Internship

**Main report internship assignment Health Concept Lab**

**Student:** Casper R. Tak

**Studentnumber:** 657313

**Client**: Rudie van den Heuvel/Jeroen Veen

**Coach:** Jeroen Veen

**High School:** HAN Arnhem

**Education:** Embedded Systems Engineering

**Date:** 17-01-2023

# Brief Description

In this report, we present the design and implementation of the Rastaban water quality monitoring system, which combines hardware and software.

The system uses a Raspberry Pi running Raspbian as the main controller and computer, and a PCB for circuit layout. Control of the hardware, GUI, and data processing is done using Python. The design process involved selecting suitable hardware, designing the PCB, and programming the firmware in Python. The system was rigorously tested to ensure accurate and reliable measurements by testing both hardware and software components separately.

# Preface

## Personal Goals

I recently completed projects S3 and S4 at the Health Concept Lab (HCL) at the HAN and decided to do my internship at the same location. During my internship, I plan to further develop my planning skills and improve my PCB design skills using KiCad6. Additionally, I hope to improve my programming skills in C, C++, and Python. I am also interested in exploring and expanding my knowledge and understanding of various topics like industrial product design.

The fluid analyzing device project will be a key part of my internship as it will allow me to achieve these learning goals. The project involves creating a smaller, improved version of a fluid analyzing device that was developed by previous S6 and S4 students. The device is being developed for Jeroen Veen, who is the client for this project. I will later write a reflection where I will conclude if I achieved my goals.

## Personal Motivation

My interest in health and care has grown significantly since completing my S3 project. I have become interested in anything related to health care technology. I was fortunate enough to come into contact with Rudie van den Heuvel, who informed me that the Health Concept Lab (HCL) was still seeking students to assist with various projects, including a water monitoring system. Rudie mentioned the opportunity to grow my PCB and Python skills, which motivated me to join the HCL. I believe that the HCL and its teachers can provide me with the support and knowledge I am seeking to gain.

Table of contents

[1 Brief Description 2](#_Toc124852766)

[2 Preface 3](#_Toc124852767)

[2.1 Personal Goals 3](#_Toc124852768)

[2.2 Personal Motivation 3](#_Toc124852769)

[4 Background information 6](#_Toc124852770)

[4.1 Water quality monitoring background 6](#_Toc124852771)

[4.1.1 Previous work 6](#_Toc124852772)

[4.2 Stakeholders 6](#_Toc124852773)

[6 Main and sub questions 7](#_Toc124852774)

[7 Objective and end product 8](#_Toc124852775)

[8 Functional Design 9](#_Toc124852776)

[9 State machine 14](#_Toc124852777)

[10 Use Case Diagram 14](#_Toc124852778)

[11 System Architecture 15](#_Toc124852779)

[12 Hardware 16](#_Toc124852780)

[12.1 Microscope led 16](#_Toc124852781)

[12.1.1 Testing plan 17](#_Toc124852782)

[12.2 Stepper motor driver 18](#_Toc124852783)

[12.2.1 UART on the TMC2209 19](#_Toc124852784)

[12.3 Thermal Control 20](#_Toc124852785)

[12.3.1 The modules 20](#_Toc124852786)

[12.3.2 Schematic 20](#_Toc124852787)

[12.4 Pi Hat i2c EEPROM interface 21](#_Toc124852788)

[12.5 12V Power GPIO 21](#_Toc124852789)

[12.6 PowerManagementSystem 22](#_Toc124852790)

[12.6.1 Power Supply 22](#_Toc124852791)

[12.6.2 Preventing Supply Noise 22](#_Toc124852792)

[12.6.3 EMC 23](#_Toc124852793)

[12.6.4 ESD protection 23](#_Toc124852794)

[12.6.5 Power regulators 24](#_Toc124852795)

[12.6.6 12V vs 24V 24](#_Toc124852796)

[12.6.7 Trace width 25](#_Toc124852797)

[12.6.8 2 VS 4 layer PCB 26](#_Toc124852798)

[12.6.9 Coil whine 26](#_Toc124852799)

[12.7 Backpower protection circuit 27](#_Toc124852800)

[12.8 PCB versions 27](#_Toc124852801)

[12.8.1 PCB V0.1 27](#_Toc124852802)

[12.8.2 PCB V0.2 30](#_Toc124852803)

[12.8.3 PCB V0.3 31](#_Toc124852804)

[12.9 Focusing the lens 32](#_Toc124852805)

[13 Software 33](#_Toc124852806)

[13.1 Components 33](#_Toc124852807)

[13.1.1 Microscope LED 33](#_Toc124852808)

[13.1.2 TMC2209 Stepper motor driver 33](#_Toc124852809)

[13.1.3 PowerGPIO, Cooling Fan, Heating resistor 33](#_Toc124852810)

[13.1.4 Voicecoilmotor 33](#_Toc124852811)

[13.1.5 Pigpio 34](#_Toc124852812)

[13.1.6 Init function (PIGPIO daemon) 34](#_Toc124852813)

[13.2 Doxygen 34](#_Toc124852814)

[14 GUI 35](#_Toc124852815)

[14.1 GUI design 36](#_Toc124852816)

[14.1.1 Final design sketch 36](#_Toc124852817)

[15 State diagram 37](#_Toc124852818)

# Background information

## Water quality monitoring background

The ultimate goal of this system is to analyze and monitor water quality. As people age and take more medications, the remains of these medications can end up in urine and feces, leading to polluted water that can harm organisms. The Water Quality Monitor (WQM) will be a relatively cheap and useful device to use for testing the quality of water or other fluids. For now the focus is on water but the long-term goal is to create a portable device that can continuously monitor water quality and potentially even test for diseases.

### Previous work

The previous groups S4 and the S6, have contributed by designing pcbs and software for the WQM. S4 has focused on making the control board that can be attached to the raspberry pi. The raspberry pi would use a camera attachment and a microscope to observe water quality. The S4 setup was relatively small, but also limited in its functionality.

The S6 group has created a fully functioning setup with heaters, pumps for (precision) fluid movement and a spectrometer. They focused on this sensor instead of a camera attachment for the pi. Sadly the setup is rather large, prone to getting detached cables, since not everything is soldered and its control interface is not intuitive.

## Stakeholders

**Jeroen Veen** is the project leader. He will be guiding me mostly through all the ESE related questions, regarding programming, but also regarding the overall process and development skills. Since Jeroen is the project leader and the client, I will be working very closely to him.

**Rudie van den Heuvel** is a IPD teacher and knows a lot about materials, mechanics and designing in general. He may be able to provide me with knowledge on how to approach this project and whenever I want to do something with materials, he can assist.

**Health Concept Lab** is the lab that has been created around 3 years ago. The goal of this lab is to create a pool of knowledge on health on both mechanical as electrical ground. The lab is populated by students from S3 and above and there some teachers, who are well known in the subjects, as well.

**HAN** teachers can be consulted as well. Johan Brussen or Francesco Ursino for example are well known with power electronics. They can help me when it comes to supplying bigger loads with power. Other ESE or ELT teachers can also be of help. For EMC I can probably go to Ico van Diemen De Jel.

# Main and sub questions

In this paragraph I will list everything I need to know to get this project started. I need a basic understanding of the project, it’s size and the required approach to work as efficient as possible. By knowing what is expected, what is not, and how I think I can approach problems, it should be possible for me to create an approach plan.

* Main: What is the goal of this product?
* Main: What will be my contribution to this project?
  + Sub: What does the client think is achievable in this timespan?
* Main: What is water quality?
  + Sub: how can we measure water quality?
  + Sub: how can we conclude water quality?
* Main: What is the current process on this project
  + Sub: what did previous groups contribute?
  + Sub: what can be used or build upon in my project?

# Objective and end product

The goal is to design and develop a stable and sturdy device, called Rastaban, for monitoring water quality. The device will be equipped with a printed circuit board (PCB), we will call this the HAT. Further there will be software needed for the Raspberry Pi and this PCB, which will control the whole system and retrieve data collected by it.

Rastaban is a project that aims to improve water quality monitoring in various locations. The device will be used to monitor water quality by taking optical images and possibly video’s of the chemical reaction that emerges when a water sample is mixed with the reaction fluid in the reaction chamber.

The device will be used by environmental scientists, engineers, students from the Wageningen University and other professionals who are interested in monitoring and improving water quality. The device should be able to operate in a range of aquatic environments, including freshwater and saltwater. It should also be durable enough to withstand harsh weather conditions and rough handling.

The project will be divided into several phases, including research and development, prototyping, testing, and deployment. We aim to complete the first compact Rastaban prototype within the next 5 months. The first version of the Rastaban will be designed and developed specifically for measuring water quality. However, we will also keep DNA research capabilities in mind during the design process, and incorporate any necessary components or features suggested by Jeroen.

My contribution to this project will be the design and creation of the HAT and it’s software to control it. After this period I deliver the hardware, software and documents. This way the device will be usable, expendable and serviceable by anyone with basic knowledge on the subject.

# Functional Design

Before creating anything, as the V-model shows, it is important to create a design. I started by creating a list of functional specifications: what should the device be able to do? Should it be able to move fluids? Should it be able to heat up the fluids? Etc. These specifications will be the red line to follow while making decisions.

|  |  |  |
| --- | --- | --- |
| **Functional specifications** | | |
| **#** | **MoS CoW** | **Description** |
| F1 | **M** | **The device is collecting data with the biosensor setup** |
| F1.1 | **M** | The device can detect particles |
| F1.2 | **M** | The device has a light source that provides sufficient light for the camera |
| F2 | **M** | **The device can control the temperature of the examination chamber** |
| F2.1 | **M** | The device can heat the examination chamber |
| F2.2 | **M** | The device can cool the examination chamber |
| F2.3 | **M** | The device can measure the temperature of the examination chamber |
| F3 | **M** | **The device can manipulate fluid movement** |
| F3.1.0 | **M** | The device can add fluids and remove fluids from the examination chamber |
| F3.1.1 | **S** | The fluid should be able to be moved forward and backwards. |
| F4 | **M** | **The device needs to be able to connect to external sensors** |
| F5 | **M** | **The device must be portable** |
| F6 | **W** | The device will not be battery powered |
| F7 | **C** | The device could have It’s own identification |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Technical Considerations** | | |  |  |
| **Part** | **(Expected) Function** | **Pros** | **Cons** | **Notes** |
| **Raspberry Pi 4** | **Main computer** | **All in one computer, good documentation, good software support** | **Chip shortage creates vulnerability for supply chain** | **The compute model 4 is not an easy alternative since there is no build in camera connector. The Raspberry Pi 4 model B seems the most suitable** |
| **TPS61158** | **LED driver** | **Flexible digital and pwm brightness control, 100:1 pwm dimming ratio, soft start build in** | **Datasheet unclear if there is a switching value of 750 MHz or KHz** | **Mistakes in datasheet it seems, wrong frequency ratings** |
| **TPS6106x** | **LED driver** | **Pwm brightness control, digital brightness control, 1mhz fixed switching frequency** | **Made for multiple leds it seems, only 80% efficient** | **led disconnects during shutdown** |
| **TMC2209** | **Stepper driver** | **High quality, good documentation** | **Not yet found, maybe over specked** |  |
| **ST L297** | **Stepper driver** | **Reputable brand, low cpu usage** | **Expensive** |  |
| **Tmc2130** | **Stepper driver** | **The flag ship version of the tmc series** | **Expensive and overkill** |  |
| **Tmc2208** | **Stepper driver** | **Just as good as the 2209** | **higher impedance and so lower output amperage than tmc2209. It also has less features than tmc2209** |  |
| **DRV8870** | **H-bridge** | **Is able to supply 3.6A of current, enough for the Peltier module most likely** | **Not the best option, a higher current version (6A) would have been ideal: DRV8874** |  |
| **IRL540SPBF** | **High load switching mosfet (logic level)** | **A logic level high mosfet that can handle very high currents (20A)** | **A little expensive** |  |
| **TPS61169** | **LED driver** | **LED current can be set with resistor** |  |  |
| **TPS92360** | **LED driver** | **LED current can be set with resistor** |  | **Seems the same driver as the tps61169** |
| **P82B96** | **I2C ESD protection IC** | **Galvanic separation of i2c lines which results in high esd level protection. One package saves all** | **A little more complicated possibly than using passive components. Expensive.** |  |

|  |  |  |
| --- | --- | --- |
| **Technical specifications** | | |
| **#** | **MoSCoW** | **Description** |
| **T1** | **M** | **A Raspberry Pi will be used as the computer and controller** |
| **T2** | **M** | **The Raspberry Pi will connect to a PCB hat (PI-HAT)** |
| **T3** | **M** | **The device will use a camera connected directly to the pi via a flat cable to take pictures and videos** |
| **T.3.1.0** | **M** | **The camera will be stationary, the lens itself will move** |
| **T3.1.1** | **M** | **The device will be able to focus and reposition the lens via motors or moving magnets** |
| **T3.1.2** | **S** | **When using magnets, the drv8838 should be utilized** |
| **T4** | **M** | **The device will have a LED to create light for the microscope** |
| **T4.1** | **S** | **The led driver that should be utilized is the cn5711** |
| **T5** | **M** | **There will be stepper motors used** |
| **T5.1** | **S** | **The motors should be chosen according to the required strength** |
| **T5.2** | **S** | **The motor resolution should be below x degrees** |
| **T6** | **C** | **Some motors and sensors could be attachable via connectors** |
| **T6.1** | **S** | **All connectors should be standard (nonproprietary)** |
| **T7** | **M** | **The device will cool the examination chamber with a Peltier module** |
| **T7.1** | **M** | **The Peltier module must be actively cooled to function and prevent damage** |
| **T7.2** | **M** | **The device will heat the examination chamber via resistive heati** |
| **T8** | **S** | **A flat cable should connect the sensors and motors data to the PI-HAT** |
| **T8.1** | **C** | **The PI-Hat should distribute power to all devices including the Raspberry Pi** |
| **T8.2** | **W** | **The PI-HAT will not have any LEDs or status indicators** |
| **T8.3** | **C** | **The PI-HAT may provide a fan connector to cool the Raspberry Pi** |

# State machine

The device will mostly be working in this continuous simplified manner.

1. Prepare system for testing (flush system clean, reach required fluid temperature etc.).
2. When the fluids are ready, pump them to the reaction chamber for measuring.
3. Capture optical images/video’s.
4. Store the captured data on a storage medium or cloud based storage.

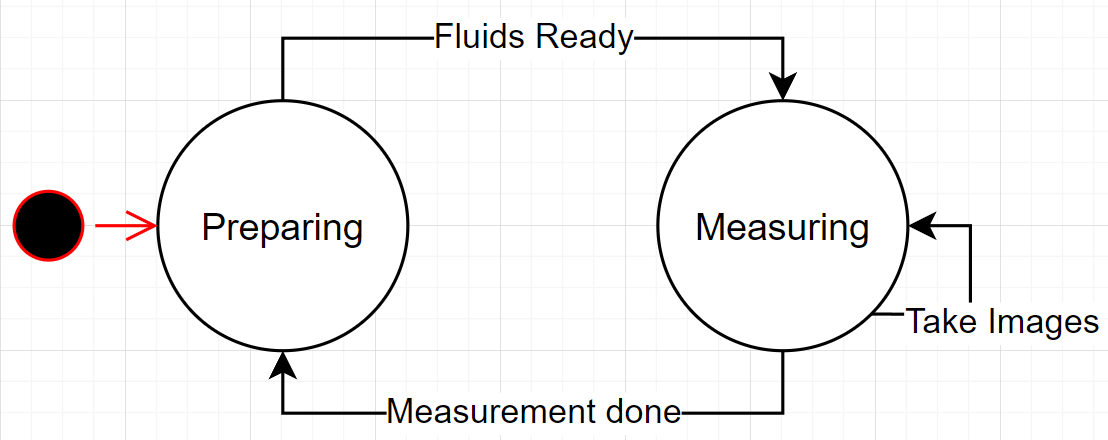


Figure 1 State machine

# Use Case Diagram

In the use case diagram we show the normal usage of the system in a static way. Here it is explained how the system functions in a block and how to Actor (user) provides input to the WQM and how the WQM may com in contact with a server solution.

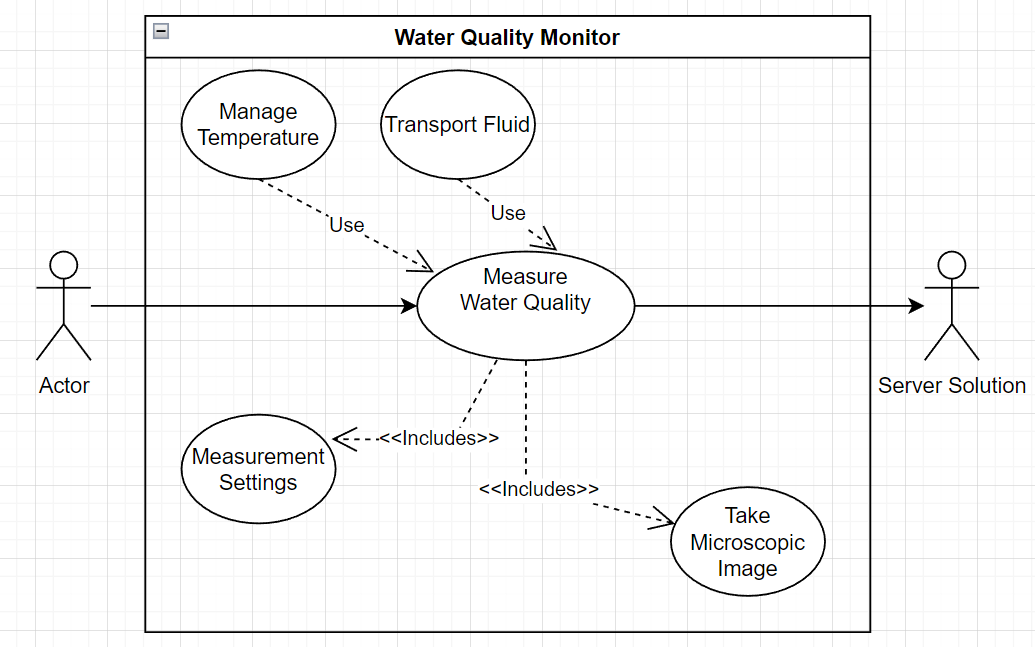


Figure 2 Use Case Diagram

# System Architecture

The system architecture gives a static view of how to system components are connected to each other.

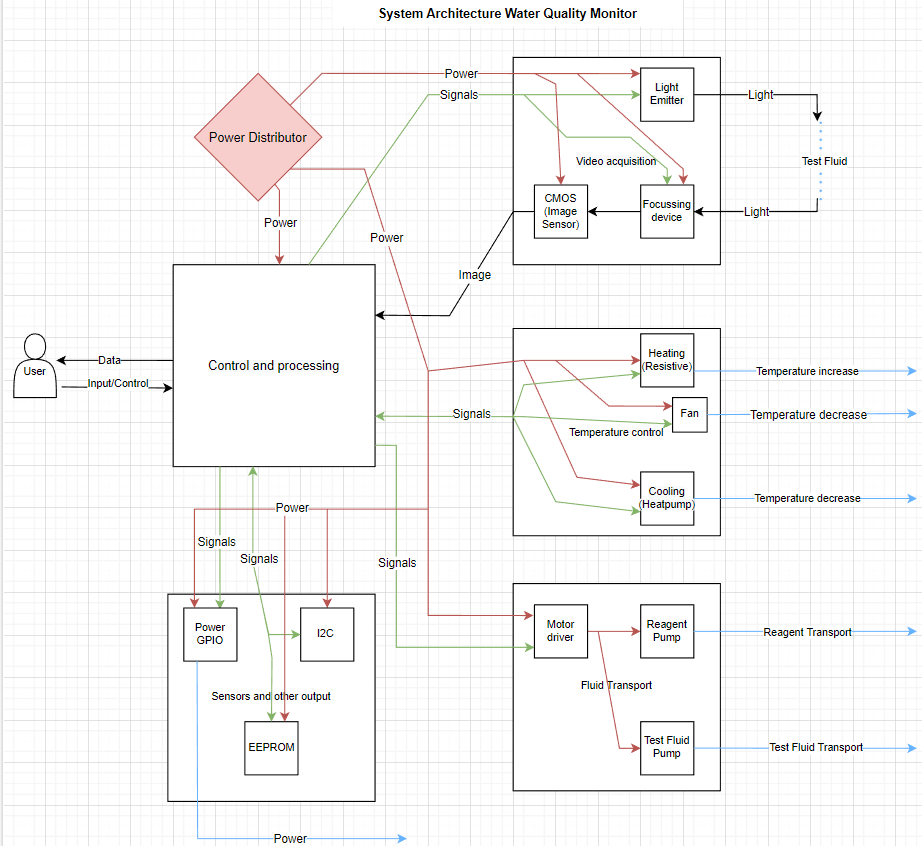


Figure 3 System Architecture

# Hardware

The Rastaban hardware consists of multiple components and categories. It controls, light intensity, temperature, movement and it has room for other sensors and output. All hardware will be described below

## Microscope led

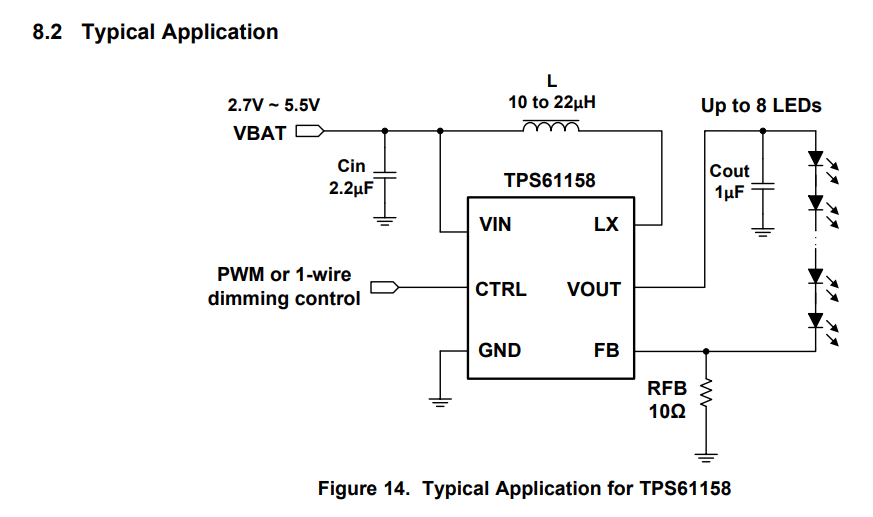
****The microscope led is the light source required for the camera to see an image in the reaction chamber. One of the options for driving the led was the TPS61158 driver.

Figure 4 Typical application TPS61158 driver

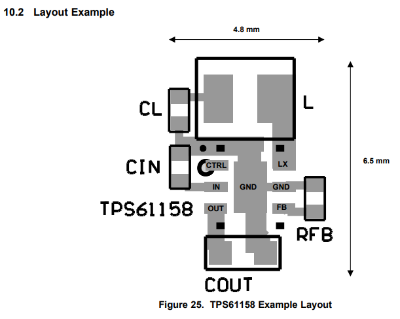
In this typical application circuit the input capacitor is to filter ripples from the power source and is there to function as a tiny buffer. The output capacitor is responsible for delivering current to the leds when the required current cannot be drawn continuously from the IC itself due to switching or power supply reasons. The inductor is there to function within the boost circuitry and has a high influence on the ripple and on maximum current and efficiency.

Figure 5 pcb recommendations

It is recommended to take advice from the pcb layout recommendations to avoid unsuspected behaviour due to unwanted resistance or capacitances.

However most of the component placements are common rules for PCB layout, so this layout recommendation can be considered trivial.

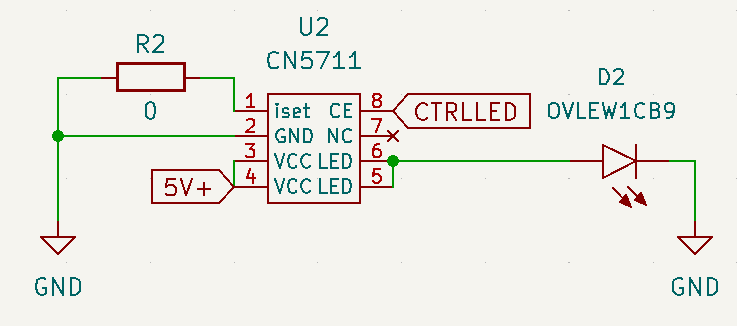
We later concluded that it was rather difficult to get our hands on the chips we wanted to use. This is why we chose to go with the CN5711 driver. This one is widely available on LCSC, AliExpress and similar shops.

Figure 6 finalized led driver circuit

The driver was used in the testing setup for the LED. We never intended to use this chip since I didn’t feel comfortable investing in a “no name/brandless” chip that may become obsolete sooner than later. However for this prototype it is sufficient. Driving the chip is done via a PWM signal, which is not proprietary, the code will always stay the same even with a new driver. R2 may be set if current limiting the driver is desired.

### Testing plan

In order to accurately capture images or videos of the reaction chamber, it is essential that the LED light used has a switching frequency that is higher than the camera's shutter speed. If the switching frequency is lower than the shutter speed, the resulting image or video may appear dark or flickering due to the incomplete exposure of the image sensor. To avoid this issue, it is necessary to determine the shutter speeds of the camera and select an appropriate LED switching frequency, particularly when dimming the light. This will ensure that the image or video is accurately captured with minimal artifacts.

Due to time limitations and other priorities I decided that testing this extensively is not smart. I tested the led by using a raspberry pi camera and my smartphone’s high speed camera starting a video capture and started dimming the led. I saw no direct flickering issues at most frequencies. I suspect that by further tweaking of the shutter speed of led switching frequencies it is possible to get even better results.

## Stepper motor driver

There are many different stepper driver models available. Since we opt to go for a compact PCB and since we don’t need to supply high currents it is safe to say that industrial grade or high current drivers aren’t going to be suitable for us. There are small drivers like the like the A4988 or the DRV8425 which are very simple to control and do not have any special functions except for micro stepping (Max 16uSteps). When searching for drivers, I found that 3D printing stepper drivers are very capable and well tested and documented by the community. So I found the TMC2208 and the TMC2209 stepper drivers.

The TMC2209 and TMC2208 are both stepper motor drivers that have identical pinouts, meaning they can be used interchangeably in terms of their physical footprint on a printed circuit board (PCB). For considerations regarding PCB layout, refer to Chapter 19 of the TMC2208 datasheet. It should be noted, however, that the TMC2209 is the recommended replacement for the TMC2208 and TMC2130 in newer designs, as it is the more current and up-to-date version of these drivers.

A typical stepper motor operates at a resolution of 200 steps per complete revolution. Through the use of micro stepping, the resolution of a stepper motor can be increased to as many as 51200 steps per revolution (depending on the specific motor). This results in a resolution of 1/256 steps. The implementation of micro stepping has been shown to reduce noise levels, improve the smoothness and accuracy of motor operation, and potentially increase energy efficiency. It is worth noting, however, that micro stepping can also decrease torque, particularly at higher speeds, which may lead to stalling. Some drivers are equipped with the ability to adjust the stepping mode based on speed in order to mitigate this issue.

Source on Micro stepping <https://www.youtube.com/watch?v=G8oGa2mawKk&t=68s>

Afbeelding met tafel

Automatisch gegenereerde beschrijving

Figure 7 Microstepping settings table

### UART on the TMC2209

The UART (Universal Asynchronous Receiver/Transmitter) interface is a serial communication protocol that allows for the transfer of data between devices. The TMC2209 stepper motor driver supports UART communication in addition to the traditional step/dir interface. Using UART to communicate with the TMC2209 can provide several benefits compared to using step/dir:

* Higher data transfer rates: UART allows for faster data transfer compared to step/dir, which can be useful for applications that require high-speed communication or precise control of the motor.
* More flexible control: UART allows for more advanced control of the motor, such as micro stepping, automatic load compensation, and stealthChop mode. These features are not available using the step/dir interface.
* Enhanced diagnostics and monitoring: UART allows for the monitoring of various internal parameters of the TMC2209, such as the temperature and current draw, which can be useful for debugging and performance optimization.
* Ease of use: UART can be easier to implement than step/dir, as it does not require the use of external pulse generators or counters.
* Using UART also removes the connection needed to step/dir. Using one RX line per driver and an enable pin to select the driver (with our current tmc2209 software).

Overall, using UART to communicate with the TMC2209 can provide improved performance, flexibility, and ease of use compared to using the step/dir interface. That’s why I settled for using this driver as our main stepper motor driver.

## Thermal Control

Thermal control will be in charge of providing energy and controlling all thermal regulating components of the Water Quality Monitoring (Rastaban) device fluids.

### The modules

The Thermal control consists of three power components:

1. Power Resistor
2. Peltier module
3. Fan (for Peltier)

These components are here to increase or lower the temperature of fluid that would be examined. The first two modules will be drawing very high currents (2A and higher) and therefore these powerlines need to be controlled and protected with care.

### Schematic

Figure 8 Thermal control components circuit

To control and protect the module and PCB from damage we need to building some passive protection. That’s why I chose some fuses that are roughly 1.25 times the value of the maximum current the module is supposed to use. When this value Is exceeded, the fuse will “blow” and the current will stop flowing.

The power mosfet used is the PMV15ENE**.** This NPN mosfet is designed to be controlled by a 3.3V logic level signal, which the raspberry pi uses as well. The mosfet is able to switch 6A and that’s why the component may seem overspecced. However, by using a higher current mosfet (lower RDS-on) there is higher efficiency because less energy is converted into resistive energy (heat). The component will also have a longer lifetime expectancy. Apart from this it also means there are no extra cooling components (heatsinks) required for this mosfet.

For the Peltier module we used a H-bridge since the Peltier module can cool or heat the same side when current is reversed. The module can help reaching higher temperatures in combination with the power resistor. When the current is reversed cooling the same surface can be achieved.

## Pi Hat i2c EEPROM interface

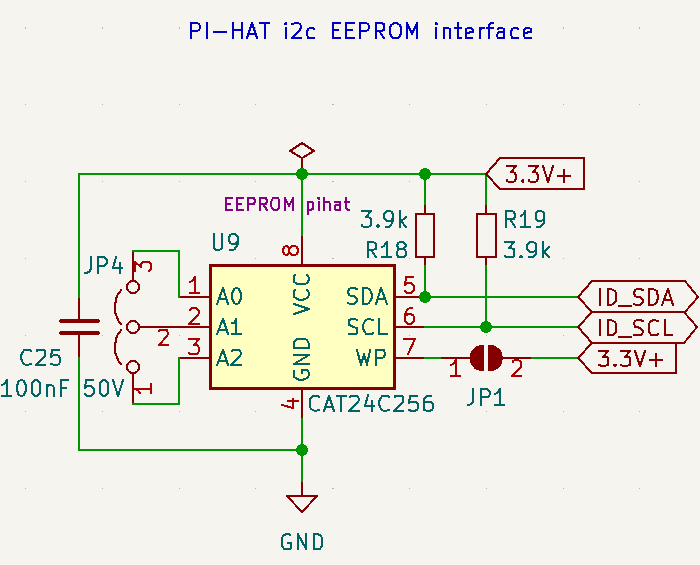
The i2c EEPROM interface was intended to create a script that would set the Raspberry Pi (RPI) in "Rastaban mode." The Sense Hat, developed by the Raspberry Pi Foundation, uses the EEPROM to identify the Sense Hat version and ID in the software used for the Hat. This allows the RPI to recognize different Hats and adjust settings, run appropriate code, and so on accordingly.

Figure 9 optional EEPROM circuit

However, we later realized that this may be unnecessary as we are programming the SD card to work specifically with this Hat and there will not be a need to support other boards. It is possible that the i2c EEPROM chip could be repurposed for safer data logging storage, as micro SD cards tend to become corrupt in RPI systems after extensive reading and writing.

## 12V Power GPIO

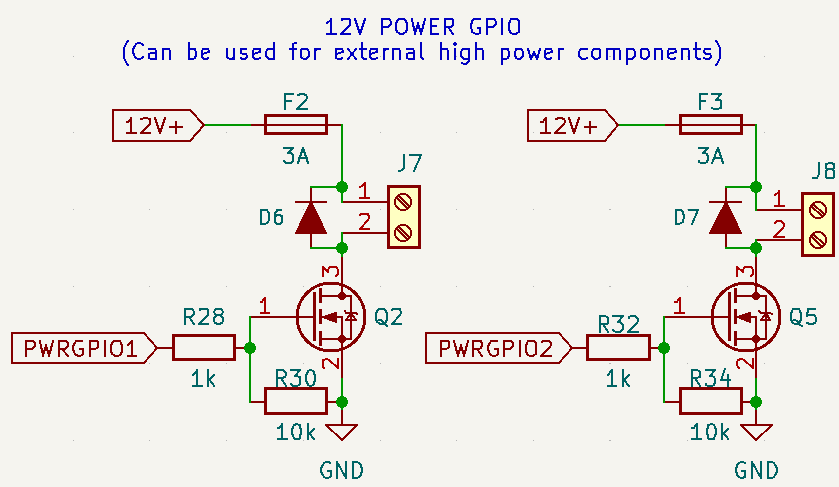
I added two power outputs for the user to connect high power loads to. The devices could theoretically draw to up to 6A, but I limited the fuses to 3A and the loads should be chosen accordingly or controlled with PWM to avoid tripping the fuse. The diodes are there to prevent back EMF destroying the mosfet or 12V supply when suddenly disconnecting an inductive load such as an motor. The 1k resistor on the PWRGPIO pin is for ESD protection of the pin and the 10k resistor is to define a state for the mosfet at all times even if the raspberry pi is disconnected physically.

Figure 10 PowerGPIO circuit

This prevents a floating pin and therefor unpredictable behaviour.

## PowerManagementSystem

The PowerManagementSystem (PMS) will be in charge of providing energy to all components of the Water Quality Monitoring (Rastaban). Rastaban has the following components that require power:

Figure 11 Maximum Power Usage



### Power Supply

This table is just an indication of the maximum power rating and it was made to give some insights. After seeing this table I think I can conclude that a 24V 10A (240W) power supply should be sufficient to run the device when it’s working in its normal consumption parameters. I don’t see the device ever drawing 11Amps. This power supply will be an external one for the time being, since it eliminates the need for a much bigger PCB and extra EMC and safety precautions. It also gives us the possibility to easily swap the PSU out if it gets damaged.

### Preventing Supply Noise

It is important to prevent noise that may be caused by some more power hungry components in the device. That’s the reason why I will try create power lines for 12V, 5.0V and 3.3V. The 12V rail will use a lot of power where the 5.0V and 3.3V are on the lower side of power usage. The use of capacitors will also smooth out the voltage dips that may occur while load becomes high.

#### Planes on PCB

There are multiple benefits to using ground planes, something that is already widely known. It improves thermals for heat inducing chips and it helps preventing EMC issues. It may however be wise to keep analog and digital grounds separated, to prevent ground loops and the noise it creates. We may use positive 12V or 5.0V planes as well, so we can transfer high currents to some loads without heating up the PCB, but this may introduce interference (EMC) with signal lines, something that is not tolerable.

Later in the project we figured that it is not possible to keep analog and digital grounds separated, since some ic’s have analog inputs ground that are connected to digital ground. If even one ic has this ground setup, than trying to separate the grounds won’t work. Adding the face that it makes routing the pcb more difficult, we will keep this idea in mind, but we won’t implement it for now. It is not smart to use 12V high power planes on the pcb since this can create magnetic fields that interrupt data lines or components.

### EMC

EMC (Electromagnetic Compatibility) refers to the ability of electronic devices and systems to function properly in their intended electromagnetic environment without causing interference to other devices or systems. In the context of PCB (Printed Circuit Board) design, EMC refers to the design practices and measures that are taken to ensure that a PCB does not emit or receive electromagnetic interference (EMI) that could affect the performance or reliability of other electronic devices. There are several strategies that can be used to prevent unwanted EMC issues on a PCB:

1. Use proper grounding: Proper grounding is essential for EMC compliance. Use a solid ground plane and ensure that all ground connections are made as direct and low-impedance as possible.
2. Use proper power distribution: Proper power distribution helps to prevent voltage drops and noise on the power supply lines, which can cause EMI. Use a dedicated power plane and ensure that the power supply lines have low impedance.
3. Use proper decoupling: Decoupling capacitors help to filter out noise on the power supply lines and should be placed as close as possible to the power pins of all active components.
4. Use shielded components: Shielded components, such as shielded inductors and transformers, can help to reduce EMI.
5. Use proper routing: Proper routing helps to reduce the length and proximity of high-frequency signals, which can reduce EMI. Use a differential pair routing technique for high-speed signals and avoid routing signals near sensitive components or traces.

By following these design practices, it is possible to prevent unwanted EMC issues on a PCB and ensure that the electronic system functions properly in its intended environment. I have tried to implement a few of these strategies like point 1, 3 and possibly 5 if the pcb size allows this. Some strategies are not viable, cost effective or mandatory to get the system working properly.

I used a ground plane on the front and back of all my 2 layer PCB designs. This makes connecting al ground connections easier but I suspect it also creates a “shield” for underlaying layers when using 4 layer PCB’s where the second layer is a ground layer. I further made sure that my coil for the buck converter is properly shielded with the same ground plane and I routed no data lines beneath the coil. Since the coil produces an electric magnetic field it is wise to do so.

### ESD protection

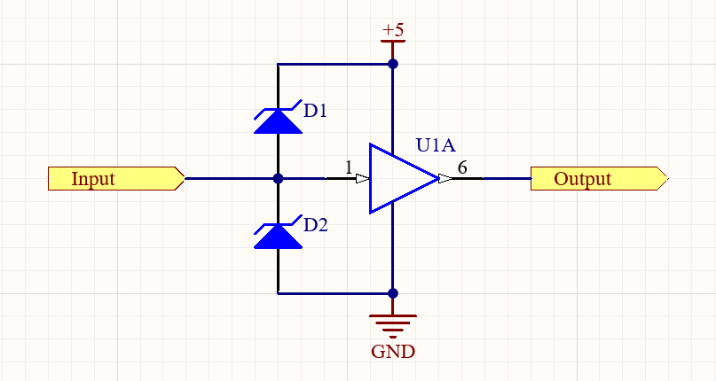
A voltage clamping diode circuit is used to prevent excessive voltages from accumulating at the input terminal of a buffer or differential input of an operational amplifier (op-amp). The circuit consists of two diodes, D1 and D2, which are reverse-biased under normal conditions. When the input voltage exceeds the supply rail voltage, D1 becomes forward-biased and conducts, limiting the voltage at the input. Similarly, when the input voltage falls below ground, D2 becomes forward-biased and conducts, again limiting the voltage at the input. This circuit is useful for protecting sensitive input stages, such as those found on I2C data lines.

Figure 12 ESD protection using TVS diodes

We later decided to implement this, as we can always omit the diodes on the PCB if it may not be necessary after all.

### Power regulators

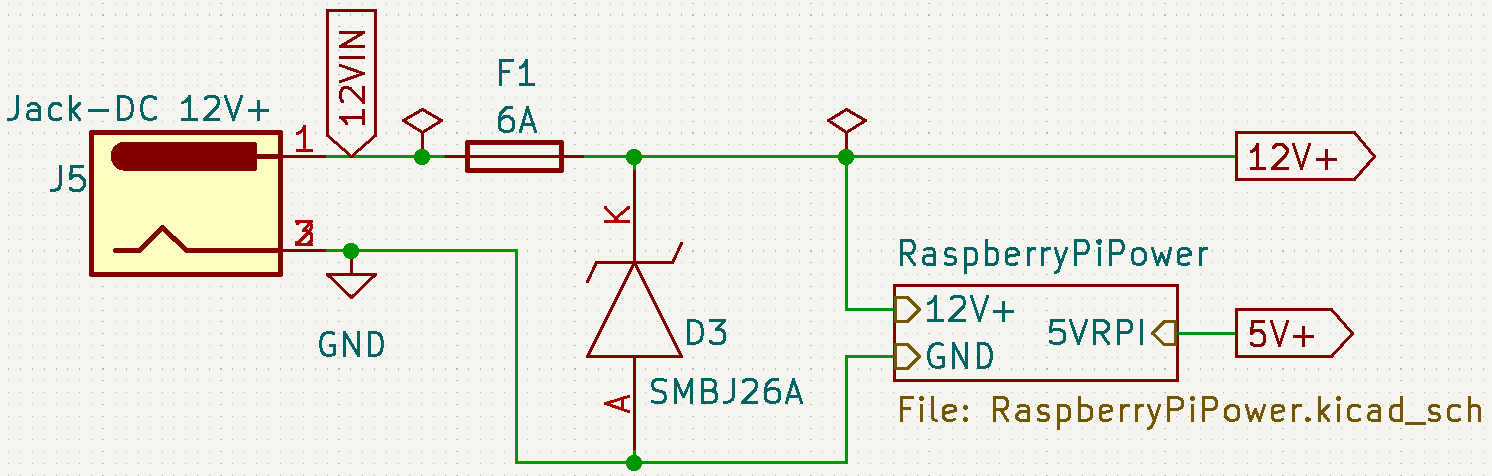
There are two main methods for adjusting the voltage in a circuit: Buck/Boost converters and linear regulators. Buck/Boost converters are more efficient and produce less heat, but they require more components and can cause noise on the supply line due to switching. Linear regulators, on the other hand, waste a lot of energy as heat but produce less noise on the output voltage. However, in cases where the difference between the input and output voltage is not significant, linear regulators can be more efficient than switching regulators. The choice between these two options depends on the specific requirements of the device. In this case, we prioritize functionality and repairability, so we would likely choose a linear regulator. We ultimately decided to use two step-down converters, both of which are switching regulators. The first one is used to power the Raspberry Pi at 5.0V and can deliver up to 3A if needed. I chose this because the Raspberry Pi will eventually be used with a camera and that can lead to higher current usage when compared to using just the raspberry without peripherals. Other components that also run on 5.0V, such as the microscope LED and the voicecoil driver (with a buck converter IC), are powered by this converter as well. The 5.0V converter also supplies the 5.0V breakout of the breakout power pin header. It is unlikely that protection will be necessary, as the LM2596 5.0 has built-in thermal and overvoltage protection. The second converter is used to provide 3.3V and is based on the TLV1117-33 IC, which can deliver up to 3A. The output of this converter is connected to the voice coil motor driver (DRV8838).

Figure 13 Power Management System circuit (simplified)

### 12V vs 24V

Since the primary and secondary stepper motors are designed to run on 12V we think it may be wise to choose a 12V supply. 24V gives us more headroom for voltage dips, but the motors can probably not handle the voltage difference. It is also less efficient to buck a higher voltage to a lower voltage, so keeping the difference lower is better. We do need to compensate for the possible voltage dips with capacitors and a PSU with a rather high current (probably 10A). I think that a 120W power supply would suffice for this prototype. As an addition we found that the Peltier module usually work on lower voltages like 12V, 5.0V and even 2.2V when they are really small (1cm by 1cm). bucking the voltage from 24 to 12 or lower and still having >3A currents will be very difficult when it comes to sizes of IC’s. using 12V is again beneficial in this case.

### Trace width

Afbeelding met tafel

Automatisch gegenereerde beschrijvingSince high currents will run through the board to some components, we need to create thicker wires other wise the resistance will be too high and the board will heat drastically, possibly leading to traces burning up or degrading over time due to temperature fluctuations.

Figure 14 Trace Width Calculator tool

I used an online calculator to get a overall indication of the track width required. Sadly it seems impossible to create a 5.62 mm trace for every high power net. I decided to go with a trace with that was still possible to route and that roughly resembled the width of the package pins of the power IC use. This may not be the most professional way to test and see if this would work, but it is the only way to go. It later seemed to work perfectly fine using 0.6mm wide traces. It would be a great experiment to check the temperature of the board with a high resolution thermal camera.

I included rounded tracks in the PCB design for two reasons:

1. Rounded tracks serve a functional purpose when signals are being transmitted at above 1Ghz.
2. Rounded tracks give the PCB a visually distinct appearance that can make it stand out. I found it nice to use since the rounded tracks remind me of water.

Overall, the inclusion of rounded tracks in the design serves mostly a aesthetic purpose.

### 2 VS 4 layer PCB

A PCB (Printed Circuit Board) consists of one or more layers of conductive material, typically copper, separated by insulating layers known as core and prepreg. The number of layers in a PCB can range from one to many, depending on the complexity of the circuit and the required performance.

Here are the main differences between 2-layer and 4-layer PCBs:

1. Number of layers: As the name suggests, a 2-layer PCB has two layers of conductive material, while a 4-layer PCB has four layers.
2. Layout complexity: 2-layer PCBs are typically limited to simple circuits with a small number of components, while 4-layer PCBs can accommodate more complex circuits with a larger number of components.
3. Signal routing: 2-layer PCBs have limited routing options and may require the use of vias (small holes that connect different layers) to route signals between layers. In contrast, 4-layer PCBs have more routing options and typically do not require the use of vias.
4. Performance: 4-layer PCBs can provide better performance compared to 2-layer PCBs, as they offer more routing options and can reduce the effects of crosstalk and noise.

In summary, 2-layer PCBs are suitable for simple circuits with a small number of components, while 4-layer PCBs are better suited for more complex circuits with a larger number of components and higher performance requirements.

In order to minimize the size of the PCB and facilitate its ease of transport and mounting on a Raspberry Pi, we chose to use a 4-layer PCB in the final design. This allows us to easily separate the power planes from the signal wires, which can improve the performance of the circuit. In future designs, it may be beneficial to include a ground plane between the signal and power planes to further improve the performance and EMC (Electromagnetic Compatibility) characteristics of the PCB.

### Coil whine

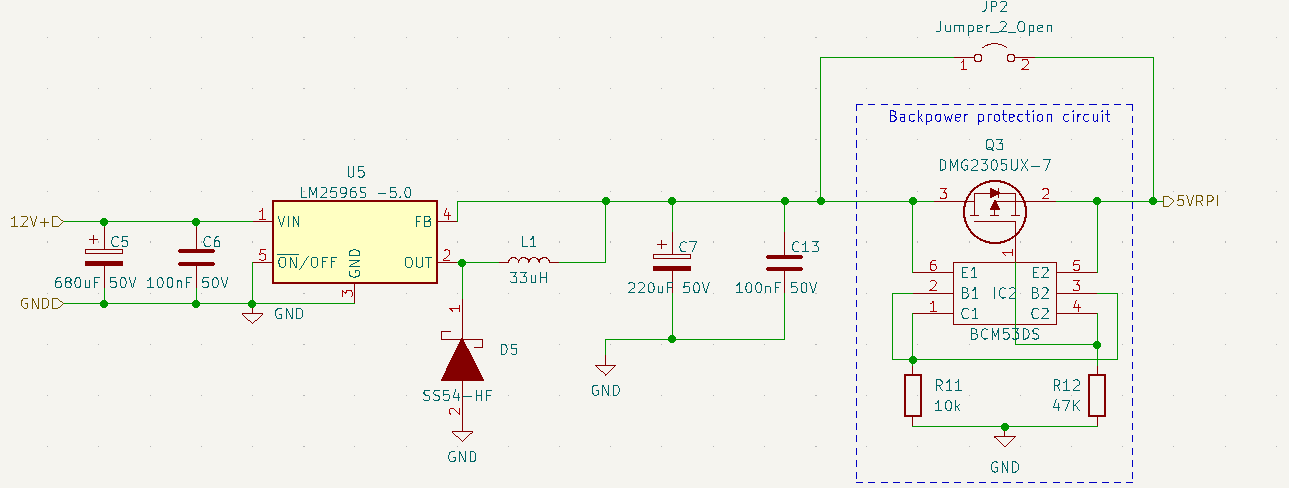
Upon connecting a minor load (300 mA) to the 5V Raspberry pi power circuit, I noticed that it generated a high-pitched whine. Upon further investigation, it was determined that the noise was likely emanating from the coil. Upon reviewing the circuit design, I realized that I had omitted the decoupling capacitor from the output of the LM2596 5.0 regulator. This likely resulted in the output producing pulses in the kHz range that were audible through the coil.

Figure 15 5V power circuit

To resolve this issue, I added an 220 uF capacitor to the output of the regulator. This effectively eliminated the whine and improved the performance of the circuit.

## Backpower protection circuit

The Backpower Protection Circuit (BPC) is designed to protect the Raspberry Pi from damage in the event that a 5V supply is connected to the micro USB port while a 12V supply is simultaneously connected to the barrel jack on the Pi Hat. However, due to a lack of necessary components, this circuit was not tested during the development of the Rastaban project. The BPC can be bypassed by using the solder jumper JP2. To further mitigate the risk of unintended power connections, the housing for the Rastaban project could be designed to cover the micro USB connector.

## PCB versions

When fabricating PCBs, it is common to go through multiple iterations before achieving a functional prototype. It is wise to test everything on a breadboard first, but this is not always feasible due to the size of certain components (not breadboard-friendly), the complexity of the circuit, and sometimes the parasitical behaviour of the breadboard, which can affect the performance of the prototype. During my internship, I designed three versions of the Rastaban HAT, as well as several simpler PCBs.

### PCB V0.1

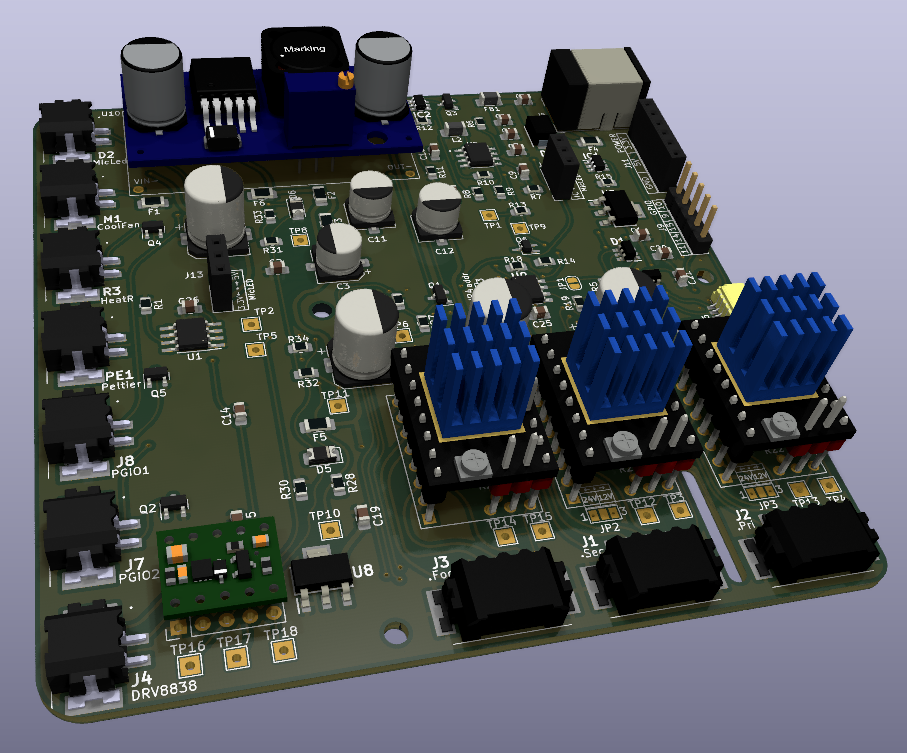


Figure 16 PCB V0.1

The first version of the PCB was designed to explore multiple options when it comes to driving the motors, led and power the components.

Successes:

* Most of the functions/ elements work
* This layout seems to be a good starting point, maybe some small refinements.
* The mounting holes and dimensions are all correct.

Sadly, the first version of the HAT had more problems than I expected, but I was able to make all the necessary improvements and get a fully functional prototype. I documented all the changes I made, so that I could easily apply them to future PCB’s.

Notes for V0.2 of the Rastaban PCB

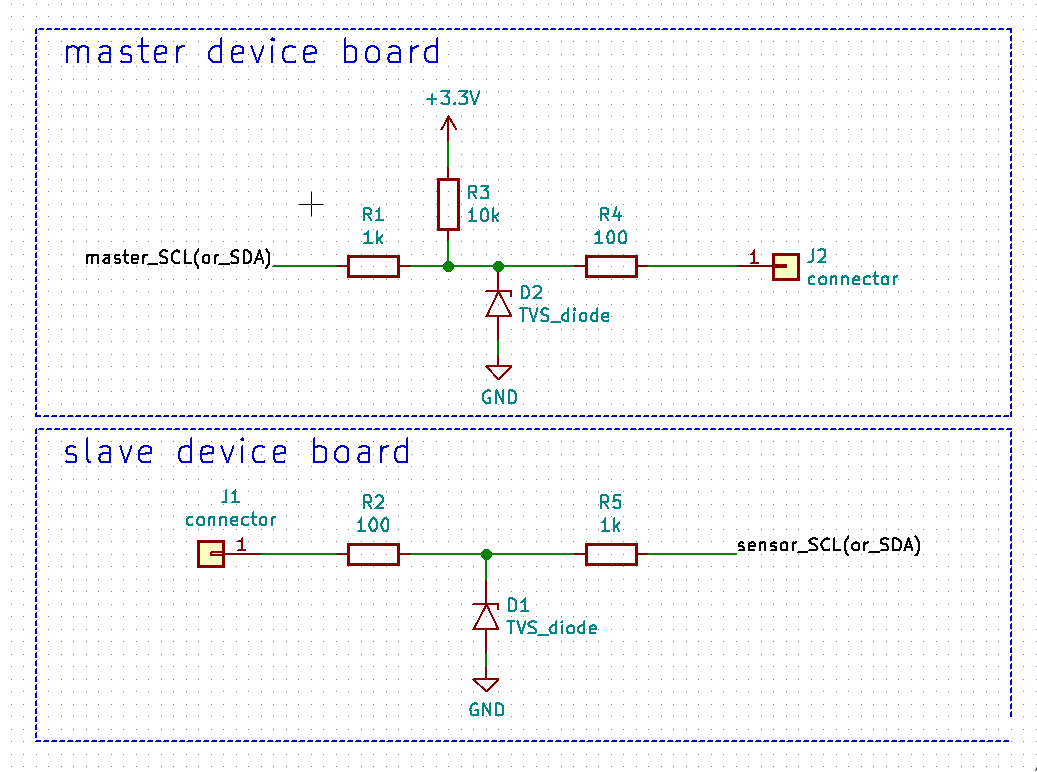
1. **CHECK THE ORIENTATION OF THE RASPBERRY PI HEADER FOOTPRINT(next time, dubbel, dubbel check)**
2. VCC pin on the drv8838 driver must be connected to 3.3V or 5.0V for it to work.
3. Try to draw high power traces away from signal traces (move connectors?)
4. For the 2209 we could use UART port, so we can control clock, microstepping etc, set parameters, delays, coolstepping, etc.
5. Buy the right parts 10uf 50V not 10uf 10V
6. FB1 will not work (made for 150mah), change for different inductor.
7. Order a lot of Phoenix ptsm connectors (both receptacle and plug)
8. Remove or find smd variants for fuses (remove some fuses/change values?)
9. Bypass backpower protection circuit (jumper)
10. Raspberry pi has decoupling capacitors (caps). Remove caps that are not necessary. Leave c11 and c8.
11. C5 should be 22uf.
12. Raspberry pi has buildin esd for gpio. Raspberry pi is connected to ground via power connector. Leave ESD footprints, but don’t solder them.
13. Simplify (or maybe even remove) raspberry pi power circuit.
14. Change inductor for rt7272A boost/buck circuit. Chip gets HOT, possibly switch to buck boost converter or worse case: external PSU for rpi (micro usb)?
15. Use a 40 pin FEMALE header on the pcb
16. Make the schematic more readable. Don’t run wires through each other.
17. Fix u7 (footprint is wrong, partly flipped)
18. Flat cable for 40 pin to 40 pin connection for easier, safer? For testing purposes.
19. Better diode polarity indication on silkscreen.
20. J12 silkscreen error: ic and fet are flipped
21. Possibly add pinout (names) to bottom of HAT
22. ENABLE focus must be using a hierarchal flag! Now it’s global, it’s not consistent.
23. All stepper motors work fine with 12V, they have more than enough torque even while using 16bit micro stepping. Remove the 24V buck boost converter.
24. Peltier module has not been tested (0.2 sense resistor not received)
25. Micled IC not tested (ic BCR420UW6-7 not received)
26. I2c hat functionality (special eeprom chip) not tested.
27. 5.0V for mic led not tested. 3.3V from u2 (tlv1117-33) works fine.
28. ESD protection not tested.
29. Maybe use the xl6009 for the raspberry pi and 5V line (4A is possibly more stable than the 3A) or copy a known design ([df robot?](https://www.tinytronics.nl/shop/en/power/voltage-converters/buck-(step-down)-converters/dfrobot-dc-dc-buck-converter-7-24v-to-5v-4a) )
30. Remove some testpoints to make pcb smaller and routing easier?
31. I2C NOT working probably due to esd protection ic. Remove the ic, it’s expensive and time consuming. Use diode alternative, which is optional:

Figure 17 ESD TVS diode option

Small discussion with Jeroen:

* Is a sense resistor a special type of resistor -> figure this out.
* Maybe 5 1ohm resistors in parallel work as an alternative for this sense resistor -> try this to find out if this is something to continue.
* Get current h-bridge working or search for alternative (make your own h-bridge with fets maybe, higher currents?)
* Try to fix the power circuit to gain knowledge on the subject.
* Remember: 1 small value cap for low frequency, 1 larger nf cap for higher frequencies.
* Drv8838 keeps turning somehow after disabling it.

Conclusions on process:

* Don’t reflow large capacitors. Do it by hand after the small parts have been soldered.
* 230 degrees Celsius is usually hot enough to melt the solder paste.
* Stencils for small footprints diy soldering, otherwise solder connections will fail.
* Always three dubbel check the most important things (footprints).
* Do some life testing so create mockups for pcb’s to test fit (also for Industrial designers).
* Check solder connections! Visual inspection! A mosfet with only 2 pins soldered instead of all three does not work.
* Don’t unplug things while they are powered on (hot plug/swap), plug things or measure things while the system is powered.

#### Current usage

I tested what the current usage of the components. Some components could not be tested and have been estimated.

|  |  |  |
| --- | --- | --- |
| Component type | Name | Current (A) usage tot. |
| LED | Microscope led | 0.077 |
| FAN | Cooling FAN | 0.150 |
| HEATER | Power Resistor (10w) | 1.180 |
| Cooler/heater | Peltier | ? 0-3.6 |
| PGPIO | X | 0-6 |
| PGPIO | X | 0-6 |
| Stepper motor (small) | Focus steppermotors | 0.5 (3 motors) |
| Stepper motor (large) | Primary steppermotor | 1.7 |
| Stepper motor (medium) | Secondary steppermotor | 1.5 |
|  |  |  |

### PCB V0.2

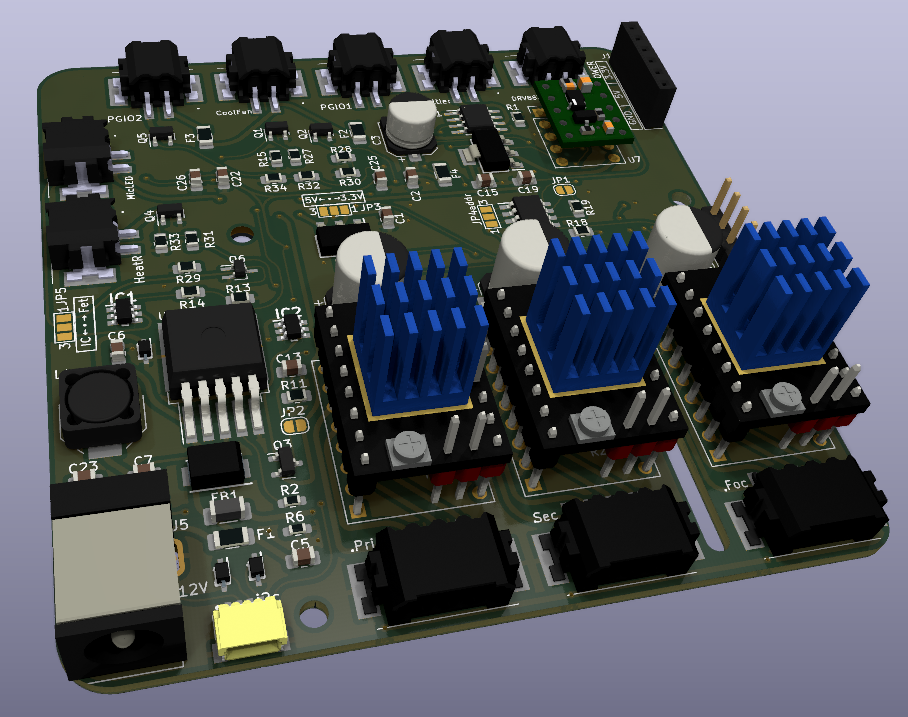


Figure 18 PCB V0.2

Successes:

* All functions/elements are working now, this version could be used for fluid testing purposes.
* This version is smaller than it’s predecessor L9,9CM; W9,9CM -> L8,4CM; W8,4CM.
* This version is capable of powering the raspberry pi through the GPIO header. External power for the raspberry pi Is no longer necessary.

Notes for V0.3 of the Rastaban PCB

|  |  |  |
| --- | --- | --- |
| # | Improvement | Finished |
| 1 | Figure out which hardware pins to use for what components led/focusmotor (drv8838) | NO |
| 2 | TMC 2209 uses different ms1 ms2 configuration for microstepping than the tmc 2208! Keep this in mind. | DONE, new design uses TMC2209 design config |
| 3 | Remove tmc2208 from design, focus on 2209 | DONE, no more tmc2208. |
| 4 | Use UART on 2209 and remove step/dir enable interface. | DONE, UART is only control way now |
| 5 | Use appropriate resistors for UART control on 2209 (resistance should decrease with increase in drivers). | DONE |
| 6 | Connect diag pin of 2209 for stall (stuck motor) indication for rpi? | NO |
| 7 | Checkout the problems with diag pin on tmc2209 (see pdf in datasheets) | NOT CHECKED |
| 8 | Change L1 footprint | Enlarged and changed to bourns square size |
| 9 | Copper zone for L1 | Copper ground zone added under L1 for MRI protection |
| 10 | Maybe move the camera connector to the left | YES, DONE |
| 11 | Corners in the hat header silkscreen. | Already satisfied in previous version |
| 12 | Raspberry pi power circuit coil whine should be resolved. | Solved, new values added to schematic |
| 13 | Footprints for all power circuit components should be enlarged/checked | Checked, DONE |
| 14 | Add logo v0.2 | DONE |

### PCB V0.3

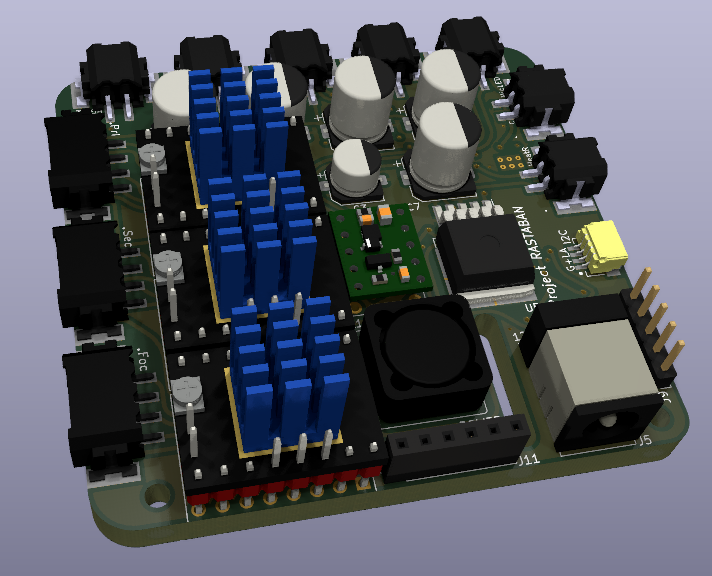


Figure 19 PCB V0.3

Successes:

* All functions/elements are still working.
* This version is smaller than it’s predecessor L8,4CM; W8,4CM -> L6.8CM; W6.5CM.
* Some footprints changed/enlarged.
* Removed step/dir connections and now using UART on TMC2209 driver.
* Added copper ground zone underneath L1 for MRI protection.
* Refined dimensions for camera cable hole.
* Newly created Rastaban logo added.

Notes for V0.4 of the Rastaban PCB

|  |  |  |
| --- | --- | --- |
| # | Improvement | Implemented/finished |
| 1 | Make the H-bridge compatible with different Peltier module sizes (different peltier module voltages) |  |
| 2 | Possibly break out more gpio pins to headers |  |

For final reference I placed the three PCB designs for Rastaban side by side to accentuate the size reduction and overall component placement improvement over the versions.

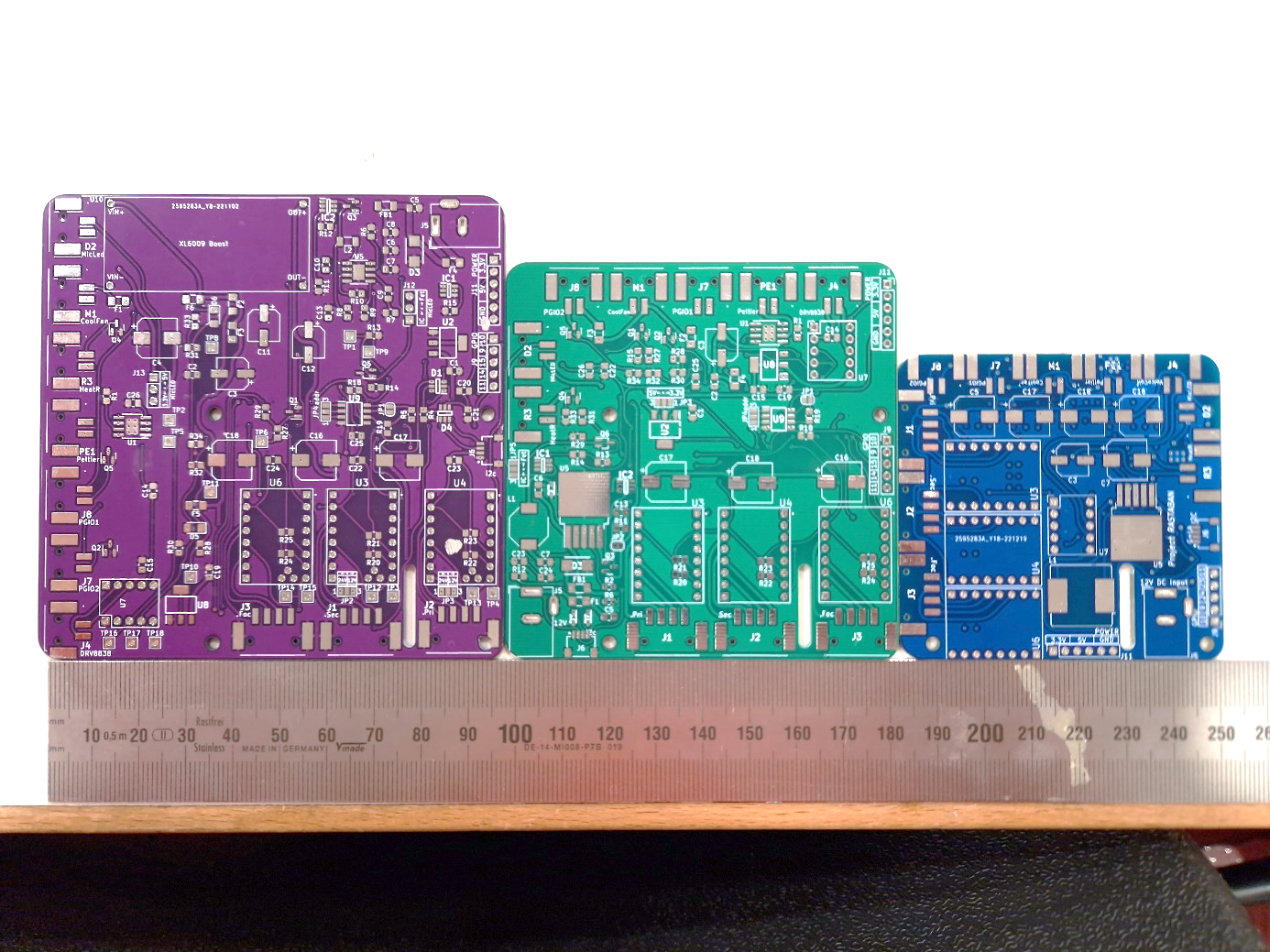


Figure 20 PCB V0.1-V0.3

## Focusing the lens

When imaging small objects through a lens, it is crucial to use a high-quality lens with sufficient magnification to clearly resolve the details of the objects. The working distance of the lens, or the distance between the lens and the object, should also be considered as it affects the ability to achieve focus.

There are several ways to adjust the distance between the lens and the object. One method is to use a voice coil, which works by magnetizing the coil to move it slightly closer or farther from the object, similar to the principle of a speaker coil. The Rastaban project can use a voice coil setup controlled by, for example, a DRV8838 driver. Another method is to use stepper motors to move the lens up and down. The design created by Jeroen uses three stepper motors in parallel for this purpose and will be controlled using TMC2209 drivers. It may be necessary to limit the current provided by the TMC2209 drivers through the UART settings in order to protect the small and vulnerable motors.

# Software

The software written for the Rastaban (HAT) has been made in the Python language. This language is chosen because it is widely used in the Raspberry pi community and it gives us the ability to use the PYQT GUI library. Other reasons are maybe for the ease of use and the reason that it is not compiled but interpreted code, which makes it faster for prototyping purposes. Most of the code is written in an object oriented style. I find this code looking cleaner and more maintainable than the regular coding style I know.

## Components

To give a short overview of the code, I will describe all components in short. I will not go into detail on the functions of the components, since the DoxyGen file is much more conclusive and is fluid in it’s versions, while this document may become obsolete sooner or later. Component list:

* Microscope LED
* Stepper motor driver
* PowerGPIO
* Cooling Fan
* Peltier Module
* Voicecoilmotor
* PIGPIO
* Init (PIGPIO)

### Microscope LED

The CN5711 is utilized to control the microscope led. The device is driven by a Hardware PWM signal send by the raspberry pi. The code can control the LED’s brightness, switching frequency, duty cycle and on/off state.

### TMC2209 Stepper motor driver

We can use both step/dir commands to control the motor drivers for “legacy” stepper motor control, as we can use UART, the more modern way to control and set the stepper motor drivers. V0.3 of the Rastaban PCB only uses UART to control the stepper motor drivers.

The TMC2209 stepper motor driver can be controlled through its UART (Universal Asynchronous Receiver/Transmitter) interface. To use this interface, a UART-to-serial converter (such as a USB-to-serial adapter) is used to connect the TMC2209 to a microcontroller or computer. The microcontroller or computer can then send commands to the TMC2209 using a specific protocol, and receive status information from the TMC2209 in return. This allows for real-time customization and monitoring of the driver's behavior. We use the following library to control the drivers via UART: <https://github.com/Chr157i4n/TMC2209_Raspberry_Pi>

### PowerGPIO, Cooling Fan, Heating resistor

The power GPIO pins are driven using software PWM as well as the Cooling Fan and the Heating resistor. This code drives the mosfet that is responsible for switching the load.

The Peltier driver is controlled using two software PWM signals. Sending to IN1 and IN2 will control the output of the H-bridge.

### Voicecoilmotor

The Voicecoilmotor code controls the voicecoil for focusing the lens. The code was designed for the DRV8838 motor driver.

### Pigpio

Pigpiod or pigpio is a utility which launches the pigpio library as a daemon.  
Once launched the pigpio library runs in the background accepting commands from the pipe and socket interfaces.  
The pigpiod utility requires sudo privileges to launch the library but thereafter the pipe and socket commands may be issued by normal users. There are several reasons why we might choose to use the pigpio library:

* It provides a simple interface for controlling the GPIO pins: The pigpio library provides functions for setting the direction and level of individual pins, as well as for configuring PWM and detecting changes on the pins.
* It is lightweight and efficient: The pigpio daemon runs in the background and communicates with the pigpio library via a socket, which means that it has minimal overhead and does not require any additional libraries to be installed.
* It is flexible: The pigpio library allows users to control the GPIO pins from multiple programming languages, including C, Python, and Perl.
* It is well documented: The pigpio library comes with detailed documentation that explains how to use the library and troubleshoot any issues that may arise.

Overall, the pigpio library is a popular choice for controlling the GPIO pins on a Raspberry Pi because it is easy to use, efficient, flexible, and well documented. That is the reason why we choose to implement it in our Rastaban code.

### Init function (PIGPIO daemon)

I created an init function for the Raspberry Pi to address unexpected behaviour of the GPIO pins when using the pigpio daemon. Some pins were unable to go LOW or did not properly utilize PWM. If you wish to run Rastaban code that utilizes PIGPIO functions, you can run the Init function once to start the pigpio daemon. The function will indicate whether the daemon was already running and display its process ID. By restart or after a shutdown of the raspberry pi, the daemon would be stopped. It is possible to run this code on start-up of the RPI, but it may be better to run this code when the final Rastaban code has been created.

## Doxygen

I chose to use Doxygen because for the following reasons:

1. Improved code readability: Doxygen can help you create clear and concise documentation for your code, which can make it easier for other developers to understand and use your code.
2. Enhanced collaboration: Doxygen can help you document your code in a way that is more accessible to other developers, which can facilitate collaboration and make it easier for others to contribute to your project.
3. Increased code maintainability: By documenting your code with Doxygen, you can make it easier for other developers to understand and maintain your code over time. This can be particularly important for larger projects with many contributors.
4. Professional-quality documentation: Doxygen can generate professional-quality documentation in a variety of formats, including HTML, LaTeX, and PDF, making it easy to create high-quality documentation for your project.

Overall, using Doxygen can help create more readable, maintainable, and collaborative software projects by providing comprehensive documentation for my code. Therefore, all the code for the Rastaban has been processed using Doxygen and is primarily documented in this way.

# GUI

It was planned to control the Rastaban project via an GUI. This could make it easier to control the setup for the end user. As said before the user may be a student, teacher or possibly a researcher who most likely has not a very wide understanding of the Rastaban code or product as a whole. This GUI should make it easy for them to interact with the device. Below is shown the function of the GUI in the system.

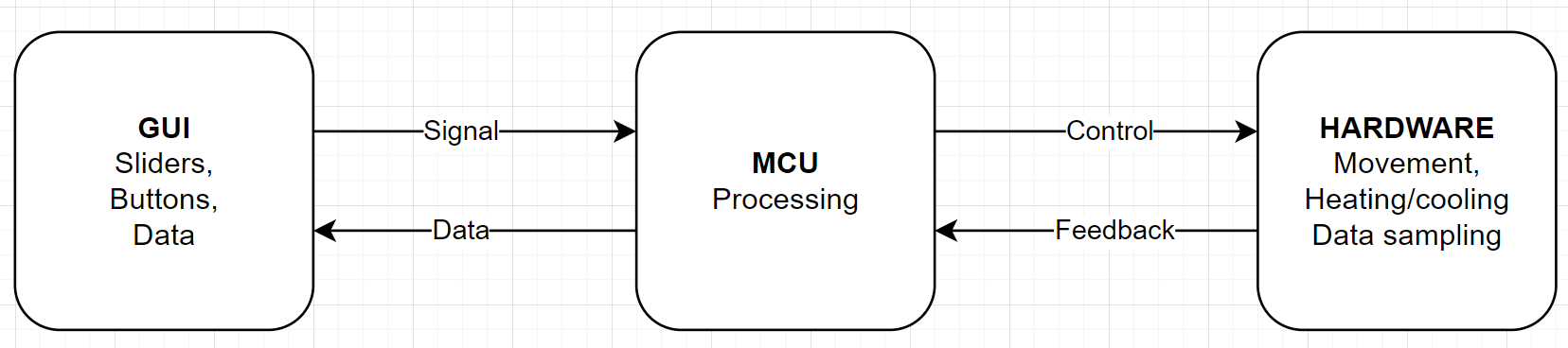


Figure 21 Application of GUI in the system

I created a crude window sketch for controlling the Rastaban project.

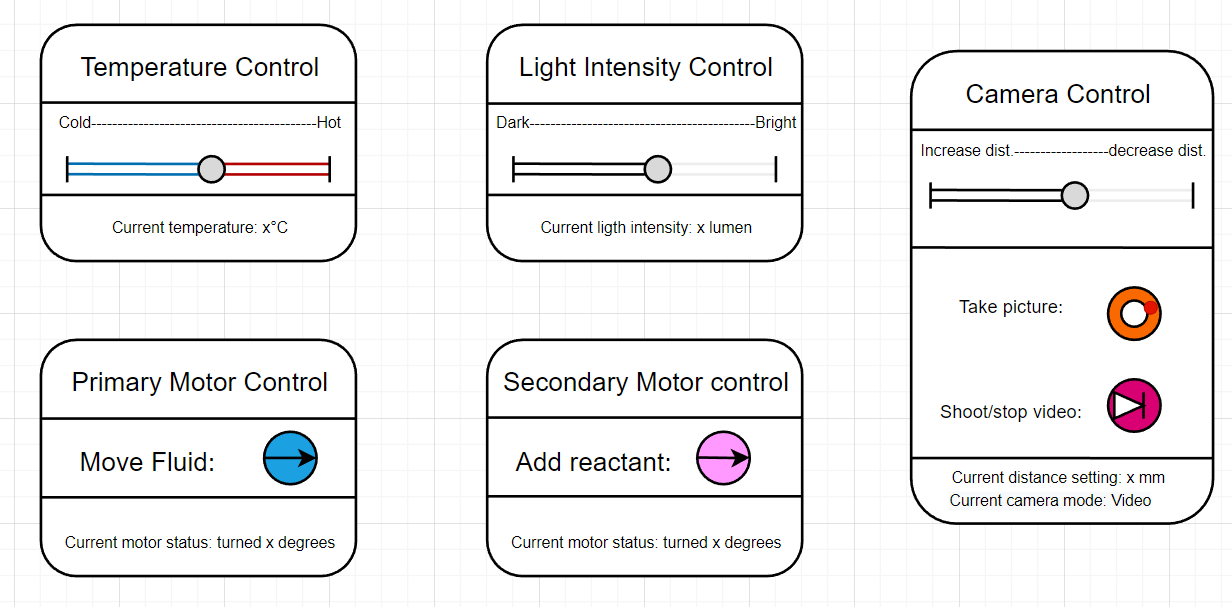


Figure 22 Windows sketch GUI

## GUI design

Figure 23 GUI widget options in one window

After having a general idea of how I wanted my GUI to function, I searched for possible input methods to put in my GUI. There are many options, sliders, dials etc. I ultimately decided to go with, in my opinion, easiest and safest option, by using only buttons and arrows that need to be clicked or scrolled on. I think that using a dial can be useful for an led for example for quick testing, but not for high power components like the Peltier or heating resistor. I think it would be dangerous to accidentally turn a dial to these high power components and a dial or slider doesn’t seem to be precise enough compared to setting numbers.

### Final design sketch

Afbeelding met tekst

Automatisch gegenereerde beschrijvingUltimately I think that functionality and safety in the GUI are of highest priority. That’s why in my final design sketch I made buttons to toggle enable/disable to specific blocks of control or the whole control window.

Figure 24 Final design sketch GUI

Too have full and precise control over the components, numbers can be set manually in the text boxes. The numbers can be changed by typing them in, scrolling on the box or clicking on the arrow, up/down.

This window/GUI may be build upon to give it functionality and add signals and slots to integrate it with the main Rastaban code.

# State diagram

This diagram gives the overall view of how the state machine will work if Rastaban would be used in the field, functioning automatically or if it would be controlled manually.

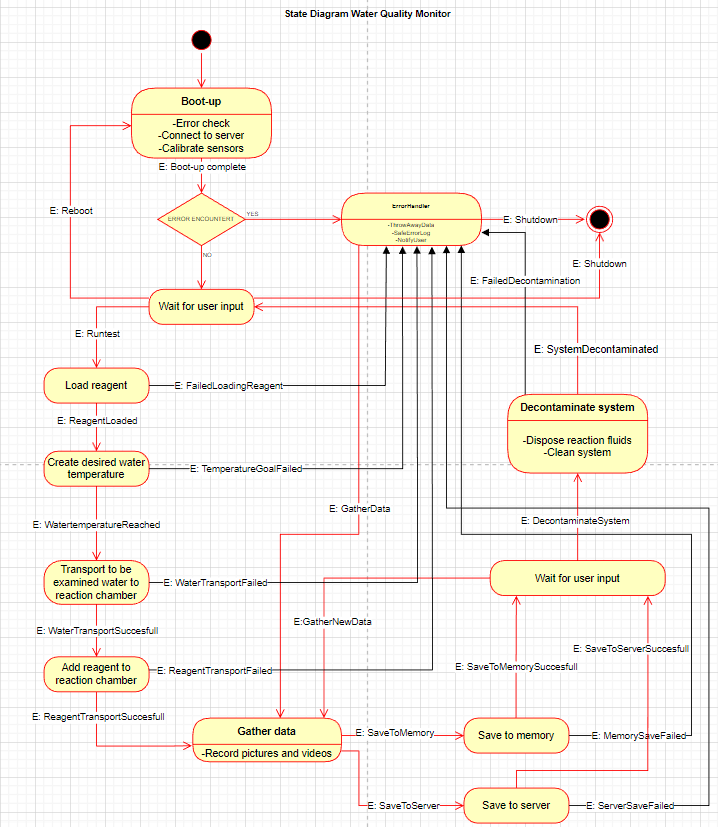


Figure 25 State diagram