THE CRANE MANIPULATOR FOR THE 3D PRINTERS

Khmil Denys

Bachelor Degree Programme (3), FEEC BUT

E-mail: xkhmil00@stud.feec.vutbr.cz

Supervised by: Jakub Arm

E-mail: Jakub.Arm@vut.cz

Abstract: The aim of this work is to design the architecture of a manipulator that takes the finished product out of the 3D printer. This article deals with the whole concept of the machine, its design in the CAD program Siemens NX12, selection of parts, software development, and real prototype assembling. The manipulator is driven by LinuxCNC ecosystem installed on a single-board computer Beaglebone Black. To control the machine, a client PC application was developed. The manipulator movements are achieved using the tooth rail gearing and closed loop stepper motors Nema23.

Keywords: Manipulator, 3D Printer, CAD, LinuxCNC, Beaglebone Black.

1 INTRODUCTION

Nowadays 3D printers are commonly used for manufacturing, prototyping or hobby. The main disadvantage of traditional 3D printers is the impossibility of printing a large number of elements without human intervention. To start a new printing, it is necessary for the operator to pull out the printed part and start another print.

An analysis of the current situation in this area has revealed that there are several solutions to this problem. The first one is printing on a belt conveyor, which rotates after printing finishes. This frees space for further printing. The second solution is to use a 6-axis robotic manipulator for removing printed parts. Each of these solutions has its disadvantages. The disadvantage of the first solution is the unstable printing surface which can cause printing problems. For the second solution, it is the high cost and limited manipulating range. [1]

This work deals with the design of a machine that solves this problem in a different way. During the work, a manipulator will be designed. This manipulator removes the printing surface with the printed parts from the printer and places a new one for further printing. All the manipulations are based on the signals from the printers.

2 DESIGN OF THE MANIPULATOR

A prototype of the manipulator is designed to operate with up to four printers. This gives a working area of 1 x 1 m. In case of a larger number of printers, the design allows extension of the working area by another 1 m in each direction.

2.1 MECHANICS

The manipulator is designed as a 3-axis CNC machine, where X is a horizontal axis, Y - vertical, and Z axis is presented as the mechanism for taking the printing surface from the printer. The machine is mounted to the front side of the printers, so it has access to each one of them.

Figure 1 shows a model of the manipulator that was created in the program Siemens NX12. For the construction of the manipulator, modular aluminum profiles 30 x 30 mm are used. These profiles give good strength and simplicity of manipulator assembling.

Linear motion is provided by a Hiwin HGR15 linear guideway and rack and pinion. Using these components, it is possible to achieve high speed and accuracy of movement over long distances. Another advantage is the possibility of expanding the working area of the machine. [2]

The printing surface pick-up mechanism designed as a carriage which can move 100 mm in both directions from the zero position. This makes it possible to work with printers from both sides of the manipulator. The movement of the printing surface on the carriage is implemented using a belt conveyor which is designed as two belts, one on each side of the carriage. When the manipulator reaches the printer coordinates, Z axis carriage moves inside of the printer and uses a belt conveyor to remove or place the printing surface.

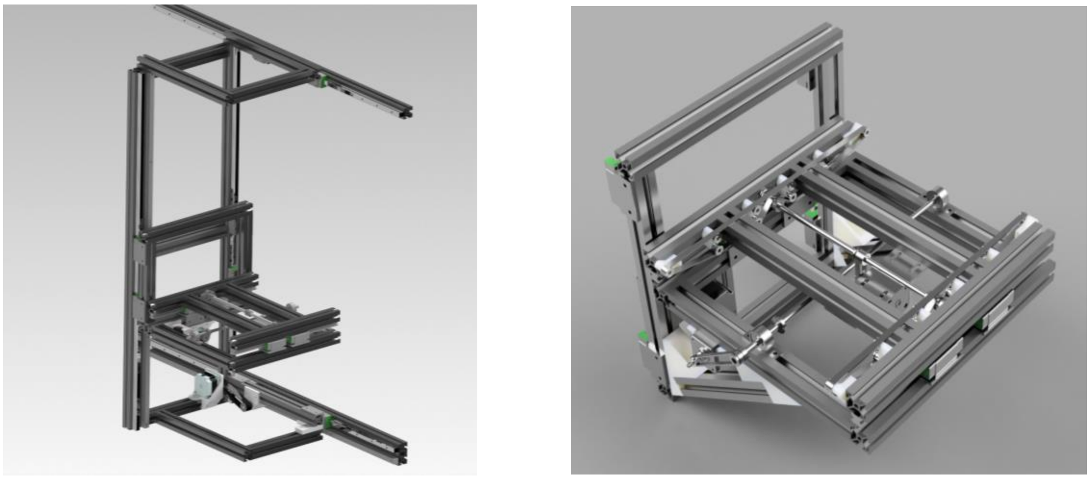


Figure 1: Manipulator and Z axis carriage

2.2 MOTORS

For the design of the manipulator, closed loop stepper motors with a torque of 0.9 N.m from the company Makerbase are used. These motors have drivers mounted on the back side of the motor and have a built-in position sensor to provide feedback.

In this case, the feedback is implemented using a magnet located on the shaft and magnetic angle sensors. Based on the output of these sensors, the driver evaluates whether the required movement has occurred. If it is not, driver will try to compensate the difference or change its state to an error.

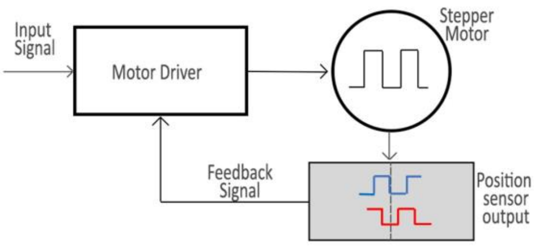


Figure 2: Closed loop stepper motor principle

The advantage of magnetic encoder against optical, that are typically installed on these motors, is smaller size, better reliability, and better accuracy (2048 pulses per revolution versus 1000 pulses per revolution) [3].

Communication between the control unit and this driver can be implemented using standard STEP / DIR signals or using the serial interface SPI or UART. Communication of this type makes it possible not only to control the movement of the motor, but also to read or change the settings of the driver.

2.3 ELECTRONICS

Figure 3 shows a schematic arrangement of the electronic components of the machine. The machine control is divided into main and secondary control components. The main component controls the movements in X and Y axes while the secondary controls Z-axis mechanism.

The main control component is Beaglebone Black. The single-board computer was chosen because of the possibility of implementing more complex service modules such as industrial I/O modules or a motor driver with industrial communication protocols support (EtherCAT, Powerlink etc.).

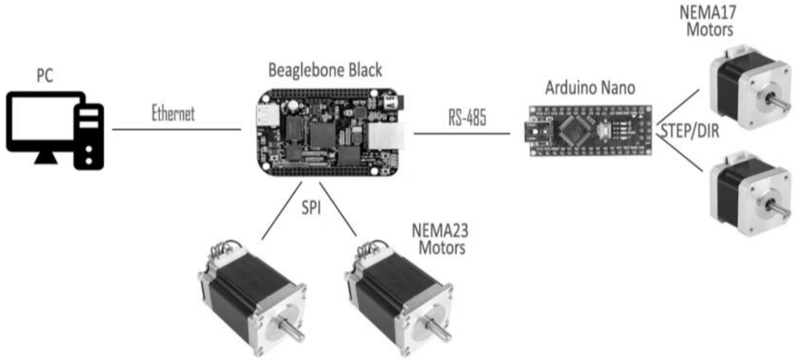


Figure 3: Electronic component arrangement

Arduino Nano was chosen as a secondary control component. Based on a command from the Beaglebone and the values from the sensors, Arduino will control the movements of the Z-axis carriage and the belt conveyor. Two inductive sensors are used for detection of the carriage limit positions. Capacitive sensors are used for detection of the printing surface on a belt conveyor, which allows using not only metal surfaces but also glass.

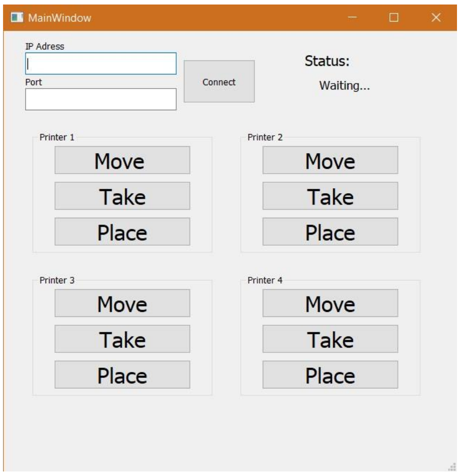
For the Arduino, a printed circuit board was designed, which makes it possible to connect up to 5 sensors using optocouplers and two stepper motors.

2.4 SOFTWARE

For controlling the manipulator, a client computer application was developed. Interface of this program is shown in Figure 4. The program is designed to work with four printers, each of which ha preset coordinates. Using the buttons, we can choose whether the machine must remove the printed part, place a new printing surface or just move to the printer. From the client app, commands are sent to the Beaglebone Black single board computer using the TCP/IP protocol in a structure format (X, Y, direction), where X and Y are coordinates and the direction determines whether the printing surface will be removed or placed.

LinuxCNC was installed on the Beaglebone, which is controlled by a Python script. The script needs a few more files for proper functionality. The overall structure of the Beaglebone software is shown in Figure 4. The first file is a *.bbio* file. This file defines functions of the individual pins of the Beaglebone. Next one used is an *.ini* file, in this file the parameters for LinuxCNC are set, such as the number of machine axes, maximum speed and acceleration of movements, etc. The last one is the *.hal* file, which serves as a layer between software and hardware and allows the script to work with various hardware types.

After running the script, first of all, all the required settings are set and then LinuxCNC is started. At the beginning of each movement, the possibility of starting the machine is checked (the Estop button is not pressed and machine is not in fault state). If all conditions are good, the manipulator starts moving to the specified coordinates. After reaching the position, Beaglebone sends a command for Arduino subsystem using the RS-485 bus [4].



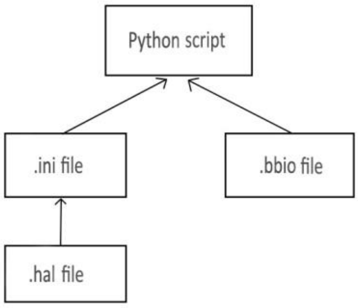


Figure 4: Client user interface and software structure

Based on the command from the Beaglebone, Arduino turns on the motor to move the carriage into the printer to a precise, predetermined location. Then it starts the sequence of removing the printing surface with the printed part, i.e. starts the movement of the belt conveyor and will wait until it receives a signal that the target position of the pad has been reached. When a pad is detected on the cart, the Arduino will send a confirmation to the Beaglebone that the command has been properly performed and the machine will continue to move. In case the pad is not detected within a certain time, an error message will be sent and the whole machine will be stopped.

3 CONCLUSION

During the work, the mechanical principle of the manipulator was designed. Based on the calculations, the necessary components were selected. Using selected components, a 3D model of the manipulator was created. According to this model a prototype of the machine will be assembled. As the next step, a client application for the computer, software for Beaglebone and Arduino, and communication between them were developed. The next step will be to build and test a physical prototype of the machine.

The developed machine makes it possible to automatically collect printed parts and start next printing. Using this machine instead of human operator, some benefits arises, such as time efficiency and cost reduction. Since the components are chosen from the hobby and semiprofessional domain, it is not yet possible to quantify the financial benefit. A distinct advantage of this manipulator is the possibility to extend a working area for more printers.

REFERENCES

[1] Turn Your 3D Printer Into a Factory: Automatically Remove Parts [online]. [Accessed 2020-03-09]. Available at: https://www.instructables.com/id/Turn-Your-3D-Printer-Into-a-Factory-Automatically-/

[2] Hiwin Datasheet [online]. [Accessed 2020-03-09]. Available at: https://www.hiwin.com/pdf/linear_guideways.pdf

[3] A1333LLETR-T Datasheet [online]. [Accessed 2020-03-09]. Available at: https://pdf1.alldatasheet.com/datasheet-pdf/view/1148490/ALLEGRO/A1333LLETR-T.html

[4] Machinekit Documentation [online]. [Accessed 2020-03-12]. Available at: http://www.machinekit.io/docs/