Bachelor Project

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Katedra řídicí techniky

Design and Fabrication of Controllable Syringe Pumps for Microfluidics

Adam Polák

Supervisor: doc. Ing. Zdeněk Hurák, Ph.D.
May 2017

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Polák Adam**

Studijní program: Kybernetika a robotika
Obor: Systémy a řízení

Název tématu: **Návrh a výroba řiditelných injekčních pump pro mikrofluidiku**

Pokyny pro vypracování:

Cílem práce je navrhnout a postavit sadu několika injekčních pump, které budou pracovat jak nezávisle tak i pod řízením z PC. Inspiraci je možno najít v mnoha projektech popsaných na webu, jako například [1] až [3]. Finální motivací pro celý projekt je experimentální výzkum v oblasti mikrofluidiky. První část práce bude sestávat z výběru vhodných komponent (krokový motor, budič, mikrokontrolér, mechanické díly, ...), stažení a upravení existujícího či tvorby vlastního 3D designu, samotné výroby mechaniky, a to nejspíše formou 3D tisku, a sestavení dílů dohromady. Na samotný návrh a výrobu mechanické časti pump bude navazovat návrh a tvorba elektroniky a software pro řízení pump. Práce bude završena sadou jednoduchých laboratorních experimentů, které prokáží funkčnost vyrobených pump.

Seznam odborné literatury:

[1] DIY Syringe Pump Using Stepper Motor. [Online], dostupné z: http://www.instructables.com/id/DIY-Syringe-Pump-Using-Stepper-Motor/, [cit. 22.12.2016].
[2] 3D Printed Syringe Pump Rack. [Online], dostupné z: http://www.instructables.com/id/3D-Printed-Syringe-Pump-Rack/, 2014, [cit. 22.12.2016].
[3] Open Syringe Pump. [Online], dostupné z https://hackaday.io/project/1838-open-syringe-pump, 2014, [cit. 22.12.2016].

Vedoucí: doc. Ing. Zdeněk Hurák, Ph.D.

Platnost zadání: do konce letního semestru 2017/2018

L.S.

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 30. 1. 2017

Acknowledgements

I would like to express my gratitute to the supervisor of this project for valuable advice and to my parents for their lifelong support.

Declaration

I declare that I have worked out this thesis independently and mentioned all used information sources in accordance with the Guideline about observation of ethical principles while preparing college final thesis.

Prague, 20. May 2017

Abstract

This project is about the design and fabrication of a syringe pump. The pump is actuated by a stepper motor, which is controlled by Arduino with UI and a desktop application for remote control. Its parametric 3D design is created in OpenSCAD. The prototype has been created and tested.The whole project is open-source, available on Github [1].

Keywords: syringe pump, 3D printing, Arduino, OpenSCAD, JavaFX

Supervisor: doc. Ing. Zdeněk Hurák, Ph.D.

Abstrakt

Tento projekt se zabývá designem a výrobou injekční pumpy. Pumpa je poháněna krokovým motorem, který je řízen Arduinem s uživatelským prostředím a desktop aplikací pro vzdálené ovládání. Parametrický design je napsán v OpenSCADu. Prototyp byl vyroben a otestován. Cely projekt je open-source, volně dostupný na Githubu [1].

Klíčová slova: injekční pumpa, 3D tisk, Arduino, OpenSCAD, JavaFX

Překlad názvu: Návrh a výroba řiditelných injekčních pump pro mikrofluidiku

Contents

Figures

Tables

Chapter 1

Introduction

This document describes the open-source syringe pump project, covering its design and function part by part, with notes about the creation process, where it is relevant. It should serve as a manual, as well as an inspiration for possible similar projects, for the whole material put together during its creation is going to be published online as an open-source design/software.

At first, some examples of syringe pumps from the market and DIY area are described, then notes about the design requirements follow, with the description of the hardware and software after that. Finally, an analysis of measured test data of the pumps functionality is made.

1.1 Motivation

The motivation behind this project is to create a cheap printable syringe pump, which could be used in various application fields, namely microfluidics. It might seem that a relatively simple device such as syringe pump should not be very expensive, but apparently the demand for these devices is not high enough or at least not many people need it, so the market does not push the price down much. The majority of these pumps are used in the medical branch, e.g. as part of hospital transfusion mechanisms, biomedical or pharmaceutical research, along with the chemical research in general. These state institutions or big research companies are not very likely to require a cheap version of the pump, but many private researchers, amateur experimenters, technology enthusiasts or just somebody who needs it for something and do not feel like investing a lot of financial resources into it, might be interested in a device, which would complete the demanded task for a reasonable price, with the possibility to assemble it themselves. The goal of this project is to design such device and fabricate a prototype with similar functionality to the commercial ones, although in a limited way, understandably.

Chapter 2

Syringe pumps

A Syringe pump, or a syringe driver, is a device, which lineary pushes or pulls the plunger of the mounted syringe, and thus causes it to expel precise volumes or flows of liquid contained inside.

Following examples of existing syringe pumps are chosen out of smaller types, leaving out the larger infusion devices and similar machines. All of them serve as mounts for the regular size syringes and are usable in a lab. The following short excerpt of commercially available ones is based on the market in Czech Republic.

2.1 Commercial

Not many dealers offering syringe pumps exist, and the selection is limited, therefore the following two examples represent noticable portion of the commonly available ones in the lowest price sector, which is our area of concern. Both of them are reconditioned products, which signalizes, that a brand new syringe pump is much more expensive, as seen with the third example. The cheapest option would be of course a used one from the second hand, but the prices vary a lot in this case, therefore it would be pointless to include them here.

2.1.1 Graseby 3100

Out of the least expensive syringe pumps available on the market, perhaps the cheapest is Graseby 3100 Syringe Pump (Figure 2.1) by British company SIMS Graseby Ltd, which is originaly a slightly older (1998), reconditioned product, available for 14 450 CZK on the online shop szo.cz [2]. It is intended for pharmaceutical administration, with fluid flow controlled by peristaltic mechanism able to perform dosing with flow 0.1 to $200\,\mathrm{mL\,h^{-1}}$ in steps of $0.1\,\mathrm{mL}$ with precision of ±2%, with build in software safety features, such as audible and visible failure alarm self testing routines on startup or incorrectly placed syringe detection, furthermore automatic sensing of its size. It can be powered either by battery or by mains power and weigths $2.4\,\mathrm{kg}$.

Figure 2.1: Graseby 3100 by SIMS Graseby Ltd [2]

Figure 2.2: Argus 600 by Codan Argus AG [3]

2.1.2 Argus 600

Slightly more expensive than Graseby 3100 is the pump Argus 600 (Figure 2.2) by Swiss company CODAN ARGUS AG. As a reconditioned product from the year 1999, it is being sold for 17 451 CZK on szo.cz [3]. It claims to have a precision of dosing of ±2% with doses between 0.1 and 999 $\mathrm{mL\,h^{-1}}$ and 0.1 mL per step. Other functions are similar to Graseby 3100.

2.1.3 Lamba VIT-FIT

Lambda VIT-FIT (2.3) is a laboratory syringe pump with capability to mount almost any syringe, because of its flexible mounting system. It should provide the accuracy of ±1 % and reproducibility of ±0.2 %, therefore it fits the laboratory requirements well. The pump weigths 3.2 kg, which makes it a a bit heavier one. The cost should be around 50 000 CZK.

2.1.4 Pump 11 Elite

The last example is not a reconditioned, but a brand new syringe pump named Pump 11 Elite (Figure 2.4) by US company Harvard Apparatus, made especially for experimental purposes. It has the precision of ±0.5%, minimal flow rate of 1.26 $\mathrm{\mu L\,min^{-1}}$. This type of pump comes in two versions - one

Figure 2.3: Lambda VIT-FIT by LAMBDA [5]

or two syringe mounting capability, and can operate syringes with volume ranging from 0.5 µL to 60 ml. This is achieved by the stepper motor with step resolution of 0.9° and precision lead screw. With microstepping, it takes allegedly 15360 microsteps per one lead screw resolution. The user interface is carried out by 4.3" WQVGA TFT color display with touch interface for easy operation. The weight of 2.1 kg makes it rather heavy for frequent transportation, but it is slightly lighter than the previous ones nonetheless. The cost of this pump is 72 479 CZK [[6]], which makes it a very expensive device, but the pricetag is by far nothing surprising in the field of laboratory syringe pumps.

2.2 Open-source projects

Since the commercial syringe pumps are so expensive, several people have already made their own projects and attempts to build much less costly devices, especially in recent years, when 3D printing is becoming more widely available and cheaper, and the DIY scene is booming. The following selection of examples of such project tries to briefly cover a few of the most interesting open-source ones published on the internet.

2.2.1 Open-Source Syringe Pump

The first example (Figure 2.5) is a project made by a team of graduate and undergraduate students at Michigan Technological University, lead by Associate Professor Joshua M. Pearce. It seems to be one with the most serious approach and documentation, they even released a paper about it [7]. The pump is constructed using customizable 3D printed part designed in OpenScad and NEMA stepper motors are controlled by Raspberry Pi with web user interface. The project is intended for any purpose from experimentation to medicine and should cost as low as 50$ per pump. However, it seems to put a little emphasis on the low volume area, stating in the paper that the performed measurement was limited to a single drop of volume cca 20 µL,

Figure 2.4: Pump 11 Elite by Harvard Apparatus [4]

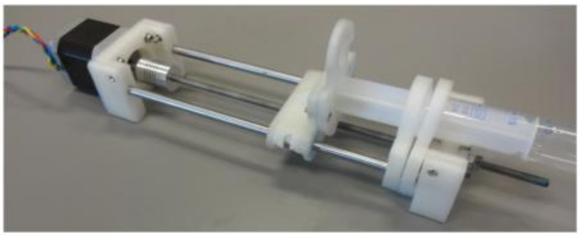


Figure 2.5: Open-Source Syringe Pump by team from MTU [7]

probably due to the relatively large syringe they used. For the needs of microfluidic research would be beneficial to use a smaller volume syringe and perform more precise measurements to test the pump.

2.2.2 Open Syringe Pump

The next example is a pump developed and published on instructables.com [8] by user naroom. The pump (Figure 2.6) is mounted on a aluminum rail, which provides the possibility of fitting it on various mount setups, but at the same time makes the whole pump rather big and clumsy for transportation. The holders are 3D printed and designed parametrically in OpenSCAD, and thus easily redesignable. Prototyping board Arduino Uno R3 is used for the control, with a shield and enclosure box for mounting on the rail. The whole

Figure 2.6: Open Syringe Pump by naroom from hackaday.io [8]

pump including the shipping costs should cost in the USA around 250$, which is more than the pump from MTU, but they did not include the shipping costs of the material in their estimates, so the real price could be higher. The author claims that the pump should be able to push precisely up to 1 µL of fluid, which sounds good, but no precision measurement is provided, so it is not certain.

2.2.3 3D Printed Syringe Pump Rack

The last and maybe the most interesting one is a syringe pump by Aldric Negrier published on instructables.com [[9]]. The pump is compact and nicely designed (Figure 2.7), the design of the 3D parts is unfortunately not parametrical though, but common one made in SketchUp, so the redesigning could be a bit more difficult. Since it has a complete enclosure and is not very spacious, the portability is definitely the best out of the three. What puts it down a bit is the lack of a power connector, which is understandable, since there is no PCB and the electronic parts are just soldered together, but it is not very practical to have just a buch of wires comming out and soldering it to a PC power supply does not really help it. The pump is controlled by a prototyping board Arduino Nano, with user interface only by commands via serial port. From the brief test included in the manual it seems that the pump is rather precise with good repeatability results. The errors are allegedly under 0.5 µL, but not a lot of data is included.

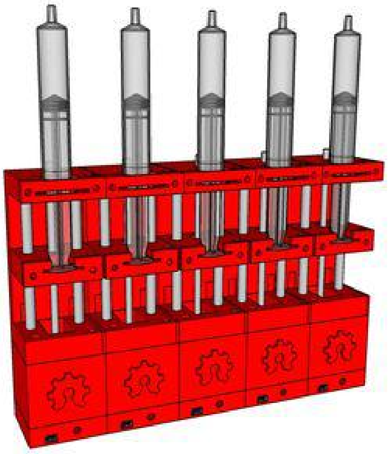


Figure 2.7: 3D Printed Syringe Pump Rack by Aldric Negries from instructables.com [9]

Chapter 3

Design Requirements

Unlike the examples from the previous chapter, our syringe pump is intended mainly for the usage in experimental microfluidics, therefore it is more important, how it performs with low volumes, than how long is it able to expel pharmaceutics as an infusion device. To formulate the goal of this project, following function requirements were formulated:

- The syringe can be placed in vertical position with the outlet facing down.
- The pump has to perform well on low volumes.
- Portability shloud be taken into consideration.
- The pump has to be controllable from PC and directly via display as well.
- Preference should be given to some widely spread and affordable embedded platform.

The first requirement is based on the need of creating the particle containing droplet, where the influence of gravity could cause an uneven distribution of particles comming out of a horizontaly placed syringe, whereas the vertical placement ensures that the particles are oriented towards the outlet of the syringe. This requirement affects the design significantly in a way that the most of the available commercial syringe pumps as well as the majority of published DIY syringe pump projects have a horizontal syringe position as default, in case of the DIY pumps often even without a possibility to place the pump in a way that the syringe is facing down, therefore some of the design elements used in those examples cannot be used here or have to be revised. This topic is further discussed in the subsection 4.1.4.

The experimental purposes require a certain precision as well as accuracy, since it is often operated with very low volumes of liquid, ranging as low as few µL. The precision and accuracy of the pump relies heavily and primarily on the transition system, which provides the linear motion, i.e. the trapezoidal rod and nut together with the stepper motor. Subsection 4.1.2 explores this in more depth.

The design should be portable, compact and lightweight, since there is no reason for it to be the opposite. 3D printed parts from fillament as PLA are very light, so the most weight will be caused by the transition system - the stepper motor with trapezoidal screw and nut, which is still not very heavy. The goal is a possibility of having a rack of a few pumps, which can be joint together in a row, connected by a snap, slide or a different kind of joint.

Since it is going to be an open-source project, which should provide the possibility for various people to get the laboratory equipment they need, but do not have the need or the resources to buy the expensive commercial one, and it is certainly not planned nor expected to be produced and distributed widely, it makes sense to use a platform, which is widespread and has a large community, so the people would be able to fabricate it themselves.

Chapter 4

Hardware

This chapter covers the mechanical and electrical part of the syringe pump design.

4.1 Mechanical part

The mechanical design consists of the 3D printed pards (holders, enclosures, coupler, etc) and the metal parts (Trapezoidal screw, nut). Folowing section is going to cover the process of designing and choosing the items with suitable parameters.

4.1.1 Trapezoidal Rod and Nut

The linear motion made possible by the leadscrew and the nut (Figure 4.1), which are connected to the motor shaft via a shaft coupler (Figure 4.2) designed by a thingiverse user loco, which proved as an sufficiently reliable and cheap alternative to the metal shaft couplers, making the overall design a bit cheaper.

4.1.2 Backlash and Deviation

At first, a ordinary threaded rod was used to test if it is possible, since it would mean a significant price reduction, because it is available in every ordinary ironmonger's around the corner. It ended up being not very suitable solution, because of a lot of backlash and Z-axis deviation, along with the linear motion not being very smooth overall. Threaded rods are simply not made for linear motion, therefore can contain inaccuracies in the threading. In the end it was chosen to use the trapezoidal screw, which is made specifically for linear motion and offer minimal backlash. A slight Z-axis deviation is still present and influences the function of the pump during the change of the direction of the linear motion - between pushing and pulling, there is a brief transition period, when the deviation is changing the angle orientation. If needed, backlash could be improved by two tightly connected nuts in a way that each of them pushes on a different orientation of the threading, as displayed on the Figure 4.3

(a) Trapezoidal screw (b) Trapezoidal nut

Figure 4.1: Trapezoidal screw and nut

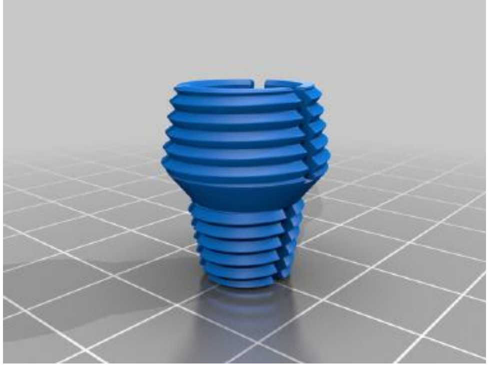


Figure 4.2: Shaft coupler 5/10 mm [[11]]

4.1.3 OpenSCAD

The 3D printed part was developed in OpenSCAD, which is a parametric CAD software using a functional programming laguage of the same name. It allows to program parametric designs, i.e. modify it via altering the variables, for example resizing the holder for a part of different size. This feature is useful because of the reusability, where it can be easily improved according to ones need.

4.1.4 Linear Motion Parts

The 3D printed part of the linear motion mechanism constist of following parts:

- Stepper mounter, which has holes for the M3 bolts for tightening of the stepper motor, hole for the motor shaft and holes for tightening the smooth leading rods, which are made of the same diameter as the rods, because it is the easiest way to attach them in firmly.
- Syringe mounter has a mechanism to mount the top of the syringe barrel. A end-switch is also present to switch off the motor when the pump reaches the limit position.
- Moving slider consists of a nut housing, linear bearings and syringe

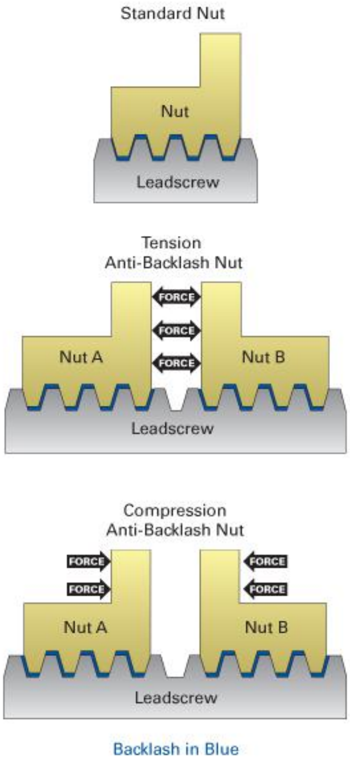


Figure 4.3: Backlash compensation [12]

plunger top mounting mechanism (Figure 4.4). The nut housing is made tight enough, so the nut holds firmly inside, since it has one side uncovered. It proved well to make the nut hole exactly the size of the nut and just a few millimeters thick wall around it, and the nut can be easily forced in using a bench vise or similar tool. The linear bearing was, similarily as with the shaft coupler, 3D printed instead of using a metal part. An OpenSCAD design, similar to what the company Igus makes [13], was already created and published on thingiverse.com by user shaa80 [14]. It functions well, just a bit of lubrication, which then stays in between the spikes, needs to be applied.

- Top part of the mechanism serves as the second mounting point of the leading rods and contains also the second limit switch.

- Backside construct provides for one thing the possibility of tightening the steady parts to it with screws and for another the hole for the wires leading from the switches. These wire holes can be covered with a cover, as displayed on the Figure 4.5.

The syringe can be mounted to the device using the tightening plates, which can be tightened to the holders with bolts (M3), which can be screwed to the mounter using the nut inserted to the nut hole.

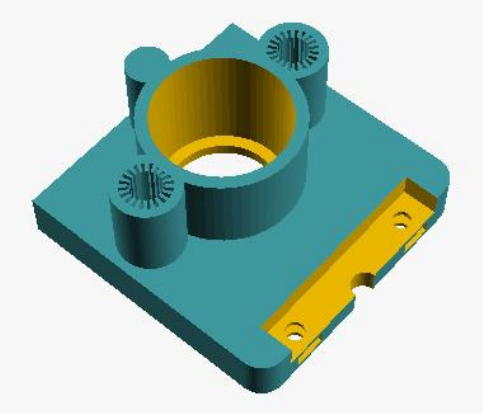


Figure 4.4: Slider under the angle

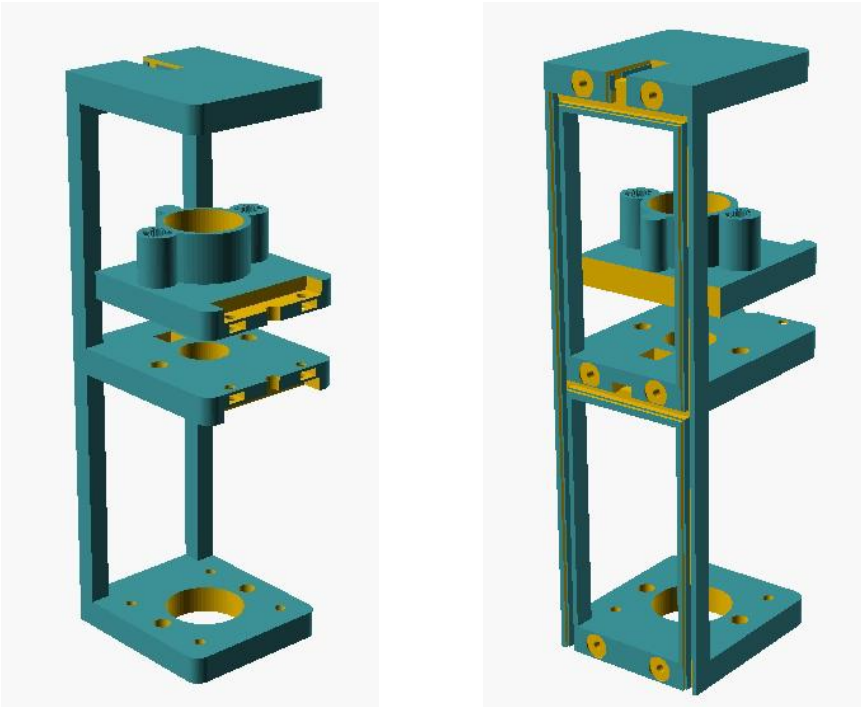
Figure 4.5: Backside with the wire cover

4.1.5 Syringe

The prototype design is adapted to Omnifix-F Duo 1 mL syringes, but can be easily changed to a different one by adjusting the constants in the file `constants.scad`. It still has to be tested to use a better, non-plastic syringe, since it might improve the performance a bit. The plastic syringes also get a bit stiff after some time.

4.1.6 Enclosure

It was initially intended to include electronics with the enclosure in one piece with the pump itself, but later a realization was reached, that it would be too bulky at the current state. After the PCB is made, which will be much more thinner than the currently used prototyping board, can be the enclosure improved in that way. The current electronics enclosure is displayed on the Figure 4.7



(a) Front (b) Back

Figure 4.6: Design of the linear motion parts

4.2 Electronics

In this section, the electronics part of the project is described. The whole schematic can be seen on the Figure 4.10. The Circuit consists of a controller Arduino Nano, stepper motor driver DRV8825, stepper motor Sanmotion F, OLED display, and some minor parts, such as push buttons or filter capacitor.

4.2.1 Controller

As mentioned before, this project is Arduino-based. Arduino is an open-source prototyping platform with libraries that assure its easy usage, making it accessible even for the beginners. Partly because of that, it has a large community worldwide, and that is the reason, why it was chosen for this project. Some may say, that it makes some things too easy and prohibits people from learning the embedded programming properly, but that is generally not a valid argument, since it makes some people interested in embedded programming, who would turn away from it otherwise. And who has the desire to learn it ”the hard way”, still can. Arduino, especially stripped-down Arduino Nano used here, is basically just Atmel prototyping kit with some extra support libraries.

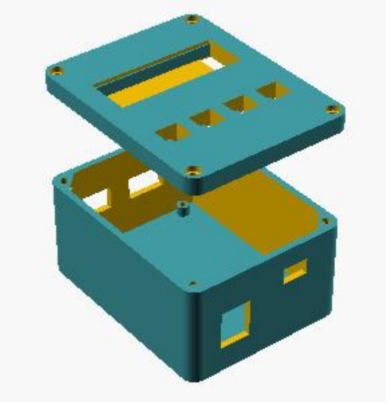


Figure 4.7: Enclosure

Figure 4.8: Sanyo Denki Sanmotion F [[15]]

4.2.2 Actuator

The stepper motor Sanmotion F SS2421-5041 [[15]] by Japanese company Sanyo Denki was chosen because of its light weight of 0.07 kg and small dimensions (height just 11.6 mm), but still provides enough torque (0.083 $\mathrm{N\,m\,min^{-1}}$) for this application. It can be changed for a different motor without much trouble. The bolt holes have the same position as the NEMA 17 [16].

4.2.3 Power Source

The Arduino Nano itself has three possibilities for the power supply [[17]]:

1. Mini USB connector,
2. 5V pin,
3. $\mathrm{V_{in}}$ pin.

The first option is obvious, the second is a bit dangerous and not recommended, because it requires regulated 5V power source and no more, sice no regulator or safety circuits are present for this option.

Figure 4.9: Display Menu

The third option is the best one for our purpose, since the V_{in} pin leads to LM1117 voltage regulator, which regulates the voltage to 5 V and has maximum input voltage of 20 V [18]. It allows the use of unregulated power supply, but the recommended and much safer way is to use regulated power supply with voltage of 12 V, so it is assured, that it does not exceed the maximum and so does not burn out. The motor is rated at 1 A maximum and the board draws much less current, 19 mA plus the current drawn from the I/O pins, which is 40 mA maximum per pin, and our circuit certainly does not draw that much, therefore 2 A power source is safely enough to power the board and the motor in parallel.

4.2.4 Display

A cheap OLED 0.91” display with 128x32 resolution has been used for the prototype. It uses I2C bus, which is benefitial because of the low amount of connections needed - only VCC, GND, clock line (SCL) and data line (SDA). It is a monochrome display, as seen on the Figure 4.9.

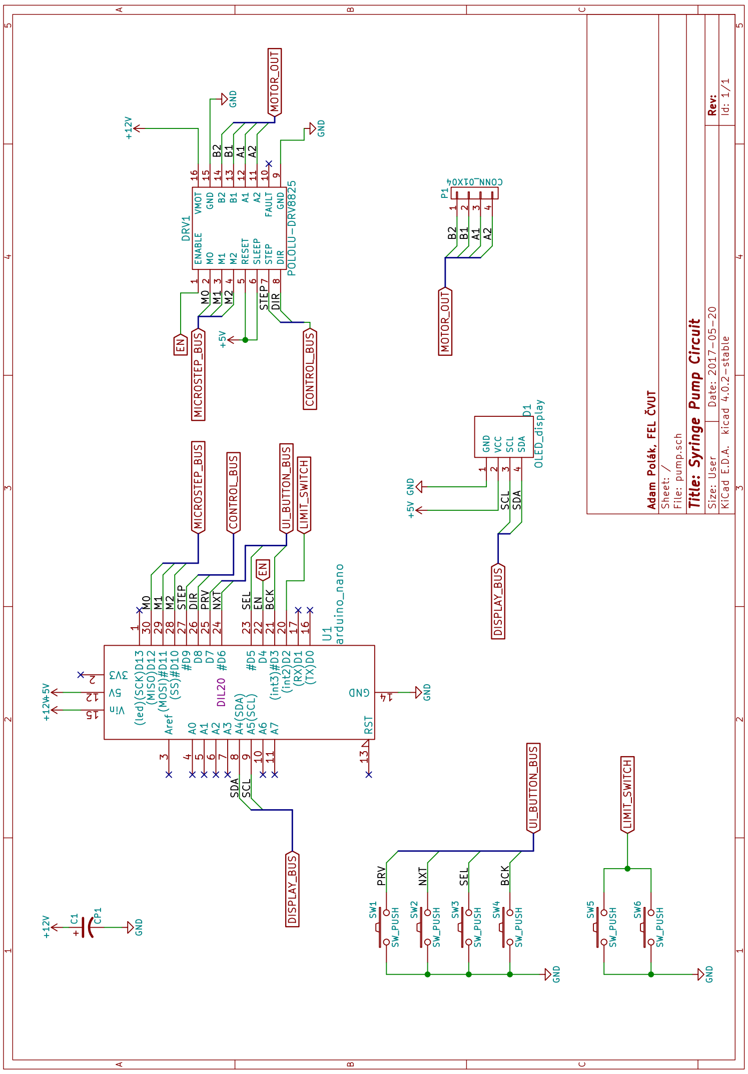


Figure 4.10: Circuit Diagram

Chapter 5

Software

The software consists of two parts - the embedded, microcontroller one and the pc side, which is a java application. These programs communicate together via a serial port.

5.1 Embedded Software

The Embedded side is responsible for the stepper motor control and display user interface and processing of the requests from the PC. At first was the program written as a simple one file C program with help of the robust Arduino libraries (which are in fact written mostly in C++), but later on became unbearably chaotic, because of the addition of functionalities, therefore it was rewritten later on using an object oriented approach and implemented in C++. In order to logically organize the structure, it was divided into classes according to the function, and so following classes were created:

- `MotorControl`
- `DisplayUI`
- `ExternalUI`
- `Calibration`

The basic function behind the class names of the classes is probably apparent - the class `MotorControl` takes care of the steper motor and contains all the neccesary members and functions for that, plus the constants providing the relation between the volume/flow of the liquid to be expelled and the motor steps/speed. The class `Calibration` includes the methods of setting these constants and saving them to the EEPROM memory. `DisplayUI` takes care of the OLED display and its user interface - menus, settings, motor requests, etc. `ExternalUI` reads the input sent from the PC app and makes arrangments accordingly. Lastly, the file `pump_main.ino` contains the loop and setup methods instead of the expected main method, because it is needed for a program upload using the Arduino IDE and USB, otherwise it would be necessary to upload via the onboard programmer. It also contains the interrupt methods and instances of the other classes.

5.1.1 Stepper Motor Control

As mentioned above, the stepper motor control is taken care of by the class MotorControl. It uses the library `AccelStepper` to control the motor, which implements the acceleration and deacceleration of the motor, according to the theory in paper [].

5.1.2 Display UI

The OLED display is controled by the class DisplayUI, which uses the library `U8glib` to display the UI. The class contains all the necessary logic behind it.

5.1.3 User Interface

The menu has several options, through which can be navigated using the buttons with symbols marking the label NEXT and PREVIOUS for item browsing and SELECT and BACK for selection/going back. The possible requests are following:

- *Set Move* contains the menu dialog for setting the precise movement via volume and flow, as well as the settings of the motor microstepping and software reset option.
- *Direct Up* option moves the slider holding the syringe top up through request to the motor, which will move it as long as it remains selected. This option is intended for manually setting the slider position.
- *Direct Down* option does the same as *Direct Up*, but with the opposite direction.
- *External Control* takes care of the serial communication with the PC, namely decoding the recieved code and setting the parameters according to it.

The item in the *Set Move* menu are:

- *Move*, which orders the motor to start running, using the saved values of flow and volume.
- *Flow* option lets the user set the flow with precision on tenth of $\mu L\,s^{-1}$. The value is set by holding the NEXT and PREVIOUS buttons, and is expected to be within the interval $(0, 30\rangle$. The flow value is also used for the *Direct Up* and *DirectDown* options from the main menu.
- *Volume* option sets the requested volume, again with precision on tenth of $\mu L\,s^{-1}$. The negative value means the pump pushes and expels fluid, the positive the opposite.The value is not limited, because the pump will stop at the limit by itself.

- *Microstepping*, which features a dialog to set the microstepping mode of the stepper motor. By default is it 16, but acceptable values are $\{1, 2, 4, 8, 16, 32\}$, as explained in the datasheet [22].
- *Software Reset* option resets the software of the microcontroller by jumping to the beginning of the program. Note that it does not reset the physical inputs.

5.2 PC program - Java Application

The PC application provides the possibility to control the pump through a GUI and serial communication code. The program was developed in Java, using JavaFX with FXML markup language for the GUI.

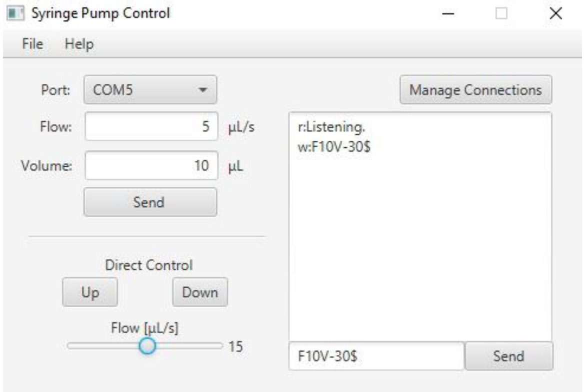
5.2.1 Object Oriented Design

The program itself consists of following classes:

- `Main` is the class that inherits from JavaFX Application class and serves as the entry point for the whole application, initializing all the graphical stages and their controllers.
- `MenuController` is a controller for the `Menu.fxml` file, which defines the GUI of the main application window (Figure 5.1). It handles all the input data and hands it over to the `ArduinoCommunicator` class, which sends it over the serial port.
- `ConnectMenuController` serves as a controller for the `ConnectMenu.fxml` file, which defines a port connecting and disconnecting dialog, as seen on the Figure 5.2.
- `ArduinoCommunicator` takes care of connecting to ports and devices and contains methods for that. It has a list of the ports and a list of connected Arduino objects.
- `SerialMonitor` is a wrapper class for the TextArea in the menu, which serves as a serial monitor and displays the communication with the device.
- `SerialReader` is a Thread, which reads the serial port of the connected Arduino and displays the written messages.

5.2.2 Serial Communication

A library ArduinoJava was created for this project using a part of existing library [19], adding the compatibility features for JavaFX and several port connection. The class `Arduino` represents the Arduino connected through the serial port and takes care of the communication using the jSerialComm



Port: COM5

Flow: 5 µL/s

Volume: 10 µL

```
r:Listening.
w:F10V-30$
```

F10V-30$

Figure 5.1: Main window of the GUI

Available Ports: COM5

Connected Ports:

Figure 5.2: Port managing dialog

library [20]. Multiple instances of this class can be created and managed, and thus control several Arduinos.

The class PortList represents a list of serial ports available.

5.2.3 JavaFX GUI

JavaFX is a software platform for creating desktop applications (among other features), replacing the old Swing and AWT libraries for GUI creation. The GUI itself is defined in a FXML file, which is a special version of XML format. It is possible to use a program for quick user interface designing, such as Scene Builder by Gluon [21].

Typically, for every FXML file exist a controller file, which contains the handlers and logic behind the GUI. This separation of logical and graphical side of the application is one of the main advantages of JavaFX over Swing and AWT, making the code clearer and easier to understand.

5.2.4 Operation Instructions

There are three ways how to put in commands to control the syringe pump through the GUI application. The first option is to fill in the *Flow* and *Volume* boxes and click on *Send* button under it. The second way is only auxiliary, usable for directly moving the syringe slider. The order is given by pushing the *Up* or *Down* buttons and holding it as long as it is needed to move the pump slider. The flow can be set by moving the slider bar underneath

Command	Code	Argument	Example
Set Flow	F	float (0,30>	F14.6$
Set Volume and Run	V	float (-1000+,1000+)	V-24.3$
Set Microstepping	M	int {1,2,4,8,16,32}	M32$
Set Acceleration Rate	A	int <1,1000>	A100$
Software Reset	X	no argument	X
Set Syringe Constant	C	float <1,1000>	C34.7$
Get Syringe Constant	G	no argument	G$

Table 5.1: Instructions for remote control

the buttons.

The third and the best way, how to control the syringe pump, is to send the commands through the serial monitor on the right. The possible codes are displayed in the Table 5.1. Note that some orders can be combined into one, for example `F10.2V-23$` sets the flow to 10.2 $\mu L\,s^{-1}$ and the volume of 23 µL. The negative volume means pushing, the positive pulling. One can think of it as the total volume of the fluid inside the syringe is either lowering, or rising. The ending delimiter is the symbol $, which has to end every command, except for the software reset, where it is not required. To send the code, the button *Send* has to be clicked, or the key *Enter* has to be pushed.

5.2.5 Calibration

When a different syringe from the default one (B.Brown Omnifix-F,1 mL) is inserted, the pump has to be calibrated. The calibration can be done so far by the desktop application only, and thus via the serial monitor by appropriate command, as displayed in the Table 5.1. It is done by sending a constant to the pump, where is it saved to the EEPROM memory and loaded on every following startup, until changed again. The constant is the volume [µL] of the expelled fluid, when the motor makes exactly one rotation (360°). It has to be computed through a simple computation:

$$\mathrm{V_{rot}} = \frac{1000 \times \mathrm{V_s} \times \mathrm{A}}{\mathrm{l_s}}, \tag{5.1}$$

where $\mathrm{V_{rot}}$ [µL]is the volume pushed by one rotation of the motor, $\mathrm{V_s}$ [mL] is the Volume of the syringe, $\mathrm{l_s}$ [mL] is the length of the marked scale and A [mm] is the ascent of the trapezoidal rod.

The pump then computes the constant for the relationship between the volume and the motor steps necessary to expel it, using following formula:

$$\text{steps} = \frac{\mu \times \mathrm{M}}{\mathrm{V}_{\text{rot}}}, \tag{5.2}$$

where steps [-] is the number of steps, μ [-] is the microstepping number, M [-] is the number of steps required for one rotation with microstepping off[1], and V_{rot} [µL] is the volume pushed by one rotation.

---

[1] Provided usually in the datasheet of the stepper motor, sometimes in degrees rotated by one step. For example 200 steps is equal to 1.8 °

Chapter 6

Experiments and Results

This chapter presents, how the pump functions, based on a precision test. Unfortunately, the tools available were a bit less precise than needed, meaning that the scales (6.2) have precision to one tenth of a µg, which means, that theoretically, it should be possible to measure objects with this precision. In reality, probably due to the age of the device or other impacts, the value fluctuated around the range approximately ±0.5 µg to ±1 µg , which is by far not ideal, considering that the measured volumes should go all the way to 1 µL (which is 1 µg for water) or even lower. In general, the measuring device should be at least by one order more precise, that the measured values, especially when the precision of one device needs to be measured by a different device. This requirement was not satisfied, although it was measured with probably what are the most precise scales on the whole faculty.

For the reasons mentioned above, the measurement should not be considered as the precision defining test, but rather as a precision testing experiment, through which a relative error of approximately 6 % was measured. What needs to be noted though, is that the precision measurement itself has an error, defined by the computed uncertainty.

6.1 Precision Test

The test carried out lied in a measuring at least five values for several volumes. The resulting difference between the requested and the measured values is displayed on the Figure 6.1.

The measurement uncertainty of type A, as defined in [23] is:

$$u_A = \sqrt{\frac{1}{n(n-1)}\sum_{i=1}^{n}(x_i - \overline{x})^2} = 0.0724\,\mu\mathrm{L}, \tag{6.1}$$

where n is the sample size, x_i is the sample member and $\overline{x}$ is the sample mean. The type B uncertainty is:

$$u_B = \frac{\Delta}{\sqrt{3}} = 0.57\,\mu\mathrm{L}, \tag{6.2}$$

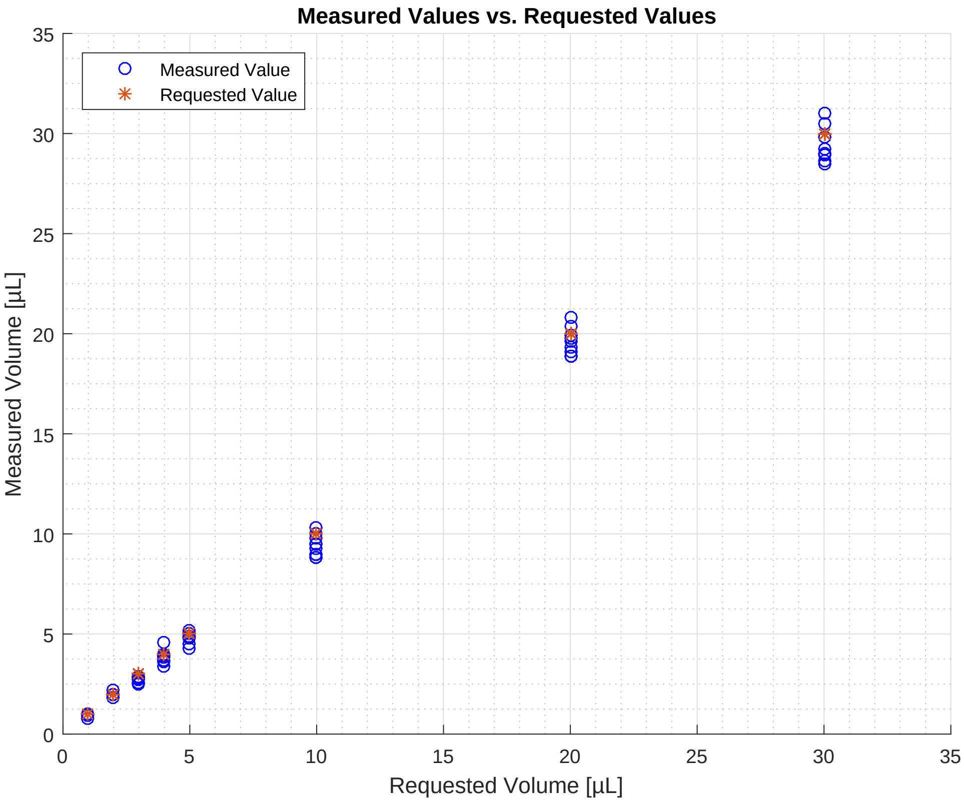


Figure 6.1: Results of the precision testing experiment

where Δ is the resolution of the scales, 1 µL in this case. The final combined uncertainty of the measurement is thus:

$$u_C = \sqrt{{u_A}^2 + {u_B}^2} = 0.575\,\mu\mathrm{L}. \tag{6.3}$$

From the graph is apparent, that the values fluctuate with deviance of approximately around ±1 µL, which is not too bad, considering the conditions of the measurement. It would be interesting to perform another one with a more precise scales and see, how it changes. During the test, it was cruitcial not to wait too long after expeling the fluid, because at this scale, the evaporation of the fluid was already apparent. Also, with the lowest volumes, it was needed to dip the needle tip into the fluid, because the water droplet has a high surface tension, which makes it unable to create a droplet of lets say lower volume, than 10 µL.

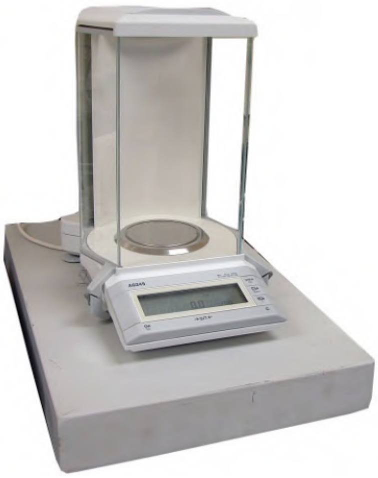


Figure 6.2: Mettler TOLEDO AG245

Chapter 7

Possible Improvements

The prototype could be improved in several ways, following ideas represent the realistic and achiavable examples:

- Producing a PCB instead of the prototype board would make it possible to significantly reduce the size of the electronics enclosure, as well as making the electronics more secure and reliable. The schematics is drawn, but it would need redrawing in order to eliminate the need of using the Arduino Nano prototyping board. Just the processor and the inevitable parts could then be used, while perserving the possibility to use to current program.
- Connecting the pump with LabVIEW or/and Simulink would be very beneficial for the usability, allowing for flexible usage during various experiments. There are tools for the Arduino platform, which provide these features [25] [26].
- Improving the syringe holding system to a more flexible one would greatly increase the convenience of the pumps use.
- If needed, a ballscrew could be aquired, which would increase the precision and decrease the axial deviance, which causes the transitional nonlinearity during the direction change.
- The enclosure cover could have a boltless joint (flip, snap, . . .), which would make the design more user friendly, with less bolts and screwing.

Chapter 8

Conclusion

A syringe pump design has been created and functioning prototype crafted (photo in Appendix B). The focus had to be on several fields - 3D design, linear motion, electronics, embedded and desktop application programming. The precision of the prototype seems acceptable, but needs to be further tested using more precise scales. The measurement showed, that the measured volumes differ from the requested ones by approximately ±5.7 %, while the uncertainty of the measurement was 0.575 µL. The pump is open for further development, as discussed in chapter 7. The cost of the prototype fabrication is around 3000 CZK, as discussed in the appendix A in more detail.

Bibliography

[1] Adam Polák (2017). *Github repository of this project.* [online] Availabe at: `https://github.com/reichenberger/SyringePump`

[2] KRÁSNÝ - zdravotnická technika s.r.o. (2016). *Injekční pumpa Graseby.* [online] Available at: `http://www.szo.cz/cs/lekarske-pristroje/?injekcni-pumpa-graseby-pumplin` [Accessed 21 May 2017].

[3] KRÁSNÝ - zdravotnická technika s.r.o. (2016).*Lineární dávkovač farmak ARKUS.* [online] Available at: `http://www.szo.cz/cs/lekarske-pristroje/?linearni-davkovac-farmak-arkus-pumparku6` [Accessed 21 May 2017].

[4] Harvard Apparatus (2017). *Standard Infuse/Withdraw Pump 11 Elite Programmable Syringe Pumps.* [online] Available at: `http://www.harvardapparatus.com/media/manuals/Product%20Manuals/11%20Elite%20Manual_5420-002REV1.0.pdf`[Accessed 21 May 2017].

[5] LAMBDA Laboratory Instruments *VIT-FIT polyvalent syringe pump* [online] Available at: `http://www.lambda-instruments.com/?pages=syringe_pump_manual_technical_specifications` [Accesed 25 May 2017]

[6] krd.cz (2017). *Injekční pumpa 11 Elite.* [online] Available at: `http://www.krd.cz/products/injekcni-pumpa-11-plus/` [Accessed 21 May 2017].

[7] Bas Wijnen, Emily J. Hunt, Gerald C. Anzalone, Joshua M. Pearce (2014) *Open-Source Syringe Pump Library.* [online] Available at: `https://www.academia.edu/8369549/Open-Source_Syringe_Pump_Library` [Accessed 21 May 2017].

[8] naroom (2015). *Open Syringe Pump.* [online] Available at: `https://hackaday.io/project/1838-open-syringe-pump` [Accessed 21 May 2017].

[9] Aldric Negrier (2015). *3D Printed Syringe Pump Rack.* [online] Available at: `http://www.instructables.com/id/3D-Printed-Syringe-Pump-Rack/` [Accessed 21 May 2017].

[10] CNCshop.cz (2016). *Trapézové šrouby.* [online] Available at: `http://www.cncshop.cz/trapezove-srouby_c` [Accessed 21 May 2017].

[11] loco (2016). *Self-centering tapered-thread Z-axis coupling [v2].* [online] Available at: `https://www.thingiverse.com/thing:602481` [Accessed 21 May 2017].

[12] Helix Linear Technologies, Inc. (2012) *The Acme Lead Screw Backlash Survival Guide* [online] Available at: `http://blog.helixlinear.com/bid/206446/The-Acme-Lead-Screw-Backlash-Survival-Guide` [Accessed 21 May 2017].

[13] igus.eu (2017). *drylin® RJMP - 2nd generation of plastic linear bearings* [online] Available at: `http://www.igus.eu/wpck/7188/DryLin__R__RJMP` [Accessed 22 May 2017].

[14] Alexander Shalnev (2016). *Customizable IGUS style linear bearing* [online] Available at: `https://www.thingiverse.com/thing:1754486` Available at: `https://www.thingiverse.com/thing:1754486` [Accessed 22 May 2017].

[15] Sanyo Denki (2008). *Sanmotion datasheet* [online] Available at: `http://www.farnell.com/datasheets/356617.pdf?_ga=2.59245736.46202373.1495480486-1020888000.1487238061`[Accessed 22 May 2017].

[16] reprap.org (2016). *NEMA 17 Stepper motor* [online] Available at: `http://reprap.org/wiki/NEMA_17_Stepper_motor` [Accessed 22 May 2017].

[17] E. Vita (2014). *Arduino Nano* [online] Available at: `http://download.arduino.org/products/NANO/Arduino%20Nano-Rev3.2-SCH.pdf` [Accessed 22 May 2017].

[18] Texas Instruments Inc. (2000, last revision 2016). *LM1117 800-mA Low-Dropout Linear Regulator* [online] Available at: `http://www.ti.com/lit/ds/symlink/lm1117.pdf` [Accessed 22 May 2017].

[19] Hidray Gupta (2016). *Java-Arduino Communication Library* [online] Available at: `https://sourceforge.net/projects/javaarduinolibrary/` [Accessed 23 May 2017].

[20] Fazecast, Inc. (2017). *jSerialComm, Platform-independent serial port access for Java.* [online] Available at: `http://fazecast.github.io/jSerialComm/` [Accessed 23 May 2017].

[21] Gluon (2017). *Scene Builder.* [online] Available at: `http://gluonhq.com/products/scene-builder/` [Accessed 23 May 2017].

[22] Texas Instruments Inc. (2010, revised 2014)*DRV8825 Stepper Motor Controller IC*. [online] Available at: `http://www.ti.com/lit/ds/symlink/drv8825.pdf` [Accessed 24 May 2017].

[23] Milan Červenka (2013)*Zpracování fyzikálních měření*. [online] Available at: `http://fyzika.feld.cvut.cz/~cervenka/vyuka/zprfm/zpracdat.pdf` [Accessed 24 May 2017].

[24] David Austin (2005) *Generate stepper-motorspeed profiles in real time* [online] Available at: `http://web.archive.org/web/20140705143928/http://fab.cba.mit.edu/classes/MIT/961.09/projects/i0/Stepper_Motor_Speed_Profile.pdf` [Accessed 24 May 2017].

[25] The MathWorks, Inc. (2017) *Arduino Support from Simulink.*[online] Available at: `https://www.mathworks.com/hardware-support/arduino-simulink.html` [Accessed 25 May 2017]

[26] National Instruments Corporation. (2017) *LINX by Digilent/LabVIEW MakerHub.*[online] Available at: `http://sine.ni.com/nips/cds/view/p/lang/cs/nid/212478` [Accessed 25 May 2017]

Appendix A

Table of Acquired Items

The important items, that have to be purchased for the fabrication of one syringe pump are displayed on the Table A.1. The minor things such as cables, buttons, bolts, etc. were not included, for their price is not very significant and they are easily acquirable. The total cost of one syringe is thus a bit more, than the total price in the table. Approximately, the total cost could be lets say 3000 CZK, which is cca. 114 EUR or 127 USD, which makes is considerably cheaper, than the commercial devices.

Item	Name	Dealer	Cost [CZK]
Leadscrew nut	TRM-BR trapézová válcová matice bronz, TRM-BR 1002	cncshop.cz	210
Leadscrew	TR trapézový válcovaný šroub, TR 1002, 15 cm	cncshop.cz	29
Syringes	Omnifix-F Duo 1 ml, 100 ks v balení	medplus.cz	470
Microcontroller	Arduino Nano	arduino-shop.cz	678
Display	IIC I2C Displej OLED 0.91" 128x32	arduino-shop.cz	206
Stepper driver	RAMPS1.4 DRV8825 driver pro krokové motory	arduino-shop.cz	117
Stepper motor	SANYO DENKI - SANMOTION SS2421-5041	farnell.com	1128
			Total: 2838

Table A.1: Acquired Items

Appendix B

Photo of the prototype

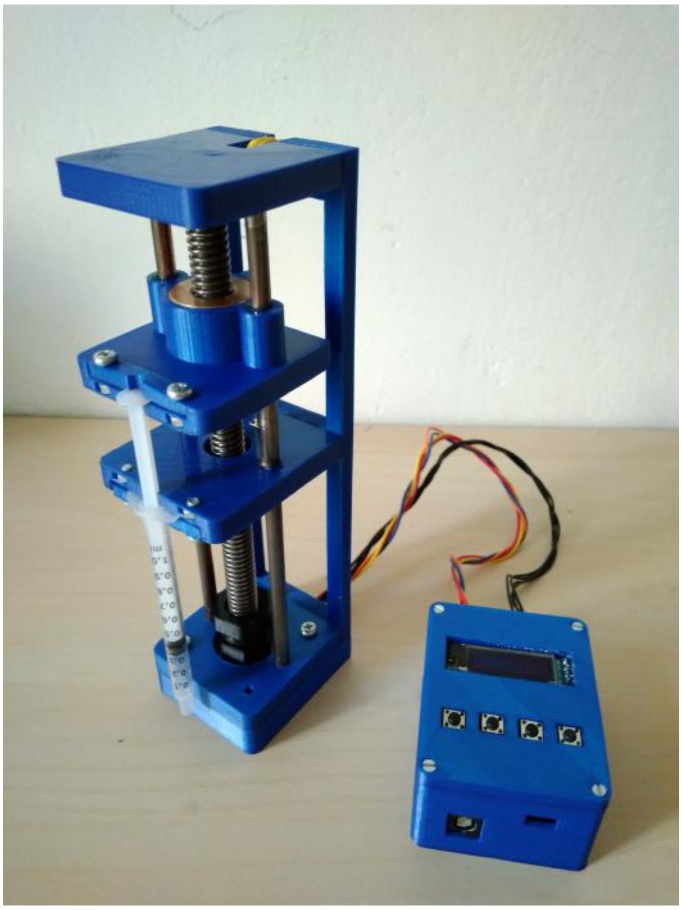


Figure B.1: Prototype of the syringe pump