Heinz-Nixdorf-Chair of Biomedical Electronics
Department of Electrical and Computer Engineering
Technical University of Munich

Microfuidic Pressure Pump

Lab Work Report

Alaeddine Yacoub
Daniel Kühebacher
Oumaima Zneidi

Lab Work at
Heinz-Nixdorf-Chair of Biomedical Electronics of the Technical University of Munich

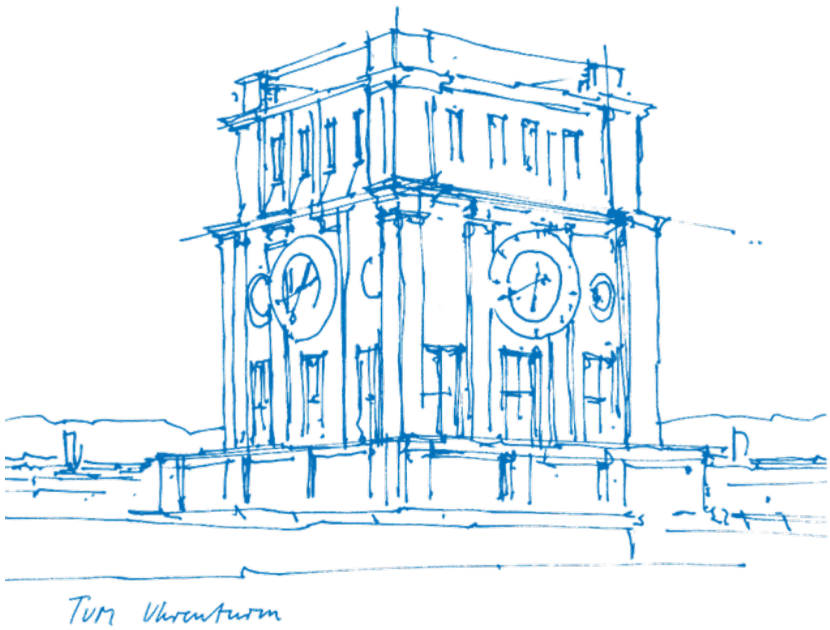
Examiner:
Prof. Dr. rer. nat. Oliver Hayden
Heinz-Nixdorf-Chair of Biomedical Electronics

Supervised by:
Mathias Reisbeck
Christian Klenk

Submitted by:
Alaeddine Yacoub: 03675497
Daniel Kühebacher: 03730933
Oumaima Zneidi: 03675592

Lab Duration
02.11.2020- 31-03-2021

Report Submitted on:
15.04.2021



Contents

1 Introduction

1.1 Motivation

Microfluidics is continuously growing to become a widely used technique incorporating high technology solutions in various fields such as biology and biochemistry. Cells testing, disease diagnosis and other applications are now of a big importance, for this purpose is the development in the field of microfluidics having a major technological priority within scientists and engineers.

The concept of Lab-on-Chip is not only offering an efficient but also a sophisticated method for the purpose of transporting fluidics. As the scientific field knows two major types of pumps, the flow-controlled pump has been used as a conventional engineering solution for the fluid displacement. However, another recently used type, pressure-controlled pump, is taking over the market due to its precision and satisfying results using the hydrostatic pressure.

The microfluidic control system that is built in this project will only necessitate a pressed-air source a power source and a computer to control the system. The microcontroller, that is integrated in the device, is connected to the computer via USB and allows the user to control the pump precisely and with minimal time delay. The pressure controller can be used as stand-alone device in the presence of the pressure source and allows the user a very flexible usage.

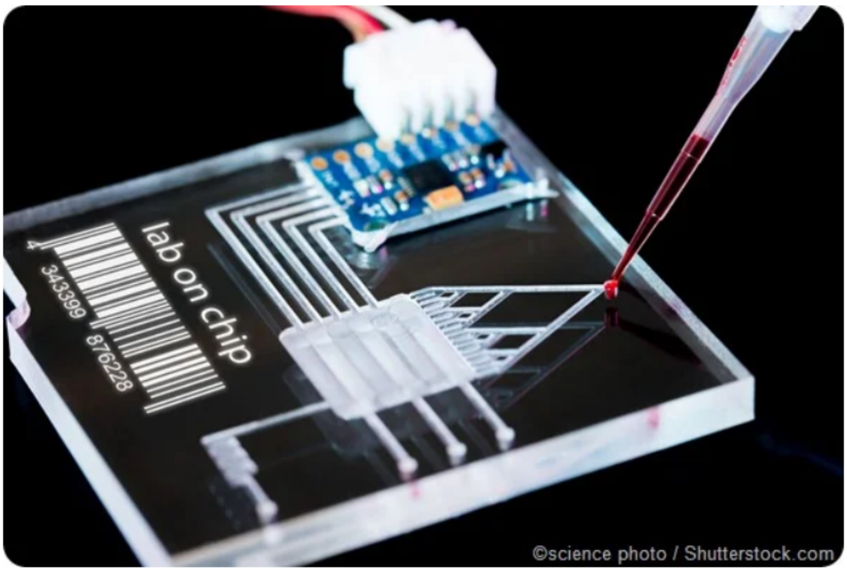


Figure 1.1 Lab-On-Chip [1]

1.2 Outline of the Work

The given tasks of the lab work was to develop a low cost microfluidic pump. Since two student teams worked on this tasks, two different concepts were implemented. Our specific task was to develop a pressure driven pump, which is capable of transporting fluids at a minimum flow rate of 0, $1\mu l/min$ to a maximum flow rate of $1ml/min$. This should be tested with a single microfluidic channel with the size l/w/h = 30 mm/ 0.25 mm/ 0.1 mm. The project schedule is presented chronologically in the following list.

1. Concept of the project
   a) Purpose of the project
   b) Theoretical research on scientific papers
   c) Mapping of the devices available in the market
   d) System concept
2. Build of the device
   a) Listing necessary hardware tools
   b) Consideration of key data, costs and form factor
   c) System development and design
   d) System integration and comissioning
3. System testing
   a) Software integration: GUI to monitor the system
   b) System characterization : stability, reproducibility, linearity
   c) Documentation of the project

2 Related Work

As the aim was to provide a low cost product that delivers precise results, a very detailed theoretical study on scientific papers was prepared to understand the basics of physics in microfluidics and available technologies that will be used as well as the necessary electronics needed for the pressure controlled pump.

The first scientific paper [2] represents a fully programmable and cost effective with low-cost components pressure source, which is composed of hardware and software parts that can be adapted to any application in the field of microfluidics as the implementation was flexible and simple. In this study, various parameters such as the accuracy, precision and noise were compared to the characteristics of a commercial pumping system. One of the applications within it proved its adaptability was a droplet Microfluidics, as water-in-oil droplets were generated over a series of multiple ratios. In this project an industrial pressure controller with an output pressure range from 0.01 mbar to 1000 mbar was used. The pressure controller had to be provided with an electrical signal (0 V to 5 V) generated by a fieldbus system. The computer was connected to the fieldbus system through ethernet and a python based software which allowed the user to monitor the whole system using a graphical user interface.

The second scientific work [3] is a thesis that also focused on building a pressure-driven pump. As the figure 2.1 shows, the hardware components that were used were an Arduino Uno as a microcontroller, an SMC Pressure Sensor (PSE532 – M5), an SMC valve (SY114-6LZ), a power supply, a reservoir, a pressure source and a touchscreen to monitor the system. As the Arduino made it possible to work with both of the pressure sensor and touchscreen due to its operating voltage, it also had to be powered through its power jack instead oft he USB connection with the computer. For the functionality of the pumping system, the pressure sensor sends the measured pressure value in the reservoir to the Arduino. The reservoir is connected from one side to a solenoid valve fort he pressure control and tot he output of pump from the other side.

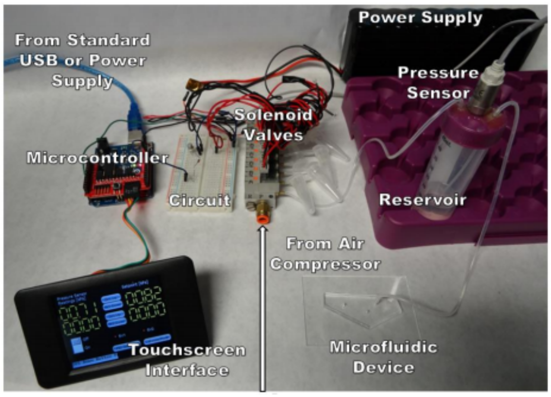


Figure 2.1 Device setup [3]

As this project aims to provide a performant microfluidic pumping system, an overlook on the market has provided us with multiple industrial examples such as the MFCS-EZ of the company Fluigent [4] that offers an easy-to-use microfluidic flow controller as it produces a constant pressure-controlled flow rate that enables reliable and repeatable experiments.

3 Concept

3.1 Requirements

The hydraulic resistance and pressure Δp of the system resulting from the given flow rates and dimensions of the microfluidic channel can be estimated using the following equation [5]:

$$Q = \int_0^w \mathrm{d}y \int_0^h \mathrm{d}z \frac{\Delta p}{2\eta L}(h - z)z = \frac{h^3 w}{12\eta L}\Delta p \tag{3.1}$$

However, with our aspect ratio being equal to h/w = 2/5 this equation delivers an inaccurate estimation by 33.6 %. Therefore we based our calculations on the more accurate following equation :

$$Q \approx \frac{h^3 w \Delta p}{12\eta L}\left[1 - 0.630\frac{h}{w}\right] \tag{3.2}$$

3.2 Calculations

Knowing that the needed results depend on the products that will be used, calculating the maximal and minimal flow rates that are expected to be delivered was necessary.

$Q_{\min} = 0,1_{\mu} l/\min = 1,67 \cdot 10^{-12} \mathrm{m}^3/\mathrm{s}$

$Q_{\max} = 1\mathrm{ml}/\min = 1,67 \cdot 10^{-8}\ \mathrm{m}^3/\mathrm{s}$

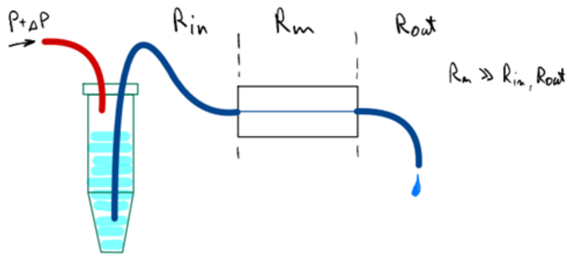


Figure 3.1 Hydraulic resistance

The calculation of the hydraulic resistance has been given through this equation:

$R_{hyd} = 12\eta\ell \cdot \frac{1}{h^3\omega} = 1,44 \cdot 10^{12} \frac{\mathrm{Pa.s}}{\mathrm{m}^3}$

After determining both of the maximal (minimal) flow rate and the hydraulic resistance, the next step was to calculate the needed maximal (minimal) pressure. For the matlab code all three Dimensions of the microfluidic channel: length = 30 mm/ width = 0.25 mm/ height = 0.1 mm were needed for the calculation of the hydraulic resistance. The maximal flow rate (1 ml/min) and minimal (0.1 µl/min) were already given for the further maximal and minimal pressure calculation.

$\Delta p = R_{hyd} \cdot Q$

$\Delta p_{\min} = R_{hyd} \cdot Q_{\min} = 2,4048 P_a$

$\Delta p_{\max} = R_{hyd} \cdot Q_{\max} = 24048 P_a$

3.3 Design

As the first figure 3.2 shows, the first concept consisted in using the industrial pressure regulator (ITV0010-0BL, SMC) which requires an input signal 4-20 mA and has to be connected with an external source for its input pressed air, it can also provide an output pressure range from 10 mBar to 1 Bar, the pressure value will be then read by a pressure sensor with the aim of having a more performant regulation cycle. The pressure sensor output signal will then be delivered to the Arduino Microcontroller allowing it to send accordingly the necessary signals to the pressure controller. A graphical user interface GUI will then be used to ensure the communication between the Arduino and the user. The second concept on the other side, as shown in the figure 3.3, consisted of using two PVQ-30 solenoid valves that can be controlled proportionally instead of the ITV0010-0BL pressure regulator. Taking the complexity of the second solution in consideration as it demanded working on two components instead of only one and also the wider range that the scalability of the pressure regulator offers as well as more promising and accurate results it was decided that the first concept was the best suited system for this project.

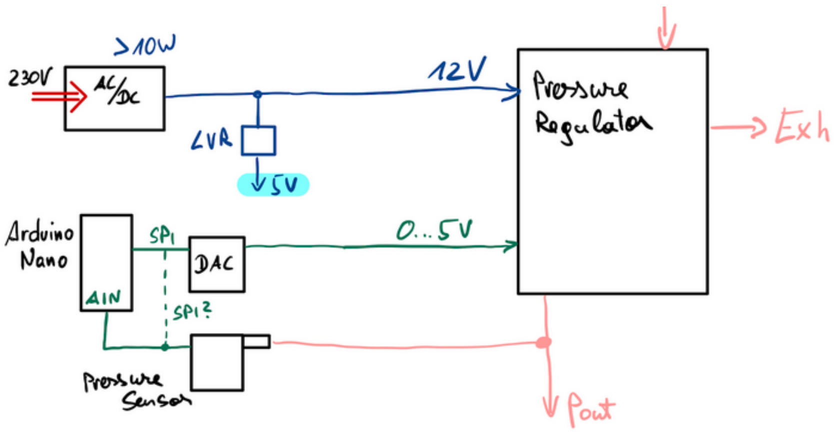


Figure 3.2 First Concept.

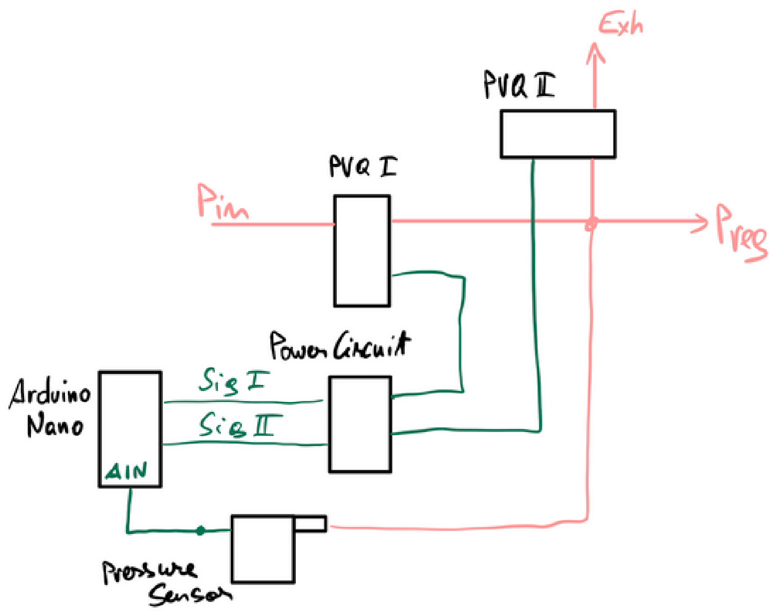


Figure 3.3 Second Concept.

4 Hardware

The following chapter describes the development and build process of the hardware for the implemented pressure driven pump. The main parts are an industrial pressure regulator, an external pressure sensor, a circuit board with Arduino to control it and a housing which holds all parts. Figure 4.1 depicts the pump from two perspectives and indicates its main components. Its overall size is 88 mm x 80 mm x 40 mm (LxBxH). The total costs of all hardware parts (excluding the 3D printed housing and necessary screws) are about 300 €. A full ordering list can be found in the appendix.

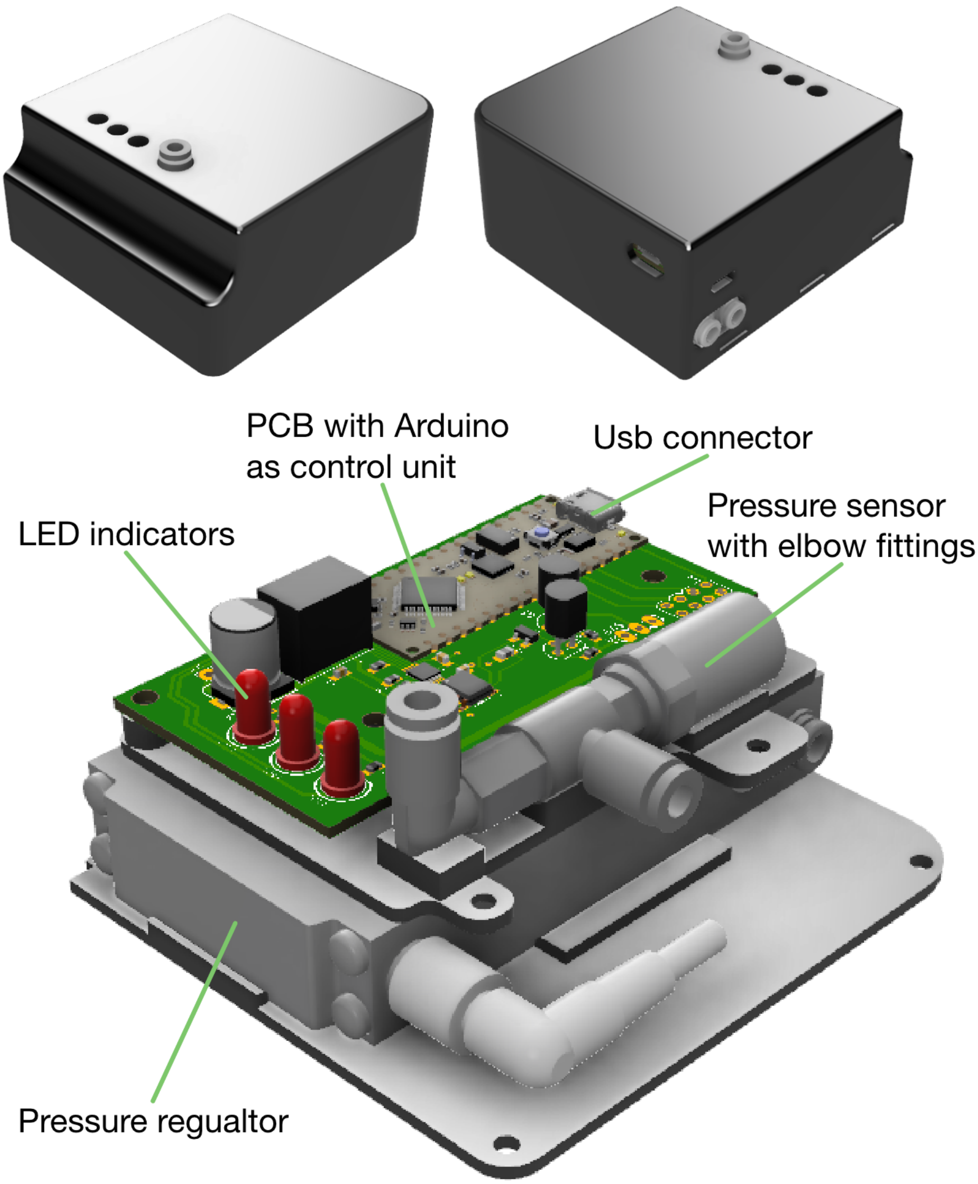


Figure 4.1 Overview of the main parts of the pressure pump hardware.

4.1 Pneumatics

4.1.1 Pressure regulator

The main functional part of the pump is the industrial pressure regulator ITV0010-0BL from SMC. It is very compact and lightweight, comprises high pressure stability and precision and can regulate pressure steplessly in proportion to an electrical signal. Furthermore the use of this industrial part reduces implementation complexity and eases scalability. Its main characteristics are shown in table 4.1. As shown in the table, a supply voltage of $24VDC \pm 10\%$ and a supply pressure of 2 Bar max. needs to be provided. The minimum supply pressure should equal the set pressure + 0,1 Bar. The output pressure can be set between 10 mBar and 1 Bar and is proportional to a 4 - 20 mA signal on the regulators input. This uncommon standard was chosen because it was listed as a standard part and fulfilled our requirements. An easier to control 0-5V version was provided by the vendor, however the the lead time of seven to nine weeks did not fit the targeted schedule.

Set Pressure Range	10 mBar to 1 Bar
Minimum Supply Pressure	Set pressure + 100 mBar
Maximum Supply Pressure	2 Bar
Power Voltage	24 V DC
Input Signal	4 – 20 mA
Output Signal (Pressure Sensor)	0 – 5 V
Fitting	One-Touch Metric $\varnothing 4mm$
Linearity	±1% F.S. or less
Hysteresis	0.5% F.S. or less
Repeatability	±0.5 % F.S. or less
Sensitivity	±0.2% F.S. or less

Table 4.1 ITV0010-0BL characteristics

4.1.2 Pressure Sensor

Though the pressure regulator features a pressure sensor, our concept envisaged an additional pressure sensor to improve the regulation performance. We decided to use the SMC PSE532-M5-L which has a significant higher accuracy than the sensor implemented in the pressure regulator (±1% F.S. compared to ±6% F.S. on the pressure regulator). Its electrical and pneumatic characteristics are shown in table 4.2. It requires a supply voltage of 12 - 24VDC and converts the applied pressure to a 1 - 5V signal. This can be measured with Arduino's internal analog to digital converter and therefore does not require additional parts.

Rated Pressure Range	0 – 1010 mBar
Power Supply Voltage	12 – 24 V DC
Analog Output Specification	1 V - 5 V
Linearity	±1% F.S. or less
Repeatability	±1% F.S. or less

Table 4.2 PSE532-M5-L characteristics

4.1.3 Pneumatic fittings and tubing

The selected pneumatic regulator comes with $\varnothing 4mm$, metric one touch fittings on the in- and outlet. To connect it with the pressure sensor, two female elbow fittings and $\varnothing 4mm$ pneumatic tubing was used.

4.2 Circuit board

As main electrical unit, a circuit board was designed and implemented. It comprises four functional blocks: power supply, a microcontroller unit (Arduino Nano Every), a voltage controlled current source to control the pressure regulator and three LED indicators. The two-layer board's size is 75 x 39 mm and it was manufactured by JLC PCB. To reduce size, SMD parts were primarily used. The schematic and a bill of material can be found in the appendix.
In the following, only the voltage controlled current source is described in detail, the power supply and LED indicator circuit are of a low complexity and don't require further description. A 24 VDC 18 W external power supply was used and converted to 5 V on the board for the control circuit and the Arduino. Figure 4.2 depicts the latest version of the board, however the implemented one has some differences which are described in section 4.3.

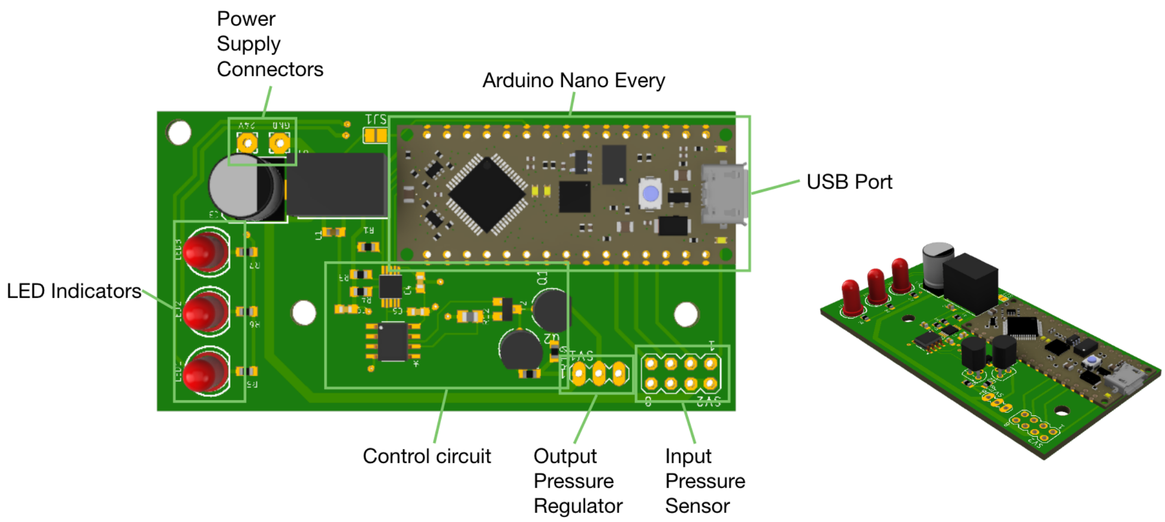


Figure 4.2 Overview of the main circuit board of the pump.

4.2.1 Voltage controlled current source

The pressure regulator is controlled by a 4 - 20 mA electrical signal. To provide this with an Arduino, an additional circuit is necessary. After some desk research we decided to implement a feedback operational amplifier circuit, which is often stated as precision voltage to current converter.
The input voltage is applied to the non-inverting port of the operational amplifier. The current to be controlled at the output causes a voltage drop across resistor R1, which is returned to the inverting port of the amplifier. The output of the amplifier is connected to the gate of transistor M1. Considering the operational amplifier characteristic of the high gain at the input, it will control the transistor in order to reach a voltage drop equal to the provided voltage on the input.[6]
The current on the output of the circuit is a linear conversion between the voltage on the input and the resistor R1.

4.2.2 Current mirror

The circuit described above comprises a current sink, however our application necessitates a current source. To fulfill this requirement a current mirror with two standard pnp transistors was implemented. The whole circuit was simulated in LTSpice and tested under controlled conditions.

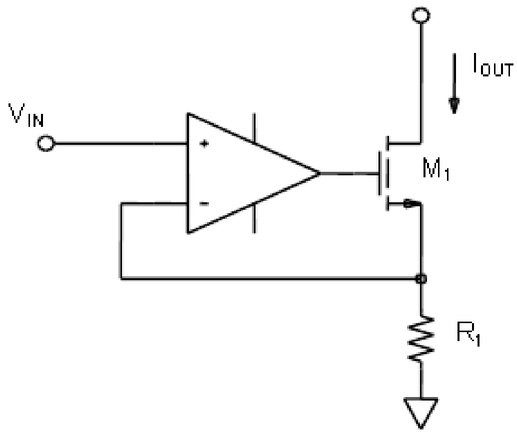


Figure 4.3 Basic feedback circuit for a precision voltage to current converter. [6]

4.2.3 Digital to analog converter (DAC)

In order to control the current, a controllable voltage source on the non-inverting input terminal of the operational amplifier is necessary. For this purpose a 10-bit DAC (MCP4728) was used. This part is connected to the Arduino via I2C bus and since it is frequently used, a code library was found online.

4.2.4 Circuit dimensioning and simulation

The functionality of the voltage controlled current source was checked with a circuit simulation in LT-Spice. Since enhancing the performance of the regulator with a superimposed PID controller was a primary goal, the circuit needed to be fast and linear. These characteristics were investigated by simulating a linear voltage rise from 0V to 4V in 1 ms and a fast drop from 4V to 0V. According to the datasheet of the pressure regulator, its input impedance is approximately 250Ω. Therefore the pressure regulator was reduced to a 250Ω resistor in the simulation.

The selection of the operational amplifier was carried out iteratively. Single supply and rail-to-rail capabilities and a decent slew rate were only important characteristics. In the first try, the LT6003 op amp was used in the circuit, however it did not meet our requirements due to its low slew rate. With the help of Analog's product finder, the LTC6242 was chosen in the next iteration.
The pnp transistors used for the current mirror did not need to meet specific requirements. Two 2N3906 were used out of convenience and their availability in the lab.
The dimensioning of resistor R1 in figure 4.4 results from the maximum input voltage of 4,096V, provided by the digital to analog converter. With $R1 = 200\Omega$, a maximum current of approximately 20,5 mA can be measured. Resistor R4 and R5 are equal and set the operating point of current mirror. Their value was determined by testing the circuit on the breadboard.

Figure 4.4 depicts the simulation circuit in LTSpice, figure 4.5 shows the simulated result. As described above, a linear voltage rise from 0V to 4V was applied to the non-inverting port of the amplifier (green line). As desired, the current through R1 (blue line) and R2 (red line) are identical and follow the applied voltage linearly. The result is a proof of concept and shows that the circuit is suitable for our application.

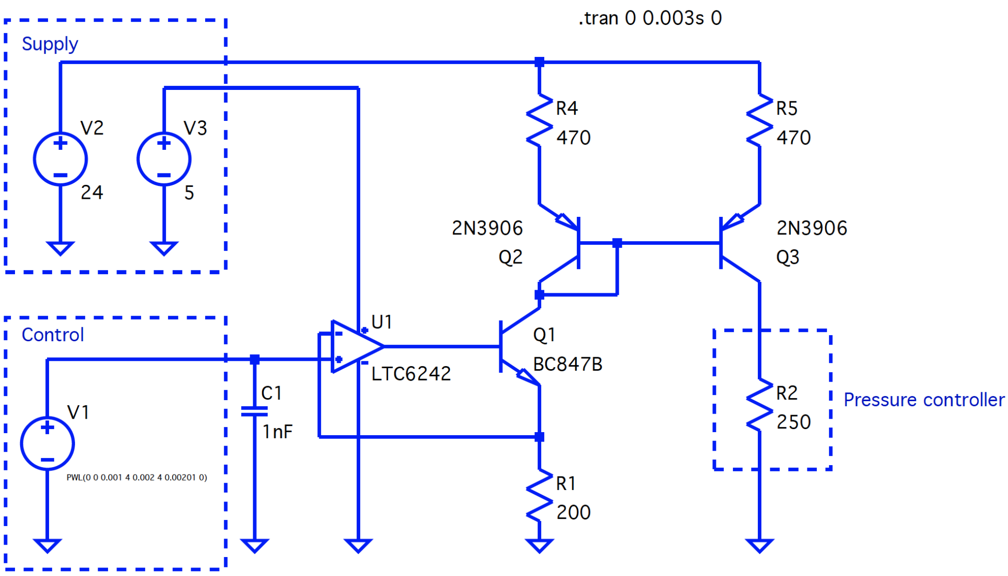


Figure 4.4 LTSpice simulation circuit

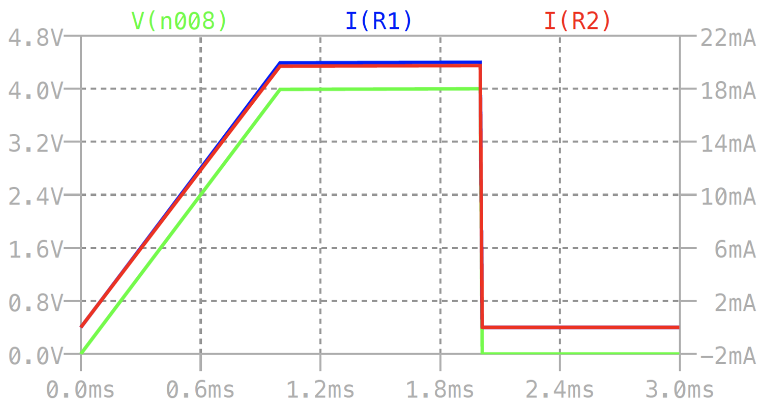


Figure 4.5 Simulation results of the voltage controlled current source.

4.3 Board design and commissioning

As described in the introduction of this chapter, a two-layer circuit board was designed in EAGLE. The selected vendor JLC PCB did not impose specific design rules, therefore EAGLE standards were used as design rules and for the design check. For ordering, *.gbr files were required which were generated with the standard EAGLE CAM processor.

4.3.1 Controlling the DAC

After delivery, the board was soldered and ready to be tested. The power supply and LED indicators worked as expected, however we had issues with the voltage-to-current circuit. Further investigation showed that the ground pin of the DAC was not connected. Since a I2C scan, conducted with the Arduino surprisingly returned the correct I2C address, finding the bug was not straightforward. However after fixing the issue with a tiny wire on the board, we were able to control the DAC as expected.

4.3.2 Current sink vs. source

Another design issue was the implementation of the wrong voltage-to-current converter. Due to lack of understanding how exactly the pressure regulator needs to be controlled, a current sink was implemented. However the regulator itself is a sink type and needs to be driven by a current source. The main difference is the direction of the current flow on the regulator's input: A sink type needs positive (ingoing) current and a source type needs negative (outgoing) current on the input port. The issue was solved with the pnp current mirror explained above and shown in the simulation. However this required external parts on top of the designed board.
Finally the board worked as expected and the simulated behavior was confirmed with sample measurements. Figure 4.6 shows the final board. The changes are highlighted in color.

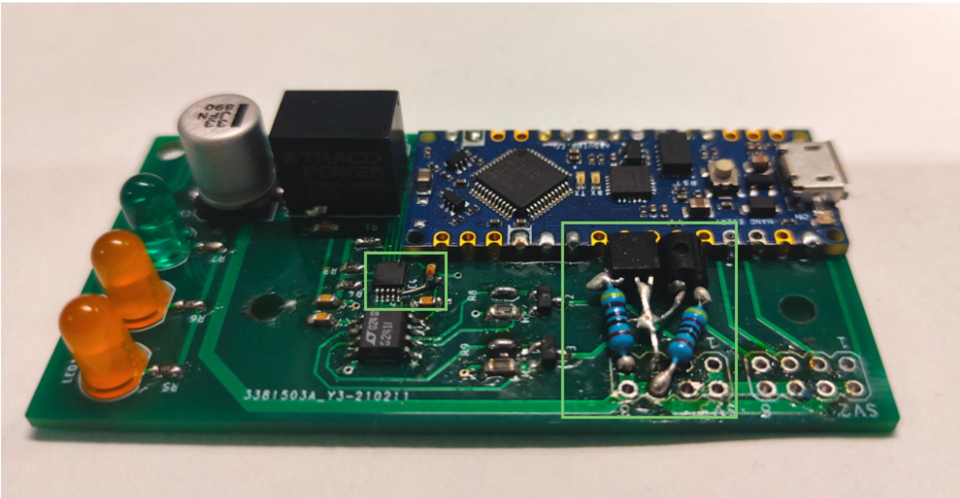


Figure 4.6 Soldered PCB with additional changes to fix design issues.

4.4 Enclosure and final implementation

The design of the enclosure was carried out in Autodesk Fusion. It does not include any aesthetic design features, instead the focus was solely on the basic functionality. It consists of three parts and was 3D printed in the lab. Seven M3x16 countersunk screws are necessary to hold all parts in place. The assembly went smooth, except that the through holes for the screws had to be extended. In case of a rebuild, extending the through holes in the CAD files is recommended.

5 Software

In this chapter, we will document the conception and implementation phases of the software shipped with our pump. We will first discuss the concept of our Finite State Machine (FSM) and then document the sketches uploaded to our Arduino microcontroller. At a later point, we will discuss and document the developed graphical user interface and propose a guide on the best practices to use it. The necessary software for this lab included the Arduino Software IDE and Laboratory Virtual Instrument Engineering Workbench (LabVIEW). Most recent versions of the programming environments can be found at the corresponding websites. Supporting documentation and programming instructions and tutorials can also be found at the corresponding websites.

5.1 State Machine

As a first step, we conceptualized a simple FSM flowchart that our Arduino program will be based on. Each of these steps were implemented in separate functions on our microcontroller. The flow chart for the logic of the Arduino can be seen in the figure below.

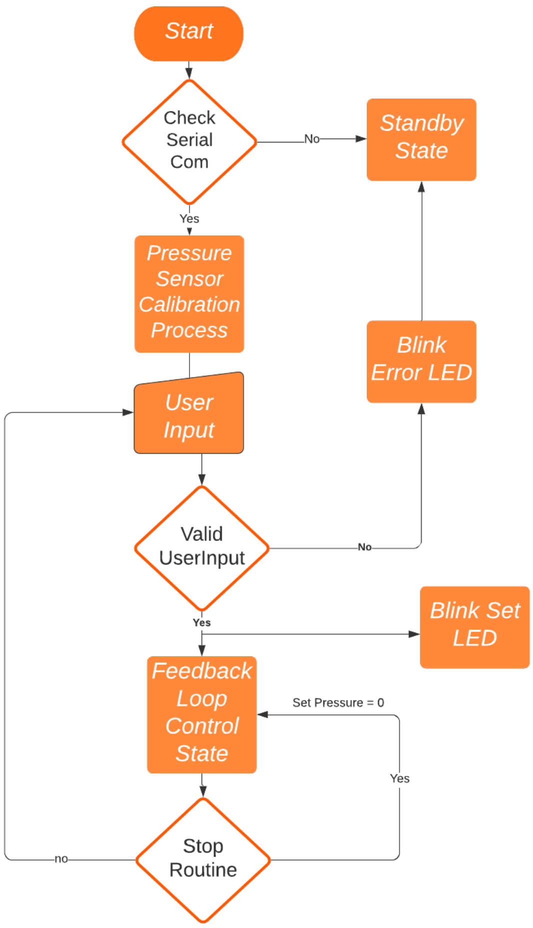


Figure 5.1 Finite State Machine Flowchart.

The main states of the logic that we implemented are:

- **Calibration State:** This State does not need user input to be activated. As soon as the pump starts, a null signal would be sent to the controller. Following this, 64 values of the pressure sensor will be stored in an array and averaged. The resulting value is our calibration value for the pressure sensor and fundamental for a correct mapping between the control signal and the ouput pressure. This value could be changed in the sketch of the microcontroller.

- **Feedback Control State:** To enter this state, the user needs to enter a valid pressure value (within our pressure range 0mbar and 1000mbar). Once the user input is validated, its value will be mapped by the microcontroller and sent to the DAC, which controls current source and the pressure regulator subsequently.

This FSM logic was implemented using the Arduino Nano Every microcontroller. All the transitions were delayed by 10ms to guarantee analytical stability. The code of this FSM can be found in the appendix.

5.2 Arduino

In our use-case the Arduino was programmed to communicate with the user interface via serial communication, sending signals to the pressure controller and receiving signals from the pressure sensor. The measured voltage from the sensor is mapped to a pressure value (according to the sensor's documentation) and will be later displayed to the user.

The Arduino first initializes the variables and the DAC MCP object and prepares the board for serial communication at a predefined baud rate.

After setting up all the important variables, the Arduino enters a loop which runs continuously until the device is turned off or interrupted by a the stop button. If there is no user input, the Arduino will read the incoming pressure values, turn off the pressure controller and calibrate the sensor by averaging the last 64 sensor readings. The system will then enter the standby state waiting for a user input over the serial channel.

Once a user input was applied, the system checks whether this variable is within our pressure range (0 mbar and 1000 mbar). If the value is indeed between the maximum and minimum allowable values, the Arduino will map this value so that the controllable voltage-to-current converter outputs a predefined current value that will control our pressure regulator. This value will be retained, unless the user inputs another set pressure or stops the system. This value can be set by the user via the graphical interface.

The sketch for the Arduino can be found in the appendix.

5.3 LabView GUI

In order to facilitate the use of our pressure driven pump, we conceptualized and developed a user-friendly graphical interface using LabView. This framework provides standard components such as buttons, checkboxes, trees, and dropdown lists, that made the usage of our program considerably more intuitive. Using this GUI, the user is able to monitor the connection status of the microcontroller, visualize the sensor value and control the output pressure in real time.

The back-end of this user interface is mainly structured around three main components which are:

- **Arduino Detection:** This first component of the interface detects whether the Arduino Nano Every microcontroller is connected and recognized by the computer. For this component to work correctly it is essential to install the latest drivers for the Arduino Nano Every board.

- **VISA Serial Resource:** This component is where the baud rate, package size, stop bits, and parity bits are declared. For further work, the baud rate is variable and can be tuned directly from the front-end. However, all serial connection parameters are hardcoded within the Arduino sketch and should not be changed

- **Serial Read and Write:** This component is responsible for the serial communication between the GUI and the microcontroller. The serial read will check every 10 ms for new user input and the serial write will then transmit it to the pump.

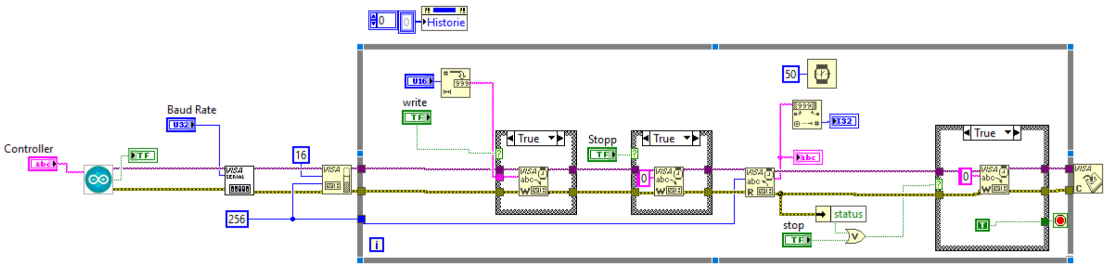


Figure 5.2 LabVIEW Back-end Interface

For this back-end, we developed a simple user-friendly frontend to be able to control the pump. A tutorial GIF can be found in the appendix in order to demonstrate the functionality of the user interface. While developing the graphical user interface (GUI), we tried to keep the interface as simple and intuitive as possible, in order not to overwhelm the user.

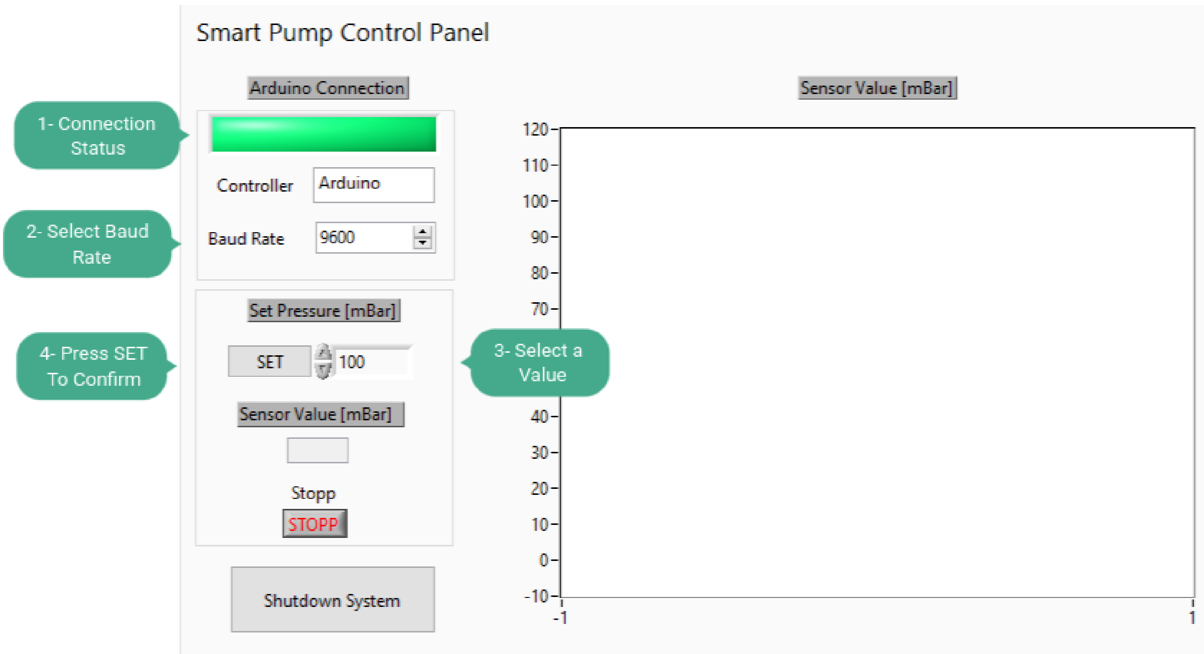


Figure 5.3 LabVIEW Front-end Interface.

6 Results and Discussion

In this chapter, we will describe the experiments carried out to test our system and then we will proceed to present the analytical results at different pressure set-points separately.

6.1 Experiments Description

To be able to emulate the daily usage of the pump in the research field different scenarios were simulated. These scenarios differs mainly regarding three factors: **set pressure**, **time** and **hydraulic resistance**. During our experiments, we were able to variate the pressure set points, going from the minimal adjustable pressure to the maximal adjustable pressure measuring the microfluidic flow during different periods of time. In our test setup, we used a 2,7 m long microfluidic pipe to achieve some hydraulic resistance and tap water as fluid. The resulting fluid flow was measured with a fluigent flow sensor and captured with LabVIEW. Since this sensor has a maximum flow rate of $120\mu l/min$, with our setup we could only apply pressures up to 650 mBar. We choose to stop the characterisation at 500 mBar, since then overshoots during value changes were captured adequately. For a characterization of the full operating range of the pump (up to 1 Bar), a higher hydraulic resistance is necessary. This can be achieved by changing the system (eg. longer pipe) or using a more viscous fluid.
Our experiments resulted in three datasets:

- **Short Term Measurements (STM):** In order to study and analyse the short term behavior of the system, we performed a recording of the microfluidic flow for relatively short periods of *20 sec* at pressure set points ranging from *2 mbar* to *500 mbar* with a sampling rate set at *50 ms*. These experiments will allow us to study if the system is able to stabilize itself and hold this stability during short time spans. Furthermore, we investigated the settling time after a value change.

- **Long Term Measurements (LTM):** In order to study and analyse the long term behavior of the system, we performed a recording of the microfluidic flow for relatively long periods of *1 min* at pressure set points ranging from *2 mbar* to *500 mbar* with a sampling rate set at *50ms*. These experiments will allow us to study if the system is able to stabilize itself and hold this stability for longer time spans.

- **Stepping Measurements (SM):** In order to study and analyse the ability of the system to switch between different pressure set points, we performed a stepped recording of the microfluidic flow. For this experiment, we changed the pressure set points, going from *2 mBar* until reaching *500 mBar* every *10 seconds* with a sampling rate set of *50 ms*. This experiment will allow us to study if the system is able to switch between states fast enough to assure a smooth workflow.

As final experiment, we analysed the behaviour of the system in a real world lab scenario. The goal was to generate droplets by using our our pressure driven pump and the syring pump of a second team conducting the internship. These experiments are documented in a video form and can be found in the appendix.

In the next chapter, these acquired datasets will be analysed using the **mean value**, **standard deviation**, **settling time** and **coefficient of variation**. The documented evaluation was reduced to a few data sets, since the final experiment clearly showed the higher importance of low pressure values.

6.2 Evaluation of the Measurements

After taking the measurements with pressure values of 10 mBar, 50 mBar and 100 mBar with a time step of 50 ms it was possible to study the linearity of the flow rate/pressure relationship over the operating range of the controller while specifying the minimal and maximal pressure. The analysis of stability is also an important step for the evaluation of the measurements, by calculating the coefficient of variation for the measuring points :

CV = the standard deviation after settling (rms noise)/mean value after settling

The rise and fall times for each measuring point are also to be considered while looking through the available results after executing the experiments with different pressure values. The steppring ability of the system has also been tested with a sampling rate of 50 ms has while taking the measurements of several pressure setpoints starting from 2 mBar, as the figure 6.1 shows.

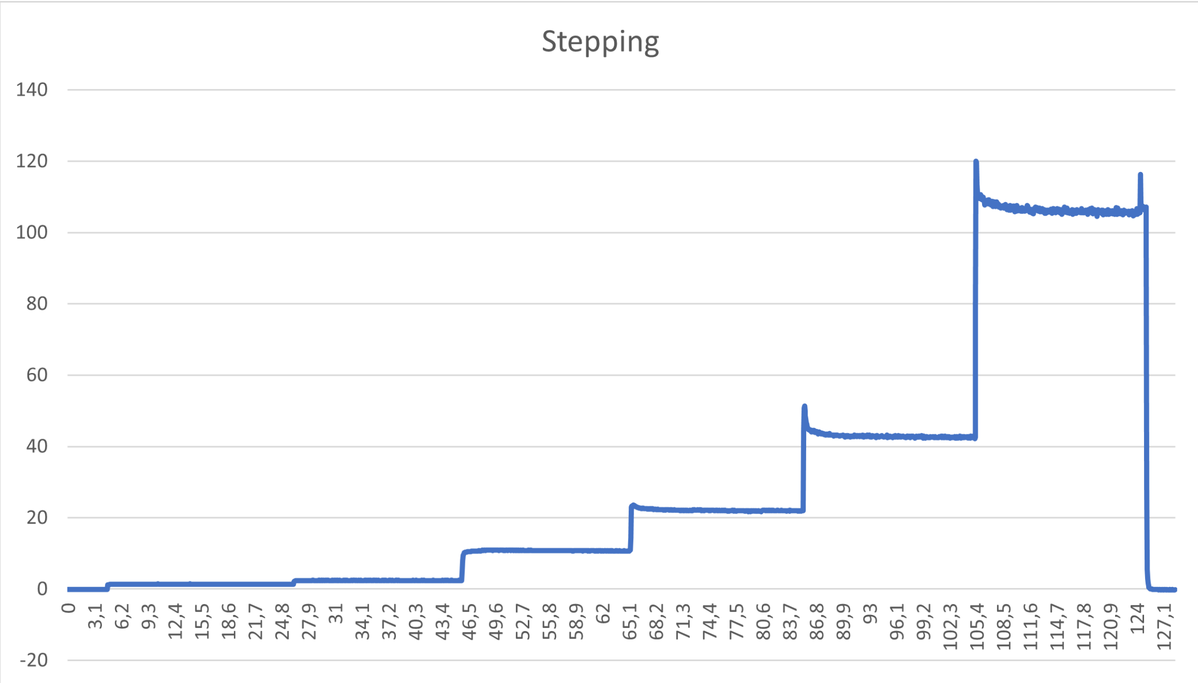


Figure 6.1 Stepping Measurements.

6.2.1 10 mBar Measurements

As the figure 6.2 shows, the measurements while having 10 mBar as the pressure value for the short term (ST) have shown that the flow rate has stabilized after a period of 1,4 s while it took approximately 2 s to stabilize by the long term (LT) experiment (settling time: until the measurements become stable and constant). As the average value of the ST flow rate is 3.2138 µl / min, it has decreased during the LT measurements to a value of 2.8983 µl / min while the standard deviation of the ST has increased from a value of 0,0196 during the ST experiment to 0.0445 during the LT measurements, which is still far below the value of 1, taking also in consideration the calculated coefficient of variation in the ST experiment 0,0061 and 0,0153 in the LT experiment, it is clear that the measured values have been indeed stable and show very good results. The linearity of the flow rate and pressure relationship over the operating range of the controller is also a significant part of the evaluation of the measurements. As the given pressure in this experiment is 10 mBar and the average value of the relatively constant flow rate (short term) is 3.2138 µl / min, the division of these two values, assuming that the linearity is described by the function y = a . x, gives us the linearity coefficient 3,111.

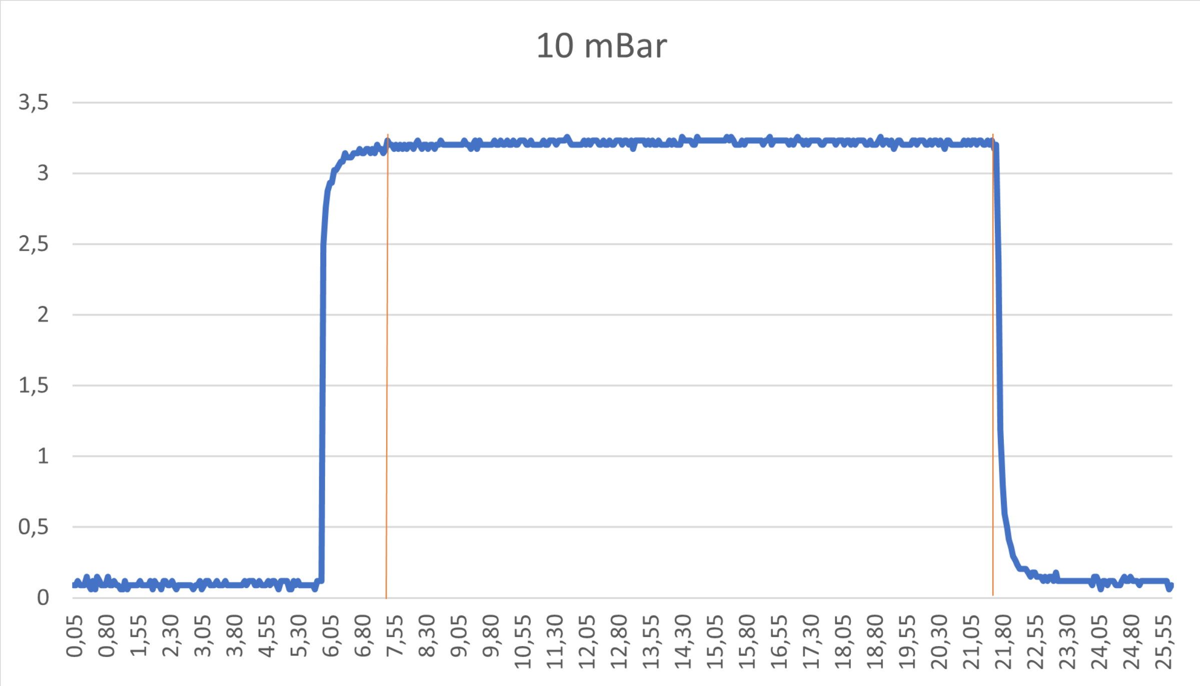


Figure 6.2 10mBar Short Term Flow rate.

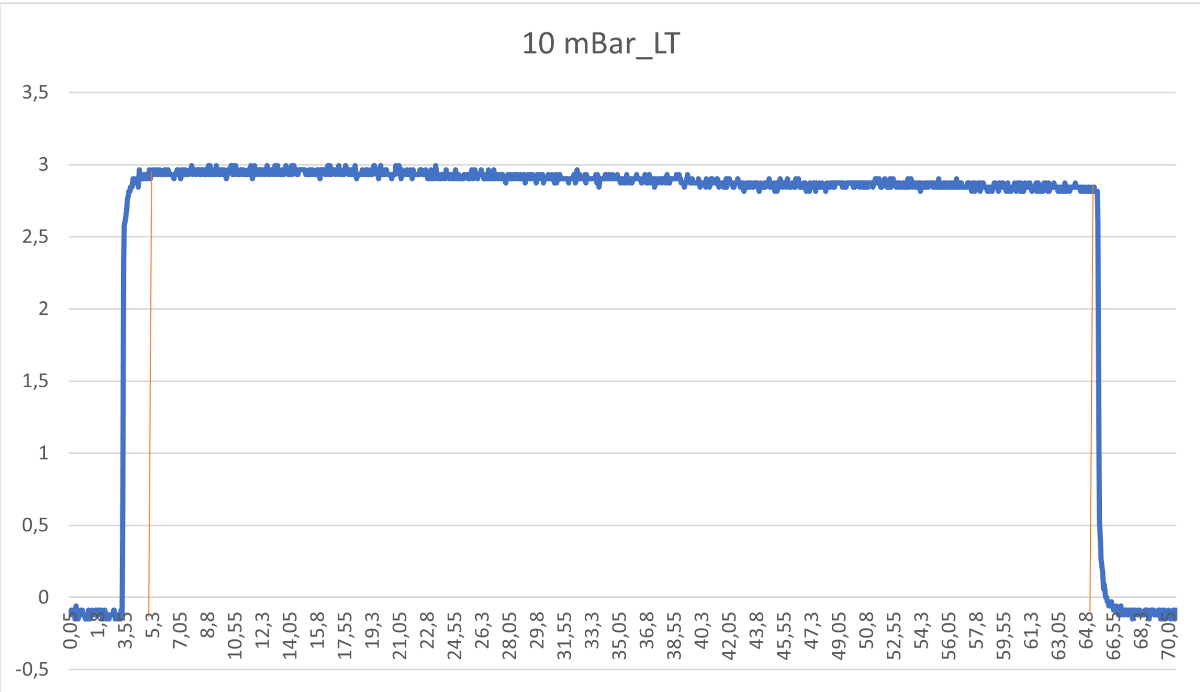


Figure 6.3 10mBar Long Term Flow rate.

6.2.2 50 mBar Measurements

As the measurements were followed by changing the pressure to 50 mBar it is obvious in the figure 6.4 that by this value the flow rate took longer than the previous experiment to stabilize since the settling time was 2,1 s. Even though the figure does not show the same stable behaviour with the pressure of 10 mBar, the standard deviation has a value of 0,1272, which also below the 1 value, indicating that the

Average	**3,2138 µl / min**
StdDev	**0,0196**
CV	**0,0061**
settling T	**1,4 s**

Table 6.1 10 mBar Short Term calculations

Average	**2,8983 µl / min**
StdDev	**0,0445**
CV	**0,0153**
settling T	**2 s**

Table 6.2 10 mBar Long Term calculations

measurements still can be considered stable. The calculated average value of the flow rate is 11.0743 µl / min giving a final result of the coefficient of variation in this experiment of 0,0115. As the pressure value has increased, it was noticeable that the results were less stable than before, but the calculations still show plausible results and stable behaviour overall. For the linearity of the flow rate and pressure relationship by the evaluation of the 50 mBar measurements, we have a constant flow rate (short term) of 11,0743 µl / min, the division of these two values, using the same function for the linearity y = a . x, gives us the linearity coefficient 4,515, which is above the last measured linearity coefficient using the pressure value 10 mBar. A further investigation of the linearity with increasing pressure values will then follow in the next experiment.

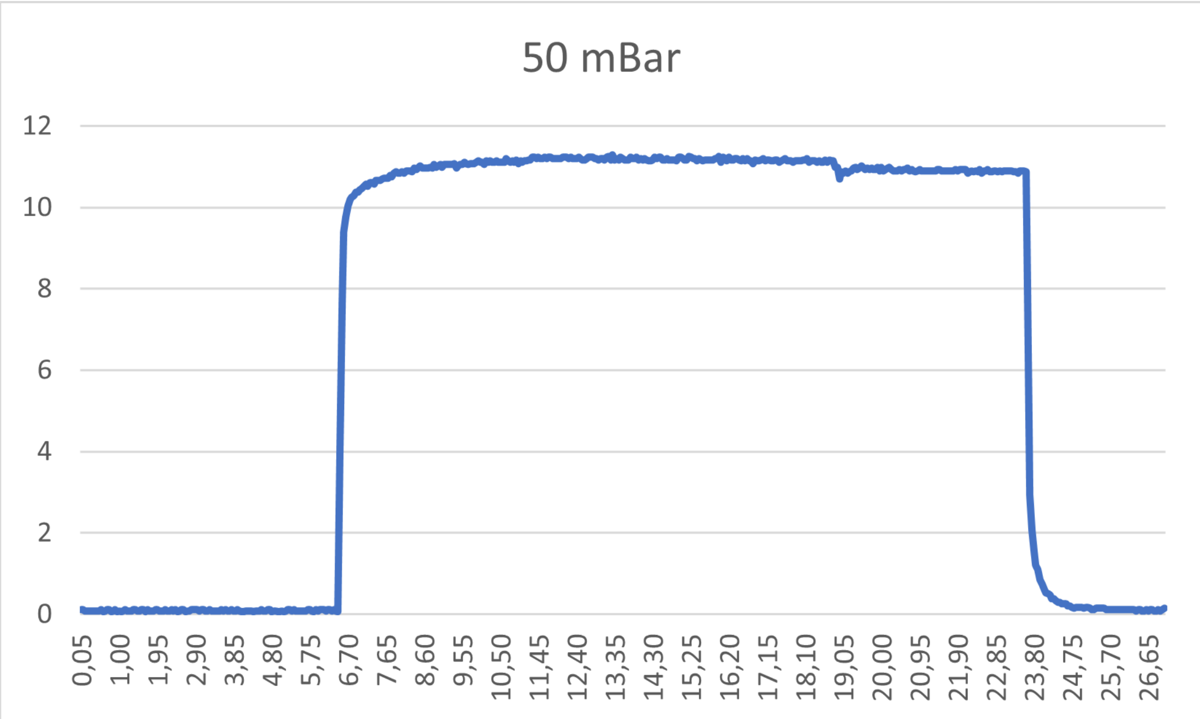


Figure 6.4 50mBar Short Term Flow rate.

Average	11,0743 µl / min
StdDev	0,1272
CV	0,0115
settling T	2,1 s

Table 6.3 50 mBar Short Term calculations

6.2.3 100 mBar Measurements

The last experiment has consisted of taking the measurements with a pressure value of 100 mBar. As the plot of the results in the figure 6.5 illustrates the stable behaviour of the flow rate with a settling time of 2,65 s it also indicates the linearity of the flow rate and pressure relationship of the controller while setting the maximal pressure. The standard deviation in this experiment has has increased with a value of 0,2688 comparing to the previous value with 10mBar and 50mBar measurement values. As the pressure increases, the flow rate shows a less stable behaviour, but in this case, it is still below 1. The calculated average value of the flow rate is 22,0608 µl / min, which is also the double of last average value with the 50mBar experiment. The coefficient of variation of the following measurements is 0,0122, which is still in the same range of the measured value of the previous experiment, showing by that an appreciable stability. Similarly to the measurements of the 50 mBar pressure value, the calculation of the linearity coefficient in this case has given a value of 4,533, proving by that the steadiness of the linearity coefficient improves by increasing the pressure value.

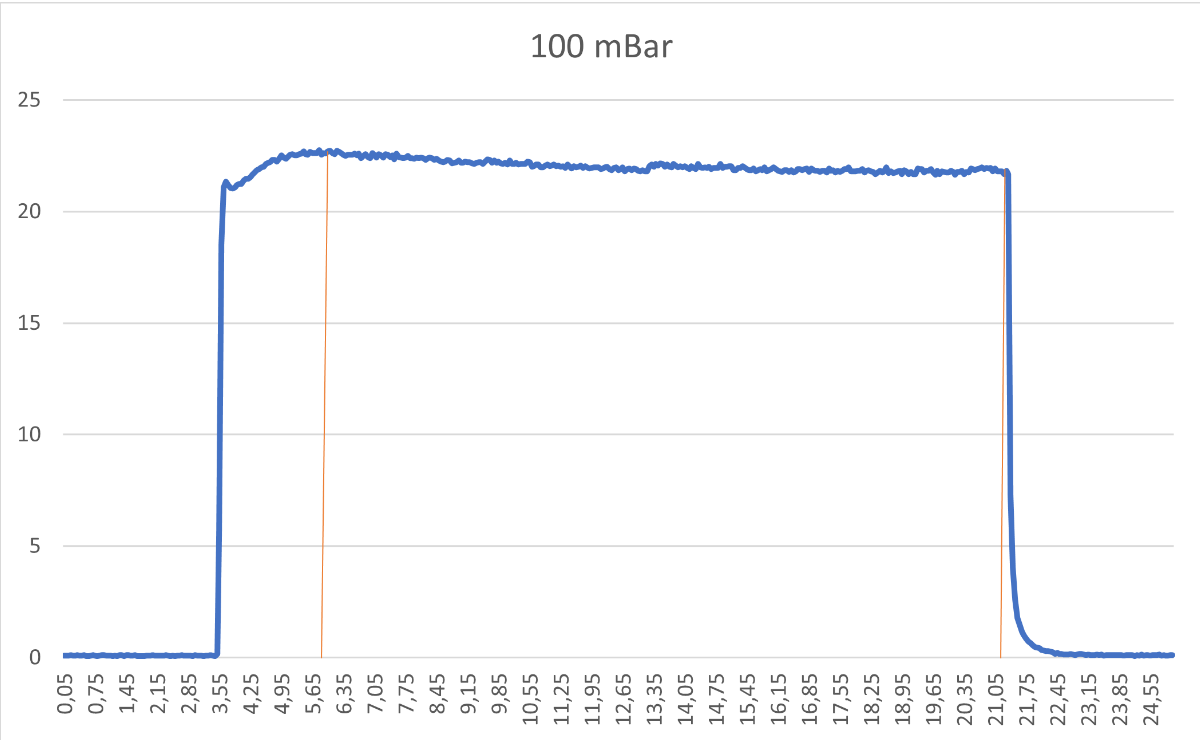


Figure 6.5 100mBar Short Term Flow rate.

Average	**22,0608 µl / min**
StdDev	**0,2688**
CV	**0,0122**
settling T	**2,65 s**

Table 6.4 100 mBar Short Term calculations

7 Conclusions & Outlook

In this chapter, the different parts of this project lab will be summarized and put into context with the conceptual framework and motivation of the work found in the introduction. The performance and costs of of our pressure driven pump will be compared to the industrial model. We will also provide an outlook on potential further improvements of our system for future lab works.

7.1 Conclusions

Controlling flow rate has always been a critical process for most microfluidic applications and research experiments. The approach of majorly downscaling these experiments might in fact increase their sensitivity and resolution but also makes them more susceptible to external factors. To circumvent this issue, we had the task to conceptualize and develop a pressure driven pump to deliver flow and provide the best control of the fluids at a microfluidic scale.

With the achieved high-precision control of the liquid flow and the subseconds response time, our pump system can largely compete with industrial alternatives at a fraction of the cost. As shown in the droplet experiments, our device system can be a viable alternative to costly industrial pressure driven pumps for most of the research application at TranslaTUM.

Due to its compact size and optimal packaging, our device is fully portable and can be easily carried around. If a user has a computer, compatible tubing and a pressurized air source to disposal, our setup can be ready to use within minutes.

Another main advantage of our pressure driven pump is that one can pressurize several reservoirs with only one pressure channel and microcontroller. This is a significant cost reduction to the setup if you want to inject different solutions sequentially.

7.2 Outlook

The developed pressure driven pump shows the high potential to utilize industrial pressure controllers as low-cost alternative to expensive lab-equipment. We could show experimentally, that the controller incorporates sufficient performance for experiments with microfluidic systems at a fraction of the costs of state-of-the-art lab equipment. Due to the inherent design of the pump with only a few mechanical parts and the use of industrial components it also comes with high reliability.

We would like to highlight the scalability of the system. With the implemented digital to analog converter up to four voltage-to-current converters can be established. Therefore, only few changes on the circuit board are necessary to control four regulators and obtain four individual pressure pumps. However, if another non-standardized controller will be used (ITV0010-2CL) which is not more expensive but comes with a higher lead time, the regulator can be controlled by voltage and the voltage-to-current converter is no longer required.

In our experiments we saw that the regulation performance of the pressure regulator is excellent. This makes the additional pressure sensor unnecessary, since a superimposed PID controller on the pressure does not lead to any improvements in control performance. Considering a system without external sensor the total costs will lower to approximately 250 € for a single pump. Any additional pressure regulator adds 200 €, whereby a quadruple pump will cost approximately 850 €.

Another modification in the future could be the implementation of a flow sensor. In many experiments, the fluid flow is an important set parameter but is only controlled indirectly by the pressure. By adding a flow sensor to the system a superimposed PID controller acting on the flow would enhance the system's usability.

Bibliography

[1] [Online]. Available: https://www.alamy.de/stockfoto-labor-auf-einem-lac-chiptechnologie-mit-einer-pipette-101777236.html

[2] P. Frank, S. Haefner, M. Elstner, and A. Richter, "Fully-programmable, low-cost,"do-it-yourself" pressure source for general purpose use in the microfluidic laboratory," *Inventions*, vol. 1, no. 2, p. 13, 2016.

[3] V. Nguyen, "Pressure driven pump for portable microfluidic devices," Ph.D. dissertation, 2014.

[4] [Online]. Available: https://www.fluigent.com/de/product/microfluidische-komponenten/mfcs-ez-microfluidic-flow-control-system

[5] H. Bruus, *Theoretical microfluidics*. Oxford university press Oxford, 2008, vol. 18.

[6] D. Mercer, "Op amp applications - advanced topics." [Online]. Available: https://wiki.analog.com/university/courses/electronics/text/chapter-4

I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.

Munich, 15.04.2021