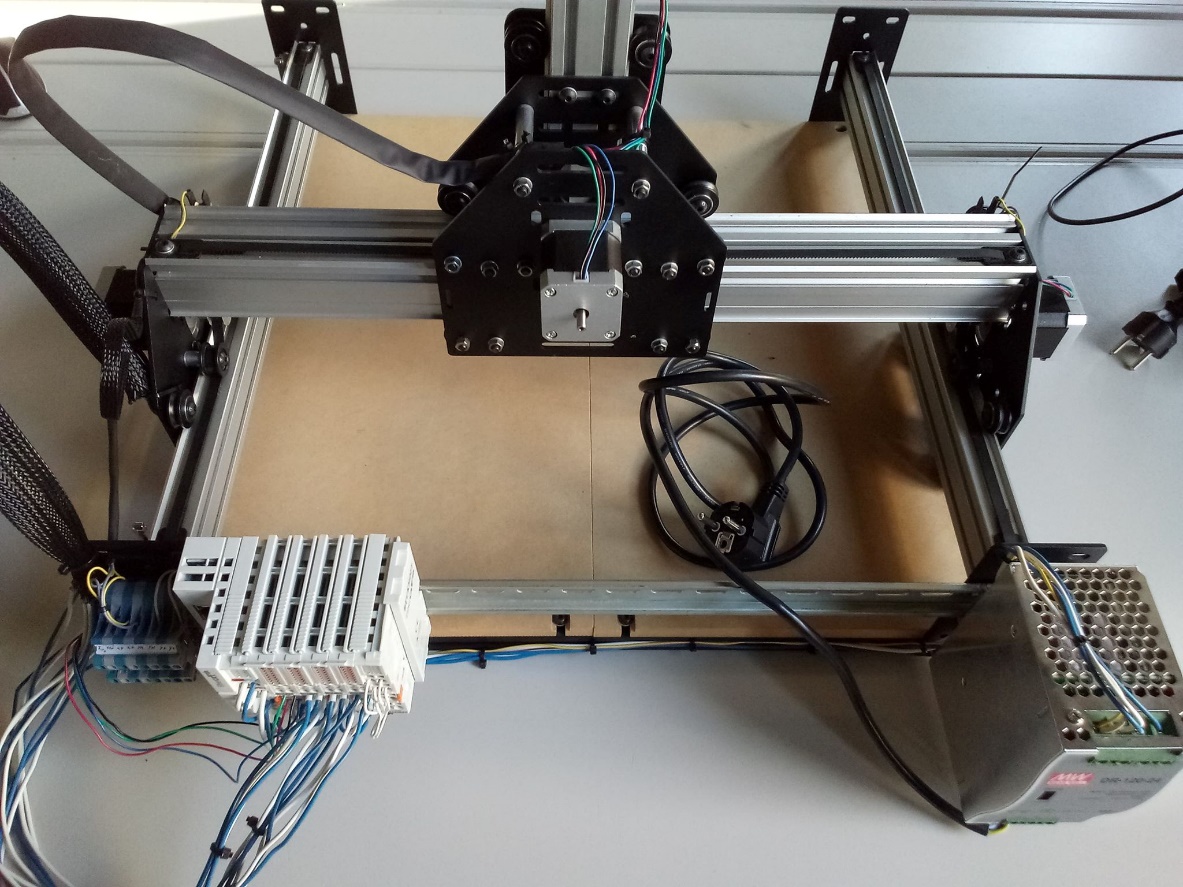
TC Robot

Multi-purpose,voice controlled automated cartethic robot



Pieter Vanormelingen

Jarni Vanmal

Cedric Diericx

Jonas Aerts

Inhouds  
opgave

Overall goal 1

Goal describtion 1

Applications 2

Component vision 5

2D vs 3D 5

Line sensors 12

Line sensor or snapshot camera ? 15

DIFFERENT SPEECH-TO-TEXT 16

Snips (.ai) 17

Daialogflow 17

Google cloud speech-to-text 17

Deepspeech (git): 17

DECISION SPEECH-TO-TEXT SELECTION 18

OPERATION OF SNIPS 19

POSSIBLE PROBLEMS 20

Snips TEXT-TO-SPEECH 21

Alternative TEXT-TO-SPEECH 22

cartetische robot 23

Commands cartesian robot 26

Show command 26

Delete command 27

Combine difference command 28

Turn objects/difference command 28

Execut commands 30

Extra commands 31

Result smart robot 32

# Overall goal

## Goal description

The cartesian robot is a Shapeoko 2 device, that is controlled by a programmable logic controller (PLC). This machine will be fully controlled by voice. The different voice commands are acquired with a speech-to-text application and send through the setup with a MQTT connection on the Raspberry Pi. A movement or a certain program will be executed based on the commands.

The goal of the project is to create a cartesian PLC construction that can be programmed in TwinCAT. The construction will eventually have multiple functionalities. Its processes will be automated and started using voice commands.

An example of a functionality is analysing an object from a picture and obtaining its contours which are then converted into g-code. This g-code is sent to the PLC and the contours are retraced.

The first section discusses the various modules and applications of the XYZ-table based on the price, difficulty and various other aspects. The second section will be about vision, how it works, and the possibilities associated with it.

In the third part speech-to-text will be discussed, how it works and what different possibilities have been considered. This is followed by the fourth section, which discusses text-to-speech.

The fifth and last section will be about the cartesian robot itself, how it operates and the specifications of the setup. The hardware.

And at the end there will be a conclusion.

## Component

## Applications

The possibilities are endless but here are some useful and obtainable applications:

* Applying glue or soldering paste using an extruder using ether a servo motor or a pump to push the liquids out. When using the pump, the pressure will have to be calibrated depending on what liquid is being used. When using a servo motor, a certain volume can be extruded.
* Sorting coins using an electromagnet.
* Picking and soring cards using a suction cup and vacuum pump.
* Edging / drilling using a Dremel tool.
* Picking up parts using a servo motor claw mechanism

The table below shows some of the different application headings. These are viewed in terms of price and difficulty to implement.

|  | hardware | price | difficulty |
| --- | --- | --- | --- |
| #1 | Glue/soldering paste | 4-15 € |  |
| #2 | Electromagnet | 3-10 € |  |
| #3 | Suction cup | 15-20 € |  |
| #4 | Drill /dremel | 15-20 € |  |
| #5 | claw | 10-20 € |  |

The prices of the electromagnet or servo motor depend on how powerful they are. A small electromagnet that can lift 2.5Kg will cost about 4 euros. Stronger ones that cost up to 25 euros can lift 25Kg.

To test the limitations of the tools, stress tests can be performed. For instance, to test the limitations of the dremel or pen, the maximum speed can be tested by looking at the inaccuracies for a specified speed.

Prices of different parts that can be used to create the different tools have been looked up on amazon to do an estimation of the costs.

The prices of the different parts that can be used to create the different tools of the cartesian robot have been looked up on amazon. This is done to estimate their price and are linked in the following points. [1]

* [Link Elektromagneet](https://www.amazon.de/BQLZR-Metall-Elektrohubmagnet-Halten-Elektromagnet/dp/B00EQ1XGO6/ref=sr_1_4_sspa?__mk_nl_NL=%C3%85M%C3%85%C5%BD%C3%95%C3%91&keywords=elektromagnet&qid=1565710141&s=gateway&sr=8-4-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEyRDJJNThZT0FTM1NUJmVuY3J5cHRlZElkPUEwNzYwMTYwUkhZOENMTkhGN1ZXJmVuY3J5cHRlZEFkSWQ9QTA0MDY0MDQyQ0lWRFZVMDFSOENNJndpZGdldE5hbWU9c3BfYXRmJmFjdGlvbj1jbGlja1JlZGlyZWN0JmRvTm90TG9nQ2xpY2s9dHJ1ZQ==)
* [Link DC-Motor](https://www.amazon.de/sourcing-map-STK-Elektromotor-Rundschacht-Spielzeuge/dp/B07L8DN584/ref=sr_1_1_sspa?__mk_nl_NL=%C3%85M%C3%85%C5%BD%C3%95%C3%91&keywords=DC-Motor&qid=1565710267&s=gateway&sr=8-1-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEyRlBOQ09HSzZFWlpQJmVuY3J5cHRlZElkPUEwMjkxOTcxMkpJT1JISVNBQlNOQyZlbmNyeXB0ZWRBZElkPUEwODY2NzQ3MlExRjFWUVE0MFo0MyZ3aWRnZXROYW1lPXNwX2F0ZiZhY3Rpb249Y2xpY2tSZWRpcmVjdCZkb05vdExvZ0NsaWNrPXRydWU=)
* [Link DC-motor](https://www.amazon.de/s?k=servo+motor&__mk_nl_NL=%C3%85M%C3%85%C5%BD%C3%95%C3%91&ref=nb_sb_noss_2)
* [Link Drill set](https://www.amazon.de/Metabo-Bohrerkassette-Spiralbohrer-rechtsschneidend-627151000/dp/B00239S7QK/ref=sr_1_13?__mk_nl_NL=%C3%85M%C3%85%C5%BD%C3%95%C3%91&keywords=bohrersatz&qid=1565710526&s=gateway&sr=8-13)
* [link Dremel set](https://www.amazon.de/Heimwerker-TwoCC-Elektrisch-Schleifer-Geschwindigkeit/dp/B07W3ZPXYQ/ref=sr_1_1?__mk_nl_NL=%C3%85M%C3%85%C5%BD%C3%95%C3%91&keywords=dremmel+tool+heads&qid=1565710709&s=gateway&sr=8-1-spell)
* [Servo claw CAD-file](https://grabcad.com/library/gripper-with-servo-motor-1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **equipment** | **object** | **differences** | **Task** |
| **Sort** | magnet, suction cup | coins, cards, components, washers | size, colour, shape | Place difference 1 at x-, y- coordinates |
| **Drill** | drill | Points and circles | none | drill holes |
| **Trace** | draw,  gluing, milling | card(boards), lids, PCB's (working area) | files on ftp (G-Code + Json) | place file 1 inside objects (scale if needed) |

There are 3 main categories of applications: sorting, drilling and tracing.

Sorting can be either done with a magnet or a suction cup. The Vision system will detect differences in size, colour or shape in the different objects such as coins. Each set of similar objects is pointed to a XY coordinate on the surface and each object in its category is placed in its respective location. (Place difference N at “X, Y” coordinate).

A drill is used to drill holes at marked locations. The locations are marked by points or circles and detected by the Vision system. In this application there are no differences to be detected.   
 (Drill Holes).

Tracing can either be done with a pen/pencil, Glue dispenser, rotary tool. The vision system detects a drawing or written text and create 2 files on an ftp server. The XYZ-table will retrace the drawing or text inside a newly marked and scanned (working)area such as lids, cards/boards and PCB’s. If a scanned drawing or text must be placed in multiple smaller areas, the image must be rescaled.   
 (Place file X inside objects)

The vision system creates a Json and g-code file on an ftp server which can be filtered by the user. The XYZ-table will point to an object or difference and the user will validate if the object is valid or if 2 differences are similar. The filtering is done by voice commands.

# Component vision

The vision part is necessary to scan object and send the data to the PLC. This data is needed to control the cartesian robot and to automate the processes.

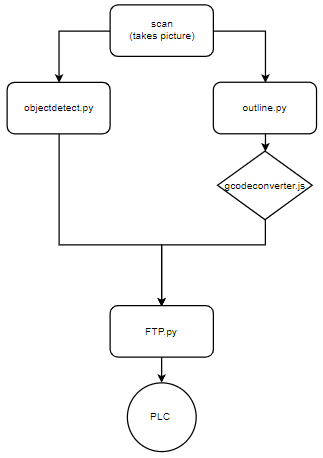
There is a choice to be made. Either a 3d line sensor or a 2d snapshot camera will be used. The specifications (resolution/accuracy) and possibilities of the different cameras.

## 2D vs 3D

3D compared to 2D allows to scan object at low contrast with its background since there’s a Z coordinate which can be filter with instead of filtering on colour differences with the background. This Z position is also needed in some other applications, for example, when it is needed to apply glue to a raised surface.

**Intro Vision**

First we show some research over vision camera’s then we show the commands we would like to integrate and then we talk about the code. The vision part is written in Python there is the python code what you can run on the raspberry Pi and .ipynb those are Jupiter Notbook files also written in python. In the Jupiter Notbook files, it’s easier to visualize all the steps and you can see what happens step by step. We got three files that we must run. The scan file is executed first Here we take a picture of the object. then we can run the object detection file and the FTP file or the outline file and the G-code converter. In the object detection file, we detect the items and get there rotated angles of the items the four corner coordinates and the centre of the items. We send this information in JSON format to the PLC using FTP. In the outline file the outline of all the items will be taken and will be converted to G-code by using the G-code converter. You can see a flowchart of the process in the photo below.



**RESEARCH VISON CAMERA’s**

Research was done on vision camera’s the prices of the vision cameras are between 100 and 200 €. More information on different vision cameras is shown in the table below.

|  |  |  |
| --- | --- | --- |
| Camera's | info | link |
| ***cognex in-sight 2000*** | - *Pressure gauge parts to check critical dimensions and / or to measure components for sorting and classification processes*  - *Can be connected to PLC*  - *Delivery time can last up to 6 months*  - *No large field of view*  - *Using easybuilder*  - *Request price via email* | <https://www.cognex.com/applications/customer-stories/electronics/ease-of-use-vision-system-helps-connector-manufacturer-achieve-zero-defects> |
| ***SENSOPART | V10-RO-A3-C-2*** | Field of view 800-600  - communicatee via Ethernet IP, Ethernet (LAN), PROFINET, SensoWeb | <https://shop.fortop.be/be_nl/catalog/product/view/id/50820/s/sensopart-v10-ro-a3-c-2-st502753-vision-sensor-robotic-advanced-c-mount-631-91082/category/1667/> |
| ***USB2.0 Camera*** | Cheap industrial camera  - Resolution of 2592x1944 (available in different resolutions)  - The prices begin at 98 euro.  - Programmable Control (features): Image size, gain, Exposure time, trigger polarity, flash polarity, etc.  - Dashing Imaging can only be controlled with the Daheng Imaging SDK. | <https://www.vision-camera.nl/5MP-USB-Camera-OnSemi-MT9P031-MER-500-7UC-L> |
| ***Adimec Q-12A65*** | Functions & features (which are interesting):  - Region of interest  - mode  - Flat Field Correction  - Programmable I/O  - HiQ mode (image averaging)  - High dynamic range mode  - Mirroring  - Internal & External triggering  - Video LUT | <https://www.adimec.com/cameras/machine-vision-cameras/quartz-series/q-12a65/> |
| ***keyence*** | - Measuring maximum/minimum dimensions of parts or products  - Measuring maximum/minimum/average inner or outer diameters or centre coordinates of O rings  - To close to the objects | <https://www.keyence.com/ss/products/vision/visionbasics/use/inspection03/> |
| ***Imprex T484016 MP CCD Camera*** | - The camera provides 4864 x 3232 resolution | <https://www.imperx.com/ccd-cameras/t4840/> |
| ***Alibaba EX130MS Machine Vision Mono Rolluik USB C-mount Camera*** | - 120 to 150 dollars  - sensor CMOS  - Uses VB, VC, C#  - software DSHOW, TWAIN, OSX, SDK, LabVIEW  - No information over the dimensions | <https://dutch.alibaba.com/product-detail/ex130ms-machine-vision-mono-rolling-shutter-usb-c-mount-camera-1627817894.html?spm=a2700.8699010.normalList.1.539f77a3c92Nbj&s=p> |

**Vision commands**

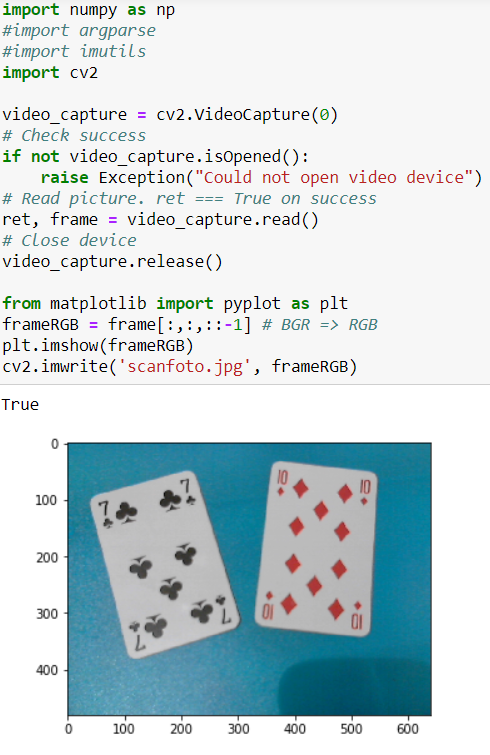
The commands to call up the python and node files are shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| Commands Rpi | Remarks | reply voice | reply command |
| Take picture | Takes pictures in python | Picture made | Blinking led |
| Scan (normal) | Scans the picture too find objects | X differences found | Show differences |
| scan (more/less) difference | Scans the differences bigger then given length | X differences found | Show differences |
| Show items | Give all the object detection data send with FTP | Show all items data | Show all items data |
| outline | Python program that gets the outline of the picture  (Opens outlineGcode.jpg) | Picture outline made | Show G-code |
| Create G-Code | Node js program that generates G-Code | G-Code file generated | / |
| Object detect | Detects objects and saves them in an array | Objects and 4 corner points + x objects found | Show points |
| Find angle and centre point  (in progress) | Gives the rotated angle and the centre point of the objects  (easier to find similarities of objects) | X items rotated and measured | Show angle |
| Show all commands | Shows all the commands | List with  Scan, outline, Create G-Code, Object detect, Change angle | / |

**CODE**

**Scan.py**

In the scan function we use the video capture function to get a picture of our objects. We use this in a separate file so we can use the picture of the objects in multiple .py files.



**Objectdetect.py**

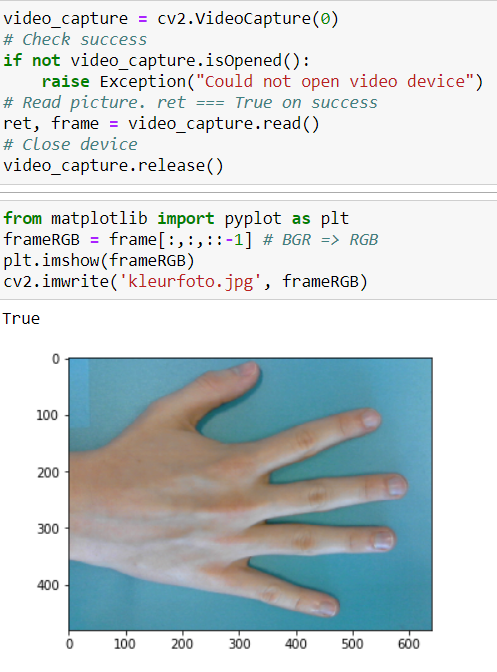
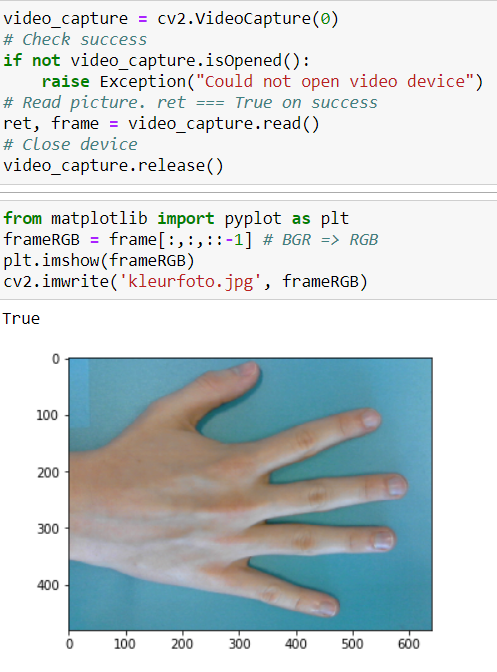
The phot made by scan.py is called up in this program and we use the order\_points function to put a box around the object. the coordinates of each object will go true a loop and be saved in a 2d array variable named box. We use math functions to get the centre point and the angle of the object. The information will be put in JSON format using json\_dump and will be saved in a .txt file.





**Get outline.py**

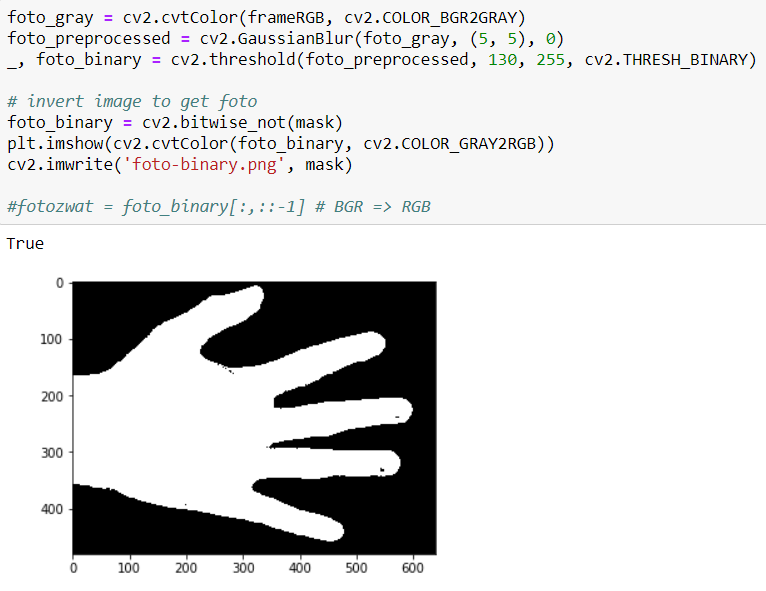
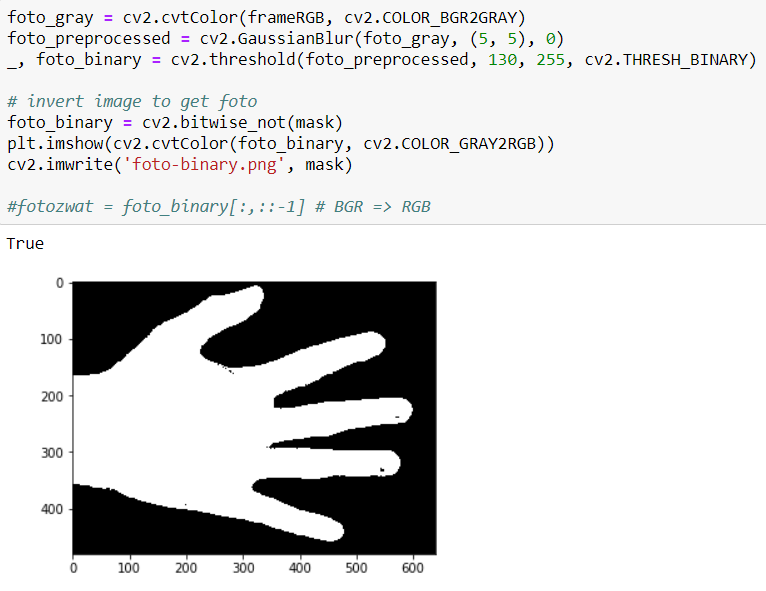
This program calls up the scanfoto.jpg from the scan.py function and use it to go further in this program.



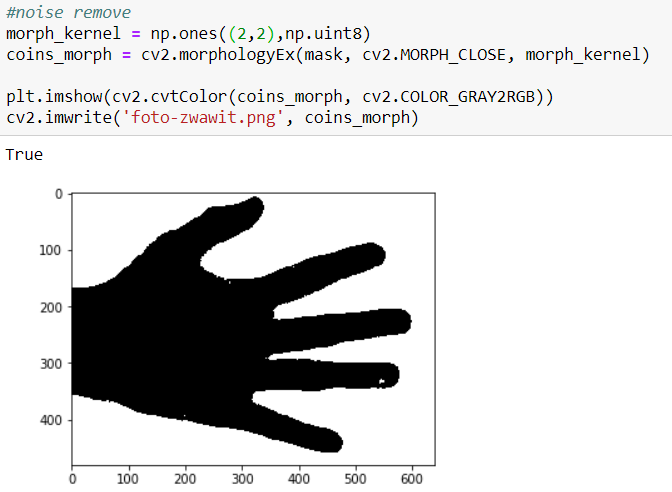
A colour filter is then applied that only the blue colours remain, and all other colours are converted to black. This ensures that all objects that are not blue turn black. Attention must be paid here to the exposure, because shadows or the reflection of light may cause parts to be displayed incorrectly.



The photo is then set in black and white to make it easier to edit.



The photo is now inverted, and part of the noise is also filtered out here, this is then saved as "foto-zwawit" we use the canny function to get the edges of the item. we save the picture as outline.jpg and use this picture in the NodeJS image to g-code converter. On the photo shown below you see a before and after from using the outline.py program.





**FTP.py**

We are using the FTP to get the connection with the plc. In this program we will send the data from the json.txt file and the data from the .gcode file.

## Line sensors

A line sensor creates a 3d image of an object by measuring the Z distance to the object line per line. This creates a profile of the object which is needed for application where a precise Z coordinate makes a difference.

To be as efficient as possible the camera must at least scan half of the surface (16 cm) at once, which means the line sensor must be at least 16cm wide. The table below gives an overview of the different options for line sensors. Only the line sensors for which a price was found are given since price played a big part in deciding which camera to use for the vision component.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Clearance | measure range | Resolutie x | Resolutie z | repeatability µm | field of view | scan rate | Prijs |
| 2x scan |  |  |  |  |  |  |  |  |
| Gocator 2440 | 183 | 210 | 0,09 | 0,037 | 1,2 | 194 | 170 - 5k Hz | 11.453 euro |
| Gocator 2340 | 190 | 210 | 0,095 | 0,037 | 1,2 | 194 | 170 - 5k Hz | 8.894 euro |
| Gocator 2140 | 190 | 210 | 0,19 | 0,037 | 1,2 | 194 | 170 - 5k Hz | 5.407 euro |
| 1x scan |  |  |  |  |  |  |  |  |
| Gocator 2350 | 300 | 400 | 0,15 | 0,06 | 2 | 365 | 170 - 5k Hz | 8.893 euro |
| Gocator 2151 | 300 | 400 | 0,3 | 0,06 | 2 | 365 | 170 - 5k Hz | 5.407 euro |

The Gocator sensors can communicate through various protocols. Most discussed and recommended protocols are CANopen and ethernet/IP. To use these protocols the right PLC terminal must be provided. For CANopen the BC500 terminal is required and ethernet the BX9000

<https://www.beckhoff.de/bx9000/>

<https://www.beckhoff.de/bx5100/>

Online specifications Gocator line sensors:

<https://lmi3d.com/gocator/product-selector?product_movement=MovingLine&part_width=180&part_height=&part_length=>

The prices are obtained over e-mail or over the phone from LMI technologies representatives.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| COGNEX | Clearance | Measure range | Resolutie x | Resolutie z | repeatability µm | field of view | scan rate | Prijs |
| DS1300 | 180 | 725 | 0.101 - 0.457 | 0.016 - 0.265 | ? | 410 | up to 10 Khz | ? |
| DS1100 | 135 | 220 | 0.079 - 0.180 | 0.01 - 0.051 | ? | 161 | up to 10 Khz | ? |

Info about COGNEX sensors:

<https://hogeschoolpxl-my.sharepoint.com/personal/20003024_pxl_be/_layouts/15/onedrive.aspx?id=%2Fpersonal%2F20003024%5Fpxl%5Fbe%2FDocuments%2FDocumenten%2FAutomation%20en%20electronics%2FIndustrial%2FTC%20Robotics%20and%20Vision%2FPOZ%201819%2FcognexDS1000%2Epdf&parent=%2Fpersonal%2F20003024%5Fpxl%5Fbe%2FDocuments%2FDocumenten%2FAutomation%20en%20electronics%2FIndustrial%2FTC%20Robotics%20and%20Vision%2FPOZ%201819>

With these sensors comes an open source SDK for configuration and real time 3D visualization. This software can be used for third party processing and communication with PLC’s.

Another brand that was considered is BASLER but for these camera’s there is no obtained pricing data and they had to sensors that had a wide enough scan area.

## Line sensor or snapshot camera?

For most applications the snapshot camera is good enough. Since The price difference is so big it is best to start with a simple snapshot camera since the price to benefit ratio for the line sensors is so bad.

Component  
Speech-to-text

The part that follows will discuss the use of speech-to-text in the project. The first section will be about the different possible speech-to-text programs that are available for implementation. It will state what the advantages and disadvantages for each program are. This is then followed by a brief decision about which speech-to-text program was chosen and why. The section that follows will elaborate on the chosen speech-to-text program. What are the functions that it has, the communication between the speech-to-text and the PLC and other aspects?

## DIFFERENT SPEECH-TO-TEXT

Various speech-to-text programs have been investigated and compared with each other. The various pros and cons will be discussed below.

It must be state in advance that the speech-to-text program/application must work offline. If the program also has online capabilities is an extra, but not imported for this project.

### Snips (.ai)

Advantages:

* Works offline/embedded
* Has its own broker
* Possible to make personal assistant
* Easy to install

Disadvantages:

* Is not available on a windows device

### Dialogflow

Advantages:

* Chat bot
* Possible to make personal assistant

Disadvantages:

* Only available on internet applications
* Needs an internet connection in order to work properly

### Google cloud speech-to-text

Advantages:

* Only translates .wav files, with speech, to a string

Disadvantages:

* Needs an internet connection in order to work properly
* There are limitations with the sound file it can convert
* Is only free for a limited time. Trial period 60 minutes

### Deepspeech (git):

Advantage:

* Only translates .wav files, with speech, to a string
* Available for free
* Possible to make personal assistant

Disadvantages:

* Only available on certain internet applications
* There are limitations with the sound file it can convert
* Difficult to install
* Not available on every system

## DECISION SPEECH-TO-TEXT SELECTION

Out of all the different speech-to-text applications, Snips (.ai) is the best option for the current project. In contrast to the other applications, it has less disadvantages that would influence the performance of the speech-to-text. Furthermore, Snips (.ai) has many advantages that benefits the current project. One of these has to do with the personal assistant, how it works, and that it has its own MQTT broker. Which makes it possible to communicate with the PLC.

**Installation and tutorials Snips (.ai)**

The following links are tutorials for installing Snips (.ai) on a Raspberry Pi. They also go more in depth how to make assistant on the Snips website. Because these tutorials provide all the necessary information, the sections that follow will not in to how to install Snips (.ai).

##### Installation tutorial for Snips (.ai) on a Raspberry Pi 3.

Imported: Snips (.ai), at this point, only works on Raspbian Sketch. It does not work on Raspbian Buster. This is because Snips (.ai) was built to work on Sketch but has not been configured to work on Buster.

[https://docs.Snips.ai/getting-started/quick-start-raspberry-pi](https://docs.snips.ai/getting-started/quick-start-raspberry-pi)

##### Tutorial for making a Snips (.ai) assistant and extra.

[https://www.hackster.io/matrix-labs/voice-control-your-lights-with-Snips-ai-and-a-matrix-device-9c54ce](https://www.hackster.io/matrix-labs/voice-control-your-lights-with-snips-ai-and-a-matrix-device-9c54ce)

## OPERATION OF SNIPS

In this section the different steps for the communication between Snips (.ai) and the different applications of the project will be discussed.

Snips (.ai) is, as mentioned before, a speech-to-text program. Which makes uses of MQTT for communication with other programs or devices. This communication protocol is also used for communication between Snips (.ai), on the Raspberry Pi, the PLC and the other applications.

The operation/communication between Snips (.ai) and the PLC will be discussed step by step in the following point.

** When starting the Snips (.ai) assistant, on the Raspberry Pi, it also activates the MQTT broker. This can be done with the “sam connect <ip address>” command. The <ip address> must be the ip address of the device. This can be the ether cat ports ip address or the wlan ip address.

 In order to make the assistant listen the “sam watch” command must be entered. This tells the Snips assistant to listen for commands.

 To give a command, the "wake word" must first be said. In this project it has been kept default to "Hey Snips".

 After the "wake word" is spoken, a sentence can be spoken that contains the different commands. Some of these commands are:

* + Scan
  + Show object
  + …

The different commands for vision, text-to-speech and the PLC (Cartesian robot) are mentioned in their respective sections.

It is also not possible to mix different commands of different applications. This means it is not possible to mix commands of vision and of the PLC, together in one sentence. Snips will only send the command of the vision or the PLC, but not both. This is due to some limitations of Snips.

 When one of more commands is detected, the Snips (.ai) program will send a Json string to the broker, with the different commands in it. This Json string will then be received by the PLC.

 The PLC and other application have a program running that listens to the broker for possible Json strings that can be sent. The different applications all have a custom MQTT topic and therefore cannot receive commands from the other applications.

 When the PLC (program) receives a Json message, it will analyse it for the different commands it contains. The different commands are then used to preform different task with the cartesian robot.

## POSSIBLE PROBLEMS

There have also been some problems showing up while using MQTT on the PLC (cartesian robot). The mayor problem that has occurred, is that the MQTT code (/program) that handles the communication between the PLC and broker, only works on the latest version of the system. In older versions, of the operating system of the PLC, the libraries about MQTT are not integrated. The system while therefore not go into the "run mode", making it impossible to run code.

It is therefore best to install the latest version of the operating system on the PLC and the other applications.

Component  
text-to-Speech

This section will be a discussion about the use of a text-to-speech application in the current project. The first part will look into the text-to-speech of Snips (.ia) itself and why it was not used. This will be followed by a section about the text-to-speech used for the current project and how it works.

## Snips TEXT-TO-SPEECH

While Snips (.ai) is a speech-to-text program/application, it also has a text-to-speech application integrated. This makes it possible for the various programs and applications to indicate if an error has occurred or what objects have been detected. In order to use the text-to-speech of Snips (.ai) a message must be sent over MQTT. The message in question, that has to be sent, is a Json message of the form:

* '{"init":{"type":"notification","text":"hello world"}}'

Most of the parameters are already set correctly, only the value that appears under "text" must be changed to the text to be spoken by the text-to-speech. In the example above, the values is set as "hello world".

Another important parameter is the topic the message will be sent to. The specific topic, in order to use the text-to-speech of Snips (.ai) is “hermes/dialogueManager/startSession”.

The disadvantage of this text-to-speech is that the sounds it produces of very low quality is. The spoken speech it produces is unintelligible and contains a lot of noise.

## Alternative TEXT-TO-SPEECH

Because of the low quality of the text-to-speech of Snips, an alternative text-to-speech was used: Say.js. In contrast to the Snips text-to-speech, the Say.js text-to-speech is of higher quality when it comes to spoken speech. The speech is fluent and does not contain any noise.

In order to use say.js a NodeJS program had to been made in order to receive messages over MQTT. This also gave it more flexibility and made it possible to add extra functionalities.

There are two topics the application is listening to:

* 'Test/Text\_To\_Speech'
* 'hermes/intent/GeneraalAlfa:Show\_commands'

The first topic is used by the different applications and programs made for the project. The message it expects to receive, is a Json message of the form ‘{"text": "This is a test"}’. The program will take the value of the parameter “text” and convert it to speech.

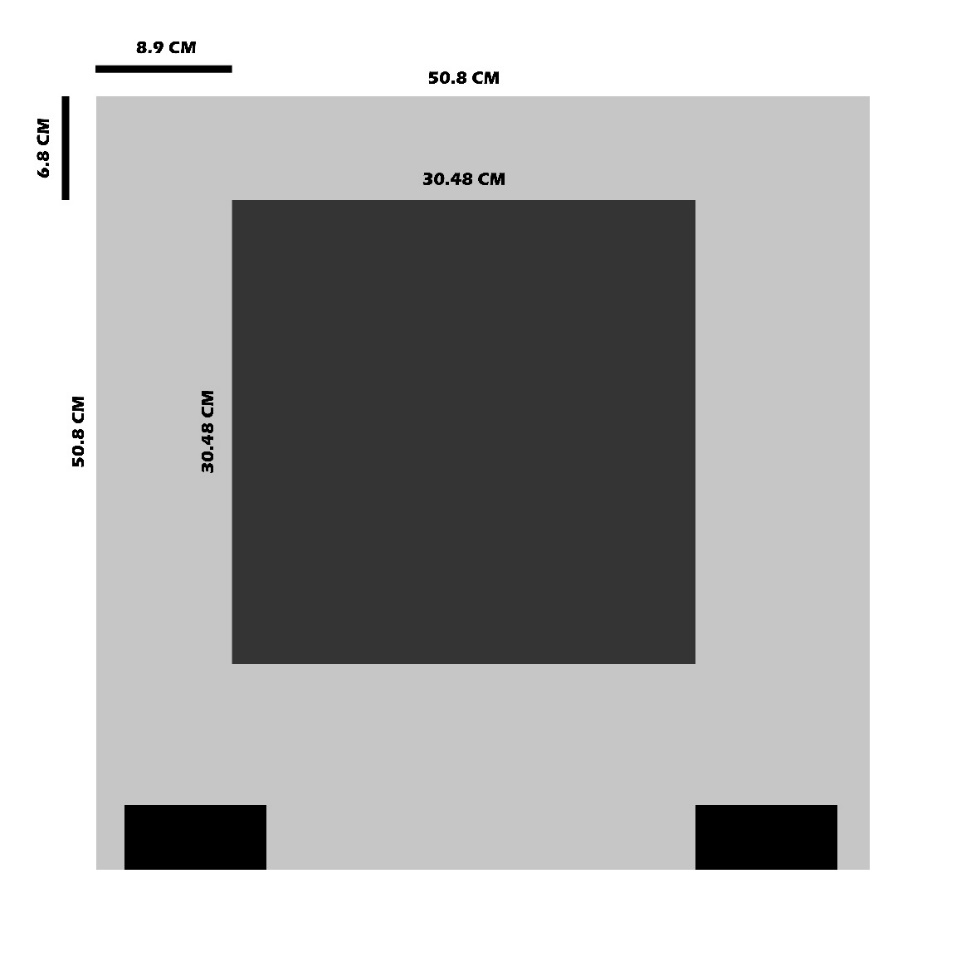
The second topic is used to receive specific commands from Snips. The Snips assistant has been configured to only send one command over this topic: “show commands”. This sends a Json message. If this message is received a list of the different commands of the project is spoken by the Say.js application.

Component

# Catesion robot

The robot consists of 3 axes, an X-, Y- and Z-axis. Each axis has a stepper motor (SM42HT47-1684B) to accurately determine the location. The Y-axis consists of two stepper motors instead of one, like the other axes. All these motors are individually connected to a stepper motor terminal (EL7031). The EL7031 EtherCAT Terminal is intended for the direct connection of stepper motors in different sizes. The slimline PWM output stages for two motor coils are in the EtherCAT Terminal together with two analog inputs.

The EL7031 can be adapted to the motor and the application by changing just a few parameters. 64-fold micro-stepping ensures particularly quiet and precise motor operation. [1] To determine the exact locations and to perform a homing, there are end-to-end contacts on the X- and Y-axes, which are connected to the digital input terminal EL1008. The entire system is powered by the DR-120-24 power supply. This power supply has an output voltage of 24 V DC that is needed for the terminals and the stepper motors. The setup surface of 50,8 on 50,8 cm, but the work surface is   
only 34,8 on 34,8 cm.



Everything is controlled by an industrial Beckhoff C6920 PLC. Beckhoff implements open automation systems based on PC Control technology. The product range covers Industrial PCs, I/O and Fieldbus Components, Drive Technology and automation software. [2]

This PLC has a windows version and can be programmed in   
TwinCAT 3. Due to the text-to-speech that happens on the Raspberry Pi, commands are sent to the PLC. These commands then ensure that a certain movement will happen or that a program will be executed, for example a g-code. All of this makes the cartesian robot multifunctional and allows the user to sort parts, drill, glue, etc.

This part of the cartesian robot is fully programmed in TwinCAT 3. TwinCAT is a software program made by Beckhoff, that makes it possible to program the PLC. This program is used to program the motion control for this device. The program consists of various functions from the motion control references. The functions make it possible to do the movements and the location determination. One of the important functions is mc\_home, this one is needed to calibrate the robot for further movements. The movement that will happen depends on the command that the PLC receives from the Raspberry Pi. The robot can move to a certain position and perform an operation, can perform various operations one after the other with the information that is noted in a Json file or even execute a g-code. The g-code or Json files can be sent to the PLC through an FTP connection between the Raspberry Pi and the PLC.

***All of this together makes this robot a multifunctional cartesian robot.***

## Commands cartesian robot

In the section that follows the different commands that are used by the cartesian robot will be discussed.

A list of commands with execution within the PLC are:

|  |  |
| --- | --- |
| Name of command | Remarks |
| show object <number> | Show all or just one |
| show difference <number> | Show all or just one |
| delete object <number> | Delete one object (or scan for all) |
| delete difference <number> | Delete one diff (or scan diff for all) |
| combine difference <number> and <number> | Combines difference <number> and <number> in one difference |
| Turn objects / difference <number> | Turn object/difference <number> |
| execute sort | ask places to put differences |
| execute drill | Ask to drill different objects |
| execute trace | Ask to trace different objects or differences |

### Show command

The "Show object" command indicates the different objects that are present the work surface. These objects are determined from the scan that is taken or the photo that has been taken of the workspace. There are two cases here.

If only "show object" is spoken, the head of the XYZ-table will go to each individual object (in order of number) in order to display which object was detected.

In the case two objects overlap, it can be seen as one object. If the XYZ-table indicates this, then these objects must be removed from each other and the work surface must be scanned again, using the "scan" command. It is recommended to do this before deleting objects.

"Show object <number>" command. This is the other form of the "show object" command. Here the "show object" command is followed by a number, indicating the object of interest. This command will ensure that the head of the XYZ-table will only go to the specified object.

The "Show difference" command indicates the different groups that are present on the work surface. These groups are determined from the scan or the photo that has been taken of the workspace.

If "show difference" is spoken, the head of the XYZ-table will go to the different differences (groups).

If "show difference <number>" is spoken, the head of the XYZ-table will go to the specified difference (group) in question.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Requirements | Reply voice | Reply command |
| Show object | The "scan" command must have been executed in advance. | Showing all object by number | Scan/delete object <number> |
| Show object <number> | The "scan" command must be executed. Best first "Show object" for total overview. | Showing object <number> | Delete object <number> |
| Show difference | The "scan" command must have been executed in advance. | Showing all differences by number | Scan/delete difference <number> |
| Show difference <number> | The "scan" command must be executed. Best first "Show difference" for total overview. | Showing difference <number> | Delete difference <number> |

### Delete command

The "delete" command is used in two ways. It can be used to remove an object and / or can be used to remove a difference (group) from the list of objects / differences.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Requirements | Reply voice | Reply command |
| Delete object <number> | The “scan” and “show object” commands must be used beforehand | X objects found | Show objects |
| Delete difference <number> | The “scan” and “show difference” commands must be used beforehand | X differences found | Show differences |

### Combine difference command

The "combine difference" command is used to merge two differences together into one difference. Combine difference <number (1)> and <number (2)>, the first number of the command stand for the difference (group) to which difference (number) 2 is added.

The “combine difference” command will change the item number ‘…..’ inside the Json file of the second difference in the number of the first difference.

|  |  |  |
| --- | --- | --- |
| Name | Reply voice | Reply command |
| combine difference <number> and <number> | Difference <number> and <number> combined | / |

### Turn objects/difference command

The "Turn objects / difference <number>" command ensures that the object in question is rotated with an angle of "x" degrees. It is possible to choose at which point it should be rotated. These can be:

* Longest side
* Shortest side
* …

Commands that belong to this are:

|  |  |
| --- | --- |
| Command | Remark |
| Longest side | The object/ difference will be turned around the longest sides |
| Shortest side | The object/ difference will be turned around the shortest sides |
| Center | The object/ difference will be turned around the centre |
| <number> degrees | Indicates how many degrees to rotate the object/ difference |
| … | … |

An object as well as a difference can be turned.

|  |  |  |
| --- | --- | --- |
| Name | Reply voice | Reply command |
| Turn object <number> | How would you like to do this? (degrees, longest side, shortest side) | See table above |
| Turn difference <number> | How would you like to do this? (degrees, longest side, shortest side) | See table above |

### Execute commands

|  |  |  |  |
| --- | --- | --- | --- |
| Commando | Head(s) | Reply voice | Extra command |
| execute sort | Magnet, suction | Where should diff x be located? | Table section “Magnet/Suction Cup” |
| execute drill | Drill | Calibrate the z axis by saying up/down and number and validate | Table section “Drill” |
| execute trace | Draw, gluing, milling | Tracing will be executed | Table section “Gluing/ milling/ drawing” |

The part that follows will discuss the different "execute" commands. There are 3 different version, depending on the head that is used by the Cartesian robot. The different execute commands are shown in the table above.

These commands are accompanied by some other commands that are needed for these commands to work properly. These extra commands are listed below for each execute command.

#### Magnet/Suction cup

|  |  |
| --- | --- |
| Command | Remark |
| X-axis <number> | X position where the object/difference must go to. |
| Y-axis <number> | Y position where the object/difference must go to. |
| Z-axis <number> | Z position where the object/difference must go to. |
| Difference <number> | Difference that the execution will be performed on |
| Object <number> | Object that the execution will be performed on |

#### Drill

|  |  |
| --- | --- |
| Command | Remark |
| Up <number> | Determines how much the Z-axis must go up |
| Down <number> | Determines how much the Z-axis must go down |

#### Gluing/milling/drawing

No extra commands

## Extra commands

The table below shows some additional commands that can be used by the PLC.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Remark | Reply voice | Reply command |
| Enable |  |  | / |
| Stop |  | Stopping program | / |
| Z to zero |  |  | / |
| Do homing |  | Going to home position | / |
| Reset | Reset system to default | System reset | / |

# Result smart robot

The cartesian robot is a Shapeoko 2 device, that is controlled by a PLC. The PLC controls all the movement of the axis. The robot is fully controlled by voice commands that the PLC receives through a MQTT connection with the Raspberry Pi. The Raspberry Pi is used to run the speech-to-text, the text-to-speech and the vision programs. The MQTT broker on the Pi will send all the available data as a JSON file and saves it on the PLC by using ftp. The software used for the conversion from speech to text is the Snips software because it has the advantage of offline usage. For the vision part everything is made in Python. Because this program has a lot of object detection functions. The locations of objects, that are detected by the vision camera and software are needed for the device to work properly.

In the end, this device can still be upgraded for other possibilities. All of this together makes this robot a multifunctional cartesian robot.