VIETNAM NATIONAL UNIVERSITY, HANOI

UNIVERSITY OF ENGINEERING AND TECHNOLOGY

STUDENT RESEARCH CONTEST

2020 - 2021

AN AUTOMATIC MICROINJECTION SYRINGE PUMP FOR BIOMEDICAL APPLICATIONS.

Nguyen Dinh Duong QH-2018-I/CQ-R

Nguyen Phuc Vinh QH-2019-I/CQ-R

Tong Tran Hoang QH-2017-I/CQ-ĐACLC

Research advisor: Dr. Nguyen Ngoc An

Post-grad student. Tran Thanh Hang

HA NOI - 2021

Abstract

In this day and age, robotics and devices are being put to use in a variety of fields, and one the most prominent is in medicine. Enhancement in accuracy, speed of operation and cost reduction are the primary motivations to design and manufacture of medical devices or robots. Based on that premise, this paper presents the design, construction, and evaluation of an automotive microfluidic pump for drug delivery applications. This device is used to infuse drugs into patients who are unable to receive them orally, at a controlled, precise volume and rate set by medical practitioners. By referencing a device named "Dual Syringe Pump", we have built a similar one with some additional functions (3 pumps instead of 2 on the original; a raspberrypi-controlled interface that comes along with a wifi module, which opens up the possibility of controlling and monitoring the device remotely using our in-house developed smartphone application), studying the advantages and disadvantages between the original and our own creation. The most important items achieved in these investigations include time, accuracy, speed of drug injection, and cost reduction. Some of the features on the original pump are quite useful in real-world usage (such as withdraw-infuse and vice versa in the same cycle), so we decided to also develop and embed those same functionalities into our system.

Keywords: Microfluidic pump; drug delivery; raspberry Pi; smartphone application.

Table of Contents

List of figures:

1. Introduction

Microfluidic pumps (or syringe injection pumps) are used for the slow injection of drugs. They are also used for the injection of blood components, such as plasma. The pump monitors the fluid pressure rate, resulting in fluid pressure control. This avoids excessive injection pressure on the patient's vein and possible pain at the injection site.

The first attempt at manufacturing an intravenous therapy device was conducted in 1492. Major developments occurred in 1970, when Dean Came invented the first ambulatory injection pump. These pumps are the size of a cassette player and are attached around the patient's waist. Drug delivery pathways for these pumps are intravenous, subcutaneous, epidural, internal, and in the spinal cord. This device is also used for morphine injection and other strong painkillers to control intensive pain and chronic cancers.

Another application for microfluidic pumps is for the purpose of producing and mixing drugs for the treatment of different types of diseases. For example, in chemotherapy cancer treatment, micro-sized droplets, encapsulated by a thin film of polymer, are injected into the patient’s vein, and follow the bloodstream to where the tumor is located. The process to produce this type of drug can be done effectively using microfluidic pumps.

The most common approach to building this type of microfluidic delivery system is by using a stepper motor that applies a continuous and steady force onto a syringe, at an accurate and controlled flow rate, as well as being more precise in terms of the target volume the user wants to dispense or withdraw.

In this paper, we propose a microfluid delivery system that is built similarly to the above-mentioned method But this is where we decide to deviate from the prototype to make ours truly original. IoT – or Internet of Things – is growing at a rapid pace, almost everything technology-related these days has an IoT element in it. Embracing that trend, we have integrated into our system the ability to control the pump and monitor the progress remotely, using a tool that is so ubiquitous, it fits in the palm of your hand: the smartphone.

The details of our progression into building this device, technical terms, and method will be discussed furthermore all the sections below.

2. Mechanical design

In this project, design and construction were carried out in three parts. The first is about the mechanical design of the machine. This section describes the structure and mechanical parts of the machine. We renovate the design. The actuator, engine and control system of the machine will be completely inside the machine, only a few blocks are visible from the outside. Viewers can only see blocks outside the enclosure thanks to the controller being inside the machine.

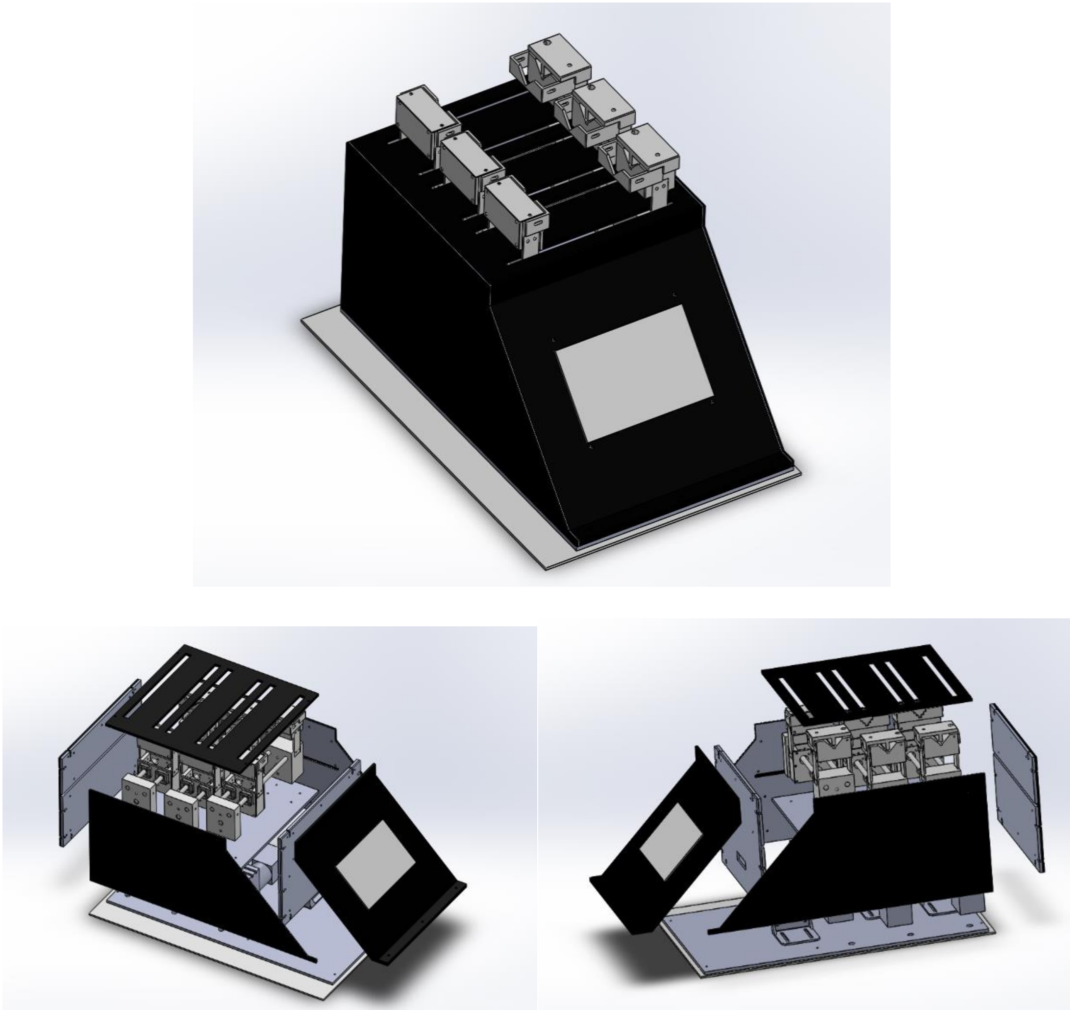


Figure 1: SolidWork of the model prototype machine

Machine specification: 482x282x211 (mm)

2.1. Lead screw actuator of the machine

For the purpose of running micropump, the flow is very small, accurate, slow speed, we use lead screw of 8mm shaft diameter, 2mm thread pitch. The structure uses the belt to drive. The motor shaft end and lead screw bar will be each fitted with a pulley. The belt is attached to the pulleys at the top of the motor shaft and lead screw. When the engine is rotating, the drive belts, the mechanical parts on the lead screw bar rotates in synchronized order.

280mm GT2 belt

Lead screw T8 step2,nut

GT2 pulley 20 teeth

Figure 2: Mechanical components

2.2. Controller components

For the purposes of running very small, precise, slow-flow micro-pumps, we use a stepper motor with gearbox with a gear ratio of 1:49, Arduino Mega controller, motor control circuit: TB6560 Stepper Motor Driver

GPLL40-49 MOTOR

Arduino Mega

TB6560 Stepper Motor Driver

Figure 3: controller components

2.3. Micro-pump movement system

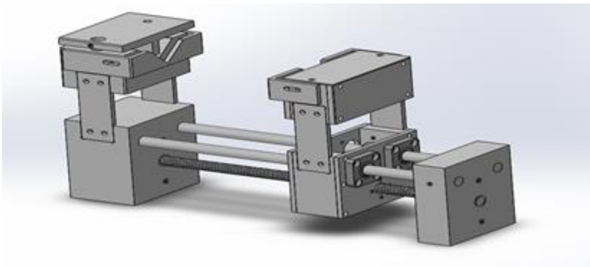


Figure 4: Movement system

In this device, we use three movement systems that work together. System includes mechanical parts:

2.3.1. **Exterior of the housing:**

- Block for cylinder body:

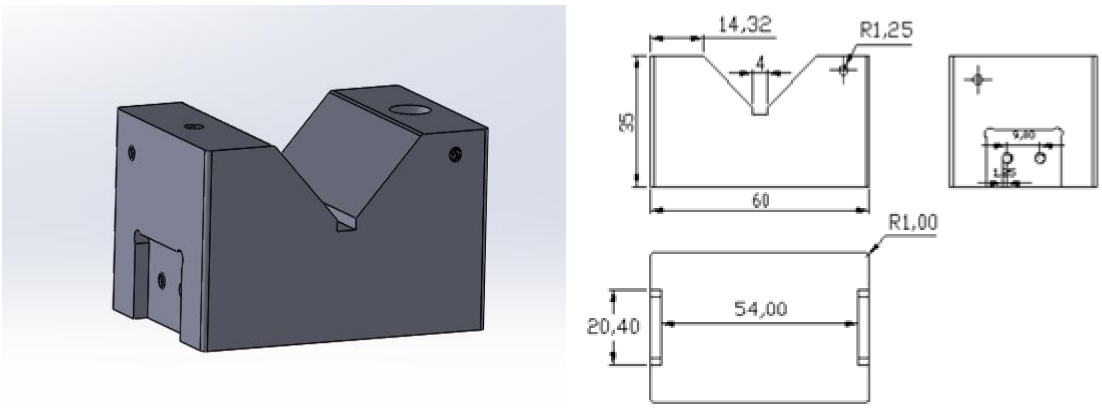


Figure 5: Syringe tube block

- The block is designed with a groove so that can accommodate any size of syringe tube.

- Syringe fixing block:

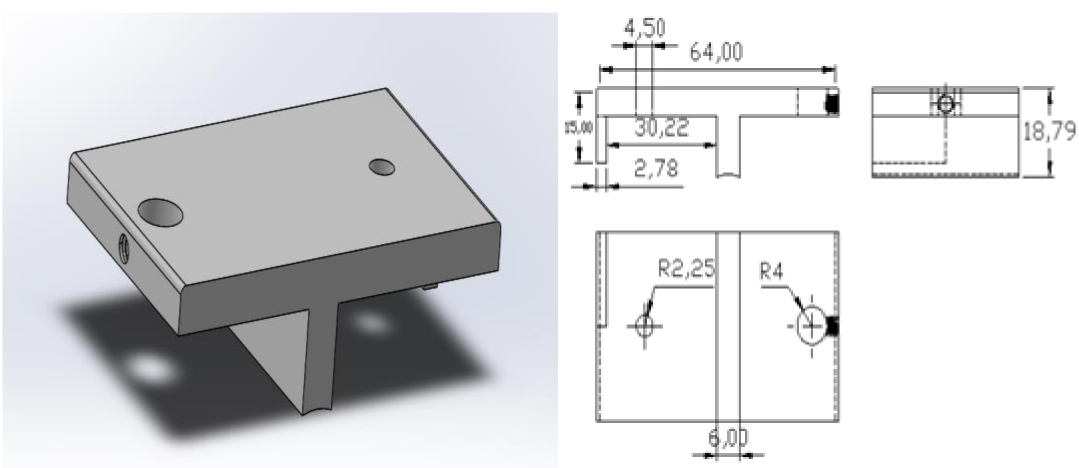


Figure 6: Syringe fixing block

- Threaded block to fix the cylinder body on the block to the cylinder body, ensuring the cylinder body does not move during work.

- Syringe flange holder:

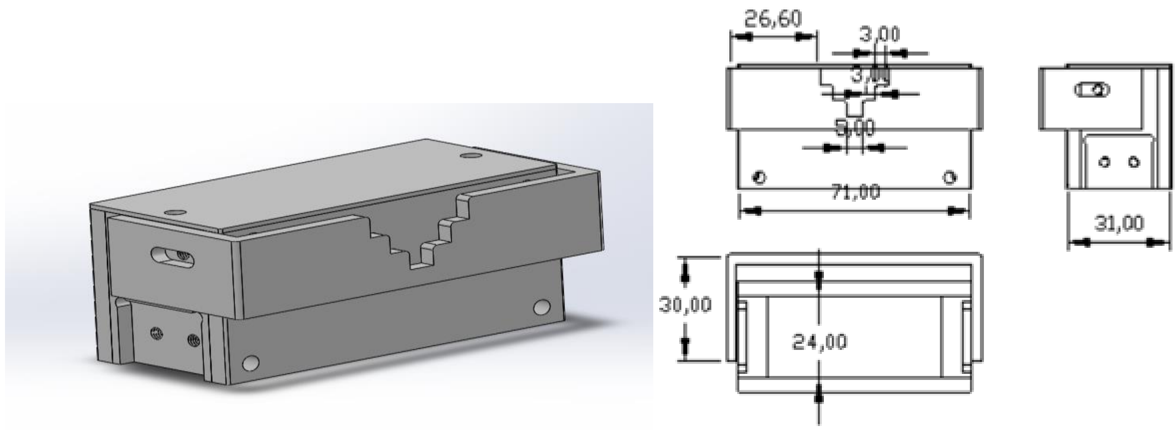


Figure 7: Syringe flange holder

- A holder is mounted on the block, which can be pulled in and out to accommodate the dimensions of many other cylinders.

2.3.2. Interior of the housing:

- Bridge of the interior and exterior elements:

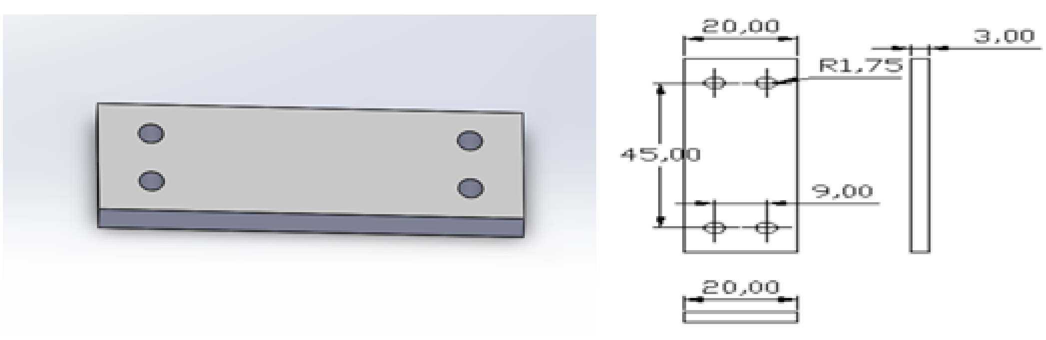


Figure 8: Bridge of the interior and exterior elements

- We use these 4 blocks. Its task is to fix the upper blocks of the machine, ensuring that when the machine is operating the blocks do not shake.

- Interior moving elements:

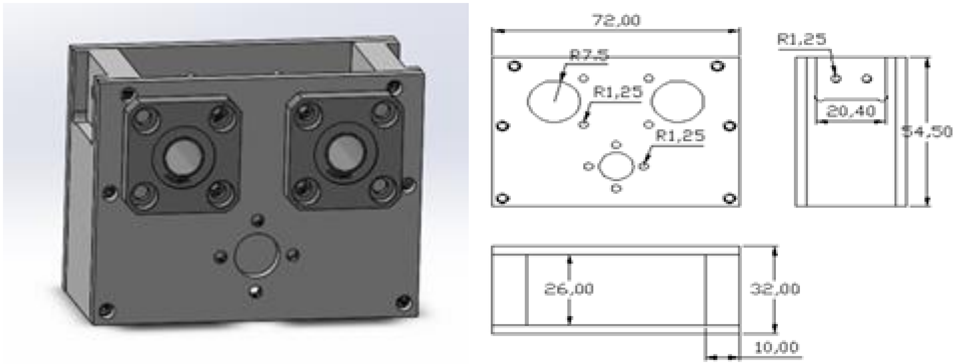


Figure 9: Interior moving elements

- This unit moves inside the machine making the outside block move. Shaft bearing LMK8UU be used to block that can slide on the slide.

Figure 10: Shaft Bearing LMK8UU

2.4. Layout of elements inside the machine

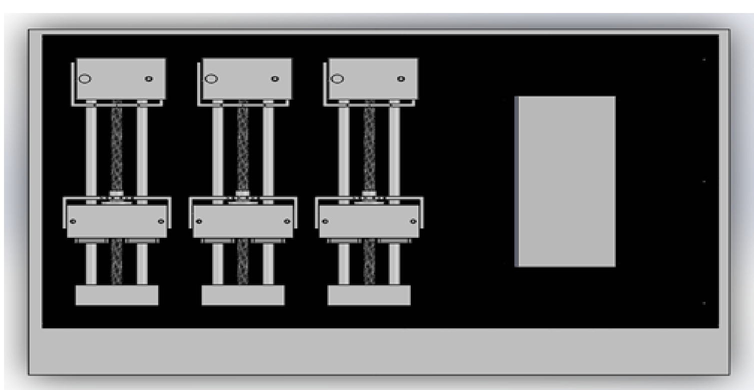


Figure 11: Top view of the exterior

The top part of the machine is a place to put the three moving systems. The interior part of the machine is where we put the motors, controllers and power supplies.

In order for the motor to not vibrate during operation, a 3D-printed holder is needed to fix the motor to the floor of the machine.

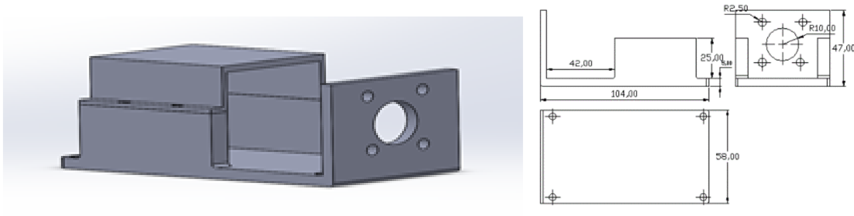


Figure 12: SolidWork model of the motor holder

3. System design

3.1. Fluidics

The principle of operation of the micro-fluid

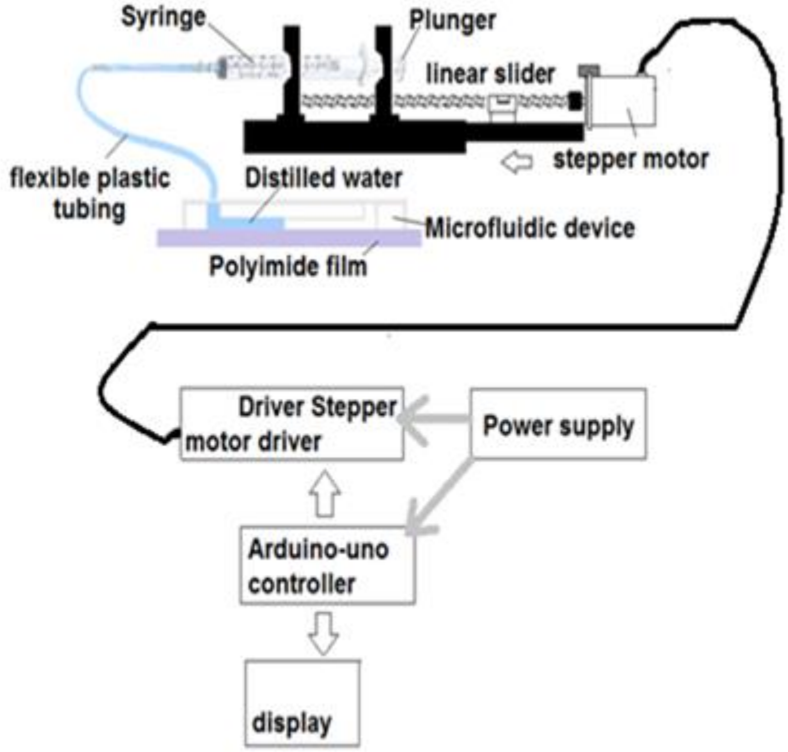


Figure 13: Working principle diagram

After being powered up, the stepper motor driver is controlled from the Arduino Uno microcontroller. The driver provides a pulse signal and direction for the stepper motor to operate. Active stepper motor transmits rotation to the lead screw. Through the screw and slider, the rotation of the stepper motor is converted to the translational motion of the piston. The slow translational piston pushes the syringe solution out in a very small flow.

Due to the need for slow speed, small flow, and high precision pumps, a stepper motor that can meet the above needs should be selected. It is very important to choose a motor that can be a micro-stepper and has a large reducer to match the requirements. The stepper motor STL4118L1804 – KGR01 meets the above requirements: with 1/16 micro stepping mode, 1:49 gear ratio, large motor torque.

Figure 14: Stepper motor used in this device

In order to be able to supply the pulse and direction signals to the motor, an additional driver is needed to convert the electrical signal into a control pulse for the motor. To meet that requirement, the TB6560 Stepper Motor Driver is a suitable choice because it can be combined with the Arduino Uno microcontroller.

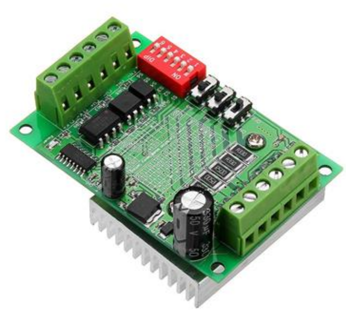


Figure 15: Driver for the motor

TB6560 motor control circuit Stepper Motor Driver can easily adjust motor control modes (full step, half step, micro-step) with the maximum micro-step mode is 1:16; input currents; stopping current; excitation Mode; delay setting through adjusting the switches.

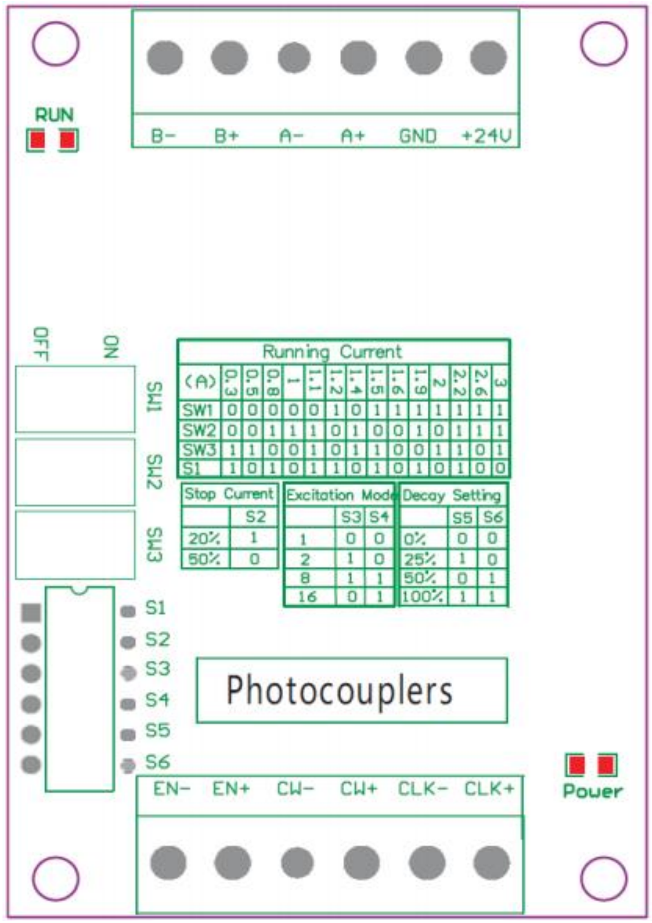


Figure 16: TB6560 port diagram

The table of connection wire:

Terminal	Connection
+24V	Power: Positive
GND	Ground
A+, A-	MOTOR COIL: Phrase A
B+, B-	MOTOR COIL: Phrase B

CLK+, CLK-	Clock for Step Speed
CW+, CW-	Direction
EN+, EN-	High Enable, Low Disable

Table 1: Connection

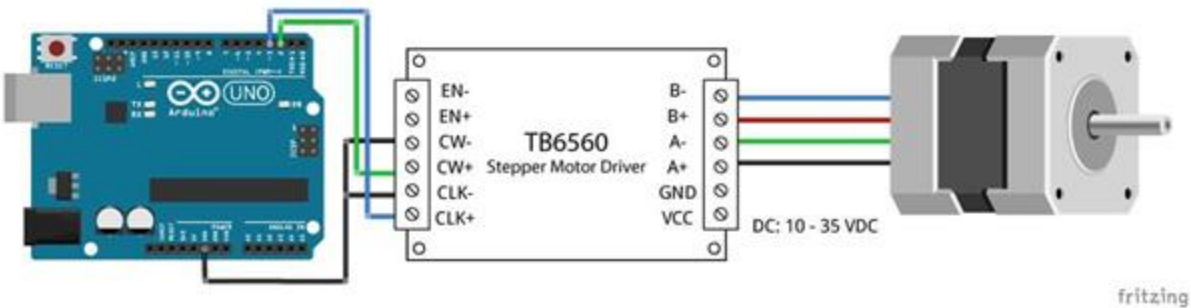


Figure 17: Wiring diagram for Arduino - Drive- Motor

2.2. Controller.

2.2.1. Principle diagram

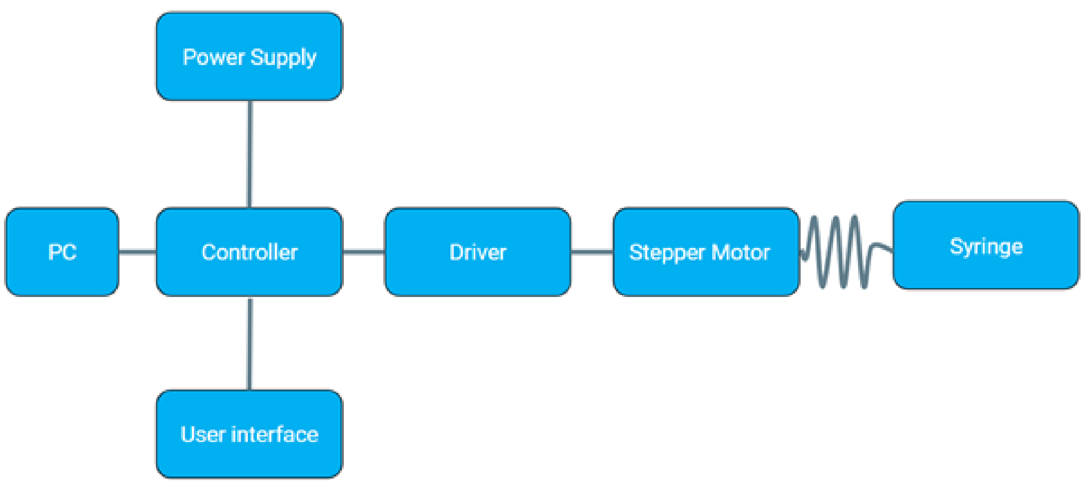


Figure 18: Diagram of the working principle

This diagram explains the working principle of the entire system. The PC sends the command to the Controller. The Controller sends the control signal to the stepper motor via the Driver. The motor then rotates the Syringe.

2.2.2. Components of the system

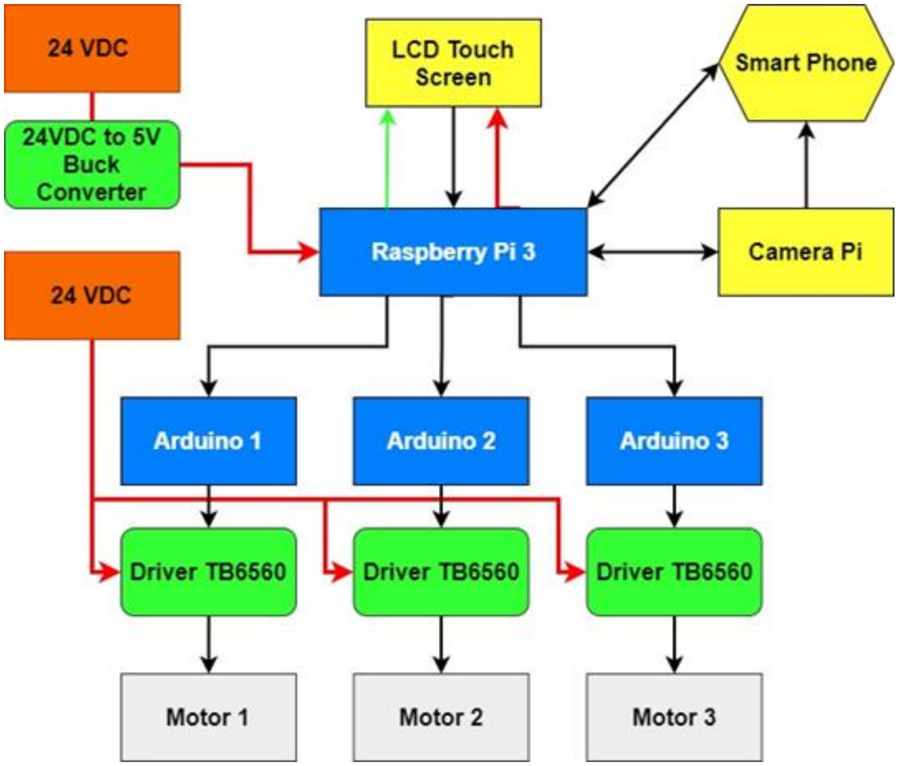


Figure 19: System deployment

Blue block: information processing components.

Green Block: electronic circuits.

Orange block: power supply.

Yellow block: electronic components.

Red line: power.

Black line: signal transmission line.

Green line: HDMI to HDMI connection.

2.2.3. Algorithm diagram

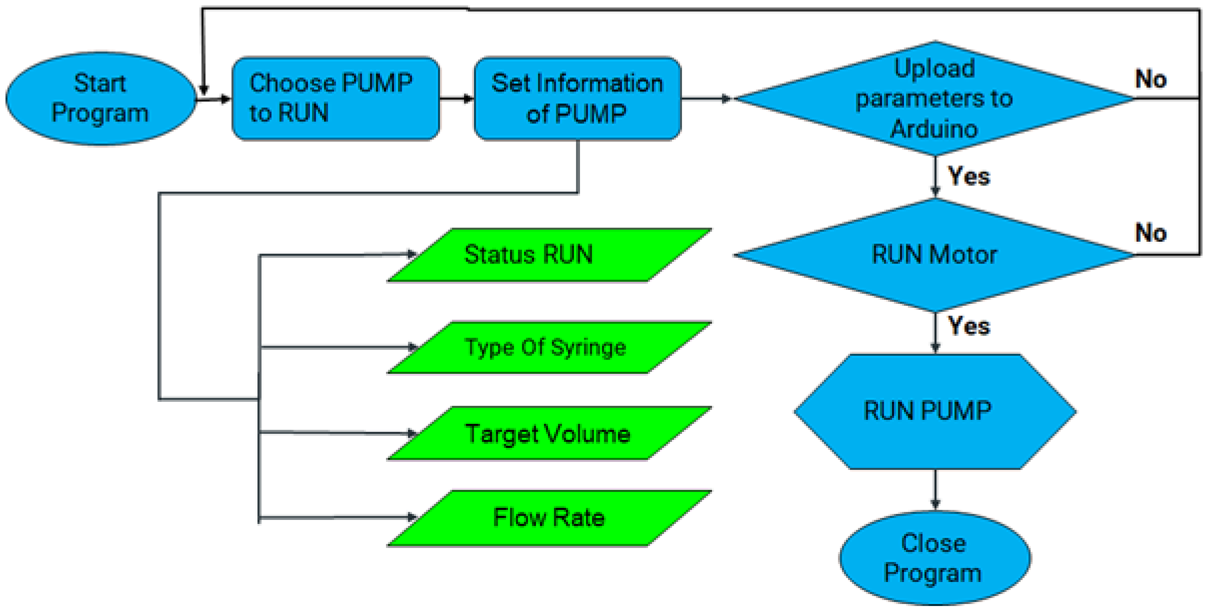


Figure 20: Algorithm diagram

1. Start the program by supplying the power to the machine. Set up all the conditions needed to operate.

2. Choose the Pump to run. Make sure that the pump already has a syringe attached.

3. Set information of pump by using the LCD touch screen. The user can set parallel the optional parameters for this pump that they want. These parameters include: status run for each pump, the type of syringe the user wants, the user's desired target volume, and the flowRate to push the solution.

4. After setting the parameters on the touch screen, the user continues to press the Set button to transmit the parameters to the Arduino. If the parameter set is successful, the user continues to press the button to run Motor. If the uploading process fails, go back to step 2 and do it again.

5. If the motor runs, the user will wait for the pump to finish its cycle according to the pre-set parameters. If the motor does not run, redo everything starting from step 2.

6. After the pump is running and finished transferring the small amount of fluid desired by the user, the pump's operation is terminated.

2.2.4. Controller block diagram

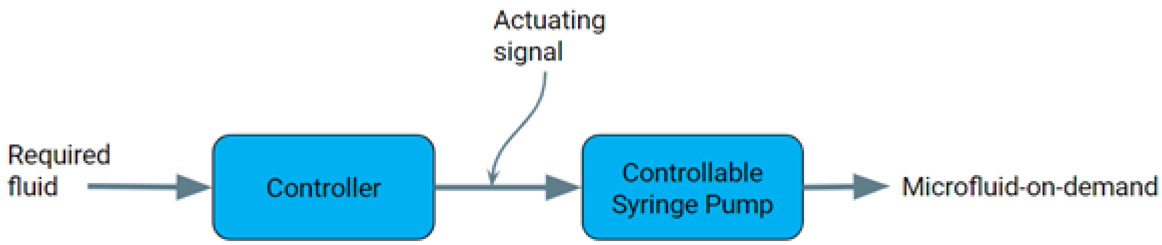


Figure 21: Block diagram of the control.

The figure above shows the control block diagram of the micro-injection syringe pump. First, the input of the system is the desired amount of solution. The required fluid is the system input that is sent to the controller. Here, the controller processes information and transmits control signals to the micro-injection syringe pump to operate. After the pump receives the signal and controls the controllable syringe pump. The output of that system is the microfluidic-on-demand.

2.2.5. Software

2.2.5.1. The user interface touchscreen.

The figure below shows the information about the touch screen user interface. This interface is designed by Qt design software and in the Python programming language. The interface is where the user communicates with the control system of the micro-injection syringe pump. The user selects the parameters and settings according to the desired output needs.

Figure 22: Interface layout.

Figure 23 shows the information about the pump running modes. This machine has 4 status of running mode for the user to pick. All of the status run modes of the pump are : Infusion ONLY, Withdraw ONLY, Infusion/Withdraw (Infusion before and Withdraw after immediately), Withdraw/Infusion(Withdraw before and Infusion after immediately).

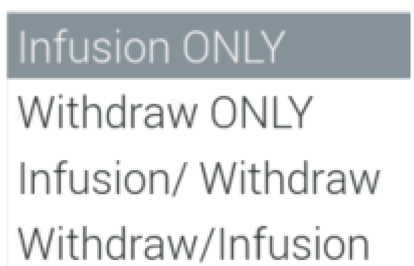


Figure 23: Running modes menu

Figure 24 displays the information about the type of the volume syringe that the user chooses to use in the machine. The system can be suitable for a variety of syringes with different volumes. The syringes used for testing are needles of Vinahankook company with different volumes 1ml, 3ml, 5ml, 10ml, 20ml, 50ml respectively. The pump is designed to accommodate the syringe with a maximum volume of 50ml.

Figure 24: Syringe’s volume options

The figure below displays the different target volume that the user can choose to run. The output volume can be chosen with a small volume equivalent from 100µl to 1000µl.

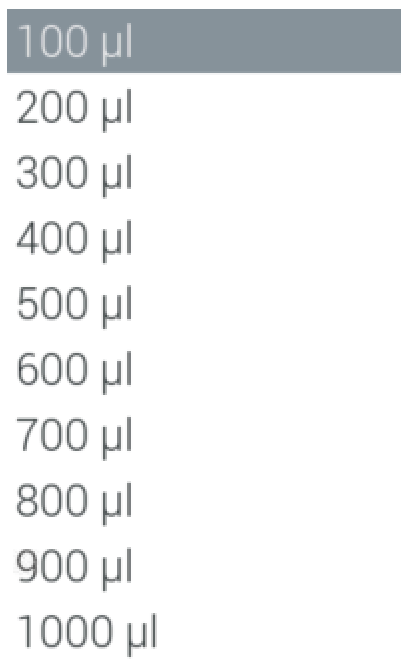


Figure 25: Target volume options

The picture below shows information of the different flow rate of this pump. The minimum and the maximum of the flow rate for the user to choose are 10 µl/min and 200 µl/min respectively.

Figure 26: Different flow rate options

4. MFPC application.

4.1. Android app block diagram.

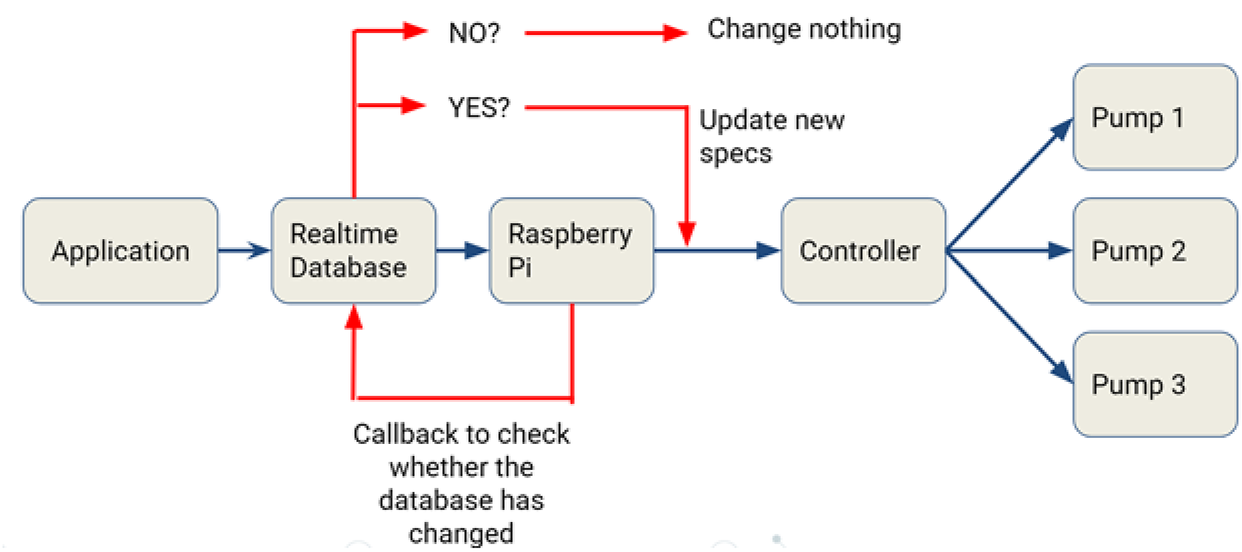


Figure 27: Android block diagram

The block diagram shown above is a representation of processes happening when the user issues a command to the system.

The Raspberry Pi receives the information issued by the application via Firebase Realtime Database. The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with their iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

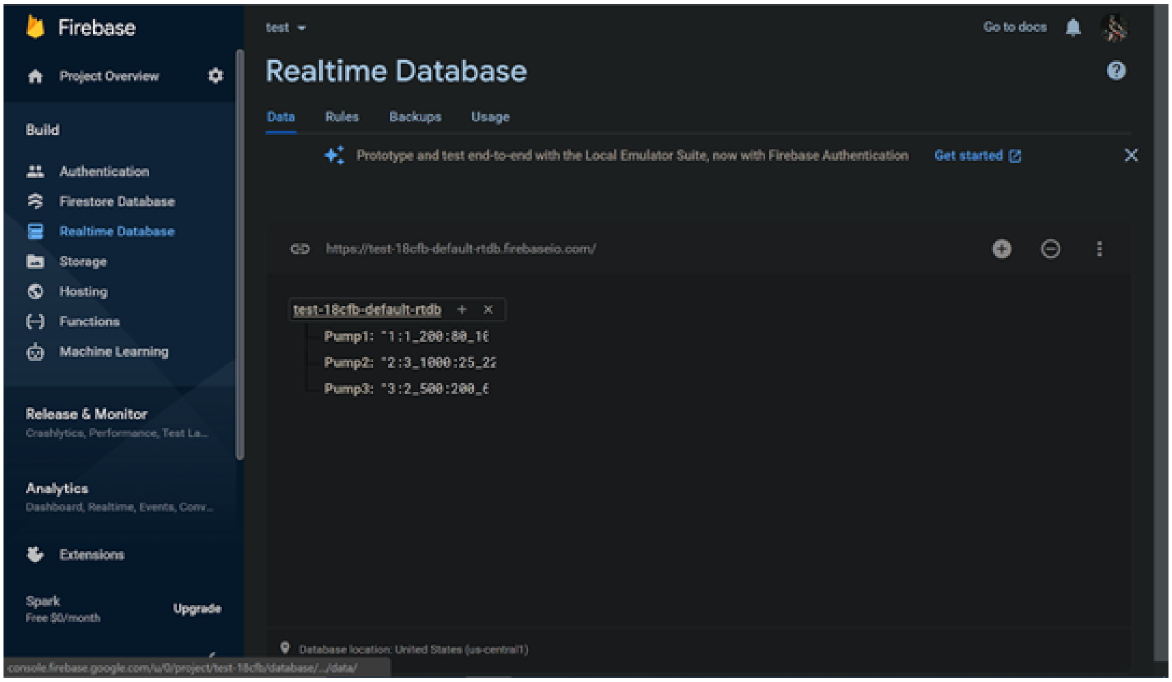


Figure 28: Layout of the Realtime Database

Every time the user presses start on the application, the specifications are automatically sent to the database.

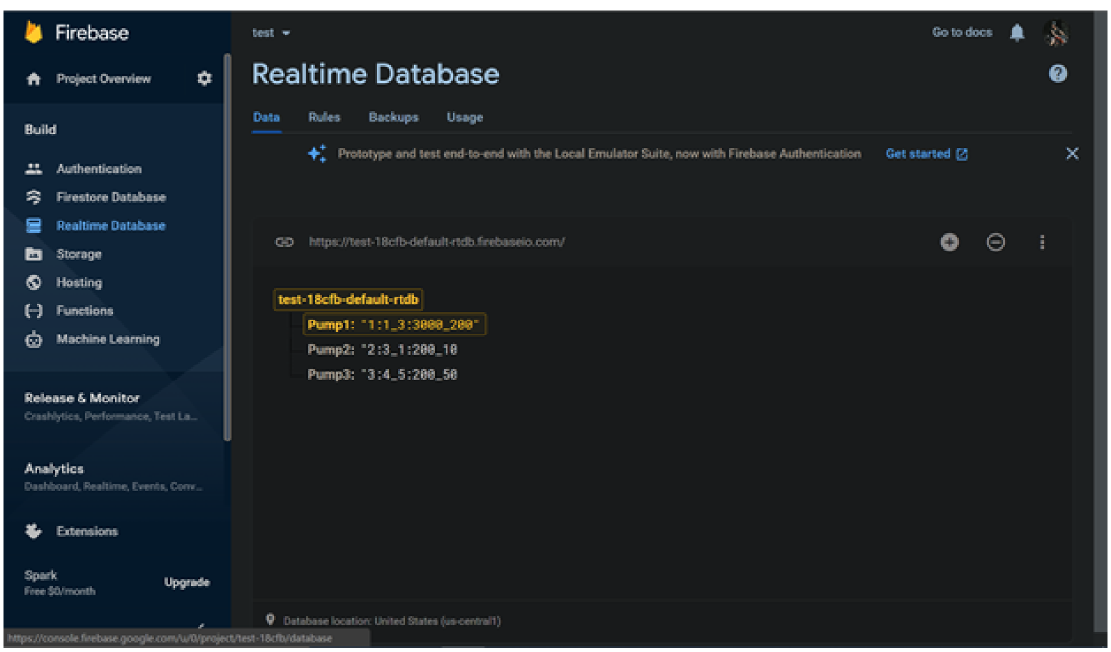


Figure 29: Realtime Database receives new specification for Pump 1.

When the database has been updated, the Raspberry Pi, at the same time, has been listening for changes. When a specification change happens in the database, the Rasp-Pi will instantly detect and pull that new specs down.

After the new specification has been read by the Pi, the same operation occurs. The Pi sends that data to the Arduino, the Arduino then decodes the new data, figures out which pump to send the information to, and tells the driver corresponding to that pump to run the stepper motor accordingly to the new specification.

4.2. Application Interface Design.

Figure 30:Application Interface

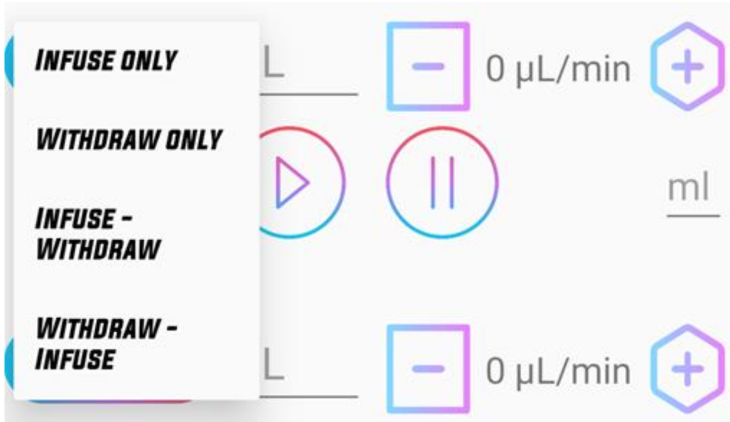


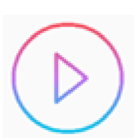
Figure 31: Dropdown menu (Spinner) to select modes

This button lets the users pick different Modes for each pump. When touched, the drop-down menu will appear, shown in Figure 2.2.

This section lets the users manually input the target volume they want to infuse or withdraw

This section lets the users set the desired flow rate by using the "+" sign (increment), or "-" sign (decrement). The value of their choice will be displayed between the two buttons in the format "X µL/min". The value can be increased or decreased one at a time if the users tap on the corresponding button. But if they tap and hold on any button, the value will increase or decrease continuously until the users lift their finger off the button.

This area is where the users will input the volume of the syringe being used.


This is the START button. After the users have set all required specifications, they will press this button, and the information will be sent to the Realtime Database, where the existing data will be replaced by the new data.

5. Experimental result

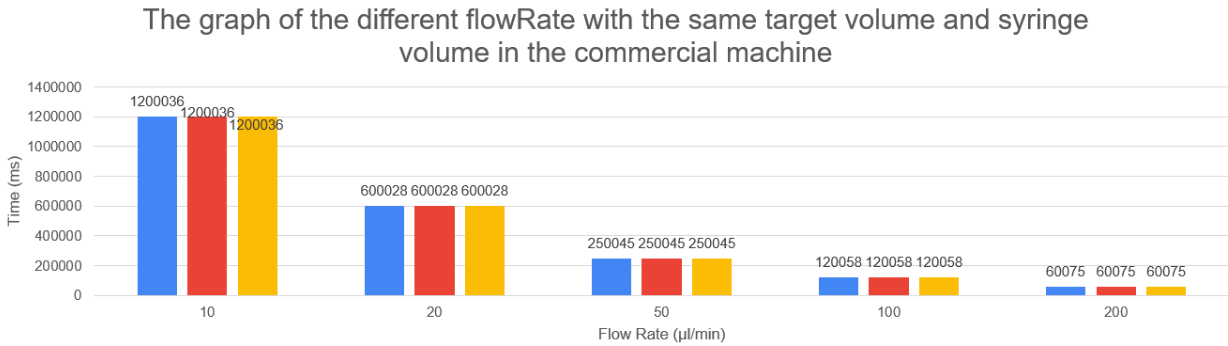


Figure 32: Graph showing the time taken to pump at fixed target volume and syringe volume at different flow rate in the commercial machine.

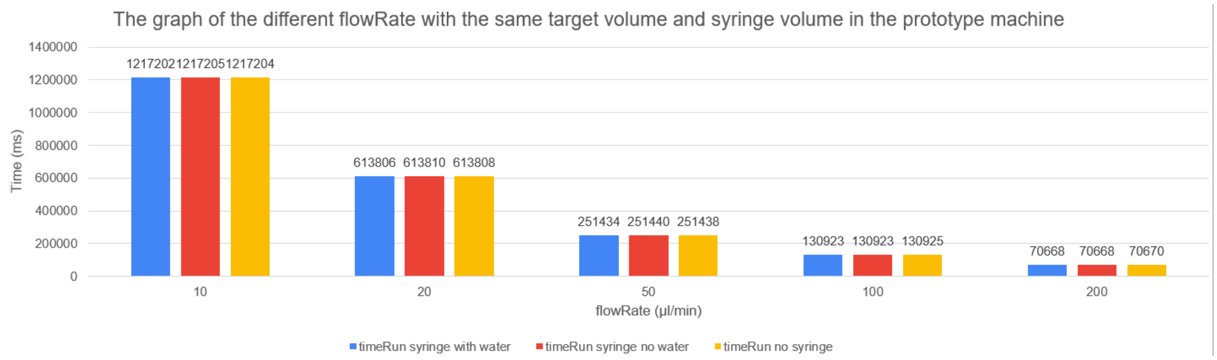


Figure 33: Graph showing the time taken to pump at fixed target volume and syringe volume at different flow rate in the prototype machine.

These graphs are the measurement comparing the commercial machine to our prototype machine. Both machines were set at the same specification: Syringe Volume is 1ml, Target Volume is $200\mu l$. We want to measure the time both machines take to deliver the same volume of the same syringe at different flow rate (10, 20, 50, 100, 200). As the chart shows, our prototype performs similarly to the commercial machine with little disparity. The Δt (difference in time) between the commercial machine and our prototype ranges from a maximum of 17 seconds at $10\mu l$/min to 600 milli-seconds at $200\mu l$/min.

6. Conclusion

In this paper, we have proposed an in-house developed microfluid delivery system for a variety of application in the biomedical field, based on existing commercially available device. And based on what we have learnt about to commercial machine, we have enhanced the originality of our prototype with additional features (3-system; control the system via the internet with our own developed Android Application).

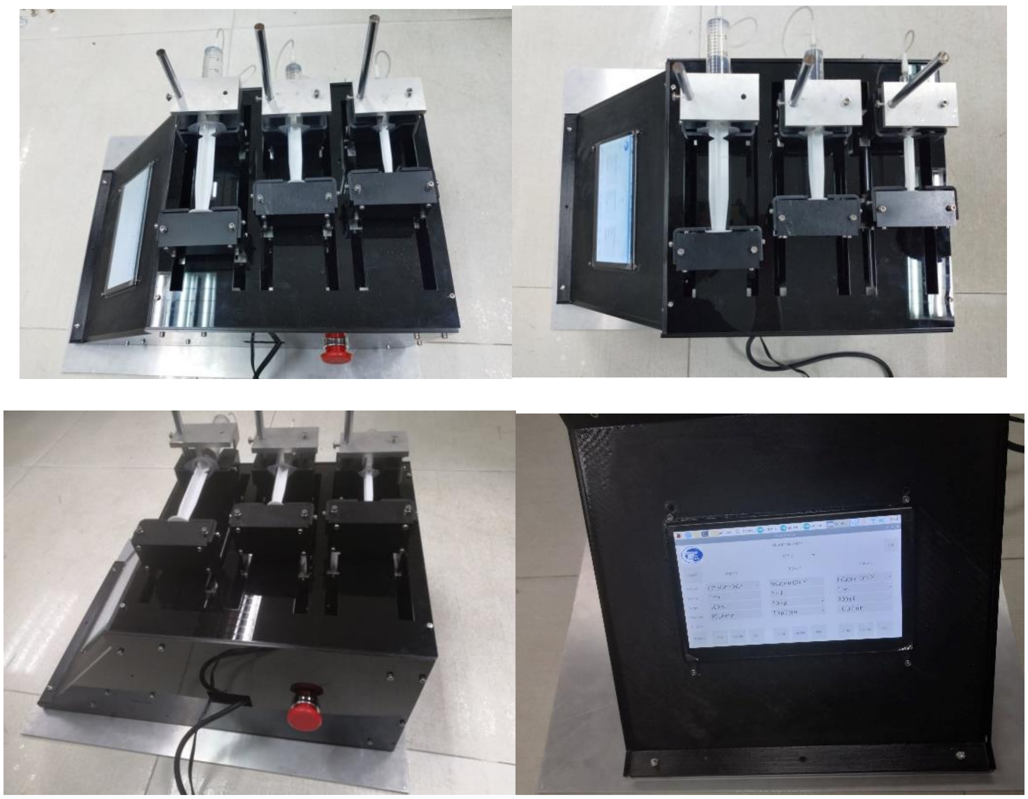


Figure 34: Real world prototype results

We have carried out the experiment to validate the functionalities of our proposed prototype versus the commercial one. The obtained results show that at greater flow rate from 200μl/min, our prototype performs nearly identical to the commercial machine, but the accuracy in time for our machine deviates further the more we decrease the flow rate. This data is very crucial for us, providing fundamentals for future development of a more accurate and effective microfluid delivery system and other applications.

References

[1] Mohsen Jafarzadeh, Fardad Farokhib, **“**Design and construction of an automatic syringe injection pump,” *Islamic Azad University, Biomedical Department, Islamic Republic of Iran*

[2] Abbas Z. Kouzani, Mile Ivankovic, Michael Fielding, AkifKaynak, Chunhui Yang, Wei Duan, Eric J. Hu, “Design and Construction of a Micropump for Drug Delivery Applications,” *The 2010 IEEE/ICME International Conference on Complex Medical Engineering*

[3] “Android Number Picker”, website: https://www.technologichron.net/

[4] “IoT Using Raspberry Pi and Firebase and Android”, website: https://www.pcbway.com/project/shareproject/IoT_Using_Raspberry_Pi_and_Firebase_and_Android.html