# Interactive Vertical Plotter with face recognition

**Andreas Babic**
Mt191075 | FH - St. Pölten
St. Pölten, Austria
Mt191075@fhstp.ac.at

**Stefan Ploderer**
Mt191069 | FH – St. Pölten
St. Pölten, Austria
Mt191069@fhstp.ac.at

## ABSTRACT

Face recognition is not a novelty; however, you can use it for very many different interactive projects. In our case it is in 60cm x 120cm vertical plotter, which can print on any vector image. The plotter, or as we call it, the "plotterer", is controlled by camera module (Raspberry Pi). The idea was, during the course IAI (= interactive installations) in the Fh. St. Pölten, to create this interactive project. The printer can take different images as input and can output any vectorized image to the user Furthermore, due to the design considerations, the plotter can be attached to nearly every wall or location in every size up to several meters. Everything is highly adaptive.

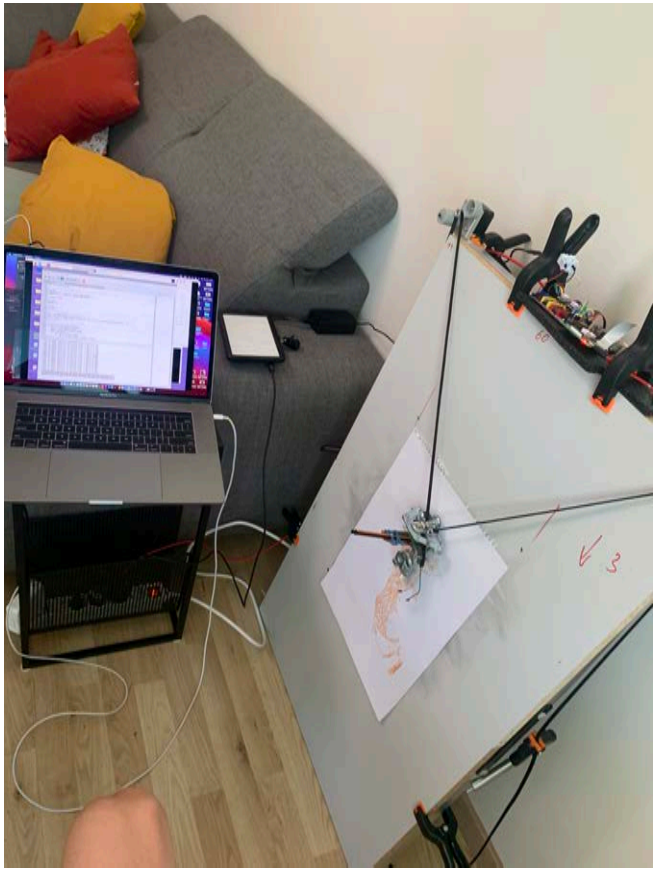
### Author Keywords
Raspberry pi; python3; face recognition; camera module

### CSS Concepts
• Human-centered computing~interactive installations

## INTRODUCTION

In the course of the subject Interactive Installations at the FH St. Pölten we set ourselves the goal to make an already existing idea of a vertical plotter interactive. Initially, the basic idea of the project was to program a face recognition with a camera module, which uses the recognized face as input and changes it into a binary edge vector with the help of an edge detection. This vector should then be used as input worthy of the vertical plotter. This works by means of two attached stepper motors, which are located at the left and right upper edge of the frame. A pen, which is held by a belt with balancing weights, then gets the input from the Raspberry Pi and starts printing.

After this initial concept, the planning process began with the first tests, which quickly presented us with major problems.



## PLANNING

The first step in the design process was to run various libraries and tests before we could determine the exact components. Primarily we wanted to get a test image from the camera module via serial to the arduino framework "polargraph", which is the control unit for the plotter.

Here several problems arose: The camera module is used for face recognition, but the quality is not high enough to apply various edge detection filters. Pixel artifacts would distort the image too much. Furthermore, it was not possible to send the specified amount of vector graphic data without overloading the serial port.

So,he planning process had to start over and the basic concept had to be revised. To keep the plotter interactive, we simplified the concept as follows: The input image for the plotter's print object should be a normal image that is vectorized using code. When this step is done, a face should serve as input for the camera module. When a face is detected, the plotter starts working. If it does not detect a face anymore, it stops and waits again for a face as input.

Then, this new plan got tested by us and showed a proof of concept, which allowed us to organize our components.

## COMPONENTS

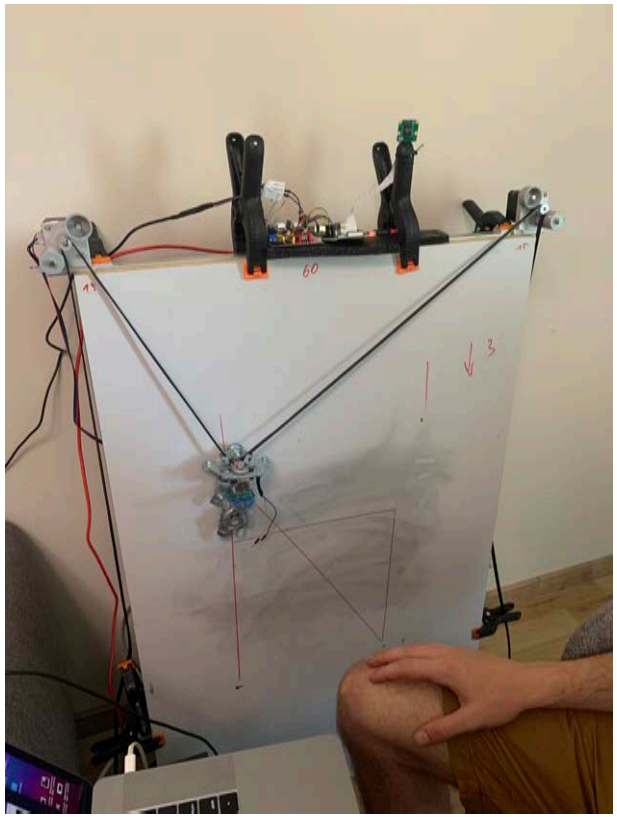
Our material list in order to successfully build the “plotterer”:

- 2x DRV8825 driver
- 2x 17HS4023 stepper motors
- 1x 5-meter belt
- 2x belt pulleys
- 1x Raspberry 4B 4GB
- Raspberry v2 8MP cameramodule
- 12 Volt power supply
- 5 Volt power supply
- 9x clamps
- Pens and Trash for Penholder
- Counterweights
- paper

## INSTALLATIONS

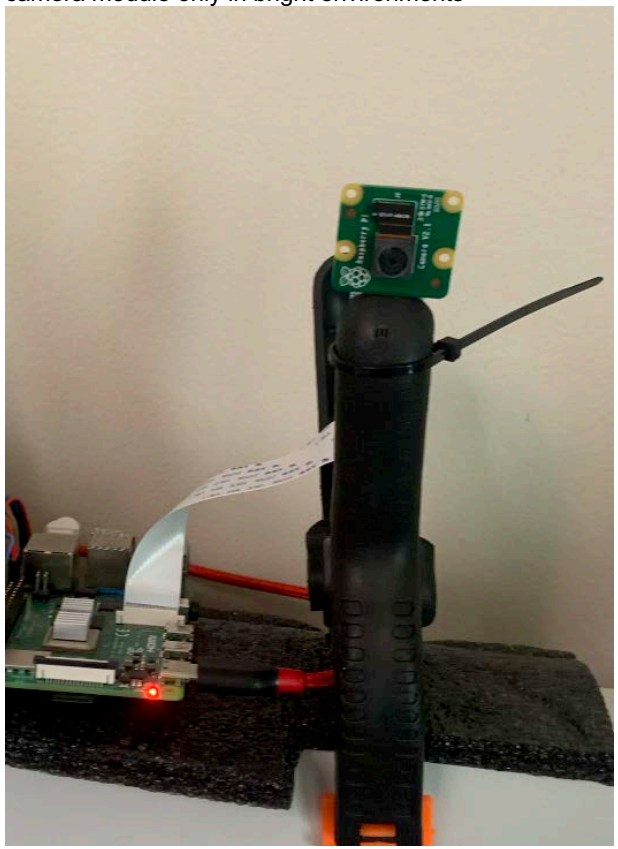
The two stepper motors were installed on the driver module DRV8825. This serves as an adjustable current limiter and simple direction control interface.

Connected to this were the 2 stepper motors which were attached to the upper left and right corners of the board. These were equipped with a belt pulley to hold the belt. Between them hangs the penholder, which was equipped with counterweights. To guarantee tension on the belt, this was also equipped with counterweights outside.

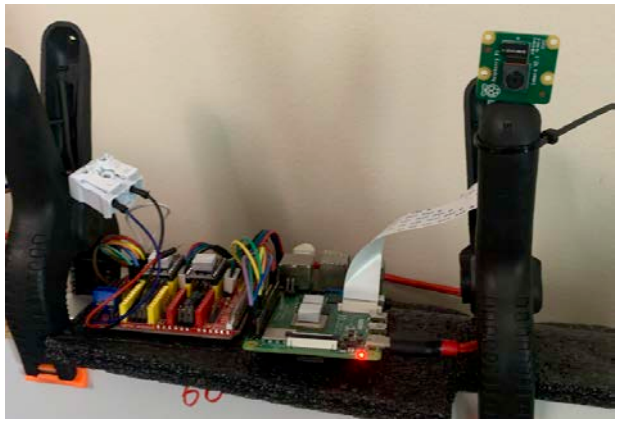


The pen holder consisted of an improvised rail made of old scrap parts from a workshop. Various screws and nuts served here as counterweights

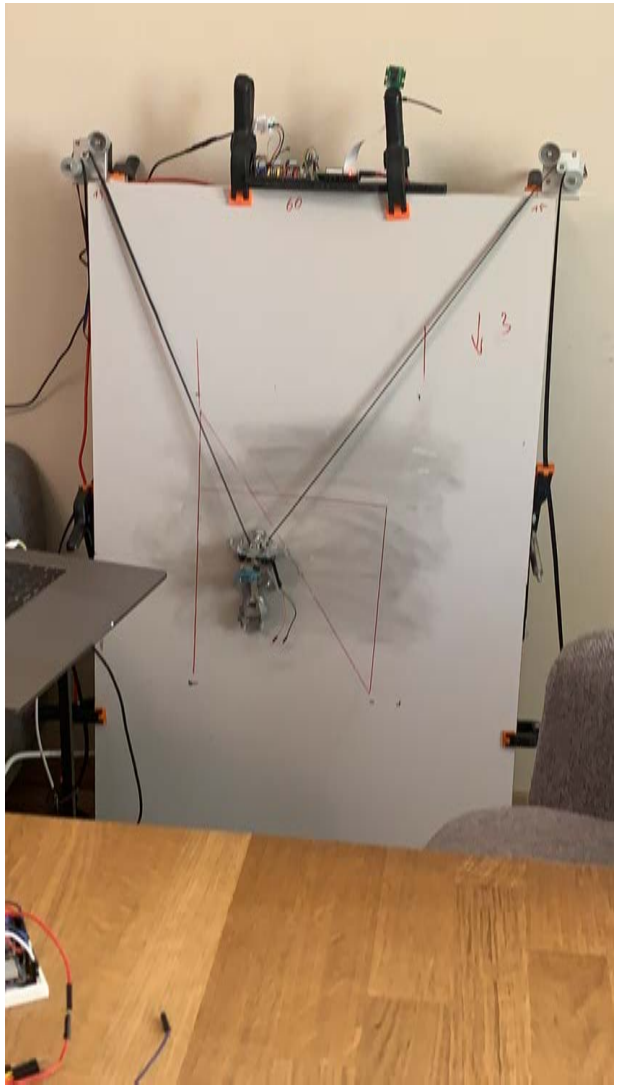
The camera module is directly connected to the Raspberry board and is powered by 12V. The camera is temporarily attached to a terminal here. We used the camera module only in bright environments



These components were then connected together on the Raspberry Pi to guarantee successful communication.



Last but not least, the control units were attached to the top of the frame. This can be mounted almost anywhere standing vertically.

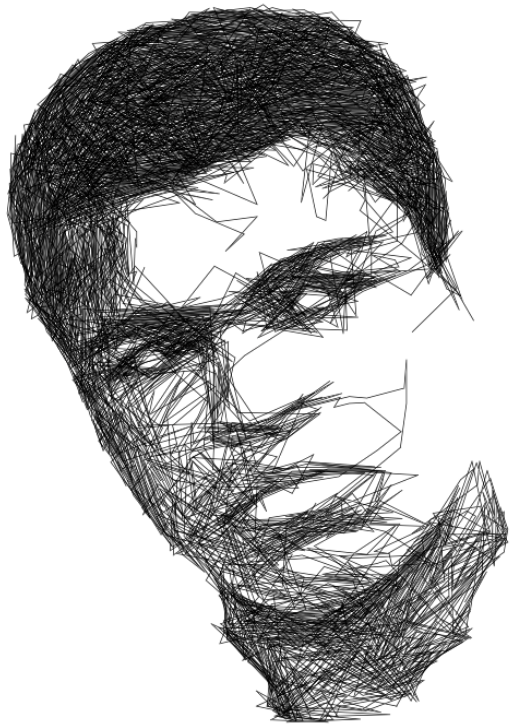


## PROGRAMMING

All operating code is created in python, due to the wide variety of libraries and application examples. Following libraries have been used:

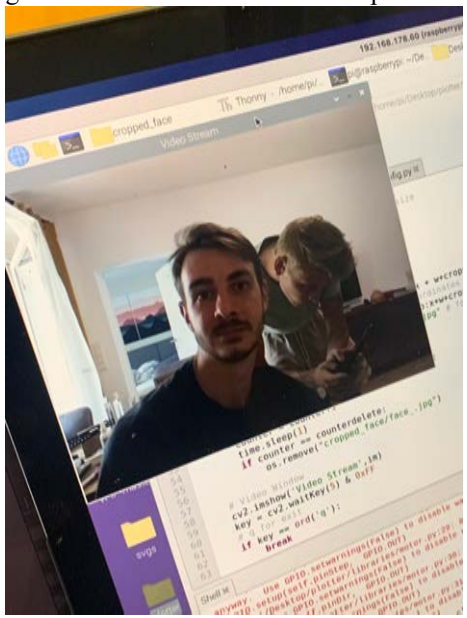
- Open CV
- GPIO
- Time
- Aso
- User

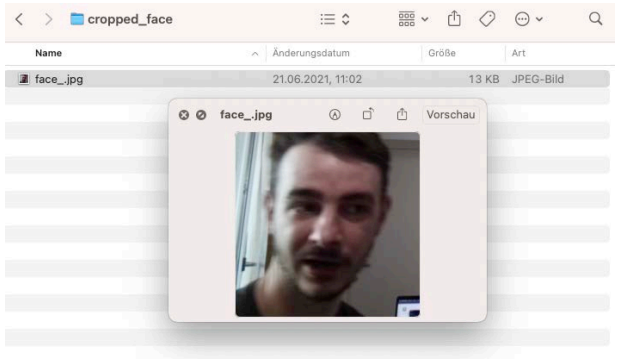
Within the Raspberry Pi interface, a project folder has been created for a successful structure. The emulator serves as a "converter" for the input vector graphics for the plotting process itself. Here, the lines are specified exactly and the vector graphic is saved in its own folder. Our goal was, to create modern fineline grafics. Following picture shows the output.



### Face Recognition & Camera Module

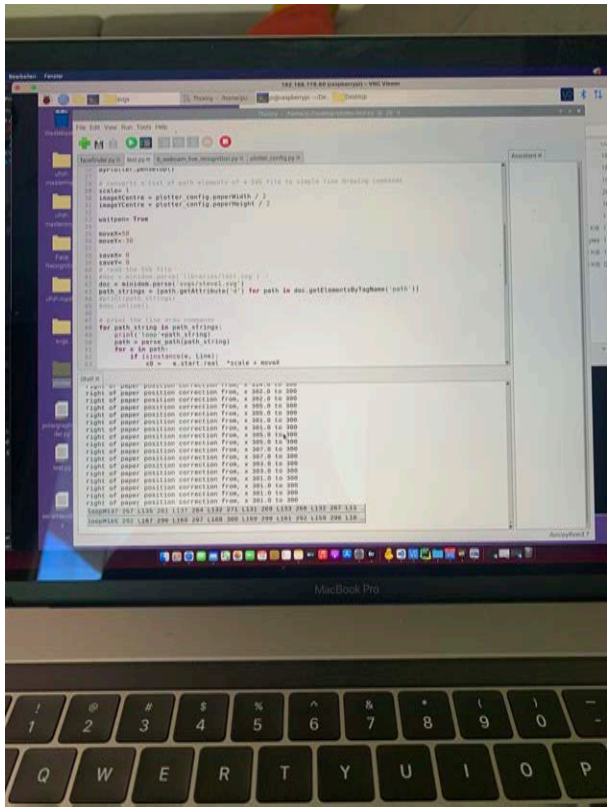
The Face recognition is run with Open CV. It basically finds the best lighted face and draws a square line around it. Afterward the code pics the recognized face and saves and JPG file in the folder “cropped_face”. The main codefile only operates if a JPG file is existing in this folder. This file gets deleted if there is no face to be regocnized currently. A new picture will be saved if a face gets detected and the process goes on.





**Drawing**

The drawing process start if a face gets detected and a JPG file is saved. The pen needs to be placed manually before the drawing process begins. The vectorized image offers coordinates of the lines which guarantees a best fitting starting point for the pen. These coordinates are scanned point by point one after the other and thus connected with a line. They also serve as input for the stepper motors, which receive the size of the paper as basic input. Then the motors start to work in opposite directions and move the coordinate points step by step.



**CONCLUSION AND FUTURE WORK**

From the very first test attempt at the very beginning of the project, we reached our limits. Our complete plan showed us big hurdles in terms of interactivity and feasibility. These problems had to be overcome and showed an in the end very successful project in the second attempt. In the beginning, the existing libraries and templates were taken too strictly and robbed us of some of our own leeway. In this semester we learned above all that in this industry a step-by-step solution process is the key to success. Solving the small problems made the successful final product possible to show the bigger picture.

The project showed the importance of the connection between the planning, building and programming process. However, the current but also final product does not meet the initially intended conditions. Therefore, we will continue to work on this project outside of the FH course in order to meet our expectations.