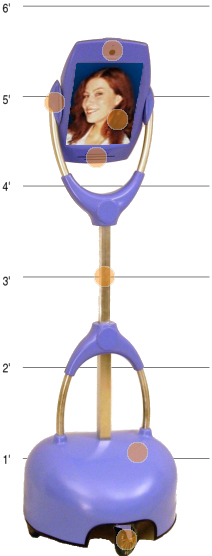
Manual ZAP ROBT Facetracker

https://idservice.zuyd.nl/nidp/images/hszuyd/zuydlogo.gif

Fernand Hendriks

[Fernand.h@hotmail.com](mailto:Fernand.h@hotmail.com)

2015

**De printplaat voeding sensoren en UDN2998W full-bridge motor driver**



GND

5V

5V

GND

VSS Logic

GND

Phase A

Enable A

Enable B

Phase B

GND

Out 1A

Out 2A

Out 1B

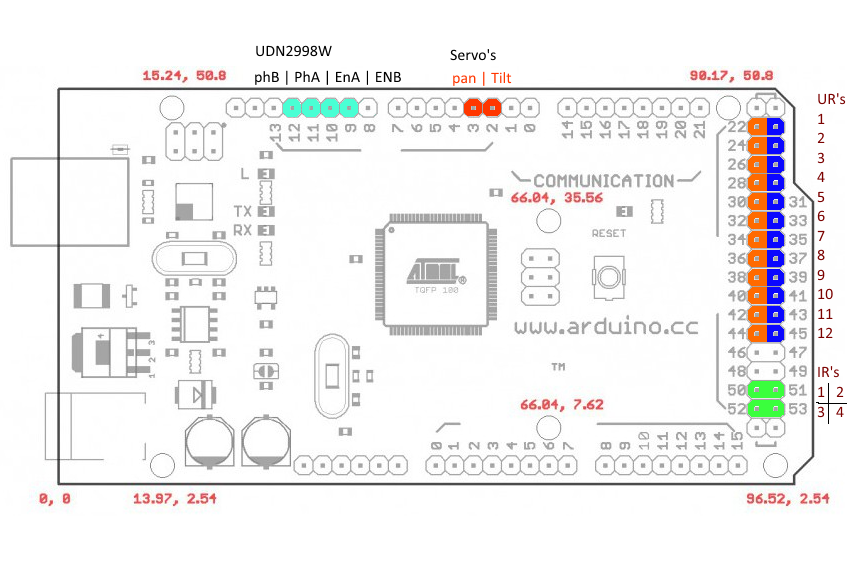
Out 2B

Voeding

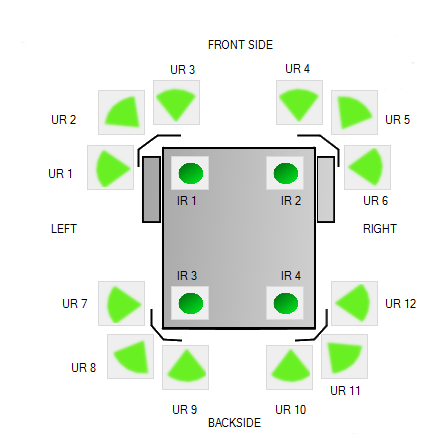
GND

Figuur 1 : PCB robot

**Aansluiting Mega 2566**

 Figuur 2 : Aansluiting layout mega 2560

**Nummering en plaatsing sensoren**

****

**Figuur 3 : numering en servo layout**

**Benodigdheden robot:**

2x arduino mega (waarvan 1 voor servo’s en h brug en 1 voor sensoren met ros node )

12x UR sensor

4x IR sensor

1x webcam

1x nema 17 Wantai stepper motor : 42BYGHWB11

2x servo’s

1x laptop met software facetracking

**Optioneel :**

Zet software op rpi2 , zoek op servoblaster en installeer, verder pas code aan zodat deze in plaats van naar Arduino naar de servo’s schrijft. Factrack.py bevat grootste deel code. Echter zit hier niet de H-brug in verwerkt. Dit moet hiervoor worden aangepast.

-benodigdheden : openCV , Servoblaster , Python ,RPI2

**Opstarten facetracker:**

* Sluit Arduino en webcam aan op laptop.
* Controleer bedrading en voeding/GND . zorg dat GND arduino met voeding hbrug en servo’s verbonden is!
* Open terminal in linux.
* Commands:
  + ython facetracker\_servo\_gpio\_edit6\_5.py <camera-number>
  + Camera number is probably 1. In case no build-in webcam is present camera-number = 0.
  + Select usb port to Arduino (options presented)

**Error handling**

Soms werkt de uitsturing van de servo’s niet. Dit heeft te maken met de verbinding tussen de arduino en laptop. Hierdoor wordt ook aangeraden om software indien mogelijk over te zetten naar de RPI2 of een sneller embedded device. Dit zodat er geen extra communicatie protocollen hoeven worden gebruikt.

Bij geen uitsturing :

Reset arduino, werkt dit niet dan reconenct arduino aan de pc.

Bij lagg :

Dit heeft met drie factoren te maken:

1 Open CV belast pc waardoor vertraging.

2 De gebruikte webcam levert vertraagd beeld. (huidige webcam(VGA uit 2000) heeft dit probleem)

3 Door de Python language, deze taal is niet optimaal. Daarom wordt het ook aangeraden om script om te schrijven in C++. Gebruik Qt om het script cross-platform te maken.

Install OpenCv linux

<https://help.ubuntu.com/community/OpenCV>

**Install RPI**

The first step of this procedure is to install the required libraries and packages using the Terminal.

sudo apt-get update && sudo apt-get install git python-opencv python-all-dev libopencv-dev

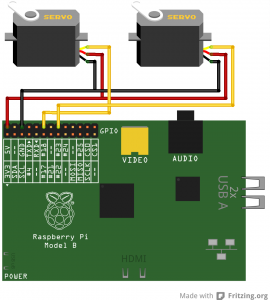
This command will pull down all of the required packages (about 215 MB worth) including the git version control system, as well as the OpenCV development headers and Python bindings. The next step is to configure the Raspberry Pi for use with multiple pulse width modulation (PWM) outputs.

The commands take the form “=” with servo number representing the desired servo (0-7 in this case) and servo position representing the pulse width in units of 10 µs. For example, to send servo 3 a pulse width of 120 µs: echo 3=120 > /dev/servoblaster To configure the servoblaster on the Raspberry Pi, first pull down the sources from Richard’s Github repo:

git clone https://github.com/richardghirst/PiBits.git

Then change into the servo blaster directory, and install the module:

cd PiBits/ServoBlaster

make install\_autostart

Figuur 4 : aansluiting servo's rpi

This command also sets up the necessary udev rules for accessing the /dev/servoblaster device. Note: using the ’install\_autostart ’ command will set up a Raspberry Pi to load the servoblaster kernel module on every boot. If you don’t want this behavior, execute ‘make install’ instead. In either case, the module will not yet be loaded so go ahead and load it into the kernel using modprobe:

sudo modprobe servoblaster

Now that all the prerequisites have been installed and the servo blaster device configured, it’s time to get the actual face tracking and servo moving code. Clone my repo from Github here:

git clone https://github.com/mitchtech/py\_servo\_facetracker

Now change directory into the created folder, and run the script like this:

cd py\_servo\_facetracker

python ./facetracker\_servo\_gpio.py 0

If you have a different servo bracket configuration, the pan and tilt axis may need to be inverted. To do so, invert the sign on the values of panStepSize and tiltStepSize. Similarly, increasing or decreasing these values will change the sensitivity of the movement, larger numbers corresponding to more degrees moved per face detection frame.