVec)
    if i<length(sVec)
        for j=1:ndrops
            Rf=sVec(j+i-1+ndrops,3);
            Ro=sVec(j+i-1+ndrops,1);
            Df=sVec(j+i-1,3);
            Do=sVec(j+i-1,1);
            aux = Ro - Df;
            DTd=[DTd;aux];
            DTcD=[DTcD;Df-Do];
            DTcR=[DTcR;Rf-Ro];
        end
    end
end
DlengthWT=DTd.*Vres;
DLenght=DlengthWT+uLambda;
```

```
% Comparing lenghts
vRealL=repmat(RealLengths,1,nExperiments)';
LengthErr=abs(vRealL-DLenght)./vRealL;
LengthMRE=mean(LengthErr)
% PLOT DROPLET LENGTH COMPARISSON
figure
stem(vRealL)
xlabel('Sample number')
ylabel('Length (mm)')
xticks(0:1:27);
xlim([0 29])
ylim([3 8])
hold on
stem(DLenght,'*')
hold off
legend('DLt', 'DLm')
grid
% Calculate Mean Error per Experiment
c=length(LengthErr);
ExpErr=[];
ResErr=[];
for i=1:ndrops:c
    for j=1:ndrops
        ExpErr=[ExpErr;LengthErr(i+j-1)];
    end
    ResErr=[ResErr; mean(ExpErr)];
    ExpErr=[];
end
figure;bar(ResErr)
%% Calculating Errors
%https://mathworld.wolfram.com/RelativeError.html
AbsErr = abs(DLenght-vRealL);

RelErr = (AbsErr./vRealL);
LengthMRE=mean(RelErr);
%%
AbsErr2 = abs(DLength2-vRealL);

RelErr2 = (AbsErr2./vRealL);
LengthMRE2=mean(RelErr2);
%% CALCULATE DROPLET VOLUME
VolT=vRealL.*Ao;
VolM=DLenght.*Ao;
VolErr=abs(VolM-VolT)./VolT;
VolMRE=mean(VolErr);
% Plot
figure
stem(VolT)
```

```
xlabel('Sample number')
ylabel('Volume (mm^3)')
xticks(0:1:30);
xlim([0 30])
ylim([1.5 3.5])
hold on
stem(VolM,'*')
hold off
legend('Vol_T', 'Vol_M')
grid

%% GET the IQR of Lambda
boxplot(lambda)
q = quantile(lambda,[0.25 0.75]);
iqrLambda = lambda(lambda>q(1) & lambda<q(2));
%% DISTRIBUTION FITTING
% Fit the Lamdba values to 'Normal' distribution
pd = fitdist(iqrLambda,'Kernel')
%% Check de fitting
x_pdf = [1:0.1:5];
y = pdf(pd,x_pdf);
figure
histogram(iqrLambda,'Normalization','pdf')
line(x_pdf,y,'LineWidth',3,'Color','r')
%legend('? Histogram','Fitted distribution')
legend({'? Histogram','Fitted distribution'},'FontSize',14)
%% Manual Bootstrapping + CI
conf=0.95;
n=1500;
v=zeros(1,n);
for i=1:n
    vaux=random(pd,1,12);
    v(i)= mean(vaux);
end
uLambda=mean(v)
Lb=quantile(v,(1-conf)/2)
Ub=quantile(v,1-((1-conf)/2))

function nDroplet = getDropletsNumber (signal)
% Extract differences between Min and Max
sDiff=(max(signal)-min(signal))/3;
% Signal Limits
lLimit=min(signal)+sDiff;
%cLimit=min(signal)+(2*sDiff);
uLimit=max(signal)-sDiff;
%Check crossing the limits by ZERO CROSSING
zcd = dsp.ZeroCrossingDetector;
numZeroCrossL = zcd(signal-lLimit);
```

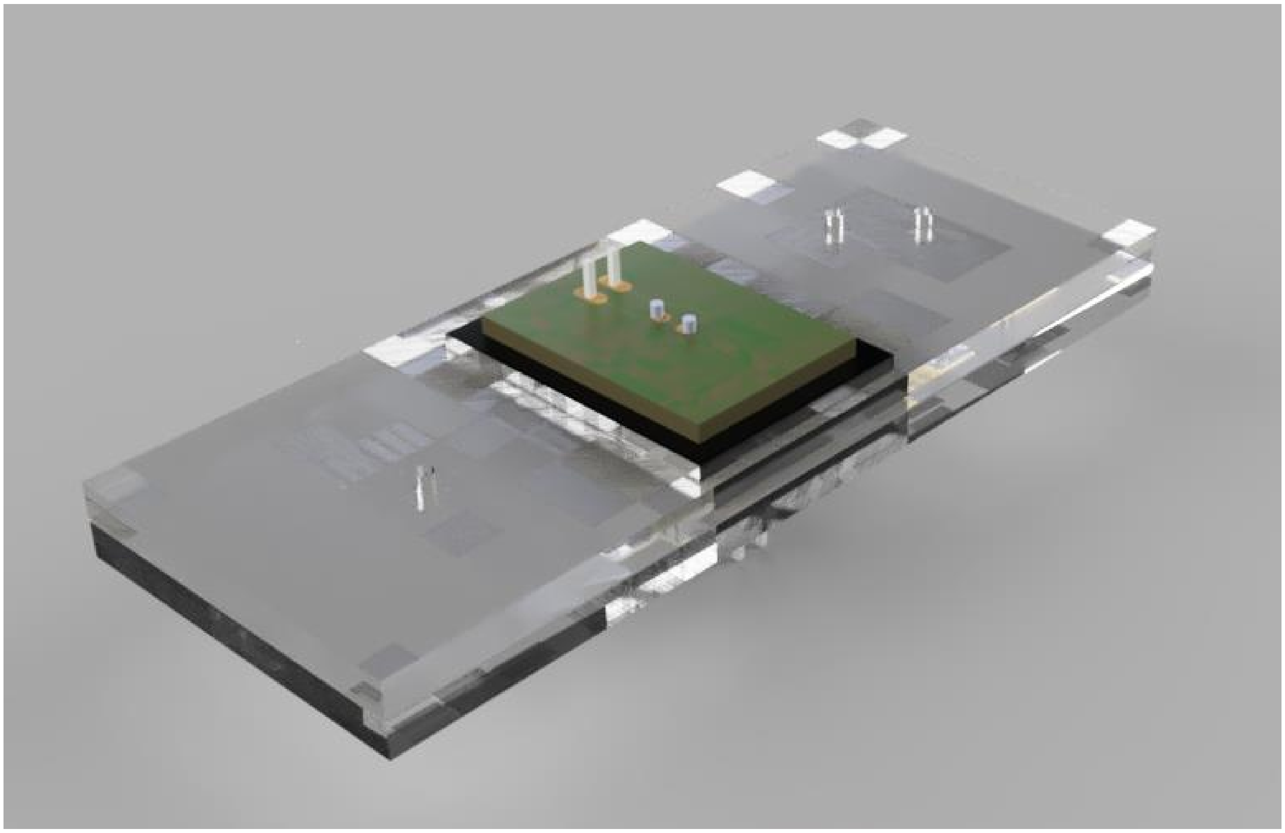
```
zcd = dsp.ZeroCrossingDetector;
numZeroCrossU = zcd(signal-uLimit);
nDroplet = (numZeroCrossL+numZeroCrossU)/2;

function lambda = estimateLambdaFunctn(vecPoints)
    Ao=pi*(0.762^2)/4;
    Vt=(vecPoints(:,5)./60)./Ao;
    dx=vecPoints(:,3)-vecPoints(:,1);
    dy=vecPoints(:,4)-vecPoints(:,2);
    m=dy./dx;
    %um=mean(m);
    rate=Vt./m;
    lambda=rate.*dy;
    %lambda=mean(lambda);
end

function vm = getVelocities(vecPoints,lambda)
    dt=vecPoints(:,3)-vecPoints(:,1);
    vm=lambda./dt;
end
```

Appendix C

Future design for a reduced microchannel and sensing area.



References

[1] G. M. Whitesides, “The origins and the future of microfluidics,” *Nature*, vol. 442, no. 7101, pp. 368–373, 2006.

[2] Y. Zhu and Q. Fang, “Analytical detection techniques for droplet microfluidics-A review,” *Anal. Chim. Acta*, vol. 787, pp. 24–35, 2013.

[3] Y. J. Chang and H. You, “Progress of Microfluidics Based on Printed Circuit Board and its Applications,” *Chinese J. Anal. Chem.*, vol. 47, no. 7, pp. 965–975, Jul. 2019.

[4] D. Moschou and A. Tserepi, “The lab-on-PCB approach: tackling the μTAS commercial upscaling bottleneck,” *Lab Chip*, vol. 17, no. 8, pp. 1388–1405, 2017.

[5] L. Duchesne and K. Lacombe, “Innovative technologies for point-of-care testing of viral hepatitis in low-resource and decentralized settings,” *J. Viral Hepat.*, vol. 25, no. 2, pp. 108–117, 2018.

[6] D. Evans *et al.*, “A Novel Microfluidic Point-of-Care Biosensor System on Printed Circuit Board for Cytokine Detection,” *Sensors (Basel).*, vol. 18, no. 11, pp. 1–15, 2018.

[7] P. Jolly, J. Rainbow, A. Regoutz, P. Estrela, and D. Moschou, “A PNA-based Lab-on-PCB diagnostic platform for rapid and high sensitivity DNA quantification,” *Biosens. Bioelectron.*, vol. 123, no. June 2018, pp. 244–250, 2019.

[8] F. B. Malcata *et al.*, “Point-of-care tests for bovine clinical mastitis : what do we have and what do we need ?,” 2020.

[9] M. Puiu and C. Bala, “Microfluidics-integrated biosensing platforms as emergency tools for on-site field detection of foodborne pathogens,” 2020.

[10] R. W. Peeling, K. K. Holmes, D. Mabey, and A. Ronald, “Rapid tests for sexually transmitted infections (STIs): The way forward,” *Sex. Transm. Infect.*, vol. 82, no. SUPPL. 5, 2006.

[11] Y. Zhu and Q. Fang, “Analytical detection techniques for droplet microfluidics-A review,” *Anal. Chim. Acta*, vol. 787, pp. 24–35, 2013.

[12] F. Mehmood, Z. Haider, and B. Yin, “A Pump Based Microfluidic Image Processing System for Droplet Detection and Counting,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 381, no. 1, 2018.

[13] N. T. Nguyen, S. Lassemono, and F. A. Chollet, “Optical detection for droplet size control in microfluidic droplet-based analysis systems,” *Sensors Actuators, B Chem.*, vol. 117, no. 2, pp. 431–436, Oct. 2006.

[14] Z. Yang, T. Dong, A. Jensen, and E. Halvorsen, “Integratable Capacitive Sensor for Identification of Microfluidic Two-Phase Flow Patterns in Lab-on-Chip Devices,” *J. Microelectromechanical Syst.*, vol. 25, no. 1, pp. 197–206, 2016.

[15] J. Panwar and R. Roy, “Integrated fusible alloy microelectrodes based microfluidic impedance cytometry for cell-in-droplet quantification,” *bioRxiv*, vol. 215, no. March, p. 278069, 2018.

[16] J. Li, Y. Wang, E. Dong, and H. Chen, “USB-driven microfluidic chips on printed circuit

boards," *Lab Chip*, vol. 14, no. 5, pp. 860–864, 2014.

[17] R. Payri, G. Bracho, J. Gimeno, and A. Moreno, "Investigation of the Urea-Water Solution Atomization Process in Engine Exhaust-Like Conditions," *Exp. Therm. Fluid Sci.*, vol. 108, no. March, pp. 75–84, 2019.

[18] S. Ma, J. M. Sherwood, W. T. S. Huck, and S. Balabani, "On the flow topology inside droplets moving in rectangular microchannels," *Lab Chip*, vol. 14, no. 18, pp. 3611–3620, 2014.

[19] W. S. Zhang, Y. N. Wang, Y. Wang, and Z. R. Xu, "Highly reproducible and fast detection of 6-thioguanine in human serum using a droplet-based microfluidic SERS system," *Sensors Actuators, B Chem.*, vol. 283, pp. 532–537, Mar. 2019.

[20] I. The Mathworks, "MATLAB (R2020a) Update 1." Natick, Massachusetts, 2020.

[21] ARDUINO.CC, "Arduino Software (IDE)." 2020.

[22] Arduino.cc, "Arduino UNO Rev 3," 2020. [Online]. Available: https://store.arduino.cc/usa/arduino-uno-rev3.

[23] Autodesk, "EAGLE education." 2020.

[24] Kingbright, "APT3216SGC SMD Chip LED Lamp." pp. 15–18, 2020.

[25] I. KOA Speer Electronics, "General purpose metal film leaded resistor." pp. 144–145, 2019.

[26] TOKEN, "CDS Light-Dependent Photoresistors." 2010.

[27] L. UNISONIC TECHNOLOGIES CO., "BC556/557/558." pp. 1–6, 2012.

[28] Kingbright, "WP7113ID T-1 3 / 4 (5mm) Solid State Lamp.," vol. 4. pp. 2–6, 2019.

[29] Autodesk, "Fusion 360, Student." 2020.

[30] I. Mircochip Technology, "megaAVR Data Sheet." pp. 1–662, 2018.

[31] A. Savitzky and M. J. E. Golay, "Smoothing and Differentiation of Data by Simplified Least Squares Procedures," *Anal. Chem.*, vol. 36, no. 8, pp. 1627–1639, 1964.

[32] I. The Mathworks, "Savitzky-Golay filter design," 2006. [Online]. Available: https://es.mathworks.com/help/signal/ref/sgolay.html?lang=en.

[33] I. The Mathworks, "smoothdata Documentation," 2016. [Online]. Available: https://www.mathworks.com/help/matlab/ref/smoothdata.html.

[34] I. The Mathworks, "Findpeaks function MATLAB.," 2007. [Online]. Available: https://es.mathworks.com/help/signal/ref/findpeaks.html?lang=en.

[35] W. Navidi, *Statistics for Engineers and Scientists*, 5th ed. New York: Mc Graw Hill Education, 2020.

[36] I. New Era Pump Systems, "NE-300 Extruder Manual." p. 10, 2016.

[37] U. Rasband, Wayne;National Institutes of Health, "ImageJ 1.53e." 2020.

[38] I. The Mathworks, "MATLAB - Kernel Distribution," 2020. [Online]. Available: https://www.mathworks.com/help/stats/prob.kerneldistribution.html.

[39] C. Bai, H. Zhang, L. Zeng, X. Zhao, and Z. Yu, "High-Throughput Sensor to Detect

Hydraulic Oil Contamination Based on Microfluidics,” *IEEE Sens. J.*, vol. 19, no. 19, pp. 8590–8596, Oct. 2019.

Curriculum Vitae

Daniel Hugo Solano Teran was born in Cochabamba, Bolivia, in 1993. He received a B.S. degree in mechatronics engineering from the "Universidad Catolica Boliviana San Pablo" university. He was accepted in the master's degree programs in Nanotechnology at Tecnológico y de Estudios Superiores de Monterrey, Monterrey Campus in January 2019.