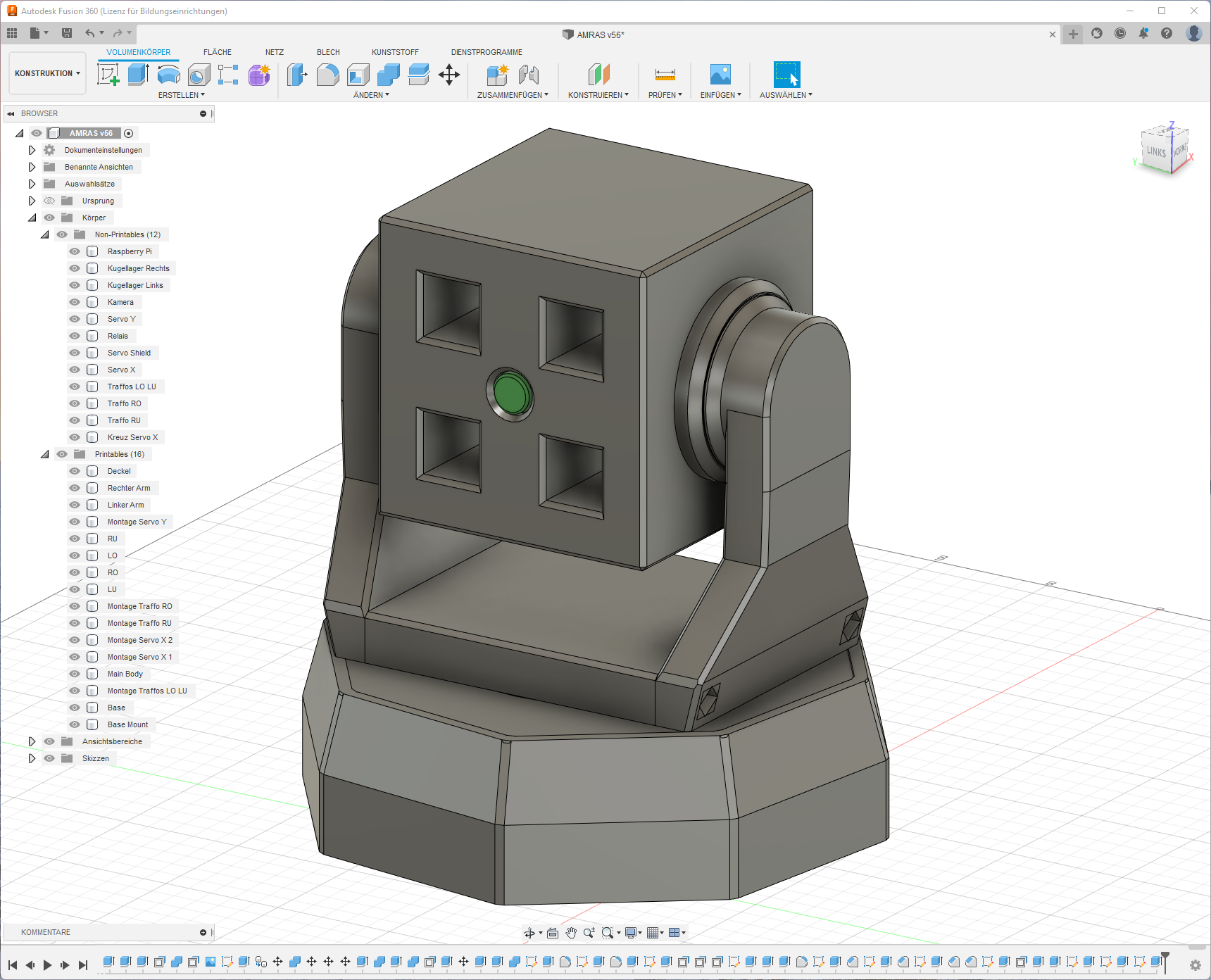
AMRAS Build Manual



Inhalt

[Introduction 3](#_Toc105767112)

[Parts List 4](#_Toc105767113)

[Assembly 5](#_Toc105767114)

[Software 1 5](#_Toc105767115)

[Installing Raspberry Pi OS 5](#_Toc105767116)

[Installing the necessary python packages 5](#_Toc105767117)

[Configuring the Raspberry Pi 5](#_Toc105767118)

[Hardware 6](#_Toc105767119)

[Dry Running 6](#_Toc105767120)

[Glueing Magnets 7](#_Toc105767121)

[Installing the Camera 7](#_Toc105767122)

[Preparing and installing the trafos 8](#_Toc105767123)

[Mounting the Raspberry Pi 10](#_Toc105767124)

[Adding the arms 10](#_Toc105767125)

[Wiring the Base Mount 11](#_Toc105767126)

[Software 2 12](#_Toc105767127)

[Running AMRAS 12](#_Toc105767128)

[Command Line Arguments 12](#_Toc105767129)

# Introduction

AMRAS is the abbreviation for the German long form “Anti-Mitbewohner-Raketen-Abwehr-System” which translates to Anti-Roommate-Rocket-Defence-System.

It uses a combination of a Raspberry Pi, a camera, some servos, trafos and methylated spirits to propel rubber rockets into anyone’s face who dares to enter its vicinity

This guide will show you how to build and start it, have fun!

I got inspired when reddit flushed [this video](https://youtu.be/U4ndTDGhc1U) onto my frontpage and I thought “that’s kinda cool, but I have a 3D printer and a Raspberry Pi, I could do that better”.

My code is based on Adrian Rosebrock’s [tutorial](https://www.pyimagesearch.com/2019/04/01/pan-tilt-face-tracking-with-a-raspberry-pi-and-opencv/) who used a similar setup to track faces. While he used a PID controller for moving the servos, I had to fall back to a simpler movement, but this will hopefully soon be fixed.

# Parts List

These are not affiliate links, just the parts I used. It should also work with slightly different hardware, but then you would need to adapt the 3D models for mounting holes

* A Raspberry Pi 3B (a newer version should also work, but this is what I have tested with
* A 5V Power supply and some way of distributing power
* [Camera](https://de.aliexpress.com/item/32668508991.html)
* [Servo Controller](https://de.aliexpress.com/item/4000468996665.html)
* 4x [Transformer/Spark generator](https://de.aliexpress.com/item/32872820437.html)
* 2x [Servo](https://de.aliexpress.com/item/32970493831.html)
* [4-Band Relais](https://de.aliexpress.com/item/4001363470793.html)
* Some [Cables](https://de.aliexpress.com/item/33060775595.html)
* 3x [Ball Bearing](https://www.amazon.de/gp/product/B076ZL83YN/ref=ppx_yo_dt_b_asin_title_o03_s00?ie=UTF8&psc=1)
* 4x [Mini Magnets](https://www.magnet-shop.net/neodym/quadermagnete/quadermagnet-10.0-x-5.0-x-3.0-mm-n40-nickel-haelt-1.4-kg) for keeping the lid shut
* [Syringes](https://www.amazon.de/gp/product/B07MC7L4PG/ref=ppx_yo_dt_b_search_asin_image?ie=UTF8&psc=1) to craft rockets out of
* 3x [Switches](https://www.amazon.de/gp/product/B07CG7JYF8/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1)
* Micro USB Cable to provide power for the Raspberry Pi (or USB-C if using a newer model)
* SD Card with minimum 8GB for the operating system
* Bunch of Screws (I used Wood Screws)  
  Raspi: 4x 2,0x10mm

Kamera: 4x 2,0x10mm

Main Body Fillings: 24x 2,9x13mm

Arms: 4x 2,9x13mm

Servo Shield: 4x 2,0x10mm

Relais: 1x 2,0x10mm

* All printed 3D parts

# Assembly

## Software 1

### Installing Raspberry Pi OS

Install Raspberry Pi OS on the SD card and boot your Raspberry Pi.

### Installing the necessary python packages

Open the terminal on your raspberry pi and enter following commands:

*sudo pip3 install opencv-python*

*sudo pip3 install smbus*

*sudo pip3 install imutils*

*sudo pip3 install "camera[array]"*

*sudo pip3 install adafruit\_servokit*

*sudo pip3 install adafruit\_motor*

### Configuring the Raspberry Pi

Enter ‘*sudo raspi-config*’ to open the Raspberry Pi Software Configuration Tool, go to ‘Advanced’ and enable Camera and I2C (and SSH, if you want to control AMRAS remotely)

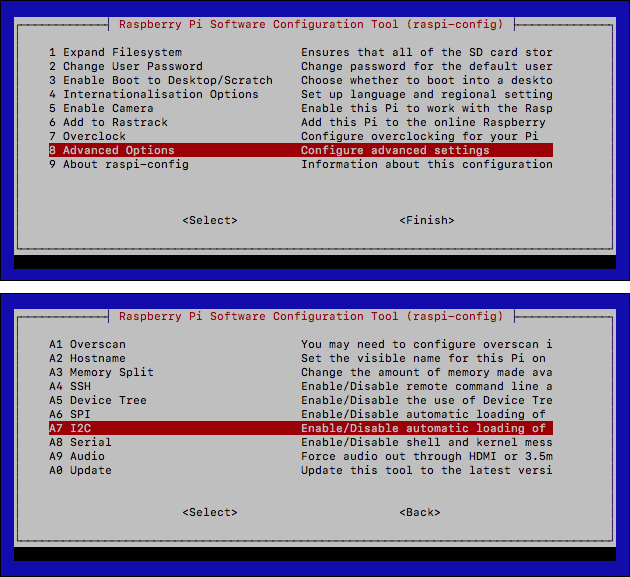


Figure 1 Raspberry Pi Software Configuration Tool

## Hardware

### Dry Running

I recommend connecting all parts according to the circuit diagram first, just so you don’t end up having any missing/broken parts which will need to be swapped. (This can be quite tedious, trust me)

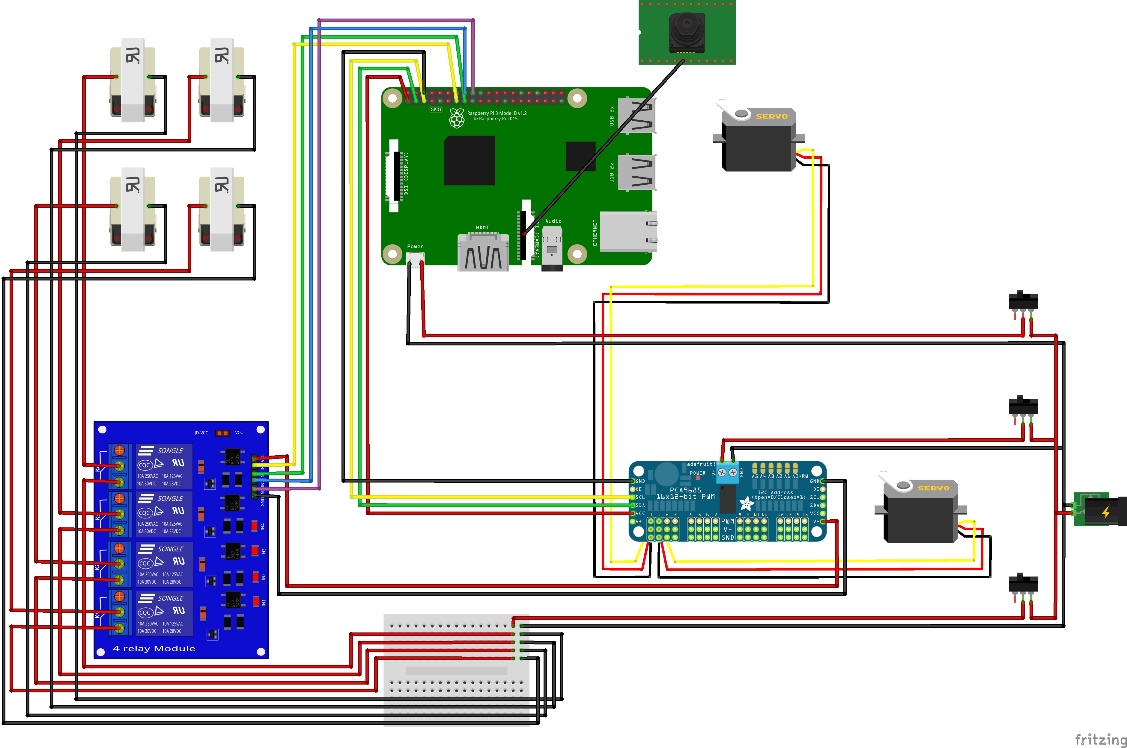


Figure 2 Circuit Diagram

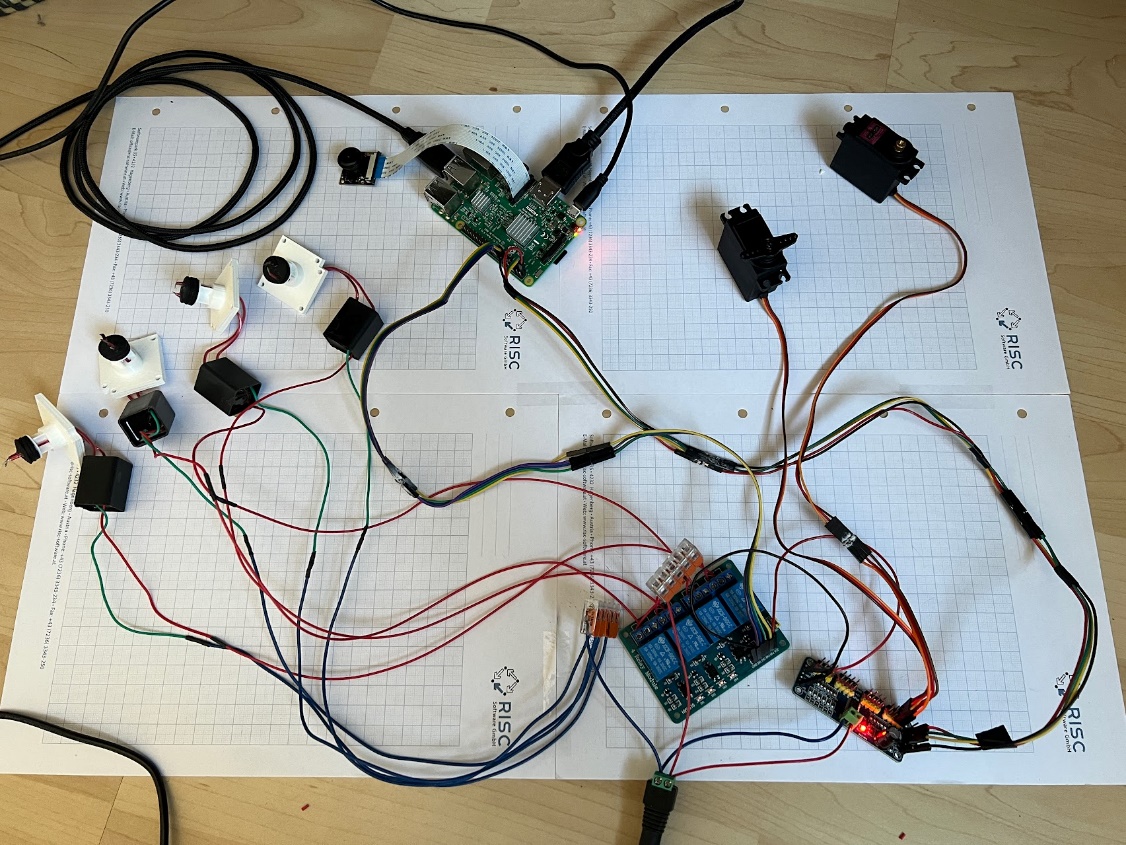


Figure 3 This is just ~90% correct, I forgot to take a new picture after making some adjustments

After making sure everything is connected properly, run test\_servos.py to check functionality.

Try not to move the servos afterwards, as this would move the home/default position of AMRAS.

### Glueing Magnets

Start by glueing the magnets for the lid in place, making sure to have the proper polarity.

Ein Bild, das drinnen enthält.

Automatisch generierte Beschreibung

Figure 4 Lid and Main Body with the Magnets

### Installing the Camera

Screw in the Camera, the ribbon cable should fit into the recess below it.

Ein Bild, das drinnen, Badewanne, Kübel, Bad enthält.

Automatisch generierte Beschreibung

Figure 5 The Camera screwed in place

### Preparing and installing the trafos

Poke two holes through the end pieces of the syringes and passthrough the output cables of the trafos through both the mount and the syringe end piece. The endpiece needs to be cut off to be slotted into the cross-holes.

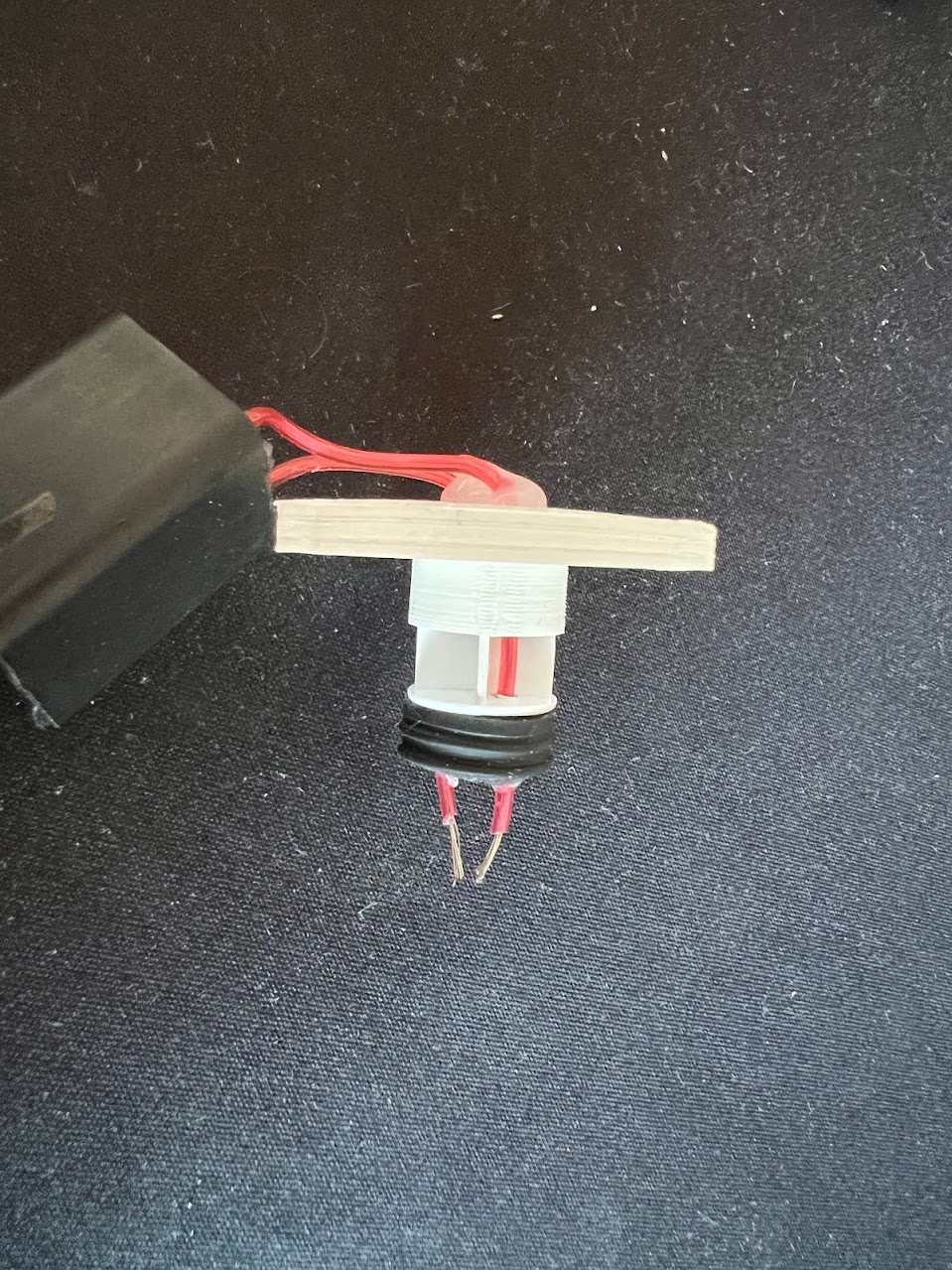


Figure 6 Installation of trafo, syringe endpiece and mount

After making sure you have enough cable length to mount the trafos in the main body, glue the cables in place.

Ein Bild, das drinnen enthält.

Automatisch generierte Beschreibung

Figure 7 Checking the cable length

The input cables of the trafos will need to be extended for about 30cm, to be able to reach through the arms, I also recommend marking them by colour, to be able to differentiate them later.

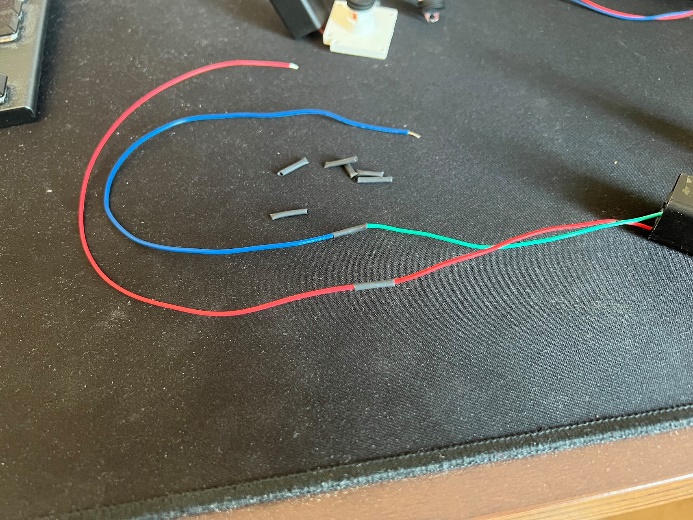


Figure 8 Cable extensions

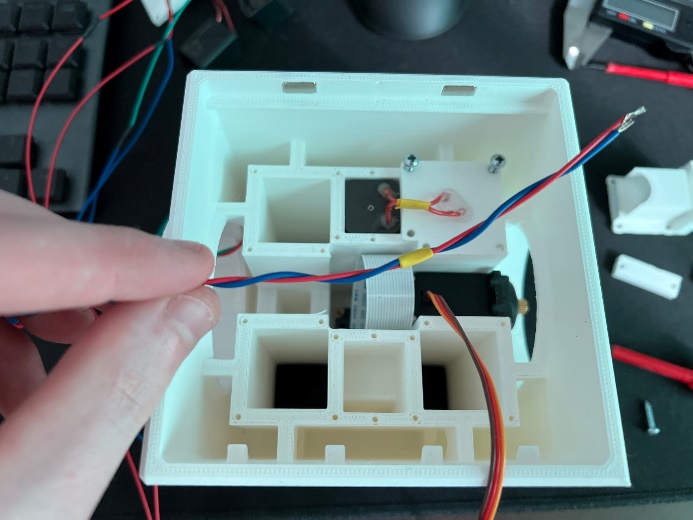


Figure 9 Colour-coding

Now it is time to mount all the trafos and servo Y, while also putting the cables in the right places.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Power for Raspberry Pi

Cables to Raspberry Pi

RU

RO

LU

LO

Figure 10 Main Body all parts installed

### Mounting the Raspberry Pi

Screw the Raspberry Pi into the provided space on the lid, making sure to pay attention to the orientation.

Ein Bild, das Text, Elektronik enthält.

Automatisch generierte Beschreibung

Figure 11 The mounted Raspberry Pi

Plug in the cables and close the lid for now.

### Adding the arms

Put the ball bearings into place and add the arms, while passing the cables through.

Ein Bild, das Elektronik enthält.

Automatisch generierte Beschreibung

Figure 12 Arms and ball bearings added

### Wiring the Base Mount

## Software 2

### Running AMRAS

If everything is connected properly, it should now be as easy as running init\_amras.py.

For calibrating I recommend plugging in the Raspberry Pi into a monitor with mouse and keyboard, afterwards it is easily possible to just run it using SSH.

You can alter the mid\_offset (line 24 in init\_amras.py) variable for calibration in case the camera is not mounted perfectly flush/centered.  
Also, you might want to play around bit with the translation multiplier (line 22), but I decided it is good enough for me, also getting the PID controller to work is the goal anyways.

### Command Line Arguments

It is possible to provide some arguments when running the program, to change the console output, the movement implementation...

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Arg** | **Type** | **Required** | **Default** | **Description** |
| -c, --cascade | String | True | - | Path to the haar cascade |
| -a, --armed | Bool | False | False | Option to arm the turret |
| -p, --pid | Bool | False | False | Option to use PID over more basic movement |
| -v, --verbose | Bool | False | False | Option to show console output |
| -I, --image | Bool | False | False | Option to draw image |

**Examples:**

*python3 init\_amras.py -c frontalface.xml -a True*

Starts AMRAS without showing any output, tracking faces, and shooting when on target.

*python3 init\_amras.py -c ball.xml -i True -v True*

Starts AMRAS while showing various values I the console, also enables camera feed and will track balls, recommended for calibrating the offset.