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:** 16-01-2023

# 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.

### Software

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.