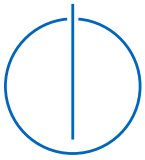
DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics: Games Enginnering

Development of control sequence programming tool for peristaltic pumps

Liudongnan Yang



DEPARTMENT OF INFORMATICS

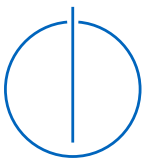
TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics: Games Enginnering

Development of control sequence programming tool for peristaltic pumps

Entwicklung eines Steuersequenzprogramms für Schlauchpumpen

Author:	Liudongnan Yang
Supervisor:	Prof. Dr.-Ing. Schlichtmann, Ulf;
Advisor:	Dr.-Ing. Tseng, Tsun-Ming;
Submission Date:	18.12.2019



I confirm that this bachelor's thesis in informatics: games enginnering is my own work and I have documented all sources and material used.

Munich, 15.12.2019 Liudongnan Yang

Acknowledgments

This thesis would not have been possible without the sedulous support of Prof. Dr.-Ing. Ulf Schlichtmann , Dr.-Ing. Tseng Tsun-Ming, and B.Sc. Yushen Zhang. They always listened and gave useful and constructional suggestions and guided me to the fix the problem as soon as possible. Therefore, I hereby want to express my great thanks, especially to Dr. Tseng for giving me this expressive and exciting Topic as well as Professor Schlichtmann.

Besides I want to thanks Mr. Yushen Zhang, who always supported me during the Bachelor Studies by doing a tutorial for various courses and providing professional advises. Furthermore, he has never once hesitated to provide me the help that I needed while composing and implementing.

I would also like to thank the Department of Electrical and Computer Engineering in Technical University of Munich for granting me access to the laboratories and computer rooms.

Last but not least the thanks will go to everybody who always generously helps me in any matter of the whole bachelor study.

Abstract

The development of microchips is getting more popular throughout the year, depending on the project people build, the form of chips can be various. Microfluidic technologies nowadays integrated with Lab on Chip devices which allow us to manipulate the behavior of the fluid inside the micro-sized channels.

However, a large amount of microfluidic projects have been developed in laboratories. Here I develop a prototype of concrolling software on microfluidic devices in the background of low-cost and portability. The project is based on resin-based chip and single-chip computer, for bio-medical or chemical technology applications.

The main purpose of the thesis is to develop the software prototype of the framework on 3D printed chips with micro-level chips which supervised the fluid through peristaltic pumps. Users are able to control the fluid's behavior by manipulating and defining the sequence of pumps, that the fluid gets reacted as the user's wish.

The chip we created is the chip that supports the chemical reaction in it, and lead to a result solution.

The Bachelor's thesis documents the process of developing the controlling system.

Inhaltsangabe

Die Entwicklung Die vorliegende Arbeit zeigt eine Applikation um zu kontrollieren peristaltic pumpe, die sich auf Mikrochips verbunden und gegebene Chemische Lösung herein zu bringen und lassen entsprechende Reaktion in Chambers von Mikrochips durchführen.

Mikrofluidik-Technologien, die heutzutage in Lab on Chip-Geräte integriert sind und es uns ermöglichen, das Verhalten der Flüssigkeit in den mikroskaligen Kanälen zu beeinflussen. Es wurden jedoch große Mengen von Mikrofluidik Projekten in Laboratorien durchgeführt, von denen es nur wenige gibt, die zu kommerziellen Ergebnissen führen.

Hierfür entwickeln wir mikrofluidische Geräte auf der Basis von Resin-chips und Single-Computern für biomedizinische oder chemisch-technische Anwendungen.Das Ziel für diese Thesis ist die Software für das Prototyp von 3D gedrukte Chip mit integrieren von Schlauchpumpen zuentwickeln, mit dem man einfach auf pumpen wie gewünschte Sequenzen und Reinfolge zugreifen und steuern kann.

keywords : microfluidic , biomedical , chemical , Lab on Chip (LoC)

Contents

List of Figures

List of Tables

1 Introduction

1.1 History

The origin of the research in the area microfluidics dates back to the late 1960s.From the book [1] on Chapter 1 we know that the main two projects have built by Stanford University and IBM, which contains different fields of knowledge as such project of gas chromatography(GC) [2], and the inkjet printer. Before that, some researchers already get the information by analyzing the fluids and calculate the fluid behavior in small diameters."They once analyze the behavior and tried to match the color and appearance of the urine with symptoms, but they did not have an opportunity to use microfluidic chips and managed to verify colors and appearance of the samples in Experimental container known as “Matulas”" [1].

1.1.1 Prototyping in microfludics

After the research of the area microfabrication [3] & microfluidics [4] , we finally saw the first prototype of “Lab-on-Chip” at Stanford and Purdue University [5] at the end of the 1960s.

A few years later Siemens [6] invented the first continuous inkjet printer used in the medical area based on the idea of William Thomson [7], who has the patent of the use of electrostatic forces to control the drops onto paper in the year of 1967.

Environmental boom in 90’s

In the year 1993, then things got slightly changed. An article of the electrophoresis based chemical analysis system on a chip was published by Andreas Manz [8] , and the microchip has the scale of 1cm by 2 cm was micromachined with a cross-section of $10 \times 30 \mu m$ in electroosmotic [9] pump. It shows that it is possible to create a Lab-on-Chip project that could be used for complex systems.

One year after, the First International Conference on Micro-Total Analysis System(μTAS) took place on Enschede The Netherlands [10], in which the position of the forum for the research results in microfluidics ; Lab-On-Chip; micro-fabrication has been stabilized and strengthened. At the same time, Chemical and Biological Microsystem

Society (CBMS) [11] organized the first workshop on the Micro-Total Analysis System (μTAS) and present its latest technologies. Then in 1998 the publication [12] by Professor George & Whiteside who is one of the leader of the microfluidics and Lab-on-a-chip community, which describes the rapid prototyping of microfluidic systems in polydimethylsiloxane(PDMS). The design and fabrication of microfluidic chips are introduced to the publication and since then it has been widely used in various studies.

1.1.2 Further research and prototyping in recent years

In 2000, an innovation group from Duke University published an article [13] of discrete liquid droplets proficient by the control of surface tension which is one of the first researches that characterize liquid handling technology [1]. It allows scientists to control the picoliter to microliter of droplets that this new method has then huge impact on biochemical based applications, immunoassays etc.

Then a couple of years later, there's a technology coming out called 3D imprinting, developed in the US by Charles Hull [14]. It shows that it is possible to fabricate mini-sized reactors for scientific used such as chemical syntheses , Bio-chemical reactions.

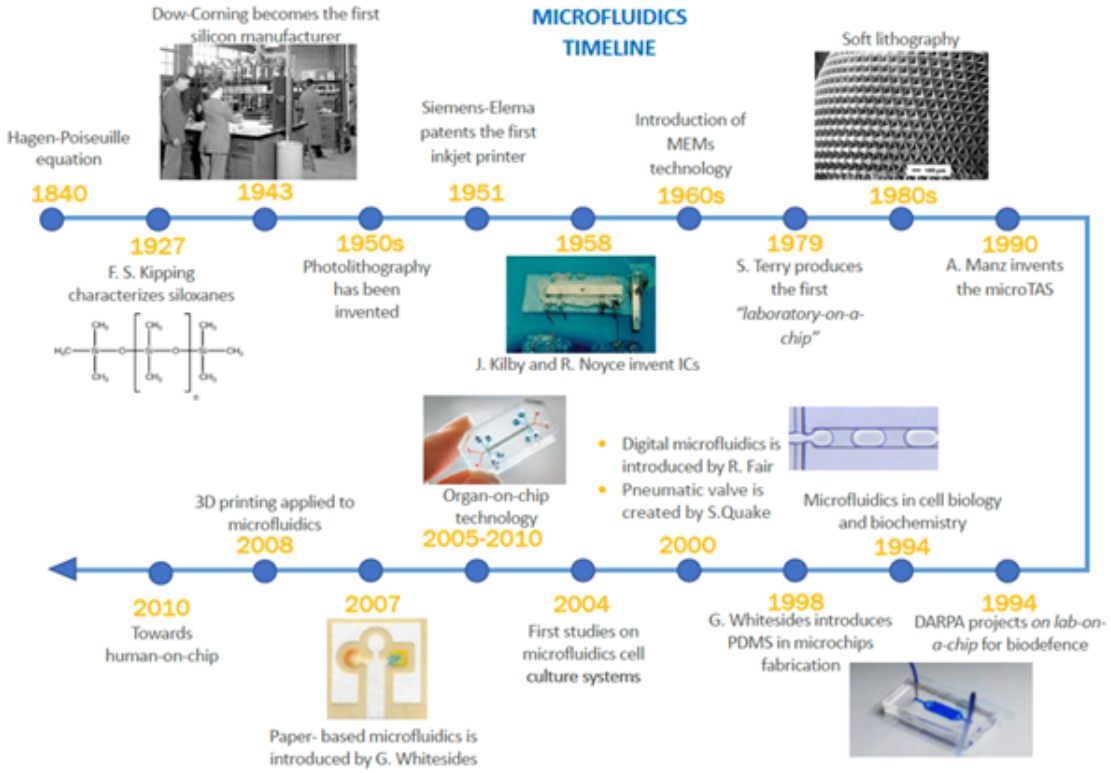


Figure 1.1: The history time line [15]

Later, the technology gets developed by scientists from different discipline. Here's an overview Figure 1.1 of the history throughout these years. The material used for those projects is polydimethylsiloxane(PDMS) which was introduced in 1998. The Figure shows different time and the different project which is done by specific research group or scientist:

- (a) the structure of PDMS
- (b) foundation of Dow Corning in Midland (1943) to work on silicones (IHS media)
- (c) the first germanium prototype IC created by Jack Kilby (1958) (Texas Instruments)
- (d) SEM image of a PDMS microstructure processing
- (e) PDMS microfluidic chip

- (f) control of discrete droplets
- (g) an example of organ-on-chip
- (h) a paper-based microfluidic device can be used as low-cost, portable diagnostic tool, resulting to be particularly useful in emergency conditions and in resource-limited areas.

Above all, the features of this new technology which could lead to the clear future is to support large-amount integrated devices . The way of building between Labs and different 3D range could be easy . Thus with those who is the leader of the third decade of microfluidic revolution, they will now have a big impact on biomedical; chemical; different kinds of areas [15].

1.2 Opportunities and challenges

Through out years of research, the challenge about to face large-scale applications potentially requires a lot of work for the fabrication of user-friendly devices and their integrations. In order to help accelerating these technologies, the technique issues need to be solved. As a matter of fact, some recent report of Lab-on-Chip devices integrated into smart phones shows that the idea of the connection between microfluidic devices with the smart phone is available by the inspiration of those features shown above. We face these opportunities and challenges coming up with an idea of building a prototype of the Lab-On-Chip device and test out the result and conclude difficulty in the chapter 6.

Referring to the use of Microfluidics, more scientists are thereby settled in this area by sharing their experiences through conferences, blogs, posts, website forum with the main topic of microfluidics. Here's a table that represents these data in Table 1.1 and Table 1.2. These resources are mostly good for those who need the environment to get familiar with the field and potentially getting involved in the community and collaboration interdisciplinary(biology, physics, and chemistry). And we can find the answer to why the community of microfluidics has been able to grow so much and should obviously influence our daily life and eventually make a huge impact on our life.

Table 1.1: Sources of information about microfluidics and lab-on-a-chip [1]

name	Editor	comment	link
Computers and fluids	Elsevier	Publish the development of numerical methods relevant to fluid flow, computational analysis of flow physics and fluid interactions	http://www.sciencedirect.com.globalproxy.cvt.dk/science/joumal/(X)457930
Lab on a chip	Royal Society of Chemistry	Publish both fabrication and applications of LOC devices	http://www.rsc.org/publishing/joumals/lc
Microfluidics and nanofluidics	Springer	Publish reports or research, techniques, and applications in microfluidics	http://link.springer.com/joumal/10404
Optofluidics, microfluidics and nanofluidics	De Gruyter	Publish mostly applications of microfluidics	http://www.degruyter.com/view/j/optof
Analytical chemistry	ACS	Publish mostly applications of LOC devices	http://pubs.acs.org/jounal/ancham
Biosensors and bioelectronics	Elsevier	Publish mostly applications of LOC with integrated sensors, e.g., electrochem., optical sensors	http://www.sciencedirect.com/science/journal/09565663
Microelectronic eng.	Elsevier	Publish mostly on techniques for fabrication of LOC devices	http://www.sciencedirect.com/science/journal/01679317
Biomedical microdevices	Springer	Publish research in the diagnostic and therapeutic applications of LOC devices	http://link.springer.com/jounal/10544
Biomicrofluidics	American Institute of Physics	Publish on theory and applications of microfluidics	http://scitation.aip.org/content/aip/journal/bmf
Journal of Micromechanics and Microengineering	Institute Of Physics (IOP)	Publish on microfabrication of LOC devices	http://iopscience.iop.org/0960-1317/
Journal of Microelectromech. Systems	IEEE	Publish on the theory, modelling, design, fabrication, assembly, and packaging of LOC and microfluidic devices	http://eds.ieee.org/jounal-of-microelectromechanical-systems.html

Table 1.2: Conferences on Microfluidics, Lab-on-chip, and MicroTAStechnology [1]

Conference	Organizer	Topics	Link
Internat.Conf. on miniaturized system for chemistry and life sciences μTAS	The Chemical and Biological Micro systems Society	Premier forum for reporting research results in micro fluids, micro fabrication, and applications of Loc and μTAS devices	http://www.microtas2014.org/
Conference on microfluidic handling systems	Freiburg University	Discuss the latest results in the field of micro fluidichandling	http://www.mfhs2014.uni-freiburg.de/
European conf. onmicrofluidics		Design, fabrication and theory of LOC device	http://microfluidics2014.eu
Point-of-care diagnostics World conference	Select Biosciences	Discuss the point-of-care diagnostic field	http://selectbiosciences.com/conferences/index.aspx?conf=POCDWC2014
Lab-on-a-chip European congress	Select Bio Sciences	Discuss innovative developments in LOC, microfluidics, and microarray	http://selectbiosciences.com/conferences/index.aspx?conf=LOACEC2014
Lab-on-a-chip and micro array World congress	SelectBiosciences	Discuss innovative developments in LOC,microfluidics, and microarrays	http://selectbiosciences.com/conferences/index.aspx?conf=LOACWC201414

2 Background

Nowadays microfluidics is a distinct and popular technological field of biochemistry related area. Thanks to the early development starting from the 1950s, the interest in developing miniature chips and patterns for use in systems to help creating complex 3D micro-tubes on semiconductors has been increased. Miniaturized sensors and other components were integrated with microcomputers or computer chips to establish a portable integrated platform to use as environmental or medical measurement systems.

2.1 3D printed chip

Concerning chip design, the design software will be able to give us a clue from the concept to the actual model. The 3D chip is currently been designed by the Computer Aided Design(CAD) program. For designers, the complete knowledge of using CAD software like AutoCAD is essential to make the workflow optimized. Thereafter, the design concept should also be implemented accordingly and carefully.

Currently, with the help of a resin-based 3D printer, the flow-based channeled microfluidic chip has been printed with the help of a precise 3D printer. The tube with a micro-sized diameter can be built through the machine.

2.2 Sequence design

As the microfluidic project become more and more popular. Different services are being provided for different kinds of users. The design of the sequence has become a significant role in the project. To illustrate and create good working sequence, supporting system has become more crucial, that on the one hand, users are able to supervise the system workflow easily and clear through the Graphic User Interface (GUI), on the other hand, it is also essential for the development team to make discussion on the optimization of the system.

parameter	value
Method	DC geared motor
Typical flow rate	0.2 ml/min
Maximum pump pressure	50 kPa
Outer dimensions	$12 \times 14 \times 31.5$
Wetted materials	Norprene
Port connection	I.D. 1.2 mm tubing
Rated valtage	DC3V
Power consumption	0.12 W
Self-priming	Possible

Table 2.1: specification of pump Figure 2.1

Figure 2.1: peristaltic pump RP-Q1.2N [16]

2.3 Peristaltic pumps

The terminal devices always have a mechanical part to interact objectively. We have chosen the Peristaltic pump, Version RP - Q1.2N in Figure 2.1 to drive and lead chemical solutions to the flow-based microfluidic chip.

There are some important parameters of the pump which may influence the result. Therefore I listed specifications in the Table 2.1

Due to the structure of the pump in Figure 2.2, the core wheels are designed to drive the solution in the pipe.The fluid in the pump goes forward when the big wheel pass through the centerline of the pump. Peristalsis will result in the movement of fluids every time that the ring turns around on the axis.

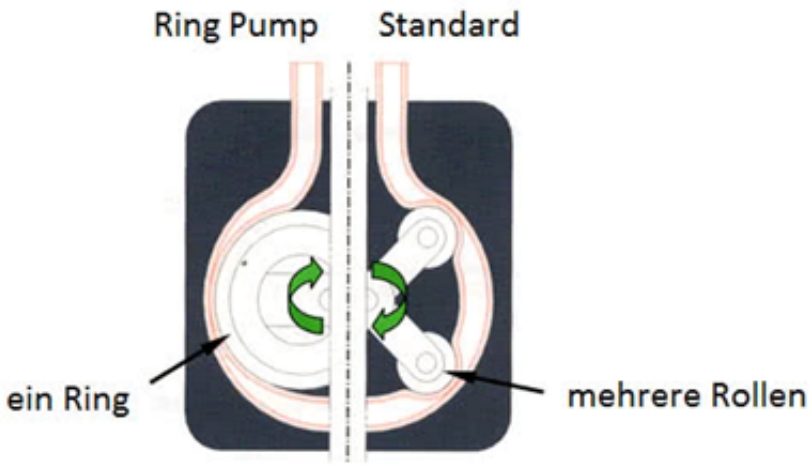


Figure 2.2: pump structure [17]

3 Method and Tools

With the background of the project and intricacy listed above, the general idea was gradually coming up to the mind that we focus on Lab-On-Chip projects. Thus, more features could potentially increase the possibility and portability.

Various kinds of tools are considered to be used, especially for the core part of controlling the hardware and send a specific signal by using APIs provided.

After discussion with the other team member, Mr. Yushen Zhang, involved in this project, we come up with an initial idea and make up our mind using the single-chip computer Raspberry Pi. To achieve the idea of the prototype. The following points are essential tools that will be described in the sections below:

Hardware side

- using pulse width modulation (PWM) to simulate signals, which are input by user
- convert signals from raspberry pi to the signals needed by peristaltic pump

Software side

- access pins on raspberry pi with Wiringpi & Pi4J APIs
- designing and structuring tool for software model–view–controller (MVC) pattern

3.1 Pulse Width Modulation(PWM)

Definition of PWM could be long and redundant so here is the definition I found in Wikipedia page which is pretty clear and understandable, "Pulse width modulation (PWM), or pulse-duration modulation (PWM), is a method of reducing 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 fast rate. The longer the switch is on compared to the off periods, the higher the total power supplied to the load. ",

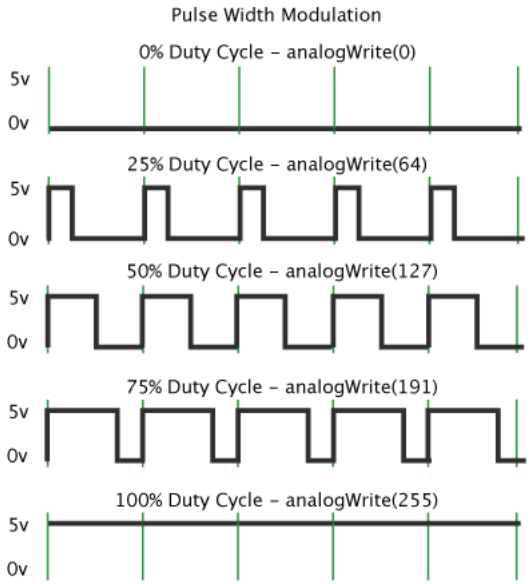


Figure 3.1: pwm signal[18]

furthermore the PWM method has been widely used in various kind of projects "PWM is particularly suited for running inertial loads such as motors, which are not as easily affected by this discrete switching, because they have inertia to react slow. The PWM switching frequency has to be high enough not to affect the load, which is to say that the resultant waveform perceived by the load must be as smooth as possible."[18].

The rate(frequency) of PWM in which power supply provision highly depends on the application. For example: the rate of electric stove need to be switched several times during a minute; for lamp dimmer, it should be 120Hz; for a motor drive the rate should be set between a few kilohertz (kHz) and tens of kHz; audio amplifiers and computer power supplies have the rate of tens or hundreds of kHz. "PWM has also been used in certain communication systems where its duty cycle has been used to convey information over a communications channel"[18].

Analyzing the signal using Pulse width modulation(PWM) in Figure 3.1 By setting different kinds of parameters in the formula given:

$$pwmFrequency(Hz) = \frac{19.2 * 10^6}{pwmClock * pwmRange}$$

The equation above shows that the frequency has the relationship between the clock and the range of pulse width modulation. Therefore, we set the clock fix. Then only by configuring the range, we can get a specific frequency. Correspondingly, we set

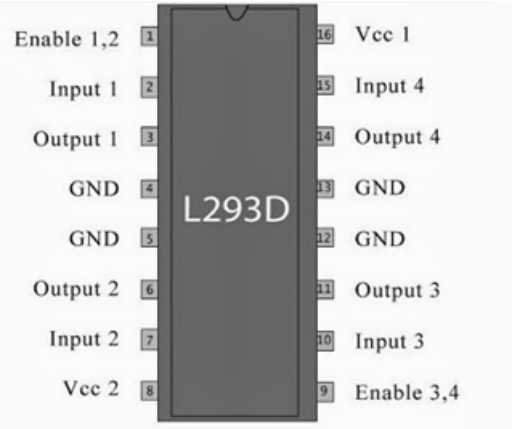


Figure 3.2: L293D [20]

primarily the clock to 1920 which is a recommended value of raspberry pi. Then we get the frequency of the project by setting Range to 100, and we get the value of the frequency of 100. We save these values as default and apply them eventually for the parameterization and specification of software development [19].

3.2 L293D chip

To drive the Motor of peristaltic pump, we need to figure out the solution that converts the PWM signal to the signal which peristaltic pump needs, Therefore we introduce a High Current Motor driver, which does the job as wish.

L293D is a high-current half-H driver, which is designed bi-directionally to provide 1A at a voltage from 4.5V to 36V on both sides as shown in Figure 3.2.

We have found out that L293D motor driver has totally 4 input pins and corresponding output pins. Intuitively we managed to input the signal from input pin, then the motors were driven by the output signal consequently.

3.3 Open sources Application Programming Interface (API)

In the project, users can control the system by interacting with the UI elements of the software thereby changing electrical potential with the PWM method on pins of Raspberry Pi. The signal will continue transport to the motor driver L293D. The motor pump will be finally driven by the output signal from the motor driver. Consequently, the fluid will be later injected into the pre-printed 3D microchips by the pumps. As a result, the reaction occurring correctly.

According to the hardware requirements. The following sections are going to shortly describe open-source APIs, that are useful tools to access the control of the pins on raspberry.

3.3.1 WiringPi

WiringPi is a pin-based General Purpose Input Output (GPIO) access library written in language C for the BCM2835, BCM2836 and BCM2837 system on a chip (SoC) devices used in all Raspberry Pi Model, which includes gpio command-line utility. Furthermore, it can be used to set up and program the GPIO pins. Read and write the pins that can be controlled by a shell script directly. A detailed structure is implemented on the Wiringpi website [21].

3.3.2 Pi4J

Pi4J provides a "friendly object-oriented Input Output I/O API and implementation libraries for Java Programmers to access the full I/O capabilities of the Raspberry Pi platform" so that the efficiency of the development therefore improved[22].

The API inherent from WiringPi API that provides full GPIO control of Raspberry Pi called Pi4J.The official website has provided some samples on the Pi4J online tutorials [23]. By understanding the API, the hierarchy is been listed as the following :

`Interface GpioPin`

This interface describes all operations over single GPIO pin.

We are interested in one of the sub-interfaces listed in the table Table 3.1

As aforementioned, we are interested in simulating PWM signals for the hardware side, the Class GpioPinPwmOutput is the first candidate that we take into consideration, then potentially using the Class in the actual implementation.

3.4 Netbeans Integrated Development Environment(IDE)

The Integrated Development Environment(IDE) can strongly improve the speed of software development in an efficient way. As a well-known IDE, Netbeans has always been adding new features which can provide a ranged convenient environment for building Java GUI application.

The installation of Java and Netbeans is simple and straight forward, thereafter the GUI design API Java-Swing is ready for use.

Classes
GpioPinAnalog
GpioPinAnalogInput
GpioPinAnalogOutput
GpioPinDigital
GpioPinDigitalInput
GpioPinDigitalMultipurpose
GpioPinDigitalOutput
GpioPinInput
GpioPinOutput
GpioPinPwm
GpioPinPwmOutput

Table 3.1: Pi4J subinterfaces

4 Design

4.1 Use case specification

The use case can be specified and categorized by different milestones :

- before the configuration
- after the configuration
- user is able to control signal output freely

When the user first interacts with the system, they should start off configuring all the settings. Thus after the configuration phase, pump intuitively initialized and being controlled by the order as user's wish.

Next, users are able to run the system by the sequence, which is composed of initial time and end time.

Moreover, the system should also be available for the user to control each pump manually. Manipulation of configured pump trough a separate interface distinct from the main workflow, in which I create the flow diagram showed in: Figure 4.2.

According to the research of wheel of emotions in Figure 4.1 by Robert Plutchik [24], user should have feeling of joy when using the software

4.2 From idea to design

The connection between software and tools is one of the most essential parts of this thesis. Therefore the model design concept has become a more and more important part of the project.

To realize the use case described in the last section. The concept design using the Model–View–Controller(MVC) pattern is one of the most suitable patterns for the project." Model–View–Controller (usually known as MVC) is a software design pattern commonly used for developing user interfaces that divide the related program logic into three interconnected elements."[25] As depicted in the Figure 4.3, the model is the central part of the pattern.

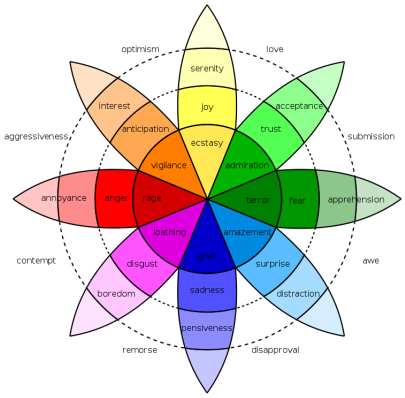


Figure 4.1: Plutchiks wheel of emotions[24]

The design of the main classes was mainly in the model part of the pattern so that the system could interact with the view and controller. Furthermore, the view part of the pattern indicates the information from the model and represent to the user in time, this idea was achieved by the Java-Swing[26] API, visualized designing tool by Netbeans IDE. Additionally, for the last controlling part, Java classes (class Pump) which contained the direct API access was introduced.

We use the Unified Modeling Language(UML) to analyze and modeling the need of the system .In following diagrams :Figure 4.5 and Figure 4.4 included all detailed design concept. Especially, the Configuration class was created to store the information of every action and pump from the list. The configuration information was transformed into the specific text file(mostly JSON string) to save and store later for reuse of the sequence. These were included in the *Util* class which implemented all the utility functions.

As mentioned above in the flow diagram The sequence diagram shows the whole process of the user accessing the system. Pretty much the same idea forms the flow diagram: Figure 4.2, Figure 4.4 shows exactly how different systems interact with each other and which information are send to where. Additionally, pump configuration required complete information of pump type meaning that pump configuration can only be based on the existing type. So as the action required complete information of the pump. In that case, the user can only create action on existing pumps.

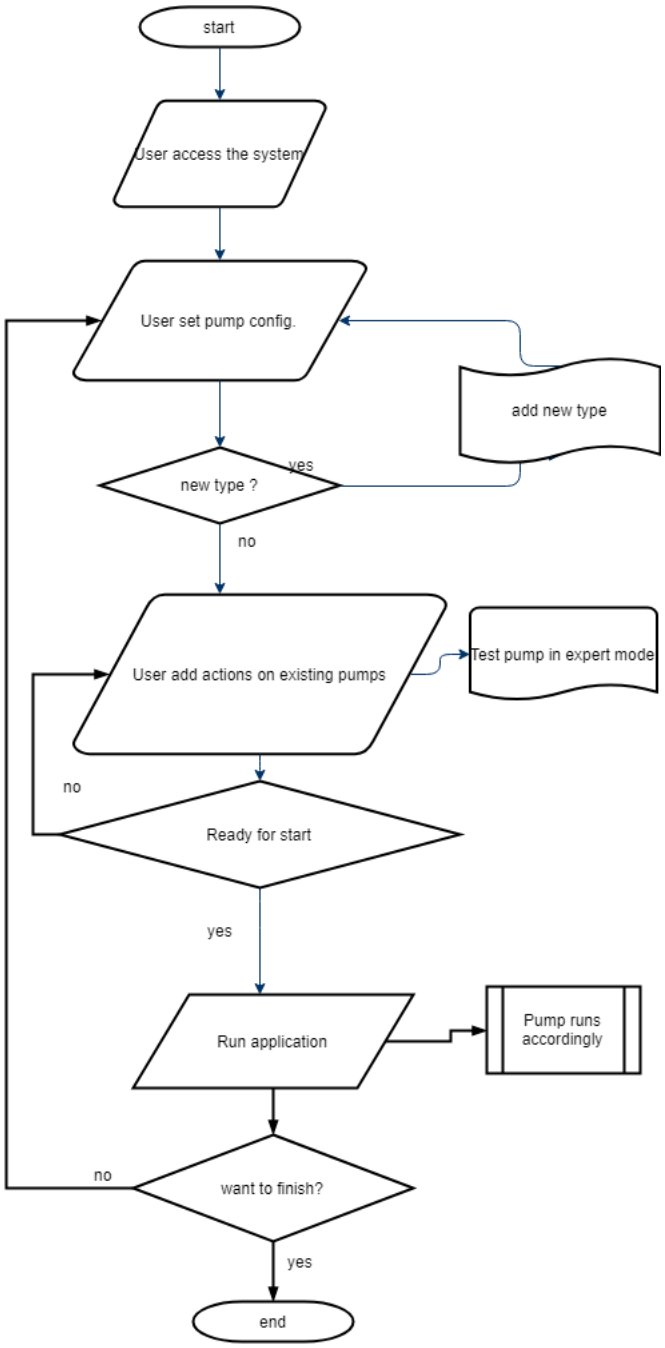


Figure 4.2: flow diagram

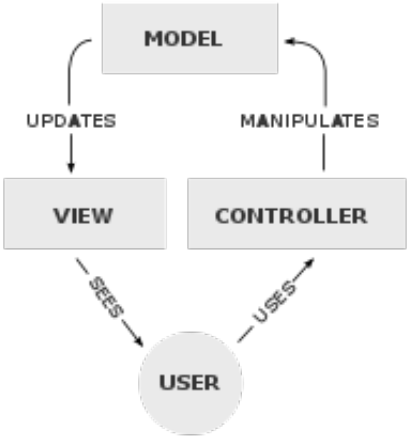


Figure 4.3: MVC-Process[25]

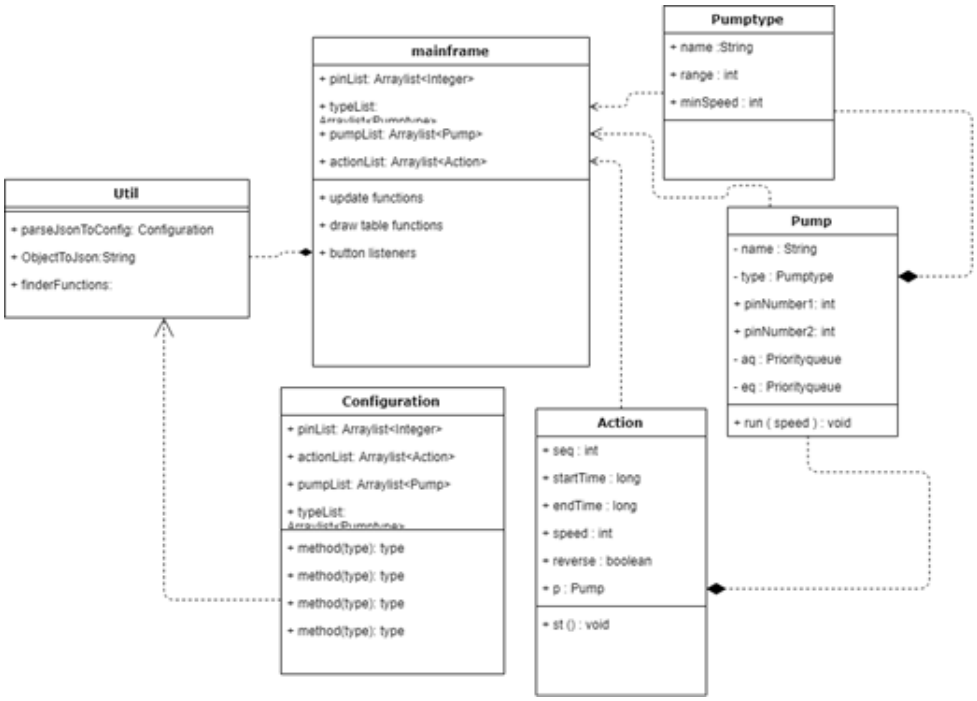


Figure 4.4: UML Class Diagram

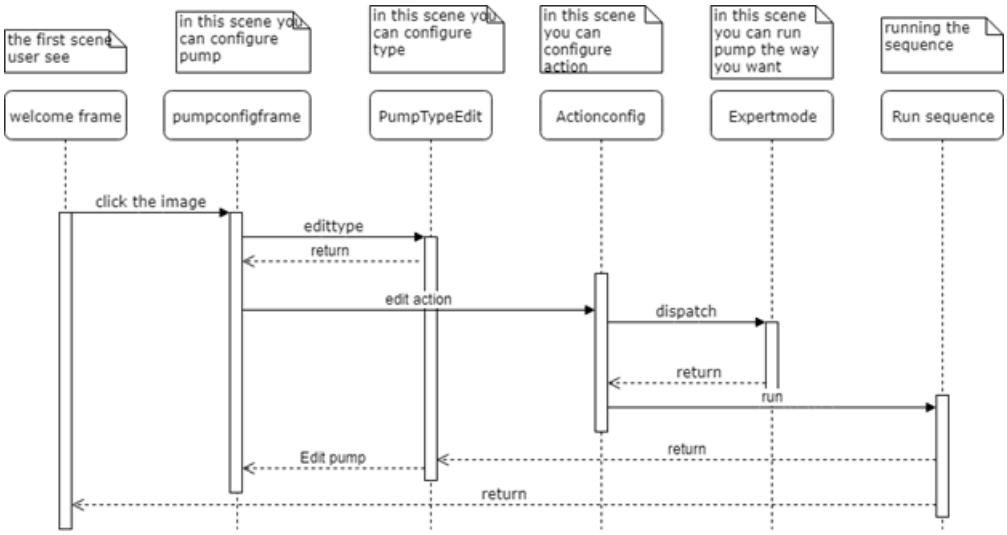


Figure 4.5: UML Sequence Diagram

5 Implementation

5.1 Graphic interface

The Graphic User Interface (GUI) has been designed and built by Netbeans IDE. Thanks to this efficient tool. It saves a lot of time. GUI for the system contains 8 frames:

- PumpTypePage
- PumpconfigMainFrame
- ActionConfiguationMainPage
- PumpActionPageNew
- PumpActionEditPage
- PumpActionPageNewLoop
- PumpActionPageNewLoop
- Welcomepage

All these frames are stored in the class *MainFrame* as well as other components designed in the frame(exp. buttons and text field components...). Overall, the three main scenes that contribute the entire program are the pump configuration scene Figure 5.1, action configuration Figure 5.2 scene, and the expert mode Figure 5.3.

The pump configuration scene is an interface in which users can select various settings and add or delete pumps. The pump type, name, speed, and pins for controlling the pumps can all be selected and implemented by clicking on the "add" button in the lower-left corner of the scene, as shown in Fig. 9. Then, the newly configured pump is shown on the right side of the list. By clicking the "EditType" button when a new pump is connected to the chip, the pump type is edited, added, or deleted on the type configuration page. Additionally, a user can delete a pump easily, with just a simple click on the trash bin on the right side of the scene. After all the pumps are configured, the user moves to the next step.

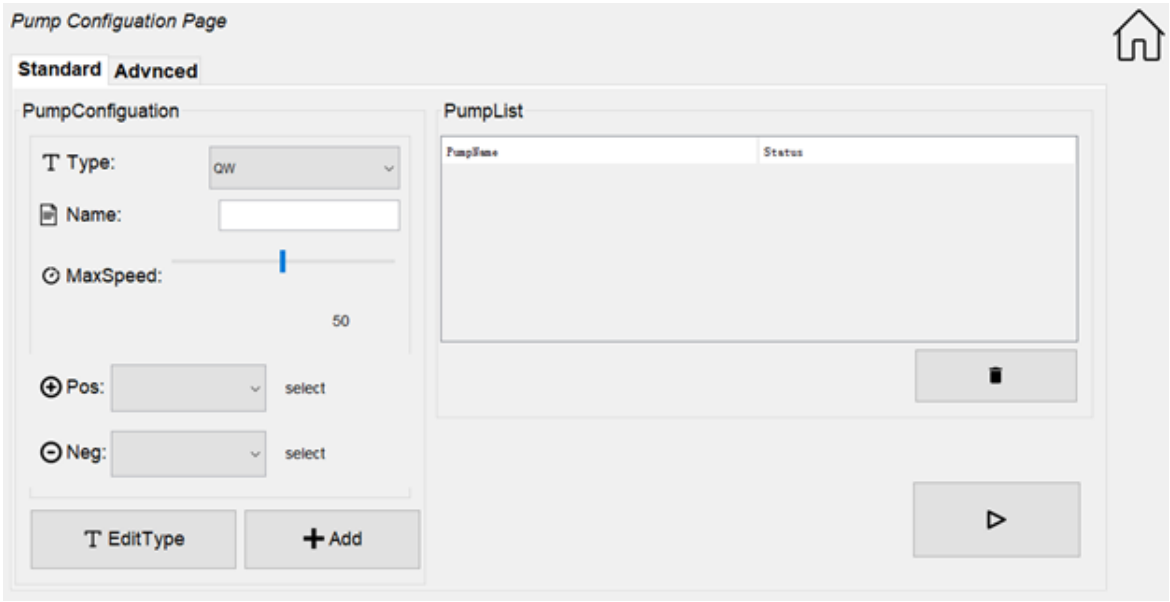


Figure 5.1: Pump Configuration

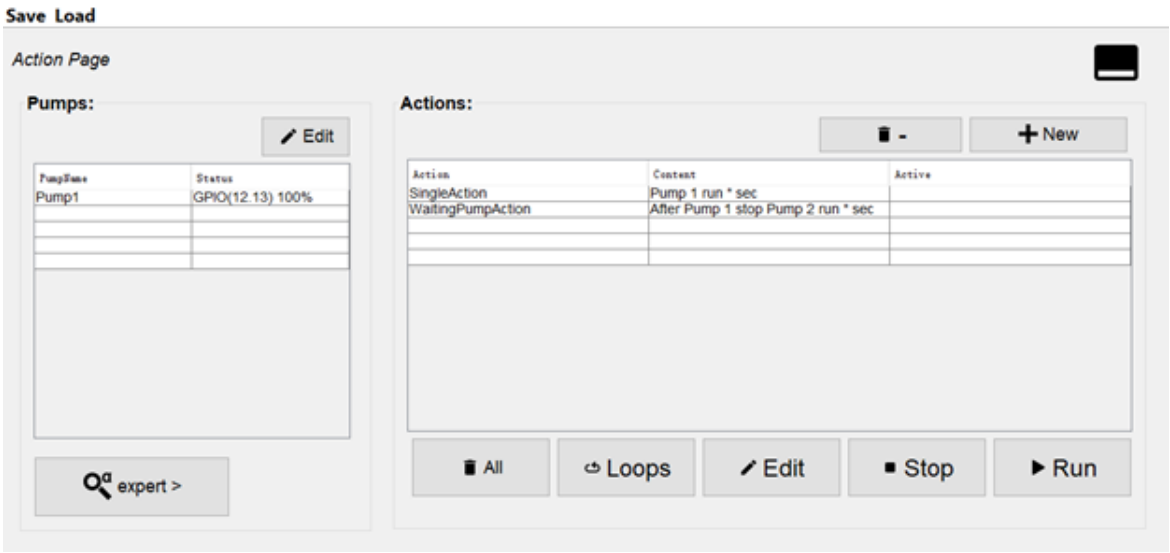


Figure 5.2: Action Configuration

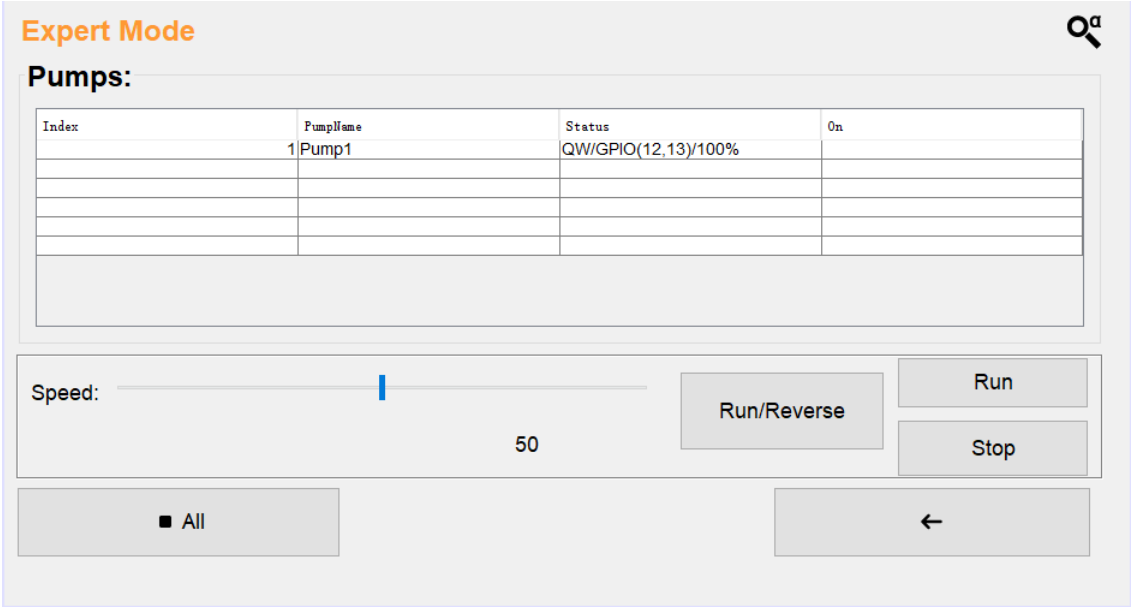


Figure 5.3: Expert mode

The action configuration scene is similar to the scene above. On the left side of the scene, a list of pumps is presented, and by clicking on the "edit" button, a user can return to the previous scene. A list of actions is presented on the right side, and by clicking "new" to add a new action, a pop-up window opens, allowing a user to configure the action with a start time and end time and choose which pump to display on the console. By clicking the "save" button, the new action is added to the list. Alternatively, a user can also add a sequence of actions by clicking the "loop" button. To delete an action, the user can either choose an action on the list and then click the "delete" button or delete all the actions together by clicking the "delete all" button.

Finally, to control pumps separately, users can enter expert mode and change the specific running speed of the selected pump. To do so, the user scrolls the bar on the corner for the corresponding pump, and this runs or reversely runs the pump at the user's command. The major function of the program, which is controlling the pump, is dictated by the script that uses the API functions to transmit the output signal to the pins connected to the pumps, as displayed in Figure 5.4

The rest of frames are composed by these 3 main part by button click or pop-up.

Here is a code snippet for set Frame properties correctly through several function call Figure 5.5

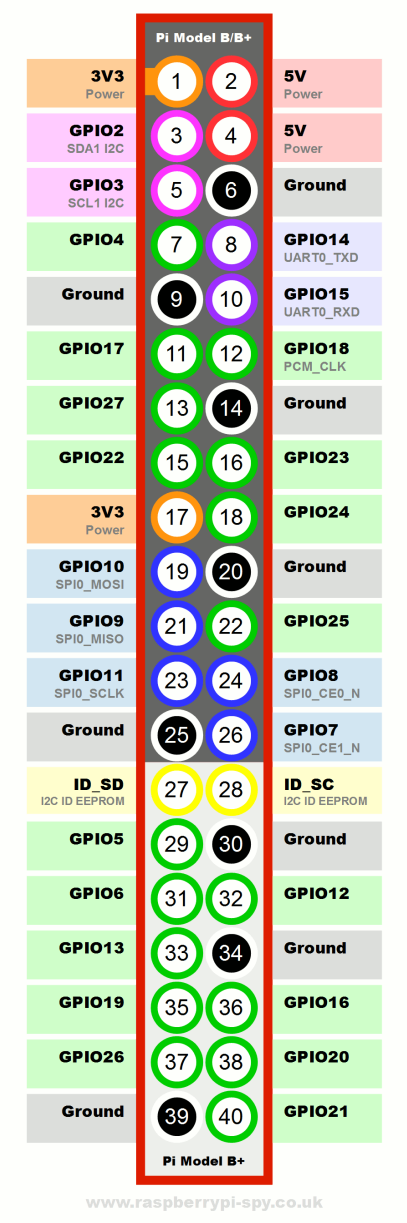


Figure 5.4: raspberry pin layout

```
PumpconfigMainFrame.setDefaultCloseOperation(javax.swing.WindowConstants.
    EXIT_ON_CLOSE);
PumpconfigMainFrame.setTitle("PumpConfig");
PumpconfigMainFrame.setAutoRequestFocus(false);
PumpconfigMainFrame.setCursor(new java.awt.Cursor(java.awt.Cursor.
    DEFAULT_CURSOR));
PumpconfigMainFrame.setFocusable(false);
PumpconfigMainFrame.setLocation(new java.awt.Point(0, 0));
PumpconfigMainFrame.setSize(new java.awt.Dimension(1100, 585));
```

Figure 5.5: PumpconfigMainFrame Setup

5.2 Defining object and properties (Model & Controller)

All functions, including show list on the table, and interactions between the backend and the UI are implemented in the mainframe-class, the Util class contains most of the utility functions that are to be used in the calculations in different classes. JSON parser functions are implemented in class Util, and they parse a configuration into a JSON string and vice versa. Furthermore, during the implementation 3 part of Object were clear and were a must-have feature in the program:*PumpType, Pump, Action*

The Unified Modeling Language(UML) class diagram in Figure 4.4 depicts the structure and classes created in this study and how they work with each other. For this project, a pump type class was created, and it stores information concerning type information and pump class. Such information is responsible for pin1 and pin2 of a pump. The action class represents the single-action object with a specific starting time and ending time, corresponding pump, and speed attribute.

In the mainframe, the type; pump; and action objects given by the UI are stored in different Array Lists, and objects can be added or deleted by manipulating different Array Lists, which is a friendly technique that enables the developer to continue developing the project.

The *pump* class contains all the information that represent each actual pump. Considering the performance and space when the amount of action get increased, a mechanism was created to guarantee the program runs in an efficient way.

The Figure 5.6 is the code snippet of *pump* class. 2 PriorityQueue Attributes are introduced as member variable of the class *pump*. Aq stands for action queue, which contains the action objects once action new action is generated. The pumps then run correspondingly with in the order of eq - execution queue. After pumps run the system transfer all action objects from aq to eq. The serialization progress locates in the

```
private PriorityQueue<Action> aq;
private PriorityQueue<Action> eq;
...
public Pump(String name, ...) {
...
    aq = new PriorityQueue<Action>(actionComparator());
}
Comparator<Action> actionComparator(){
    Comparator<Action> actionComparator = (Action a1, Action a2) -> {
        if(a1.getStartTime() - a2.getStartTime() < 0){
            return -1;
        }else if (a1.getStartTime() - a2.getStartTime() == 0){
            return 0;
        }else {
            return 1;
        }
    };
    return actionComparator;
}
```

Figure 5.6: Aqeq

constructor when pump once get created.

The specific class of configuration stores the type list, pump list, and action list and can transfer them to a JSON file or re-import them by clicking the "load" button. This transformation function are implemented in the class *Util*, here in the Figure 5.7 there's some code examples of transformation. The regeneration of configuration information was implemented strictly sub-sequenced and found from the corresponding class. A code example of finding a pump by the given name is in Figure 5.8.

Besides, there was a problem with implementing the pump when we made the first approach by following the instructions of Pi4j Git-Hub repository [27].

In the beginning, the test went fine and the pump got controlled correctly, and a bug appears later during the test. However, we founded that not all GPIO support PWM [28]. As described in the forum, so we decided to switch from the method provided by official instruction to software PWM, in which CPU of Raspberry Pi simulates the PWM process.

```
import com.google.gson.*;
import com.google.gson.reflect.TypeToken;
import com.google.gson.stream.JsonWriter;
...
public static Configuation parseJsontoConfig(String json ) throws
    Exception{

    JsonObject jsonObject = new JsonParser().parse(json).getAsJsonObject()
        ;
    String name = jsonObject.get("name").getAsString();
    ////////
    //parse type classes
    JsonArray typesjson = jsonObject.getAsJsonArray("types");
    Iterator it = typesjson.iterator();
    ArrayList<PumpType> types = new ArrayList<PumpType>();
    while(it.hasNext()){
        ...
    }
    ////////
    //parse Pump class
    JsonArray pumpsjson = jsonObject.getAsJsonArray("pumps");
    ArrayList<Pump> pumps = new ArrayList<Pump>();

    it = pumpsjson.iterator();
    while(it.hasNext()){
        ...
    }
    ////////
     //parse Action Class
    ...
```

Figure 5.7: Util class Code

```
private static int findPumpByName(ArrayList<Pump> pumpList, String str)
    throws Exception{
    if(str.equals("")|| str == null){
        System.out.println("can't␣find␣the␣pump␣the␣given␣pump␣name␣is␣
            null");
        return -1;
    }
    for(int i=0 ; i < pumpList.size();i++){
        Pump p = pumpList.get(i);
        if( p.getName().equals(str) ){
            return i;
        }
    }
    return -1;
}
```

Figure 5.8: Find pump function Code

6 Test and conclusion

6.1 Test

Setup

The test of the project started by setting up the pump and Raspberry Pi. As plotted in Figure 6.2, the project setup was generated and simulated in the online testing tool [29] from Auto Cad Company.

Next we built the project as shown in the Figure 6.3.

The final aspect of the study was to test the software through experiments.

In particular, two experiments were undertaken. For these, the project setup contains four parts:

- Wiring (black+,white- ,red+,brown- ,shown in Figure 6.1)
- Raspberry Pi
- Pump sequence
- Connection of all parts

Experiment

Once the project setup was completed, all the parts were connected. The first experiment concerned a reaction that involved mixing colors in the 3D printed chip in Figure 6.5. The performance was satisfactory, in general, and the pumps worked according to the sequence shown in the program. The second experiment was spherification Figure 6.4, in which two different chemical solutions, calcium lactate and sodium alginate, were mixed and produce the little bubbles . The results of the experiments demonstrated that the aforementioned reactions can be accomplished on the 3D printed chip. However, due to the hardware limitation, some issues must still be fixed. Thus, further development is still required. Further efforts will be undertaken in the hope of making a meaningful contribution to the field.

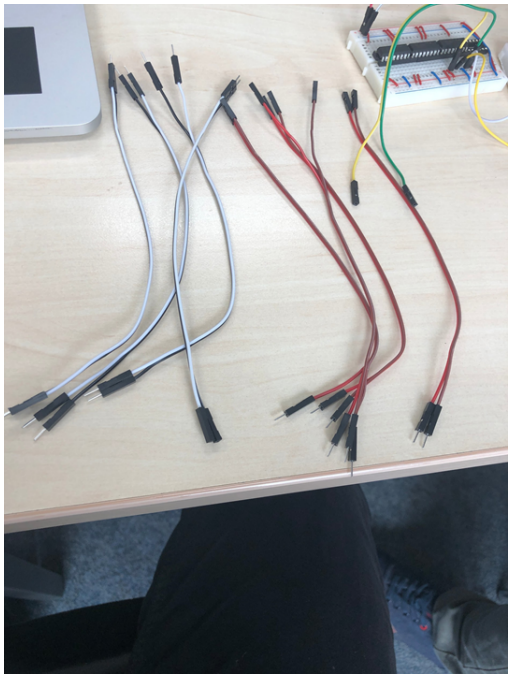


Figure 6.1: Wiring

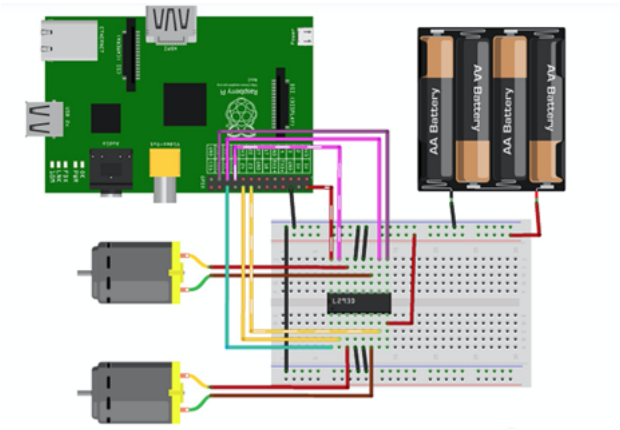


Figure 6.2: Project Setup

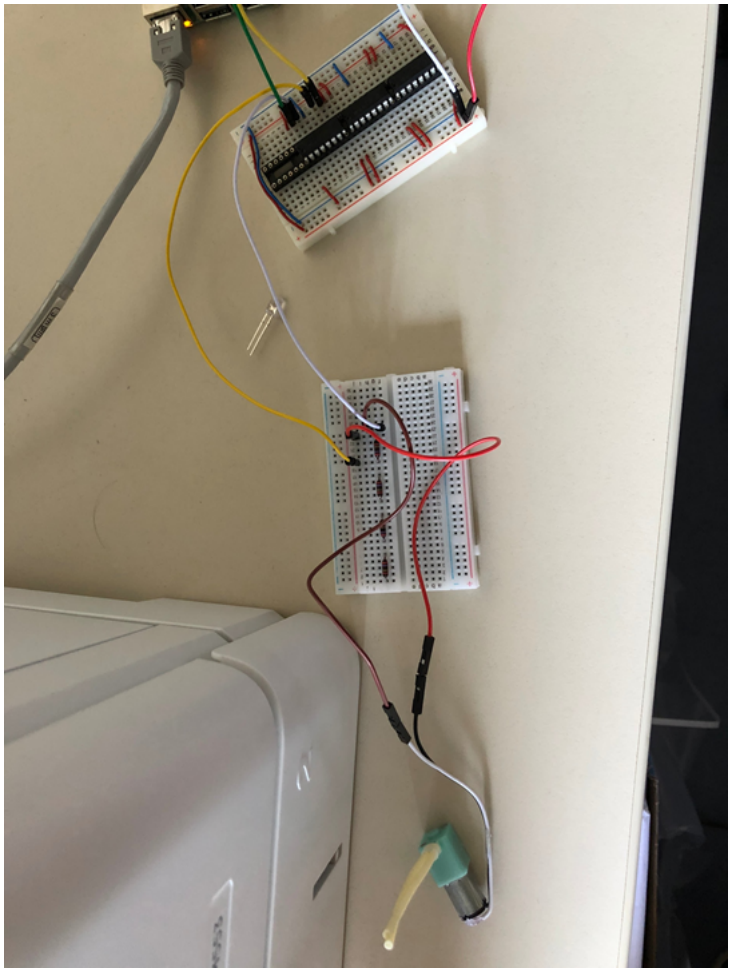


Figure 6.3: Main Setup

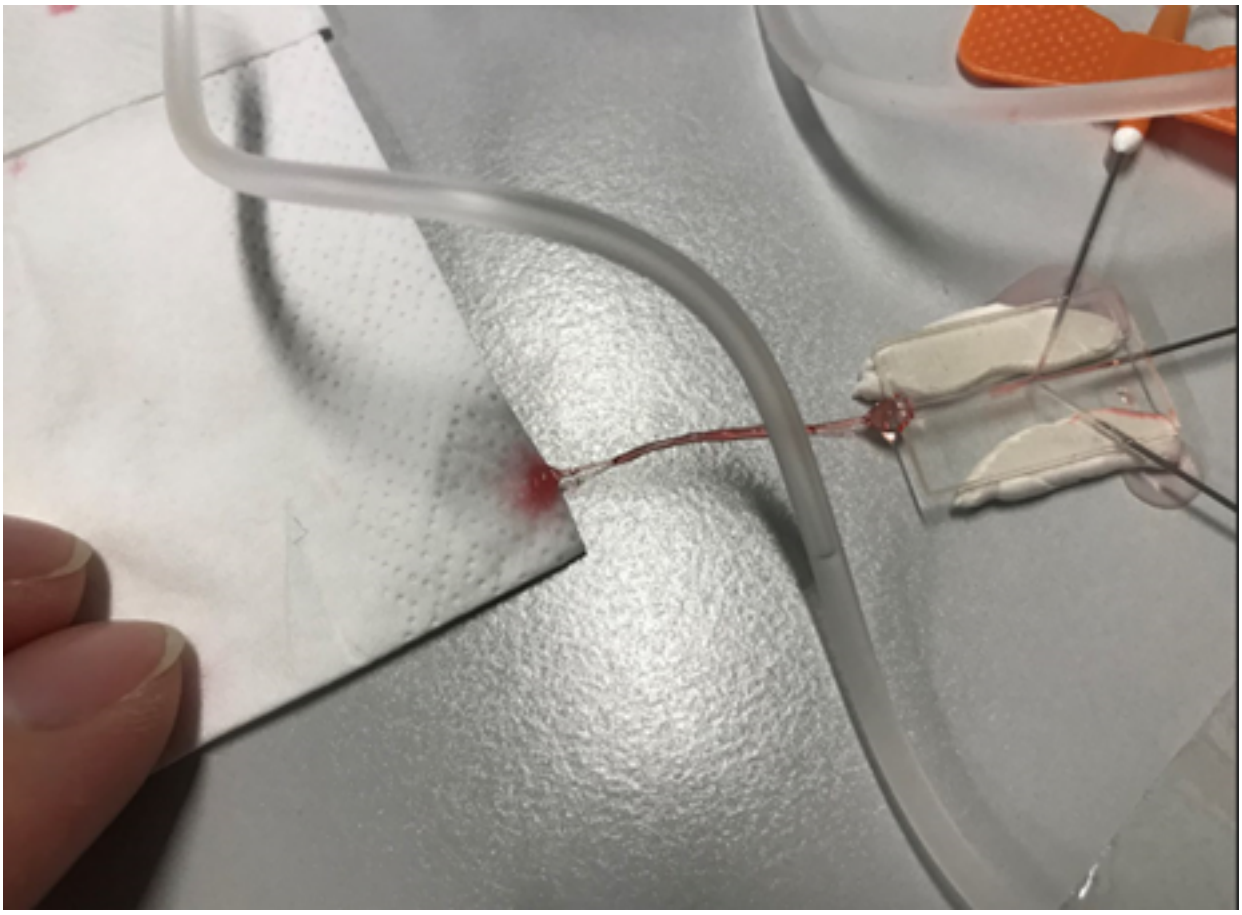


Figure 6.4: Spherification

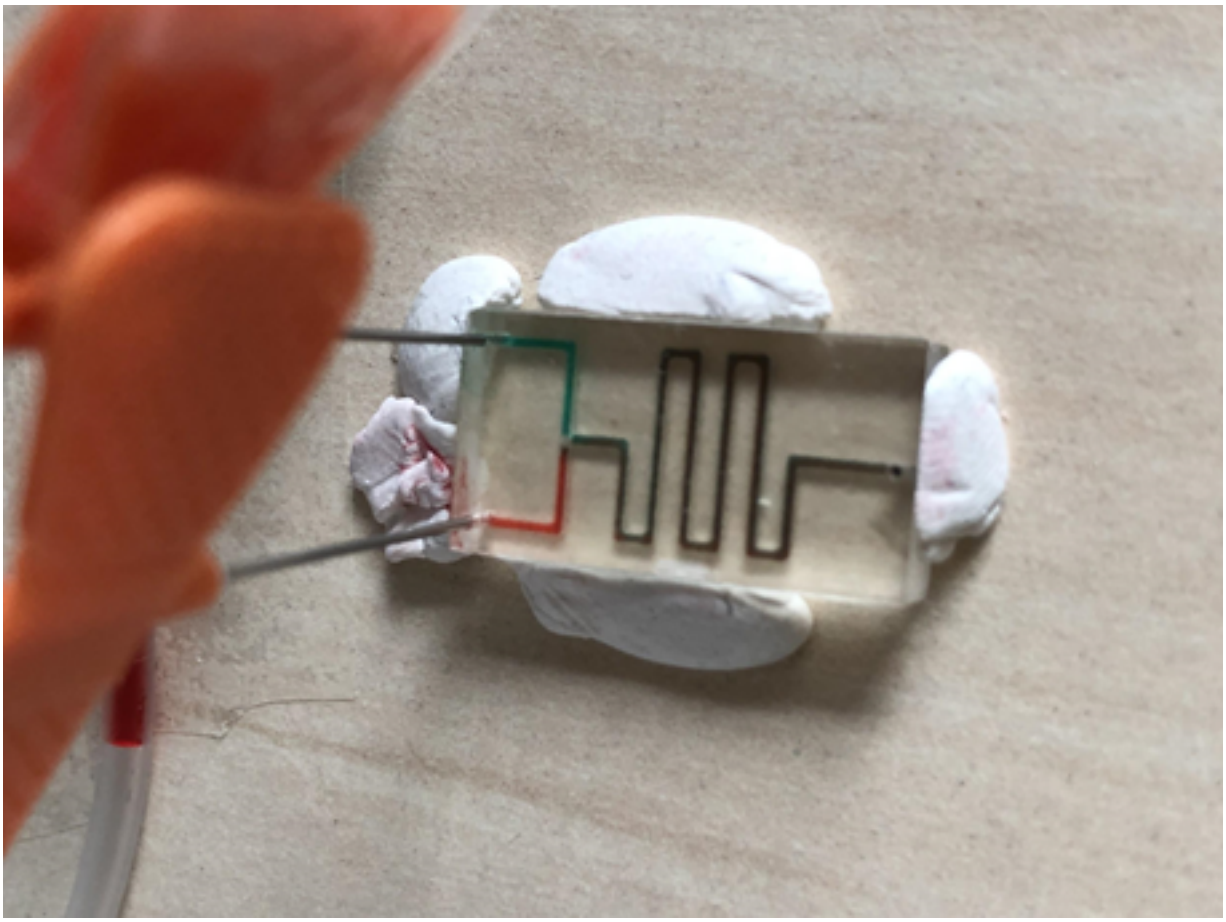


Figure 6.5: ColorMix

6.2 Conclusion

The results of the experiments demonstrated that the aforementioned reactions can be accomplished on the 3D printed chip.The problem is then solved and out prototype is consequently finished.

However, due to the hardware limitation, in some cases the pump didn't run as our wish. The reason why fluids coming out irregularly is that as aforementioned in the chapter1.2 and the Figure 2.2 the fluids will only be driven if the rings go through middle line and press the pipe. Thus, development in the future is still required. Further efforts will be undertaken in the hope of making a meaningful contribution to the field.

Finally, in the extensively researched area of Lab-On-Chip, scientists and researchers have engaged in various types of research to achieve portability and low cost.

Bibliography

1. Castillo-Leon, J., Svendsen, W., Dimaki, M., Arima, V., Akram, M., Miserere, S., Neumann, C. & Kipling, G. *Lab-on-a-Chip Devices and Micro-Total Analysis Systems A Practical Guide* ISBN: 978-3-319-08686-6. doi:10.1007/978-3-319-08687-3 (Nov. 2015).
2. Wikipedia. *Gas chromatography — Wikipedia, The Free Encyclopedia* [Online; accessed 10-December-2019].
3. Wikipedia. *Microfabrication — Wikipedia, The Free Encyclopedia* [Online; accessed 11-December-2019].
4. Wikipedia. *Microfluidics — Wikipedia, The Free Encyclopedia* [Online; accessed 10-December-2019].
5. Seemann, R., Brinkmann, M., Pfohl, T. & Herminghaus, S. Droplet based microfluidics. *Reports on progress in physics* **75,** 016601 (2011).
6. *Siemens Website* https://new.siemens.com/global/en.html.
7. Sharlin, H. I. *William Thomson, Baron Kelvin* Nov. 2019.
8. Harrison, D., Fluri, K., Seiler, K., Fan, Z., Effenhauser, C. & Manz, A. Micromachining a miniaturized capillary electrophoresis-based chemical analysis system on a chip. *Science (New York, NY)* **261,** 895–897 (1993).
9. Theeuwes, F. *Electroosmotic pump and fluid dispenser including same* US Patent 3,923,426. Dec. 1975.
10. Van den Berg, A., Olthuis, W. & Bergveld, P. *Micro Total Analysis Systems 2000: Proceedings of the μTAS 2000 Symposium, held in Enschede, The Netherlands, 14–18 May 2000* (Springer Science & Business Media, 2013).
11. *Chemical and Biological Microsystems Society (CBMS)* https://cbmsociety.org/ (2019).
12. Duffy, D. C., McDonald, J. C., Schueller, O. J. & Whitesides, G. M. Rapid prototyping of microfluidic systems in poly (dimethylsiloxane). *Analytical chemistry* **70,** 4974–4984 (1998).

13. Pollack, M. G., Fair, R. B. & Shenderov, A. D. Electrowetting-based actuation of liquid droplets for microfluidic applications. *Applied Physics Letters* **77,** 1725–1726 (2000).

14. Wikipedia. *Chuck Hull — Wikipedia, The Free Encyclopedia* [Online; accessed 11-December-2019].

15. *FIVE SHORT STORIES ON THE HISTORY OF MICROFLUIDICS* https://www.elveflow.com/microfluidic-tutorials/microfluidic-reviews-and-tutorials/history-of-microfluidics/ (2019).

16. *Transfer pump RP-Q1 Series RP-Q1.2N-P20A-DC3V* https://www.takasago-fluidics.com/products_pump/transfer/RP-Q1/RP-Q1.2N-P20A-DC3V_15.html.

17. *Model RP-Q Series Miniature Peristaltic Pump* https://www.clarksol.com/wp-content/uploads/2019/07/RPQ1.pdf.

18. Wikipedia. *Pulse-width modulation — Wikipedia, The Free Encyclopedia* http://en.wikipedia.org/w/index.php?title=Pulse-width%20modulation&oldid=925657978. [Online; accessed 30-November-2019]. 2019.

19. KerryKerry, U. & kev. *Control Hardware PWM frequency* https://raspberrypi.stackexchange.com/questions/4906/control-hardware-pwm-frequency.

20. *L293D* https://www.st.com/zh/motor-drivers/l293d.html.

21. *WiringPi* http://wiringpi.com/.

22. Savage, R. *Pi4J* https://pi4j.com/1.2/index.html.

23. Pi4J. *Pi4J/pi4j* https://github.com/Pi4J/pi4j/.

24. *Robert Plutchik* https://en.wikipedia.org/wiki/Robert_Plutchik.

25. *Model–view–controller* https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller.

26. *Swing (Java)* https://en.wikipedia.org/wiki/Swing_(Java).

27. Pi4J. *Pi4J/pi4j* https://github.com/Pi4J/pi4j/blob/master/pi4j-example/src/main/java/PwmExample.java.

28. Stianeikeland. *PWM no work on all gpio pins? · Issue # 20 · stianeikeland/go-rpio* https://github.com/stianeikeland/go-rpio/issues/20.

29. *3D design thinkcad* https://www.tinkercad.com/things/dyEDELFcKWM-thinkcad.