**INTRODUCTION**

An interest in how things work and watching programs based on science and how things are made, have been a big inspiration for this project. However, the main source of inspiration for this project came from watching YouTube videos. These videos included topics such as the writing of algorithms, specific the video by Buekban, C., 2014. *Youtube.* [Online] Available at: <https://www.youtube.com/watch?v=WShhcGl3A6g>; DIY 3D printers, Arduino projects, specific video by GUY, T. E., 2018. *Youtube.* [Online] Available at: <https://www.youtube.com/watch?v=pe71pKhP9ME&t=2s>, and other similar topics. Some of these videos included clips of self-made / DIY printers, which could be improved upon. Thus, the idea was formed of conducting research, writing an algorithm to convert digital images into physical images and building a DIY printer.

# BACKGROUND INFORMATION

Some of the earliest inspiration for writing an algorithm to convert digital images in physical images, came from path finding algorithms such as A\*, which finds the shortest distance between all the points. Many of the later ideas came from the flood searching algorithm. This algorithm provided the bases on which the digital images were scanned to detect dots. The algorithm starts at a point and checks all the spaces around it for a potential dot and also has path finding applications. The distance algorithm’s inspiration came from the popular travelling salesman problem where the goal is to create an algorithm that finds the shortest path between all the dots.

The review of videos online provided a lot of insight and information on how to build the physical printer, specifically the video RCLifeOn, 2017. *Youtube.* [Online] Available at: <https://www.youtube.com/watch?v=XIk-w5OSVh8> . This video shows how DIY printer kits are assembles and thereafter reviewed. This was a great bonus since it provided insight into what an end user would expect from a DIY printer. Using COM ports enables one to open and connect to a device using serial communication, which made it easy to use wireless communication to transmit instructions to the DIY printer.

The main challenge in writing this algorithm, was to create an algorithm that detects and sorts all the points the printer will have to print.

The results from the data analysis (survey) indicated the following as being a high priority for users:

|  |  |  |
| --- | --- | --- |
|  | | Priority % |
| 1 | Quality | 87 |
| 2 | Quantity | 80 |
| 3 | Printing Abstract Image | 79 |

# Problem

The problem to be solved, is to determine whether it is possible to successfully write an algorithm to convert digital images that the user draws in the created software, into send able instructions that will be transmitted to a DIY printer.

**AIM**

The aim is to increase awareness and understanding of how an algorithm can be written and applied to a DIY printer application that can effectively, reliably scan and reproduce images.

# Hypothesis

Is it possible to write an algorithm that can convert a digital image on the user’s computer into instructions that can be transmitted (sent) to a home-made printer in order for it to convert the virtual image into an actual 2D image on a page?

# method including variables

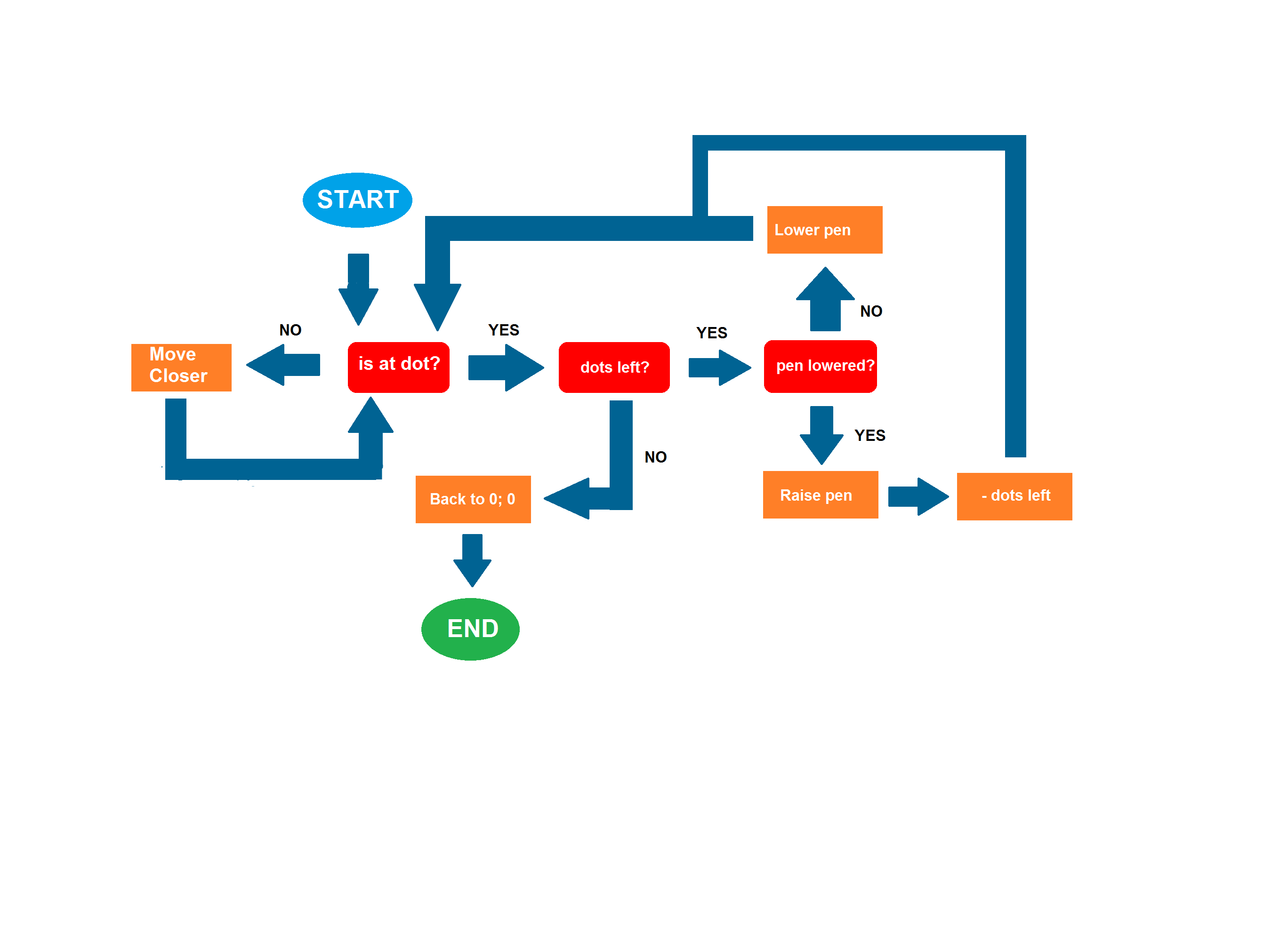
The application is written in C# and utilizes a pre-existing library to control and communicate with the stepper motors in the DIY printer. The main method creating the software was based on trial and error. All the algorithms used, had one thing in common, they start by scanning for a printable dot in a linear fashion. The linear algorithm then take these dots and print them in the exact order as they were initially scanned by the computer with the exception of going from left to right, every second row instead of right to left. This was done to decrease the overall printing time.

The user interacts with the algorithm through the GUI that contains the following aspects:

* Page orientation.
* Printer information.
* Printer status.
* COM Port the printer is connected to.
* Pen position on page.
* Scan button.
* Connect button.
* Start button.
* Settings and file preferences.

An additional functionality allows the user to draw an image in the created software. This is done by checking whether the mouse button is down and where on the page the cursor is. It then uses these coordinates to draw a black pixel on an image. This image is then displayed on the GUI and represents the picture for the printer to print. The program also includes a functionality to allow the user to import pre-made images.

When the user presses the scan button the software checks each pixel of the image and then add all the black pixels (coloured pixels) it finds to an array. Next the software checks the current location of the pen (printing head) with respect to the pixel it will draw and moves the printing head accordingly to the correct position along with raising and lowering the pen as needed. Then it moves on until all the pixels have been drawn before the printing head returns to the (0; 0) coordinate. Refer to the flowchart for the actual steps that will be followed by the algorithm, in order to “read” and print an image on the DYI printer.

**FLOW CHART OF HOW THE ALGORITHM WORKS**

# Results - iNCLUDING GRAPHS AND ANALYSIS

## Results of survey (questionnaire)

A survey was conducted using a questionnaire. The sample consisted of the following respondents:

* 20 Fellow Information Technology students in Hartbeespoort High;
* 60 Information Technology students in other High schools in the region;
* 10 friends;
* 10 Information Technology Teachers and family members.

The feedback was analysed using basic statistical techniques, such as averages etc.

The following results were obtained for male and female participants:

|  |  |  |  |
| --- | --- | --- | --- |
| **Question No** | **Question** | **Male (%Avg)** | **Female (%Avg)** |
| 1 | Quality | 73 | 88 |
| 2 | Quantity | 85 | 67 |
| 3 | Abstract Image | 88 | 83 |

Results from the questionnaires for different age groups were also analysed, resulting in the following:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Question Number** | **Question** | **17 - 20** | **21 - 30** | **30+** |
| 1 | Quality | 87 | 83 | 94 |
| 2 | Quantity | 80 | 92 | 72 |
| 3 | Abstract Image | 83 | 50 | 50 |
| 4 | Printing Speed | 72 | 75 | 89 |
| 5 | Draw Own Image | 71 | 67 | 50 |
| 6 | Import Externally | 81 | 92 | 89 |
| 7 | Saving Function | 67 | 83 | 94 |
| 8 | Printing Status | 71 | 75 | 50 |
| 9 | Page Orientation | 72 | 75 | 83 |
| 10 | DIY Printer | 79 | 75 | 50 |

Refer to the graphs for a visual representation of the results obtained in the survey.

**RESULTS OF ALGORITHM TESTING**

Five different algorithms were used for testing the hypothesis. The different algorithms consisted of the following:

1. Linear (1000 ms);
2. Linear (775 ms);
3. Distance (775 ms);
4. Distance (dots); and
5. Distance (1000 ms).

### Criteria used to compare the algorithms

The following criteria were used to compare the different algorithms:

1. Quality of printing.
2. Reliability.
3. Accuracy.
4. Repeatability.

The analysis of the results of the different tests conducted with the different algorithms yielded the following:

### ANALYSIS OF RESULTS

**Survey Results**

The analysis of the results indicated that even though there are overlaps in requirements, there is also some differences in the expectations of male and female respondents.

In addition there was also a reasonable difference in the requirements of the different age groups.

The overall impression from the data analysis is that the older generations prefer quality, saving functions, printing speed and importing from an external source. Whilst quality is also important for the younger generations, they seem to focus more on printing abstract images, printing their own images and building DIY printers. In conclusion the best option would be to write an algorithm that would be able to satisfy most of these preferences.

### Testing data results

When the average of the data obtained for each algorithm is compared, the linear (775 ms) is the overall best algorithm, if measured against the criteria of quality, reliability, accuracy and repeatability.

It was therefore concluded to use this specific algorithm for solving the posed hypothesis.

# Discussion:

There were various modifications made to the algorithm and printer including:

* Changing the x – axis movement system of the printer.
* Adding a piece to decrease the amount by which the pen can “wobble”.
* Changing the GUI Layout to be more user friendly.
* Changing GUI look to be more visually appealing.
* Changing the printing speed of the printer to print faster.
* Changing the algorithm that sorts all the points that need to be drawn to decrease movements printer needs to execute.

In the end the algorithm chosen was largely influenced by the preferences of the possible users. Since the possible users did not indicate speed as being a top priority but rather chose quality as being more important, the final algorithm focused more on quality than on time.

The testing clearly indicated the (775 ms) Algorithm being the overall best algorithm that would be able to positively answer the hypothesis.

## Anomalies:

The software sometimes does not transmit the data to the printer fast enough. This results in a run time error occurring in the software.

## Limitations:

The print size that the algorithm can currently print is a size 126 x 181 (portrait) and 181 x 126 (landscape). In addition the printing speed is currently very slow.

# Conclusion:

Through testing various algorithms, it was shown that it is possible to write a reliable and working algorithm to covert digital images into instructions that can be transmitted to a DIY printer for printing physical 2 D images. Even though it does have some limitations regarding speed and printing quality, the algorithm can successfully functions as was intended and the hypothesis could be proven successfully.

# future research:

* Conduct testing trying to print dots from left to right, move down and then to print from right to left.
* Improve printing speed.

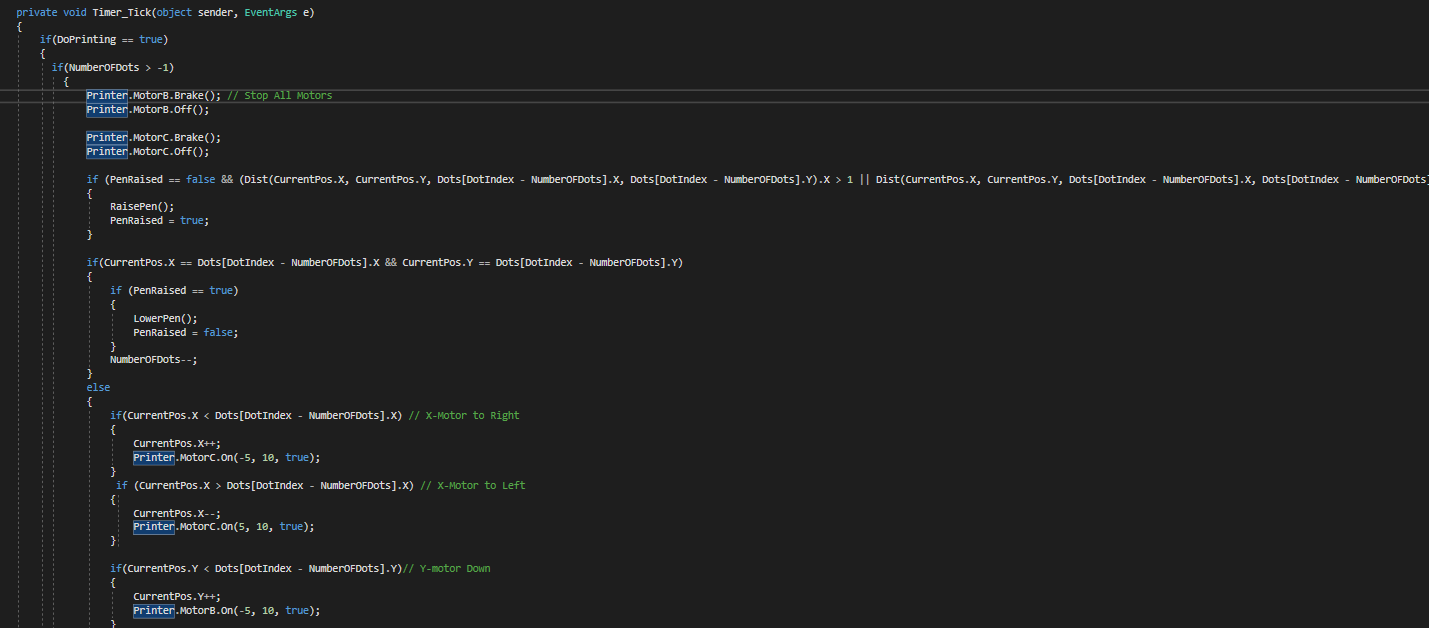
# Acknowledgements:

Mrs. C. Muller (Science Teacher): Mentor

Mrs. G. van der Merwe (Information Technology Teacher): Inspiration

Mrs. S. Smith: Aid in analyzing and interpreting data.

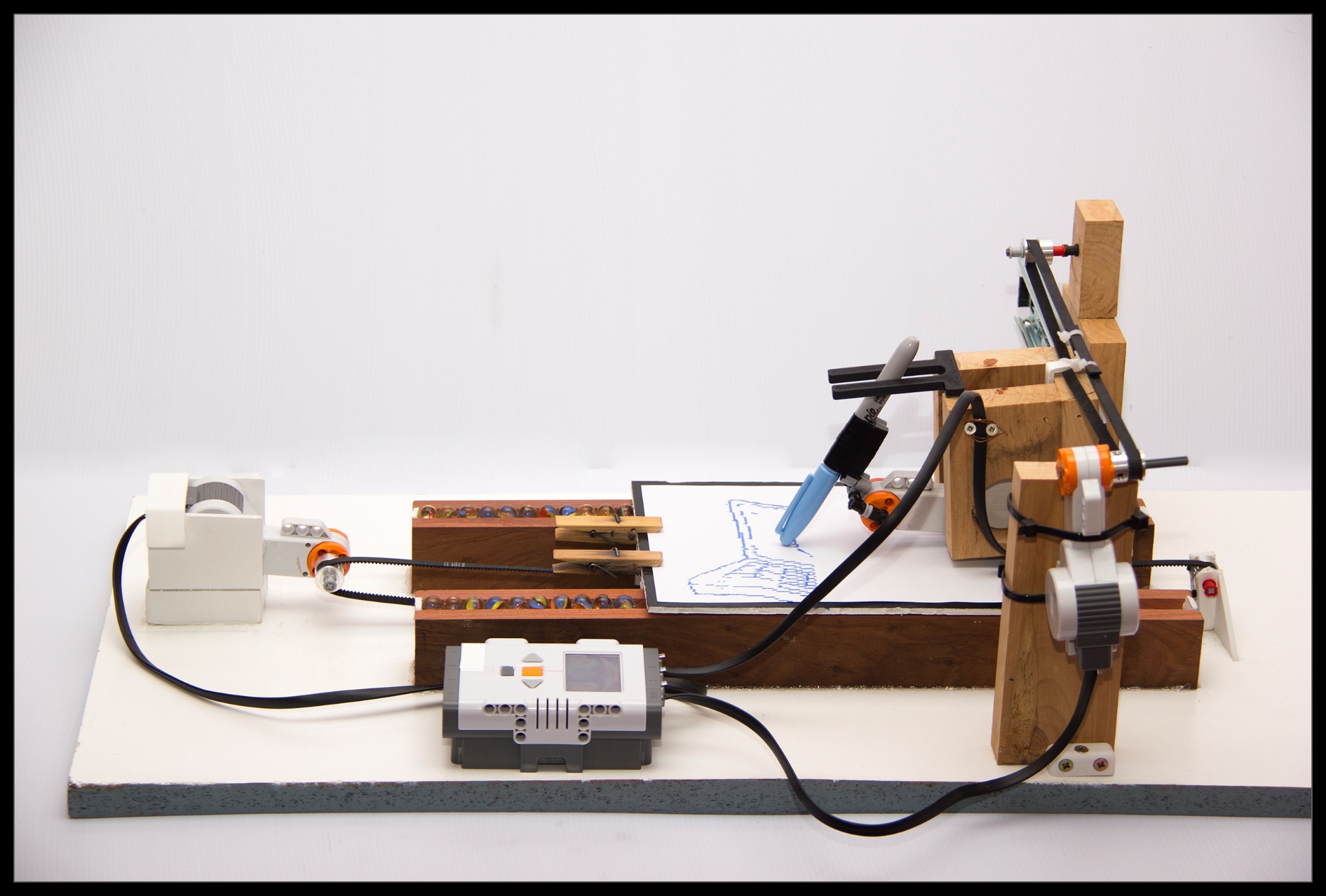
**PHOTOS**

****

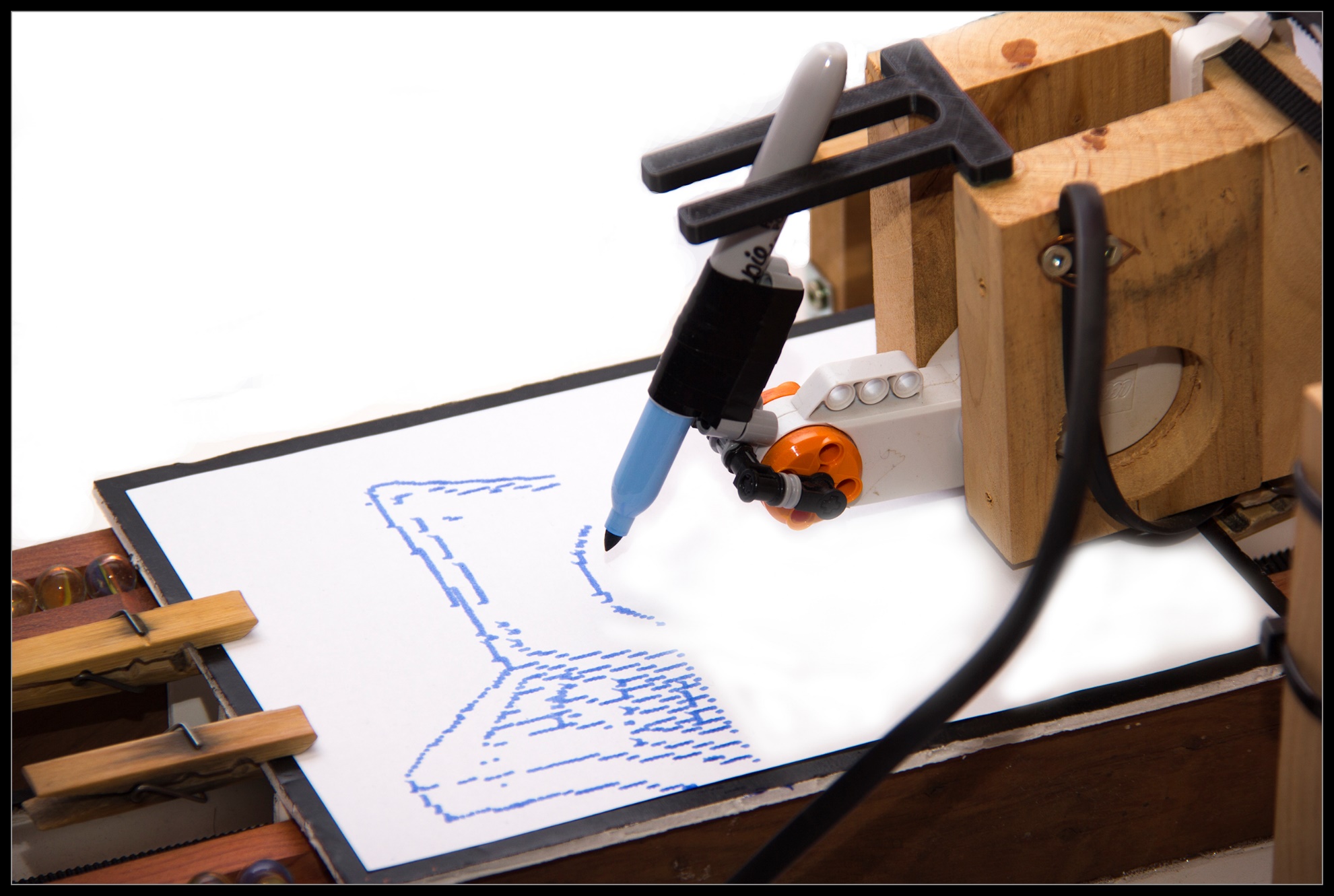
**Extract of algorithm in C#**

****

**Testing of distance algorithm (775 ms)**

****

**Side view of DIY Printer**

****

**Close-up of image printed using algorithm on DIY printer**

**Images printed with the algorithm and DIY printer**