Technische Universität München
Department of Informatics

Prototyping a Portable 3D-Printed Microfluidic System Controlled Using a Single-Chip Computer

Master's Thesis in Informatics

Yushen Zhang

Technische Universität München
Department of Informatics
Department of Electrical Engineering and Information Technology
Institute of Electronic Design Automation

Prototyping eines portablen 3D-gedruckten Mikrofluidsystems gesteuert mit einem Ein-Chip-Computer

Prototyping a Portable 3D-Printed Microfluidic System Controlled Using a Single-Chip Computer

Master's Thesis in Informatics

Yushen Zhang

Advisor:	Dr.-Ing. Tsun-Ming Tseng
Supervisor:	Prof. Dr.-Ing. Ulf Schlichtmann
Submission Date:	15.12.2019

Confirmation

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Date: 15.12.2019 Signed: Yushen Zhang

Abstract

In the past few decades, microfluidic systems have evolved rapidly and are nowadays widely used in different areas. With time, many types of microfluidic systems have evolved. In contrast to a conventional microfluidic system, I propose a portable microfluidic system, which has the advantage of small-sized, usable on-the-go, and easy to access. This work introduces the designed prototype of the proposed portable microfluidic system, including the control hardware and software, and the accessible production of a microfluidic chip, as well as the usability of the prototype. The state-of-the-art microfluidics system, the used hardware controller and microfluidic chip production methods will also be elaborated. Prospective optimization directions are discussed at the end.

Abstrakt

In den letzten Jahrzehnten haben sich die Mikrofluidsysteme rasant weiterentwickelt und sind heutzutage in verschiedenen Bereichen weit verbreitet. Mit der Zeit wurden viele Arten von Mikrofluidsystemen entwickelt. Im Gegensatz zu einem herkömmlichen Mikrofluidsystem schlage ich ein tragbares Mikrofluidsystem vor, das den Vorteil hat, dass es klein, mobil und leicht zugänglich ist. Diese Arbeit zielt darauf ab, den entworfenen Prototyp meines vorgeschlagenen tragbaren Mikrofluidsystems, einschließlich der Steuerhardware und -software, und die zugängliche Herstellung eines Mikrofluidik-Chips sowie die Verwendbarkeit des Prototyps vorzustellen. Das hochmoderne Mikrofluidsystem mit dem verwendeten Hardware-Controller und den Produktionsmethoden für Mikrofluidik-Chips werden ebenfalls ausgearbeitet. Um die Potenziale dieses Prototyps aufzuzeigen, wurden am Ende mögliche Optimierungen und weitere Möglichkeiten in Aussicht gestellt.

Acknowledgments

I would like to give everyone who supported me during this work my special thanks. Especially, I would like to give my advisor Dr.-Ing. Tsun-Ming Tseng and to my supervisor Prof. Dr.-Ing. Ulf Schlichtmann warm thanks for providing me this exciting topic as my master's thesis. With their support and help, I was able to finish this thesis without hassle. I am deeply convinced that all the practices I have had during my work will help me in the future, whether it is in my academic life or in my career. Finally, I would like to thank Mr. Liudongnan Yang very much for his work and contribution to the program implementation part of this thesis. He precisely followed the requirement of my software design and implemented the program very well.

Contents

List of Figures

List of Tables

1 Introduction

During the past few decades, microfluidics became more and more important in the area of biology [1, 2] and chemistry [3]. While this technology is the science of manipulating and controlling fluids in a range of microliters, it largely reduces the size of laboratory experiments. It emerges as a powerful tool in many ways based on the advancements in the microfluidic system.

Microfluidics allows experiments to be done simply on a small-sized chip. Currently, to have a lab on a chip is not very convenient as we might think. On the one hand, conventionally, the corresponding microfluidic chip for the laboratory experiment needs to be produced in a special environment with inconvenient procedures. On the other hand, to perform the experiment, large equipment such as stationary pumps are required. Thus, scientists still need to perform their experiments in the lab and cannot gain the benefits of the lightweight and small size of a microfluidic chip. In fact, these benefits open many possibilities, especially the potential of portability.

In this thesis, I construct a prototype of a portable microfluidic system. I call it "portable" in the sense of lightweight, transportable, usable on-the-go and easy to access. It has all the necessary components, but in a small and portable design, to perform an experiment.

The prototype is divided into two separate designs. The first part is the controller part. The controller is based on a conventional, accessible single-board-computer, connected with miniature pumps. Upon that, I design and implement a corresponding control software that can be used via a touchscreen for easy user interaction.

While three-dimensional (3D) printing has emerged as a potentially revolutionary technology for the fabrication of microfluidic devices [4], the second part is designing

exemplary microfluidic chips for my prototype, which can be produced in an easy, cost-friendly and accessible way, namely using the 3D printing technology.

Eventually, I use these chips connected to my prototype system to demonstrate the feasibility and usability of the prototyped microfluidic system.

This thesis starts with a short background review in the second chapter, elaborating on the state-of-the-art microfluidic system. I will especially talk about how current microfluidic systems are constructed, their size and the different pumping technologies. Furthermore, I will discuss the fabrication methods of a state-of-the-art microfluidic chip.

In the third chapter, I discuss the concept of the prototype. I concentrate on the design of the hardware and software of the controller, and further, in the fourth chapter, I show how 3D printable chips are designed. Moreover, I design demonstrative experiments with the corresponding microfluidic chip for the prototype system.

The fifth chapter shows the results. The final result of my designed prototype is explained. I shortly elaborate both the hardware and software of my prototype, then I continue to the 3D-printed chip and the demonstrative experiments performed with these chips and my prototype system.

Finally, I give a summary on the prototype I have designed. Specifically, the focus will be on further improvement possibilities and other integrations opportunities to enhance the system.

2 Background

With the advancement of computer science, many traditional manual works have become digitalized. So is the control of microfluidic and the fabrication of microfluidic chip. In this chapter, I discuss the state-of-the-art microfluidic chip fabrication and controlling system.

The core of a microfluidic system is undoubtedly the microfluidic chip. To produce a chip, there are many ways to fabricate the necessary channels and feature dimensions. In the following section, I elaborate on the current fabrication techniques and methods.

In fact, there are several different ways to fabricate a microfluidic chip or device. Prototyping techniques, including hot embossing [5, 6], injection molding [7, 8], and soft lithography [9], and direct fabrication techniques such as laser photoablation or laser micromachining [10], photolithography [11] and x-ray lithography [12] are the most conventional methods used for fabrication of microfluidic chip currently. Figure 1 and Figure 2 depict the fabrication process by using prototyping techniques and direct fabrication techniques.

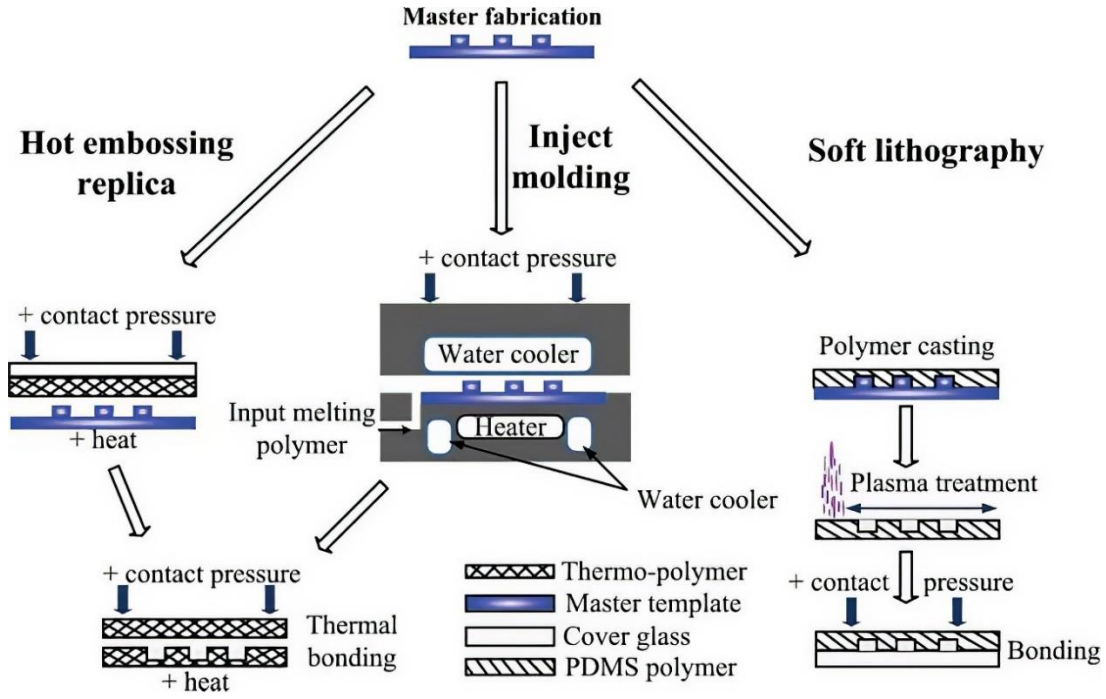


Figure 1: Illustrations [13] of the typical fabrication procedures by the prototyping techniques (hot embossing, inject molding, and soft lithography).

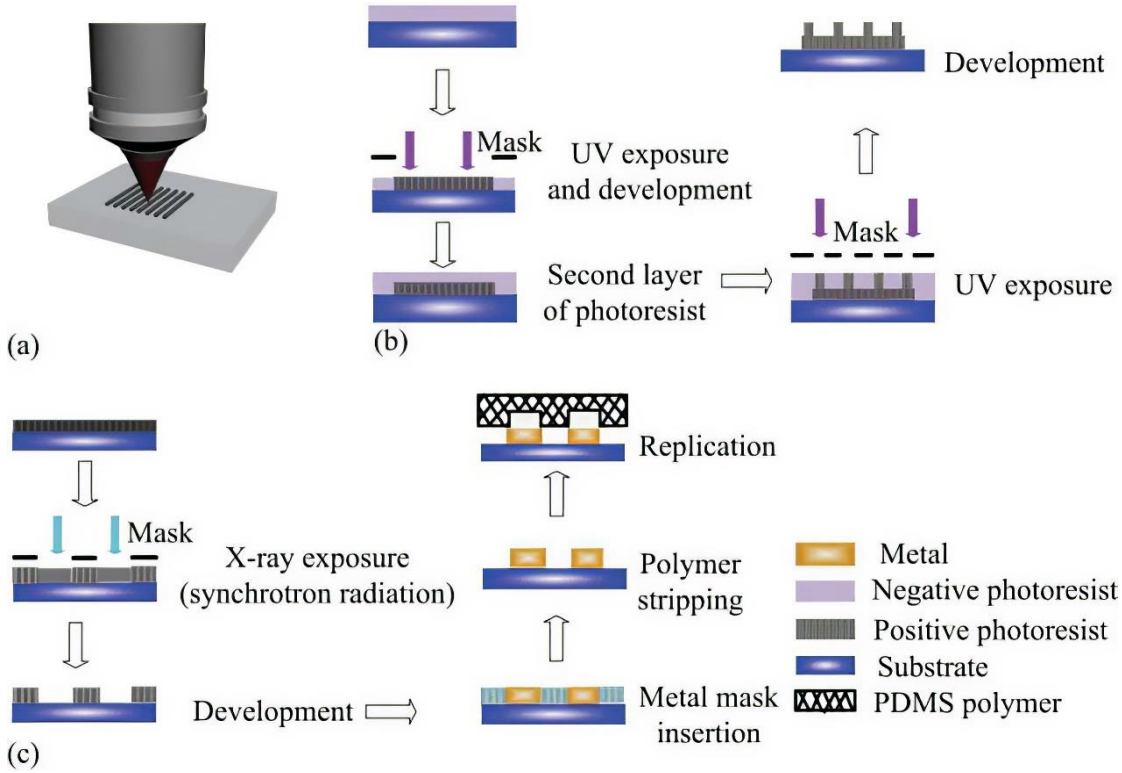


Figure 2: Illustrations [13] of the typical fabrication procedures by the direct fabrication techniques: (a) laser micromachining, (b) photolithography/optical lithography, and (c) x-ray lithography. The "metal mask insertion" step in x-ray lithography is usually realized by electroplating.

Wu et. al. [13] have summarized the advantages and disadvantages of the above-mentioned techniques in Table 1.

Methods	Advantages	Disadvantages	Ref.
Hot embossing	Cost-effective, precise, and rapid replication of microstructures, mass production	Restricted to thermoplastics, difficult to fabricate complex 3D structures	[6]
Injection molding	Easy to fabricate complex geometry, fine features, and 3D geometries, low cycle time, mass production, highly automated	Restricted to thermoplastics, high-cost mold, difficult to form large undercut geometries	[8]
Soft lithography	Cost-effective, able to fabricate 3D geometries, high resolution (down to a few nm)	Pattern deformation, vulnerable to defect	[9, 14, 15, 16, 17]
Laser photoablation	Rapid, large-format production	Multiple treatment sessions, limited materials	[10]
Conventional photo-/optical-lithography	High wafer throughputs, ideal for microscale features	Usually requires a flat surface to start with, chemical post-treatment needed	[11]
X-ray lithography	High resolution to fabricate nano-patterns, absorption without spurious scattering, able to produce straight smooth walls	Difficulties in the master fabrication process, time-consuming, high-cost	[12]

Table 1: Summarization [13] of conventional microfluidic fabrication techniques.

As I can see from the illustrations and the table, conventional microfluidic chip fabrication techniques involve many specialized resources, materials, and skills beyond the capability of many biology and chemistry scientists [18, 19]. Especially, dedicated skills and expensive cleanroom infrastructures are required [20, 21].

In recent years, additive manufacturing, or three-dimensional (3D) printing has attracted significant attention in manufacturing. In the field of microfluidics, 3D printing offers the capability to directly print complex microfluidic devices with low-cost consumer-grade desktop printers, changing the way in which microfluidic chips and devices are conceived, designed, and manufactured [4].

The most commonly used 3D printing methods are fused deposition modeling (FDM) printing, PolyJet printing, and DLP-SLA printing. Figure 3 illustrates the three different 3D printing methods [4]: (a) FDM method, in which molten plastic is extruded through a heated nozzle, according to the G-code. The features are formed by moving the nozzle in the XY-plane until the current layer was complete when the build platform was then dropped by a set level before moving to the next layer. (b) PolyJet method, where two sets of four-micronozzle arrays (build and support material, respectively) are spraying microdroplets of polymer to form the device. Following each pass, UV lamps polymerize the material before the layer is leveled by a roller and scraper. (c) DLP-SLA method, in which UV light was projected onto a build platform immersed in liquid photopolymer. Following each exposure, the build platform was raised then lowered back into the resin bath for the next layer

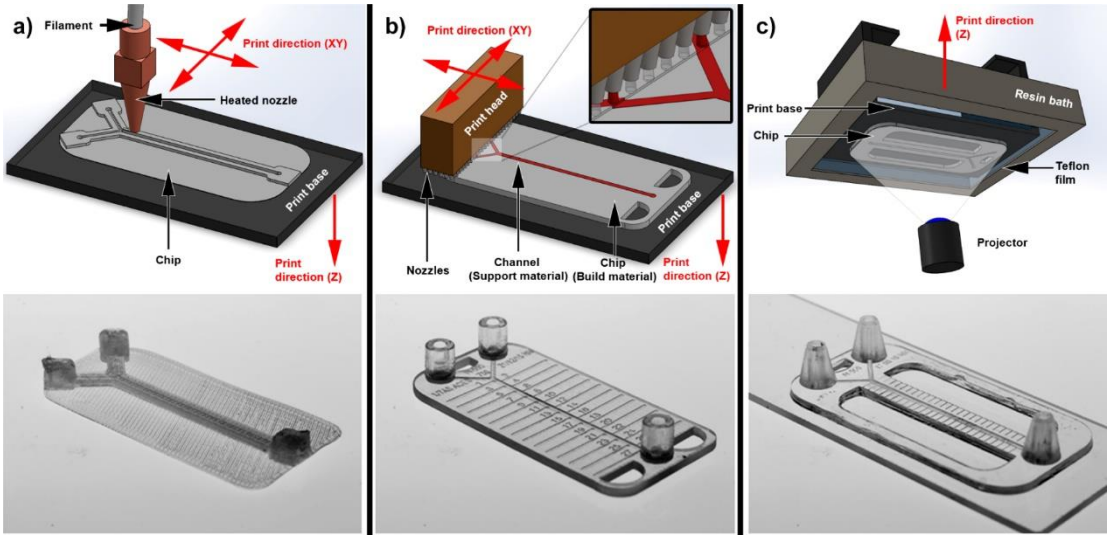


Figure 3: Different 3D printing methods [4].

While the microfluidic chip is an important part of the whole microfluidic system, to perform a corresponding experiment a microfluidic control system is required. Many commercial products for controlling flow in a microfluidic chip exist. As of to date, there is no commercial microfluidic control system product, which is being labeled as portable, exists. Indeed, the most widely used systems are the in-lab system.

Most of the controlling systems are based on a microfluidic flow control system and a microfluidic pressure controller. As for example, the company Elveflow® provides the flow control system with pressure stability up to 0.005% and a sensor resolution up to 0.006%. The next Figure illustrates the principle of such an in-lab microfluidic control system.

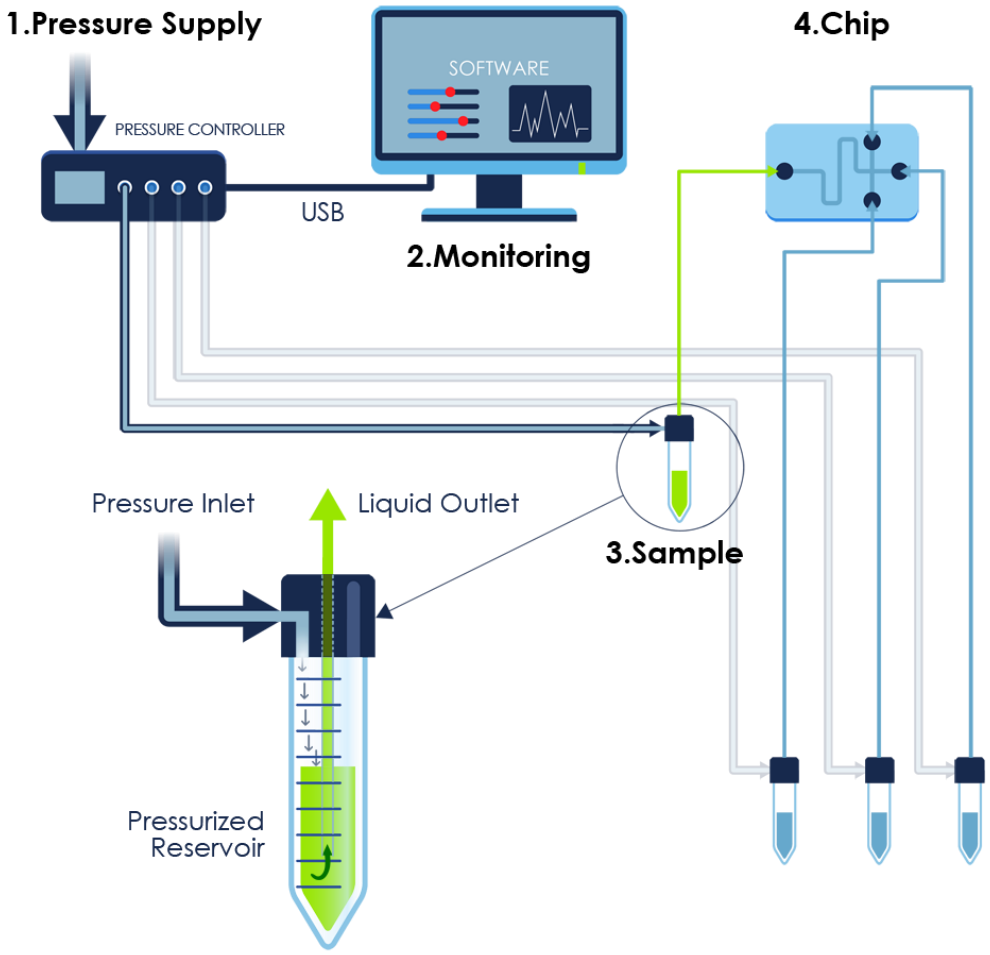


Figure 4: An illustration showing the principle of a conventional microfluidic system.

Illustration based on [22].

Typically, the pressure controller is connected to a computer, loaded with corresponding controlling software to program the experiment. According to Elveflow®, such a pressure controller has a dimension (length × width × height) of 240 × 223 × 80mm, and weight of 3.1kg [22]. Furthermore, the system can have other accessory devices connected, such as a flow or pressure sensor. The next Figure shows an illustration of such a setup.

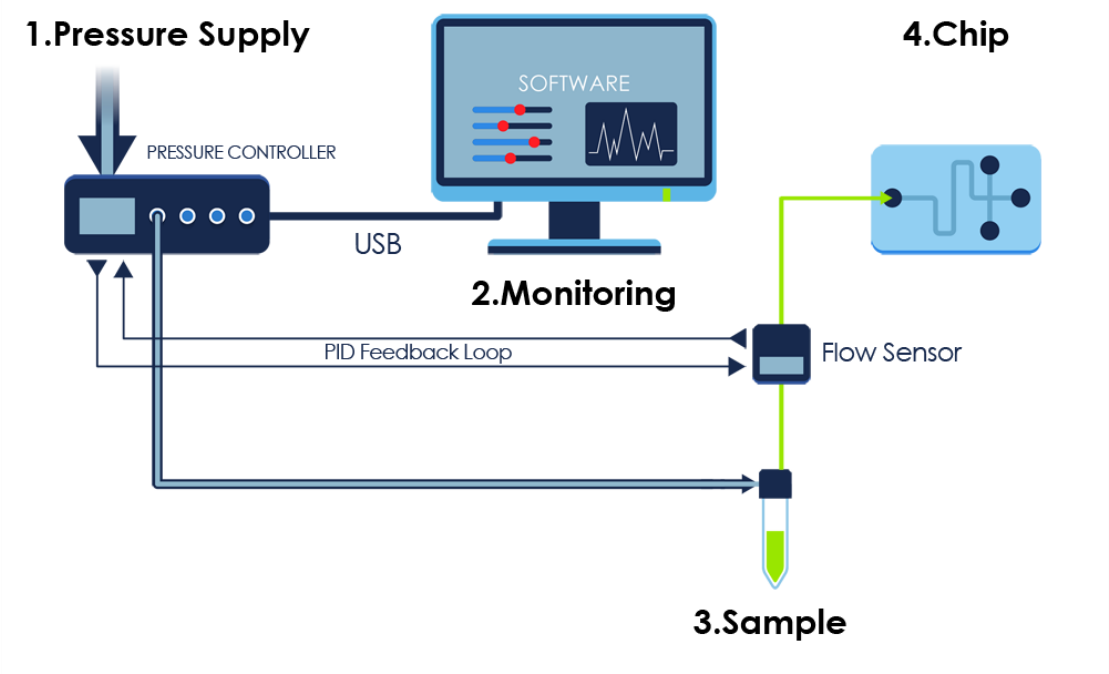


Figure 5: An illustration showing a flow sensor is connected in a microfluidic system.

Illustration based on [22].

In conclusion, currently, the fabrication of a microfluidic chip with conventional methods is a very cumbersome process. For a normal lab-on-chip user, to produce a chip using any of these methods is difficult. Not only because these techniques require the user to master dedicated skills and to have expensive resources but also require a special environment such as a cleanroom. But by using 3D printing techniques, users can easily fabricate a microfluidic chip with a consumer-grade 3D printing device. And in this thesis, I work with this new method of fabrication.

Besides this, the commercial state-of-the-art microfluidic system can only be used in-lab. With its big size and heavyweight, it is not suitable for using on-the-go. Thus, in the following chapters, I discuss a new idea of bringing all this into a small-form-factor system, which I call it a portable microfluidic system.

3 Design Concept

In this chapter, I talk about the concept and design of my prototype portable microfluidics system. I start with the hardware and software of the programmable controller and then continue with the design and specification of the 3D-printed microfluidics chip, finally I discuss the connector design, control mechanism, and some experiment designs, which show the possibilities and feasibility of my system.

3.1 General Idea

A conventional microfluidic system, as I have talked about before, consists of multiple components. The most important component is the microfluidic chip, which, then, is surrounded by big stationary pumps and maybe a computer to automize the control of these pumps.

In my concept, I design everything in a portable fashion. As I have discussed before, currently, the microfluidic chip itself is small and lightweight, but the equipment needed to perform the experiment on the chip is large and not portable in the sense of transportable at all.

Producing a microfluidic chip in a conventional way by a factory requires many steps and processes. This makes the access of a microfluidic chip very costly. Instead of using factory-produced chips, in my prototype system, I use 3D-printed microfluidic chips. The 3D-printer has the advantage of easy to access and the user can print the desired chip in-home or anywhere with a 3D-printer on-demand at a relatively low cost, which fulfills my concept of portability.

Further, I minimize the pumping system. My prototype system uses a small, transportable

device, which controls small sized pumps. This device has a touch screen for user interaction and allows the user to program the desired experiment on it and let it perform the experiment automatically on the microfluidic chip. This means I integrated the pumping system and a PC into one small-sized device. I call it a portable programmable controller.

All in all, my prototype has two main parts: (1) a 3D-printed microfluidic chip and (2) a small-sized and lightweight portable programmable controller. Figure 6 shows the concept of my prototype.

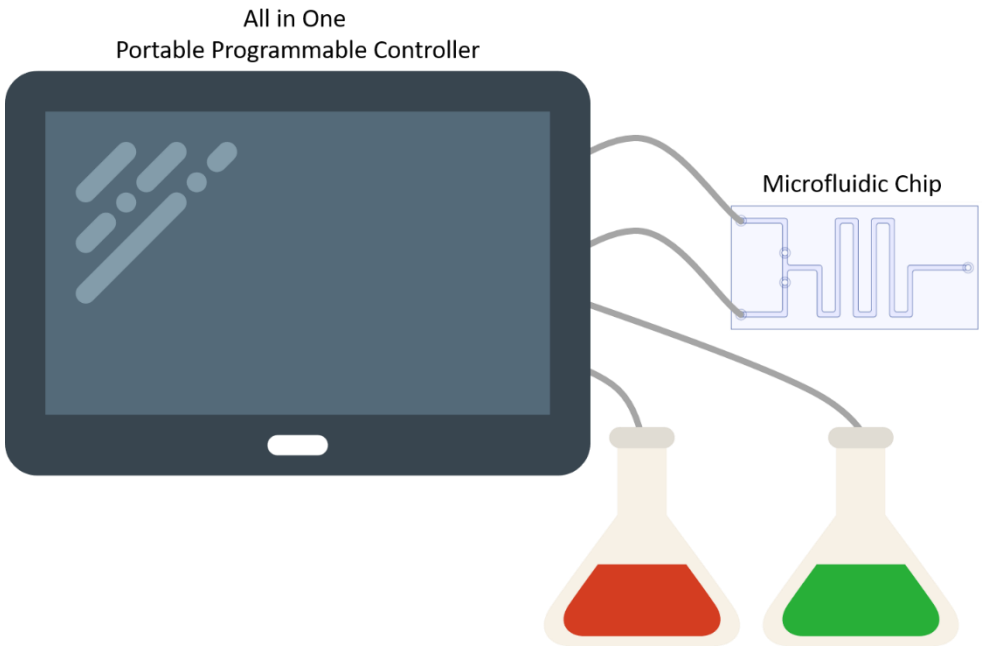


Figure 6: Concept of a portable programmable controller with a 3D-printed microfluidic chip.

3.2 Programmable Controller

3.2.1 Hardware Design

A conventional microfluidics system uses one or multiple stationary pump systems, which deliver the fluids into the microfluidic chip. To automize the delivery process, pumps are

connected to a PC. With the corresponding software, pumps are then controlled by the PC and software. The following figure demonstrates the workflow of a traditional microfluidic system.

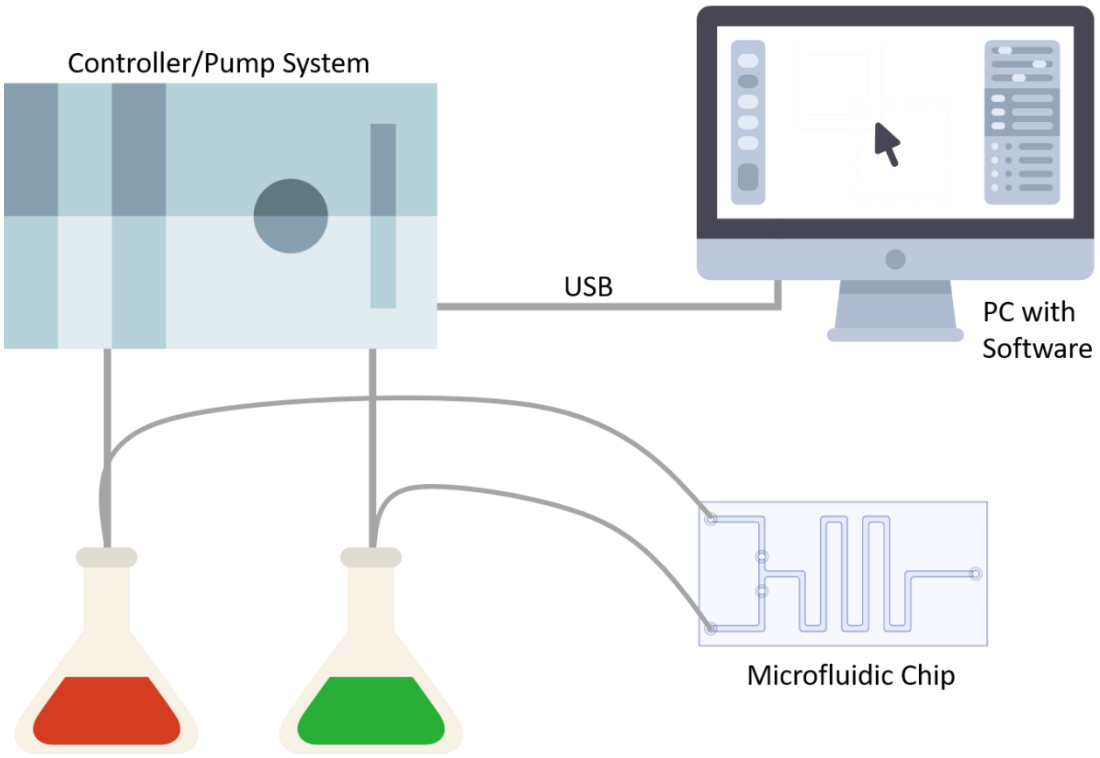


Figure 7: Illustration of conventional microfluidic system.

In my prototype, I realize the basic function of a conventional microfluidic system into a small form factor device. This device includes a fully functional computer for programmable and automized control of pumps with a touchscreen for human interaction, four small-sized pumps to demonstrate the ability of fluid and airflow control and a rechargeable power supply unit to allow the device being portable. With this design, I can perform the desired experiment on the go. To realize this design, I use the following parts, denoted in Table 2.

Component	Specification	Function
Raspberry Pi 3 Model B+	SoC: Broadcom BCM2837. CPU: 4× ARM Cortex-A53, 1.2GHz. GPU: Broadcom VideoCore IV. RAM: 1GB LPDDR2 (900 MHz) Networking: 10/100 Ethernet, 2.4GHz 802.11n wireless. Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy. Storage: microSD. GPIO: 40-pin header, populated	The heart of my prototype: runs software and controls the pumps
7-inch Capacitive Touchscreen	Resolution: 1024 × 600 px Screen Type: IPS LCD Touch Type: Capacitive Multi-Touch: Yes, up 5 points Input via: HDMI, Micro-USB	Touchable monitor for user-interactive programming
Aquatech Miniature Peristaltic Pump RP-Q1.2N-P20A-DC3V	Flow speed: 0.2 ml/min Pressure (max): 500 mbar Power Consumption: 0.12 W Rated Voltage: 3 V – 5 V	For transfer the fluid and air into the Microfluidic chip
L293D quadruple half-H driver	Supply-Voltage: 4.5 V to 36 V Channels: 4 Output Current: 600 mA per channel Peak Output Current: 1.2 A per channel	To amplify the signal of the Raspberry Pi and drives the pumps
5 V Micro USB Power Supply	Output Voltage: 5 V Output Current: 3.1 A	Power supply for the entire prototype

Table 2: Components used for the prototype system.

I decided to use a single-board-computer-based concept. The single-board-computer I use in this thesis is the Raspberry Pi 3 Model B+, which is widely used for DIY maker projects. This single-board-computer has a Broadcom BCM2837B0 quad-core 64-bit at 1.4GHz processor with 1GB of RAM, which runs a full-sized Linux system and is powerful enough to run a self-written software. Further, it features a Broadcom Videocore-IV graphics processor to provide graphical output. And most importantly, it has 40-pin GPIO (General Purpose In-/Out-put) ports, which I use to send signals to the pump (in fact, to the driver IC, which amplifies the signal and drives the pumps). Figure 8 shows the single-board-

computer Raspberry Pi 3 Model B+ from above.

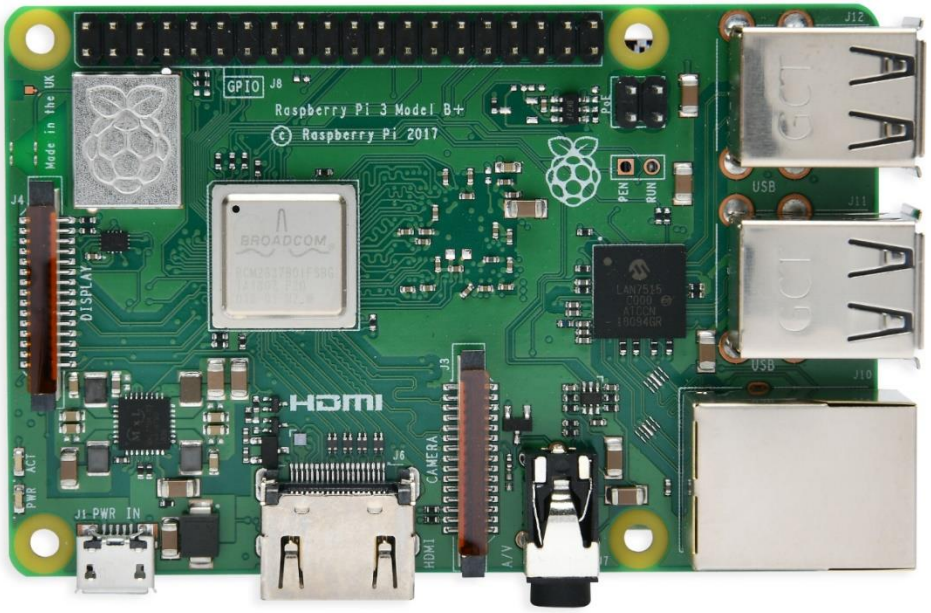


Figure 8: Raspberry Pi 3 Model B+.

Along with the single-board-computer, I use a 7-inch touchscreen to provide user interaction. This 7-inch touchscreen has an IPS panel and has a resolution of 1028 × 600 pixels. The touchscreen provides a capacitive touch function and can be connected via the HDMI connector. The touchscreen gets its power simply through the USB connection, which also enables the touch function. This means that my device will supply a user interactive software interface, where the user can input and operate via the touchscreen. Figure 9 shows the touchscreen that I am using from above.

Figure 9: 7-inch IPS capacitive touchscreen.

The pumps in our proposed system can be changed and replaced with different models depending on the usage of the user. For budget reason, the four pumps that I use here for the prototype and for the later demonstration of feasibility and usability are from Aquatech, the model RP-Q1.2N-P20A-DC3V. This pump model, though is not the most precise one, is a peristaltic type of pump, and provides a flow speed up to 0.2ml/min with a pressure up to 500mbar. Furthermore, it consumes up to 0.12 watt of power. The rated voltage of this pump is from 3V ~ 5V. Figure 10 shows this pump from the side.

Figure 10: Aquatech RP-Q1.2N-P20A-DC3V.

While the pump consumes 0.12 watt, the GPIO pin of Raspberry Pi 3 Model B+ provides just 3.3V, and the current it will drain can be up to 0.04A with I = P / U. According to the specification of Raspberry Pi 3 Model B+, the average output current of GPIO pins is only 3mA. Moreover, the GPIO pins are meant for signal exchanges and not for direct driving of devices. Thus, to not fry the single-board-computer, I decided to use a driver/amplifier IC to amplify the signal sent from the GPIO pins.

The driver IC I use for my prototype is the well-known L293D. The L293D is a Quadruple Half-H Drivers IC and is designed to provide bidirectional drive currents of up to 600mA at voltages from 4.5V to 36V, which perfectly matches to the pumps. The bidirectional current supply allows the pump to both push and pull. Each L293D can drive, in my application, 2 pumps, one pump on each side. For the power supply, the L293D is connected to the 5V power output pin of Raspberry Pi 3 Model B+. Figure 11 shows an L293D IC.

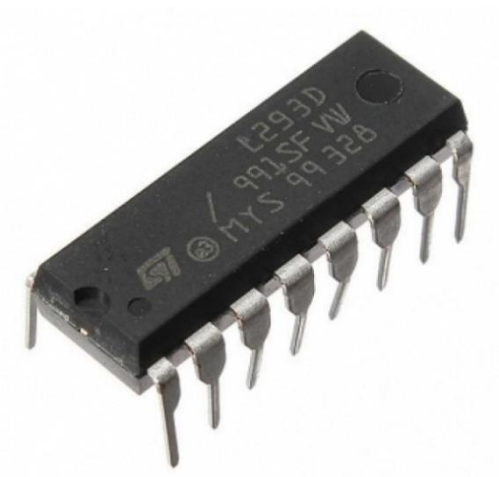


Figure 11: L293D driver IC.

This means I use constant 5V to drive the pumps. In fact, a big concern for this setting is to regulate the speed of the pump. Since the driver IC only amplifies and outputs the voltage I supply to it, I am not able to regulate the voltage to change the speed of the pumps. This way of regulation of speed is called the DC regulation. Instead, I focus on

the signal sent to the IC, and will discuss the way of speed regulation of the pumps later in section 5.2 “Hardware and Software”.

Figure 12 shows a functional diagram of L293D, which depicts how a pump is connected to one of the L293D and where the signal and power come in, and Figure 13 shows the pinout of the L293D.

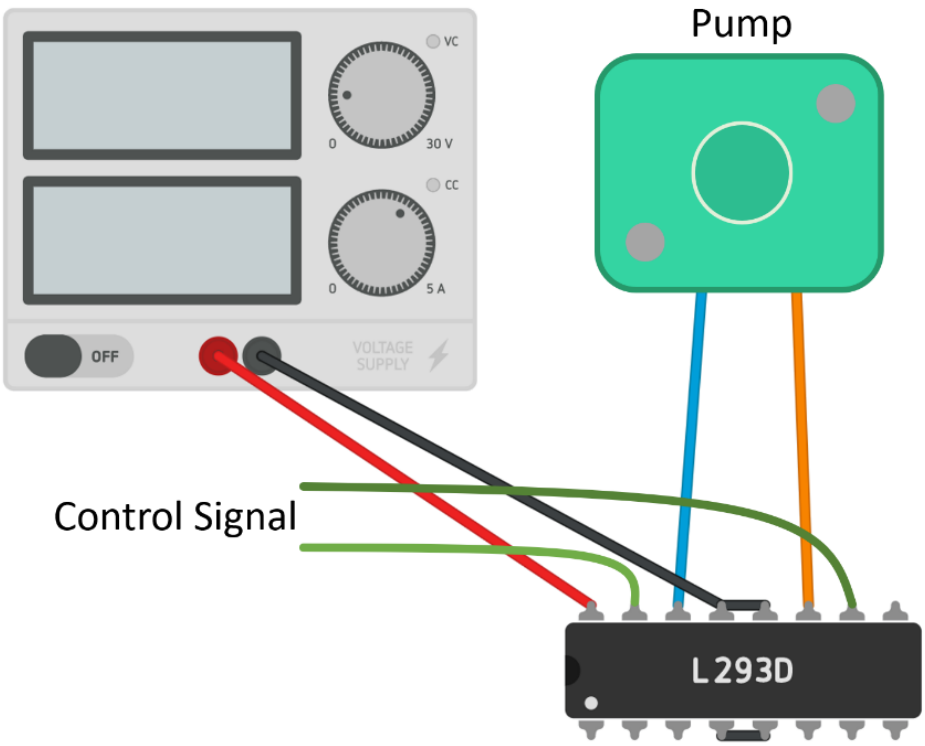


Figure 12: Illustration of how L239D and the pump are connected.

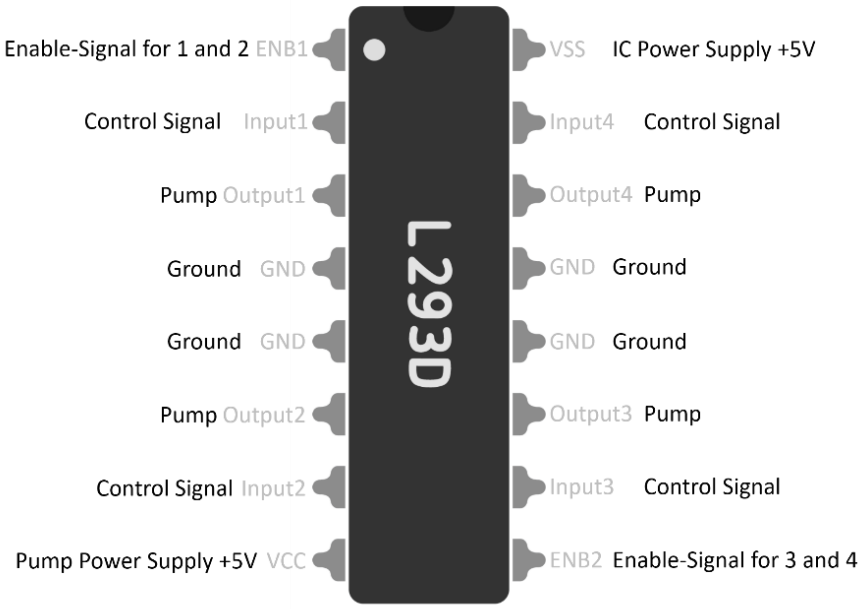


Figure 13: Pinout of L293D.

Finally, for my prototype, I use a 5V/3A USB power supply to drive the whole prototype devices. To allow my prototype to be portable, especially transportable. I can change the USB power supply later to a mobile power bank so that I can use this device on the go. Figure 14 depicts a schematic diagram of how all the components mentioned above are connected.

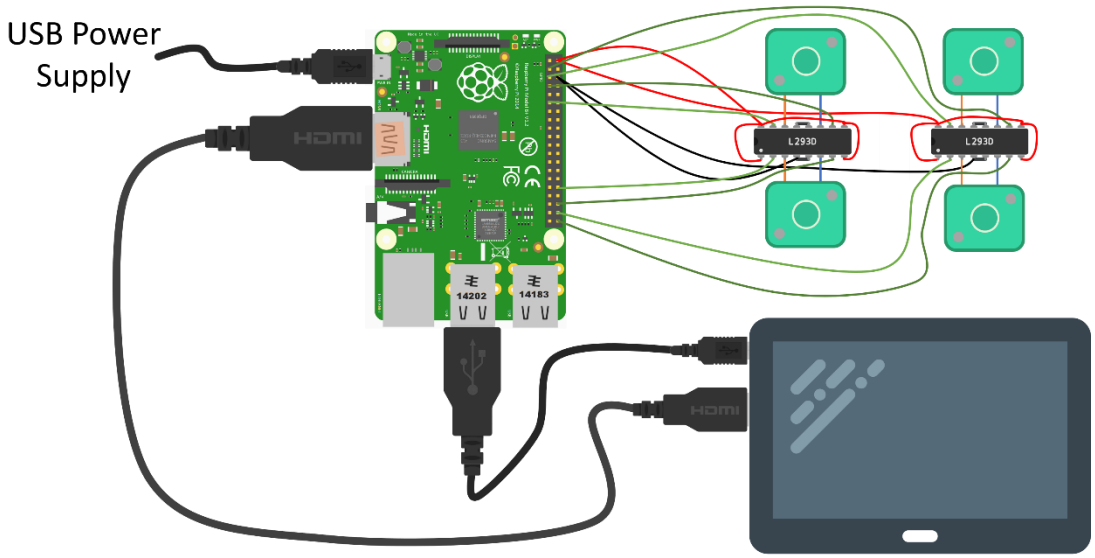


Figure 14: Illustration of how all components are connected.

Further, I design my in- and output connection. For simplicity, I did not decide to use a mature and costly connector. Instead, I choose the medical infusion needle to connect the chip with the pump. The infusion needle I use is the 25G 3/4" 0.5 × 19mm Butterfly Winged Infusion Set from the Japanese company NIPRO. This needle set has an integrated 3/4" long tube directly connected with the 0.5 × 19mm needle. At the ending of the tube, there is a female Luer-Look fitting, which makes the tube-needle-set to be easily replaceable. Therefore, by using this infusion set I do not need an extra tube to connect it between the pump and the needle. Figure 15 shows the infusion set I am using for this work.

Figure 15: 25G 3/4" 0.5 × 19mm Butterfly Winged Infusion Set.

On the pump side, I use a male Luer-Lock fitting, which the female connector screws into. A male Luer-Locks is glued to each of the pump's tubes. In this way, I can easily replace a tube and needle by simply screwing out the old one and screwing in a new one. Figure 16 shows a tube-needle set connected to a pump via the Luer taper.

Figure 16: Infusion set connected with the pump.

3.2.2 Software Design

After I have talked about the design of hardware of the programmable controller in my prototype, I discuss software design in this part.

In this part, I first start with the selection of an operating system I use on my single-board-computer. As I have introduced, the Raspberry Pi Model 3B+ is driven by a Broadcom® BCM2837B0 quad-core CPU, which is based on ARM® Cortex-A53 (ARMv8) 64-bit SoC (System on Chip) architecture. Currently, many Linux distributions, Windows IoT and Windows on ARM operating systems are supporting this CPU. I use the official Linux distribution of Raspberry Pi – Raspbian, which is also the most often used operating system on Raspberry Pi, as the operating system for my prototype.

The Raspbian includes many essential libraries and desktop PC grade applications,

including the support of Python, Java and not to mention, the library for GPIO access on system-level – the WiringPi [23].

Since I am using a touchscreen to provide a user-interactive interface, a requirement of my software is to have a graphical user interface (GUI). The GUI should be in full screen and each component, such as button, should be big enough for the user to touch.

Many makers are using python language coded program to access the GPIO pins, though to provide a GUI with python is not convenient, and many python GUI frameworks are based on other languages such as Java. Furthermore, python is an imperative language, which cannot provide the efficiency which a precompiled program could provide. Thus, for my purpose, I decided to use Java to write the program. Java provides its own GUI library – Swing, and is also cross-platform. That means, when I later want to upgrade my prototype with another single-board-computer, even based on another CPU architecture, I am still able to reuse the Java code. Moreover, there is a library for Java, called Pi4J [24], which provides the GPIO access in Java based on WiringPi. This makes Java the best choice for my application.

To provide an easy and intuitive user experience, my programmable controller needs a well-designed control software. In the following, I elaborate on the design concept of the software.

The main functionality of the software is to provide the user with a programming interface to define a sequence of operation of the pumps. Each operation can have its own action setting, including the model or type of the pump is being used for this action, the flow direction, the speed of the flow and the operation duration. Although, for demonstrative purpose I only use one pump model in my prototype system, the system should be able to connect different models and types of pump. To allow the user to perform experiment

with different types or models of pump, the control software provides the possibility to choose, define and edit the attributes of a pump. These attributes define the model and type of a pump, which includes e.g. the maximum and minimum allowed speed of that pump type. While I am using DC pump, to define the minimum speed is essential, because a signal of a too slow speed might result in a non-rotating state and gives a constant short-circulate which can burn the IC and the pump.

Moreover, all these settings and sequence of operations must be able to be saved and reused in the future. Besides these basic functions mentioned above, there should also be an "expert mode", where each pump can be controlled manually, by its speed and direction.

To promote usability and based on the software requirements stated above, I have given the software the following design. The graphical user interface has five windows, all running in full screen. In order to make it easy to use, I decided to have the software in a wizard-like fashion, where the programming procedure is divided into steps, and the user will be guided step by step. The user will see the first window after starting the program.

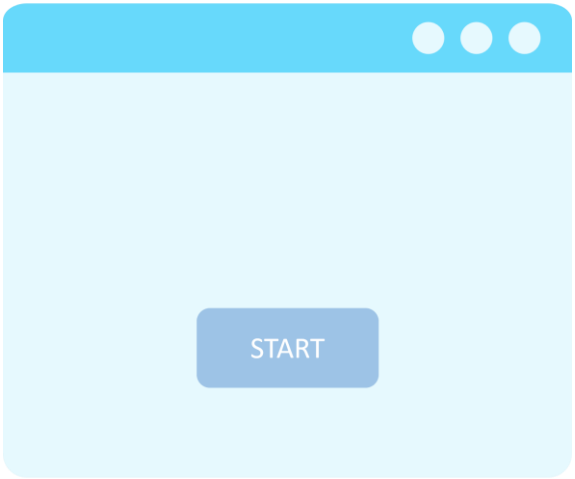


Figure 17: A design of splash screen of the program.

This first window is a splash screen, where a logo and a welcome message are shown.

During this, necessary components, such as stored pump type list, are loaded. A button indicated with a start sign is located in the middle.

After the user has clicked on the start button, a full-screen window for pump-pin mapping definition appears. On this window, the user has many options. The first option is to define where the pumps are occupied, i.e. to define which type of pump is connected to which pins with which maximum speed is allowed for this execution. Moreover, the user can give a name to each pump to better distinguish in the later step.

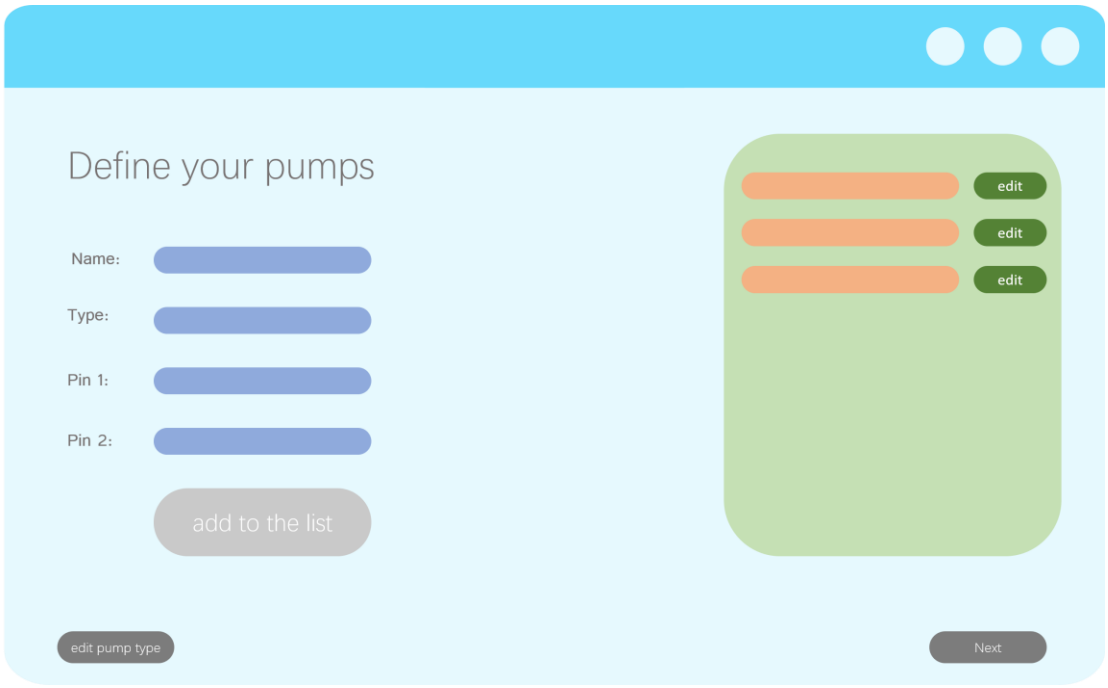


Figure 18: A design of pump definition window.

While the type of pump is involved in this window, the user can also edit the pump type list here. This will open another window, where the user can edit or define a new pump type with the following attributes: (1) type name, (2) allowed minimum speed in percentage (where 100% means full power in 5V) and (3) allowed maximum speed. All this should then be saved in a file automatically, which will also be usable for the next time.

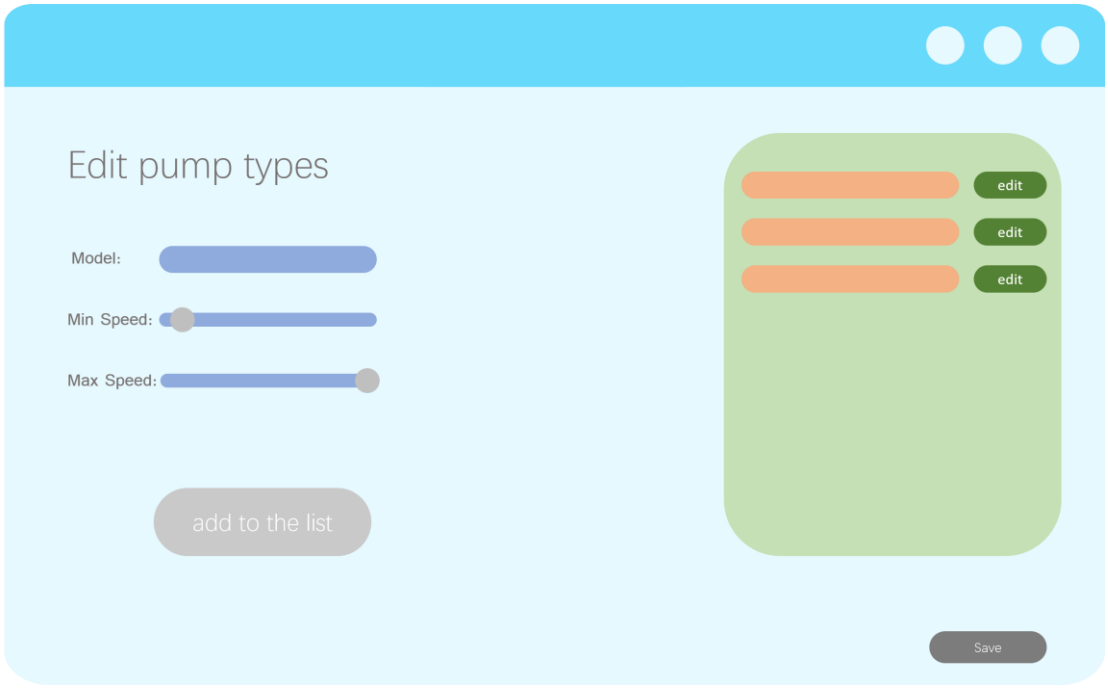


Figure 19: A design of window for pump type edit.

Back to the full-screen window, the user also has the option to save the progress, while he can reuse this configuration for the next execution.

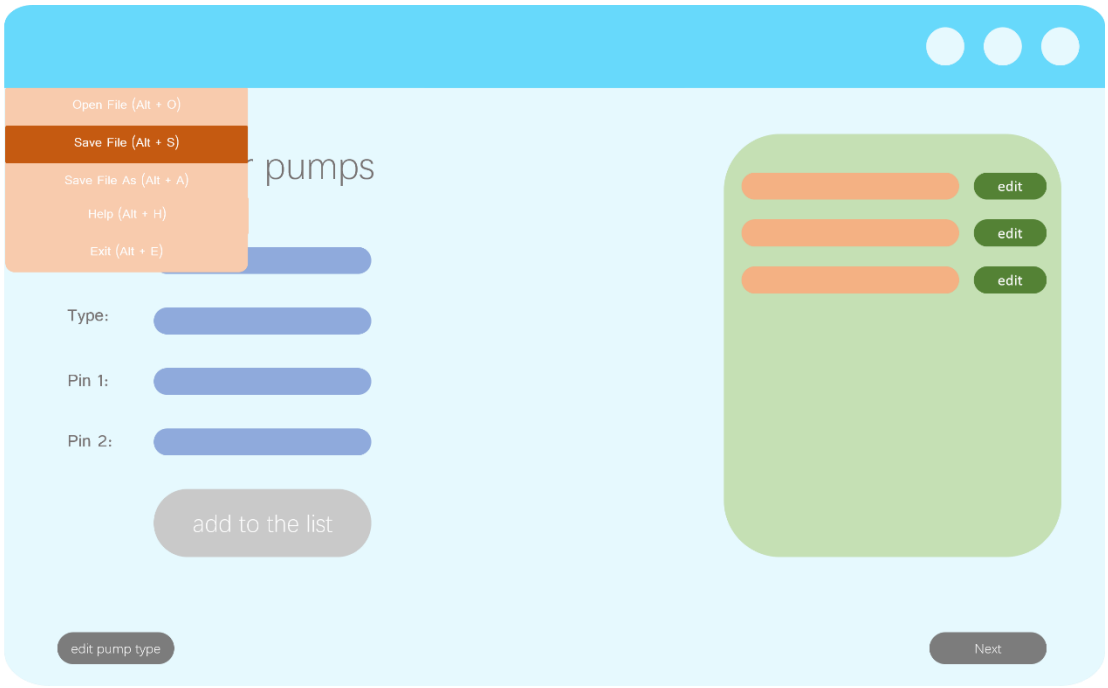


Figure 20: A design of pump edit window with menu.

After the user continues, the main window shows instead. On this window, the user can

program the execution sequence of pump actions. Figure 21 shows a drawing of the main window.

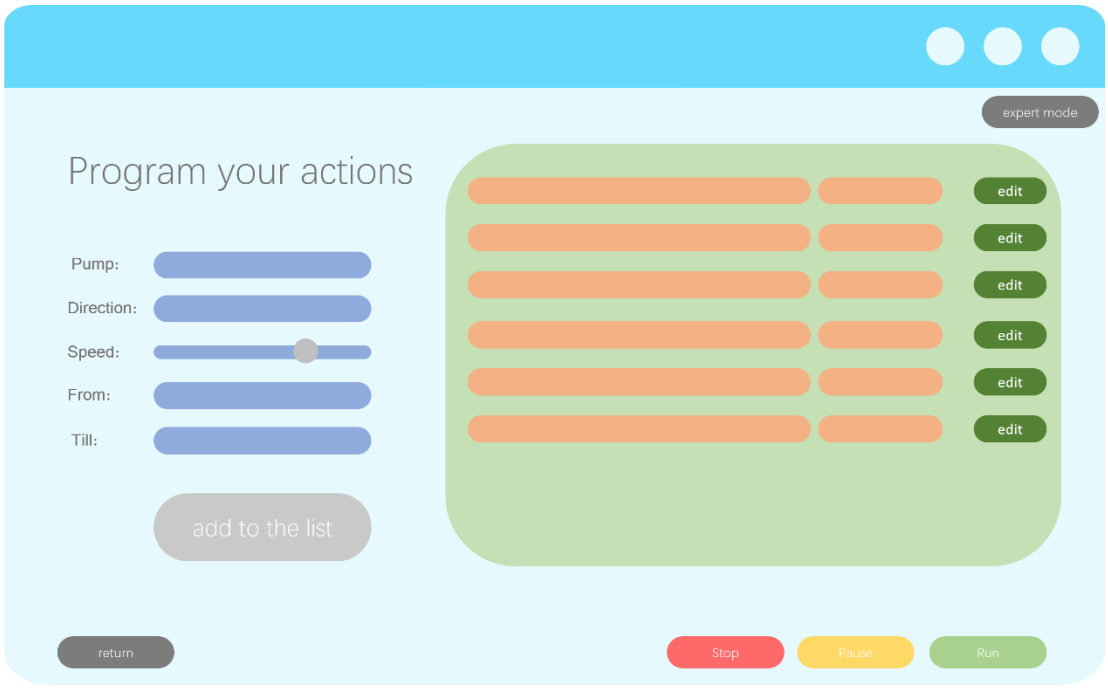


Figure 21: A design of sequence programming window.

The first option here is to go back, where the user can re-define the pump-pin mapping or the speed. The next option, which is a core of this step, is to add actions. An action consists of the action's starting and ending time, the involved pump, the speed of the pump during this action and whether the pump should push or pull the content.

In the action sequence list, the user has the options: edit and delete. The edit option will allow the user to re-configure the action and the delete option will delete the selected action. After all actions have been correctly defined, the user has the option to execute the sequence of actions. While executing the sequence, the pumps will do these defined actions in chronological order. During this, running actions are highlighted to indicate which of these actions are currently executing. A pause option can pause the current action until the execution button is clicked again, and a stop button can fully stop and reset the execution.

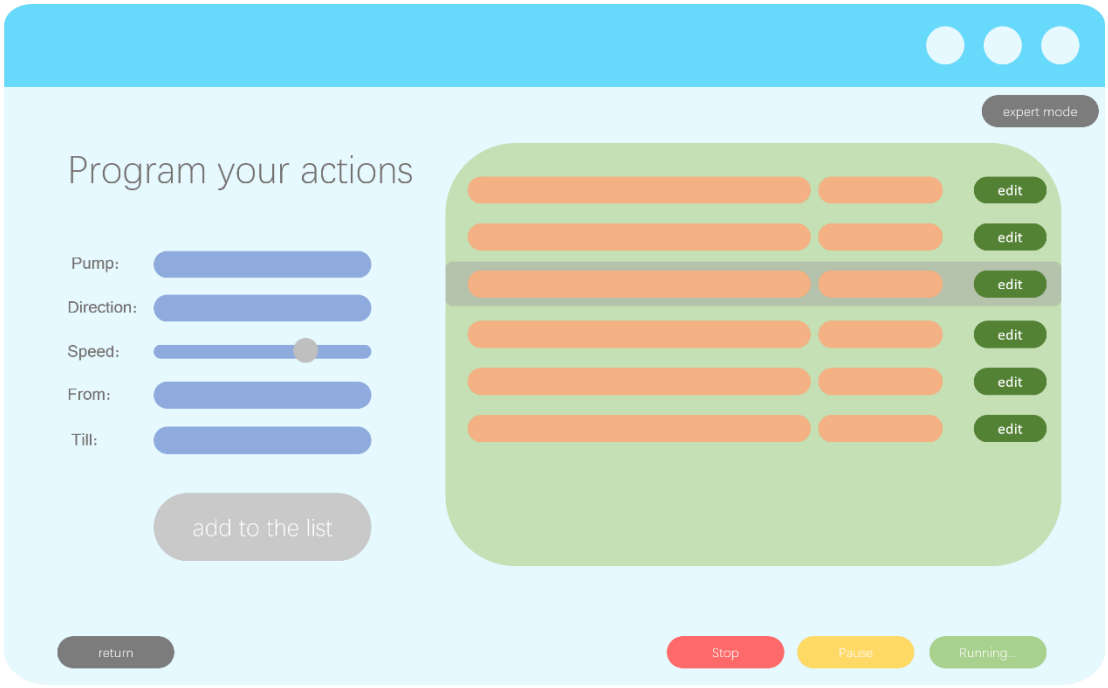


Figure 22: A design of sequence programming window with selected action.

Furthermore, the user can save the configuration. This save function will also save the configuration of pump-pin mapping, as it is a part of the action sequence definition, while the action sequence depends on it.

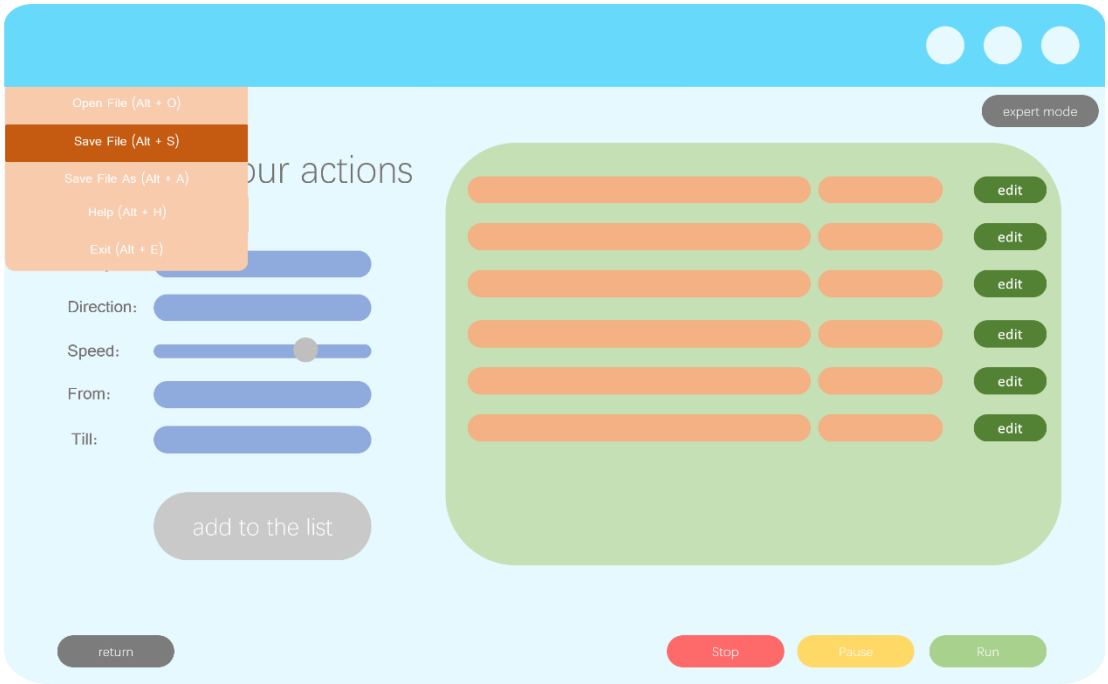


Figure 23: A design of sequence programming window with menu.

Finally, the main window provides a so-called expert mode. Open up this mode, the user will see another window, in which manual control of defined pumps is possible. Here, the user can access each individual pump, which was defined in the previous step, and let it push or pull with the desired speed. The control is as simple as just click on push or pull and slide the speed on the slider.

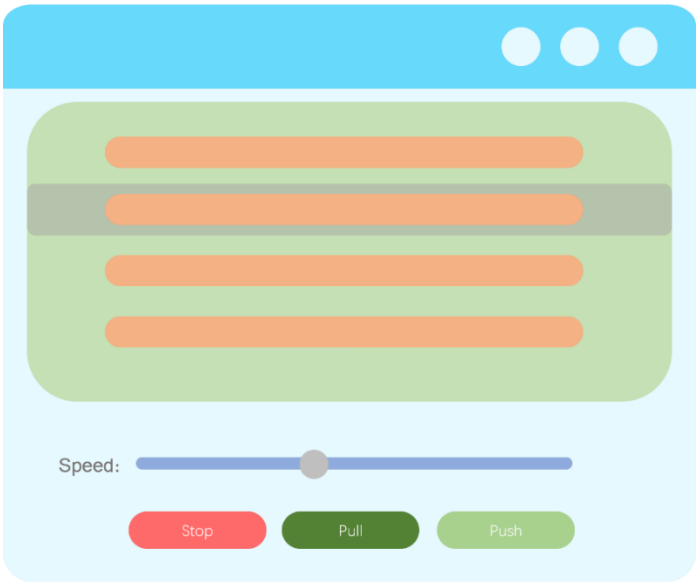


Figure 24: A design of expert mode window.

The Figure 25 depicts the structure of the software interface.

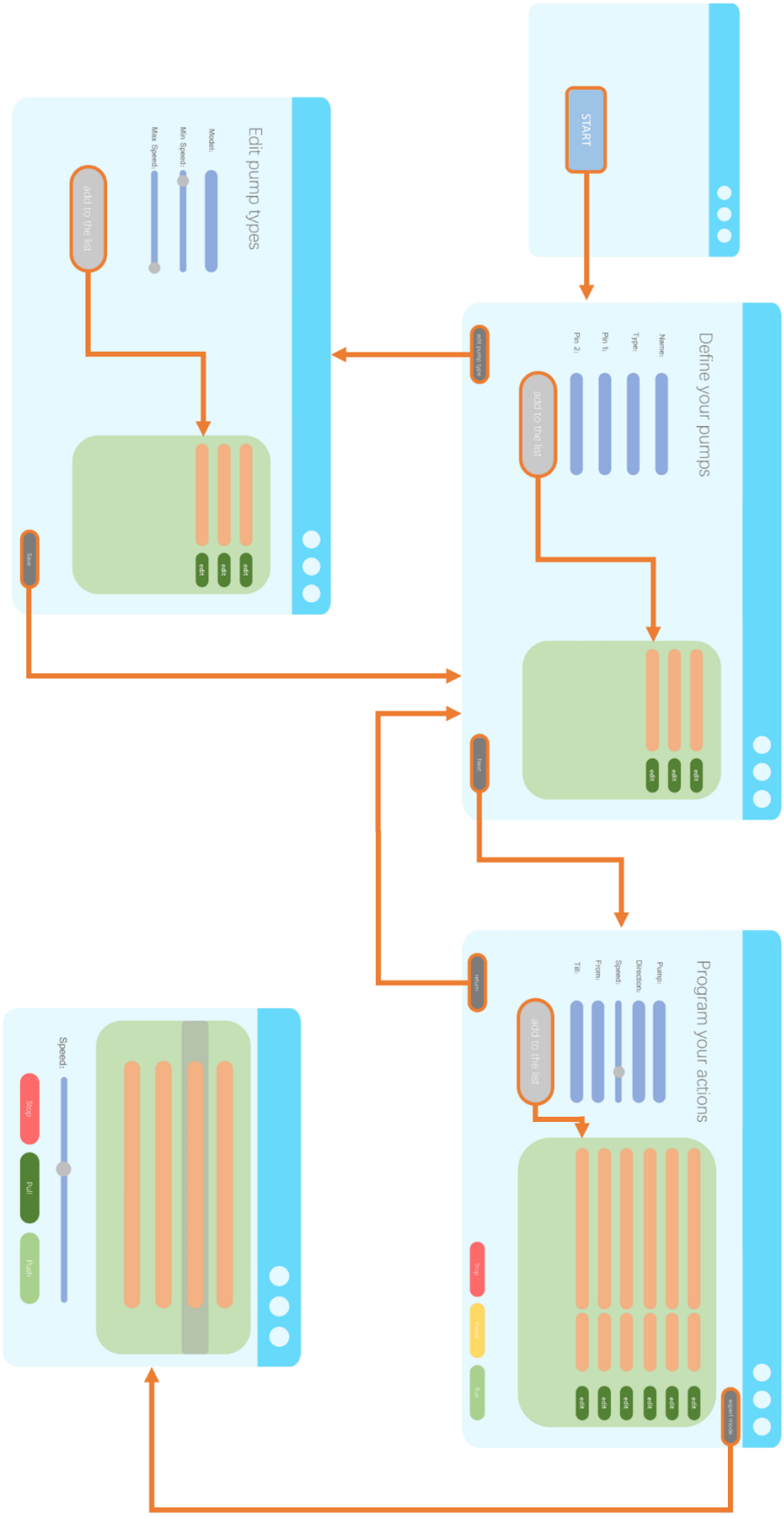


Figure 25: Illustration of the structure of the software interface.

The next Figure shows a user interaction flow of programming and executing a sequence.

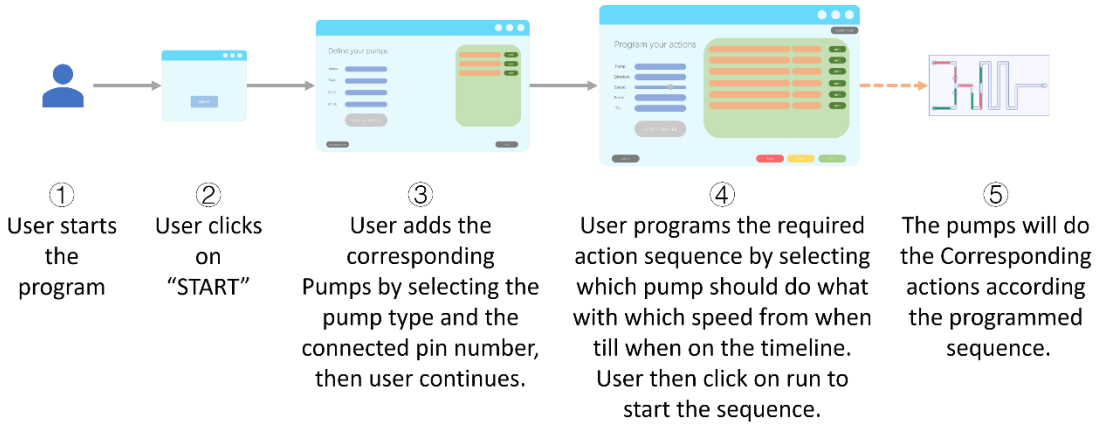


Figure 26: Illustration of user interaction flow of programming and executing a sequence.

4 Implementing Chip Construction with 3D Printing Technology

In this part, I continue with the core of a microfluidics system – the microfluidic chip. In my prototype system, I want to have the microfluidic chip in a portable fashion, which means, it should have the characteristics of easy to access, easy to produce and transportable. While 3D-printer nowadays becomes more and more popular, the resolution of 3D-printer has also increased tremendously. A consumer-grade light-curing 3D-printer, based on LCD technology can have a resolution up to 100µm. In my prototype system, I am using a more mature LDP-Laser 3D-printer.

Like I have discussed in the previous chapter, microfluidics chips are conventionally produced using glass, paper, PDMS or PMMA. With LDP 3D-printer I am using resin to produce my designed microfluidic chips. In this thesis, I work with MiiCraft® + 3D-printer (Rays Optics Inc., Hsin-Chu, Taiwan). MiiCraft® + is an LDP light-curing 3D-printer, which can produce objects up to a resolution of 56µm. For all printing works, I use MiiCraft® genuine transparent resin BV-007. Since the characteristic of light-curing 3D-printer, to directly print multilayers of channels in a chip is nearly impossible, which is explained in the next section. And because of the characteristic of the resin, printed objects are hardened after curing. Therefore, some modules/devices that appear in conventional factory-produced microfluidic chip need to be re-designed for the 3D-printed chip. I designed my chips with computer-aided design (CAD) software for Windows (Autodesk® Inventor® Fusion, Autodesk Inc., San Rafael, CA, USA). In the following, I first discuss how to design a 3D-printable chip and elaborate on the facts and requirements on chip design. Afterwards, I talk about how these important modules need to be re-designed in order to provide similar functionality on a 3D-printed chip. Lastly, I show some chip designs which later demonstrate the functionality of my prototype system in combination with my programmable controller.

Category	Item	Specifications
Printing Area	X/Y Resolution	450ppi (56 microns)
	Z axis (layer thickness)	5 microns as one step from 5~200 microns 30/50/100 microns with MiiCraft Resin
	Speed	3cm / hour (Z@100 microns)
		2cm / hour (Z@50 microns)
	Build Size	43mm × 27mm × 180mm
	Material	MiiCraft Cream Resin (Z min. = 30 microns)
		MiiCart Blue Resin (Z min. = 30 microns)
		MiiCraft Clear Resin (Z min. = 50 microns)
		*3rd party resin verified by MiiCraft
Post Curing Area	Power of Light	18W UVA Lamp
Software	Feature	MiiCraft Builder powered by Materialise: - Support generation - Model slicing - Auto repairing during model import process - Printing Parameter setting
		MiiCraft Printer Control Software: - Network Print (Browser) - Printing Status Monitoring
	Input File type	STL
	Operating System	Windows 7, Windows 8
System Properties	Printer Size	20.8cm (L) × 20.5cm (W) × 33.5cm (H)
	Printer Weight	8.5kg
	Interface	RJ-45
	Power input	100-240V, 50/60 Hz, 2.0A
Operating Environment	Temperature	15-25°C
	Humidity	40%-60% RH

Table 3: Fabrication, and production factors of microfluidic devices for the 3D-printer I use in this work.

4.1 Design of 3D-Printable Chip

Resin 3D printing is known for its great ability to achieve very nice, highly detailed parts, all thanks to a technology called vat polymerization, which is more complex than fused deposition modeling (FDM), the most common (and least expensive) 3D-printer technology.

Vat polymerization utilizes a photopolymer in the form of a viscous fluid, resin, and a light source to cure the resin and therefore build the desired parts.

There are two main types of vat polymerization technology: stereolithography, or SLA, and digital light processing (DLP). The primary difference between the two is the light source used to cure the resin and produce the desired part.

In my work, as I have stated, I use a DLP light-curing 3D-printer from MiiCraft® – the MiiCraft® +. The light source DLP 3D-printers use is a digital light projector screen. Whereas in SLA, a laser cures only a single point at a time, a DLP printer's screen projects the image of an entire layer all at once. As a result, DLP printers generally print faster than SLA printers.

The device used for controlling where light is projected in DLP 3D printers is called a digital micromirror device (DMD). This is sort of like the galvanometers in an SLA printer, but much more sophisticated.

The DMD is at the "heart" of every DLP chipset. A DMD contains hundreds of thousands or even millions of small micromirrors that direct the light and create the pattern of a layer.

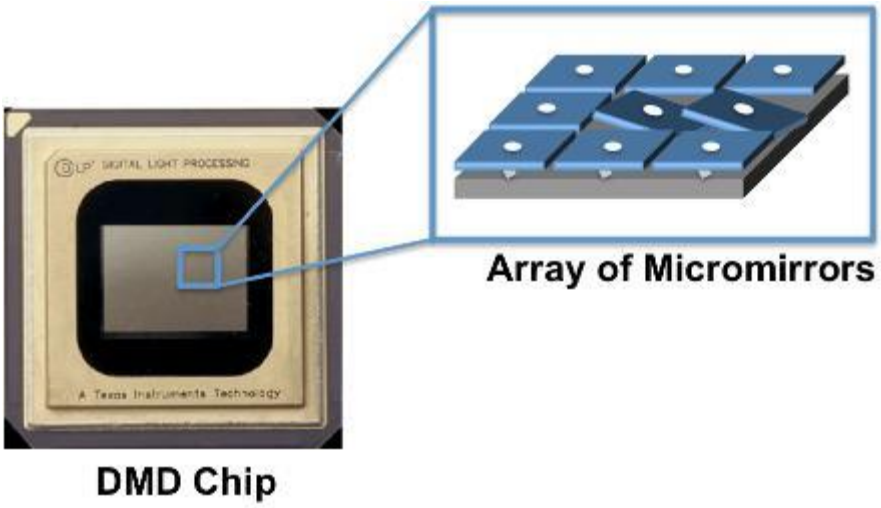


Figure 27: Illustration [25] of a DMD Chip.

Since DLP 3D printers use digital light projector screens as a light source, the image of a layer is made from pixels. Once 3D, these become voxels — tiny rectangular prisms visible in all three axes. To produce a 3D object, the build platform is poured into the vat leaving only one layer in height between itself and the bottom of the vat. When the image of a layer flashes, it cures the resin forming a solid layer. The build platform moves up and the process is repeated until the part is complete.

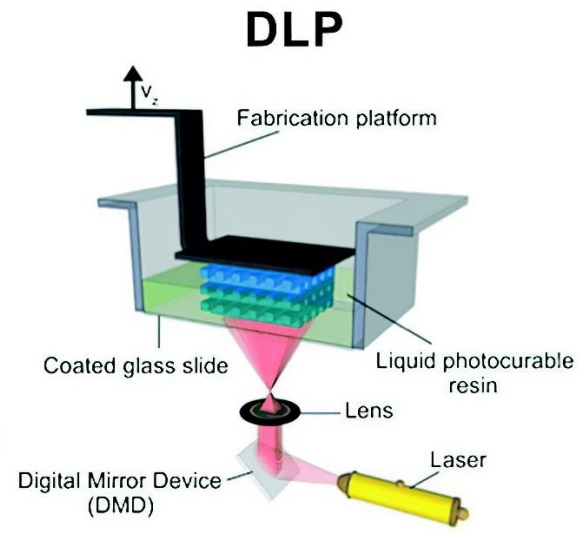


Figure 28: Illustration [26] of how a DLP 3D-Printer works.

The accuracy of MiiCraft® + is not that high, though its resolution is stated as 56 × 56 ×

50μm, in my actual test, but because of the temperature and characteristics of the resin, I am able to achieve a clear channel up to a diameter of 300μm. Thus, in my chip design, I define channels with a diameter greater than 300μm.

I use the MiiCraft® resin BV-007, which is a clear-transparent resin, to produce my chips. This resin has the advantage of being transparent, and thus we can clearly see what is going on in the chip. Nevertheless, the characteristic of transparency also has its side effects.

As I have talked about above, the printing procedure of a light-curing 3D-printer is by curing layer by layer. A 3D-Object, in digital form, will be sliced into layers with a thickness of given accuracy of resolution, e.g. in my work 50μm.

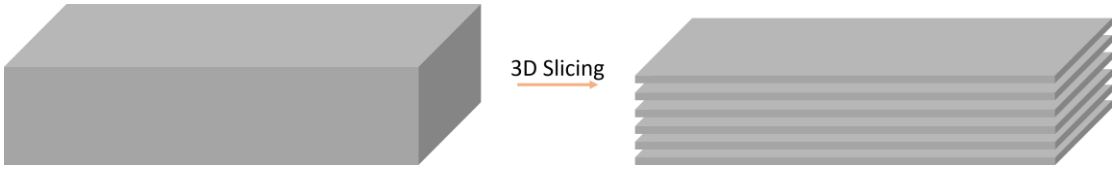


Figure 29: Illustration of a 3D object slicing.

Then, the printer cures every layer by lighting the resin with a given power of light and time, as described before. While the layers should be lighted and cured one after another, this is not the truth for transparent resin. The transparent property of the resin allows light to travel through it and penetrate previous layers. While this penetration will help to harden previous layers, undesired curing occurs, as well. This situation is especially undesired when areas in previous layers are not wished to be hardened, i.e. places where should be empty, are cured during lighting further layers. This will affect my work in the creation of a clear channel inside a chip. Figure 30 depicts the process of curing a transparent resin with the above-described side effect. For better viewing, the curing object is depicted as brown color, though in reality, it is transparent.

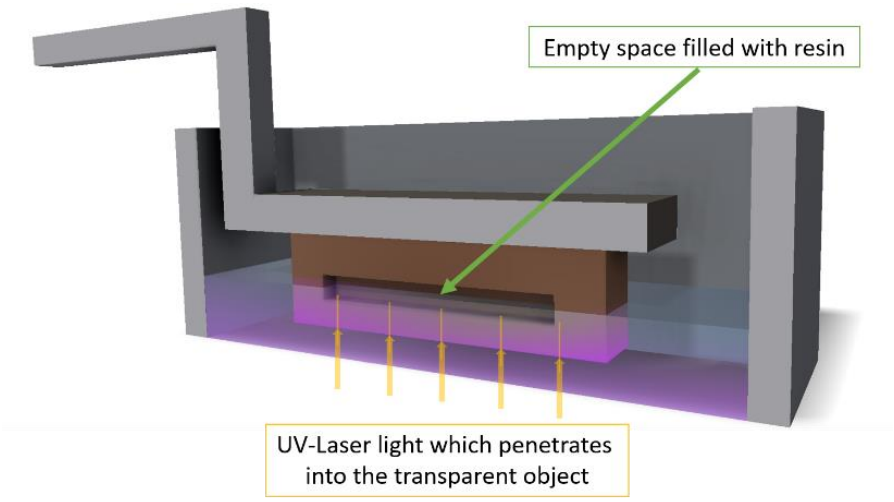


Figure 30: Illustration showing how blockage appears in a channel.

As Figure 30 is depicted, when a channel or a space is embedded in the object, resin will be left inside the space. This is since the resin bath will immerse the lastly printed layers. While the layers which have the empty space will leave the space over, but the next layers which seal the space above will also seal a part of the resin fluid into the space. This is due to the strong cohesion and adhesion characteristics of the resin. The adhesion force makes the fluid stick to the wall of the space. When the empty space is minimum, e.g. my fluid channel of the microfluidic chip, which has a diameter of ~400μm, will be fully filled with the resin. Since the resin is transparent, the UV light will cure the latest layer and go through the layer which is curing and penetrate previous layers. Though the penetration will be weakened layer by layer, this penetrated UV light will still cure the fluid in that nearest layer with empty space gradually.

Like the photo imaging process, the curing process of the resin depends on the power intensity of the UV light and the exposure time [27]. While the UV light is penetrating to the inner layers, the power intensity of penetrated light will gradually weaken. But since the time of exposure matters as well, the resin fluid left inside will still harden by weakened UV light when it has been exposed for a long period of time [27]. The layer will be exposed if the light can still travel through the layers after it. The number of layers depends on the penetration depth of the resin. Unfortunately, I am not able to retrieve this important

attribute of the resin I am using. But, during my 3D-printing experiments, it shows that the penetration depth of the resin I am using with the UV light in my MiiCraft + is much higher than four- or five-layer thickness. That means for my application, printing a microfluidic chip which has a channel diameter of 400μm, that is exactly eight layers, cannot be realized when the chip, for instance, has a total thickness of 6 millimeters and the channel is settled in the middle between the top and the bottom, as shown in the Figure 31. In such a case, the resin remaining in the channel or empty space become cured and a clear channel will not be possible to achieve. I will talk about the shape and dimension choice of my 3D chip including connector design later in this chapter.

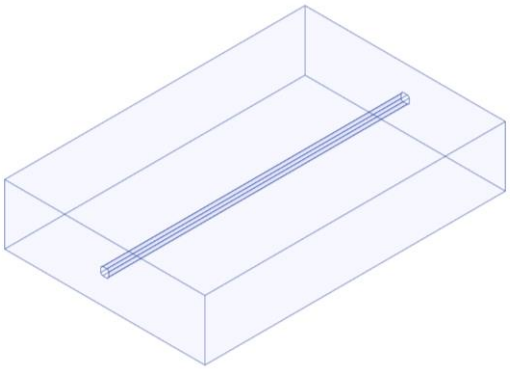


Figure 31: Illustration of a CAD designed microfluidic chip with a channel in the middle.

To solve this problem, I propose two different but similar solutions. Both are based on the minimization of light exposure and the time of exposure.

The first solution I propose is to shift the channels to the base, which is the layers exposed at the very end. As depicted in Figure 32, shifting the channels towards the end/base of the object will largely reduce the time of exposure to the empty spaces.

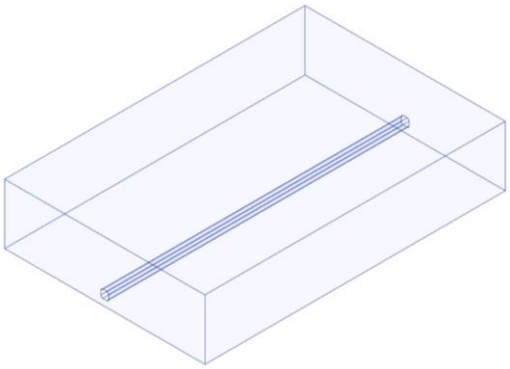


Figure 32: Illustration of a CAD designed microfluidic chip with a channel at the base.

Only the last few layers will allow UV light penetration to the space, and after the last layer has been printed, the 3D-printer will stop emitting UV light and because the exposure time to the resin in the empty space was not enough to fully cure the resin, I am able to wash out the remaining fluid inside by using isopropanol alcohol. In this way, I can have a clear channel at the end.

Another solution I propose is to let the channel fully unexposed. This means I design my chips in that way where the channels look like being engraved on the chip. I do not seal the channels into the chip but to let it be a groove on the chip. Figure 33 shows the concept of the solution, while to better see the channel I put the chip upside down, as the channels will be printed at the last.

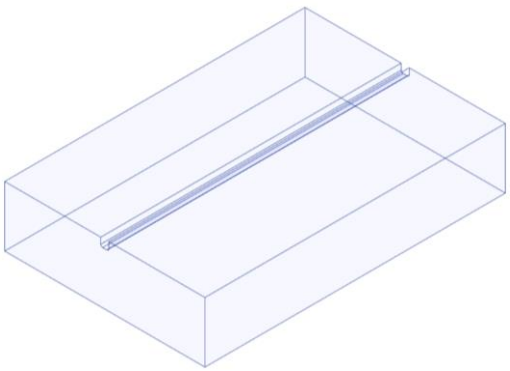


Figure 33: Illustration of a CAD designed microfluidic chip with a channel on the top of the chip without sealing.

This solution left the channels being opened. To seal the channels (this process is called lamination), I use other materials, such as a piece of clear tape. After the tape is fully pasted to the chip and sealed the channel, I, again, use isopropanol alcohol to wash out the remaining glue on the tape.

With these two solutions, I am finally able to print chips with a clear channel on a scale of hundreds of microns.

Now, I define how the chip is connected with the pump. While I decided to use the Butterfly Winged Infusion Set as my connection, this infusion set, as I stated before, has a needle with a diameter of 0.5mm. This needle needs to be inserted into the chip so that the fluid can be pumped through this connection into the chip's channels.

Accordingly, I design the connector on the chip as a hole of 0.5mm diameter, which passes through the chip from the surface to the channel. At the top side, I give the connector a chamfer by 0.5mm/20 degrees for easier insertion of the needle. The next Figure shows how a proper printable chip is designed. This chip has a channel diameter of 400µm and a connection diameter of 500µm. The needle can directly stick into the hole and being held by the thick chip body.

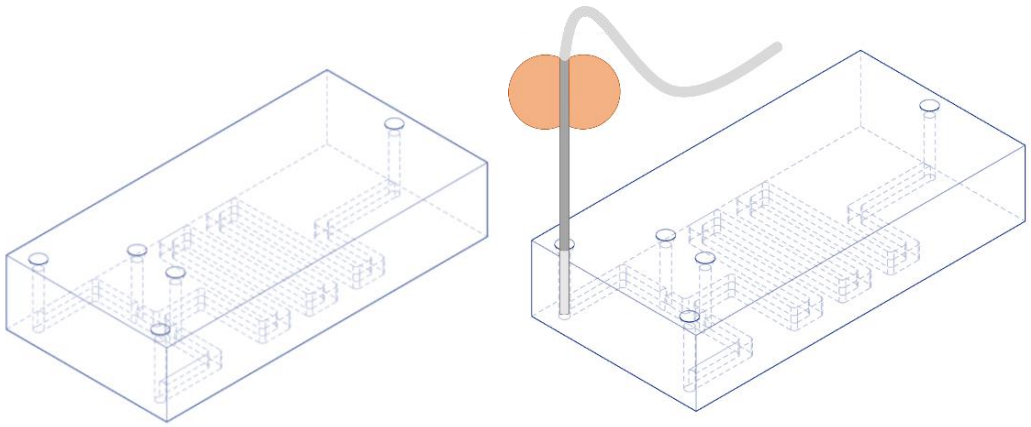


Figure: Illustration showing how the connection is designed and how it is used to connect the pump using the infusion set connection.

After I have defined how my chip should be designed in order to be printable and usable for my prototype, I can start to design experiments and the corresponding chips to demonstrate the use of my prototype system. To direct the fluid flow in the chip, I need to design a control mechanism to replace the valve of a conventional microfluidic chip, so that I can control the flow of my fluid inside the chip and I can show how my programmable controller can control the flow according to the user's input.

4.2 Design of Active Control for 3D-Printed Chip

The valve is widely used in continuous-flow Microfluidic chip to control fluid flow. Pneumatic valves, which are possible in a conventional factory-produced chip with flexible materials like PDMS, while 3D-printed objects are hard and not flexible, they are not achievable in 3D-printed chips. To realize an actively controllable system that allows me to perform a manipulation of continues flow of fluid on my 3D-printed chip, I need to design a substitution for the valve for my "hard chip".

Most conventionally fabricated microfluidic chip uses PDMS membrane to create a pneumatic valve. This type of valve within a chip can be controlled by adding or releasing pressure to the corresponding place in a second channel layer, normally called as control layer. Figure 35 depicts a pneumatic valve with a flow channel layer and a control channel layer.

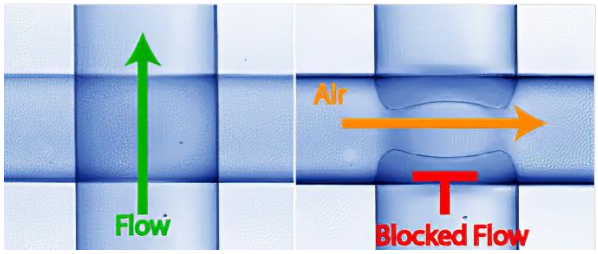


Figure 34: Pneumatic Valve [28].

While using a resin 3D-printed microfluidic chip is not flexible at all and moreover, printing multiple layers of channels is not achievable directly, I find another way to block flow as a substitution of this type of valve.

A basic idea to cut-off or to stop a flow in a channel is to add a counter-pressure against the flow direction. Figure 36 depicts the idea of two flows, one from the left side, one from the right side, with the same level of pressure, are pushing each other, which will stop both flows.

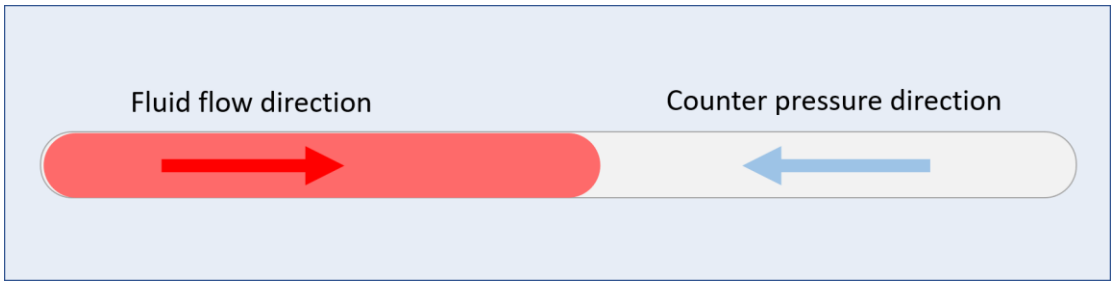


Figure 35: Proposed flow control mechanism.

And if I release the pressure from the right side a bit, I can observe a slow-down of the flow in comparison with the flow without the counter-pressure.

With this physical phenomenon, I realize a flow and control channel in a one-layer mechanism. I use one flow channel layer and add pressure adding points, i.e. I open air inlet ports at corresponding places, in which additional pressure is desired to slow-down, to stop or cut off a flow. Figure 37 shows how a counter-pressure can stop a fluid flow.

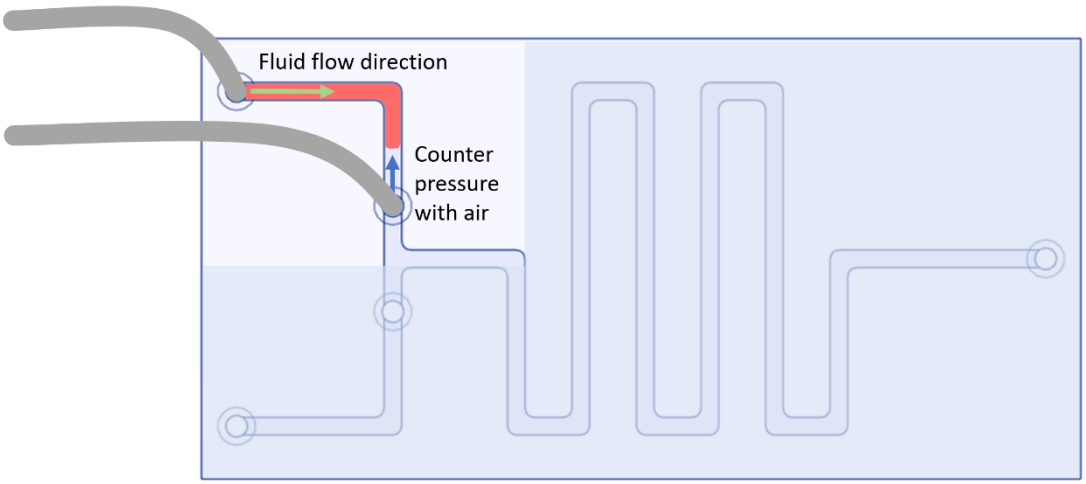


Figure 36: Illustration of flow control in a microfluidic chip.

By using this mechanism, I am able to design a 3D-printed microfluidic chip taht can have the controllable flow functionality.

4.3 Experimental Arrangement and Corresponding Chip Design

In the following, I discuss different possible experiments I have designed. These experiments and their corresponding chips are designed to demonstrate the possibility of being printed by the 3D-printer and the functionality of my prototype system.

The first experiment I designed is to use the active control mechanism I stated before to control the fluid flow. As Figure 38 depicts, I have designed three reservoirs in the shape of the letter “e”, “d” and “a” which is the abbreviation of our institute. There is one fluid inlet, which is connected to the three reservoirs through three separated channels. Each channel has a pressure-adding-point to allow adding counter-pressure to stop the fluid flows into the corresponding reservoir.

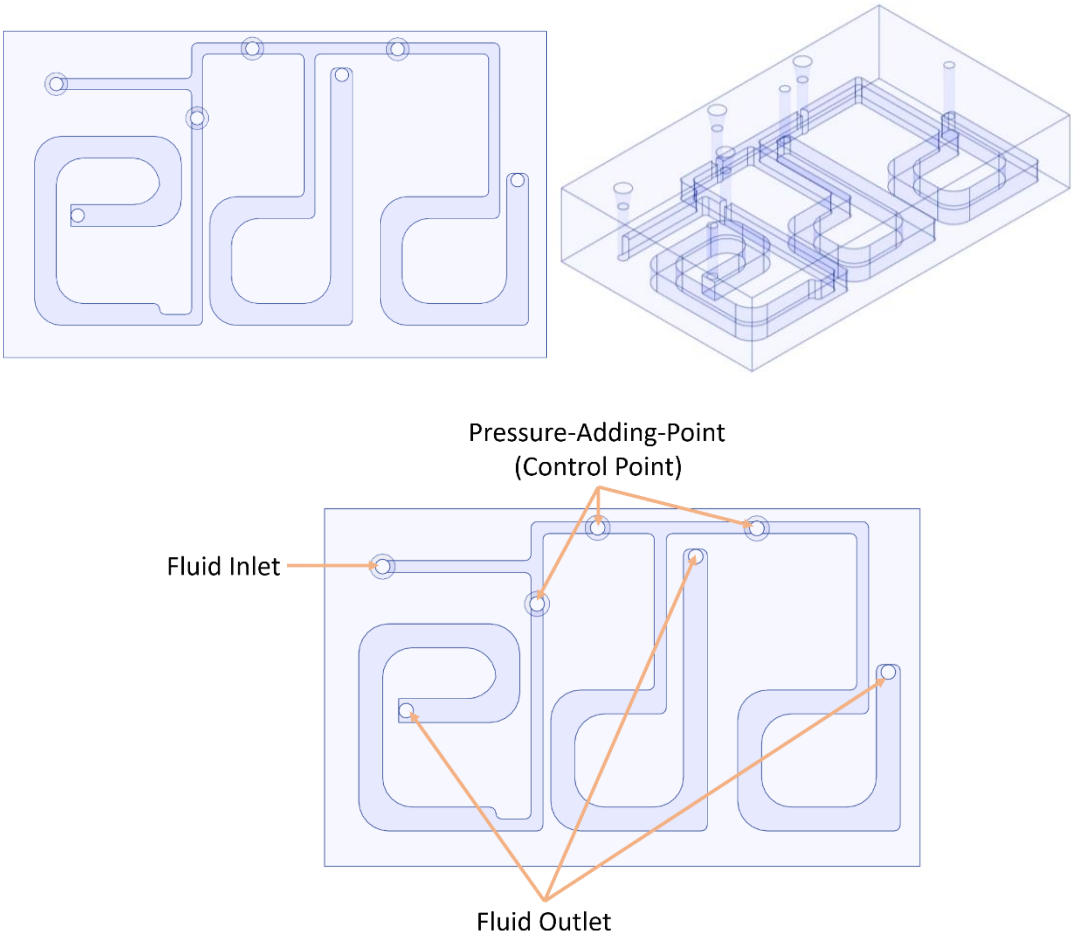


Figure 37: Chip design 1 – "e", "d" and "a" reservoir chip.

This chip has a dimension of 20 × 12 × 4mm. Each channel has a diameter of 400μm.

In this experiment, I want to infuse a colored distilled water from the inlet of the chip. By using my prototype system, I program a sequence of pump actions to control the fluid to flow into "e", "d" and "a" reservoir with a different amount.

The second experiment I designed is based on the classical serpentine mixing pattern. Here, I want to show the control of fluid by direct control of pumps. Furthermore, I want to show a fluid separation experiment on this chip, where the fluid is separated into short segments. This can demonstrate the possibility of using my prototype to realize the droplet experiment for, for instance, single-cell catching.

For this experiment, I have designed two chips with different channel density. Figure 39 depicts the first chip with a dimension of 10 × 20 × 4mm. Each channel has a diameter of 400µm. The distance between each vertical channel is 1067µm. The next chip is depicted in Figure 40 with a higher density of the serpentine channel, where the distance between each channel is 400µm. This chip's dimension is the same, which is 10 × 20 × 4mm with a channel diameter of 400µm.

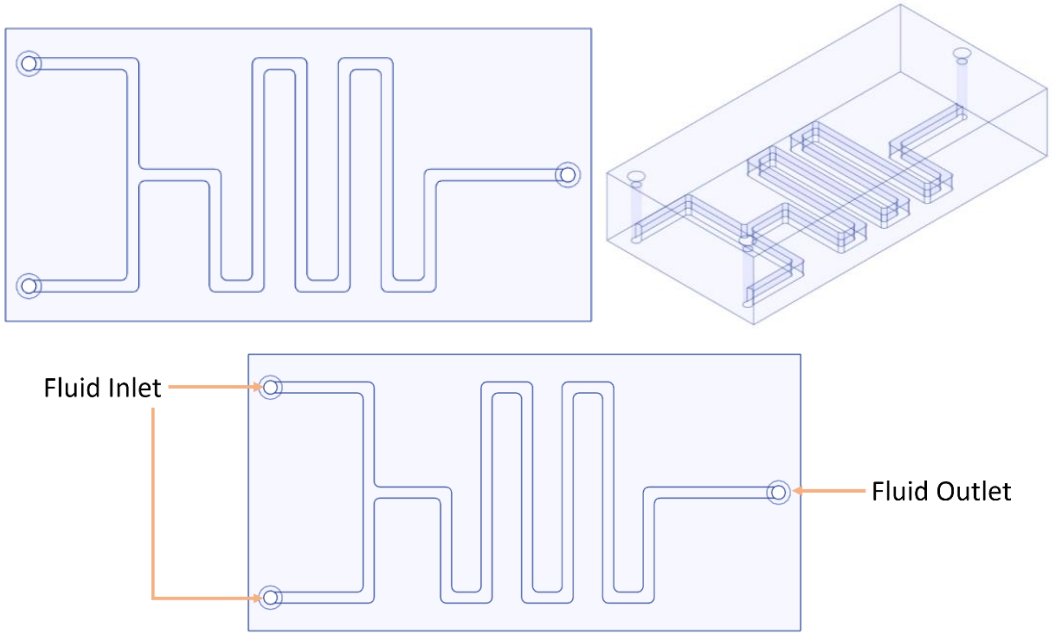


Figure 38: Chip design 2 – serpentine channel (low density) chip.

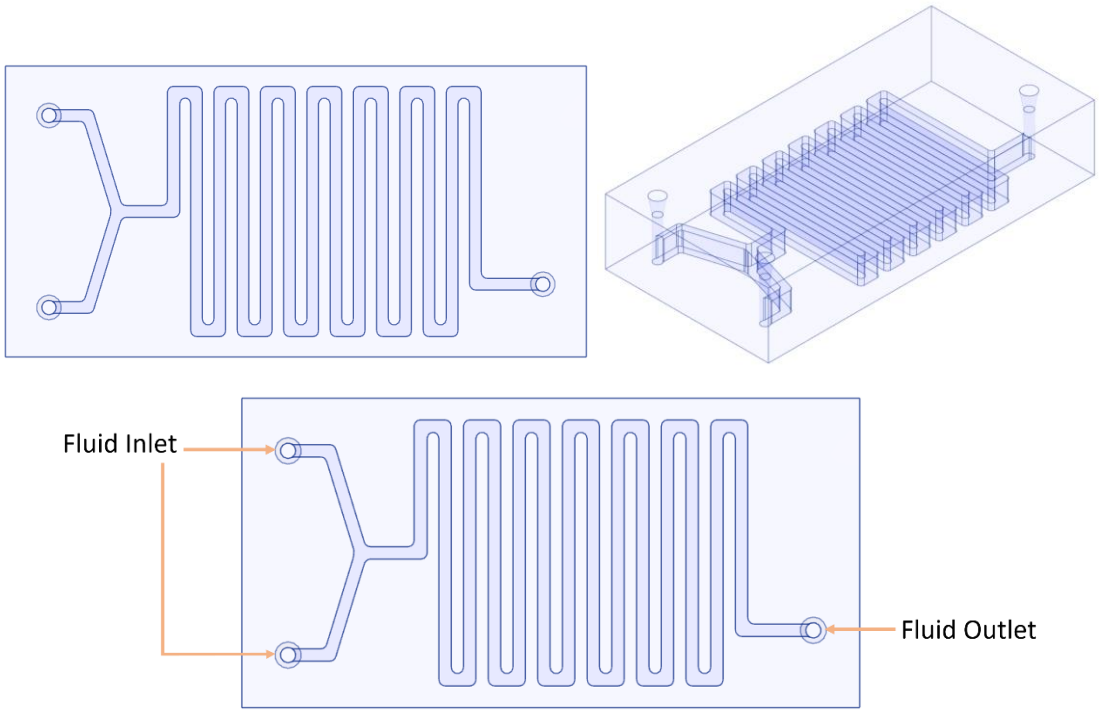


Figure 39: Chip design 3 – serpentine channel (high density) chip.

The third experiment I designed is to demonstrate the ability to print high accuracy module inside a chip. I designed a passive mixing device. In this chip, I have a chamber with a matrix of very small cylinders to allow flowing fluids being mixed when they leave the chamber. This design is inspired by the picoliter-volume mixer with intersecting channels of various lengths and a bimodal width distribution proposed by He et al. [29].

Figure 41 depicts the mixer device, which has a dimension of 10 × 20 × 4mm. Each channel has a diameter of 400µm. The mixing chamber has a dimension of 1500 × 3000 × 1200µm with each cylinder with a diameter of 250µm.

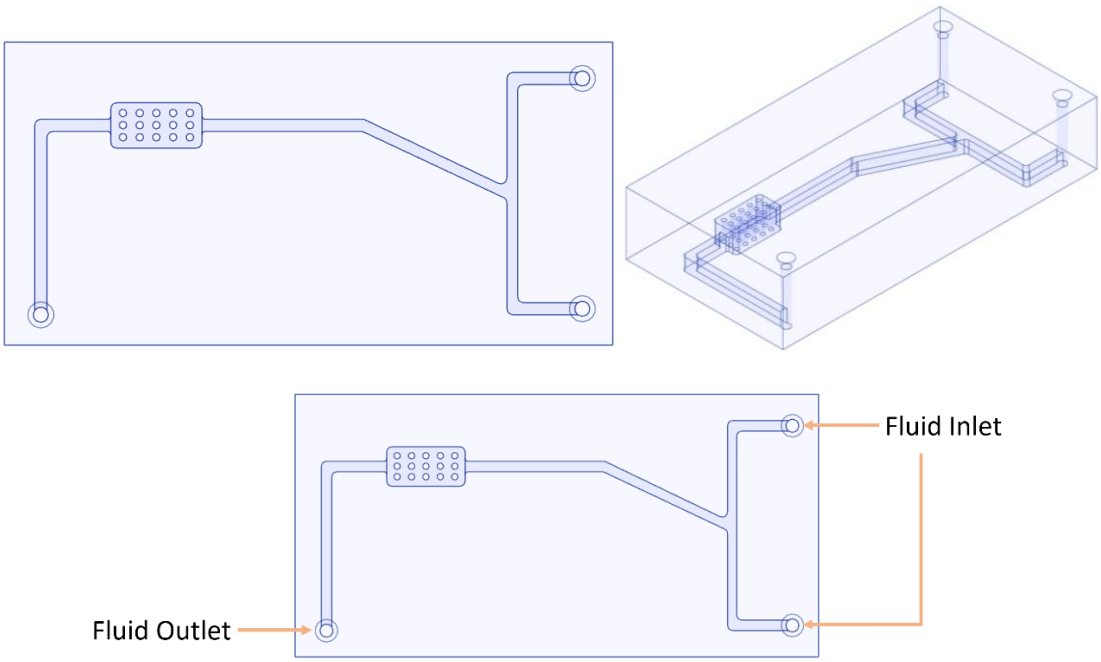


Figure 40: Chip design 4 – matrix cylinder chamber chip.

The last experiment I designed is to use my prototype system to produce small eatable pills. I choose the "spherification" experiment, which is a culinary process that employs sodium alginate and calcium lactate. With this experiment, I want to realize the microfluidic approach to generating capsules of biopolymer hydrogels. Zhang et al. [30] and Tumarkin et al. [31] have reported this type of approach, which inspired me to use a 3D-printed microfluidic chip and my prototype to try this experiment. I designed a chip with a different type of inlet and outlet. Figure 42 depicts the chip I have designed for this experiment. This chip has three inlets, where in the middle a colored sodium alginate solution will be infused. The other two inlets, both located at the left- and right-hand side of the middle inlet, are used to infuse the calcium lactate solution. I use color in the sodium alginate solution to indicate the final product. While the sodium alginate solution encounters the calcium lactate solution, a chemical reaction happens [32]. This will slowly cure the fluid sodium alginate solution and the outside of the sodium alginate solution becomes a coat. This reaction will result in small roe like balls. For this experiment, my designed chip has a dimension of 20 × 12 × 2mm. The inlets are designed as a flat insertion. The three channels have a diameter of 550µm, and the reaction chamber which is also the

outlet has a diameter of 1000µm, and a height of 750µm.

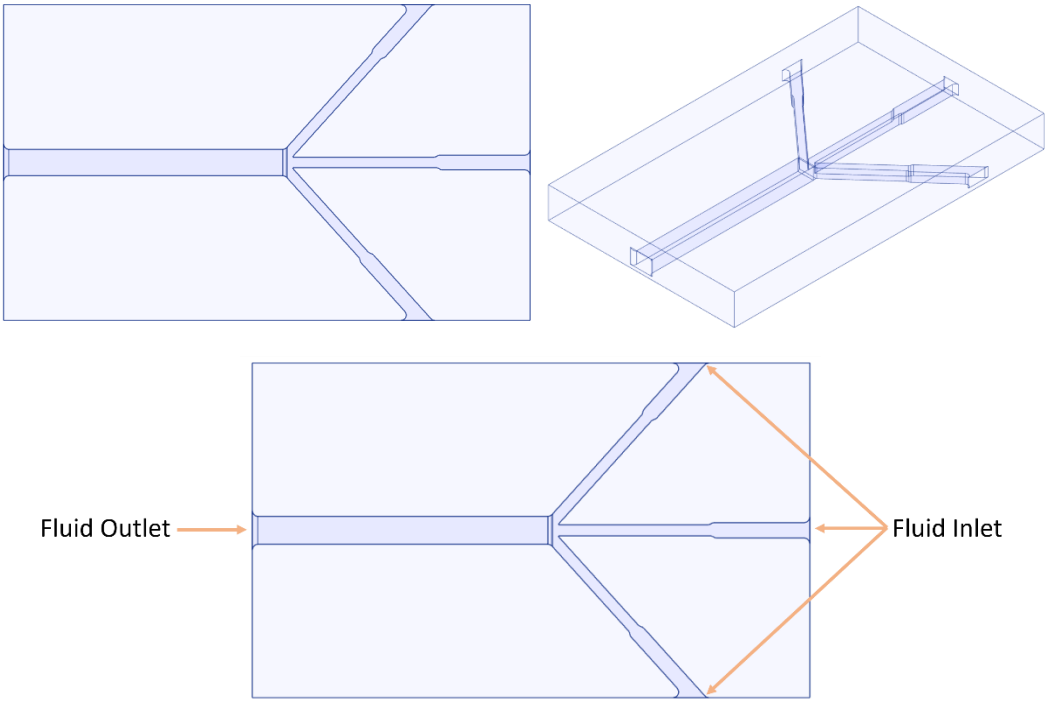


Figure 41: Chip design 5 – "spherification" chip.

5 Result

In this chapter, I discuss the result of my prototype of a portable microfluidic system, which I have described in the previous chapters. In the first part, I talk about the result of 3D-printed chips that I have designed and talked in the previous chapter. Besides, I discuss the quality of printing microfluidic chip with a 3D-printer. Next, I show the hardware and software part of my prototype system. I briefly show how the hardware is connected and setup, and how the control function is realized and how the implementation of the software I designed is structured. Finally, I show how the entire prototype works and how the experiments I mentioned in the previous chapter work out on with my prototype system and the result of each experiment.

5.1 3D-Printed Microfluidic Chips

In this part, I show the printed result of previously designed chips. I also discuss findings commonly occurs on each printed microfluidic chip.

To print the Microfluidic chips, I have sliced every 3D chip design into layers of 50µm thickness by using the accompany software from MiiCraft®, namely the MiiUtility, and setup the MiiCraft® + DLP Laser 3D printer to print every layer with a exposure time of 7 seconds except the first two layers, which will be exposed for a longer time, namely 60 seconds, to ensure the object will not fall down from the fabrication platform (catcher). The printing speed is set to "Slow" for the best accuracy.

The first chip is the "e, d and a reservoirs" chip. This chip is designed to have no base, which means all empty spaces, including the channel, will be unexposed. After the chip is printed, I washed the chip, especially the groove, with isopropanol alcohol. Finally, I sealed the bottom with the clear tape and put the chip into the UV chamber for a post-curing

process of about 2 hours. Afterwards, I injected isopropanol from each opening port (e.g. inlet or control point) to wash out the glue of the tape remained inside. The Figure below shows the printed chip from different perspectives.

Figure 42: Printing result of chip design 1 (Top, Down and 45 ° View Angle).

The next Figures show two chips which are the serpentine mixing device. The first chip was printed directly with a base thickness of 200µm, i.e. 4 layers of cured resin. After it is printed, I washed the chip with isopropanol alcohol and injected isopropanol from each opening port (e.g. inlet or control point) to wash out the remaining resin fluid. Afterwards, I put the chip into the UV chamber for a post-curing process for about 2 hours. The second chip with a higher density of channels was printed without the base. After the chip is printed, I washed the chip, especially the groove, with isopropanol alcohol. Finally, I sealed the bottom with the clear tape and put the chip into the UV chamber for a post-curing process of about 2 hours. Afterwards, I injected isopropanol from each opening port (e.g. inlet or control point) to wash out the glue of the tape remained inside.

Figure 43: Printing result of chip design 2 (Top, Down and 45 ° View Angle).

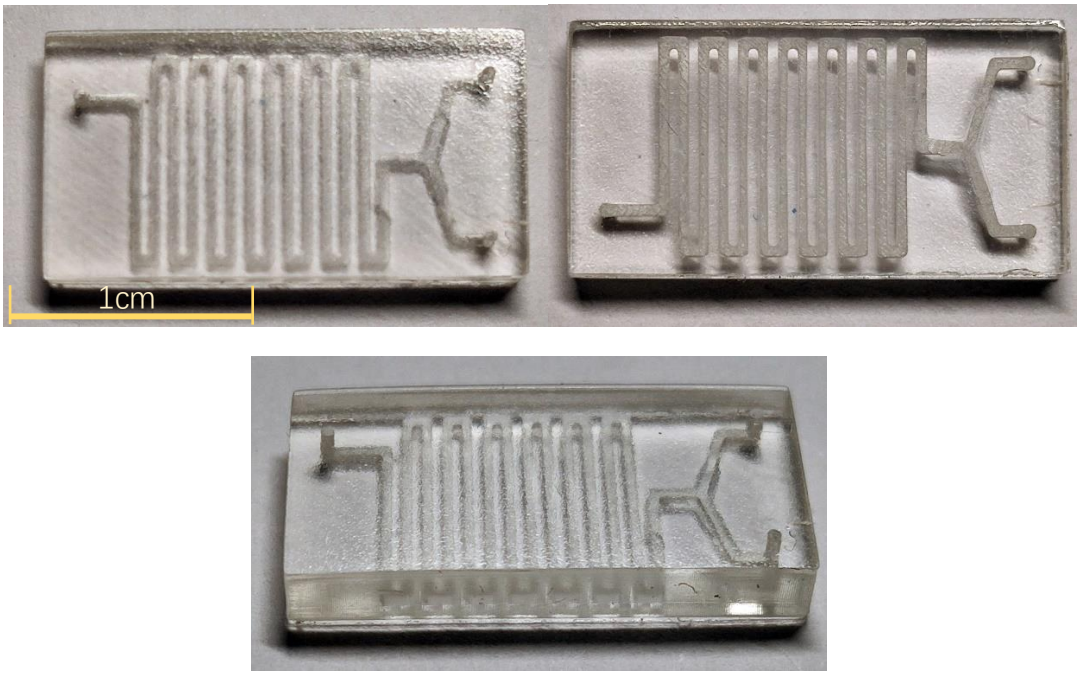


Figure 44: Printing result of chip design 3 (Top, Down and 45 ° View Angle).

I continue with the mixing device. The next Figure shows the mixing device I designed with the matrix cylinder chamber. Here, I can clearly see the small-sized cylinders are well printed and is noticeably well arranged. This chip was printed directly with a base thickness of 200µm, i.e. 4 layers of cured resin. After it is printed, I washed the chip with isopropanol alcohol and injected isopropanol from each opening port (e.g. inlet or control point) to

wash out the remaining resin fluid. Afterwards, I put the chip into the UV chamber for a post-curing process of about 2 hours.

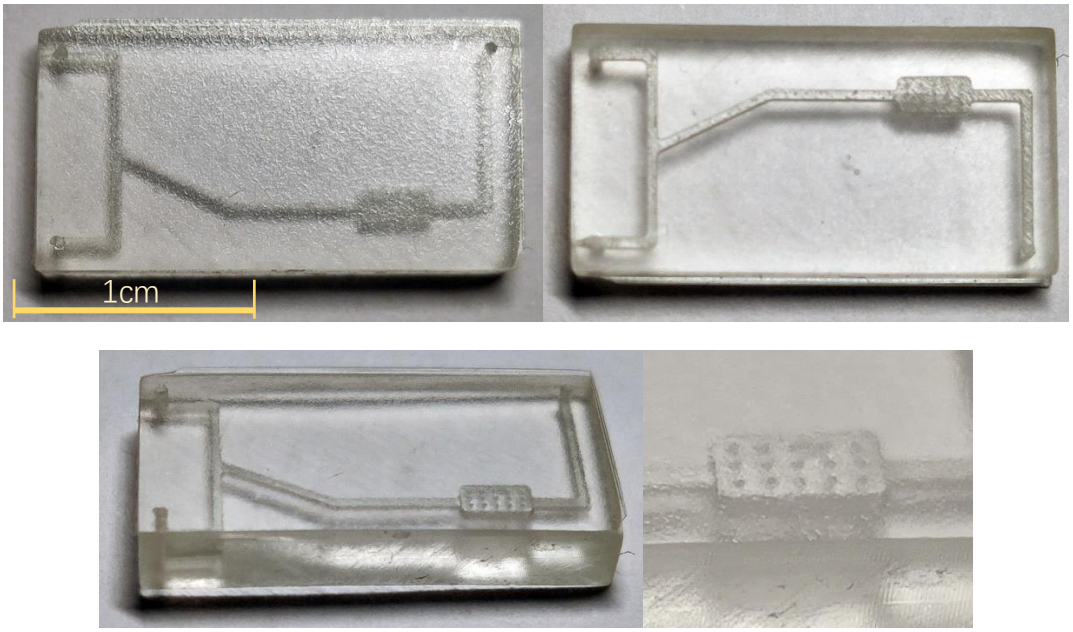


Figure 45: Printing result of chip design 4 (Top, Down, 45 ° View Angle and Closeup).

The last chip I have designed is for the "spherification" experiment. This chip was printed without the base. After the chip is printed, I washed the chip, especially the groove, with isopropanol alcohol. Finally, I sealed the bottom with the clear tape and put the chip into the UV chamber for a post-curing process about 2 hours. Afterwards, I injected isopropanol from each opening port (e.g. inlet or control point) to wash out the glue of the tape remained inside. Figure 47 shows the result of this printing.

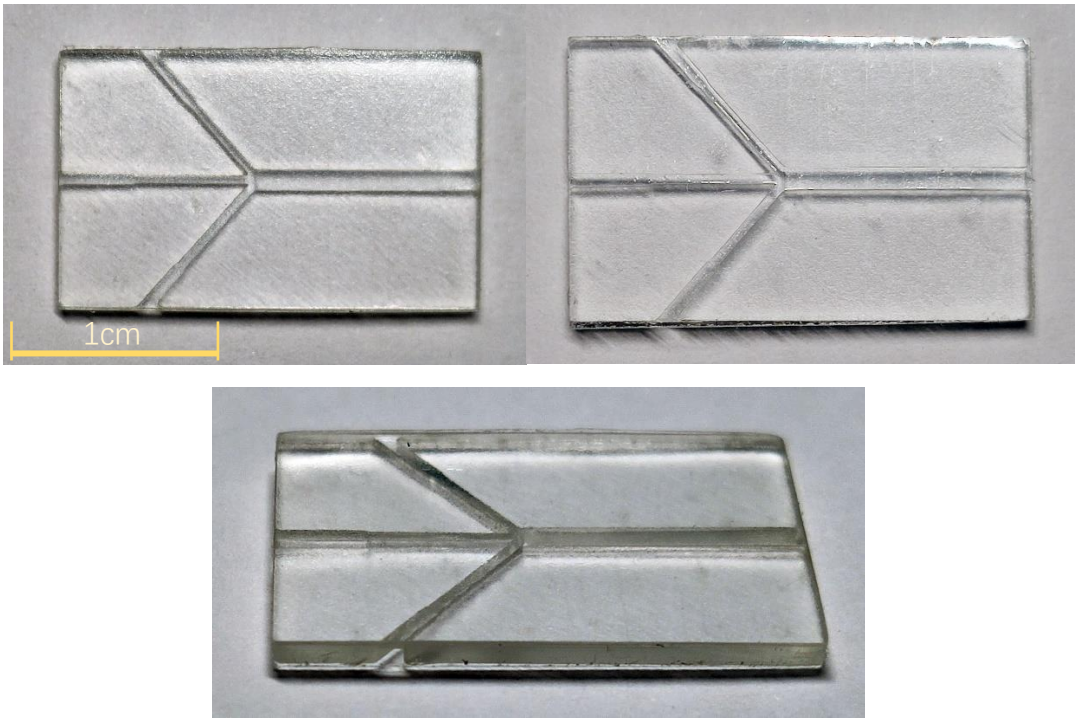


Figure 46: Printing result of chip design 5 (Top, Down and 45 ° View Angle).

When I take a closer look at these chips, I can observe some problematic facts. The first and most obvious fact I have noticed is the uneven surface in the channel. Small protrusions are noticeable on the walls on the channel. This problem could due to the fact that the 3D-printer's accuracy of printing and also caused by an unclean post-printing process of a chip, i.e. washing channel with isopropanol alcohol, where the alcohol was not 100% pushed out of the channels. The remaining small pieces of alcohol resin solution can be the cause of these small protrusions, which are alcohol resin solution cured during the post-curing process.

Another fact that I have noticed is that almost every chip is not 100% a cuboid, while they are designed as cuboid form, but the printed results are most likely deformed. On one hand, the printer's accuracy is not that great, since the MiiCraft® + printer I am using is a mid-range consumer-grade printer. On the other hand, the characteristic of the resin may play a big role, as well. The temperature and humidity can largely change the printing quality and results in deformation during the printing and curing process.

But all in all, the designed microfluidic chips are well printed, in the sense of usable to demonstrate my prototype system. I will show my prototype system using these chips in a later section. Finally, it matches my definition of portability, i.e. to build and produce a microfluidic chip in a short range of time and with accessible materials and devices, besides the problems I stated above.

5.2 Hardware and Software

I realized my hardware setting according to the definition I have given in the previous chapter. The form factor of my prototype system, including the programmable controller and the pumps, is relatively small compared to traditional stationary laboratory setup. Figure 48 shows a photo of everything connected. The size of all components together is about a DIN-A4 paper, when I flatten them on the desk. Thus, it complies with my definition of portability. It is possible to further compact the size of the prototype, by bringing all components into a suitable case, which unify them into one device.

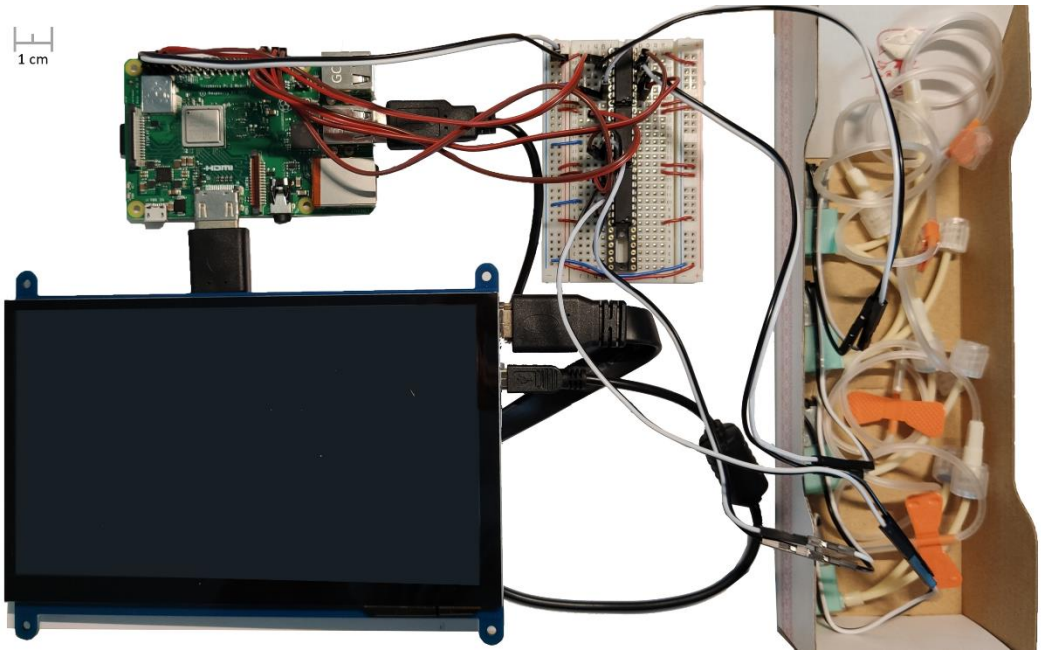


Figure 47: Prototype hardware setup.

On this prototype, I am using the "Raspbian with Desktop" (version April 2019) as the

operating system. The control software is implemented in Java. Java has the advantage of portability, i.e. when I later upgrade my single-board-computer into another more powerful one, I can easily port this software into a new hardware environment.

While I use Java as my implementation language, I need a library to access the GPIO of the Raspberry Pi 3 Model B+. The library I decided to use is the "Pi4J", version 1.2. In fact, the "Pi4J" library is based on the "Wiring Pi", which is a part of the "Raspbian" operating system. In the following, I discuss in which way my software is controlling the output signal of the single-board-computer, which then will affect the speed of the pump.

The pump I am using for this prototype, the Aquatech Miniature Peristaltic Pump RP-Q1.2N-P20A-DC3V, has a rated voltage from 3V ~ 5V. To control the speed of the pump, I need a way to regulate the power I supplying to the pump. A common way to control the speed of a pump, or in fact, to control the speed of that motor, is to regulate the supplied voltage. This would not work in my prototype since the hardware I am using does not support the so-called "DC Regulation". And further, the rated voltage of the pump is down to 3V, while I can use a lower voltage to drive the pump, but the speed of the pump becomes unstable. For this reason, I use another commonly used technique to control the speed of the pump, namely the Pulse-width modulation, or for short PWM. PWM is a method to reduce the average power delivered by an electrical signal, by effectively chopping it up into discrete parts. The average value of voltage (and current) fed to the load is controlled by turning the switch between supply and load on and off at a defined rate.

By using PWM, I can adjust the rate of signal-on and -off to slow down or speed up the spinning speed of the motor in the pump, while I am still supplying a voltage of 5V, as I have designed before. This can ensure the stableness of the motor and ensure the high pumping force the pump.

But unfortunately, Raspberry Pi 3 Model B+ has only two PWM connections: (GPIO18, GPIO12) and (GPIO13, GPIO19), while I need at least four connections for my four pumps. To solve this problem, I use the software PWM mode provided by the "Wiring Pi", which is also available in the "Pi4J" library. With software PWM mode, I can bring any GPIO pin into a software simulated PWM pin. In the software, I define a fixed frequency, which is the frequency of the 'on' time within a given time, to 100Hz, and adjust the speed by regulating the of the duty cycle, which is the proportion of "on" time to the regular interval.

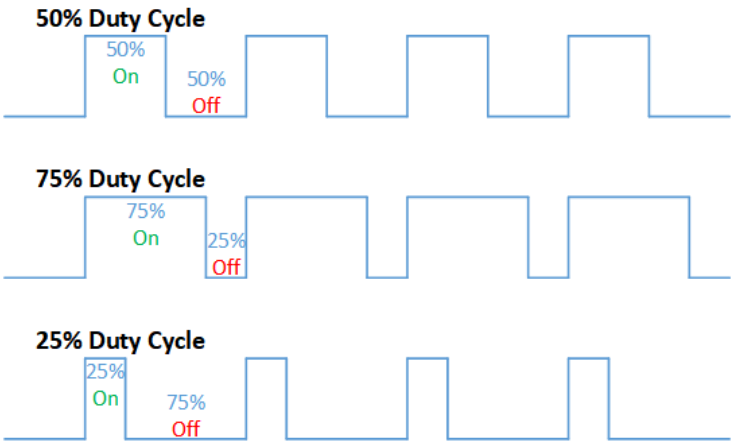


Figure 48: Illustration of signal of PWM [33].

To allow my team to implement the software, the program is divided into multiple classes: Action, Configuration, MainFrame, Pump, PumpType and Util. Each class represents a type of object.

First, the class "MainFrame" implements all the Graphical User Interface (GUI) windows. The windows are these I have designed and talked about in the previous chapter. Figure 50 shows a screenshot of the pump configuration window of the final implementation.

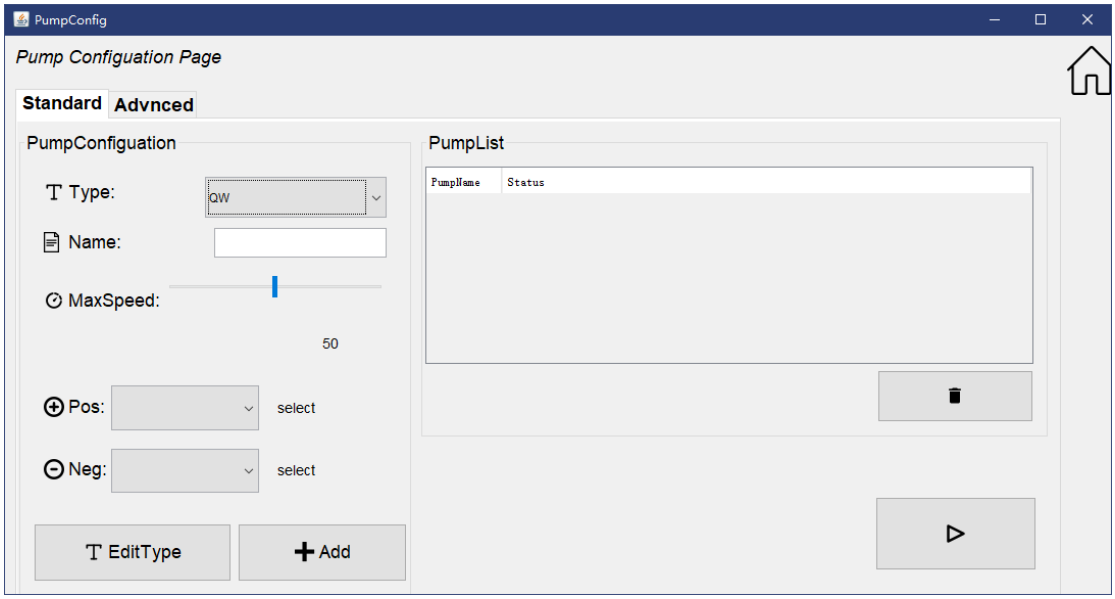


Figure 49: Screenshot of the implemented control program.

Further, other interactions and objects are abstracted into different classes. The class "Pump" represents a configuration of a real-world pump, which defines the different attributes of a pump, such as its type, allowed maximum speed, pin headers, and name. The class "PumpType" represents a type of pump, while different pump has different characteristics, especially an allowed minimum speed (while a too low speed/short PWM duty cycle will result in a constant short circuit because the motor will not spin in that case). The characteristics will be represented by the class's attributes. The class "Action" represents a pump-action, in other words, it describes what a pump will do from when until when, and these properties will be saved as attributes, while each action has one ability, namely to perform the action. Finally, we have the "Configuration" class, which represents the whole inputs of a user, including the defined pumps, programmed action sequence, etc. This will then be used with the class "Util" to provide the user with the function of saving and loading. This allows the user to reuse the input and save the time of re-entering the information.

5.3 Results of Test Experiments

As the final step of my work, I use the 3D-printed chips to perform the designed experiments with my prototype. I want to demonstrate the feasibility and usability of the prototype through these experiments. As Figure 51 shows, all my experiments will be performed with the shown condition. In each experiment, I stick the 3D-printed chip on the desk by using a piece of sticky clay. Since I am demonstrating the prototype, I use distilled water as my fluid sample. I use liquid food dye, red and green, to color the distilled water (except the "spherification" experiment). Instead of using a reservoir to store the fluid samples, I let the sample be stored in the tube, which is connected to the needle. In all the following experiments, pump actions are set to push the fluid sample or the air.

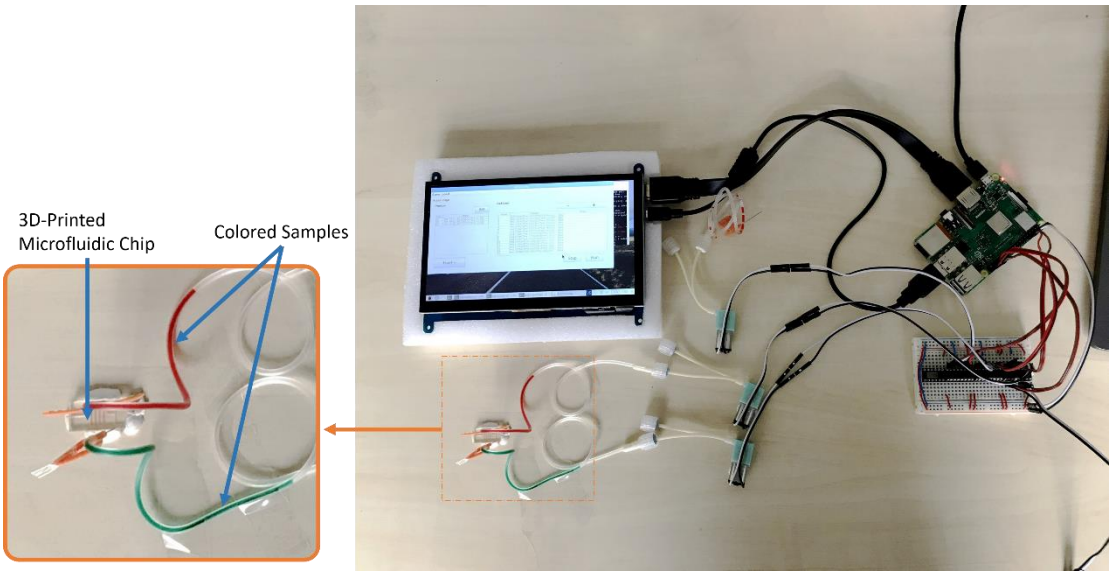


Figure 50: Experiment Setup.

5.3.1 Test Experiment 1: Active Control of Fluid Direction

In this experiment, I want to control fluid flow in the chip. I use the "e, d and a reservoirs" chip to demonstrate the active controlling function of the prototype. As I have described before, the chip has only one fluid inlet port and three controlling ports. I connect all four ports to the controller's pumps. Figure 52 shows the naming of the connected pumps I

have defined in the program.

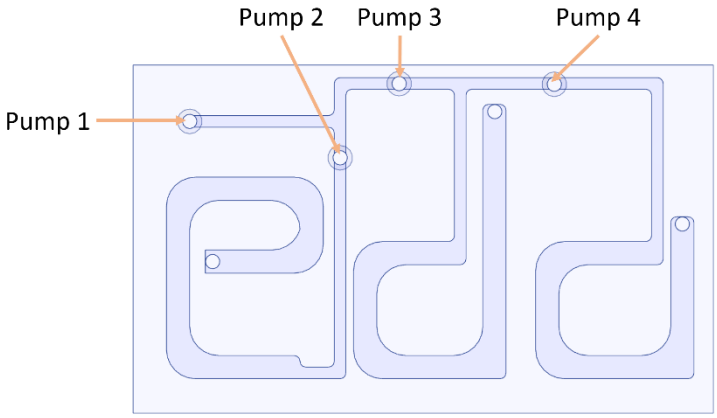


Figure 51: Connection description of experiment 1.

To demonstrate the active controlling mechanism, I designed the following experiment. I let the fluid flow into the chip constantly through the inlet port (Pump 1). By pumping counter-air-pressure on different controlling ports (Pump 2 – Pump 4), I can direct the fluid into different reservoirs with different volumes of fluid. I want the fluid flows to the "e" reservoir with the highest volume, in the "d" reservoir with a little less volume of fluid and in the "a" reservoir with the least volume of fluid. To realize this experiment, I have programmed the flowing program sequence, as shown in Table 4.

Pump Name	Time (From – Until) in min:sec:ms	Speed (in %)
Pump 1	00:00:000 – 00:30:000	17
Pump 3	00:00:000 – 00:10:000	35
Pump 4	00:00:000 – 00:15:000	35
Pump 2	00:20:000 – 00:30:000	35
Pump 3	00:17:000 – 00:30:000	35
Pump 4	00:19:000 – 00:30:000	35

Table 4: Program Sequence of Experiment 1.

The following Figure shows photos of the experiment at different time points in chronological order.

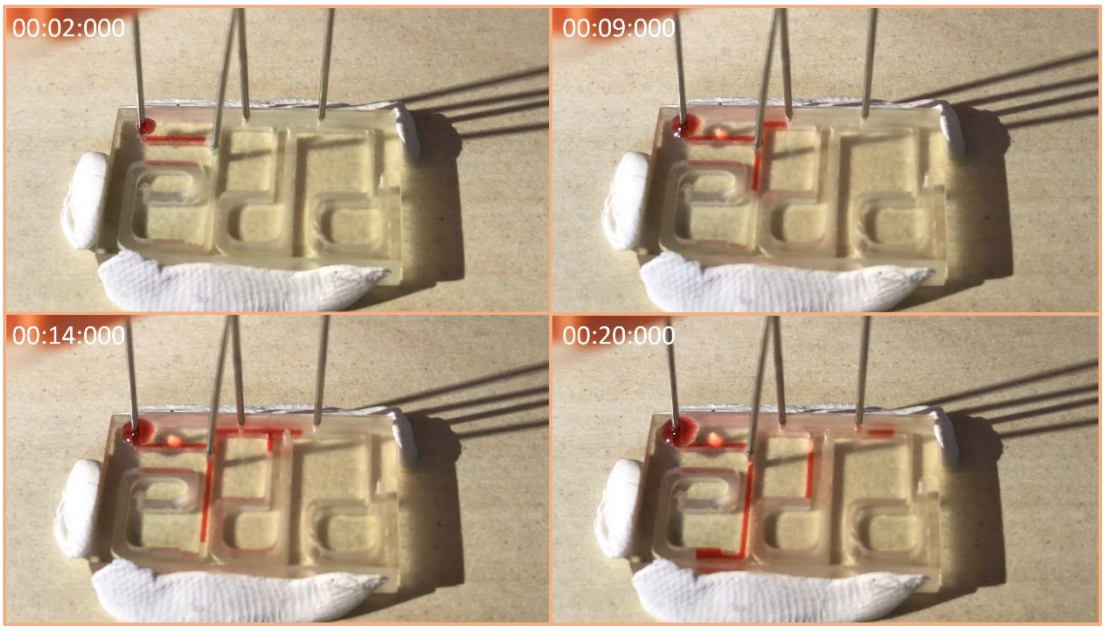


Figure 52: Flow situation of experiment 1 at different stages.

As the result of this experiment, I correctly controlled the fluid flow as I have designed. This demonstrated the active controlling of the prototype system is achievable.

5.3.2 Test Experiment 2: Mixing

In this experiment, I test the two serpentine mixing chips and the matrix cylinder chamber mixing chip with my prototype system. I let two different colored fluid samples, red and green, flow into the two inlet ports separately. Figure 54 shows the naming of the connected pumps I have defined in the program. I use the same configuration for both the serpentine mixing chips and the matrix cylinder mixing chip.

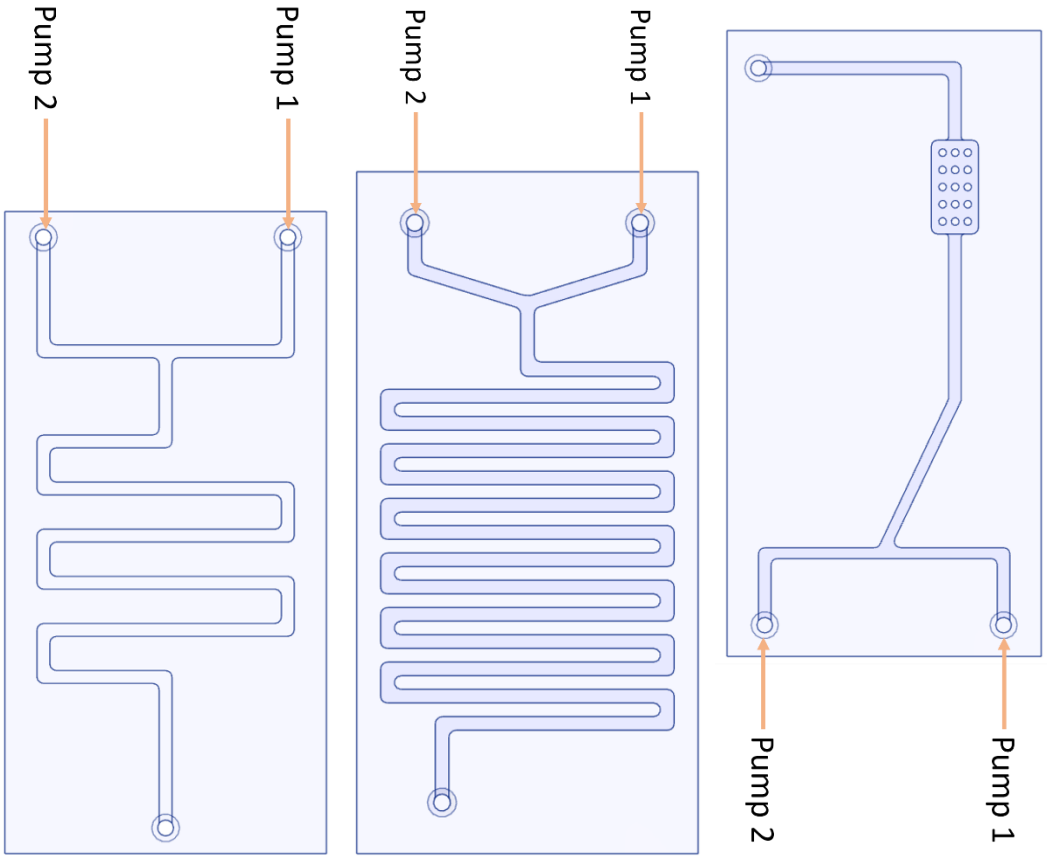


Figure 53: Connection description of experiment 2.

In this experiment, I continuously supply the two inlet channels with the colored fluid samples, to allow them being mixed in the serpentine formed channel and in the matrix cylinder chamber. Thus, the program sequence is as simple as follows, showed in Table 5. This sequence is used for both the serpentine mixing chips and the matrix cylinder chamber mixing chip.

Pump Name	Time (From – Until) in min:sec:ms	Speed (in %)
Pump 1	00:00:000 – 00:30:000	17
Pump 2	00:00:000 – 00:30:000	17

Table 5: Program Sequence of Experiment 2.

Here, the fluid samples will be mixed while they are flowing in the serpentine formed channel [34] and in the matrix cylinder chamber, respectively. The next Figure shows the

mixing process of the two fluids in the two serpentine mixing chips and in the matrix cylinder chamber chip. The piece of stuff at the right side is used to suck the effluence.

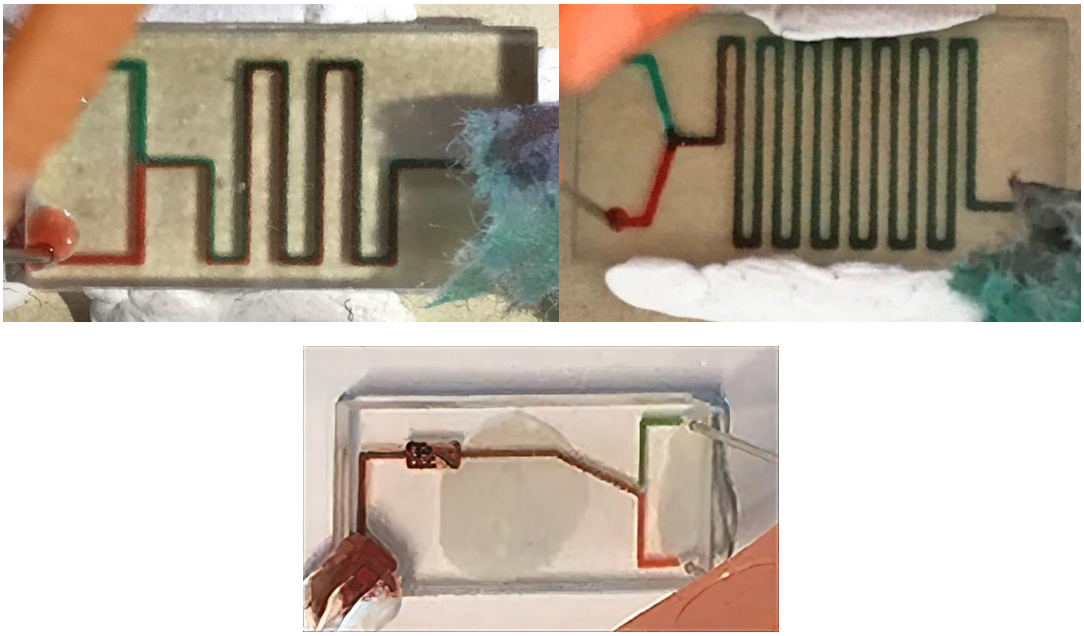


Figure 54: Mixing process in two different serpentine mixing chips, with different channel length and density, and in a matrix cylinder chamber mixing chip.

5.3.3 Test Experiment 3: Droplet

In this experiment, I use my prototype system to generate droplets by separating the fluid with air. Here, I again use the serpentine chip, since it has two inlets. I use one to supply the fluid, and the other one to supply air. By alternating the supply of fluid and air, I can split the fluid and generate drops. Figure 56 shows the naming of the connected pumps I have defined in the program.

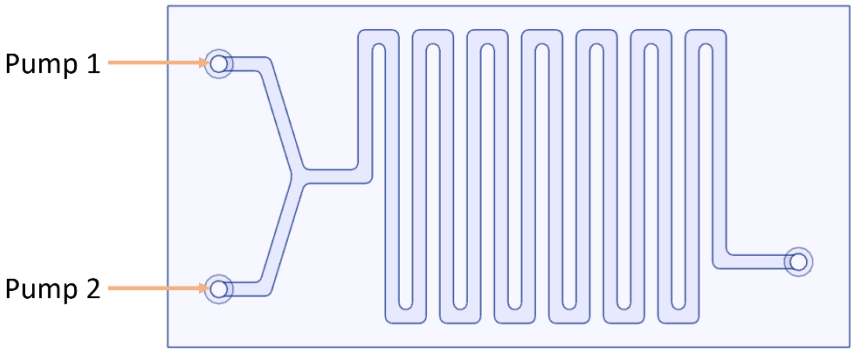


Figure 55: Connection description of experiment 3.

By controlling the time of pump activity, I am able to generate different drop sizes. The next table (Table 6) indicates a program sequence of generating one smaller drop.

Pump Name	Time (From – Until) in min:sec:ms	Speed (in %)
Pump 1	00:00:000 – 00:00:500	17
Pump 2	00:00:000 – 00:30:000	35

Table 6: Program Sequence of Experiment 3 to generate one drop.

The generation of the drop running in the chip's channel looks as follows, showed in Figure 57.

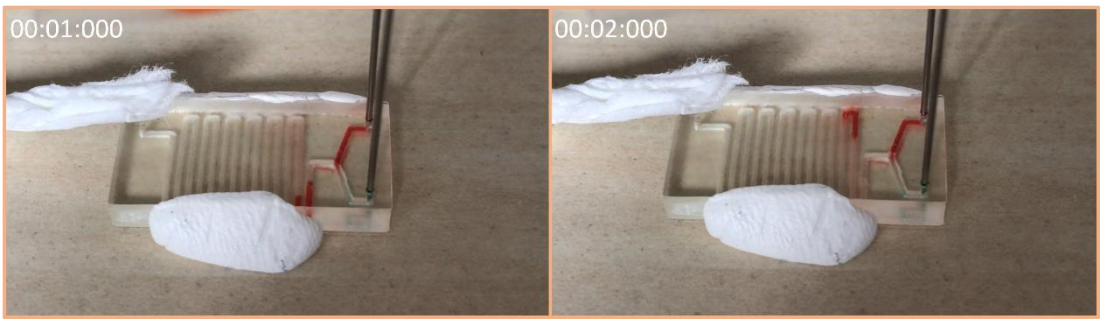


Figure 56: Single drop generation.

The next program sequence, indicated in Table 7, allows me to continually generate multiple bigger drops (compared to the previous one).

Pump Name	Time (From – Until) in min:sec:ms	Speed (in %)
Pump 1	00:00:000 – 00:01:000	17
Pump 2	00:01:000 – 00:03:000	35
Pump 1	00:03:000 – 00:04:000	17
Pump 2	00:04:000 – 00:05:000	35
Pump 1	00:05:000 – 00:06:000	17
Pump 2	00:06:000 – 00:30:000	35

Table 7: Program Sequence of Experiment 3 to generate multiple drops.

The generation of the drops running in the chip's channel looks as follows, showed in Figure 58.

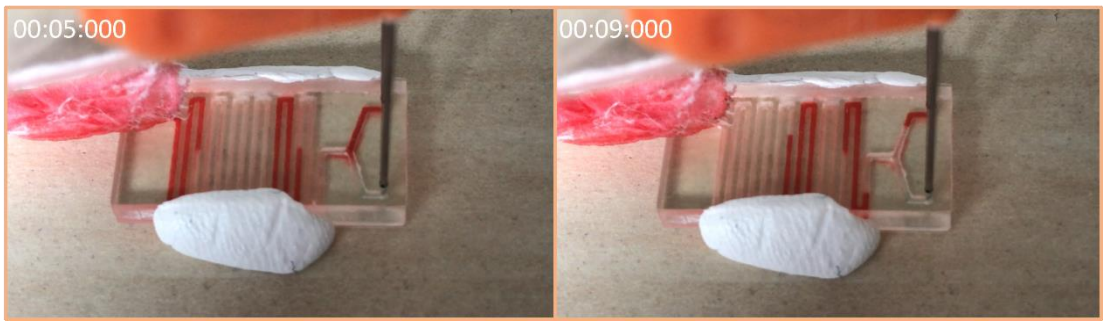


Figure 57: Multiple drops generation.

5.3.4 Test Experiment 4: Spherification

In this experiment, I perform the "spherification" experiment. The 3D-printed chip has three inlet ports. I connect all three ports to the controller's pumps. Figure 59 shows the naming of the connected pumps I have defined in the program.

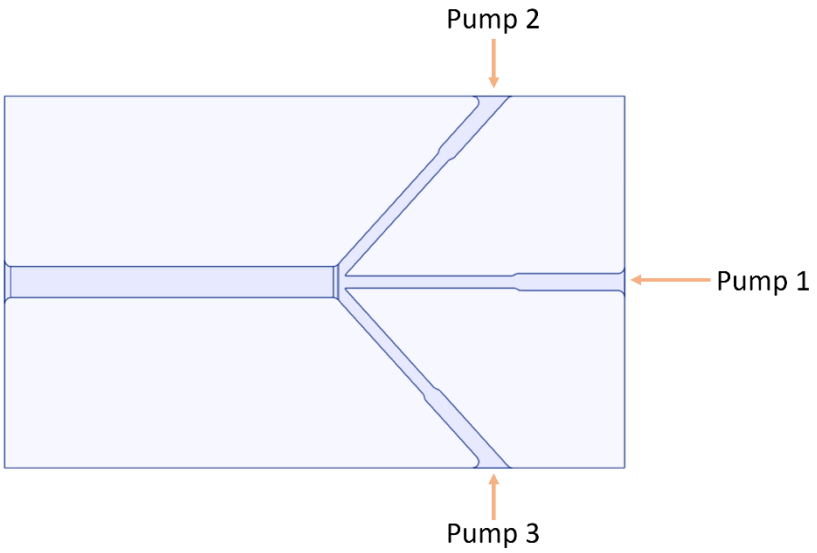


Figure 58: Connection description of experiment 4.

To perform this experiment, the one in the middle is supplied with the sodium alginate solution. The side inlet ports supply the calcium lactate solution. The experiment is designed as follows. The supply of the sodium alginate solution and calcium lactate solution alternate. While a small volume of sodium alginate solution is pushed into the reaction channel, the calcium lactate solution will be pushed out and cut off the sodium alginate solution to let it form into a drop and cover it. The calcium lactate cover will react with the sodium alginate drop and produces a small hydrogel ball. This process will be repeated to form small pills. The following program sequence, shown in Table 8, is used for this experiment.

Pump Name	Time (From – Until) in min:sec:ms	Speed (in %)
Pump 1	00:00:000 – 00:00:500	17
Pump 2	00:00:500 – 00:01:000	35
Pump 3	00:00:500 – 00:01:000	35
Pump 1	00:01:000 – 00:01:500	17
Pump 2	00:01:500 – 00:02:000	35
Pump 3	00:01:500 – 00:02:000	35
…	…	…
Pump 3	00:10:500 – 00:11:000	35

Table 8: Program Sequence of Experiment 4.

Unfortunately, the result of this experiment is not as I have expected. Figure 60 shows a photo of the result. As I can see, the product was not multiple hydrogel balls, but a long string of reaction product – hydrogel string.

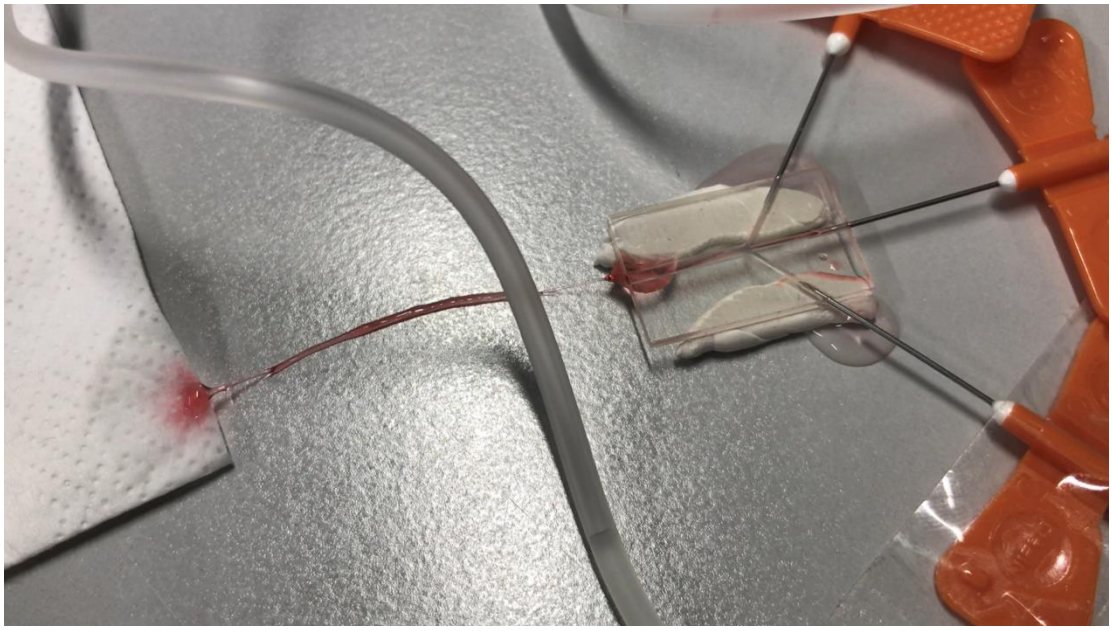


Figure 59: Result of experiment 4 – a string of hydrogel.

This unexpected result might have many causes. But, since I am designing the prototype of a portable microfluidic system and using this experiment to verify the feasibility of the system, I will not address the causes in detail.

5.4 Discussion and Conclusion

While performing the designed experiments, I found several problems. Besides the unsuccess of experiment 4, there are some common problems occurring in every experiment. The biggest problem I had is the stableness of the pump. As I have stated before, the pumps I am using is the Aquatech Miniature Peristaltic Pump RP-Q1.2N-P20A-DC3V. This peristaltic pump is, in fact, a ring-type peristaltic pump. Figure 61 depicts the difference between a standard peristaltic pump and a ring-type peristaltic pump.

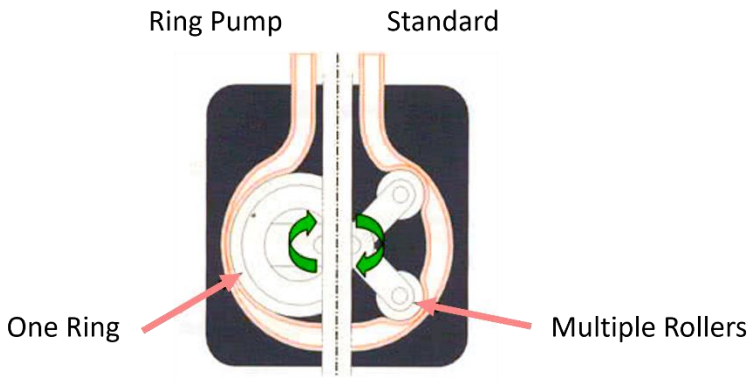


Figure 60: Difference between a ring-type and standard peristaltic pump [35].

As an advantage of the ring-type peristaltic pump, its ring rotor presses more gently on the tube over the smaller and sharper rollers, the stress of the tube is reduced, and longer life of the tube is achieved. But there is also a big disadvantage. While the ring rotor moves to the top, as shown in c) of Figure 62, there is nearly no pressure directly to the tube. At this time point, the fluid inside the tube will not move until the ring rotor moves further to a point where it can press the tube. Thus, instead of continuously moving fluid, I can see a haltingly moving phenomenon of the fluid when the pump is running at full speed.

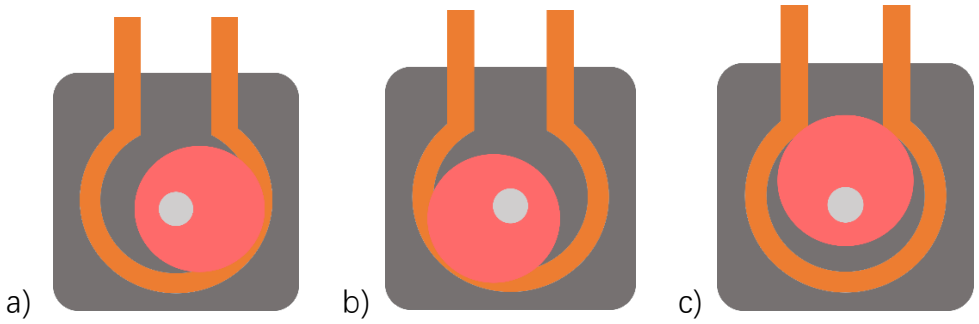


Figure 61: Rotation of the ring rotor in a ring-type peristaltic pump.

Since the pump has a rated discharge rate of 0.2ml/min, which is too fast for my microfluidic application, I have to downscale the discharge rate by down speed the pump using PWM. This results in an effect of the uncertainty of fluid movement. While the rotor moves slower, the time of fluid is moving becomes longer, but the time of fluid is not moving becomes longer, as well.

In many of my experiments, I have to retry the experiment multiple times to get the expected result. Especially, when multiple pumps are involved in an experiment, while I never know where the ring rotor currently is, I have to try until every pump is in a state where the rotor is pressing the tube and the fluid or air is coming out.

Another problem I have observed is the leakage of the fluid sample at the inlet connection ports, which can be seen in the figures shown in each experiment section. Because the design of my connection ports is not mature, by simply using a needle to connect the tube to the chip, such kind of leakage was foreseeable.

But, indeed, these problems are not affecting the prove of the feasibility and usability of the prototype system. To the first problem, as I have stated in the very beginning, the pump can be changed depending on the usage of the user. By switching the pump with another more precise and lower discharge rate pump, I can achieve a better result. In fact, the programmable controller is being proved to be able to control all of the four pumps, and

by changing the pump model would not change the control logic. And the same holds for a redesign of the connection ports of each microfluidic chip.

In fact, I defined the portability as transportable, usable on-the-go, lightweight and easy to access. The result, indeed, fulfills my definition of portability. On the one hand, I used a consumer-grade 3D-printer to fabricate the microfluidic chips I have designed. It is an easy and convenient process in comparison to the conventional fabrication process. Although the 3D-printed chips have some problems, such as deformation, in general, these chips are in a usable state and are used to perform experiments for the feasibility demonstration. On the other hand, my prototype system is, in comparison to a regular in-lab microfluidic system, relatively small and lightweight. The size of the prototype system is less than 20 × 20 × 20cm and weighs less than 600g in total. By switching the power supply of the prototype to a mobile power unite such as a power bank, I can truly use this prototype system on-the-go. Finally, by performing all the experiments I have mentioned above, I can verify the usability and feasibility of my prototype system.

6 Summary and Prospect

In this thesis, I have first discussed the state-of-the-art microfluidic system and the fabrication methods of the microfluidic chip. Then, I have realized a prototype of a portable microfluidic system. I defined the word portable as small-size, light-weight, usable on-the-go and easy to access. The prototype system consists of two different parts. The first part is a portable programmable controller. This controller is based on a single-board-computer, namely the Raspberry Pi 3 Model B+. This single-board-computer is the heart of the whole programmable controller. It is connected to a 7-inch touchscreen, which is used to provide an interactive user experience. Furthermore, I use the single-board-computer to control four miniature peristaltic pumps, with three L239D drivers in between. On the pump side, I use Luer tape lock as the connection for the needle-tube set. This allows me to exchange the tube and needle with ease.

On this single-board-computer, I installed Raspbian as the operating system. On this operating system, I put our Java implemented control software. This software has different functions, but mainly provide the interface for programming the program sequence, which indicates the actions of the pumps. The whole programmable controller has only a size less than 20cm × 20cm × 20cm and weighs less than 600g.

The second part of this prototype system is the microfluidic used to perform experiments. I designed some demonstrative experiments and used a consumer-grade 3D-printing technique to fabricate the corresponding microfluidic chip. By using the 3D-printing technique lab-on-chip users can easily fabricate their microfluidic chip. In comparison to conventional fabrication methods used to fabricate microfluidic chip, lab-on-chip users can simply use a consumer-grade 3D-printer to fabricate a microfluidic chip in a very short time. This enables the user to instantly get the corresponding microfluidic chip for their experiment, which then can be used with my portable programmable controller on the go.

Lastly, I have verified the feasibility and the usability of the constructed prototype system by performing previously designed experiments with both the programmable controller and 3D-printed microfluidic chips. Though I encountered several problems while performing the designed experiments, in fact, none of the problems was able to refute the concept of a portable microfluidic system and the feasibility and usability of the prototype. Overall, the prototype of a portable microfluidic system has been realized in this thesis.

While the programmable controller I have designed is still in a prototype and only provides the most simple and basic functions such as controlling the flow by defining the run time of a pump, this portable microfluidic system can be improved or upgraded in the future in many ways. With the development of technology, computer vision becomes a very sophisticated technology today. Adding a camera to the programmable controller can help the program to "see" what is happening. While the Raspberry Pi 3 Model B+ has both a dedicated camera connector and a standard USB connector, adding a camera to it is very simple. By adding a camera to the programmable controller, the programmable controller can provide more advanced control based on what it has "seen". For instance, it can control the pump based on the location of the current fluid sample flow. Depending on where the front of the flow is, the controller can decide whether the next program step should be triggered or not.

Another improvement I can do in the future is adding a flow rate sensor and pressure sensor on each pump. Since the Raspberry Pi 3 Model B+ provides over 40 GPIO pins, I do have enough spar pins to add these kinds of sensors. The programmable controller then can regulate the flow depending on the flow rate and the pressure measured by these sensors. Of course, I can also upgrade the system with application-specific sensors, such as a color sensor, which can indicate application-specific data and provide it directly to the programmable controller, which can then decide the next step based on that, or even save the data for further processing.

Further, to ensure the stability of an experiment, the ambient temperature is also very important [31]. I can build a platform to hold the microfluidic chip and add a heating/refrigerating plate under knees to the platform. With a temperature sensor, the programmable controller can automatically regulate the experiment ambient temperature depending on the user setting and the temperature sensor data. With such a setting, certain experiments, requiring a strict temperature condition can also be performed with my prototype system.

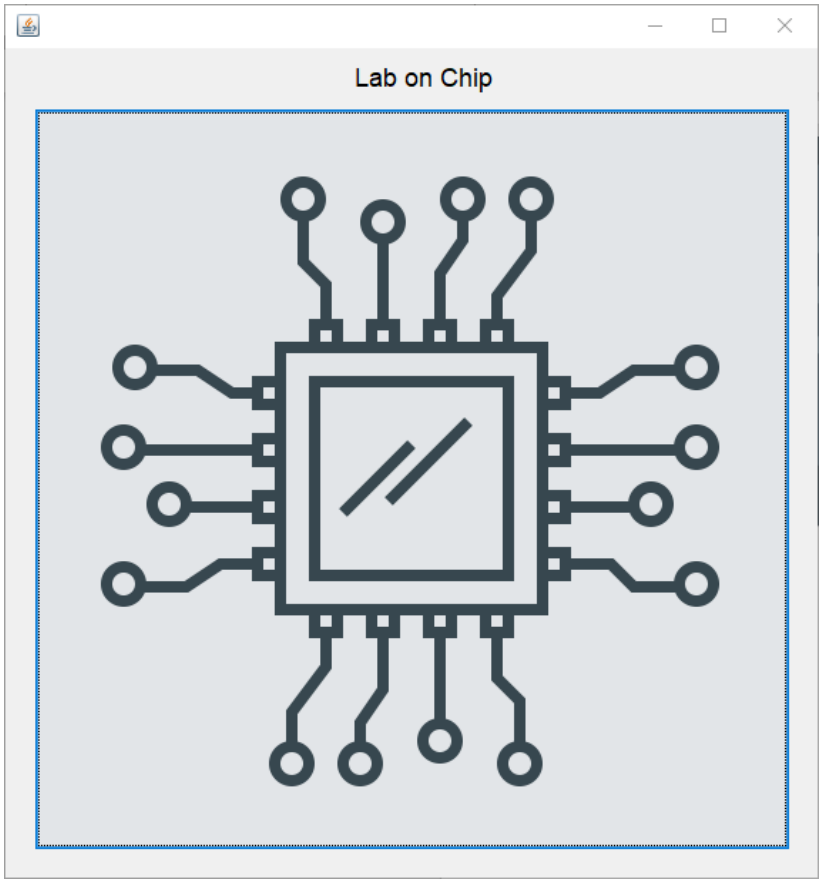
Besides functional upgrade or improvement, I can also upgrade the main board of the programmable controller. While I have stated in the third chapter, the Raspberry Pi 3 Model B+ provides only two pairs of hardware controlled PWM pins. Thus, I use software simulated PWM to connect all four pumps. A little problem of the software simulated PWM is that it is, as the name already indicates, software simulated, i.e. the timing of PWM is calculated by software using CPU power. This may have a problem when the CPU is fully loaded. In this case, the timing of PWM cannot be calculated promptly, and the control becomes unprecise and inaccurate. To solve this problem, I can add another IC between the Raspberry Pi board and the Driver IC. I could, for instance, add an Arduino board between, and program the Arduino to respond to the signal sending from the Raspberry Pi board. Accordingly, the pumps are controlled by the Arduino board.

Altogether, I believe this prototype of a portable microfluidic system showed its potential. It cannot only perform designed experiments but because of its small form factor and light-weight, it can be carried to any places and perform the wanted experiment. Especially, for medical usage, a small-sized microfluidic system can help the doctor to perform small examinations or tests, such as blood tests or urine tests, while on a home visit. I am very excited and are looking forward to the progress and development of small-sized truly portable microfluidic devices and system.

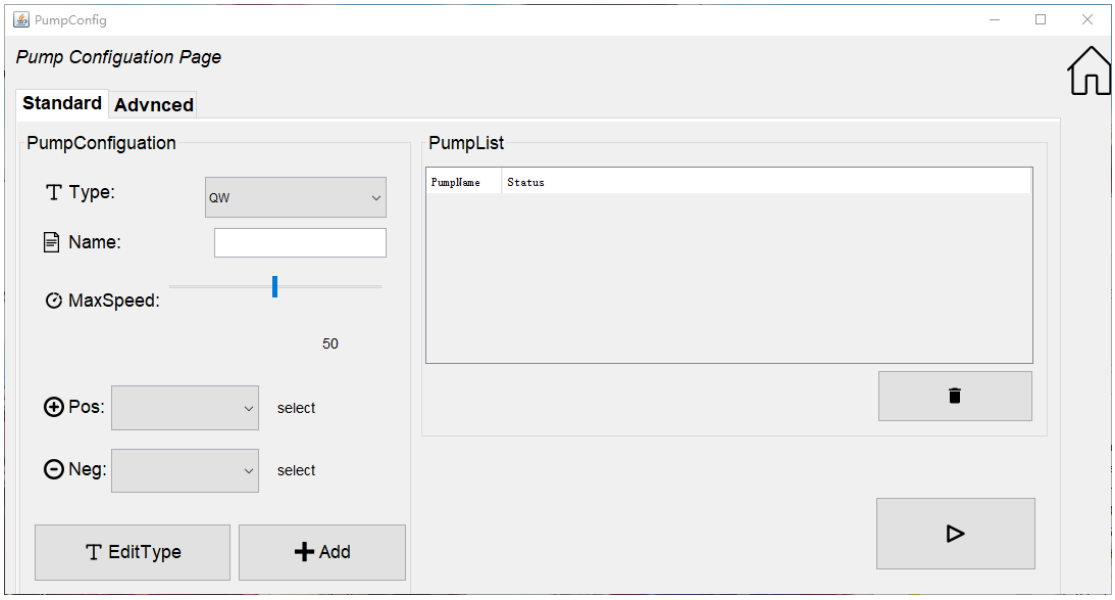
A Appendix

The software our team has implemented for this thesis (controlling software on the programmable controller) can be downloaded from Mr. Liudongnan Yang's GitHub at: https://github.com/yldn/MicroControl

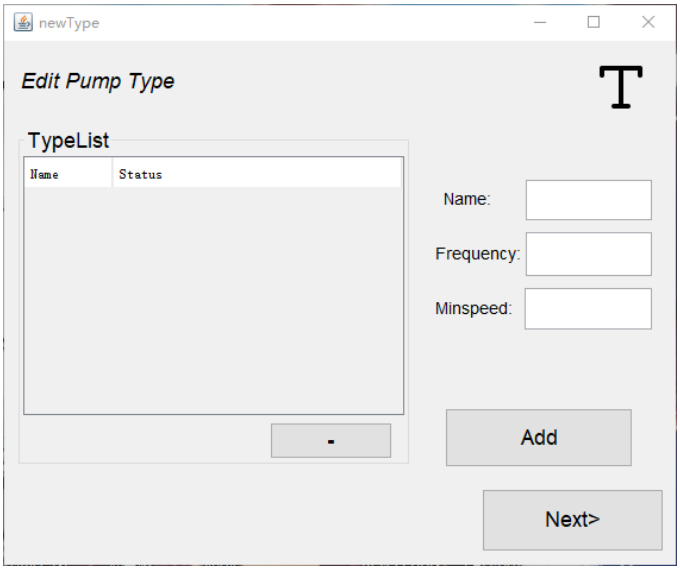
Screenshots of the controlling software:



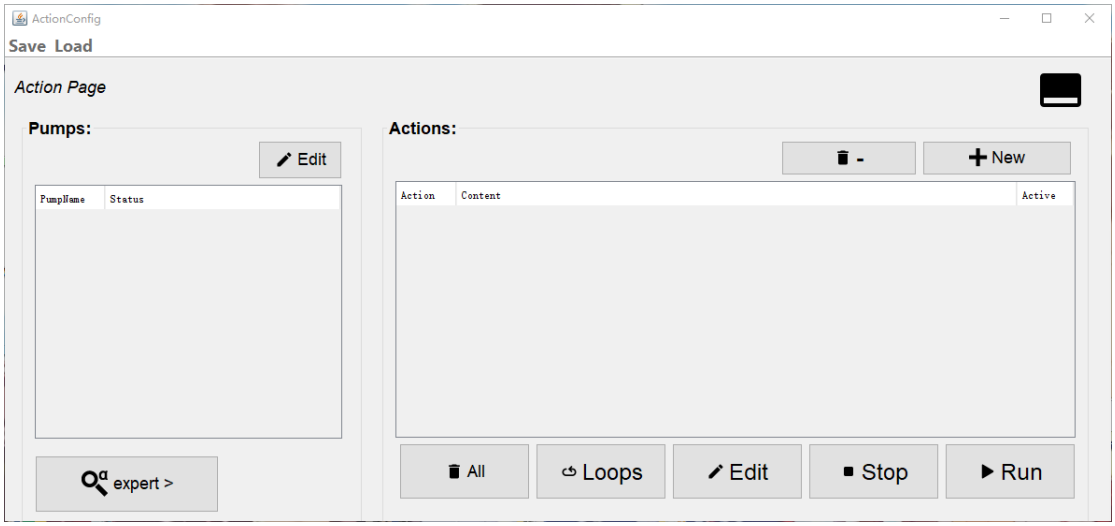
[The splash screen window]



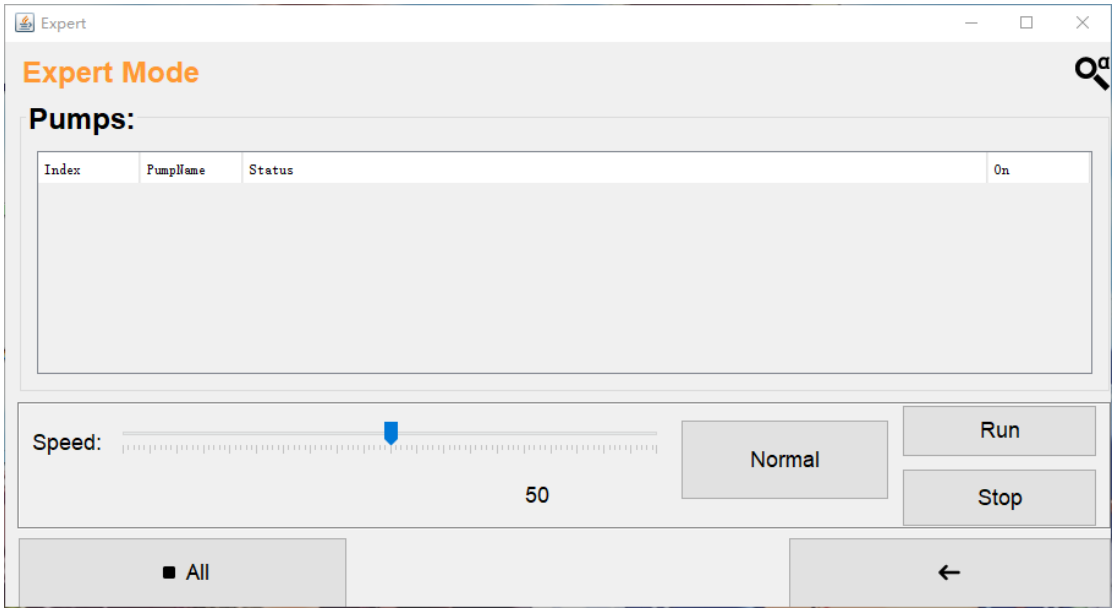
[The pump configuration window]



[The pump type configuration window]



[The program sequence window]



[The expert (advance) mode window]

Reference

[1] J. Son, R. Samuel, B. Gale, D. Carrell and J. Hotaling, "Separation of sperm cells from samples containing high concentrations of white blood cells using a spiral channel," *Biomicrofluidics,* vol. 11, no. 054106, 2017.

[2] M. Guo, A. Rotem, J. Heyman and D. Weitz, "Droplet microfluidics for high-throughput biological assays," *Lab Chip,* vol. 12, no. 2146, p. 2012.

[3] A. Jafek, S. Harbertson, H. Brady, R. Samuel and B. Gale, "Instrumentation for xPCR Incorporating qPCR," *Anal. Chem.,* vol. 90, p. 7190–7196, 2018.

[4] N. P. Macdonald, J. M. Cabot, P. Smejkal, R. M. Guijt, B. Paull and M. C. Breadmore, "Comparing Microfluidic Performance of Three-Dimensional (3D) Printing Platforms," *Anal. Chem.,* p. 3858−3866, 2017.

[5] H. Becker and U.Heim, "Hot embossing as amethod for the fabrication of polymer high aspect ratio structures," *Sens. Actuators, A 83,* p. 130–135, 2000.

[6] S.-Z. Qi, X.-Z. Liu, S. Ford, J. Barrows, G. Thomas, K. Kelly, A. McCandless, K. Lian, J. Goettert and S. A. Soper, "Microfluidic devices fabricated in poly(methyl methacrylate) using hot-embossing with integrated sampling capillary and fiber optics for fluorescence detection," *Lab Chip 2,* pp. 88-95, 2002.

[7] R.-D. Chien, "Micromolding of biochip devices designed with microchannels," *Sens. Actuators, A 128,,* pp. 238-247, 2006.

[8] U. M. Attia, S. Marson and J. R. Alcock, "Micro-injection moulding of polymer microfluidic devices," *Microfluid. Nanofluid. 7,* pp. 1-28, 2009.

[9] J. C. McDonald, D. C. Duffy, J. R. Anderson, D. T. Chiu, H.-K. Wu, O. J. A. Schueller and G. M. Whitesides, "Fabrication of microfluidic systems in poly(dimethylsiloxane)," *Electrophoresis 21,* pp. 27-40, 1999.

[10] J. Rossier, F. Reymond and P. E.Michel, "Polymer microfluidic chips for electrochemical and biochemical analyses," *Electrophoresis 23,* pp. 858-867, 2002.

[11] H. Cao, J. O. Tegenfeldt, R. H. Austin and S. Y. Chou, "Gradient nanostructures for interfacing microfluidics and nanofluidics," *Appl. Phys. Lett. 81,* pp. 3058-3060, 2002.

[12] T. Mappes, S. Achenbach and J. Mohr, "X-ray lithography for devices with high aspect ratio polymer submicron structures," *Micoelectron. Eng. 84,* pp. 1235-1239, 2007.

[13] J. Wu and M. Gu, "Microfluidic sensing: state of the artfabrication and detection techniques," *Journal of Biomedicao Optics 16(8),* August 2011.

[14] T. W. Odom, J. C. Love, D. B. Wolfe, K. E. Paul and G. M. Whitesides, "Improved pattern transfer in soft lithography using composite stamps," *Langmuir 18,* pp. 5314-5320, 2002.

[15] F. Hua, Y.-G. Sun, A. Gaur, M. A. Meitl, L. Bilhaut, L. Rotkina, J.-F. Wang, P. Geil, M.

Shim and J. A. Rogers, "Polymer imprint lithography with molecular-scale resolution," *Nano Lett. 4,* pp. 2467-2471, 2004.

[16] Q.-B. Xu, B. T. Mayers, M. Lahav, D. V. Vezenov and G. M. Whitesides, "Approaching zero: using fractured crystals in metrology for replica molding," *J. Am. Chem. Soc. 127,* pp. 2467-2471, 2005.

[17] Z.-W. Li, Y.-N. Gu, L. Wang, H.-X. Ge, W. Wu, Q.-F. Xia, C.-S. Yuan, Y.-F. Cheng, B. Cui and R. S. Williams, "Hybrid nanoimprintsoft lithography with sub-15 nm resolution," *Nano Lett. 9,* p. 2306–2310, 2009.

[18] J. Love, D. Wolfe, H. Jacobs and G. Whitesides, "Microscope projection photolithography for rapid prototyping of masters with micron-scale features for use in soft lithography," *Langmuir 17,* pp. 6005-6012, 2001.

[19] S. Mitra and S. Chakraborty, Microfluidics and Nanofluidics Handbook: Fabrication, Implementation, and Applications, FL, USA: CRC Press and Taylor & Francis Group: Boca Raton, 2012.

[20] C. Sun, N. Fang, D. Wu and X. Zhang, "Projection micro-stereolithography using digital micro-mirror dynamic mask," *Sens. Actuators A Phys. 121,* pp. 113-120, 2005.

[21] P. Preechaburana and D. Filippini, "Fabrication of monolithic 3D micro-systems," *Lab Chip 11,* pp. 288-295, 2011.

[22] Elveflow, "Elveflow Microfluidic Flow Control Products," [Online]. Available: https://www.elveflow.com/microfluidic-flow-control-products/flow-control-system/pressure-controller/. [Accessed 08 11 2019].

[23] Wiring Pi, "Wiring Pi," 2019. [Online]. Available: http://wiringpi.com/. [Accessed 13 11 2019].

[24] Pi4J, "The Pi4J Project," 05 03 2019. [Online]. Available: https://pi4j.com/1.2/index.html. [Accessed 13 11 2019].

[25] The Listening Post CHRISTCHURCH, "DLP Projection Technology," [Online]. Available: https://www.listeningpost.co.nz/News-Tips/Glossary/Dark-Chip-dlp-dc3-dc4-__I.125618??. [Accessed 13 11 2019].

[26] R. Pires, "SLA vs DLP: The Differences – Simply Explained," 12 10 2019. [Online]. Available: https://all3dp.com/2/dlp-vs-sla-3d-printing-technologies-shootout/. [Accessed 13 11 2019].

[27] H. Biwu, H. Bofen, W. CHEN 和 W. Lizhuan, "Determination of the critical exposure and the penetration dep th of the photosensitive resin for Stereolithography," *Information Recording Materials, Vol. 8, No. 1,* pp. 59-62, 2007.

[28] H. Chen, S. Potluri and F. Koushanfar, "BioChipWork: Reverse Engineering of Microfluidic Biochips," in *IEEE 35th International Conference on Computer Design*, 2017.

[29] B. He, B. Burke, X. Zhang, R. Zhang and F. Regnier, "A picoliter-volume mixer for microfluidic analytical systems," *Anal. Chem. 73,* p. 1942–1947, 2001.

[30] H. Zhang, E. Tumarkin, R. Peerani, Z. Nie, R. M. A. Sullan, G. C. Walker and E.

Kumacheva, "Microfluidic Production of Biopolymer Microcapsules with Controlled Morphology," *J. AM. CHEM. SOC. 9 VOL. 128, NO. 37,* 2006.

[31] E. Tumarkina and E. Kumachevaab, "Microfluidic generation of microgels from synthetic and natural polymers," *Chem. Soc. Rev., Vol. 38, No. 8,* p. 2149–2496, 08 2009.

[32] J. Sun and H. Tan, "Alginate-Based Biomaterials for Regenerative Medicine Applications," *Materials 6,* pp. 1285-1309, 2013.

[33] Thewrightstuff, "Examples of PWM with different duty cycles," 6 11 2019. [Online]. Available: https://en.wikipedia.org/wiki/Pulse-width_modulation#/media/File:Duty_Cycle_Examples.png.

[34] C.-Y. Lee, C.-L. Chang, Y.-N. Wang and L.-M. Fu, "Microfluidic Mixing: A Review," *Int. J. Mol. Sci.,* pp. 3263-3287, 12 2011.

[35] Aquatech Co., Ltd., "About Ringpump | Aquatech : Development, Production & Sales of Peristaltic pump & Micro pump," [Online]. Available: http://www.ringpump-aquatech.co.jp/ring_pump/about.html. [Accessed 12 11 2019].

[36] L. C. Waters, S. C. Jacobson, N. Kroutchinina, J. Khandurina, R. S. Foote and J. M. Ramsey, "Microchip device for cell lysis, multiplexPCR amplification, and electrophoretic sizing," *Anal. Chem.,* pp. 158-162, 1998.

[37] T. Ichiki, T. Ujiie, S. Shinbashi, T. Okuda and Y. Horiike, "Immunoelectrophoresis of red blood cells performed on microcapillary chips," *Electrophoresis,* pp. 2029-2034, 2002.

[38] P. C. H. Li and D. J. Harrison, "Transport, manipulation, and reaction of biological cells on-chip using electrokinetic effects," *Anal. Chem.,* pp. 1564-1568, 1997.