POA Internship

**Main report internship assignment Health Concept Lab**

**Student:** Casper R. Tak

**Studentnumber:** 657313

**Client**: Rudie van den Heuvel

**Coach:** Jeroen Veen

**High School:** HAN Arnhem

**Education:** Embedded Systems Engineering

**Date:** 15-12-2022

# Preface

After attending both the S3 and S4 projects at the Health Concept Lab (HCL) at the HAN, I decided that this would also be a great place to do my internship. Since I have some learning goals left from my S4 project, I would like to finish those in my internship period. The goals I Have set are as follows:

* Improving and applying my planning skills to create more structure in my work approach
* Exploring and improving even further on PCB design by designing a PCB with KiCad V6
* Improving my programming basics and skills withing the language C, C++ and Python
* Explore and expand my horizon

The assignment that will allow me to get to these learning goals will be the fluid analysing device project. This project will focus on getting a better, smaller version of the fluid analysing device that was build by the S6 project students from previous years. The project is developed by Jeroen Veen, who is therefor my client.

This system will eventually be used as a mean to analyse water quality. Since people use a lot of medicine, especially since they get older and older, the urine and defecation contains medicine remains. This results in water getting polluted with these remains and that is harmful for organisms. The goals in the future will be to create a portable device that can easily and continuously monitor the water. Even further in the future the device will not only be used for fluid analysing but also recursive testing for (multiple) deceases.

Inhoud

[Preface 2](#_Toc122011525)

[Motive 4](#_Toc122011526)

[Companies and individuals involved 4](#_Toc122011527)

[Main and sub questions 4](#_Toc122011528)

[Goal and end product 5](#_Toc122011529)

[Theory 5](#_Toc122011530)

[Schedule 5](#_Toc122011531)

[Design 6](#_Toc122011532)

[GUI 11](#_Toc122011533)

[Statemachine 11](#_Toc122011534)

[State diagram 12](#_Toc122011535)

[Use Case Diagram 13](#_Toc122011536)

[System Architecture 14](#_Toc122011537)

[KiCad 15](#_Toc122011538)

[Why KiCad6? 15](#_Toc122011539)

[Plugins 15](#_Toc122011540)

[How to learn KICAD6 17](#_Toc122011541)

[The design approach 17](#_Toc122011542)

[System Components 19](#_Toc122011543)

[Microscope led 19](#_Toc122011544)

[Testing plan 20](#_Toc122011545)

[Driver circuit decision table 20](#_Toc122011546)

[Stepper motor driver 21](#_Toc122011547)

[Later findings 21](#_Toc122011548)

[Micro stepping 21](#_Toc122011549)

[UART on the TMC2209 22](#_Toc122011550)

[UART Problem with multiple drivers 22](#_Toc122011551)

[Conclusion multiple drivers UART error: 26](#_Toc122011552)

[Thermal Control 28](#_Toc122011553)

[The modules 28](#_Toc122011554)

[The circuit 28](#_Toc122011555)

[EMC 28](#_Toc122011556)

[PowerManagementSystem 29](#_Toc122011557)

[Power Supply 29](#_Toc122011558)

[Preventing Noise 29](#_Toc122011559)

[Planes on PCB 29](#_Toc122011560)

[EMC 29](#_Toc122011561)

[ESD protection 30](#_Toc122011562)

[Power regulators 30](#_Toc122011563)

[12V vs 24V 30](#_Toc122011564)

[Trace width 31](#_Toc122011565)

[Coil whine 31](#_Toc122011566)

# Motive

My interest in health and care have grown exponentially since my S3 project. I got interested in almost anything health care technology related. I got into contact with Rudie van den Heuvel, who told me that the health concept lab was still looking for students that could help with several project including the water monitoring system. He told me about the possibilities of growing with my PCB and Python skills and this is what got me motivated to join the HCL. I think the health concept lab and it teachers are able to provide me with the help and knowledge I seek to gain.

# Companies and individuals involved

**Jeroen Veen** is the project leader. He will be guiding me mostly through all the ESE related questions, regarding programming. 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 ELC teachers can also be of help. For EMC I can probably go to Ivo van Diemen De Jel.

# Main and sub questions

The soul reason of creating this POA is to gain an 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 things, it should be possible for me to create a planning.

To get a understanding of the project’s size and the current challenges, I need to know what the problems are.

Problem: there is currently no all in one compact and functional driver board for the device.

# Goal and end product

The goal is to create a water analysing device. My goal would be to create a printed circuit board along with software for the raspberry pi that will control the monitoring system and retrieve all the data collected by it.

# Theory

See OneDrive documents.

I’m going to approach this project via the V-model

# Schedule

See schedule document.

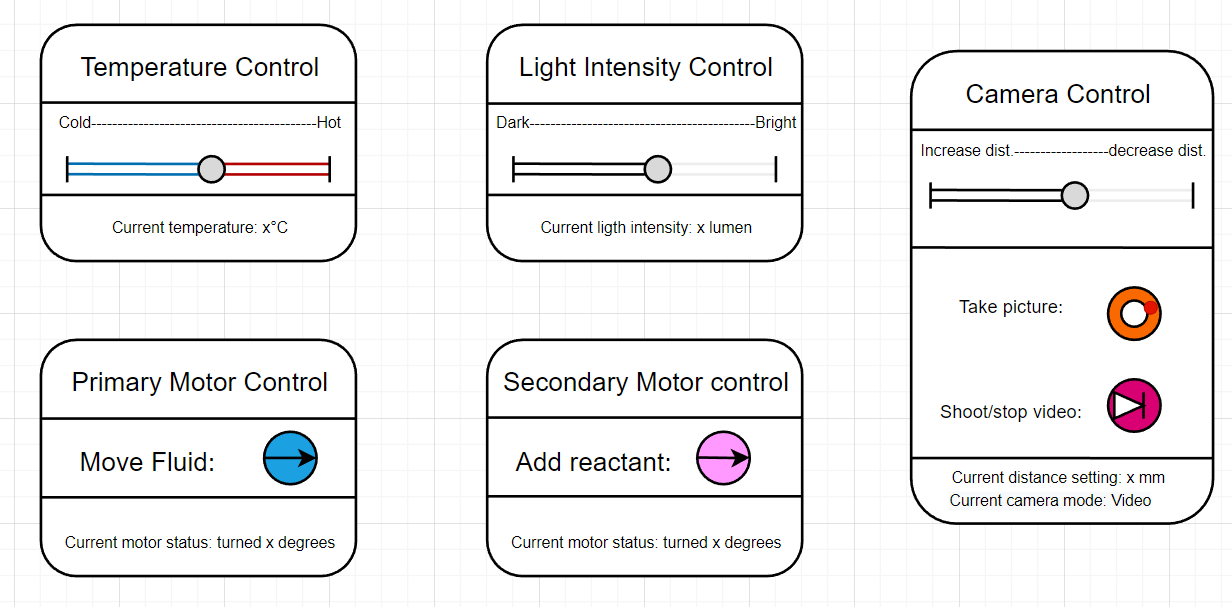
# Design

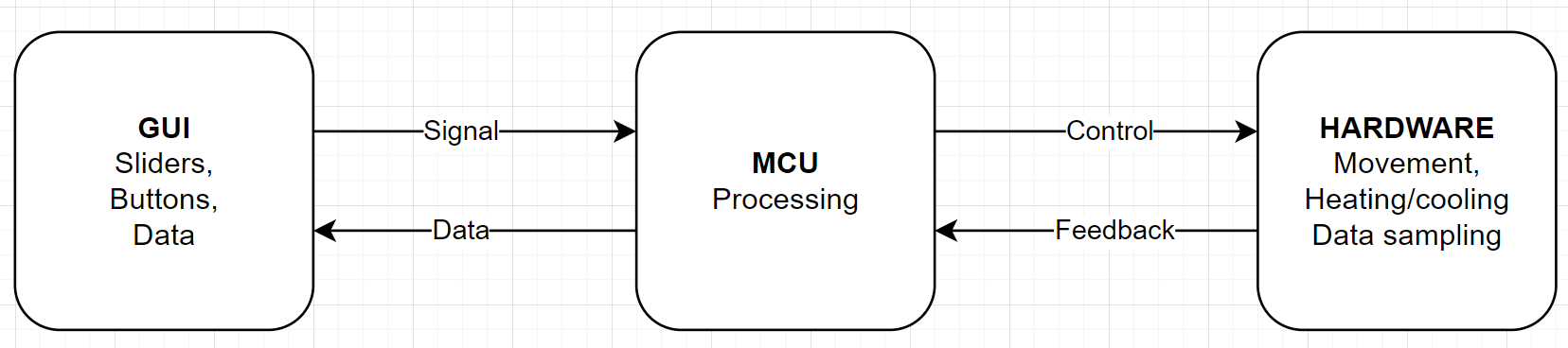
|  |  |  |
| --- | --- | --- |
| **Functional specifications** | | |
| **#** | **MoS CoW** | **Description** |
| F1 | **M** | **The device is collecting data with the biosensor setup** |
| F1.1 | **M** | The device can detect particles |
| F1.2 | **M** | The device has a light source that provides sufficient 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 is able to decontaminate the examination chamber** |
| F5 | **M** | **The device needs to be able to connect to external sensors** |
| F6 | **M** | **The device must be able to be carried by men** |
| F6.1 | **S** | The device should be as small as possible |
| F6.2 | **S** | The device should be as light as possible |
| F6.3 | **W** | The device will not be battery powered |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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.** |  |

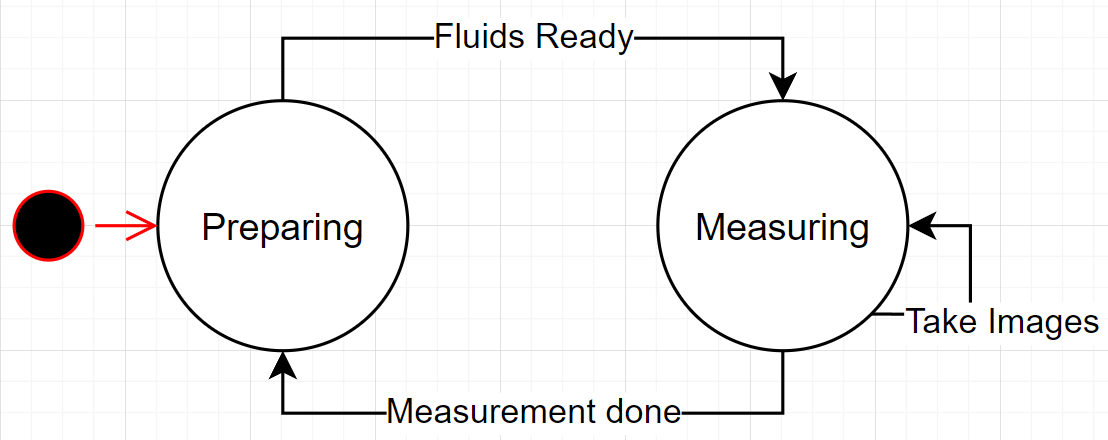
|  |  |  |
| --- | --- | --- |
| **Technical specifications** | | |
| **#** | **MoSCoW** | **Description** |
| **T1** | **M** | **A Raspberry Pi will be used as the computer and controller** |
| **T2** | **M** | **The Raspberry Pi will connect to a PCB hat (PI-HAT)** |
| **T3** | **M** | **The device 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** |

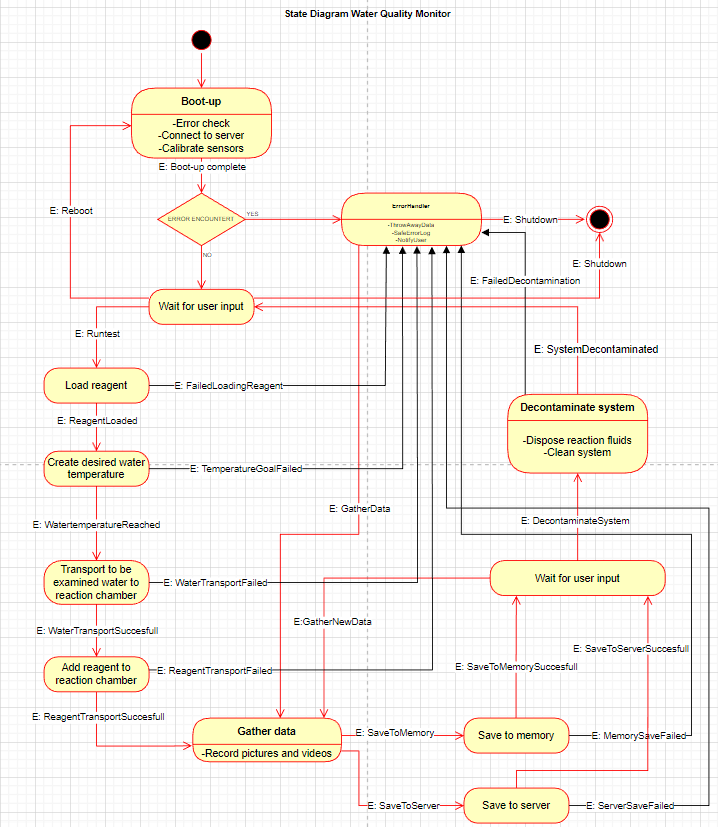
# GUI

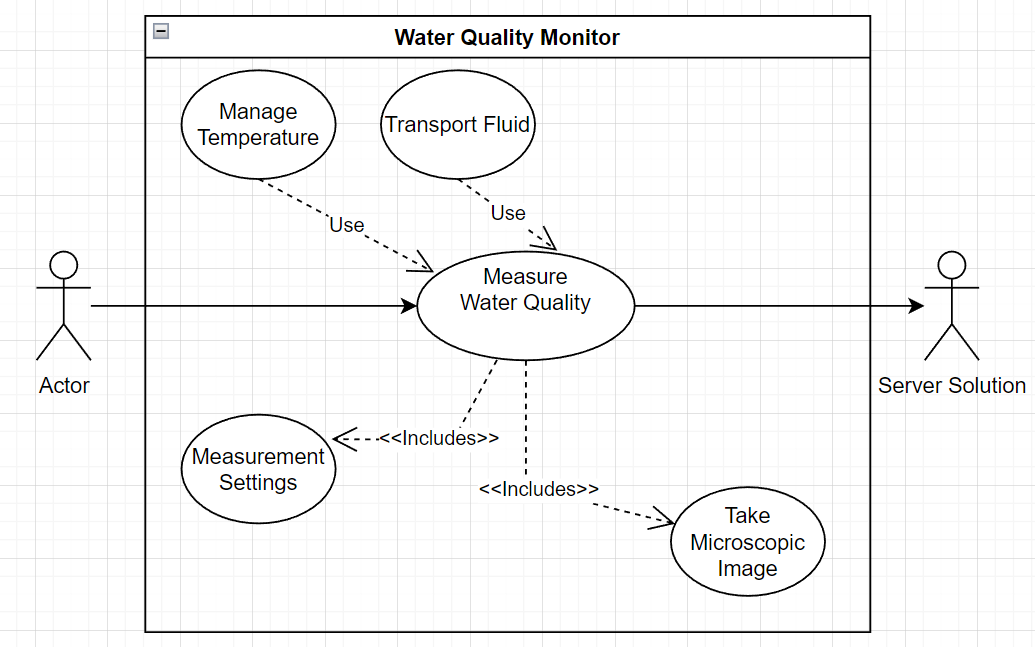




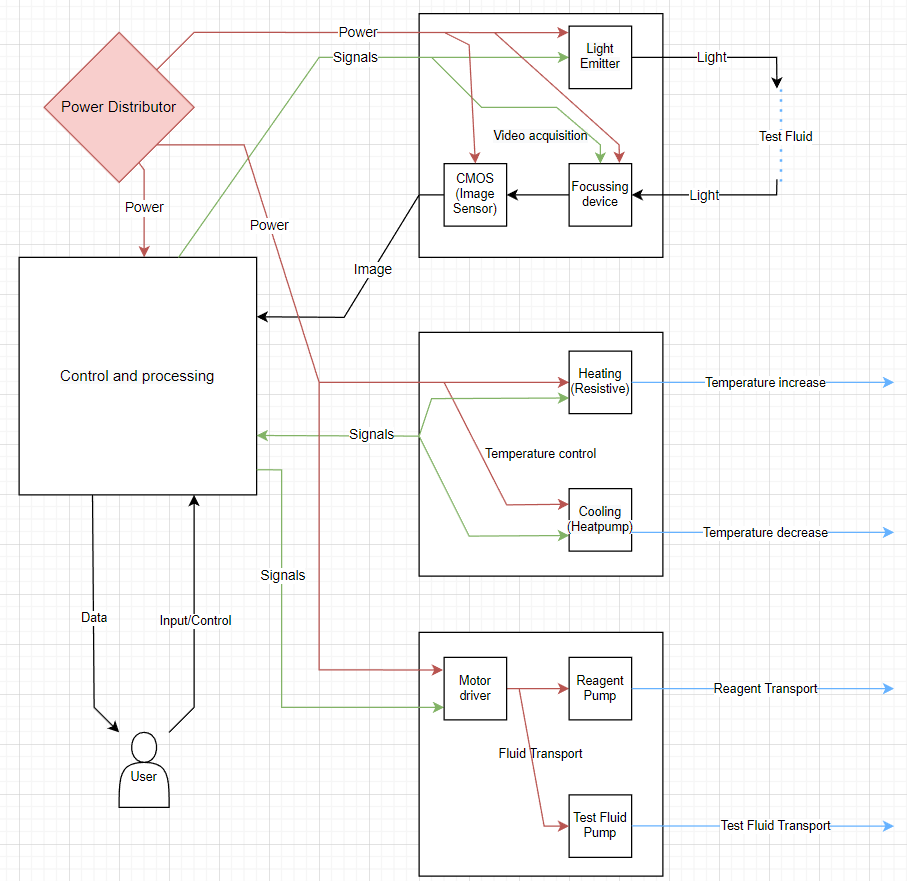
# Statemachine



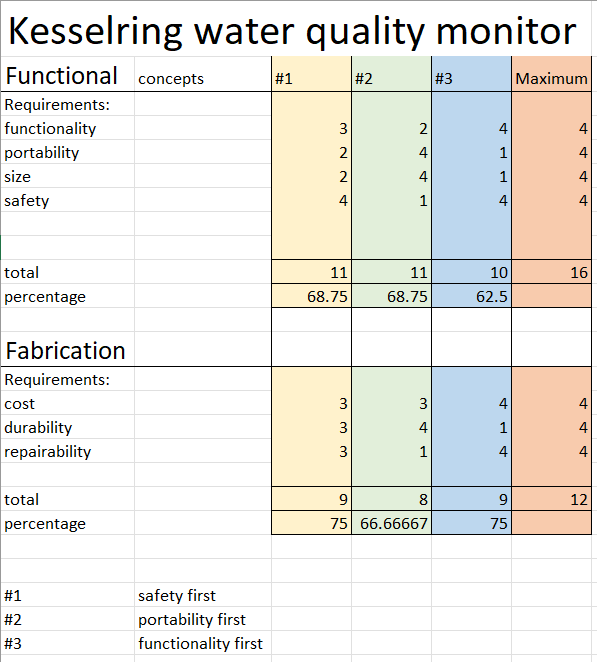
State diagram

Use Case Diagram

System Architecture



Kesselring



# KiCad

## Why KiCad6?

Afbeelding met tekst

Automatisch gegenereerde beschrijving

The Rastaban project requires a printed circuit board (PCB) to be used and moved safely from one place to another. Additionally, a PCB is the best option for producing the product in larger quantities. I chose KICAD 6 for several reasons.

First, it is open-source and free, making it accessible to many hobbyists and professionals. Second, it is a good software to learn how to design PCBs. Previously, I used EasyEda, which is proprietary. However, this comes with ads, limited access to certain features, and increased vulnerability to losing or stealing designs. In addition, I was unable to install useful plugins like an interactive BOM file, 3D model archiver, and fabrication toolkits. These limitations led me to stop using EasyEda. However, it should be noted that using EasyEda's library may be the fastest and cheapest way to produce a fully assembled PCB. If you plan to assemble your PCBs yourself, this is no longer an issue. KICAD 6 also has plugins that allow you to easily obtain LCSC part numbers for PCB assembly with JLCPCB.

## Plugins

To create an even better PCB and make my workflow easier, I downloaded a few plugins. The plugins I downloaded are as follows:

|  |  |  |
| --- | --- | --- |
| Name | Function | Download location |
| KiCAD JLCPCB tools | This plugin allows you to search the JLCPCB parts database, assign LCSC article numbers to your parts, generate production files for JLCPCB and much more. | <https://github.com/Bouni/kicad-jlcpcb-tools> |
| Interactive HTML BOM | This plugin generates convenient BOM listing with ability to visually correlate and easily search for components and their placements on the pcb. | Build into KICAD6 |
| PCB action tools | Annular Ring Checker, Snap Selected Footprint(s) to Grid, Fabrication Footprint Position, Move Selected Drawings to chosen Layer, Export pcb technical layers to DXF, Checking 3D missing models | Build into KICAD6 |
| Archive 3D models | Copies footprint models to the project local subfolder and remaps all the links within the used footprints. | Build into KICAD6 |
| Place Footprints | Arrange sequentially numbered footprints or footprints from multiple hierarchical sheets in linear, circular or matrix arrangement. This plugin works on footprints already present in the layout, so that layout and schematics stay in sync. | Build into KICAD6 |
| Round Tracks | Algorithmically smooth tracks in a predictable manner. Useful for flex PCBs, or just because it looks cool. | Build into KICAD6 |
| Length matching | Track Length Calculator | Build into KICAD6 |
| Freerouting | Auto router for Kicad. It draws all the connections between components for you. Be warned: Auto routing should never be used carelessly, always check the results. | Build into KICAD6 (requires Java) |

## How to learn KICAD6

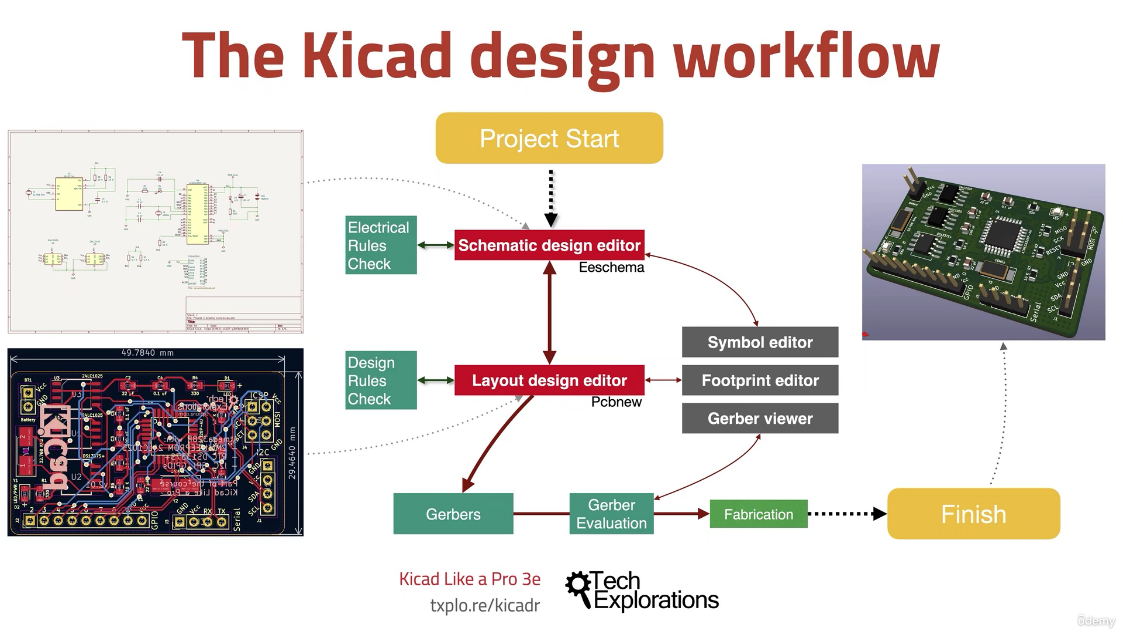
Afbeelding met tekst

Automatisch gegenereerde beschrijving

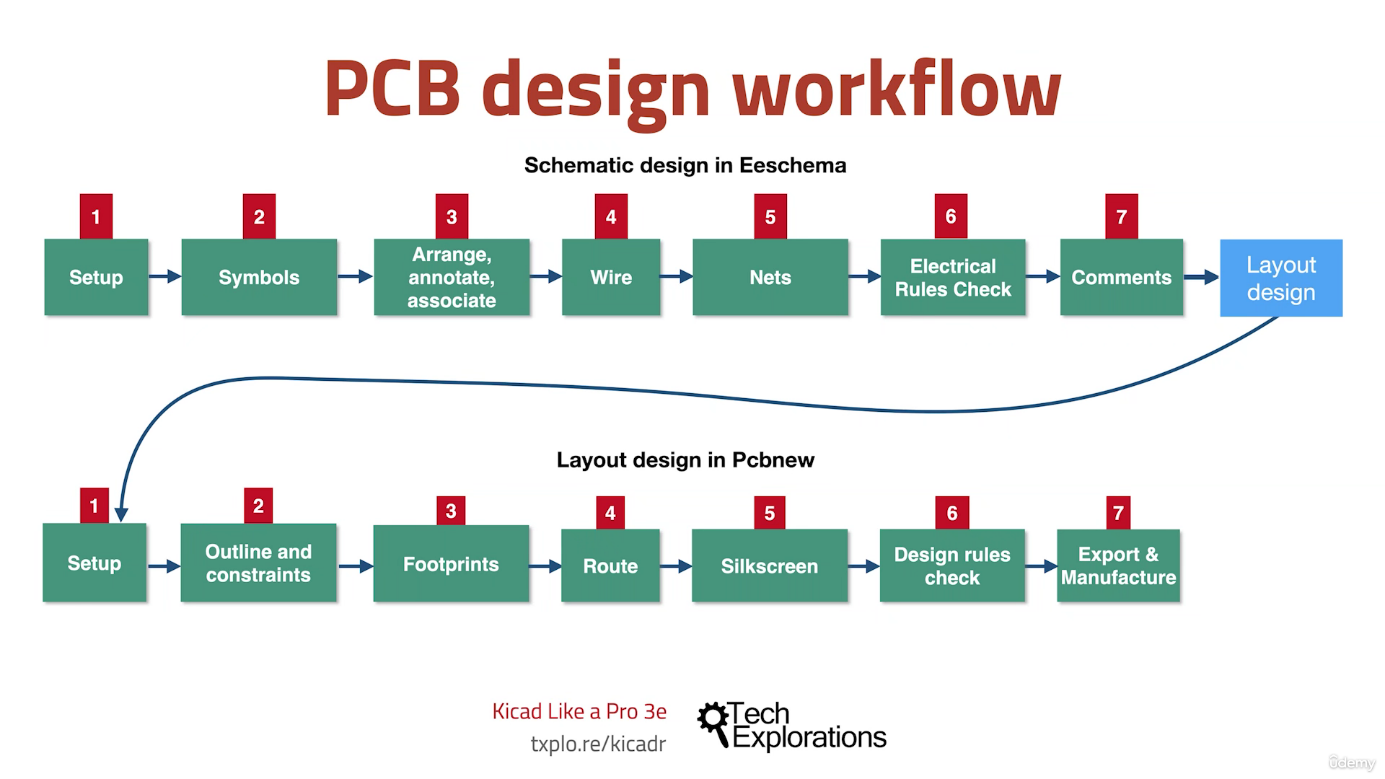
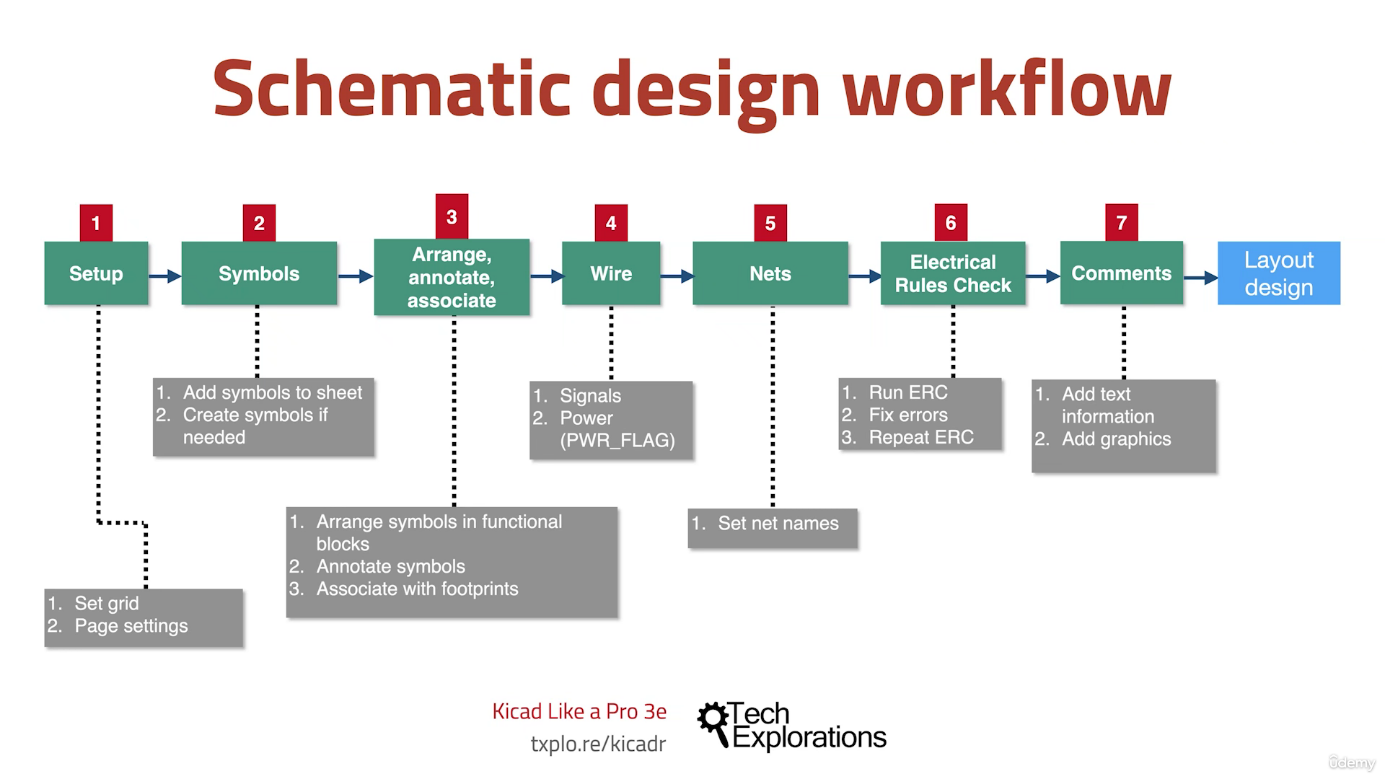
I learned KICAD6 by following [an online course.](https://techexplorations.com/so/kicad-like-a-pro-3rd-edition/) But I could also have used tutorials on YouTube or read books on it. KICAD itself is not difficult to use, but there are a lot of buttons, some of which are important, some are a bit redundant, which can be confusing. The best way to learn KICAD is to just start creating a schematic with some (maybe 4) parts and connecting them together. Then you can start creating a PCB in the “PCBNEW” section.

## The design approach

Creating a PCB is always following trough 2 stages (as shown in the picture right)

1. Create the schematic
2. Create the PCB

After having done this, your design is finished. Of course there are more details to both design stages an those are described in the following pictures.



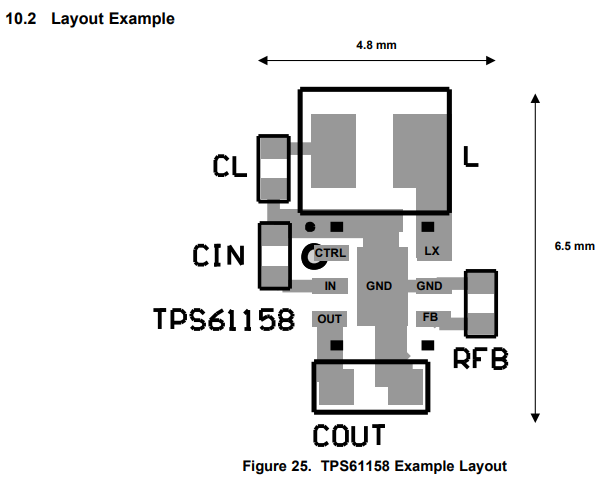
# System Components

## Microscope led

****

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.

It is important to follow the pcb layout recommendations to avoid unsuspected behaviour due to unwanted resistance or capacitances.



### Testing plan

When making a video or taking a picture of the reaction chamber, it is important that the LED light has a switching frequency that is higher than the shutter speed of the camera. If this is not the case, it could be that the image will appear to be darker or completely dark and that flickering will occur while recording a video. Therefor we need to know all shutter speeds of the camera and then we need to find what frequencies we need to drive the LED with, especially when dimming the light.

Our expected shutter speed: 60

Minimal shutter speed: 300usecs (<https://forums.raspberrypi.com/viewtopic.php?t=323983>)

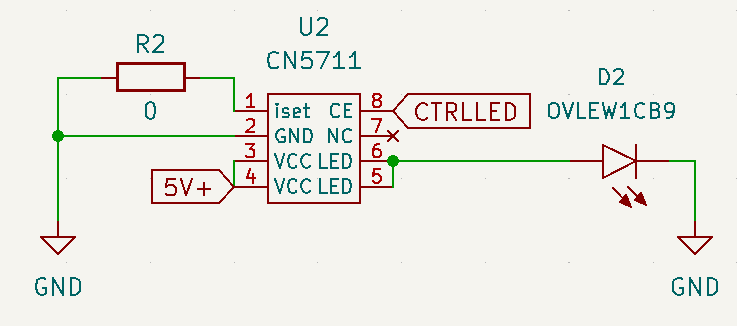
Python control camera: <https://raspberrypi.stackexchange.com/questions/99304/shutter-speed-and-exposure-time-of-picamera>

About shutter speed and fps: <https://camerajabber.com/what-shutter-speed-for-filming-moving-subjects/#:~:text=The%20180%2Ddegree%20Shutter%20Rule%20states%20that%20whatever%20the%20framerate,What%20is%20this%3F&text=Therefore%20if%20you%27re%20shooting,speed%20should%20be%201%2F120th>.

## Driver circuit decision table

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nr.** | **Functionality** | **1** | **2** | **3** | **4** | **5** | **6** | **Points** | **Weightfactor** |
| **1** | **Dim the led without visible flickering** | - | 1 | 1 | 1 | 1 | 1 | 5 | 5 |
| **2** | **Provide enough power to fully brighten LED** | 0 | - | 0 | 1 | 1 | 1 | 3 | 3 |
| **3** | **Enough brightness levels** | 0 | 1 | - | 1 | 1 | 0 | 3 | 3 |
| **4** | **PWM and digital control** | 0 | 0 | 0 | - | 0 | 1 | 1 | 1 |
| **5** | **PWM only control** | 0 | 0 | 0 | 1 | - | 1 | 2 | 2 |
| **6** | **Highest energy efficiency as possible** | 0 | 0 | 0 | 0 | 0 | - | 0 | 0 |

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.



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.

## Stepper motor driver

2209 and 2208 have the same pinout, this should mean that they are interchangeable when it comes to the footprint on the pcb.

**Afbeelding met tekst

Automatisch gegenereerde beschrijving**

For layout considerations **check chapter 19** of the tmc2208 datasheet.

### Later findings

The TMC2209 is the definitive replacement of the TMC2208 and TMC2130, so there is no good reason to keep using the depreciated TMC2208 in further designs.

### Micro stepping

Normally a stepper motor has 200 steps per complete revolution. Micro stepping can give you up to 51200 steps per complete revolution (depents on the motor of course) or 1/256 steps.

Benefits: improves noise levels and motor smoothness/accuracy and could improve energy effiency.

Drawbacks: decreases torque (especially at higher speeds) which could lead to stalling (some drivers can change stepping modes accordingly to speed).

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

Afbeelding met tafel

Automatisch gegenereerde beschrijving

### UART on the TMC2209

Using the UART function of the TMC2209 we gain control many new functions including:

* Access to the control register
* Programmable 256 uStep sequencer
* Pulse generator
* SpreadCycle
* StealthChop
* Access to OTP

It is be possible to remove the STEP and DIR connections if we use UART to control one motor.

### UART Problem with multiple drivers

When connection multiple drivers you are able to read/write commands to the drivers individually using the same bidirectional line:

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Strangely enough I not figure out why the above configuration did not work.

I configured one driver to be 01 and the other 00 by connection the ms1 and ms2 to GND and VCC the way it was shown in figure 4.1 above. Somehow I always get this message: Afbeelding met tafel

Automatisch gegenereerde beschrijving

These are the configurations I tested:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | PDN UART | TMC software address settings | Result | Additional notes |
| TMC driver A | Connected to RX | 1 | Functional | Changing one address to 0 also makes it NOT functional |
| TMC driver B | Floating | 1 |  |  |
|  |  |  |  |  |
| TMC driver A | Floating | 0 | Functional |  |
| TMC driver B | Connected to RX | 0 |  |  |
|  |  |  |  |  |
| TMC driver A | Connected to RX | 0 | NOT Functional |  |
| TMC driver B | Floating | 1 |  |  |
|  |  |  |  |  |
| TMC driver A | Floating | 0 | NOT Functional |  |
| TMC driver B | Connected to RX | 1 |  |  |
|  |  |  |  |  |

Conclusion: The driver UART control only works if there is one driver connected to RX and assigned address to this driver is selected in software (according to the ms1 + ms2 settings you chose). You can however use 2 drivers if you go with the “more than 4 drivers solution” which Trinamic provided. This could be achieved with transistors I suppose.



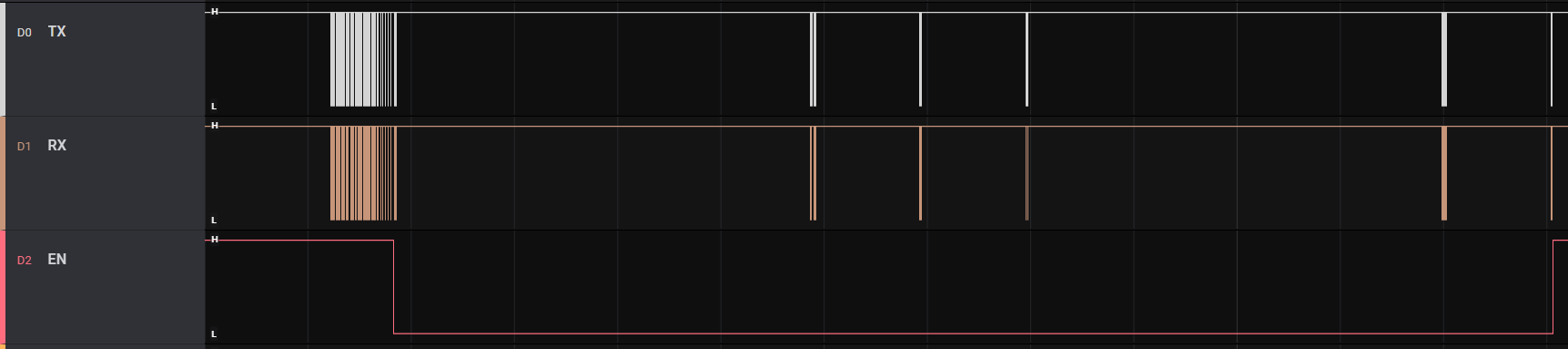
If reading from the driver is not required, using the write only function as seen in figure 4.1 of the datasheet is a viable option. This way we only need to use n+1 pins of the raspberry pi to control the motors, where n increases with every driver added (enable pin). This would treat all steppers as the same and would control them using the enable pin and sending data over the RX pin.

Configuration (test script 05: vactual.py) **RX and TX are flipped** **on the picture:**

Afbeelding met tekst

Automatisch gegenereerde beschrijving

This is what a successful transmission, with a few rotations and direction changes should look like.



When zoomed in it is clear that the TX deviates from the RX line data. Afbeelding met tekst, computer

Automatisch gegenereerde beschrijving

Further inspection shows that it does sometimes match the RX line

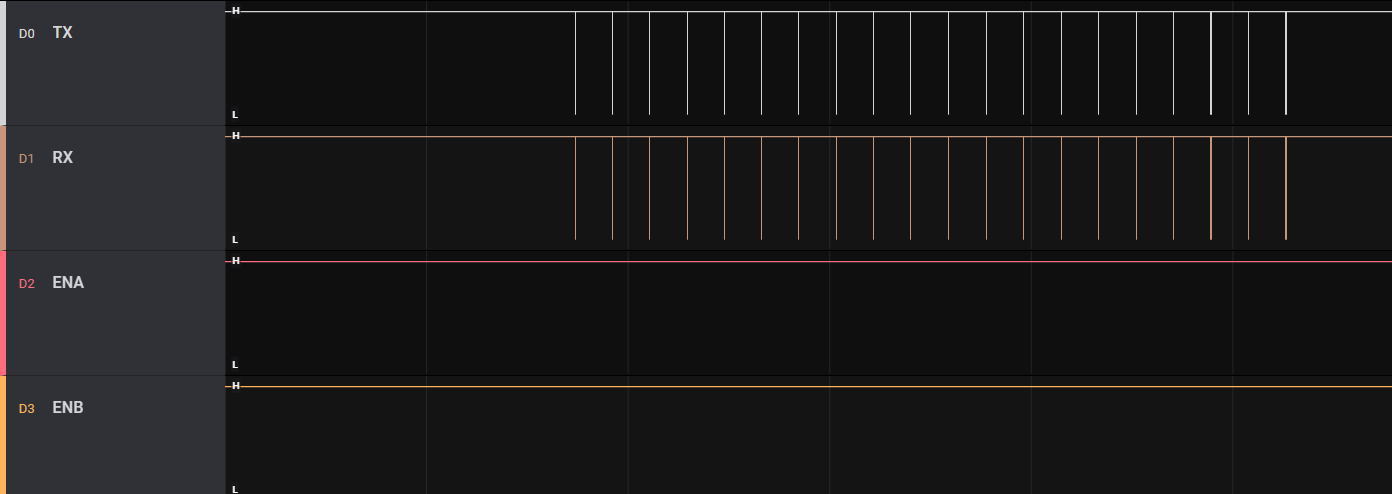
It is however always the case that TX continues sending after RX is done with a package.

It is however always the case that TX continues sending after RX is done with a package.

Using these settings (test script 06: multiple drivers.py) **RX and TX are flipped** **on the picture:**

Afbeelding met tekst

Automatisch gegenereerde beschrijving

This is what a unsuccessful transmission looks like: 

When zoomed in it seems that TX receives exactly the same information as it sends out. While, as stated earlier, it is expected that TX sends more data after RX is finished.Afbeelding met tekst

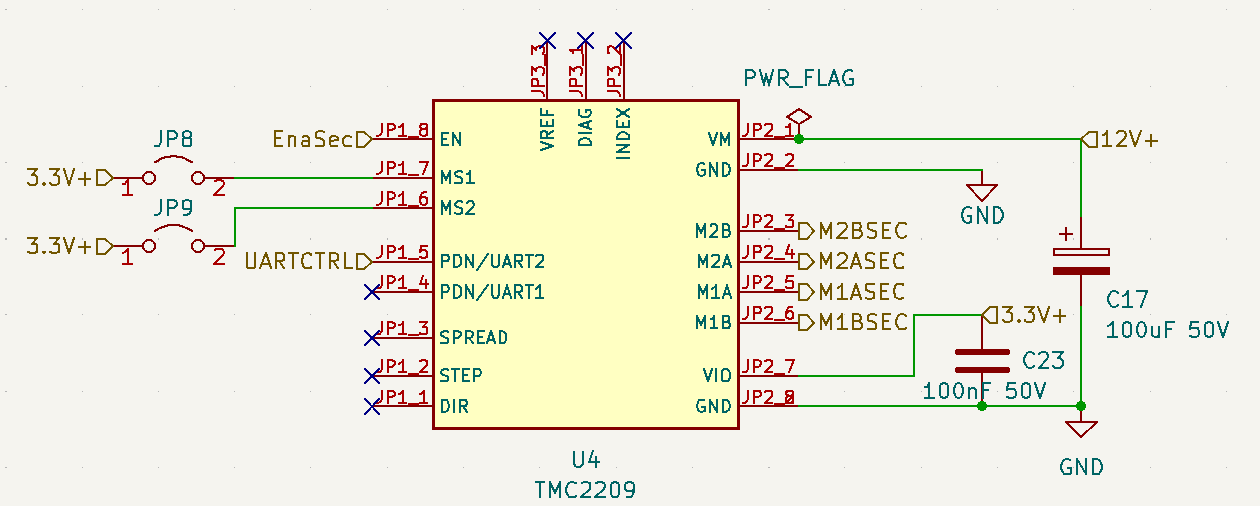
Automatisch gegenereerde beschrijving

### Conclusion multiple drivers UART error:

I can conclude that somehow TX is getting the same information as it sends out on the RX line.

Fix found by Thomas Ijsseldijk: lowering the TX resistor to 500ohms increases the voltage available for the IO of the TMC driver which results in the TMC being able to read the signal. It appears to be very important to choose the right resistor value. It is now possible to control the stepper motors via UART. This reduces the pins required on the raspberry pi with 7 pins (removing all step and dir pins)

The final configuration for the stepper motor drivers is as following:



We decided to go with the TMC2209 and we operate it in UART mode only. This means we need to connect to PDN UART and use one enable pin per driver to select it. 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.

### The 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.

### EMC

EMC or electromagnetic compatibility is a significant part of every PCB design. We don’t want the high currents that flow through these components to disrupt the signals lines. Therefore I need to consider the position of the power wires.

## PowerManagementSystem

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

### Power Supply

This table is just an indication of the possible power consumption 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 whole device at full power, something that will not be typical but should be accounted for. 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 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 5.2V 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.

### EMC

EMC or electromagnetic compatibility is a significant part of every PCB design. This starts at the schematic level by making sure there are enough decoupling capacitors, 0 ohm star grounding and keeping HIGH power LOW power, digital and analog circuits as isolated as possible.

### ESD protection

A typical voltage clamping diode circuit is shown below. The main responsibility of this voltage clamping circuit is to limit the accumulation of voltages on the input terminal of the buffer. Note that this could also be applied to the differential input on an op-amp. The operation of this circuit is very simple and, under normal conditions, diodes D1 and D2 are reverse biased. Whenever the voltage at the input is larger than the supply rail voltage, then diode D1 is forward biased and conducts. Similarly, when the voltage at the input falls below ground, then diode D2 is forward biased and conducts from the ground towards the input. This circuit seems to be suitable for at least the i2c data lines in our design.



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

### Power regulators

There a 2 main types of upping or lowering the voltage in a circuit. We can use Buck/Boost converters or linear regulators. The main benefit of using a buck/boost converter is its high efficiency and therefor lower heat dissipation. Sadly it comes at the price of many components (PCB space) and a noise on the supply line due to switching. A linear regulator on the other hand wastes an extreme amount of the energy supplied to it as heat, but the output voltage does not create as much noise. A great bonus is that if the voltage difference is not great between input and output, than sometimes the linear regulator is more efficient than a switching.

Choosing what type fits our voltage rails really depends on what is more important in our device’s case. We care more about functionality and repairability at this stage of the device so using a linear regulator would most likely be sufficient in most of our cases.

### 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.

### Trace width

Afbeelding met tafel

Automatisch gegenereerde beschrijving

### Coil whine

## Pi Hat i2c EEPROM interface

## 12V POWER GPIO

## Focus motors