Construction of Vertical Scanner for Laser Analysis of Gel Samples

Jiří Pech[1], Milan Novák[2], Ladislav Ptáček[3], Jana Kalová[4]

Faculty of Science, University of South Bohemia, Branišovská 1760, 370 05 České Budějovice, Czech Republic, e-mail: pechj@prf.jcu.cz[1], novis@prf.jcu.cz[2], lptacek@prf.jcu.cz[3], jkalova@prf.jcu.cz[4]

Abstract: **This paper is focused on the construction of vertical scanner with linear translation used for gel samples analysis via laser. Scanner construction is based on the construction of RepRap 3D printer. The paper describes the process of choosing this type of construction and then the construction itself. Advantages and disadvantages of this solution are recorded too.**
Keywords: **Industry 4.0, 3D prints, Arduino, Raspberry Pi, G-Code.**

I. Introduction

This paper describes the design, the construction and the testing of the vertical scanner, which was built for the Institute of the Physic and Biophysics [1]

The construction is based on the construction of the 3D printer Rep Rap Rebelix (figure 1) founded on the Rambo motherboard, supported by the Marlin firmware. All of the components are open software or open hardware, so we could use them and modify them as we needed.

The single-board computer Raspberry Pi with the attached touch display is used as the control computer. The control program is written in the Python language.

It means, that this article describes the applied research.

II. Task

Institute of Physics and Biophysics [1] assigned us following task: "Design a vertical scanner with linear translation that will be used for gel samples analysis via laser. The scanner has to consist of a frame, which will be moving in two directions: left - right and up - down. But in the direction backwards – forwards it will be stable. Next task is to cover the sample fully and to not stop at any position." The Physicists searched the Internet, but they haven't found any satisfying solution. Only one of the found solutions was little bit fair but it was very expensive (about 1000 Euro). In addition, this scanner's recommended working position is horizontal, and we need the vertical working position because our lasers have horizontal rays.

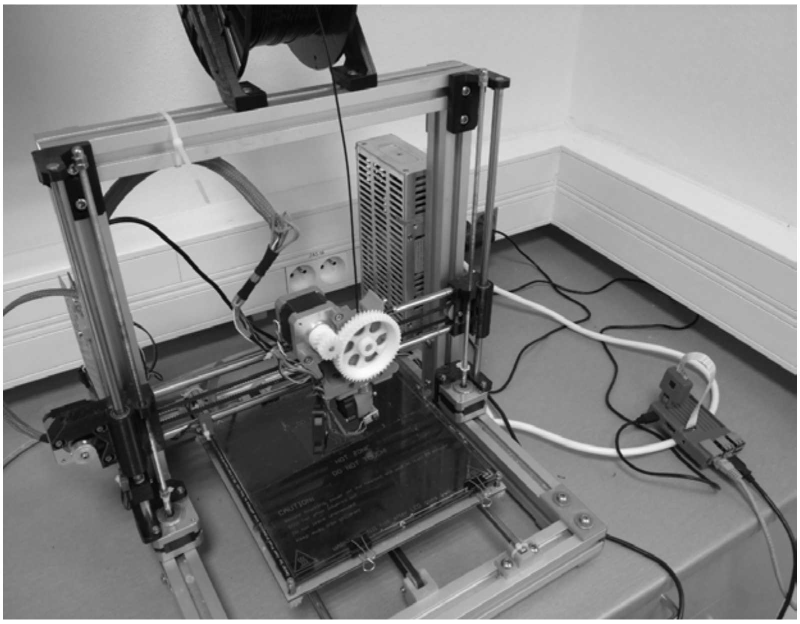

Fig. 1. 3D printer Rep Rap Rebelix with the control computer Raspberry Pi (on the right side).

So, it was clear, that if we want to satisfy them, we have to design absolutely new construction.

III. Methods

After the acceptance of this challenge, we started thinking about the construction of the scanner. First, we thought about using linear motor, but we discovered the fact that all of the available linear motors are very expensive. So, we decided to use a motor with a screw-thread.

We have the 3D printer Rep Rap RepRap model Rebelix [2] at our department. Its construction enables precise movement of the instrument in three dimensions. The extruder of the filament represents the instrument in this case.

We had the idea, that we have similar problem, but we need to move only in two dimensions – left - right and up - down.

So, we decided, that we base the project solution on this 3D printer construction. This was not a problem because the Rebelix printer is designed as the open hardware. It also means that the list of the parts used is available on the Internet.

We prepared the initial scheme of a scanner and ordered all the needed parts. As the control board we have chosen the Mini-Rambo [3] board, the new version 1.3 (Figure 2). Its delivery from the USA was the longest waiting period during the construction.

The Mini-Rambo is as a matter of fact Arduino Mega with included drivers for the five motors. So, the advantage of this board is the possibility to use Arduino IDE to program it. First, we thought, that we would write completely new program to control our scanner. Later we decided to use the original firmware, which is used to control 3D printers. Again, it was not a problem, because it is open-source software.

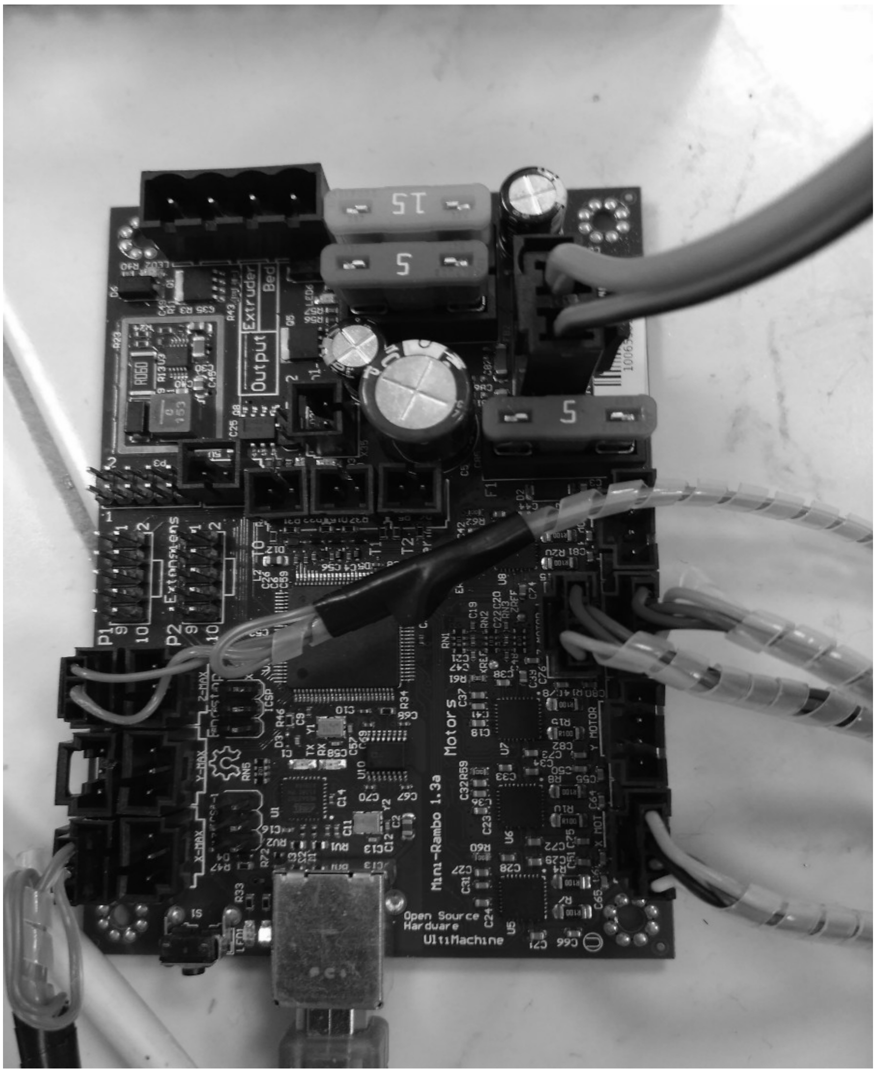

Fig. 2. Control board Mini-Rambo 1.3.

This means, that we control this board by the G-Code[1] commands. We needed leading computer, which would send the G-Code commands to the board.

IV. Construction

After arrival of all the parts (except the board) we began the construction. With the help of our technicians we prepared stake, threads and sockets. In comparison to construction of 3D printer 3 years ago, we had simpler task because there are motors with connected screw-thread available now, so it is not needed to join screw-thread to motor.

Next, we needed the parts printed by 3D printer. We used some parts directly from 3D printer, but we also needed to prepare and print some new parts. We made them by the online support CAD software Tinkercad [6]. Then we printed all the parts on our printer.

Then it was necessary to write a code to send desired G-Code to the control board. It was required to cover the entire frame with gel by laser ray by moving the scanner. So, we decided to use Lissajous curve for the movement of the frame.

1 G-Code is the language for the digitally controlled machines. I will show an example of the G-Code later.

As the programming language we selected Python for its simplicity and good portability and as the operating system for the control computer we used Linux. The computer is connected by USB cable type A.

The advantage of the presented solution was that we could test the first versions of our program on the 3D printer. It was important because we had all construction complete except for the board, which arrived two months later.

Finally, three months later, we had the construction complete (Figure 3 and 4) and we could test it. The meanings of the items in the figure 3 are:

1. Stepper motor for the movement of the frame in the x axis direction.
2. Stepper motor for the movement of the frame in the z axis direction.
3. Coulisse for the screw-threads.
4. Screw-threads.
5. Leading threads.
6. Coulisse for the leading threads.
7. Gear transmission for the movement of the frame.
8. Frame for the holding of the sample.
9. Aluminum profile for the stability of construction.

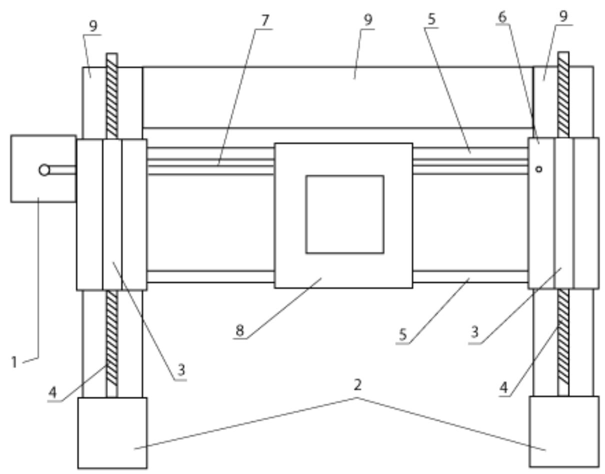

Fig. 3. Scheme of the scanner.

V. Testing and Tuning

All the problems, which we solved during testing the scanner, are described in this chapter.

First big problem was the selection of the correct firmware for the Mini-Rambo board. First, we tried the Marlin firmware, but we had a problem with end-stops[2]. So, we tried alternate firmware RAMPS, but it was even worse. So, we used the development version of the Marlin finally and we had to make some changes in the source code. For example, we needed to disable setting for unused y axis (forwards – backwards direction). We had to change level of the current for motors because with the original settings the motors are overheated.

We had to test the developed program, which generated G-Code, carefully. First, we used relative coordinates but finally we used absolute coordinates to eliminate rounding induced errors. Here is an example of the source code and example of the generated G-code.

VI. Example of the Source Code

```
import serial
ramecek =
serial.Serial("/dev/ttyACM0",baudrate=115
200,
timeout=0)
ramecek.write("G21\n".encode())
XS = 80
ZS= 40
stred="G0 X"+str(XS)+" Z"+str(ZS)+"\n"
ramecek.write("G90\n".encode())
ramecek.write("M92 X100 Z390\n".encode())
ramecek.write("G28 X Z\n".encode())
ramecek.write(stred.encode())
```

This code sets up a communication between computer and control board and then sends G-Code commands via USB cable. The whole program is much bigger and complicated and has about 320 lines of the code.

VI. Example of the G-Code

And here is the example of the G-Code for the moving frame of the scanner:

```
G0 X71.601 Z32.191
G0 X73.018 Z31.602
G0 X74.09 Z30.734
G0 X74.694 Z29.738
```

These are the commands for the movement of the frame along the x axis (right - left direction) and along z axis (up - down direction). For the matter of the interest there are about 2000 G-Code commands for one complete covering of the frame with Lissajous curve.

VII. Problems and Calibration

The biggest problem presents the requirement to make small pauses between commands. It is not possible to send a command after a command without the pauses because the control computer generates the commands faster than the scanner can perform them. The duration of the pauses depends on four variables – two dimensions of the sample and two parameters of the Lissajous curve. If the pauses are too small, the control board can't correctly process G-Code to the scanner in time or the scanner can't execute (more precisely finish) all the commands. In opposite, if the pause is too big, the movement is abrupt and then the scanner shakes.

So, we have made the algorithms in which the system is learning itself. First, we estimate the duration of the pauses according to all enumerated parameters and then the

2 End-stops are the tools which prevent the moving parts from getting out of the leading bars

calibration starts. Control computer tries to work with estimated pauses and reads answers from the control board. If there is any error, the system stops for a while and prolongs the pause. First calibration ends when the system passes all the commands without any error five times. When there appears no error during all the calibration, the pauses are cut by 30 % and calibration repeats.

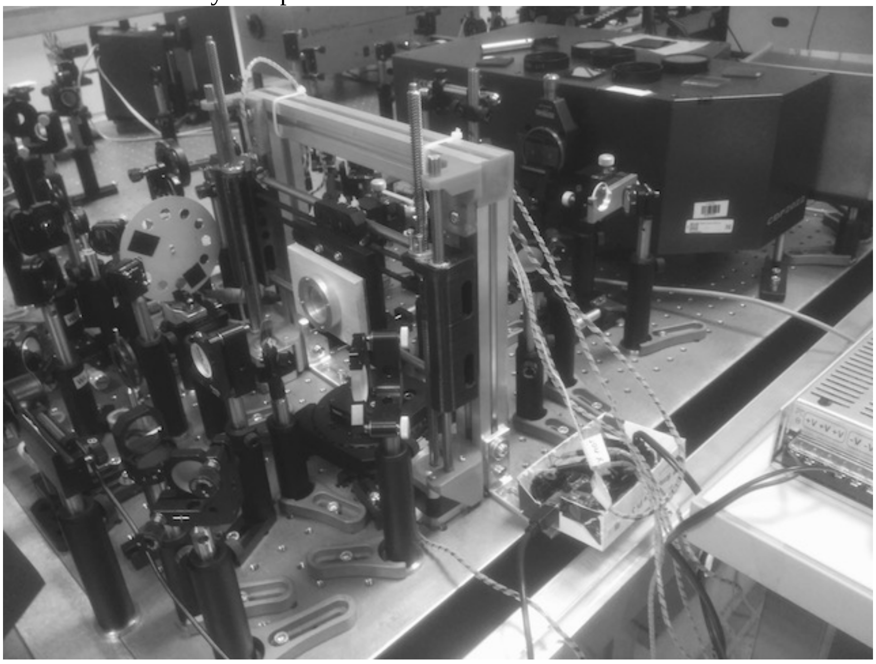

Fig. 4. Final form of the scanner.

After this first calibration or calibrations there comes the second calibration. The whole program runs once more with the same conditions as during the real use later. The system watches for the errors again and if any appear, the appropriate pause is prolonged. Again, the process ends when no error appears five times.

After second calibration computer notifies the user that everything is ready to run, and user can start the lasers.

During all the development we worked on the normal PC, but we prepared one-board computer Raspberry Pi 2 with attached touch display as the control computer for the real use.

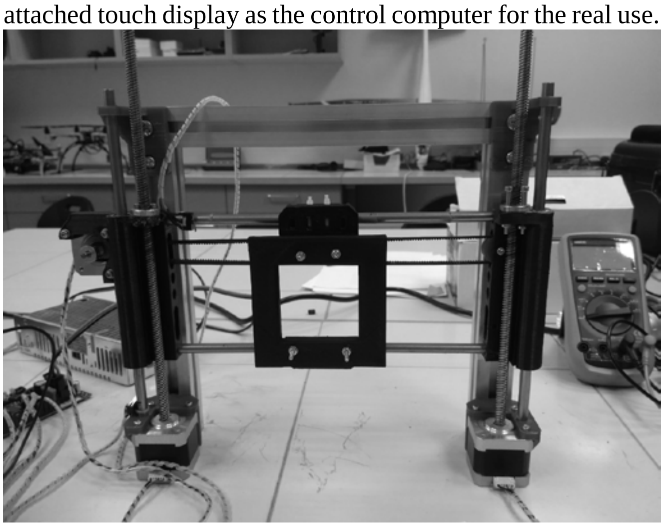

Fig. 5. The scanner on the working position.

VIII. Conclusion

We managed to make the required scanner, which was named the Lissajouse scanner. The submitter is satisfied with our work and scanner will serve to the research. The system partly fulfils the standards of the Industry 4.0 because it is self-learning. It would have to read the dimension and the position of the sample itself to meet the standards completely.

The Lissajouse scanner is used for moving of this kind of the sample where is necessary to excite the researched molecules so that no previous history of the sample (e.g. illumination) has any meaning. (e.g. the gel situated in the test-glass) The excited molecules in the sample are researched with the methods called pump-and-probe, which is common in the scope femtosecond optical spectroscopy.

We got very valuable experiences during this project and these experiences can be used in following constructions. We will be very pleased if it helps anybody to make any similar construction.

References

[1] Institute of Physics and Biophysics [online]. České Budějovice, 2017 [cit. 2017-04-03]. Available from: http://www.prf.jcu.cz/en/ufy/

[2] Marlin - RepRapWiki [online]. 2016 [cit. 2017-04-03]. Available from: http://reprap.org/wiki/Marlin

[3] Mini-Rambo 1.3 - Ultimachine [online]. 2017 [cit. 2017-04-03]. Available from: https://ultimachine.com/products/mini-rambo-1-3

[4] Raspberry Pi - Teach, Learn and Make with RaspBerry Pi [online]. Raspberry Pi Foundation, 2017 [cit. 2017-04-03]. Available from: https://www.raspberrypi.org/

[5] Rebelix - 3D tiskárna [online]. Praha, 2017 [cit. 2017-04-03]. Available from: http://reprap4u.cz/rebelix/

[6] Tinkercad: Create 3D digital designs with online CAD [online]. Autodesk, 2017 [cit. 2017-04-04]. Available from: https://www.tinkercad.com