

A Technical Report on Fluid transportation network

Beketova Aktoty

Kaidarov Assanali

Doschanov Askhat

Rychshanova Aimira

Alimkulov Rustam

Mannanov Ruslan

Yestay Abilmansur

Zholaman Temirlan

**November, 2018**

**Contents**

* **Introduction and aim**
* **Technical content**

1. Design calculations & a flow diagram depicting the control process and the system operation.
2. Description of pump’s design including justification of design choices.
   1. Basic requirements for design
   2. Rotor design
   3. Casing design
   4. Material and printing
3. Description of the controlling process including justification of coding/implementation choices.
4. Results from all benchmark cases
5. Results from measuring cases
6. Images from all stages of construction, assembly and operation.
7. Challenges we faced and how we solved it.

* **Conclusion**
* **Reference sources**
* **Appendices**

**Introduction and aim**

The main aim of the given work is to demonstrate engineering skills including all (mechanical, electrical, civil and etc.). We were obliged to perform a detailed design and analysis of the fluid transportation system, particularly the functional pump that is maximumly effective both in manufacturing and in the application. For the practical demonstration of the features of our own pump, we were given the task to design a system where liquid transportation occurs from tank to another tank including special requirements (rate of flow, overflow mechanism, liquid level and etc.).

We as a team divided the obligation of such a big job. Beketova Aktoty and Kaidarov Assanali were responsible for the report of all the steps that were conducted. Also, one of the main things - 3D modeling of all components of our pump was the obligation of Doschanov Askhat and Rychshanova Aimira. Alimkulov Rustam and Mannanov Ruslan were obliged to implement the codes of the basic operation control and additional cases. Yestay Abilmansur and Zholaman Temirlan supposed to present our work. However, we interacted with each other and everyone was aware of other processes. As all of these parts are interrelated with each other, we shared all the data and results that we got.

The main stages involved in this report:

* Performance of a detailed design and analysis of the fluid transportation system so that it can achieve the functional characteristics & objectives.
* Designing, manufacturing, and assembling of a functional peristaltic pump with the prescribed specifications.
* Design and implementation of a system control process on a provided microcontroller.
* Assembling and using the system to demonstrate attainment of prescribed functional characteristics.

**Technical content**

1. **Design calculations & a flow diagram depicting the control process and the system operation.**

**Design calculations.**

In our pump design we used following elements: Spinner and case for the spinner.

The Spinner:

In our pump we designed pump with 3 bearings at the end with 120 degree angle between them.

Bearing holder height 9mm.

Bearing stand height 4.2mm.

Height of the base 3.6mm.

Height of the motor shaft 12.4mm.

**Overall radius is 28.5mm.**

Radius bearing holder is 3.965mm.

Radius rotor hole 2.5 mm.

Radius screw hole 2mm.

Nut hole dimension 4mmx7.5mm.

**The case for the spinner:**

In this design we had layers with different heights.

Height of the first layer 2mm.

Height of second layer 3mm.

Overall height 17mm.

Overall radius 39mm.

Radius rotor hole 11mm.

Radius of the first layer 28.5mm.

Radius of the second layer 36mm.

Radius of the third layer 39mm .

Radius screw hole 1.575mm.

Radius tube hole 3.651mm.

We have used the following measurements in order, to provide small, cheap pump, which can sustain minimal flow rate of 0.5L, and which can pump for maximum time without damage. Also, bearing, rotor and nut holders are designed in such way, that they are little bit bigger, due to the fact, that there is small error after printed plastic changes his volume slightly. Also, the shape of our spinner gave us opportunity to have maximum rotates per minute, while not consuming much plastic. Also, there is some cavity between walls in casing, supported by

**Code logic flow chart.**

1. **Description of pump’s design including justification of design choices.** 
   1. **Basic requirements for design**

Peristaltic pumps’ mechanism is based on squeezing a length of tubing to push fluids. The main issues that we have to focus on when we choose the pump’s design were such points as the flow rate of liquid, reasonable waste of PLA material during printing, consistency of printed components of the pump with the motor setup for moving and secure connection of them. In this case, a key point for our pump was the capability of providing a flow rate of from 0.5 to 1 L/min assuming water as the main liquid. First of all, our pump with chosen parameters was 3d-modeled in an appropriate CAD(Computer-aided design) package. We used Rhinoceros (Rhino 3D) CAD application software developed by Robert McNeel and associates.

* 1. **Rotor design**

Considering the rotor, we minimized the overall size as possible (28.5 mm). The smaller the radius of the rotating object, the smaller the momentum that facilitates the work of the motor. We chose the rotor that has 3 rollers. We would choose 4 or more rollers. However, for consideration, this amount was optimum in case of PLA waste and Transferring fluid volume. For the same reason, we cut the edges of the rotor between rollers that are not so essential for the work. We made the upwelling on the surface of the base, that is, the height of the stands of rollers is bigger than the height of the base for 0.6 mm. It was for the precaution just in case the rollers collapsed with other components. The radius of the hole for the engine was 2,5 mm that did not fit the engine itself. So we cut the hole a little bit to reach consistency.

* 1. **Casing design**

Moving to the casing component, there are 4 openings for screws that connect the engine and the casing component. These crews have their heads that may be a barrier for rolling of the rotors. To solve this problem we did the upwelling on the surface of the first level base. That is if the height of the first level is 2 mm, the second one's height is 5 mm. Overall radius of the casing is 39 mm. Also, we made the cut on the lower part of the casing in the form of an arc for the same reason as it was mentioned in the cut of the rotor (for minimizing of the waste of PLA used). We accurately chose the radius of the holes for the tubings. Because they have not to move, stand static during the liquid transferring. In our case, it is 1.575 mm.

* 1. **Material and printing**

Referring to the material of our pump, we initially tried to use ABS filaments for printing. However, when the lower parts were printed it started to deform itself. So taking into consideration the fact that the material is not consistent, in the end, we decided to use PLA.

For the testing of our design for printing, we conducted the preparation in CURA software. All the parameters were set and standardized. The time, needed for printing was checked. For the small one that is rotor 42 min, for the big one that is casing 2 and half hours were needed.

1. **Description of the controlling process including justification of coding/implementation choices.**

Program starts with an execution of main code, at the same time call back function GPIO.add\_event\_detect(24, GPIO.RISING, callback = level\_control)

is running in second thread, because we need an immediate response to tank 3 level overflow.

It allows us to perform such activity: wherever line of main code is executed, this function will call level\_control function when Tank 3 becomes full/float switch is up, and main code is interrupted/stopped until float switch is down/tank 3 is empty.

def level\_control(channel):

if GPIO.input(24):

print ("Tank 3 is full")

pi.set\_PWM\_dutycycle(STEP, 0) # PWM off

else:

print ("Everyting is fine")

motor\_online()

GPIO.add\_event\_detect(24, GPIO.RISING, callback = level\_control

def motor\_online():

pps = 300

for n in range (8):

pi.set\_PWM\_dutycycle(STEP, 128) # PWM 1/2 On 1/2 Off, 128

pi.set\_PWM\_frequency(STEP, pps)

pps = pps + 100

sleep(1.5)

There is defined function motor\_online() which controls motor speed. It increases speed of motor in a “for” loop with a 1.5 seconds interval

We used another function analog\_read as it was in the manual to measure pressure difference.

def analog\_read(channel):

spi.max\_speed\_hz = 1350000

r = spi.xfer2([1, (8 + channel) << 4, 0])

adc\_out = ((r[1]&3) << 8) + r[2]

return adc\_out

Then we are entering the while loop with try. It executes flow rate and pressure difference code. And whether pressure difference is high or low to maintain solenoid.

try:

while True:

rate\_cnt = 0

pulses = 0

time\_start = time.time()

while pulses <= 5:

gpio\_cur = GPIO.input(flowrate\_pin)

if gpio\_cur != 0 and gpio\_cur != gpio\_last:

pulses = pulses + 1

gpio\_last = gpio\_cur

rate\_cnt += 1

time\_end = time.time()

output = rate\_cnt/(time\_end-time\_start)

print("Water flow is", output)

reading = analog\_read(0)

print("Pressure is", reading))

if reading == 30:

GPIO.output(valve\_pin,GPIO.HIGH)

else:

GPIO.output(valve\_pin,GPIO.LOW)

sleep(0.1)

1. **Results from all benchmark cases**
2. As it was planned, we launched the water of amount 0.5 L from Tank 1 to Tank 2. When exact 0.5 L was successfully transferred stopwatch showed 57.15 seconds.
3. When we launch the pump again, we observed that for 2 minutes 1.128 L of water was transferred from Tank 1 to Tank 2. That is, the flow rate is about 0.5L per minute. Respectively, after 2.5 minutes we stopped the flow. In the result 1.46L of water was in Tank 2 overall.
4. 0.8L/min rate was not up to our system, as it cannot sustain, maintain this pressure in the tubes. So, we completed an alternative version of the task. That is, as initially, the flow of water of 0.515L/min rate was launched steadily for 3 minutes.
5. We started fluid from Tank 1 to Tank 2. We had overflow mechanism that directed to Tank 3. At Main 2 pipe the flow rate was 0.516L/min. And the pressure value was 21kPa. We observed water level in Tank 3 in the result. As the Normal operation the valve between Bypass 1 and Bypass 2 was closed. It was automatically opened when Blockage state occurred. Then, the level of water in Tank 3 was controlled too. It rose above the sustainable level, the system stopped working automatically. However, when we reduced the amount of water, it started working again.
6. **Results from measuring cases**
7. Pressure values from the pressure sensor in 1 minute
8. The system is working in Normal mode (Main2 pipe is used)

|  |  |  |
| --- | --- | --- |
| **№** | **Time (s)** | **Pressure (kPa)** |
| 1 | 10 | 20 |
| 2 | 20 | 22 |
| 3 | 30 | 22 |
| 4 | 40 | 20 |
| 5 | 50 | 25 |
| 6 | 60 | 23 |

1. The system is working in bypass mode (Bypass 1&2 pipes are used)

|  |  |  |
| --- | --- | --- |
| **№** | **Time (s)** | **Pressure (kPa)** |
| 1 | 10 | 52 |
| 2 | 20 | 47 |
| 3 | 30 | 48 |
| 4 | 40 | 47 |
| 5 | 50 | 49 |
| 6 | 60 | 50 |

1. Measurement of flow rate over a period of 5 minutes:

|  |  |  |
| --- | --- | --- |
| **№** | **Time (min)** | **Flow rate (L/min)** |
| 1 | 0.5 | 0.54 |
| 2 | 1 | 0.55 |
| 3 | 1.5 | 0.54 |
| 4 | 2 | 0.54 |
| 5 | 2.5 | 0.55 |
| 6 | 3 | 0.55 |
| 7 | 3.5 | 0.55 |
| 8 | 4 | 0.54 |
| 9 | 4.5 | 0.54 |
| 10 | 5 | 0.54 |

1. Measurement of viscosity

Hagen-Poiseuille law:

(m3/s)

Q – Volumetric flow rate

P- Pressure difference

- Radius

- Fluid viscosity

- Length of tubing

Then, we derive from the equation above the equation of viscosity:

(Pa\*s)

Calculating the exact value of viscosity gives:

mPa\*s

1. **Images from all stages of construction, assembly and operation.**
2. **Challenges we faced and how we solved it.**
3. We did not have enough fitting from solenoid valve to tube. So we had to improvise with glue and other tubes a little thicker. We inserted a small tube into a bigger tube and then it was all glued together.
4. When we connected the pressure-sensor it was difficult to make it not leak. It took us 4-5 attempts to make it not leak. Unfortunately, during the demonstration, the sensor itself was sealed and therefore it did not give the measurements. But after the presentation we replaced the sensor and fixed this problem.
5. Water leak: Everything was leaking, but a couple of layers of glue and electrical tape solved this problem.
6. Design: The design was good, but we had a problem in that the rotor was tight for the hole. Solution: to melt the edge a little and insert pieces of paper to other places.
7. The problem with the code. During the presentation, we had a small error with the code due to which the water level sensor did not work, but we fixed by reimplementing the code for it.
8. Flow sensor: we attached the flow sensor to the wrong place, and therefore our motor could not start spinning without twisting, and our water did not start moving without suction.

**Conclusion**

**Reference sources**

**Appendices**