Internship

**Personal Reflection internship**

**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

# The internship

In my internship I have come in contact with many new experiences. I had to design hardware, software and had to carry this out alone, although I received guidance of Jeroen Veen. I also got introduced to new programming languages: Python and PYQT. This is a write down of the subjects that I learned or got introduced to:

* Python programming language
* PYQT programming language
* Draw.io diagram/drawing modelling software
* Autodesk Fusion 360
* Linux (Raspbian)
* PCB design in Kicad6
* 3D printing with PLA
* 3D printing with SLA
* Food grade silicon production
* General fabrication materials
* Educating students about electronics and the design process
* Trello
* Github desktop
* Visual Studio Code

Not all of these subjects are explored as extensive as others.

# Overall reflection on my process

## Project setup phase

In the first weeks of my internship I discovered that I was really focused on doing everything in the design process as it was shown in the V-model. I think that in the beginning of a project this is the best way to start. However I do think that planning or creating a very detailed plan of approach is a waste of time mostly. I noticed that I knew not enough about the subject to create a real plan of approach so next time I will start by doing a lot of research on the subject. And creating a Trello planning board accordingly. All gathered information can written down and with that information I plan to create my first functional specifications. This seems to me, the best approach for the setup phase next project.

## Project general phase

### Selecting components process

After finishing the functional and some of the technical design, I had to do a lot of research on components, chips and functionality in general. I noticed that I didn’t have my goals and my specifications clear enough to select the right chips, drivers or other components. This resulted in choosing many chips with features I ought to be useful one day whilst they may have been overspecced heavily for the applications, even for future plans. This resulted in not being able to make choices and wasting time. I selected a chip, added complimentary components and figured out later that it was not a suitable chip to begin with, because I forgot X or Y, which was an specification required by the client.

Next time I will make sure that my technical specifications matchup with my functional specifications and I will ask my client to review the components I chose before implementing them into a design.

### Testing on breadboard

I think that when I was creating my first PCB for Rastaban, I should have spend time testing things first on breadboard. This way I would have found a lot of problems that could have been fixed before ordering a PCB. Now the first PCB had many, many mistakes on it. Some which where unfortunately not really easy to spot and some mistakes could have been easily prevented if I had just tested it once on breadboard. Of course I was limited by the time to chip components and some components are not available in breadboard friendly packages (although there are workarounds) and testing them are time consuming.

It was not wrong from me to choose to order a PCB so soon without testing everything, it is not always wise to spend the time testing everything, since setting up tests can take much time too. But testing the bare minimum would have been smart. Don’t implement something that is complex and may behave differently from how you think it might.

I can conclude that next time I will be testing all components that I can test easily on breadboard, will be tested and the ones that are not easily tested must be researched heavily before implementation.

### Documentation

While working on my project I had to write documents of course. These documents should contain everything that reflects my design choices and a summary of my work.

I was very motivated this project to write good documentation on the design process and tests. I followed the V-model and I think that I have made good documents. I could however improve on the way I write down my specifications/requirements and I could work on my diagrams. I think that the tables that I am using for showing my requirements are a bit to simple, not detailed enough and difficult to read. I could improve it maybe by looking up a standard made by others. I do like the colouring I used for the tables. The diagrams should maybe be reviewed more often. Some of the diagrams may be created later in the project process, since loads of things get altered half way in the project.

### Learning goals

During my internship, I planned 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.

#### Kicad6 and PCB workflow skills

I think that I succeeded in improving my kicad6 skills. I have designed 3 prototype PCBs and I think that I learned especially when to start designing and when to make improvements and how. I mean to say that it is not always a smart move to create a redesign, especially minor revisions, right away. It could be time consuming to reroute every wire on a pcb if some connections change. Next time I will be testing a PCB and note every improvement I see. After having tested the PCB extensively I will start a revision. I also learned implementing new footprints and librarys using SnapEda or Library Loader. The latter is really easy to use with the Mouser website and it’s components.

Another thing I learned is to always check the footprints that you download from third partys. Some are fine, others are simply wrong. This can lead to non functional prototypes, long hours of debugging and damaged components. The last thing I would like to add is the importance of 3D models. I noticed that using 3D models in your design can potentially save you from real world problems. Especially connectors can be a real problem if you don’t remember to position them correctly. Using 3D models also has another benefit: you can export your design as one .STEP file and import it to a program like Solidworks or Autodesk Fusion 360. You can easily design things like a casing around the PCB this way. Something that especially in multidisciplinary design teams like ours is a big win in time.

#### Programming skills

I learned new programming languages during my internship: Python and PYQT. I think I learned the most of object oriented programming with Python. I learned the way to think will making classes. How functions give features to classes. The language itself really speaks to me. I feel more comfortable at this moment using Python than C++. I think this language gives me more confidence in programming and I expect that it will help me understand C++ and other languages better as well.

Designing software was a challenge for me. The goal was to create software that was easy to adapt to new hardware, pin configurations and make the software easy to understand and maintain.

I think that I learned much from the object orientated programming way I used with my Python programs. Next time I would like to implement the code more into a whole. I now left my code like many small scripts that all do their function, which is fine, but it feels like I created only testing scripts and not a fully functional product.

#### Other subjects I learned

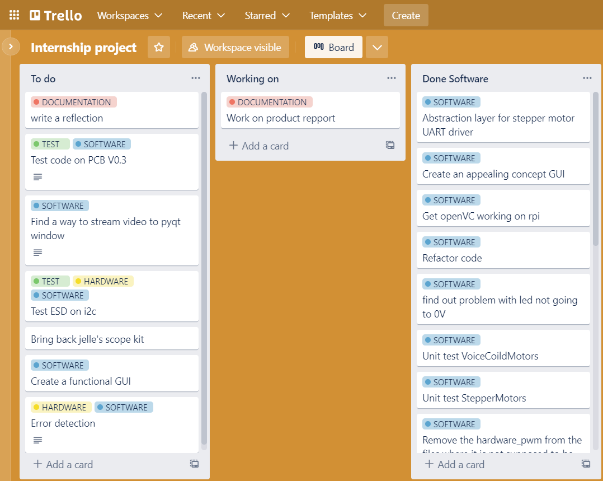
Like I said in the beginning, I learned many things and to many to describe in this reflection. I want to talk about these bullet points:

* Educating students about electronics and the design process
* Autodesk Fusion 360
* Trello
* Github desktop

On request from Rudie, but also my own preference I decided to help with the UCD2 lesson for the second year industrial product design students. This included helping students with their work by listening to their questions, forming a general idea of the problem, asking for validation on my conclusion and then generating an answer that is adjusted to the skill and understanding of the students. It was important that I wouldn’t take over all the work of the students. They had to learn how to program and troubleshooting for themselves, but I did assist where needed or where I thought trouble shooting was beyond the scope of the learning goals.

Overall I would say that it was a useful learning experience and I think that I could see myself doing work as a teacher later when I’m a seasoned electrical engineer or industrial project designer. For now it is to soon, I want to explore my designing skills.

I used Autodesk Fusion 360 to model a chocolate bar to gain more experience on 3D design, materials like Wood, PLA, SLA and Silicon. I learned some basic functions from Fusion 360, learned how to use fabrication machines and how to improve on my designs without spending much time and money on useless revisions. I also kept material usages and spillage in high consideration.

Trello is a tool that I used to create a general overview of what I had to do, what I was doing (I really do forget that) and what I already have done. I like that I can move the blocks from row to row and back. This way I can easily see what I may need to do again for example. Using the labels functions for software, hardware, documentation and tests (or other custom labels) gave me a quick overview of what I have been doing the most or what needs to be done in a specific category. I will absolutely use this tool again and it is really useful in a team settings as well.

I used Github desktop to improve my version management. I found using Github desktop really easy to use and I almost never had any problems with it. I only experienced one problem with Github and it was solved relatively quickly. A benefit of using Github desktop is that I can download all the files to the raspberry pi with a few command lines.

I did notice that is it faster to use SSH in visual studio code to run python scripts, so for coding I still prefer that, but for moving files or big chunks of new code, using Github is great.

### Writing things down

I learned to write things down in the middle of a conversation. This may sounds stupid but it really changed things up for me. By writing conversations down I am able to remember what we talked about and I don’t have to worry about any details getting lost or forgetting that the conversation happened as a whole. I will write more conversations down in the future and possibly record conversations if that is agreed upon.

## Troubleshooting

When troubleshooting the TMC2209 driver UART connection I was debugging for hours and was convinced that the problem lay in the hardware of my PCB. I was so convinced because I thought that I had seen the UART connection working with two boards on a breadboard of mine. I was so convinced that it has worked once, that I completely ruled out software or settings on the RPI as a problem. Of course my parameters could have been changed, but I just redownloaded the code I worked with the first time and everything “worked” somewhat. I think that I can conclude that seeing something “work” doesn’t mean that it is working the way I think it is working and ruling things out that way is time consuming.

## Ending of the project