Universitetet i Sørøst-Norge

Bacheloroppgave

SFHO3201 Karoline

Predefinert informasjon

Startdato:	05-05-2019 10:00	**Termin:**	2019 VÅR
Sluttdato:	23-05-2019 10:00	**Vurderingsform:**	Norsk 6-trinns skala (A-F)
Eksamensform:	Bacheloroppgave		
SIS-kode:	222 SFHO3201 1 BO 2019 VÅR		
Intern sensor:	Radmila Juric		
Intern sensor:	Karoline Moholth Mcclenaghan		

Deltaker

Navn:	Marco Tande
Kandidatnr.:	55
USN-id:	888658@usn.no

Gruppe

Gruppenavn:	Gruppe 2 Argos
Gruppenummer:	10
Andre medlemmer i gruppen:	Jan Olav Møen, Emil Wegger Hultin, Vegar Woldstad Skogen, Vetle Mørch Tørjesen, Kenneth Boje Halvorsen

Sensur av hovedoppgaver

Universitetet i Sørøst-Norge
Fakultet for teknologi og maritime fag

Prosjektnummer: **2019-02**
For studieåret: **2018/2019**
Emnekode: **SFHO3201-1 18H Bacheloroppgave**

Prosjektnavn
Argos 3

Utført i samarbeid med Kongsberg Defence & Aerospace.

Ekstern veileder: Alexander Gosling, André Ruud, Håvard Omholt

Sammendrag: Fortsettelse av tidligere bacheloroppgaver, Argos 2016, Argos 2.0 2018 og sommerprosjekter siden 2015. Argos har målet å kunne øke situasjonsforståelsen til soldater i pansrede kjøretøy, ved hjelp av 360 graders utsikt med lav-forsinkelses kameraer. Våre hovedoppgaver er:

- Implementasjon av kommunikasjon med Virtual Battlespace.
- Utvikle ny mekanisk løsning for økt brukervennlighet.
- Utvikle et nytt elektrisk system for å løse tidligere problemer med ustabil strøm og støy.

Stikkord:

- Dimished Reality
- Thermoelectric Cooling
- Modular Lightweight Design

Tilgjengelig: DELVIS. Kildekode er ikke offentlig tilgjengelig.

Prosjekt deltagere og karakter:

Navn	Karakter
Vegar Skogen	
Jan Olav Møen	
Kenneth Halvorsen	
Marco Tande	
Emil Hultin	
Vetle Mørch Tørjesen	

Dato: 14. juni 2019

________________	________________	________________
Radmila Juric	Karoline Moholth	Alexander Gosling
Intern Veileder	Intern Sensor	Ekstern Sensor

ARGOS

Documentation

Vegar Skogen, *Software Engineer*
Jan Olav Møen, *Software Engineer*
Kenneth Halvorsen, *Mechanical Engineer*
Marco Tande, *Mechanical Engineer*
Emil Hultin, *Electrical Engineer*
Vetle M. Tørjesen, *Electrical Engineer*

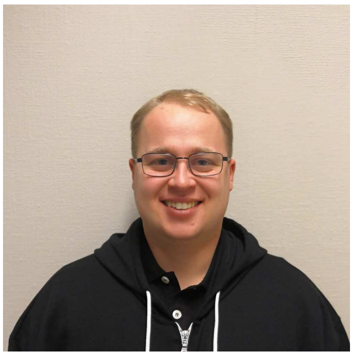
May 23, 2019

Bachelor Project 2019
Group Nr. 2
University of South-Eastern Norway
Faculty of Technology, Natural Sciences and Maritime Sciences
Project owner:
Kongsberg Defence and Aerospace

i Group Overview

Vegar Skogen
v.w.skogen@gmail.com
948 94 449

Project leader
Lead Software Developer



Jan Olav Møen
Jomoen96@gmail.com
907 56 703

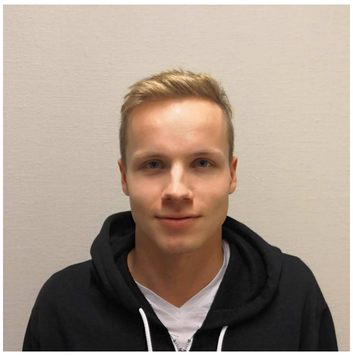
Responsible for UML and testing

Kenneth Halvorsen
Kenneth.b.hal@gmail.com
986 60 933

Responsible for material and simulation

Marco Tande
marcotande@hotmail.com
938 64 641

Responsible for Solid Works and prototyping



Emil Hultin
Emil@hultin.com
954 69 797

Responsible for electronics production

Vetle Mørch Tørjesen
Vetle24@gmail.com
903 68 592

Responsible for control engineering

ii Chapter Description

Background:	The background in this document describes in short what this bachelor revolves around and some essential information about the legacy system that the whole project began with.
Problems To Solve:	The problems to solve in this document contains the main problems of this project and what will be changed to some extent. It also goes more in depth on how the system works over all and how it holds together. Each main problem is divided into different sections so that one can receive a greater understanding of the bigger picture.
Project Methodology:	The project methodology describes our methodology and how our group being multidisciplinary affected our choice of method. Within this chapter is also the pillars of our project, namely requirements, testing and risk.
Modeling and Designing the System:	Modeling and designing the system is how we intend to solve the problem and design the new system.
Implementation and Production:	The process of designing and developing the final product. Changes to the product during the project. In addition to how you will produce the product.
Challenges:	The challenges we have met throughout the project.
Argos Final Product:	How the final product functions and interact within its boundaries.
Conclusion:	The conclusion summarises how the project went and concludes with our thoughts and experiences in hindsight.

Contents

1 Background

Project Argos is a see through armour system that began development in 2015 by Kongsberg Defence and Aerospace (KDA). The project is mostly developed by several summer interns and bachelor projects, with the help of exceptional guidance from in-house engineers and university supervisors. With the help of high resolution cameras with low delay, an operator can sit within a vehicle and view the surroundings trough Virtual Reality (VR) goggles. This way, the operator is still protected from the environment and has no need to expose them selves to unnecessary danger.

The set-up that we inherited at the beginning consist of four Mako cameras. These are linked to a network interface card(NIC) within the computer, so that the Argos system can interpret this. To contain and protect the cameras during operation we have two boxes. These boxes are mostly build out of acrylic and metal, which are the attached to a wooden plate so it can be mounted on the roof of a car. We also have a power supply that runs on batteries to provide power to the rest of the system when it is used in the field. A part that provides directional information is the Inertial Measurement Unit (IMU). This sensor provides the possibility for the person operating the system within a vehicle, to follow the cars movement when turning without the view drifting. If this sensor would not be in the system and the car were to turn, the operator would still view the same direction as the operator viewed before the car turned.

1.1 Scenario of Demonstration

Project Argos is a system designed to improve the situational awareness of soldiers when in a armored fighting vehicle (AFV). One possible scenario is for an AFV driver, that while driving, information comes in from other allied units, that a group of enemy units has been spotted in the direction the AFV is headed. Based on this information the Argos system can immediately display the location in the drivers heads up display (HUD), together with additional information that has been gathered.

Since Argos for now is only a Research and Development (R&D) project, and thus only really used as a proof of concept, a scenario that is closer to this bachelor project, is for a demo in a car, carried out by an engineer at KDA. The engineer is on his own able to set up the system with minimal effort, in any car he was able to reserve for the demo. During the demo the engineer is then able to check on the battery level between each person/group testing the system to ensure that the system does not run out of power while driving. Once the demo is over, the engineer is then able to dismount the system from the car to place the system components in storage, until the next demo.

2 Problems To Solve

Based on previous experiences made by both bachelor groups and summer interns, it was determined that a complete reinvention was needed for all hardware in the system. The previous hardware had been both designed and manufactured by only three electrical, and one mechanical engineering students over just two months. Already during the summer of 2016 comments were made regarding mistakes, since then the problems have only grown more obvious to anyone observing or working on the system.

Because of these experiences there was determined a list of things which needed to be redone in a more proper, well-planned manner. Because of this feedback, we have been given the job of effectively scrapping all old hardware, replacing it with new, more future proof designs. In addition to this, because Argos is meant to be able to show targets, like allies and enemies, but it can't actually be used in a real life situation as to collect this data. A feature was requested to be able to simulate a real scenario for Argos in a Battle Management System (BMS), to then be able to transmit this simulated data to the Argos system. This is so that the Argos system will be able to record the data, to manage later playback of the targets.

2.1 Legacy Mounting System

The current solution for the camera roof mounting, could be considered an extremely tedious process to mount on a car. This is done by using multiple screws and extension irons. They are held in place with bolts to a wooden plate as shown in Fig.2.1. This wooden plate is then connected to the pre existing roof rack of a car. Even though it is a tedious process to mount, the solution is highly maintainable, since all the products are available in hardware stores. We want to make it easier to mount and dismount the camera boxes to a car, but also make it more streamlined.

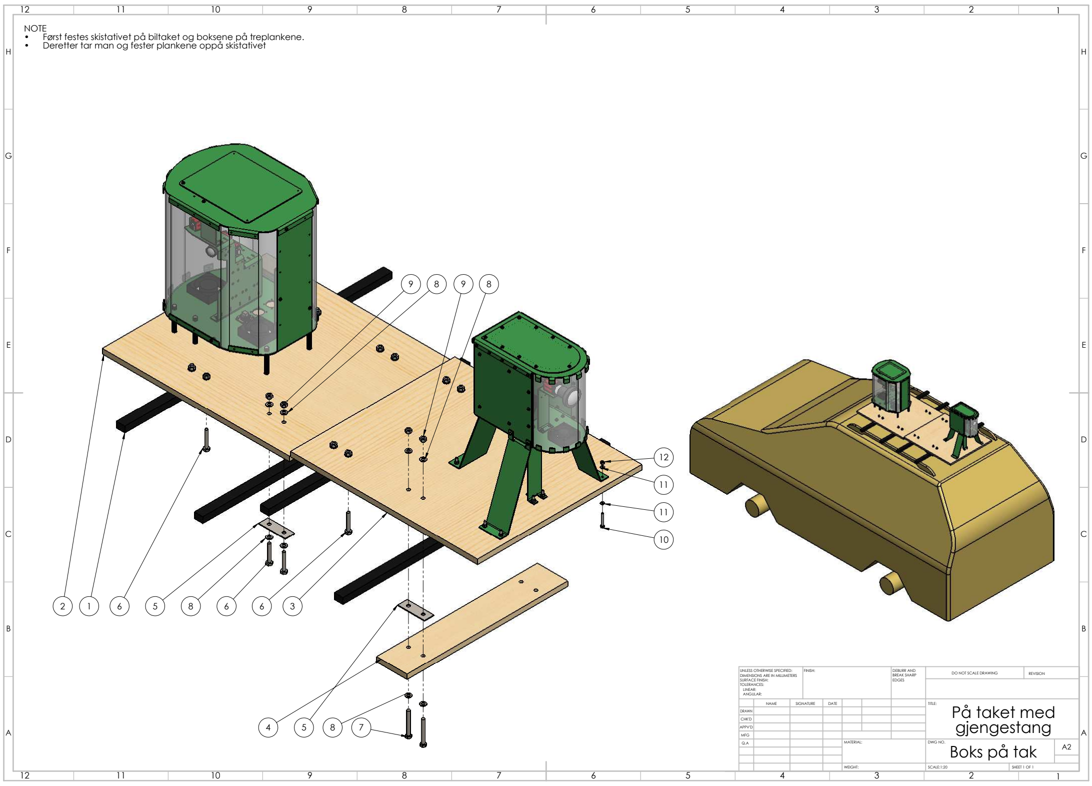


Figure 2.1: Argos 3 former mounting solution

2.2 Environment Protection For Cameras

In our project, we were asked to design a new solution to protect the cameras from the environment. Kongsberg Defence and Aerospace want a more streamlined and less time-consuming process in all aspects when interacting with the Argos system.

The system we inherited requires two people to mount. The boxes are mounted on large wooden frames, which make them hard to handle. The mounting solution also wastes time and could be vastly improved. In addition to this, the cameras don't have an auto adjustment to lighting. The light sensitivity adjustment needs to be manually adjusted on the camera lens. Getting access to the cameras in the current boxes is a whole ordeal. The covers have several screws as shown in Fig. 2.2 and are sealed with silicone sealant. This means getting access to the cameras is a slow and tedious process if you were to seal it when putting it together.

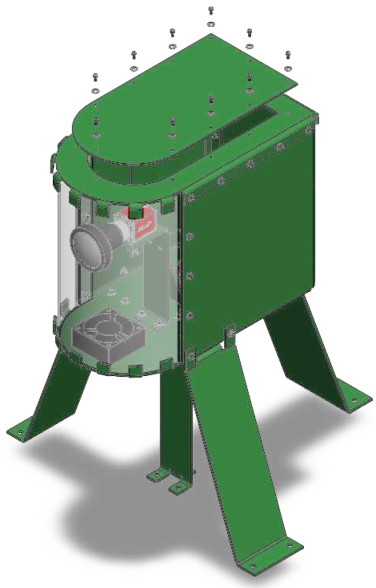


Figure 2.2: Argos legacy rear camera box

2.2.1 Field of View

We discovered when exploring design ideas and investigation that the cameras are not optimally placed. The field of view of the cameras has a dead zone as they're placed in the old boxes. The dead zone is small and unnoticeable in the image when using the Argos system, probably due to image rendering. The camera mounting position has to be taken into account when making the new boxes to fix this problem. There should be no dead zone between the cameras. We would rather have an overlapping image than a dead zone. Overlap can be fixed in software. In addition to this, the technical information from the manufacturer is not correct. The field of view does not match the stated values in the product data sheet.

2.3 Inertial Measurement Unit Vibration Attenuation

An inertial measurement unit (IMU) is a gyroscope, accelerometer and magnetometer combined into one. It measures specific force, angular rate and magnetic field. For our project, we use it for measuring our position while we are driving. In our project we're using three IMU bricks with a fourth master brick as seen in Fig. 2.3. The reason for stacking several of these bricks on to each other were an effort to receive more reliable values/readings. Vibration attenuation or vibration dampening is believed to be required for the IMU to be stabilized while in the field.

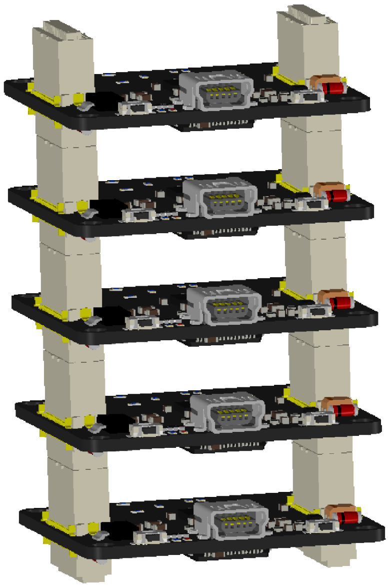


Figure 2.3: IMU Stack

The previous year the summer students had a problem where the IMU would lose track of where it was and would interrupt their driving. Their solution to this problem was rubber bands in a basket they used to stabilize the IMU see Fig. 2.4. While this solution solved the problem temporarily, we want to design a more permanent solution.

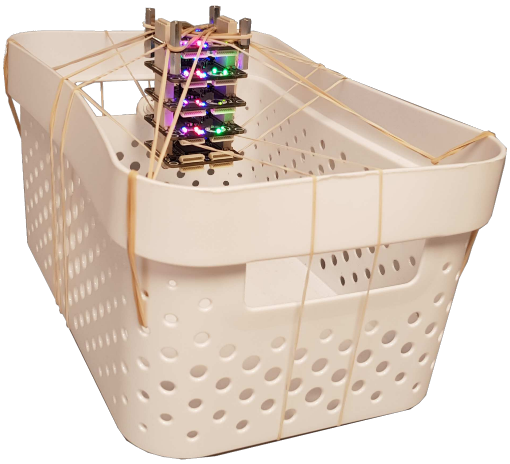


Figure 2.4: Former IMU Vibration Attenuation System

We've been doing some internal testing of the IMU using Brick Viewer- and VR software. In this testing we found out that the IMU loses track when we move it whenever we use Brick Viewer. When we're using VR in unreal engine the IMU is working flawlessly. We started looking through some of the documentation from last summer project and found out that one of the last things that was disabling the magnetometer in the IMU. While in the Brick Viewer we can't disable the magnetometer so we have to investigate to see if it's the magnetometer that makes the IMU lose track.

While researching on the internet we found that the most common way drones have done vibration attenuation [66], is using anti-vibrating rubber dampener [67]. Drones uses vibration attenuation for the flight controller, which is an IMU due to the sensitivity of the component.

2.4 Legacy Power System

When we overtook Argos, the computer that runs the Argos software was powered from a power inverter. Power inverters are used to invert the DC voltage of a battery into AC voltage that emulates the power grid. There are multiple ways of designing a power inverter, some much better than others [74]. The power inverter we inherited has what is called a modified sine wave output. Modified sine wave means that the output does not look like an ordinary sine wave, but is instead an approximation of a true sine wave.

We made measurements of the power inverter using a Keysight DSO1072B Digital Storage Oscilloscope. The output of the inverter is shown in Fig.2.5. By examining the output we see that the inverter is a modified sine wave inverter with only three steps. The higher the step count the higher quality the output wave is [75].

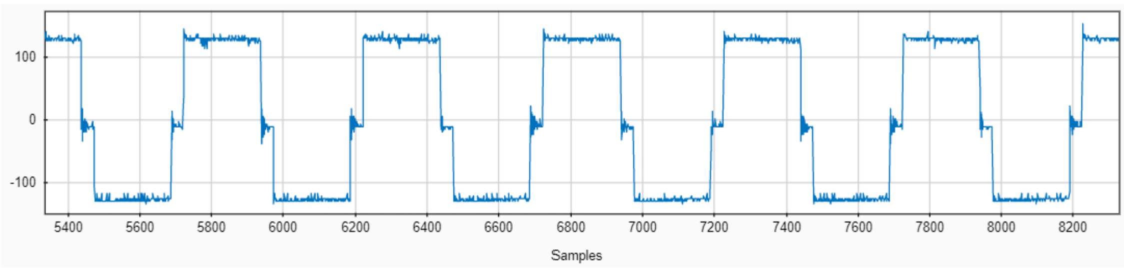


Figure 2.5: Inverter output measurement, $f_s = 90kHz$
(y-axis not to scale)

Fig.2.6 shows a typical signal from the power grid. In this figure, we can clearly see a distinct sine wave. There is some noise in this example, but the noise has much lower impact on the quality of the signal than the distortion of the modified sine wave [73].

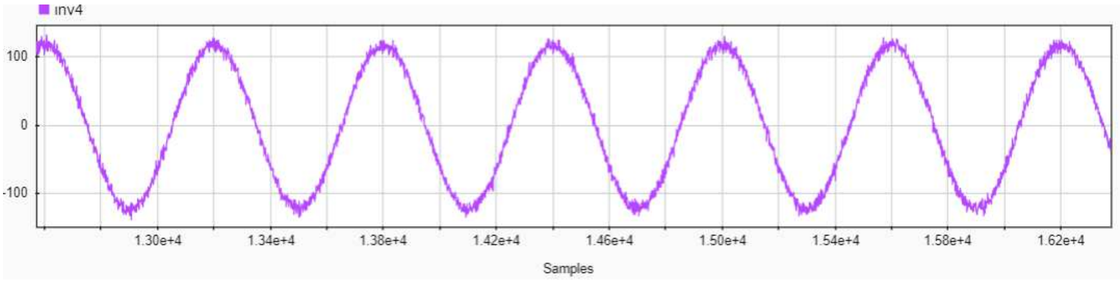


Figure 2.6: Typical power grid

The bottom figure shows the power spectrum for both of the signals. The power spectrum essentially shows where the power lies within each of the signals [76]. Ideally there should just be one slender peak all the way on the left side of the spectrum, like the power grid. But the measured signal from the inverter has many other peaks and valleys throughout the spectrum. These other peaks are called harmonics and are the cause of many problems.

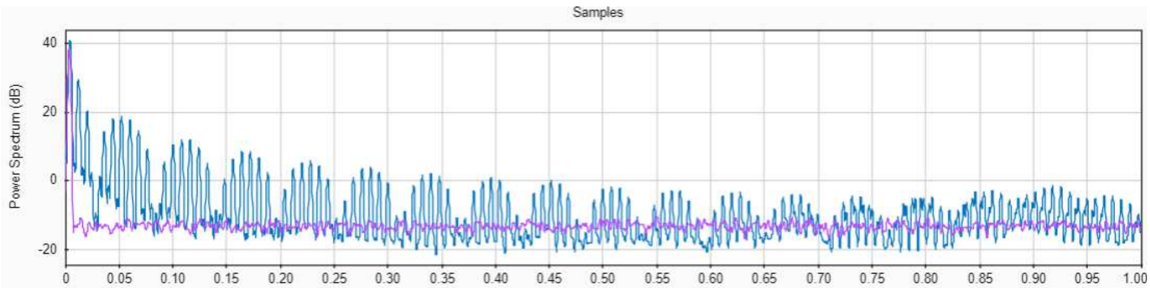


Figure 2.7: Inverter power spectrum
blue: Inverter, magenta: power grid

Total harmonic distortion (THD) is the unit of measurement used to define level of harmonics in voltage or current waveforms [68]. The standard definition of THD, known as THD_F, is defined as the ratio between the power of the harmonics and the fundamental frequencies.

$$THD_F = \frac{\sqrt{\sum_{n=2}^{\infty} I_n^2}}{I_1} \tag{2.1}$$

The inverter that we now use, is a three-step modified sine wave inverter which was added to the system in 2016. This provides a THD of as much as 28% at full load, which is a lot since the computers power converter does not have any contingencies for receiving anything different than a true sine wave and therefore can cause damage to the system within. Optimally we would want a THD less than 5%, since this is the IEEE-519 standard [69]. This can maybe be achieved by adding additional filters or change out the inverter with a true sine wave inverter with low THD.

We also plan to add functionality so that we can hot-swap the batteries when the charge starts running low. By doing this, theoretically at least, we can prolong the runtime of the system indefinitely.

2.5 Estimating The Battery State of Charge

We received a requirement from our employer that it should be possible to monitor the remaining charge of the batteries and view the information in virtual reality. We divided this into two sections, electrical and software. In this section we will focus on the electrical system that will send the necessary information to the computer.

Doing research on this, we found many battery monitoring circuits that were pre-made and mostly developed for boats or cars. Some problems with theses were the fact that these just displayed the values on the screen, while we needed the values electronically, since it is to be accessed in the virtual reality. Going forward we dived a bit deeper and found out about Coulomb counters, which is a circuit that is made for measuring electrons passing trough a certain point. A problem that were quickly discovered, was the fact that Coulomb counters aren't made for high power circuits like the one we are designing, considering the massive current of a hundred ampere. Because of this, we made the decision that the best way forward would be to construct our own.

Monitoring the state of charge can be very intricate, because there are very many factors that can create deviations from the estimated charge [82][83][84]. In addition to this, many of the parameters that are required, may only be measured when the battery is at without load. To understand why some parameters can only be measured when the battery is at rest, a good example would be voltage. A battery that is fully charged and put under heavy load, could indicate that the battery is almost out of charge, due to the voltage in the battery decreasing as the load increases. In addition to this, if we would only measure voltage, a battery that is almost empty, could be interpreted as close to fully charged once it is back at rest [78]. If the same measurements only were to be done while the battery is at rest, the difference from fully charged to empty could be as little as half a volt for a 12 volt battery.

Since there are so many parameters for measuring the state of charge for a battery, we have decided to only use a fraction of these as we only will measure voltage and charge depleted from the battery. The reason for this choice would be that our employer only wanted a estimate of the charge and a indication of when the battery is running low, so that it would be either possible to change the battery or turn the system off. By only measuring the voltage and charge depleted from the battery, the estimation will be something close to a fuel gauge on a car or a boat, in other words, only give a close approximation.

2.6 Monitoring And Regulation of Temperature

We were given the requirement to monitor the temperature of the camera environment, so that the temperature does not go above or below the cameras operating temperature. Given this information we presented the option to our customer, that regulating the temperature could also be an option. This suggestion was well received and if this would be possible, it should be implemented. Knowing this, the task to regulate temperature became a reality.

The temperature sensor that was already implemented in the system did not read the temperature correctly, which was one of the challenges for us to solve. After understanding how much interference the inverter made on the DC part of the circuit, we got an understanding that this might be the cause of the miss information the sensor provided. This is something that our customer confirmed, since some of the summer interns had uncovered this when searching for problem. Knowing this, we know that the power for the sensor must either be filtered or provide another power source.

2.7 Mismatch With Software Architecture and Software Implementation

Originally the system had UML diagrams, explaining both the system and software architecture. There had also been designed class diagrams. Diagrams was made so that anyone working on Argos in the future could easily understand how the system works, and get to work as quickly as possible. However, because of transitional problems from the bachelor project to the summer interns in 2018, the diagrams no longer matched the system. In combination with all of the changes we are making during this project, it has been requested that the architectural diagrams be updated to correspond with the true state of Argos.

2.8 Software Communication with External Battle Management System

Argos needs a system to receive simulated tactical information and to mark targets of interest. Battle Management Systems is designed for this use, and such systems have been designed by many different publishers. Kongsberg Defence and Aerospace (KDA) are already using Virtual Battlespace (VBS) in other simulations, so Argos is required to use it as well. VBS is published by Bohemia Interactive and is used to simulate battle situations for training and review. VBS is also used to get an overview of a battle situation. The software is created for use in a military setting and is mainly sold to other corporations that manufacture systems for military use.

This VBS plugin will be an entirely new part of the Argos system. The problem is to script a plugin to VBS3, to establish a connection between the Argos system and VBS. This requires an understanding of VBS scripting. The plugin will need to work as a server and collect information from the VBS Mission, format it and send it to the connected Argos program. The plugin will be written in C++ and the VBS script needs to be a part of this plugin. Because VBS is an application aimed towards the military market, there’s a lack of public information about the software and how to use it.

Other than the plugin, we need a demonstration mission in VBS. This mission will need to have markers and vehicles. The vehicles should be separated into two teams and move around, ideally at random. The mission should be located close to the university.

The Argos application also needs an implementation for connecting to VBS, receiving the data, and finally displaying the targets to the operator in the HUD. Having looked at the target system already used internally in Argos, we knew we wanted to base the new implementation on it as much as possible.

2.9 Software Application Target Tracking

Implementing target tracking was one of the features requested of the bachelor team of 2018. That way a position could be marked, and become visible both in the map, as well as in the HUD. Unfortunately during the summer of 2018, some changes were made which affected the tracking, effectively breaking the functionality.

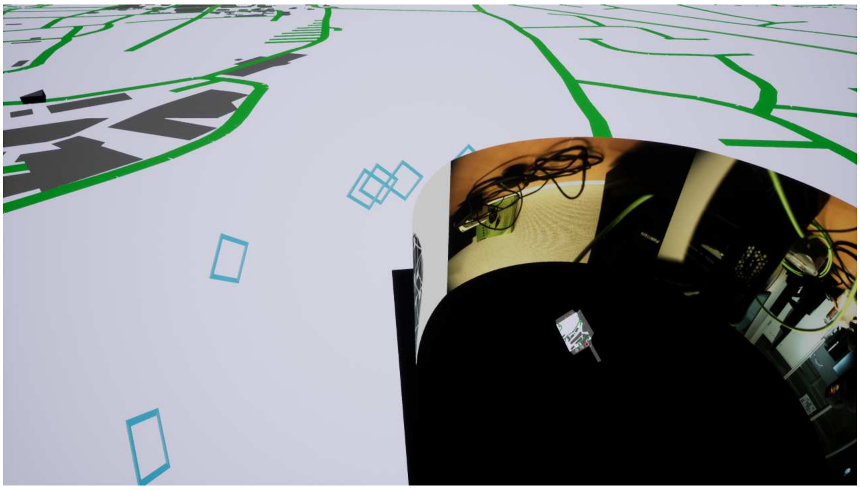


Figure 2.8: Placment of the markers at the beginning of the project

The summer interns in 2018 decided to move away from a main menu, and instead use a single unified map, with all the functionality collected in one place. This should make the software more efficient in use and also allowing live switching between the feeds from the Mako cameras and the Insta 360 camera. The camera spheres ended up getting a different scale, which meant that the HUD markers would end up outside of the camera sphere as shown in Fig. 2.8.

Displaying Information about targets when looked at in the HUD, is a functionality originally previously implemented. This functionality was broken the same way as the markers themselves, in that the HUD would spawn in outside of the camera sphere. Yet another problem that was uncovered was that the markers would be displayed in the HUD from the perspective of the camera sphere, not from the vehicle relative to the map. This also meant that the distance to the marker was way off.

2.10 Broken GigeVision Recording and Playback in Software

Recording the video streams from the GigeVision cameras is a functionality which remains unfinished from previous projects. The same applies to playback of the same recordings. While this is not a highly prioritized problem in the system, the customer has expressed some desire to regain the functionality from the original Argos application. While both features had their implementation started previously, attempting to record footage from the cameras, would silently fail if a predefined directory did not exist before starting the recording. Although the file names for each of the four cameras would be dynamically set to Front, Back, Left or Right.

```
/** Setup recording */
FString filePath = FString::Printf(TEXT("C:/GigeRecording/%s.mp4"), *this->cameraName);
fileName = *filePath;
```

Figure 2.9: Code determining where to save the recording

The application would always attempt to write the recording to the directory as decided in the manner shown in Fig. 2.9. Assuming the directory existed it would successfully write one file for each camera. However if the directory did not exist, it would not give any warnings, thus leaving the user to discover the missing recordings after the fact. Because of the application never actually creating the directory, the user has to manually create a folder named "GigeRecording" on the C drive. This also meant that if a new recording was started, while a recording was already present, it would overwrite the old recordings. After changes to the user interface, actually starting or stopping a recording also became impossible, as there was nothing connecting the recording functionality to the user interface.

2.11 Software Changes Accommodating Changes in Electrical System

Argos has traditionally been able to receive sensor data, telling the application the temperature in the camera boxes. However, as the electrical system would be redone with different sensors, the software needed to accommodate these changes. The customer also requested that a warning be displayed in the heads up display when the battery charge reaches a threshold or the camera box temperatures goes over a limit. Enabling the warning feature would allow the Argos user to perform a clean shutdown, instead of potentially damaging the computer or cameras. Some work had been done on the warning feature earlier. However, the only useful part of the earlier work was that it could connect to a serial device using a the Unreal engine 4 plugin called UE4Duino [112]. As it turned out the previous attempt had been attempting to accommodate the old sensor system used in Argos 1, and would therefore mostly need to be replaced by a new implementation.

2.12 Official Argos Website Update

Ever since 2015, Argos has had its own website, projectargos.net, hosted by Kongsberg Defense and Aerospace [111]. One of the regular requirements to anyone working with Argos is that they also update the website with what has been done by that group, as well as who the group consisted of. While we were working on updating the website, we observed that adding a new group was a very tedious process of copy pasting the last group and color codes. The tedious work is a consequence of a color coding system used to identify the different disciplines of group members. Another issue was that every Argos team has a year logo, displaying the year of the team, as well as being color coded according to whether they are a current, bachelor or intern group as shown in Fig. 2.10. However, no font had ever been agreed upon to write the year, and the exact color used had not been documented, meaning that for a uniform website we had to entirely recreate the logos since 2015.

Figure 2.10: year logos for 2015

3 Project Methodology

Argos 3 is a further development of the previous Argos 2.0, meaning we would need a project model that doesn't expect being able to do everything from scratch. We are also a multidisciplinary group with our own preferred ways of working, meaning we didn't want to force the entire team to try to follow the same methodology.

We decided to adopt Agile as a foundation, since it not only allows, but encourages the flexibility and ability to change, that we need as a group [8]. We ended up effectively replacing any references to "software" in the manifesto with the more applicable "functionality" [2]. In this sense Agile gives us more guidelines on what to prioritize and how to approach possible change.

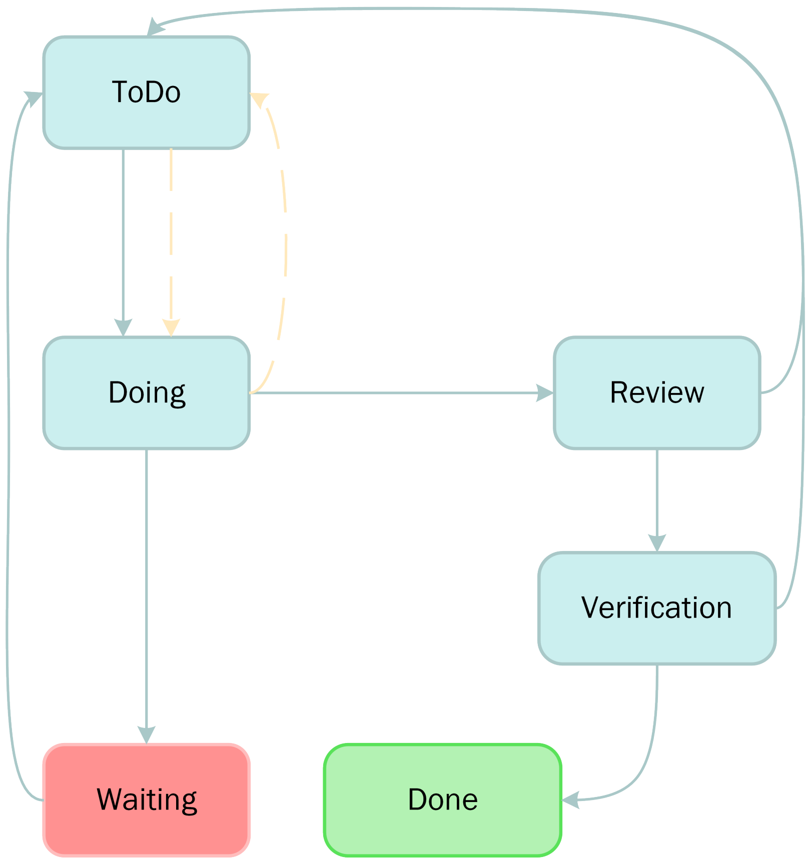


Figure 3.1: Argos 3 task process [18]

One of the most important things to us as a multidisciplinary group, and with Agile as our foundation, is coming to an agreement on how to communicate across the disciplines. In particular we need to be able to do this early on so that the software architecture can be adjusted efficiently. While we did Extreme Programming, we came to the conclusion that it was too determined to get straight to implementation, seemingly skipping design entirely [6]. Because of this we narrow things down a bit with the more specific Kanban, which specifically gives us the Kanban board with cards and different lists. Using Kanban should allow us to quickly discovering problems as they arise, rather than hiding them for later, unlike the waterfall model [4]. Our choice of

Kanban is also fitting, because not everyone one team team knew each other ahead of time, meaning there was no way to predict how well the members would work together. While that may be a problem, Kanban has been suggested to be able to act as an interface for communication between the team members while we get to know each other [5].

Taking inspiration from other project processes, we can further refine Kanban into a more specific process [3], specifying how we want each task to move between the different lists as shown in Fig.3.1. We can can also optimize each members use of time by limiting the number of tasks in Doing at any given time. By limiting each member to at most one task we are ensuring that each member knows exactly what they are supposed to be doing at any time, while also being able to keep track of the whole teams progress [17].

3.1 Agile Industrial Design

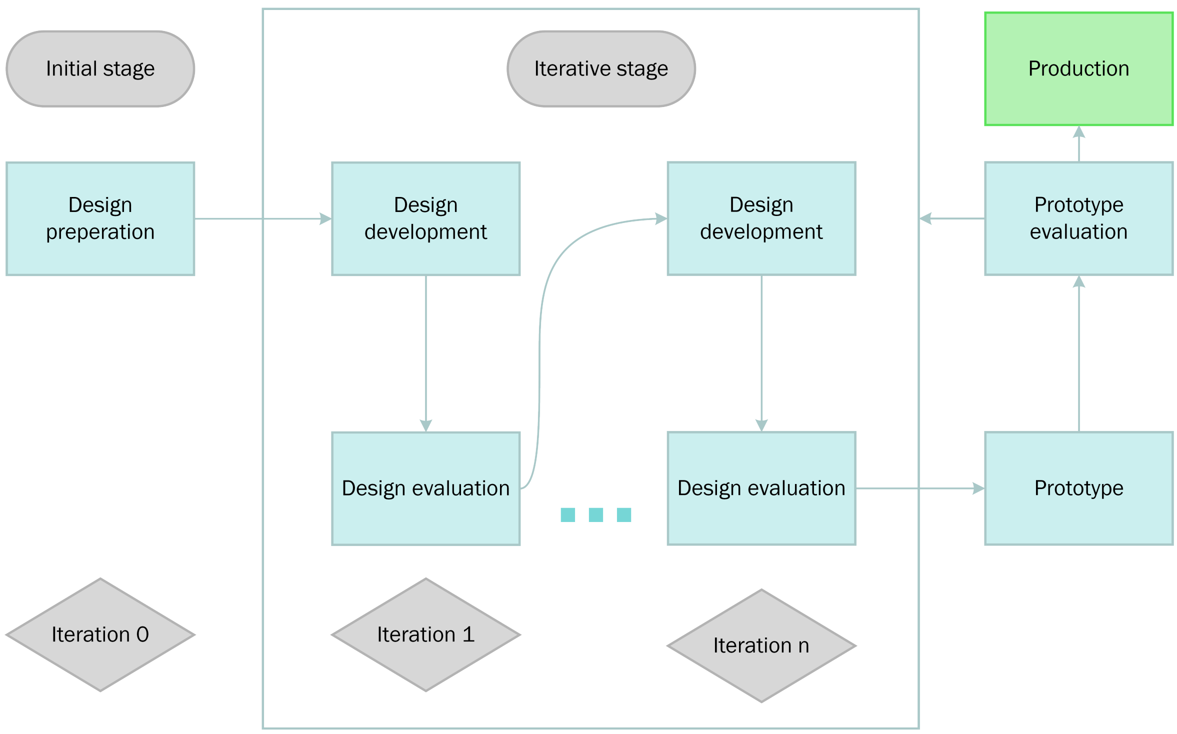


Figure 3.2: Agile Design Process [12]

As Mechanical Engineers we've adopted the agile industrial design method [1]. With some changes to the model. In Fig.3.2 we have added three extra steps which is prototype, prototype evaluation and production. We added these additional steps because of a limited budget, as well as being more limited in the time we have to manufacture designs.

In the initial stage we have design preparation. Which contains gathering information, mapping requirements and constraints of the objective with our customers. When the customer is satisfied with the requirements and constraints. We move to the iterative stage [19]. Here we go through multiple iterations until our product fill our customers requirements. The number of iteration needed will vary for every project. Each iteration in the iterative stage contains a design development stage, where we design our interpretation of the product. During the design evaluation, the stakeholders has a chance to give feedback on the design whether they are happy with it, or if they want anything changed. We then move to the next design development iteration with the new constraints and requirements in mind. This way we have a lot more focus on individuals and interaction [11] [13]. The customer is always involved in the design process, ensuring a product they will be satisfied with.

When the customer is happy with our designed product we will move to the last stages which are prototype, evaluation and production. Since our project require us to produce a finished product we have to include the three last steps. In the last iteration we will create a prototype and test it, so it meets the necessary requirements. The prototype can be either a physical model or CAD [20] If the requirements then are met, we can move on to production. If the customer does not approve of the design or it does not meet the requirements, we will have to begin a new iteration.

Since we have such a close cooperation with the customer it's easy to make changes to our design, which is a core part of a agile working method [7]. Since we have a strict time limit on our project we decided an agile model is best. this is due to the reason of being very adaptive and our project is prone to changes [9] [10], which is something an agile model will embrace.

3.2 Daily Meetings

Taking inspiration from other project models, we agreed early to have daily meetings at the beginning of each day [15]. Doing this would make it easier for the team members to keep track of our total progress and be able to give feedback or suggestions [14][16].

3.3 Project Plan

At the very start of the project, as soon as we had a very rudimentary grasp of what we needed to achieve during the project. We made a very temporary project plan, in which we planned to complete all of the tasks in a very sequential order, as shown in Fig. 3.3. With our first project plan, we had accounted for that some tasks could only be reliably completed after some predecessors had been completely, or at least partially completed.

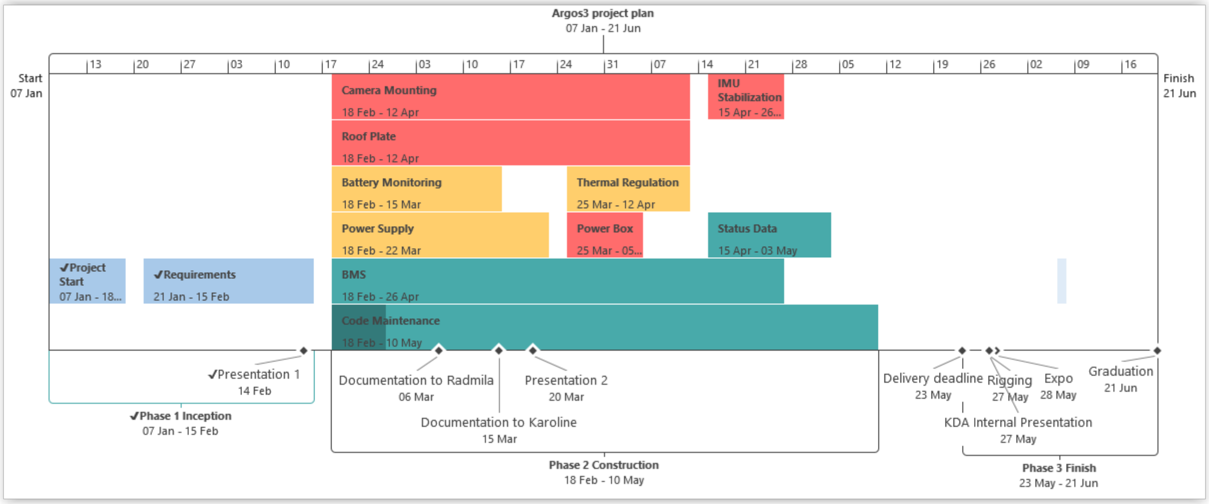


Figure 3.3: Argos 3 initial project plan

After our second presentation, it became very clear that the initial plan, would not be accurate to how we would actually be working. Our employer also requested a more broken down plan, so that it became more clear as soon as we experienced delays, this new plan is shown in Fig. 3.4. As it turned out, we would also not be able to carry on with the project as sequentially as originally hoped. The reason we ended up working a lot more consecutively was to ensure we would finish the design for everything, so that we could order everything we needed as soon as possible to begin production. Based on advice from previous bachelor groups we also made sure to plan ahead so that we would end the project with a week, dedicated only to finishing the documentation.

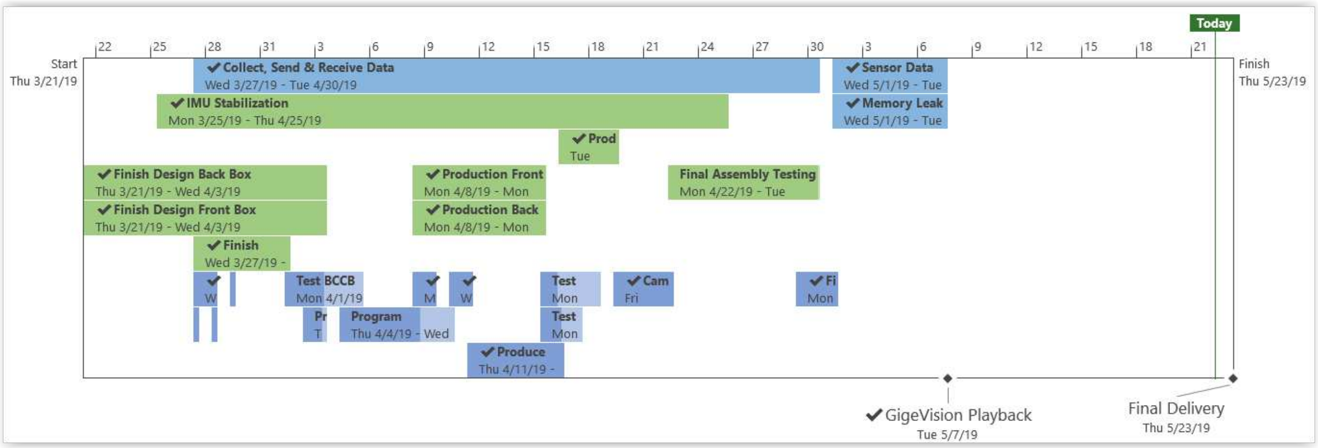


Figure 3.4: Argos 3 Project Plan

Since the beginning of the project it was intended that everything would be documented as the various tasks were completed, which we hoped would give us a reasonable amount of time at the end. During the end of the project we would only have to correct minor issues related to documentation, as well as completing the final testing. Later in the project however, it became clear that various delays would become an issue, hence delaying the dates for production and testing. As a result of the experienced delays, and production taking longer than we had planned, we lost our one week documentation buffer.

3.4 System Requirements

We received requirements from our customer when we began the project in January 2019. After understanding what each requirement meant, we began to break them down into different categories and sub-requirements. The reason we broke them down is to receive a greater understanding of what the customer wants and how we will achieve the goals of the project. To break down the requirements, we took one of the requirements the customer had given us and tried to understand which building blocks were needed to complete this requirement. Once we had found and understood these smaller building blocks, we merged and divided them into so called sub-requirements. The thing that was most important for us when we broke the requirements down, was to never come to a certain solution. The reason for this is that we only want the sub-requirements to guide us to a correct solution and not make us jump to one at the first thought.

In Table 3.1 the main functional requirements can be viewed. The full list of all requirements, including the main requirements, can be viewed in the appendix in Section A "List of Requirements".

Table 3.1: Argos 3 main functional requirements

F.1.1	The system shall have a BMS simulation that simulate and send data to Argos	A
F.1.2	The system shall be able to display BMS information overlay from an external data link	A
F.1.3	The Argos vehicle/system shall be accessible/visible in the BMS	C
F.1.4	The Argos application and BMS shall be capable of two way communication	A
F.2.1	The power source system for Argos in a car shall be redesigned and supply a stable power and voltage to the Argos system	A
F.2.2	The power source system shall be easy to mount and dismount in a vehicle	A
F.2.3	The power source system shall supply status information that can be accessed or viewed in the VR goggles	A
F.3.1	The camera system shall be redesigned so that it is possible to access the cameras in a short time period (3 minutes)	A
F.3.2	The cameras shall not be affected by the environment	A
F.3.3	The camera system shall take no longer than 5 minutes to mount/dismount on a car	A
F.4.1	The system shall use an IMU (Inertial Measurement Unit) align the system when used in a vehicle	A
F.4.2	The IMU shall be stabilized in order to prevent drifting when used in a vehicle	A
F.5	The Insta360 camera can be mounted on a vehicle	B

3.5 Risk Assessment

There will be many risks when we work with this project, and they will have different consequences for our progress. When we assess the risks involved, we will score each one with numbers between 1 and 5 for the probability and for severity, as we can see in Fig.3.5. The probability for a risk is assessed by doing an educated guess on the likelihood for a risk to occur and rate it there after. The Severity of a risk is assessed by the additional cost if it were to happen, the performance cost and what effect it would have on the schedule [22]. By multiplying these two numbers we will get an overall score that we can compare with the Risk Matrix. This gives us an indication on which risk we should prepare for, so we can minimise the consequences.

Green indicates that it is only a minor inconvenience. Orange is a medium problem that we should prepare for. Red could become a huge problem and will probably make the project halt to a standstill. We need to prepare for this, so the consequences will not be so severe [21].

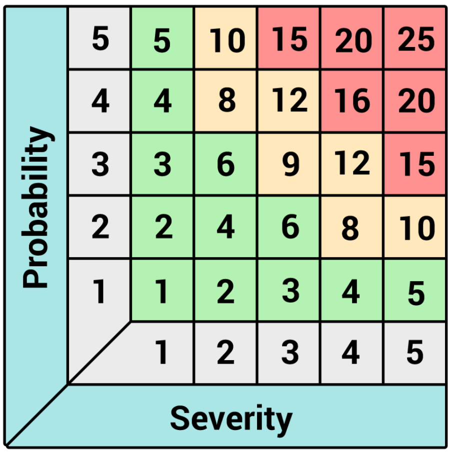


Figure 3.5: Risk Matrix

3.5.1 Risk Table

The Risk Table is sectioned into an ID, the Risk description, probability and severity score, and a total risk level. There is also mitigation row for each risk. The ID starts with *"R"* to indicate a risk, and a number to separate the risks from each other. The probability and the severity is included so we can easily understand the risks. The Risk level indicates the severity of the risk.

In Table 3.3 one can see a partial list of the hardware risks. The full list of risks can be viewed in Section C "List of Risks" in the appendix.

Table 3.3: Partial list of hardware risks

ID	Risk	Probability	Severity	Risk level
R.1	GPS failure	2	1	2
Mitigation	We have more GPS's to replace any failed GPS units			
R.2	Development PC failure	1	1	1
Mitigation	All files are backed up on all computers and on KDA's server			
R.3	Argos PC 1 failure	2	2	4
Mitigation	All files are backed up on all computers and on KDA's server and we have 2 other Argos pc's we can use			
R.4	Argos PC car failure	3	3	9
Mitigation	We have a second pc capable of running VR			

R.5	Argos laptop failure	3	2	6
Mitigation	All files are backed up on all computers and on KDA's server and we have 2 other Argos pc's we can use			
R.6	Car failure	1	2	2
Mitigation	We have multiple cars at our disposal			
R.7	Office network failure	1	4	4
Mitigation	We have several switches at our disposal			
R.8	Camera failure	1	5	5
Mitigation	Making sure the requirements for the camera system are fulfilled and ordering repair or buying new ones			
R.9	Mechanical connection failure	2	2	4
Mitigation	Running tests, simulations and prototyping			
R.10	Power failure while testing in car	3	5	15
Mitigation	All files are backed up and someone without VR goggles are sitting in the passenger seat			

3.6 Testing

We will begin by identifying what requirement we want to run a test on, so that we can start identifying what kind of scenario we want to run the test in, this should cover end to end functionality of the product. When the scenario is established, we can begin to define the test case [23]. The test case is a set of instructions that define how we will validate the test and identify positive or negative feedback. To meet these requirements the test case should contain test data, expected results and conditions [24].

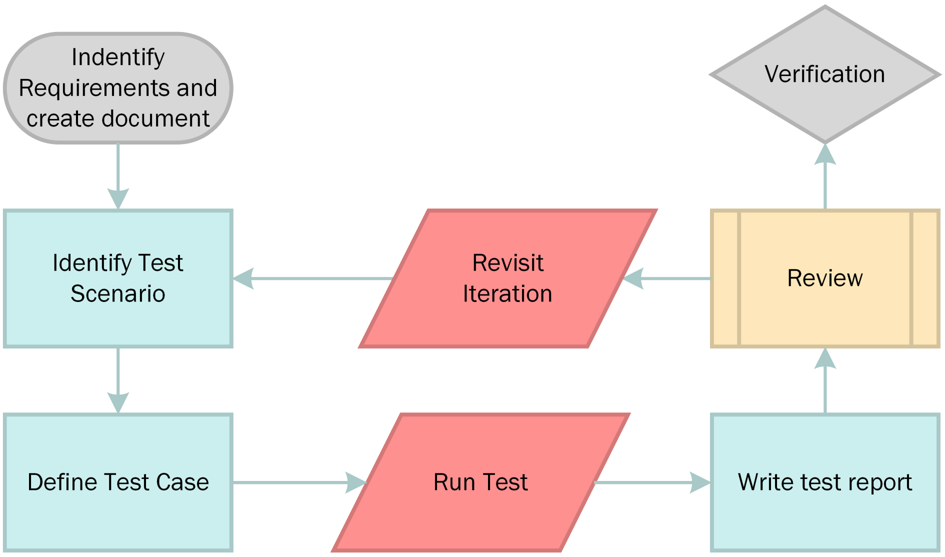


Figure 3.6: Visual demonstration of test procedure

After identifying test scenario and defining test case, the test can be run. When we run the test it is also important to run the test with values that the system do not expect, this can be done by applying different error values.

The tester will then write a report on how the test went, so the rest of the group will be able to review the test, to either be approved and sent to "verification" or make it revisit an iteration and go trough the whole process again.

The full test report can be viewed in the appendix, under the Section B, "Test Report".

4 Modeling and Designing the System

4.1 Mounting System Proposition

We started out with researching how the mounting systems on roof racks and skiboxes for cars are done today. We found Thule's FastClick system, that is used on the Thule Touring ski boxes. Since the mounting to the roof rack is such a *plug and play* method, we decided to have a deeper look into this product. We asked the company if they could share either some technical data or the 2D drawing/3D model. Sadly we got the answer "We're sorry to tell you that we do not share out technical drawings with the public, not even for scholar projects..." [25] Because of this we decided to create our own 3D model of the Thule FastClick system, to get a visual representation of the mechanism, as seen in Fig. 4.1.

Figure 4.1: Thule FastClick

Our CAD model of the FastClick system is non-functional and is only used to get an estimate of the proportions. The FastClick system has a built in torque indicator, for a safe and secure assembly, it also fits on crossbars with a maximum width of 80mm. We have decided to design something including this system, since this is the best time saving system we found. After a brainstorming session, we came up with some different design alternatives for the mounting system. We decided to put up a decision matrix for the different alternatives we considered as seen in 4.1.

Table 4.1: Decision matrix for concept design

Key criteria	Impor-tance	Roof plate	Roof bars	Directly mounted on roof rack	Scaffold bars	Legacy system
Cost	2	4	4	3	3	5
Produce time	5	4	4	1	3	5
Aerodynamic	1	4	5	5	3	4
Stability	4	4	3	3	5	4
Easy to handle	4	4	4	2	4	1
Easy to understand	3	5	4	3	5	4
Looks	1	4	5	3	4	1
Easy to substitute	3	5	4	1	4	5
Mounting time	5	4	5	3	4	3
Weight	4	4	5	5	5	2
Sum		134	135	86	131	110

4.1.1 Potential Mounting Systems

Crossbars One of our ideas was to design some bars that would be fastened on the crossbars from the front to the back of the car like seen in Fig. 4.2. After setting up the Decision matrix we found that this was one of the highest scoring systems. So we decided to create a quick design to see how it would work and to get a visualisation of how it would be.

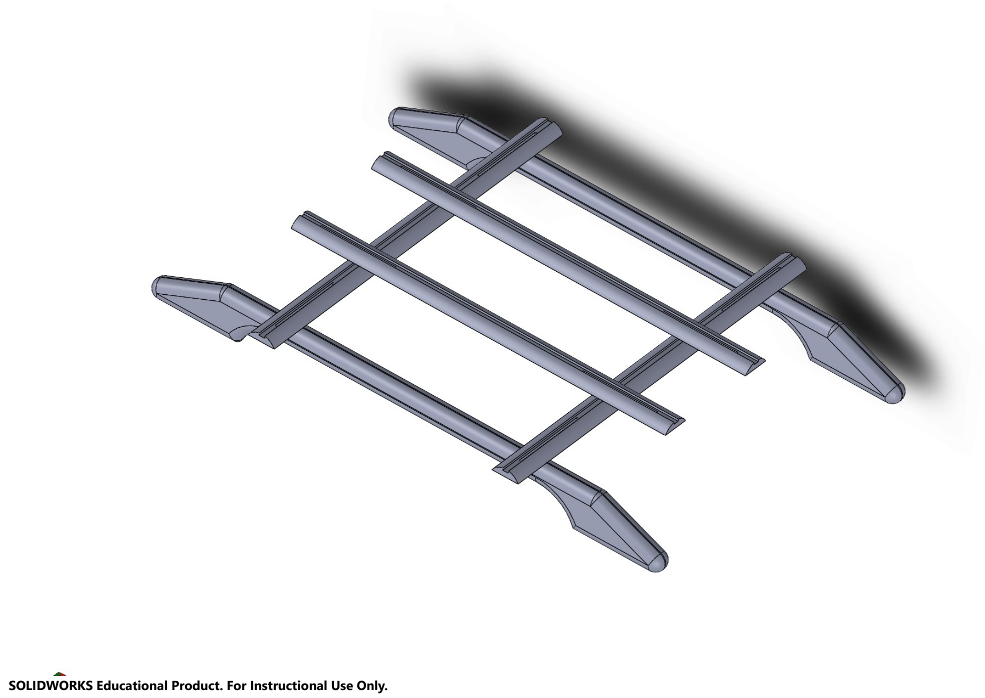


Figure 4.2: Roof crossbars

Scaffold Bars Scaffold bars was a system that was thought up from thinking about rigidity for the cameras directly. It almost scored as high as the roof plate and crossbars but we decided not to investigate it any further. Mostly because it is a very time consuming to design and produce.

Directly Mounted Camera Boxes We thought about not having a separate mounting but building the mounting in the camera boxes so you can mount the cameras directly to the crossbars of the car. This system would cause a lot of problems for all the cars that only have two crossbars to connect to and not four crossbars. Since to get it rigid and mounted securely we need atleast two bars to connect to. This system scored the lowest score of all our brainstorming system so we decided to scrap this idea.

Roof Plate Roof plate is the system currently on the roof of the car. These are made of wood and have too many screws. Our design would get rid of the screws and create a quick release system for the mounting of the camera boxes to the roof plate. We decided to have a set point for the FastClick on the front of the plate so it would always connect the plate to the front crossbars at the same place. Since the distance between each crossbar is different from one car to another, we made an adjustable point to connect to the back crossbar.

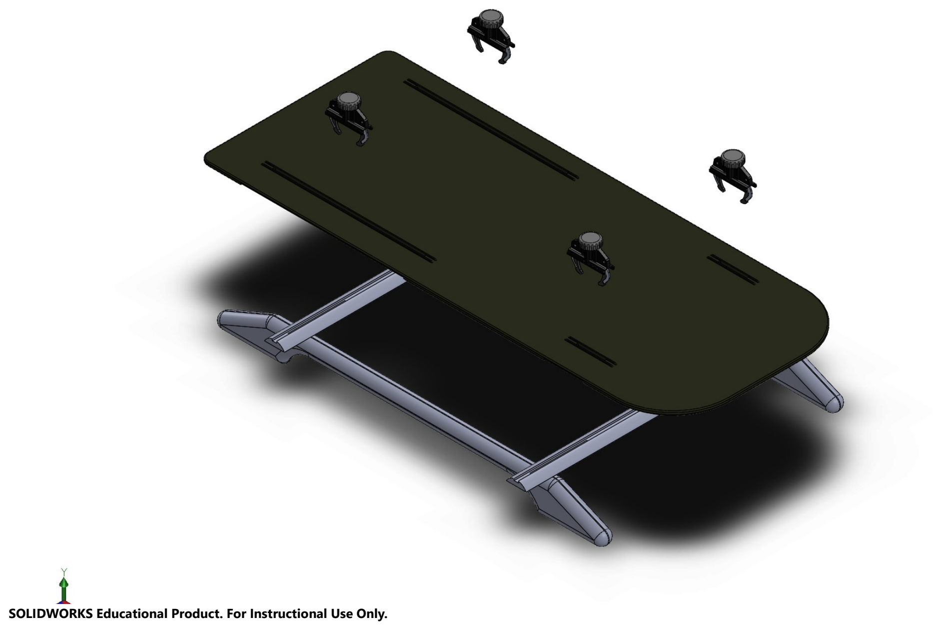


Figure 4.3: Roof plate

4.1.2 Deciding On Mounting Design

After careful consideration we initially planned to go for a roof plate. We took inspiration from an ordinary roof luggage rack. We think the idea of using Thule FastClick, as seen in Fig. 4.1, is innovative and smart. Our design process started with a plain roof plate with Thule FastClick in mind. The design of the roof plate we initially decided on is shown in Fig. 4.4.

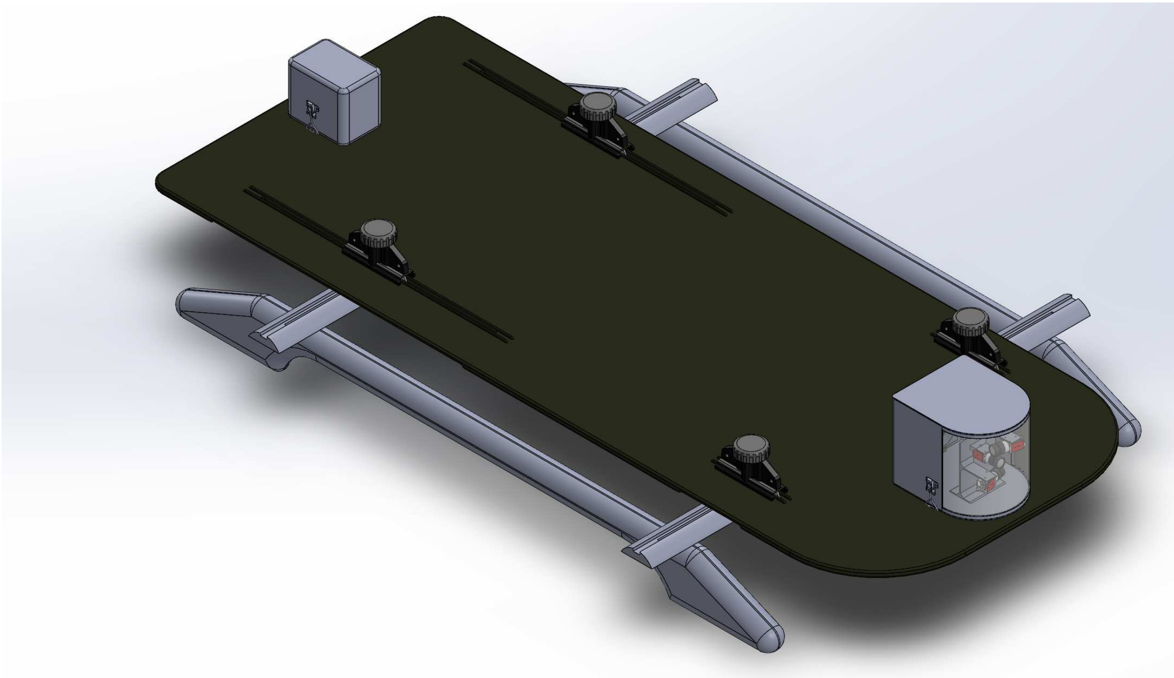


Figure 4.4: Roof plate with camera boxes

With a few minor setbacks and a lot of time spent on design of the camera boxes, we had to reconsider the mounting system. Making the roof plate out of carbon fiber would take to much time to produce, time we didn't have. We were forced to come up with a less time-consuming solution to meet the deadline. We brainstormed a new design that would take less time to implement, and actually turned out to be a more versatile design solution. We decided on using aluminium profile system in the form of roof bars as seen in Fig. 4.5. The first draft of the system is shown in Fig. 4.6.

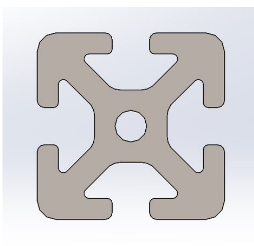


Figure 4.5: Cross sectional area of the aluminium profiles we are considering

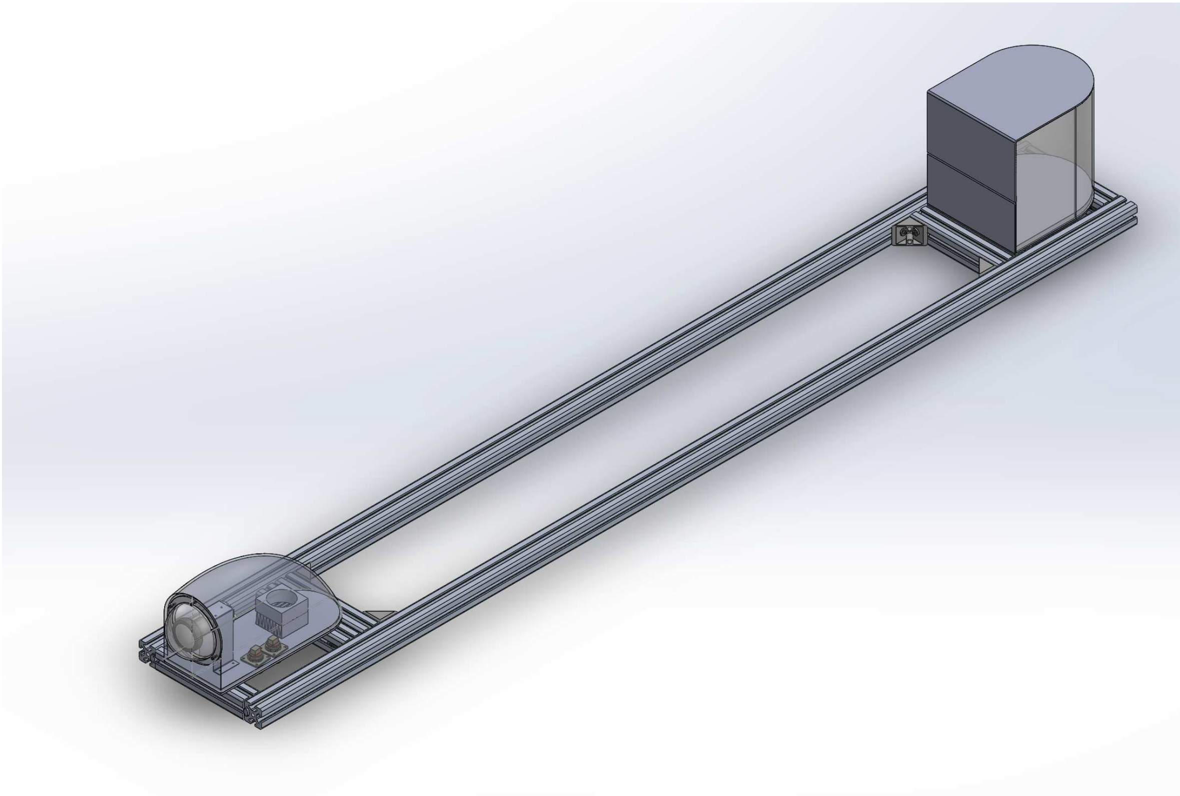


Figure 4.6: Early roof mounting system design with camera boxes

We did calculations to decide on the dimensions of the aluminium profiles. We had various options to go for considering the dimensions. The initial thought was to choose a 40mm x 40mm aluminiums profile. First we made an educated estimate of the weight of the heaviest box. We agreed on 5 kilograms. The distance between support and load is adjustable, but we estimate 450mm from measurements done on the car. In Fig. 4.7 and 4.8, we see a free body diagram(FBD) of the mounting system.

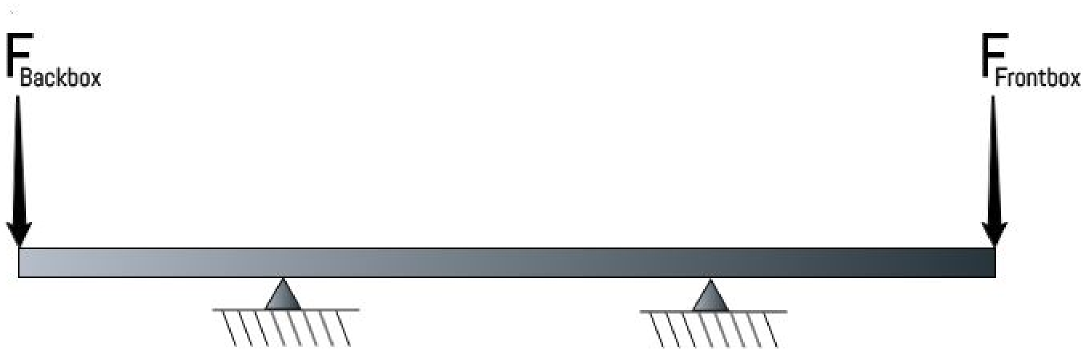


Figure 4.7: FBD of the mounting system

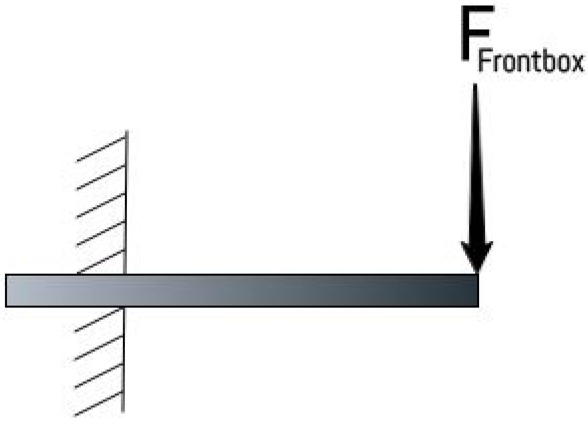


Figure 4.8: Simplified FBD of the mounting system

Using the simplified FBD we have a quick way to calculate the deflection and stresses in the aluminium profile when deciding what to order. Values are taken from the producers datasheet [26]. The deflection is described by the equation

$$f = \frac{FL^3}{3EI} \tag{4.1}$$

for this particular case [26], [27].

We did some fast hand calculations and found out that 40mm x 40mm was more than enough and looked into a smaller dimension. We decided to go with 30mm x 30mm instead, which gives us a deflection in one profile of 0.657mm.

The bending stresses induced in the profiles are described by the equation

$$\sigma_{Bending} = \frac{M_B}{W} \tag{4.2}$$

in bending cases [27]. In our case the moment is produced by the force applied 450mm from the support. The stress induced in one profile is 10.2 MPa. The tensile strength of the aluminium used in the profile is 245 MPa [26]. This means we have a safety factor of 24 [28].

To verify our calculations we ran a simulation in Solidworks. Getting a simulation on the whole system proved to be rather difficult, so we decided to simulate the system we did calculations on. The values of the simulation proved to be similar to our calculations. As seen in Fig. 4.9 and in Fig 4.10 the deflection and stress induced in the beam is 0,521mm and 9,564 MPa respectively.

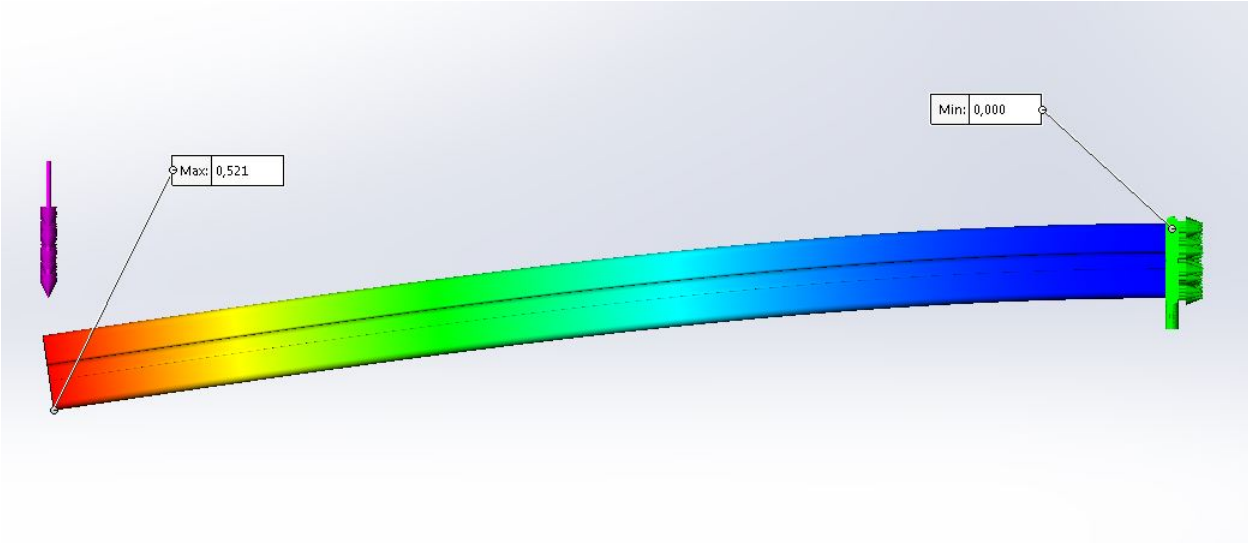


Figure 4.9: Simulation result of deflection using 450mm distance and a 50N force on the simplified system showing a value of 0,521mm

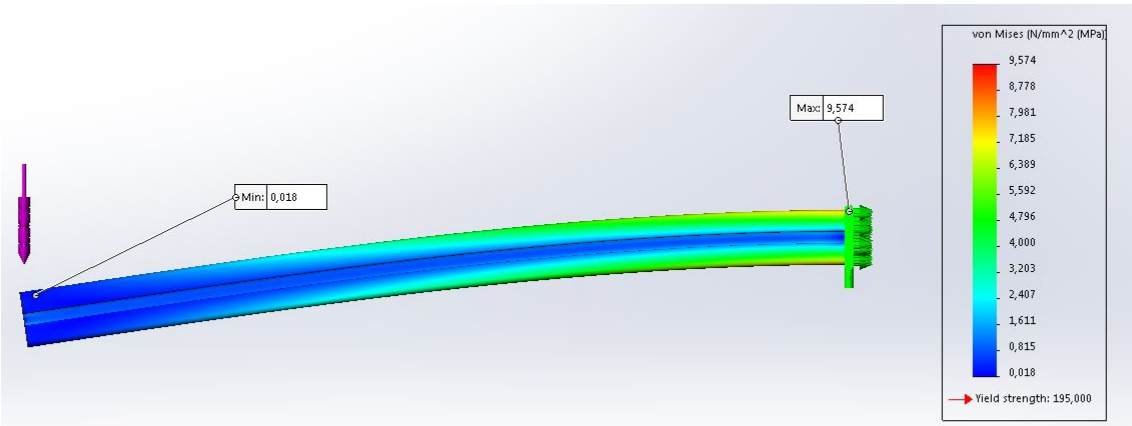


Figure 4.10: Simulation result of stress induced using 450mm distance and a 50N force on the simplified system showing a value of 9,564 MPa

After fighting the analysis tool on several occasions, we managed to get a simulation on the whole system showing increased stress. This was expected, but with a safety factor of 24 from previous calculations, we were not worried. It's still well inside the yield of the aluminium. The calculations from the simplified free body diagram did not take into account that the beam has overhanging loads on both sides. With a difference of about 7 MPa, it's minuscule compared to the yield strength of 245 MPa. The deflection did not increase. Simulation result is shown in Fig. 4.11 for deflection, and Fig. 4.12 for stress.

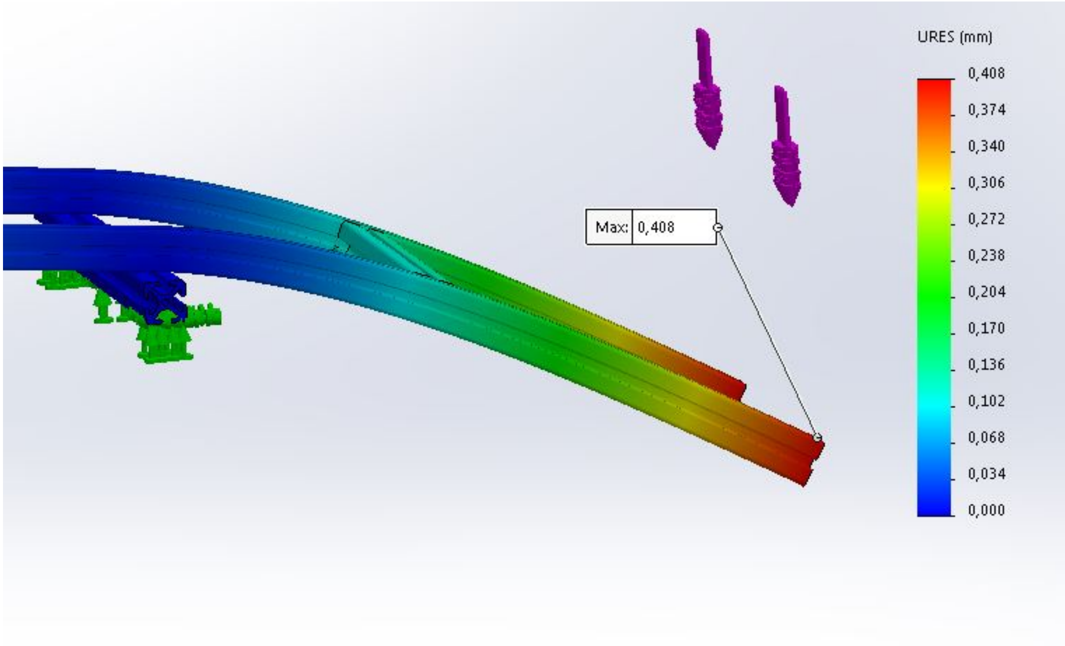


Figure 4.11: Simulation result of deflection using 450mm distance and a 50N force on the whole system showing a value of 0,408mm

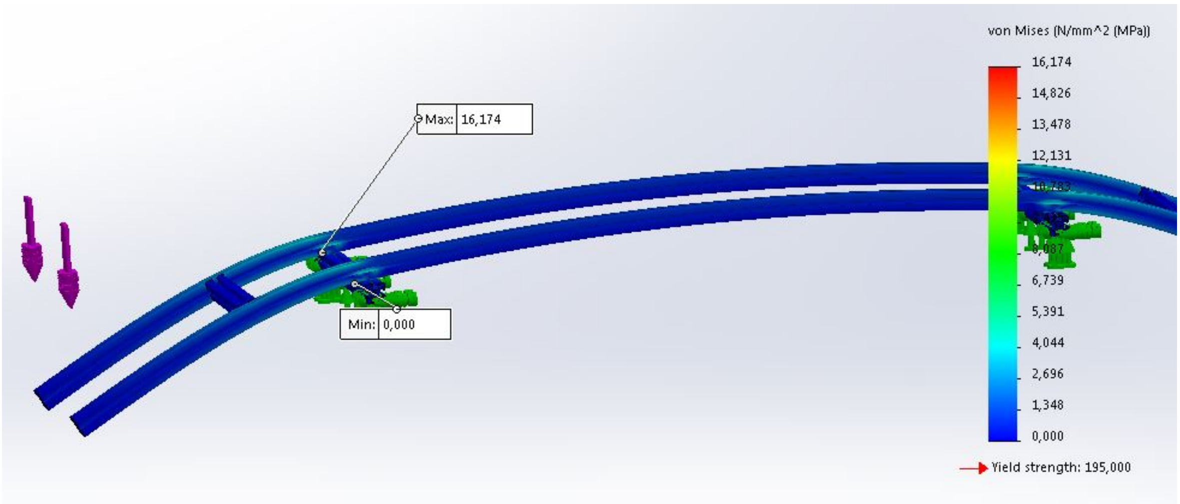


Figure 4.12: Simulation result of stress induced using 450mm distance and a 50N force on the whole system showing a value of 16,174 MPa

As the design progressed we were able to measure digitally what the mass of our system would be. This is handy to know since our system is to be mounted on a car roof and needs to be lifted. We had seen the mass/m in the datasheet and had a general idea of the mass in mind when going forward with the design. With computer calculations, we were able to figure out that the mass was about 7 kg as seen in Fig. 4.13. This is an acceptable mass to lift above the head.

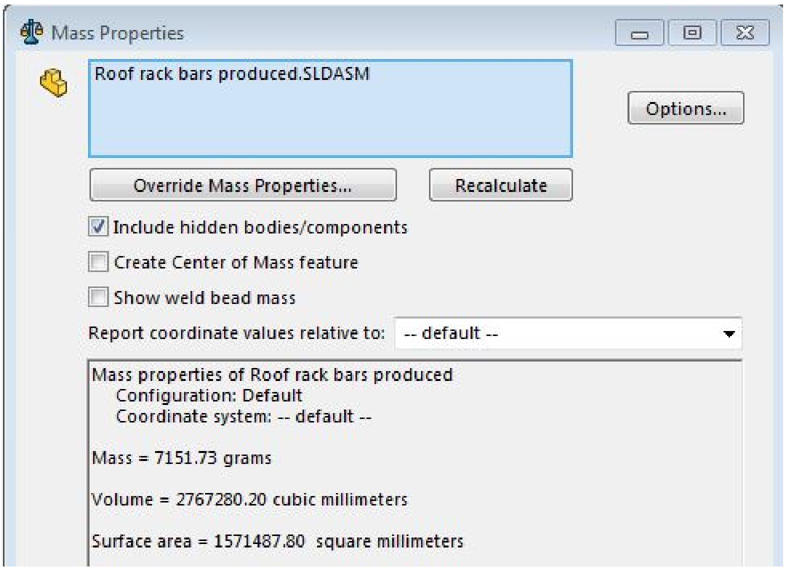


Figure 4.13: Solidworks calculations show the weight of our design, which is calculated to be 7,151 kg

We decided to go ahead and order components so we could start production. The design uses eccentric locks with hooks as a tightening mechanism. In Fig. 4.14 you see the final design of the roof mounting system we decided to go with.

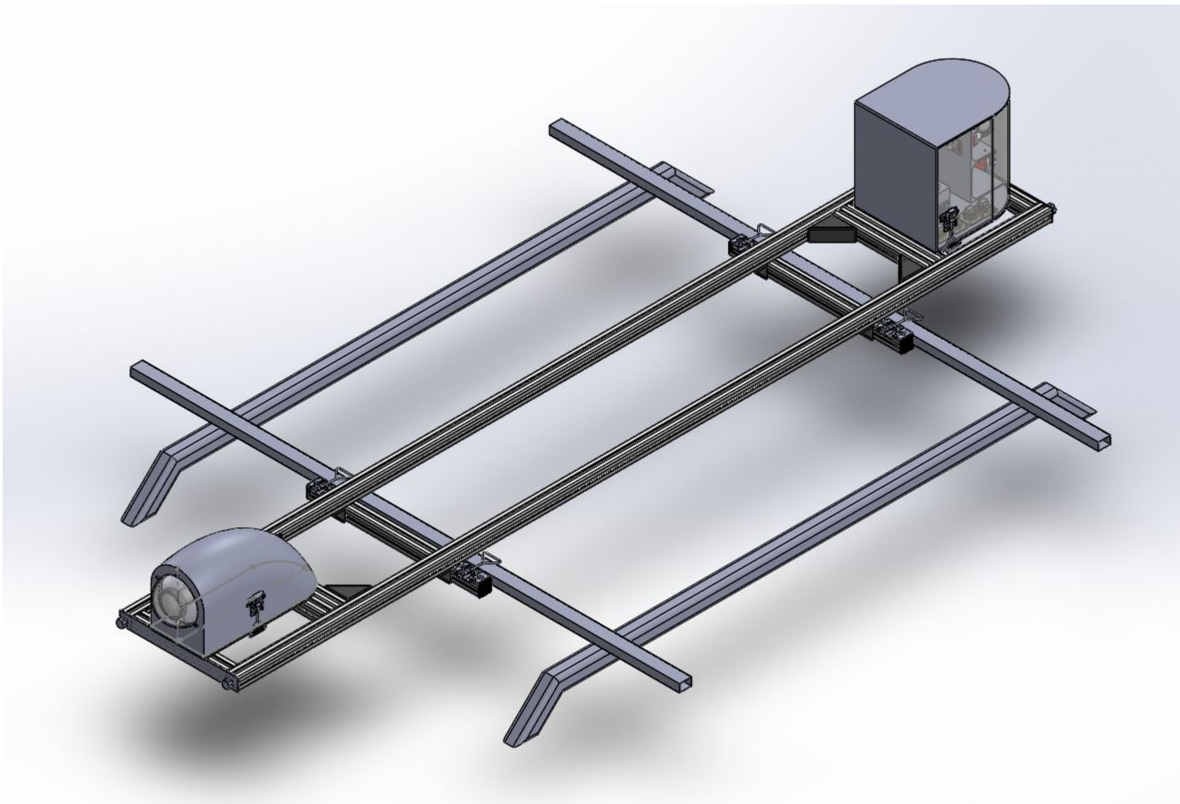


Figure 4.14: Final design of the roof mounting system with camera boxes

4.1.3 Galvanic Corrosion

Using two different metals in conjunction with each other have a risk of inducing galvanic corrosion. The energy potential between two metals will induce a low current that can cause one metal to corrode faster than normal [61]. When using aluminium we need to be careful about using steel screws, bolts, washers and nuts in harsh environments [62]. This means environments that contain salt, which means mostly coastal areas and on the sea. Roads are salted during winter so testing on a road that is not salted is preferable. Parts of the camera boxes and the roof mounting system are going to be aluminium, so this needs to be taken into consideration. To reduce the galvanic corrosion potential we need to use aluminium or zinc coated/stainless steel applications.

4.2 Camera Shielding Propositions

To begin with, we did some calculations on the old Argos front camera box, by estimating necessary information with help from a compendium in fluid mechanics [29]. We calculated the area, A, of the box using computer assisted design tool, SolidWorks, to be about 0.122 m^2. This was achieved using old Argos 3D models. We estimate the drag coefficient, c_D to be about 0.8. For an extreme case, we estimate the velocity, U, to be 28 m/s(slightly above 100 km/h). We also estimated the density of air to be 1.2 kg/m^3. Using the formula for the force applied:

$$
\begin{aligned}
F &= \frac{1}{2}\rho c_D U^2 A \\
F &= \frac{1}{2} \cdot 1.2 \cdot 0.8 \cdot 28^2 \cdot 0.122 \\
F &= 45.9N
\end{aligned}
\tag{4.3}
$$

This gives us a force of about 46 N.

To find different options for a new camera protection system, we started looking at the old camera boxes. We tried to figure out if it was possible to have only one or maybe several said boxes around the roof of a car. It was decided to plot a decision matrix for the different options previously discussed. Decision matrix is a tool for evaluating the best option in a process [48], [49]. As seen in Table 4.2 we discovered several factors that had to be included in our evaluation.

Table 4.2: Decision matrix for concept design

Key criteria	Importance	One camera box	Two camera boxes	One box for each camera
Cost	3	4	4	3
Produce time	5	4	4	2
Aerodynamic	1	3	3	3
Stability	4	3	3	2
Easy to handle	4	4	4	3
Looks	1	3	4	5
Easy to substitute	2	1	1	1
Mounting time	5	5	4	3
Weight	4	3	4	5
Design complexity	4	3	4	5
Number of new parts	2	3	3	1
Wiring complexity	3	5	4	2
Amount of sensors	3	5	4	2
Placement	3	4	5	2
Sum		165	166	124

We score the key criteria on a scale of one to five and multiply with the importance then add them together. This will help us see the best option and help us do good decisions. The highest score will be the best choice. We based the scoring system on the same model we use for risk, developed by United States Department of Defence [22].

Cost We estimated the cost of one camera box and two camera boxes to be insignificant. An option where we make 1 box for each camera would increase the material used. Also, the number of sensors and heating elements would increase. This means the cost would naturally increase.

Production time We estimated the production time to be about the same for one or two boxes. If we choose one box the design will be more complex. We would need transparent material 360° around the cameras. With two boxes we don't get the complexity so the assembly would be easier and less time-consuming to make. Time is limited in the project so we need to be smart about it.

Wind resistance The boxes are to be placed on the roof of a car, where wind resistance would be a criterion to consider [38], [40], [37], [39]. In agreement with our customer, we will design according to what the system needs now, and not what the system might need in the future. The speed the car will be driving while using the Argos system is minimal. From equation 4.3, formula for air resistance, we can see that the most important factor is U, which is the speed of the air relative to the object [29]. Since it's squared, a doubling in velocity would quadruple the force applied on the box. With relatively low speeds the force will remain low as long as we do a reasonable shape in design. We concluded that this criterion is not a high priority and therefore gave it a low importance score. It is still going to be in our minds while doing design.

Stability The stability of the boxes are fairly important. Shaking of the image is not preferred. Getting stability for four different boxes we estimated to be difficult, The boxes would need to be placed at the edges of our mounting plate. The edges are not supported and would act as a cantilever beam, prone to vibrations, though the stiffness of the plate would decide the impact of the vibrations [41], [42], [43].

Easy to handle We want to make the boxes as easy as possible to handle. This means carrying, mounting and lifting should be doable by one person. It's not in the requirements, but it's something we aim for. We estimated the options for one and two camera boxes to be equal. One box will be heavier and bigger, but you would need fewer trips to carry the cameras. Two boxes option would be lighter and smaller, and therefore easier to handle. Four boxes we estimated to be a hassle, too many trips back a forth to carry all the cameras. This is fairly important, as we want a good user experience [44].

Looks The looks of the boxes are not important since this is a research project and not a commercial product. The main focus is that it's going to be functional. We estimated the one box for each camera would look better, but again it's not important.

Easy to substitute None of the options will be easy to substitute if a box breaks, but we will have blueprints of the boxes so a box can be reproduced.

Mounting Time Our goal is to make the mounting solution tool-less and easy. This means the amount of time difference between the three options under consideration is minuscule. With the same mounting process on each option, we estimated one box to be the fastest, giving this the highest score. Weight is an important criterion. The boxes will be lifted overhead to the roof of a vehicle. Strength differs from person to person, so we want to make it lift-able by the majority of adults. The optimal choice would be 4 boxes in this category since it means less material to lift.

Design Complexity The design complexity will take away from our time. A more complex design would take more time than an easier design. We think this criterion is important. One box for each camera would be the easiest. We could do the same design for each box, making the design process faster. One box we estimate to be hard to accomplish, we would need a 360° field of view with a transparent material.

Number of New Parts The parts we need will vary with each option. With four boxes, we will need more new parts than with the current solution which has two. We estimated two boxes and one box to be equal. We will need new parts for both, but they can be different for each option. Some parts can be reused in all cases, but we scored four boxes the lowest. Considering the budget, reusing parts could be beneficial.

Cable and Wiring Complexity There will be a lot of wiring and cabling with a new system of 4 boxes. The cameras need Ethernet and the sensors and heating units need connections as well. The current system has many of the connections we need in the new, which we can reuse. Wiring in one contra two boxes will be easier which makes one box option score highest.

Amount of Sensors The amount of sensors will vary with each option. With one box we will have the least amount of sensors which is optimal. With four boxes we will have the most number of sensors. This is why one box option scored the highest.

Placement: Placement of four boxes in different positions would make the field of view weird and it would need a lot of image processing which we don't have time to implement in the project. With one box we would have issues with the amount of car roof we would get in the images of the different cameras. Two boxes seem optimal giving it the highest score.

4.2.1 Material Options

The protection system will consist of several parts with different material and because of this, we decided to do the same for deciding material, as we did for deciding how many boxes would be optimal, namely a decision matrix. The most important aspect of this project is time and therefore weighted "Produce time" and "Availability" the highest factor as seen in Table 4.3.

Table 4.3: Decision matrix for material options

Key criteria	Importance	Steel	Aluminium	Carbon fiber	Wood
Cost	2	2	3	5	3
Produce time	5	3	3	2	3
Professional machining and bending	3	2	2	5	5
Looks	1	4	5	5	1
Corrosion resistance	2	4	5	5	1
Weight	4	3	4	5	1
Availability	5	4	4	5	5
Easy to maintain/treat	2	3	3	5	5
Sum		75	84	105	78

Cost The cost of the material will vary vastly. The main structure will contain most of the same material for simplicity and time-saving. The design will vary with the material we choose. The amount of material we are going to use on the boxes, we estimate to be low. This means the impact on the budget will not be massive, but it will still play a part. The best choice in this criterion will be carbon fiber as we have material free at our disposal in Krag composite lab. Our customer is Kongsberg Defence and Aerospace which is the main contributor to the composite lab on Krona.

Produce Time Choosing carbon fiber we have more influence in the process since we will be making the whole product ourselves. Making carbon fiber composite is a slow process [46], but we're ready to take that sacrifice in the form of work hours. Compared to choosing something that needs to be machined and/or bent which require a professional to do which is a risk factor.

Machining and Bending To process metals, we will need a professional. We can process the carbon fiber ourselves in the composite lab. Potentially we could use Kongsberg Defence and Aerospace own composite lab if we need larger parts for our project. We can also process wood in labs/workshops on Krona.

Looks We think carbon fiber will look cool, but it's not an important criterion for the final product since this is a product for a research project, not a commercial product.

Corrosion Without some sort of treatment metals will corrode(I.E. sealer/varnish coating) [31]. We need to be careful about galvanic corrosion too. Galvanic corrosion could potentially "eat up" one metal if we use different metals in connection, making it corrode faster [30]. Carbon fiber composite corrosion will be insignificant in the projects lifetime [45]. The carbon fiber composite will corrode the aluminium if used as a conductor [47]. Wood will rot.

Weight In addition to the top criteria, the weight of the final product is also in our focus. The product will need to be lifted overhead to the roof of a car. Not every person is of equal strength so the product should be lightweight.

Availability We have limited time and this is connected to the availability of the material. This is shown in Table 4.3 by rating it five out of five in importance. The university has its own composite lab. We have the lab and the material free at our disposal.

Easy To Maintain/Treat The material should be easy to maintain, where carbon fiber scores high [45]. Steel or aluminium would need coating [31]. Varnishing is it's own profession and could be a challenge to do right. Wood would also need treatment which is fairly simple to do.

Transparency material The transparent section of the boxes will need to consist of some sort of glass in the form of either polymer or ceramic. To decide on the material we developed a decision matrix to help decision-making easier.

Table 4.4: Decision matrix for glass options

Key criteria	**Importance**	**Ceramic glass**	**Polycarbonate glass**	**Acrylic glass**
Cost	3	2	3	3
Weight	5	2	5	5
Availability	3	3	4	4
Easy to maintain/treat	5	5	5	5
Resitance to scratches	4	3	2	5
Easy to process	5	2	5	4
Working temprature	5	5	4	4
Resistance to UV	2	5	4	5
Impact resistance	2	1	5	2
Sum		109	142	145

Cost We valued the cost to be a three out of five in importance. We assumed that processing a ceramic glass with the required shape for our boxes would be expensive. A better choice, which shows in the score, is a polymer-base glass. The amount of glass we need is small, so the cost of the different polymer-base choices would be negligible.

Weight Ceramic glass is heavier than that of a polymer-base(I.E. Lexan, a type of polycarbonate, has a density of 1.20 g/cm^3 [53]), while commercial silicate glass has a density of >2.1 g/cm^3 [55]) which gives the ceramic glass a very low score. The difference in our polymer-base options are negligible [53], [54].

Availability All options are readily available, but ceramic glass would need extra time which would need to be planned for. Cheap acrylic glass would be most convenient since it's in stock at local stores.

Easy To Maintain All options would be treated/maintained the same way. Cleaned with any commercial cleaner suited for the material.

Resistance To Scratches Acrylic glass has better resistance to scratches than polycarbonate [52], [50], [51]. You can get polycarbonate that has been treated, which gives it good resistance to scratches. However, treated polycarbonate sheets are only for flat applications [56]. Ceramic glass does not have good resistance to scratches.

Easy To Process This is a very important point since our time is limited. Polycarbonate can be bent and formed while cold. Acrylic glass would need heat treatment to be bent or formed. Ceramic glass would not be easy to form, so we would have to pay a professional to do it for us.

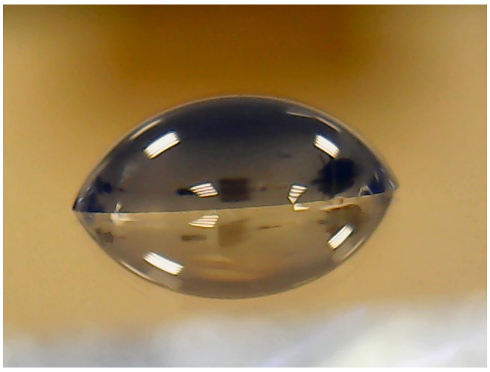
Working Temperature Polycarbonate can potentially withstand temperatures of up to 145 $^\circ$C [53]. Acrylic glass has a service temperature of up to 70 $^\circ$C [54]. Realistically, the service temperature will never exceed these ranges.

Resistance To UV-Rays Polycarbonate does not deal with sunlight very well. It can get a yellow color when exposed to UV-rays over time. There are options for treated polycarbonate which can handle UV-rays, but then again these sheets are only for flat applications [56]. Acrylic glass will not get significantly damaged by UV-rays [54].

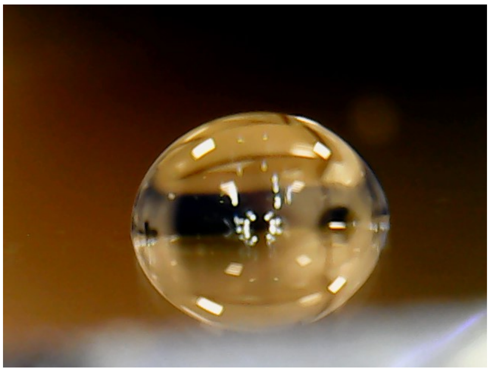
Impact Resistance As shown in the data sheets the acrylic glass [54] has a low impact strength compared to polycarbonate [53], [52]. Acrylic glass will shatter when introduced to an impact of sufficient magnitude(1/11 that of polycarbonate), where polycarbonate is not prone to shattering [50], [51]. Ceramic glass will also shatter when introduced to a high enough impact.

4.2.2 Water Repellent Glass

According to requirement F.3.2.2, the cameras shall always have a clear view. This means the glass of the camera boxes should be able to repel water if they are used while raining. We had a cheap brand water repellent available, so we were able to test if water repellent substance worked on acrylic glass. The main purpose of test T.4.1 is to compare contact angle before and after treatment of the acrylic glass. When a surface is repellent to water, also called hydrophobic, the contact angle is larger than when it's not repellent to water [32]. As we can see in Fig. 4.15a and 4.15b, the contact angle is significantly increased after treatment.



(a) Water droplet on acrylic glass before Biltema rain repellent treatment

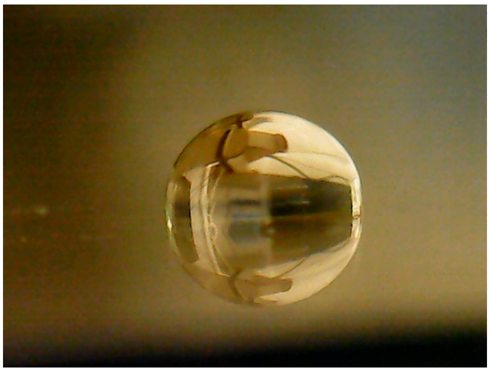


(b) Water droplet on acrylic glass after Biltema rain repellent treatment

Figure 4.15: Pictures of water droplets on acrylic glass before and after repellent coating treatment

We ordered a brand rain repellent to test if it was better than the previous. We did a new test, T.4.2 to get data. From Fig. 4.16a and 4.16b we can see that the contact angle has increased as in the previous test, T.4.1. When we compare Fig. 4.15b and 4.16b, we can see some slight differences in contact angle. Fig. 4.16b has the best potential to repel water as it has the largest contact angle.

(a) Water droplet on acrylic glass before Rain-X rain repellent treatment



(b) Water droplet on acrylic glass after Rain-X rain repellent treatment

Figure 4.16: Images of water droplets on acrylic glass before and after trademarked rain repellent coating treatment

Test reports for contact angle can be found in Table B.20 and Table B.21.

4.2.3 Camera Field of View

When we started the designing phase for the new camera boxes we started by finding the angle of view of the cameras [35]. According to the datasheet for the cameras they have a horizontal view angle of 81.9°, vertical of 61.2°, and diagonal of 102.9°[33]. We decided to model the angle of view and field of view in SolidWorks to see the overlapping or blind spot in the former system, you can see the modelled angle of view in Fig. 4.17.

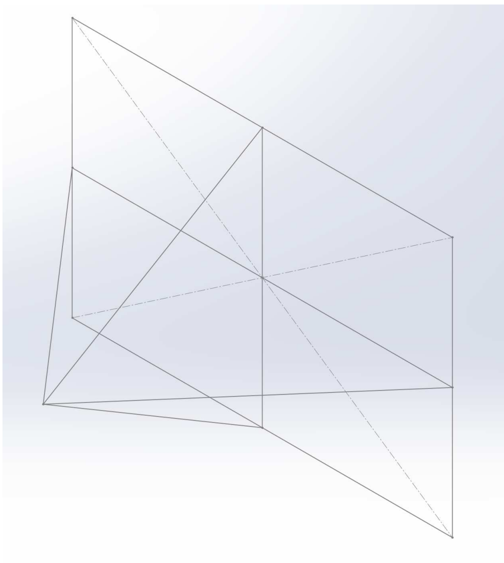


Figure 4.17: Camera angle of view

As seen in Fig. 4.18 we have a huge blind spot within the former Argos camera box. We did a visual test to see check how much information we were losing see Table

B.42 for test report. We confirmed that the former system had blind spots between the cameras.

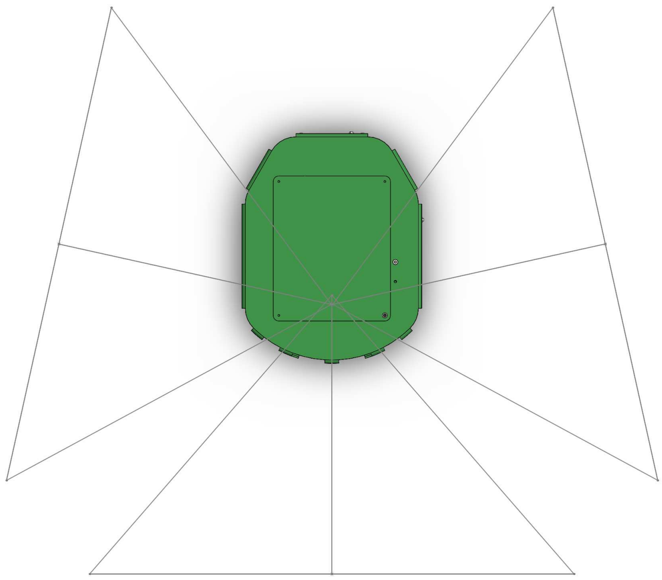
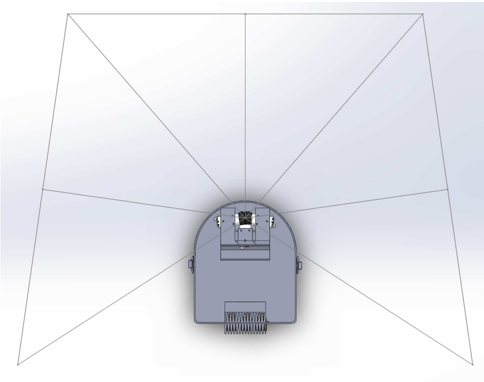
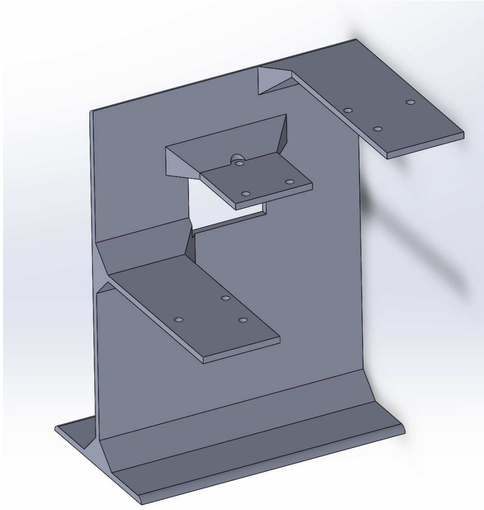


Figure 4.18: Former front box angle of view

To check the new viewing angle we created a 3D printable model we could mount the cameras on and did a new test of the viewing angle as seen in Fig. 4.19a and Fig. 4.19b. According to the datasheet for the Kowa lenses has 81.9°is the horizontal angle [33]. But when we did our test see Table B.43 it failed because we had a lot of overlapping in the picture. We made this first test to both test the viewing angle and making sure that the camera mounting isn't obstructing the view of the cameras. We found out that the viewing angle was wrong, because the cameras had a larger angle of view than the datasheet provided. However the camera mounting see Fig. 4.19b was accepted because it did not obscure the vision.



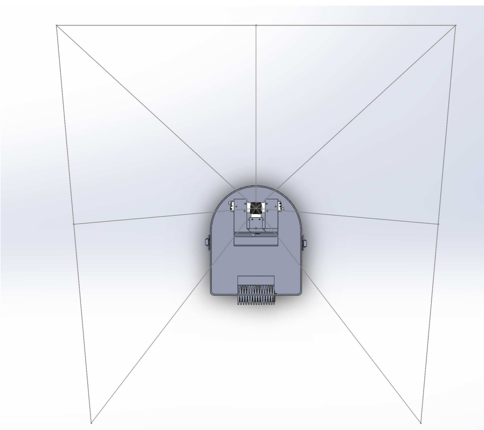
(a) New camera box with 81.9°camera angle



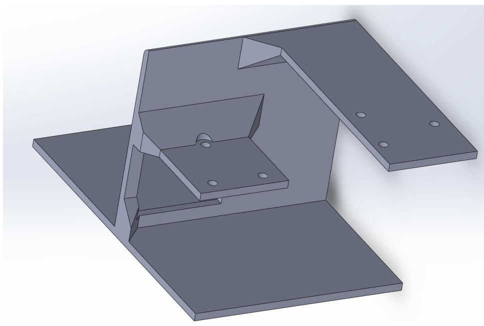
(b) Testing rig for 81.9°view angle

Figure 4.19: Camera angle 81.9°design

The second test was made with just increasing the angle of view from 81.9°to 95°[33]. With this solution we made another test rig with the new measurements see Fig. 4.20a and Fig. 4.22b. The Fig. 4.22b wasn't a good design for a test rig as it required a lot of support material to be 3D printed. And the modelled angle for the Mako cameras was again too small compared to the real Angle of view so this design failed see Table B.45 [34] .



(a) New camera box with 95°view angle



(b) Testing rig for 95°view angle

Figure 4.20: Camera angle 95°design

From the camera datasheet we had for the Mako G-223C, we have at distance=500mm horizontal and vertical is 927 x 492mm respectively [34]. We modelled this in SolidWorks too see Fig. 4.21 and quickly saw that this angle was less then the angle of view at the second test 95°. Therefore we knew that this angle was too small to be the right one.

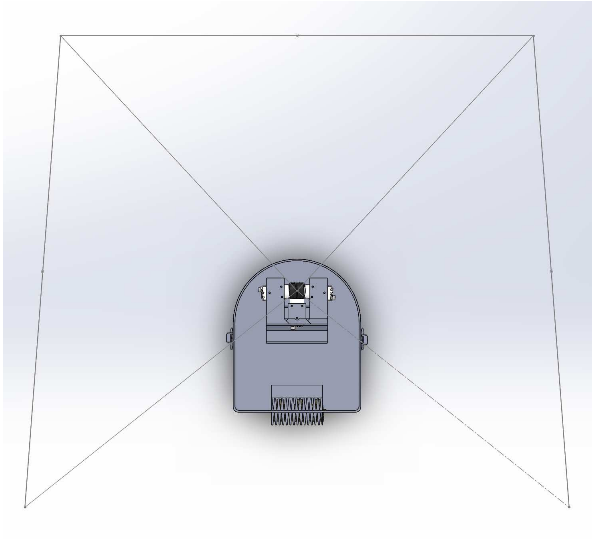
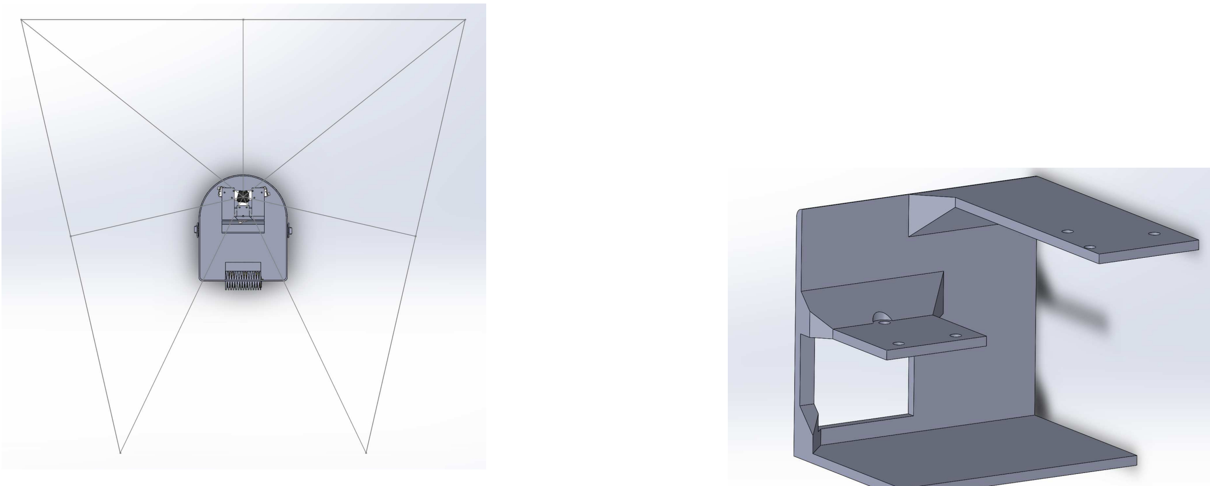


Figure 4.21: Front box with the FOV

Since we couldn't make sense of the connection of the theoretical angle of view and the physical viewing angle we decided to try to use the diagonal angle as the horizontal angle 102.9°[33]. And we made some changes to the testing rig to make it easier 3D printable for easier testing see Fig. 4.22b. The way we designed the new testing rig makes it possible to print without using support material. This test ended up as inconclusive because the camera seam had overlapping higher up and had a blind spot lower on the camera feed see Table B.45.



(a) New camera box with 102.9°camera angle

(b) Testing rig for 102.9°view angle

Figure 4.22: Camera angle 102.9°design

After having an inconclusive test see Table B.45 we had a meeting with Bonnie Nordahl Albert Uchermann who is an associate professor in the department of optometry, Radiography and Lighting Design for USN. He confirmed that our way of modelling the camera view was correct and gave us some information about worse image quality at the edge of the cameras FOV. He also gave us some information about camera focal length and how the lens distance from the sensor could change the angle of view and field of view of a camera creating an uncertainty in the angle of view provided by the datasheet [33], [36]. He advised us that the best method to find the correct angle of view is to run more tests.

We decided to try 98°angle of view because it is between 95°as we already have tried which had too much overlapping and 102.9°which had overlapping high in the seam and blind zone lower in the seam as you can see in Fig. 4.23a and Fig.4.23b. We have decided to go for 98°because at this point we can't optimize it any further, we have a blind zone about 400mm out from the seam and after that we get overlapping. We have made sure to design with the correct focal length so the AOV originate from the correct point. So theoretically we should only have overlapping at the same point in the seam and not a blind zone closer to the cameras and overlapping further from the cameras. Since we don't need any information that is within 400mm of the cameras we have decided to go with 98°AOV as this is the closest we have come to a perfect seam.

(a) Blind zone in the middle of the seam

(b) Overlapping high in the seam

Figure 4.23: Visualisation of barrel distortion

We have also learned that our lenses has -10.7% barrel distortion which leads to narrowing in the middle of the camera feed and stretching higher in the camera feed [33], [57], [58], [59]. That is the cause of our overlapping high and low in the seam and a blind zone in the middle of the seam at higher AOVs. This can be solved in two ways, either buy new lenses which has less distortion or you can fix it in software. We're looking at ways to solve it choosing the latter after discussing it with our costumer. We also learned that barrel distortion can cause the incidence angle < exit angle [59].

4.2.4 Front Camera Box

After sorting out the problem with the AOV for the front camera box we could continue the design of the front camera box. The front camera box will use three Mako G223 cameras to get a combined 245°view angle. As we established earlier however this angle was larger than the datasheet provided. We have decided to continue with a eccentric lock mechanism for quick access to the cameras to fulfil requirement F.3.1. We use plexiglas because we need the glass to hold the same position at all times. Lexan can be cold bended and we do not wish this property in our design, we need the glass to hold the same position.

Iterative Design Process Front Camera Box

We use a iterative design process to create the design of our camera boxes. With multiple design iterations. Our first iteration had a couple of ideas put together see Fig. 4.24. In this first iteration we have the connectors and peltier connected to the back wall of the camera box. This caused a complicated matter as of disconnecting the cables when lifting the lid and getting the camera box waterproof.

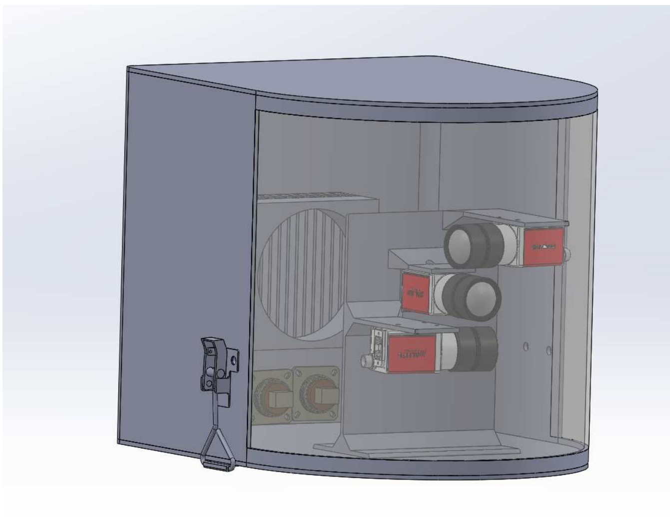


Figure 4.24: Front camera box iteration 1

We tried a dome design putting the cameras inside a dome shaped plexiglas see Fig. 4.25. But since dome shaped plexiglas is significantly more expensive in addition we would have trouble fitting everything we needed we decided not to go for this solution.

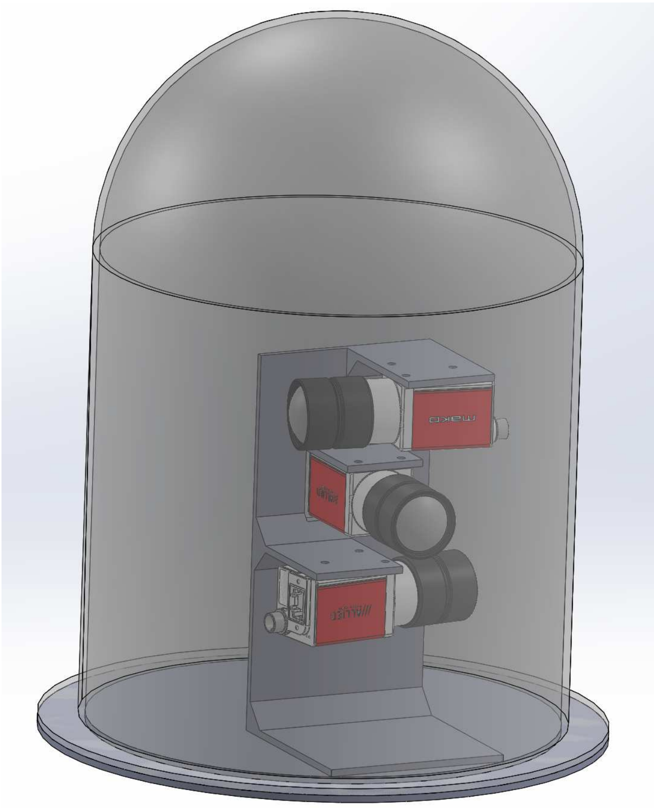


Figure 4.25: Front camera box dome design iteration 2

To get the box waterproof it was more beneficial to design the whole lid with the glass as a single part and holding it in place with force against the rest of the structure, so this was the solution we ended up going for. We will press the plexiglas on rubber seals on the bottom to secure it and make the camera box waterproof see Fig. 4.26.

Figure 4.26: Front camera box iteration 3

For our last camera box iteration we had some last minute wishes from our electrical engineering department to put the peltier directly underneath the camera stand. We modelled this in SolidWorks and measured our physical product to see if we could implement it. We also needed some space for the circuit board and the electrical components. We also decided to divide the connectors for two at each side see Fig. 4.27. We have just made a place holder for the circuit board to make sure it fits inside the camera box.

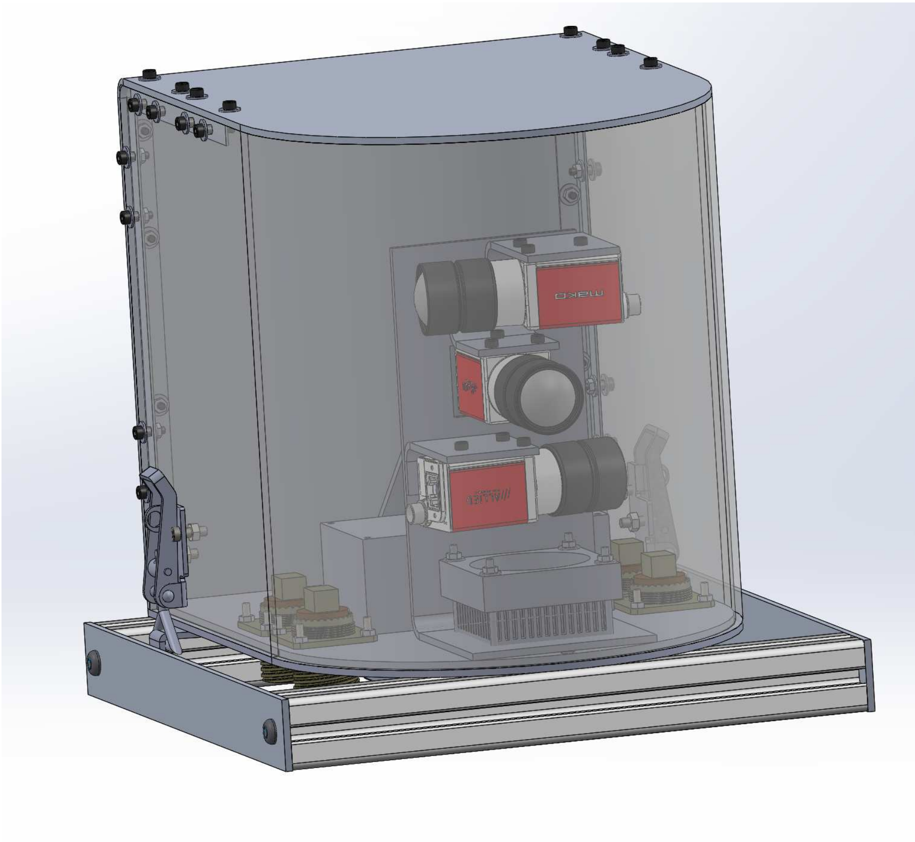


Figure 4.27: Front camera box final iteration

Lid For Front Camera Box

The lid for the front camera box is designed as three parts. You have the plexi glass which is bended in an arc and with an 3mm aluminium sheet bended to fit over it. With angle iron screwed to fit the parts together and a TEC7 sealant to make the structure waterproof. All holes will be at least 1mm larger than needed to compensate for thermal expansion in the plexiglas. We will produce the plexiglas by laser cutting it in the shape we want it, with the correct placement and size of all the holes. After we have to heat it up and create a mold for the glass and bend the heated acrylic around the mold. It is essential that the whole surface is cooled at the same rate so we don't get any warping. The aluminium lid we have to bend and after use a jigsaw to cut out the form we want, we don't have a machine to do this process for us so we have to use freehand. We were also made aware from our external supervisor that loctite to fasten the screws could cause a reaction with plexiglas causing it to crack, because of this we decided to use locknuts.

Figure 4.28: Lid for front box

Camera Stand Front Camera Box

Because of the focal length of our camera lenses is 6mm we had to do some changes in the design of the camera stand for the front camera box. The AOV originates 6mm from the camera sensor, therefore we designed a camera stand where we use this information. After reading the data sheet for the cameras it's also recommended to mount the cameras on a heat sink such as a metal bracket. With this new design we couldn't make the new camera stand as a single aluminium sheet or 3D printed see Fig. 4.29 We have to use inserts to create this new camera stand. We also decided to create an extra support arm to make the upper cameras more stable.

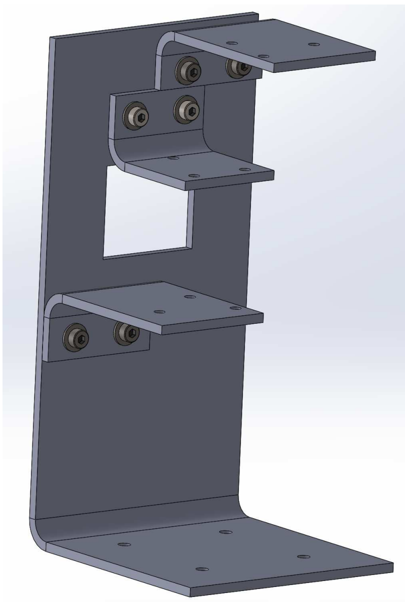


Figure 4.29: Camera mounting for front box

Enclosure Temperature Regulation

In order for the cameras to properly function in all environments here in Norway, we have decided that we will have to control the environment around the cameras. This means heating and cooling for the cameras. The electric department has chosen to regulate temperature using a thermoelectric Peltier element [97]. We will connect cooling ribs to each end of the Peltier and a small fan on the inside part of the box to get circulation inside the camera box. We have looked on the possibility to directly regulating the temperature of the cameras by connecting the Peltier to the camera stand using it as a thermal conductor. We are also if it would create a problem with condensation on the camera rig, and the cameras are not waterproof so we need to avoid getting condensation on them. This lead us to continue with our initial plan to regulate the environment around the cameras instead. We have also had continuous conversation with our team members from electrical engineering to make sure to have enough space to mount the thermal control inside the camera box.

Bottom Plate

The bottom plate of the front camera box will be two 3mm aluminium sheets connected together by screws and using silicon sealant and screws to join them together. We will also have a rubber seal at the joints where the plexi will press into to make it water resistant.

4.2.5 Rear Camera Box

The rear camera box will only contain one Mako G223 Camera with a fish eye lens with an AOV of 185°x 140°[60], [34]. The idea is to have a top cover and a bottom base. We are able to take of the cover using the same locking mechanism as the front box, an eccentric lock. This will fulfil requirement F.3.1, which enables us to access the cameras in a short amount of time. The components of the box will be attached to the bottom base, so that you're able to take off the top shell without detaching any cables or destroying components. The design process went forward with this in mind. Having carbon fiber as an option in material, we could make any shape we wanted. The box also needs to protect the camera from factors like temperature and humidity/water according to requirement F.3.2.

Iterative Design Process Rear Camera Box

In Fig. 4.30 we see the first design of the rear camera box. The thought was that it would be easy to carry and handle, but the production time would be immense if it was going to be made in carbon fiber.

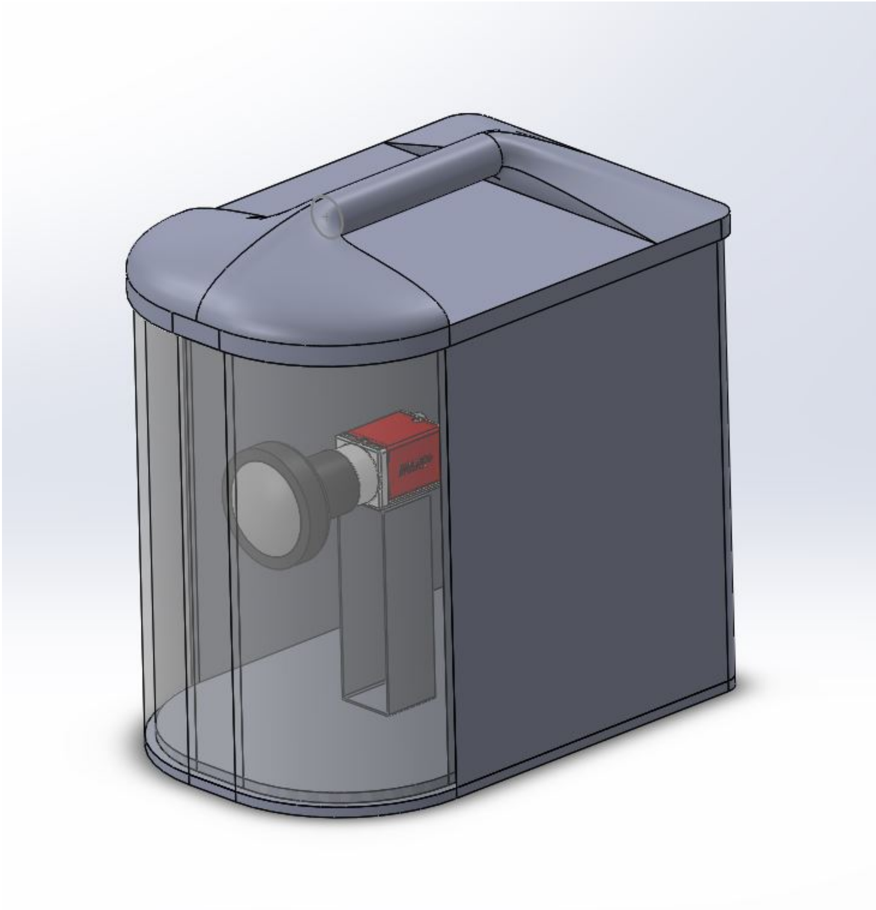


Figure 4.30: Result of the first design iteration of the rear camera box

We developed a new design in the next iteration seen in Fig. 4.31. The idea of a handle on the box was scrapped. The size of the box was reduced compared to earlier iteration.

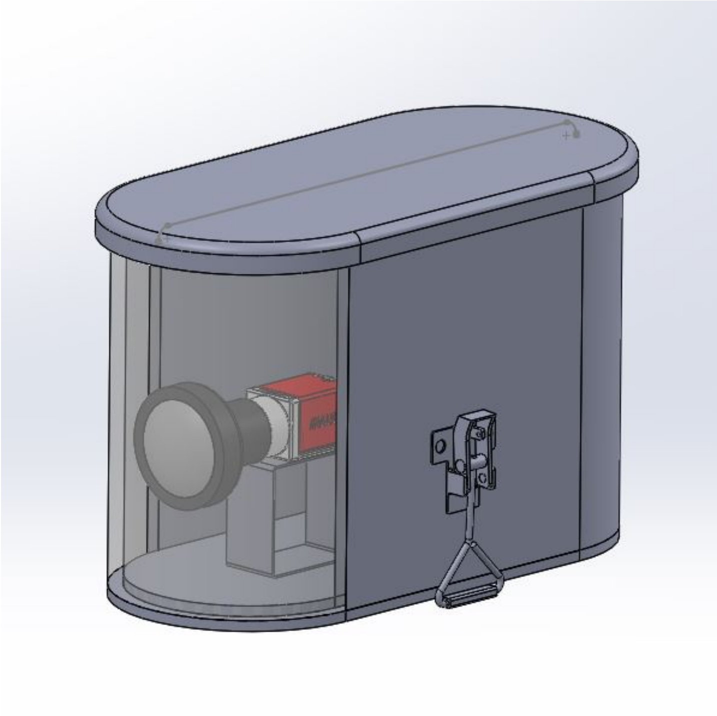


Figure 4.31: Result of the second design iteration of the rear box

In the third iteration we had an idea of shaping the glass like a dome as seen in Fig 4.32. The box is even more compact than the previous iterations. To see if the glass and the locking mechanism was functional, we decided to make a prototype of the box seen in Fig. 4.33. We connected the camera to check if it would work and confirmed that the design was functional.

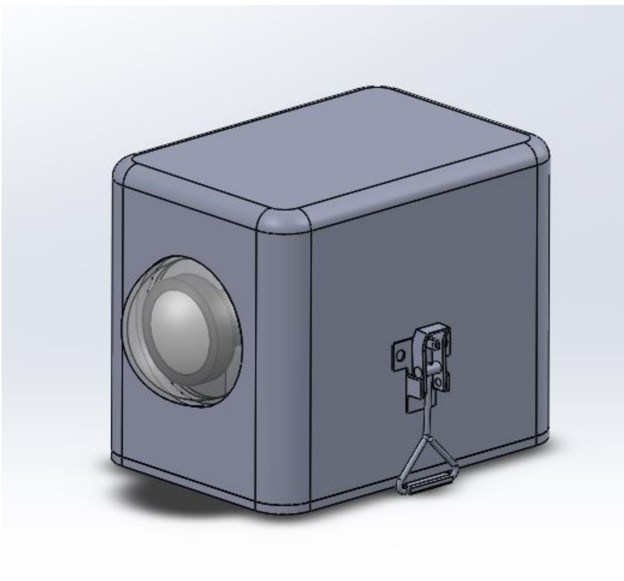


Figure 4.32: Result of the third design iteration of the rear camera box

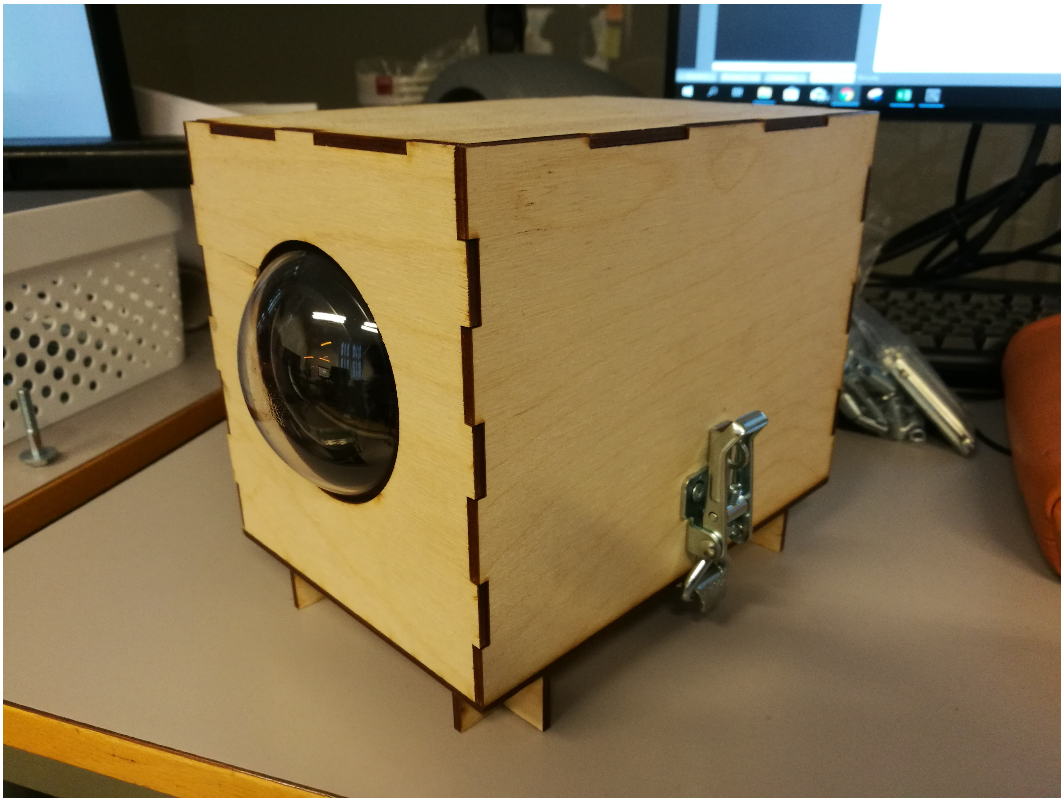


Figure 4.33: Prototype of the third iteration of the rear camera box

At this point in the process, we had decided that we wanted to develop on this design. In the fourth iteration we began shaping the box to be more aerodynamic [37]. It was also made bigger to fit the thermoelectric peltier element for heating/cooling, fans and heat sinks. In Fig. 4.34 we have developed a more streamlined shape, but a problem would occur where the eccentric lock did not have a plane surface to be mounted on.

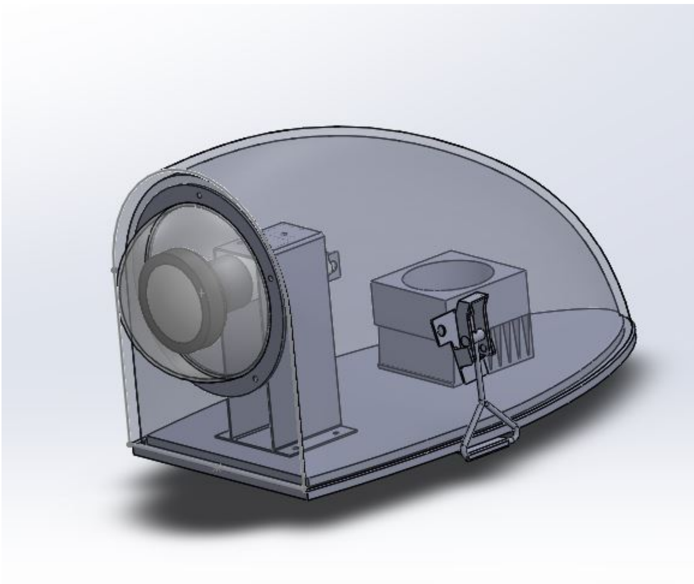


Figure 4.34: Result of the fourth iteration of the rear camera box

In the fifth and final iteration of the rear camera box we started with detailed design. We fixed the problems with the previous iteration. We added contact connections and bolts. This is the design we went on to production with. We decided that the top shell of the box would be made out of carbon fiber considering that it was the best option for material giving us a light but strong camera box.

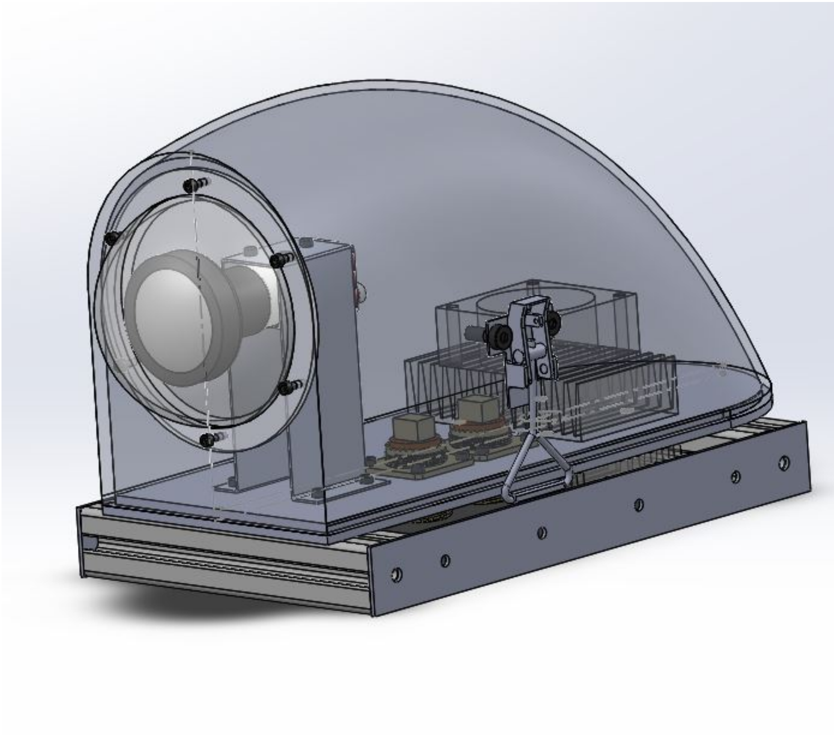


Figure 4.35: Result of the final iteration of the rear camera box

4.2.6 Bottom Plate

In the same way that the front camera box will consist of two 3mm aluminium sheets as a base, the rear camera box will do the same. Connected together by bolts and sealed tight so it is water resistant. Between the top shell and the bottom base there will be a rubber sealing strip/gasket to keep water from entering.

Glass Dome

We propose a dome shape of the glass. If the glass were to be blow moulded we need to calculate the thickness. Using the definition of density, we can use the volume to calculate the thickness of the glass before and after blow moulding [64]. The volume will be the same as we assume constant density and mass before and after forming has occurred.

$$\rho = \frac{m}{V} \tag{4.4}$$

Using equation 4.5 we can calculate the volume of a cylinder [65]. Using a thickness of 3mm and a radius of 50mm, which corresponds to the design of the glass dome, will equate to V = 23562 mm^3.

$$V = \pi r^2 h \tag{4.5}$$

We can then calculate the thickness of the glass assuming constant volume and uniform distribution of the mass. Using equation 4.6 we end up with a thickness of approximately 1,54 mm [65].

$$\begin{aligned} V &= \frac{2}{3}\pi(R^3 - r^3) \\ r &= \sqrt[3]{R^3 - \frac{V}{\frac{2}{3}\pi}} \end{aligned} \tag{4.6}$$

As the dome is not facing the driving direction, the need to look any further into the problem is not necessary. The stresses in the glass will be low so the we deem 1,54mm thickness is more than enough for shielding the camera.

4.2.7 Aerodynamic Simulation

We did a wind tunnel simulation using SolidWorks. SolidWorks does not have a solver for drag coefficient. So we have to use the equation solver to create a equation to solve for the drag coefficient for our system see Fig. 4.36 and Fig. 4.37. According to our simulation we get approximately a drag coefficient of 0.40 for the front camera box.

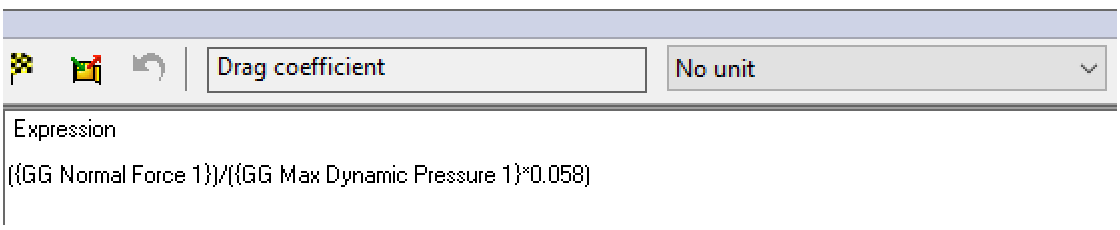


Figure 4.36: Drag coefficient equation

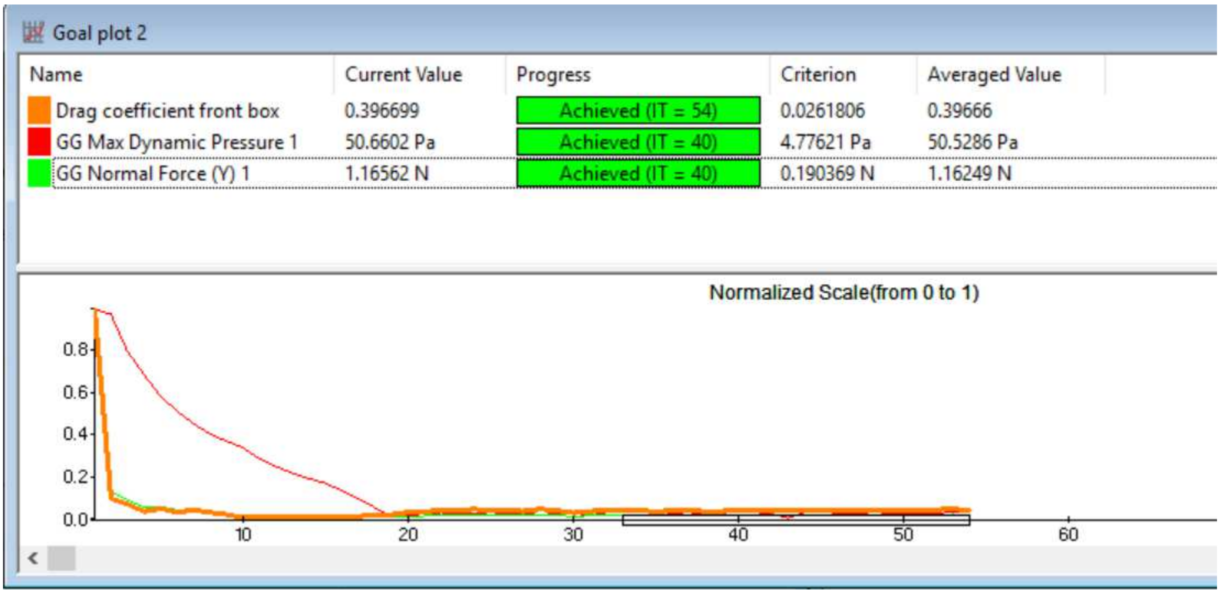


Figure 4.37: Drag coefficient calculation for flow simulation front camera box

From the flow analysis we calculated the drag coefficient for the rear camera box to be lower than that of a car. In Fig. 4.38 we see that the drag coefficient calculated to be about 0.22.

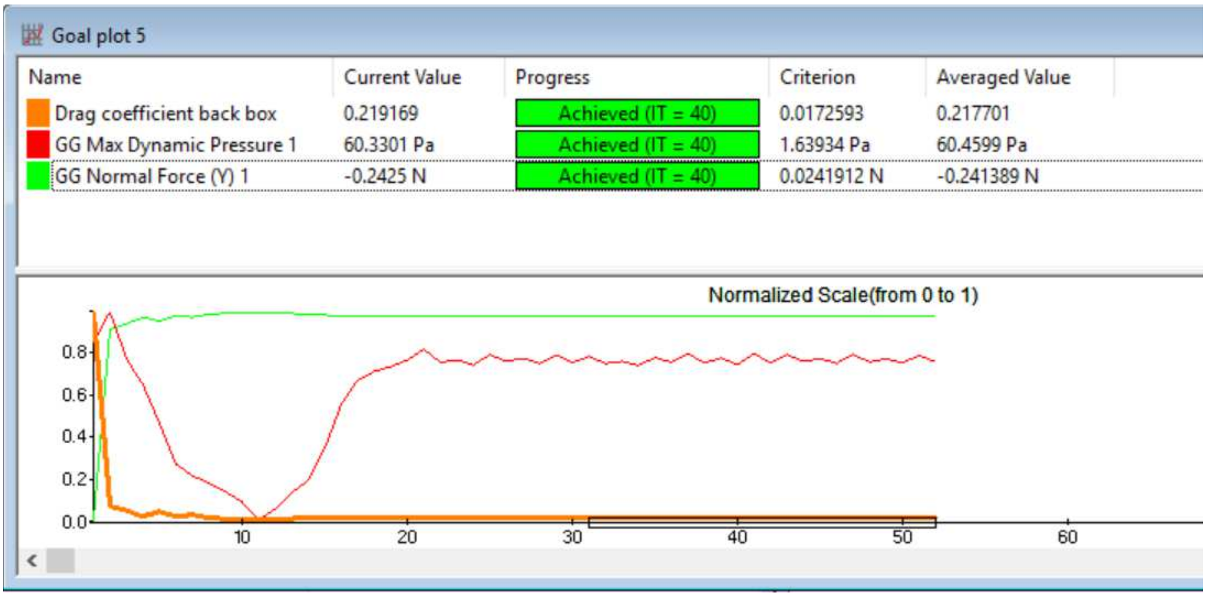


Figure 4.38: Drag coefficient calculation for flow simulation rear camera box

In our simulation we also can also analyse the wind pressure on the camera boxes and the maximum force applied. We ran tests with a wind speed of 25m/s and got the results 42N maximum force and 0.1013MPa maximum pressure see Fig. 4.39 and Fig. 4.40.

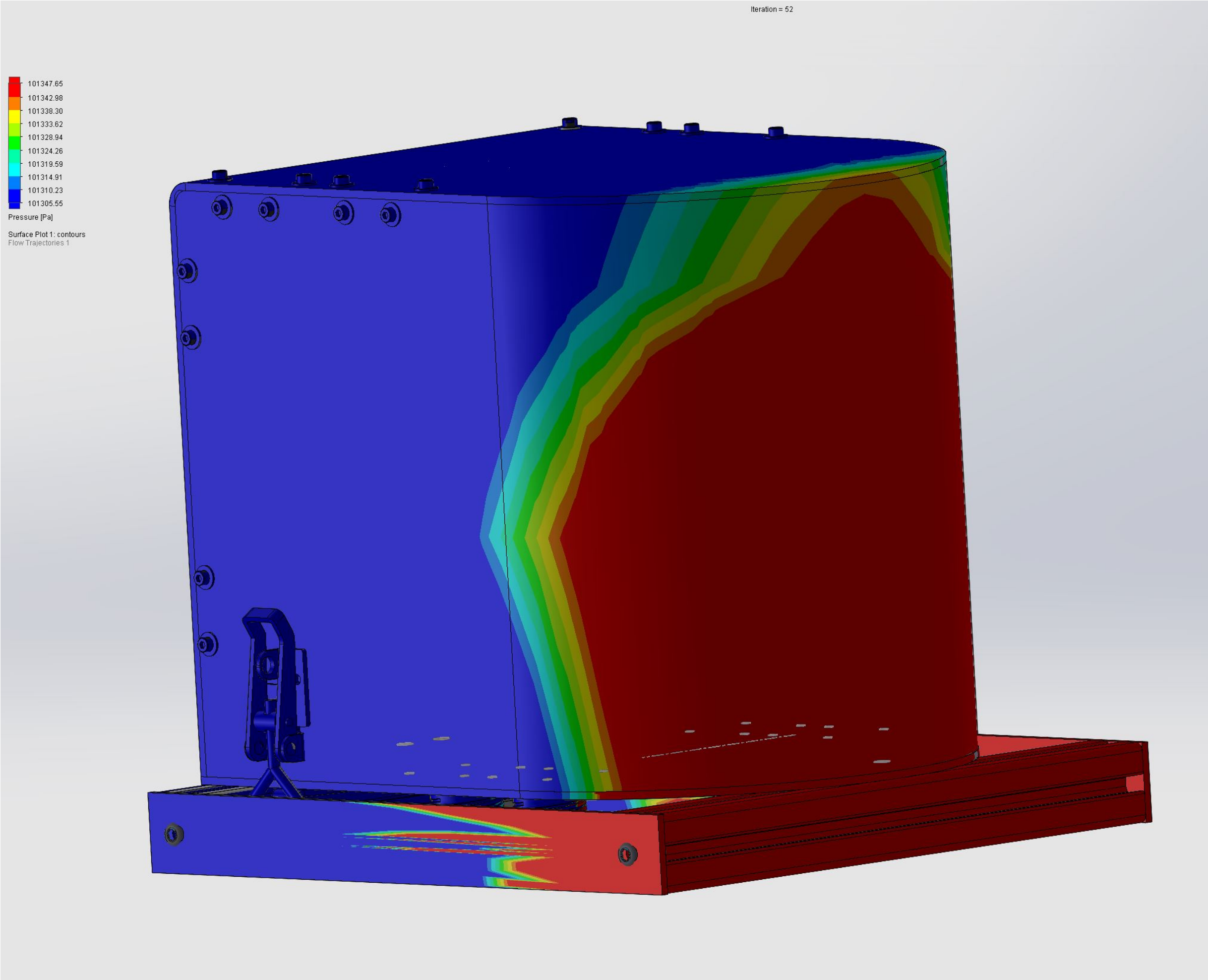


Figure 4.39: Pressure simulation at 25m/s for front box

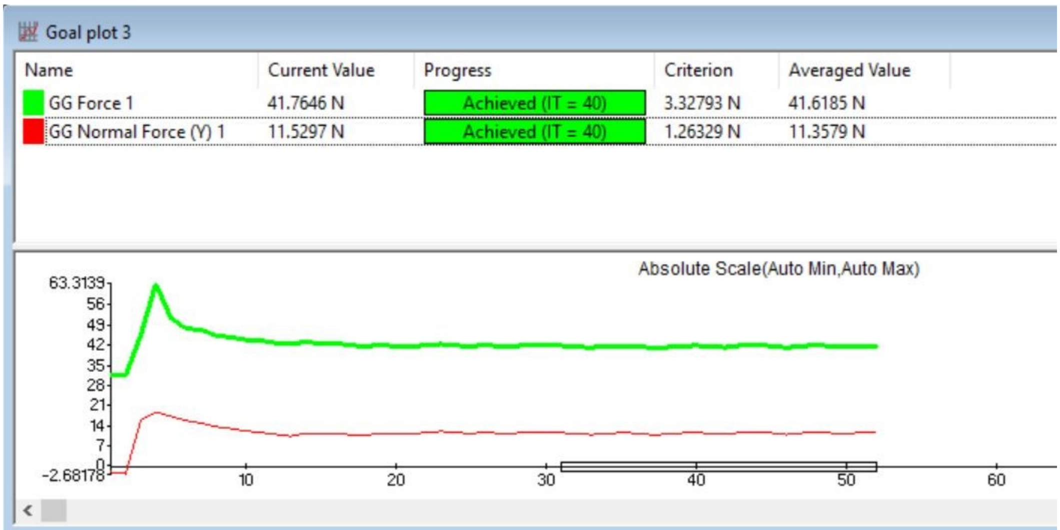


Figure 4.40: Maximum force applied the front box

To see the amount of pressure we are dealing with we decided to do computer aided analysis of the rear camera box system the same way as the front camera box.

Running the simulation with a speed of 25 m/s we ended up with a maximum pressures around 0.102 MP a. This pressure is where the box meets the air as seen in Fig. 4.41.

Figure 4.41: Pressure analysis from flow simulation in 25 m/s winds

From such an analysis we also get an estimate for the flow trajectory of the wind around the object. This will show us how the wind moves around the object, if we have any troublesome spaces where wind will get trapped. In Fig. 4.42 we see that the arrows indicate that the wind flows smoothly around the object.

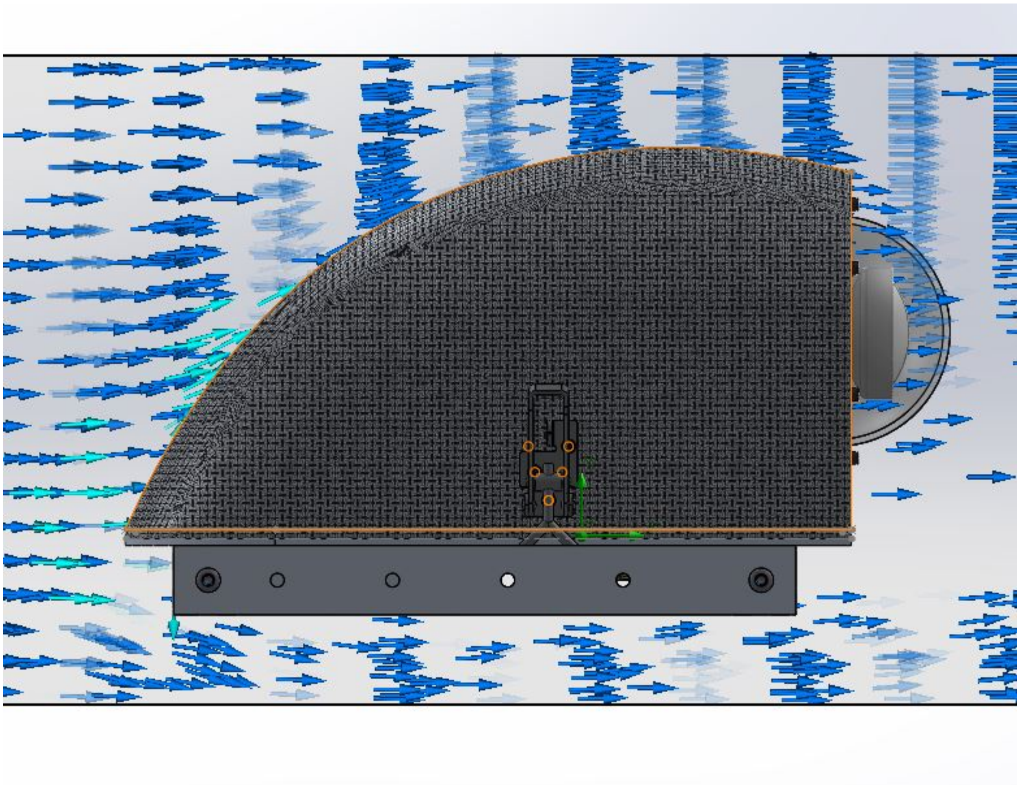


Figure 4.42: Flow trajectories from 25 m/s winds

4.3 Vibration Attenuation Propositions

Since there was a drone available with vibration attenuation, we decided to test the system for our IMU. The vibration attenuation for drones were smaller than the one needed for our IMU, which is why we had to design a new ones using the same rubber dampener see Fig. 4.43, this model is 3D printed using PLA because we don't need any specific material properties. Other than the vibration dampening in the rubber.

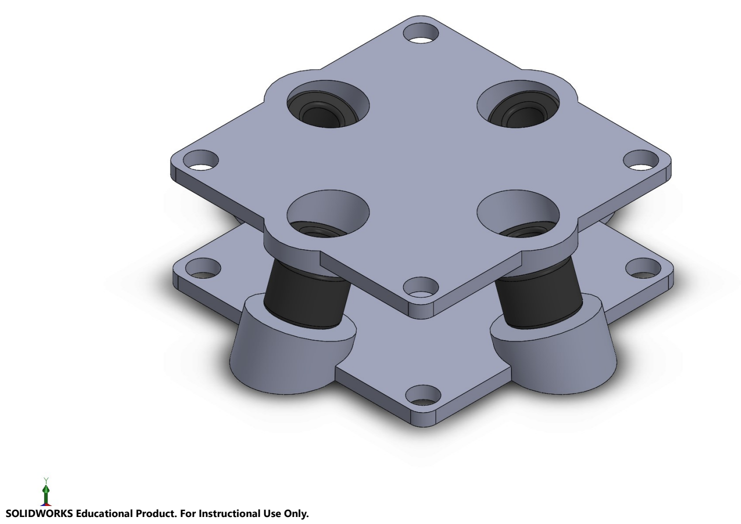


Figure 4.43: IMU attenuation

To check if we're fulfilling the requirements we made a test for the IMU see Table B.6. Our first test T.1.1 was inconclusive because of uncertainty of the mounting in the car. It was hard to tell if the IMU was drifting or if it was moving around inside the car, due to a person was holding it in place. This in return adds a whole lot of other complications and irregularities, which is why we deemed the test inconclusive.

For the second and third test we designed a testing rig for mounting the IMU inside a Nissan Leaf see Fig. 4.44. We dimensioned the testing rig so it would fit in the armrest between the driver and passenger seat. The thought was to have a tight fit so the IMU sat in place and didn't have any play. While this mounting method worked the second test was inconclusive because the cable connection was mounted to the side instead of straight ahead. This made us unsure if the slight drifting we had was a result of us dragging the connection, this test then was concluded as inconclusive see Table B.7. The third test we did we made sure the connection was mounted straight ahead. In this test we only had one problem which was drifting when driving over some speed bumps. We then concluded that this mounting solution was failed see Table B.8. Causing us to have to design a new vibration attenuation.

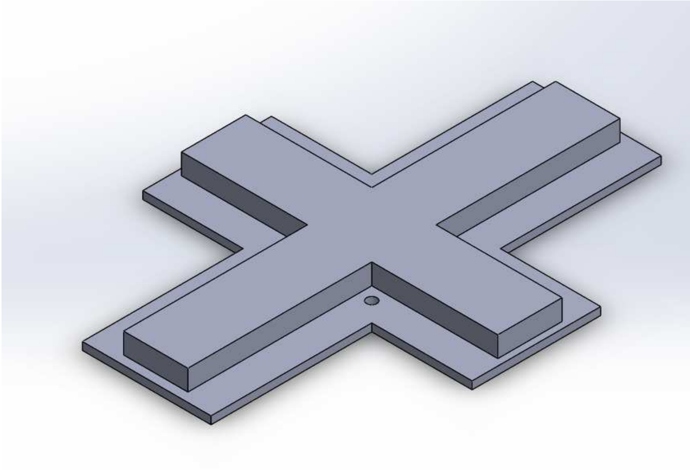
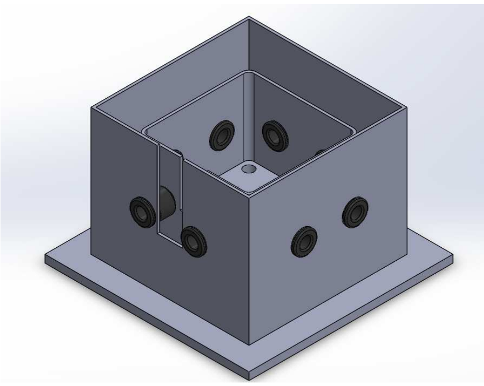
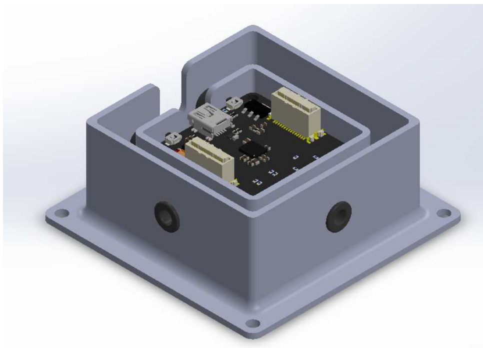


Figure 4.44: IMU testing rig

Since the drifting with our previous design occurred when we drove over some speed bumps. We decided to create a new design that is less rigid and has more dampening effect. Our first thought was using the same design as previous only having the dampening mounted in the y-axis instead of the x-axis and so we came with the designs Fig. 4.45a and Fig. 4.45b. We thought Fig. 4.45a might be too rigid too because we had two rubber balls on each side. So we scrapped that idea and designed Fig. 4.45b. This model is 3D printed and is a lot less rigid than the previous.



(a) IMU vibration attenuation V3



(b) IMU vibration attenuation V4

Figure 4.45: 3D printed design for vibration attenuation V3 and V4

In our forth test we deemed it as failed because we had too much drifting in the IMU see Table B.9. We did some extra research because we found it weird that we got no improvement except when we drove over speed bumps. We decided to turn on the magnetometer to get external reference for the IMU positioning. After doing our fifth test see Table B.10 we noticed a considerable improvement in drifting by the IMU. We had slight drifting at start up of the test run but since the IMU behaved normally after we believe this is because the IMU is initializing at startup. We have to do another test with the same setup and let the IMU initialize on the next test run to see if our design worked. This design worked considerably better than the previous because the drifting in the IMU was so small it's almost unnoticeable. One more test is necessary to ensure our design fills the requirements.

The sixth test was done with the magnetometer turned on and with an IMU stack instead of a single IMU. This test was accepted see Table B.11. We had no drifting at all in this test and the dampening was sufficient enough. We noticed that since we used a IMU stack and it makes the structure a little wobbly and some vibration at the top of the IMU stack. Therefore we had to design a higher structure to support the IMU stack. As you can see in 4.46 we designed a larger structure for the IMU stabilization so we reduced the vibration at the top of the stack. Since our design now is accepted we decided to test in a fossil fuel car. With this we still had no drifting and everything was working according to plan see B.13. So our final design is now ready.

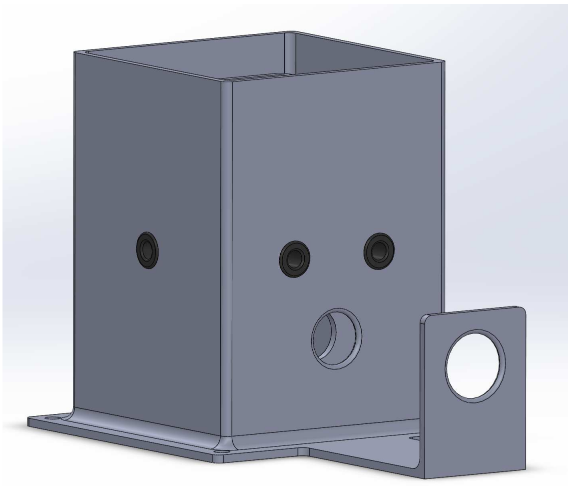


Figure 4.46: IMU vibration attenuation V5

4.4 Power System Propositions

There are two main ways of designing the power system. The first is to keep using an inverter to power the computer. The second way is to totally remove the inverter and run everything off DC. There are also variations on both of these solutions with their own benefits and draw-backs.

4.4.1 Potential Power Systems

Power Inverter The simplest solution is to keep the general system architecture already implemented and simply replace the inverter. In this solution the computer is powered by an inverter that produces minimal interference and a *true* sine wave. While everything that is run off of DC revives its power directly from the batteries. The general architecture of this system shown in Fig.4.47. The main advantage to this system is that it is the easiest to implement. Another advantage is that it has a relatively high efficiency, even though we go from DC to AC and back to DC inside the computer. One disadvantage of this system is that the low voltage electronics are all connected to the batteries without much separation, all the while the inverter could be causing havoc on the DC power line. Even if the the output of the inverter is nice and stable that does not mean that the input is.

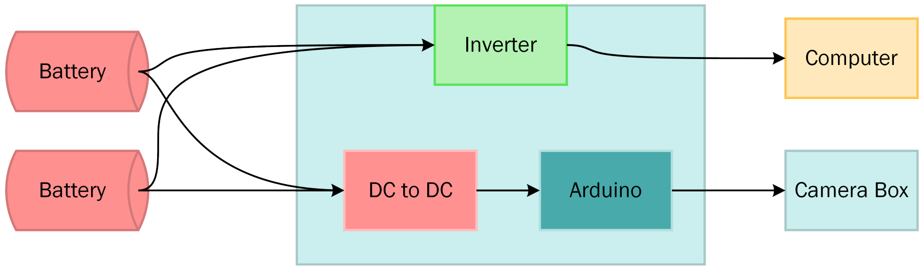


Figure 4.47: AC solution 1 - architecture

Another solution that implements the inverter is to have all the power need be extracted from the AC output of the inverter. This means that the only thing connected to the batteries is the inverter. The general architecture of this system quite similar to the previous and is shown in Fig.4.48

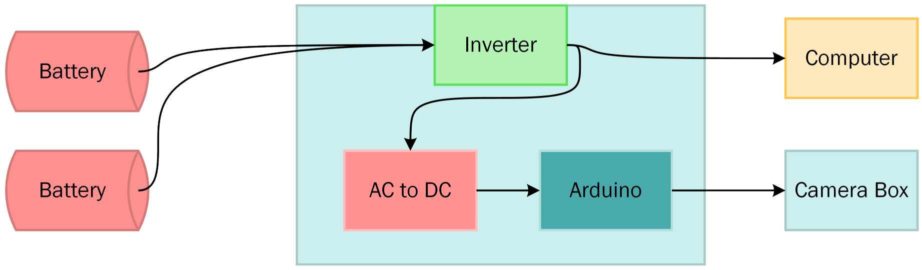


Figure 4.48: AC solution 2 - architecture

All DC One possible solution would be to ignore the AC part of the system and only use DC to DC converting for the power system. Why this would be an advantage is because of the weight and size would be reduced massively. In theory, we could get rid of the whole external power case and connect the computer directly to the battery controller, see Fig 4.49. The challenges with this is the need of high wattage for the system as a whole and one trade-off of the DC to DC system would be efficiency and modularity [70]. The reason we will need such high wattage for the system is due to the high-end graphic card that the computer needs to run the Argos system. Since this system will continue to be under development further on, modularity is something that may be very high prioritised.

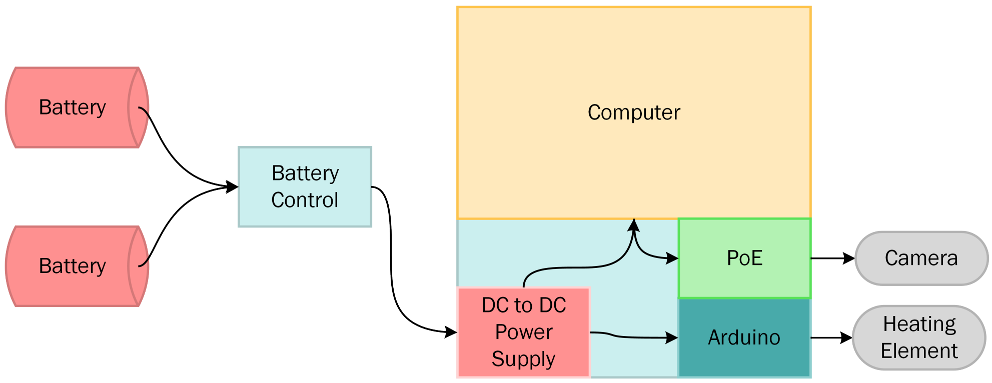


Figure 4.49: Illustration of the DC to DC power system inside the computer

To convert a signal from DC to DC one often takes the use of switched-mode regulation. With the use of Field Effect Transistors (FETs) you switch the input on and off in necessary duty cycles to charge either inductors, transformers or capacitors. The cycles variates on different converters, but they mostly use the same principles. The duty cycle determines the voltage level of the output compared to the input, but also helps determine the value of the inductor [72] [71]. Something that makes this circuit intricate, is that you want a shortest possible duty cycle, so that its possible to create the smallest possible converter with minimal ripple. The problems with creating such a low duty cycle is that the trade-off are both efficiency and heat [70].

The reason for modularity might become a problem, is because when the DC to DC system will be designed, we want to have the size and weight as low as possible. This causes the wattage to be as low as necessary and adding additional components would become a challenge. In addition to this, using a pre-fabricated inverter, which would be the case in DC to AC system, adding a new part to the system would just be as simple as plugging it into the socket as one does at home. While for the DC to DC system, each new part of the system might have different voltages that would have to be accommodated for.

Power Consumption Mitigation Next unit of computing (NUC) is one other option that we have considered to simplify our power solution since it is a small and powerful computer that runs on DC current. This is a system that runs mostly on 19 volt DC, which works extremely well when a small sized, but processor heavy computer is needed.

The problem we face when considering a NUC, is that the Argos system requires high graphical performance that the NUC as of Feb. 2019, it cannot provide. This would lead us to need a separate graphic card that also would require an additional power supply. This supply would provide a 12 volt signal with more than 300 watts to run the graphic system. Also the NUC has only up to two network outputs, which would create the need for a switch so the cameras get the Power over Ethernet (PoE). The system is also very new, so the price for these are also something to consider.

Maybe in the future this would be a great solution for the system, but that would not be until it can support such graphics as the system needs.

Informing the customer To find how we were to find a final solution for the system, we needed to be sure that we knew what the customer wanted. We decided to create a decision matrix as seen in Fig. 4.50. This would make it easier to understand the options we had developed through research and to show what that would be the best way forward. By estimating the values for each possible system with gathered information to back it up, we managed to weight each system in a way that it would show what we believe to be the best way to move forward. We then proceeded to contact our customer and find out what that would be most important for the end product so that we could have a guideline.

Evaliuation Criteria		Reference Design		DC-DC External		DC-DC Internal		DC-AC back	
	Priority - 5 is high	Rating	Weighted	Rating	Weighted	Rating	Weighted	Rating	Weighted
Modularity	4	0	0	-2	-8	-3	-12	0	0
Physical size	2	0	0	4	8	5	10	0	0
Weight	2	0	0	5	10	5	10	0	0
Power efficiency	4	0	0	-2	-8	0	0	1	4
Moneatry Price	2	0	0	-1	-2	-4	-8	2	4
Time to implement	4	0	0	1	4	2	8	0	0
Life time	2	0	0	1	2	2	4	1	2
Availability of parts	2	0	0	-2	-4	-2	-4	0	0
Ease of instalation	5	0	0	4	20	5	25	0	0
Total Score		0		22		33		10	

Evaliuation Criteria		DC-AC front		Hack PC PSU		Create		UPS	
	Priority - 5 is high	Rating	Weighted	Rating	Weighted	Rating	Weighted	Rating	Weighted
Modularity	4	4	16	-3	-12	-3	-12	4	16
Physical size	2	-1	-2	4	8	3	6	-2	-4
Weight	2	-2	-4	5	10	4	8	-5	-10
Power efficiency	4	1	4	4	16	4	16	3	12
Moneatry Price	2	2	4	4	8	2	4	-20	-40
Time to implement	4	5	20	-3	-12	-5	-20	5	20
Life time	2	1	2	-2	-4	-4	-8	0	0
Availability of parts	2	3	6	-3	-6	-5	-10	-1	-2
Ease of instalation	5	-1	-5	5	25	5	25	3	15
Total Score		41		33		9		7	

Figure 4.50: Decision matrix for power supply system

4.5 Battery Controller

Since we want to estimate the battery state of charge, we need to measure the voltage of each battery separately. If the batteries were to be parallel coupled as in the legacy system this could become a problem. For us to measure each of the batteries separately, the batteries would have to be parallel coupled within the battery monitoring circuit. If the batteries were to have different state of charge, the batteries would try to balance out the voltage difference across each of the batteries, which could induce a high-power surge from one battery to the other [77]. At best, this would cause a small spark and maybe some heat dissipation. If the voltage difference were to be much higher, this could melt the cables and destroy the batteries. To avoid the problem with voltage differences across the batteries, an implementation of a battery hot-swapping was requested, but not as a top priority. At first glance, switching from one battery to the other does not seem very hard to do. All that is necessary is a switch. But if the system should be running constantly, many problems can arise. To switch power on and off with automation, there are two main components that work extremely well, relays and MOSFETs.

Relay Relay is a combination of an electromagnet and a mechanically switch. When the electromagnet is activated, the switch can open, close or switch from one pin to the other, depending on the relay. While an electrically operated switch like the relay is very suited for an operation like switching batteries, the relay has its drawbacks. Since the switch is mechanical, there is a chance that the switch can differ from open to closed when introduced to vibration. Automotive relays do exist and are very reliable, but with the slight chance of switching when not intended, we would try our best to avoid the usage of them.

MOSFET A metal–oxide–semiconductor field-effect transistor (MOSFET) is a transistor often used for switching operations and in difference with the relay, has no moving parts. When switching a N-Channel MOSFET from on to off, instead of disconnecting totally as the relay, the transistor goes from closed, to partially open, to fully open. The state of the transistor is decided by the voltage at the gate V_G, compared with the voltage at the source V_S. For the transistor to go from closed to open, the voltage between the gate and source, V_{GS}, has to exceed the threshold voltage $V_{GS_{(Th)}}$.

$$V_{GS} = V_G - V_S \geq V_{GS_{(Th)}} \tag{4.7}$$

As shown in the equation above, the gate-to-source voltage V_{GS} has to be equal or grater than the threshold voltage$V_{GS_{(Th)}}$.

The transistor is closed if

$$V_{GS} < V_{GS_{(Th)}} \tag{4.8}$$

4.5.1 Switching Protection

Since the system shall be able to operate during a switch from one battery to the other, some kind of control circuit will be necessary. If we were to open and close the switches simultaneously, the power could turn off for some milliseconds, which could result in the computer shutting down. Another alternative, if we were to use MOSFETs, the resistance could increase for a millisecond and the transistors could explode. Since neither of these options were tempting, a circuit had to be implemented to avoid this situation. We came up with two different alternatives to solve this problem.

Diodes Diodes would be a very easy solution that would in theory allow both the batteries to be active at the same time. However, diodes induce a voltage drop of 0.7 volts, which could result in problems when the circuit has a large load. Another option would be to only have the diodes active while the switching is taking place. If the circuit were to be under large loads and high currents, the slight voltage drop for a couple of milliseconds should not affect the circuit to much. Having an additional line for the diodes would increase the number of switches naturally.

Battery Controller Integrated Circuit A battery controller Integrated Circuit (IC) allows the voltage to pass trough when power is connected. However, implementing additional components allows the IC to allow current to flow from one side to the other, but not the other way. This behaviour is very similar to a diode, but without the voltage loss. Some of the ICs also allow for surge protection in the circuit, which would protect the remaining part of the controller. While this sounds very great, the ICs only work with MOSFETs.

4.6 Estimating The Battery State of Charge

In order to know the state of charge of the batteries we need to know how much charge is being drained from the batteries [85] [86].

$$Q_{battery}(t) = Q_{battery}(t_0) - Q_{used}(t) \tag{4.9}$$

The flow of charge is the electric current. The current is defined as the derivative of the charge with respect to time [79].

$$i = \frac{dq}{dt} \tag{4.10}$$

We then integrate both sides of the equation to get the total charge

$$q = \int i \, dt \tag{4.11}$$

We can see now that in order to measure the charge we need to integrate the current emanating from the battery.

An operational amplifier (op-amp) can perform various mathematical operations depending on how it is configured. One of those operations is integration. The op-amp symbol is shown in Fig.4.51. Op-amps are fairly simple only have a few set of properties that defines them [80]. The first is that it has a very high input impedance. What that practically means is that virtually no current follows into the plus and minus terminals. Another property is that the voltage at the plus and minus terminals is identical. The most basic integrator is the inverting integrator. It produces an output that is the inverse of the integral of the input current. The problem with this, is that it requires a negative voltage source to produce the negative output voltage. Another problem is, because the output is negative, that it becomes much harder to measure using a micro-controller. Most micro-controllers can measure between 0V to 5V. Because of these problems we decided to use a non-inverting integrator. The non-inverting integrator has the same function as the inverting integrator except that it has a positive output voltage.

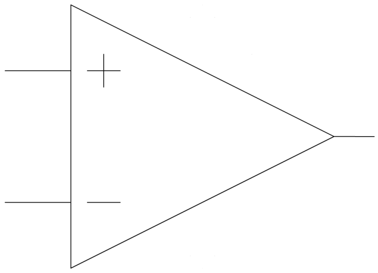


Figure 4.51: Op-amp symbol

Fig.4.52 is the schematic for the current integrator we designed. In order to derive

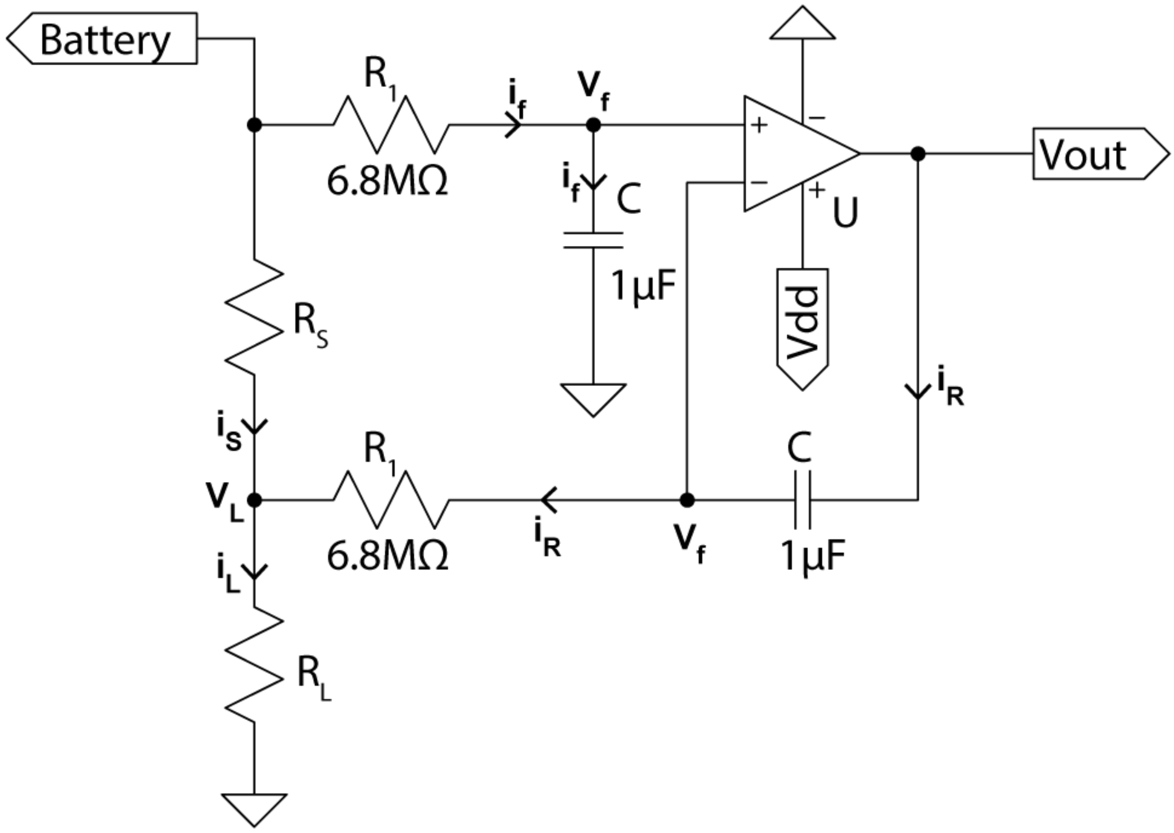


Figure 4.52: Coulomb counter op-amp

the function of the op-amp we need three equations. The first equation is the current that goes in towards the positive terminal (i_f) . To calculate this current we use Ohm's law.

$$i_f = \frac{V_s - V_f}{R_1} \tag{4.12}$$

Where V_s is the source (battery) voltage, V_f is the voltage at the positive terminal and R_1 is the input resistance to the integrator. Since (nearly) no current enters the positive terminal all of the current must go through the capacitor (KCL [81]). The current through a capacitor is the capacitance of the capacitor times the derivative of the voltage over the capacitor with respect to time [81].

$$\frac{V_s - V_f}{R_1} = C\frac{dV_f}{dt} \tag{4.13}$$

The second equation is derived from the negative terminal. This side is very similar excepts it incorporates feedback from the output.

$$i_R = \frac{V_f - V_R}{R_1} = C\frac{d(V_o - V_f)}{dt} \tag{4.14}$$

$$\frac{V_f}{R_1} - \frac{V_R}{R_1} = C\frac{dV_o}{dt} - C\frac{dV_f}{dt} \tag{4.15}$$

The third and final equation is the current that flows through R_s, which is, using Ohm's law

$$i_s = \frac{V_s - V_R}{R_s} \tag{4.16}$$

If we then substitute in the first equation into the second and solve for $\frac{dV_o}{dt}$ we get

$$\frac{V_s - V_R}{R_1 C} = \frac{dV_o}{dt} \tag{4.17}$$

Now we can substitute in the third equation

$$\frac{R_s i_s}{R_1 C} = \frac{dV_o}{dt} \tag{4.18}$$

Now the final step is to integrate both sides of the equation in order to express the output voltage of the op-amp as a function of time.

$$V_o = \frac{R_s}{R_1 C}\int i_s \; dt \tag{4.19}$$

or more precisely

$$V_o(t) = \frac{R_s}{R_1 C}\int_{t_0}^{t} i_s(t) \; dt + V_o(t_0) \tag{4.20}$$

We can also write this as a function for charge used

$$q_{used}(t) = \frac{V_o(t) R_1 C}{R_s} \tag{4.21}$$

we can then use this function to estimate how much charge is left in the battery

$$Q_{battery}(t) = Q_{battery}(t_0) - q_{used}(t) \tag{4.22}$$

There are some drawbacks of this design. The first is drawback is that the current that is being integrated (i_s) is technically not all of the current that is being drawn from the battery. This is not a problem however because i_s is the vast majority of the total current. i_s is in typically in the range of around $60A$ to $150A$, while all of the other measuring currents are in the range of $2\mu A$ to $20\mu A$. That is about a seven orders of magnitude difference in scale. So the error from this can be considered to be zero. The reason that the difference is so large is because the input resistance of the integrator (R_1) is about 7 orders of magnitude larger than the sensing resistor (R_s).

$$\frac{R_1}{R_s} \approx 10^7 \tag{4.23}$$

With an ideal operation amplifier, a design like this would be sufficient for the Coulomb counter. However, no operation amplifier is ideal and the voltage difference across R_s is so small, it can be hard for the operation amplifier to detect the difference. To circumvent this, it is possible to implement a instrumentation amplifier as seen in Fig 4.53.

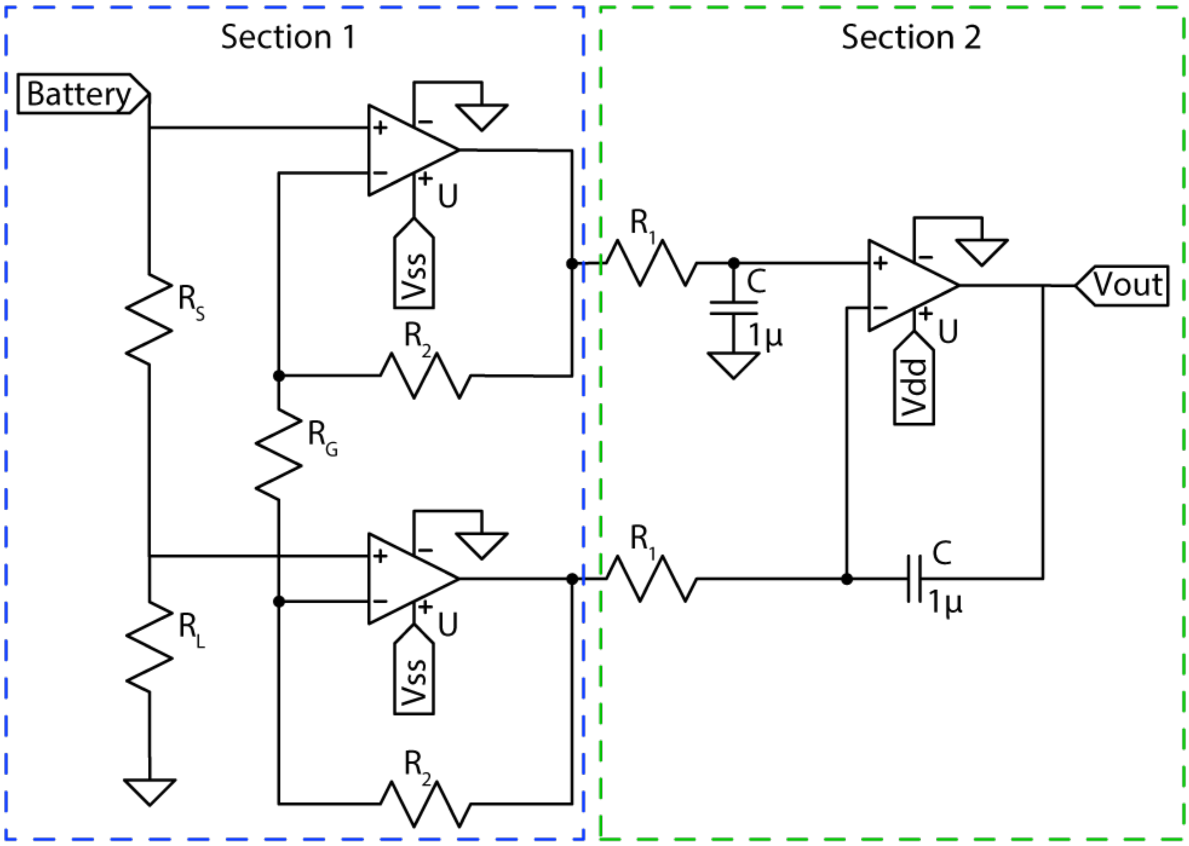


Figure 4.53: Coulomb counter with instrumentation amplifier

A instrumentation amplifier amplifies the difference between each of the inputs, given that the output is below the source voltage for the operation amplifier(Vss). This is a normal application for sensing small voltage differences from sensors, since a good operation amplifier normally needs around 1mV difference to output a signal. The amplification for the first section can be expressed as followed:

$$A_1 = 1 + \frac{2R_2}{R_G} \tag{4.24}$$

4.7 Controlling the Camera Temperature

Due to the temperature constraints of the cameras the thermal system must have the ability too cool when the air is too hot. However the outside temperature can also be too cold, so the thermal system also has to be able to heat the air inside the camera enclosure. Which means that the thermal system has to be able to both increase and decrease the temperature of the camera enclosure. This creates a dilemma. Either we have two separate systems that work together to increase or decrease the temperature, or we have one system that can do both.

4.7.1 Choosing the Right Temperature Control System

There are three main ways of heating using electricity: Joule heating, infrared heating, and induction heating. These heating systems would be part of a system that has one heating element and one cooling element that would work independently of each other.

Joule heating Joule heating (a.k.a. ohmic/resistive heating) is by far the simplest way of heating and the most common. Incandescent light bulbs, toaster ovens, hair dryers, and space heaters are a few among many devices that use Joule heating. A Joule heating element is simply a conductor that becomes hot when a current is passed through it [93]. The heat radiating from joule heating elements heat up the surrounding air. The power of heating from joule heating is based on the current and the resistance of the conductor.

$$P \propto I^2 R \tag{4.25}$$

where P is power, I is current, and R is resistance.

Infrared heating The biggest infrared radiation (IR) heater, that everyone has experienced, is the sun. IR heating, as the name suggests, works on the principle of radiating infrared waves/light to heat up people and objects. IR consists of the parts of the electromagnetic spectrum that has longer wavelengths than visible light, but still has shorter wavelengths than radio waves. Since IR is electromagnetic radiation IR does not need a medium like air to propagate. IR can pass through air without the air absorbing too much of its energy. This makes IR very good when remote heating is required, but heating of the air is either not possible or undesired. Infrared light can also be reflected, and can therefore be focused to heat specific areas [94].

The distinction between joule heaters and IR heaters can become a little muddled, because the both types of heaters produce, at least, a little of the other's type of heating. Joule heaters produce some IR heating and IR heaters produce some joule heating.

Induction heating Induction heating produces works in a vastly different way than the other two methods mentioned above. Induction heating works placing a magnetic material (like steel) inside an alternating magnetic field. The magnetic field is made by running a large high frequency current through an inductor (coil) [95]. When the magnetic material is placed inside the magnet field something called an eddy current is produced inside the material [96]. The eddy current flows through the electrical resistivity of the material producing a power loss resulting in heat generation.

$$P = RI_{eddy}^2 \tag{4.26}$$

Induction heating is normally used to perform very localized heating in metals, often to extreme temperatures. It can be used to heat non-magnetic materials, but only indirectly. The induction stove top is the type of induction heater most people are familiar with.

Cooling something is much harder than heating something. That is why there are so few options to choose from.

Air cooling Air cooling is by far the easiest and most common cooling method. It works by replacing the warm air surrounding a hot object with new cooler air from somewhere else. This new cool air will absorb some heat from the hot object and then be replaced by more cool air to continue the process. The major flaw in this method of cooling is that it is highly dependant on the ambient temperature of the air being blown onto the object. This method only brings the temperature of the hot object closer to the ambient air temperature. If the ambient air temperature is too hot or too cold, this method will not work. This method is also very efficiency of this method is also highly dependent on the total surface are of the object. The total surface area of the object can be increased however, with the help of cooling fins. This method usually requires fans to move the air.

Liquid cooling Liquid coolers transfer energy by circulating a liquid from a warm spot too a cooler one. The liquid is typically either water or oil. These liquids absorb much more energy by volume than air, which make them excellent for cooling purposes. Liquid cooling is typically used to cool specific components that produce lots of heat. Liquid cooling could, in theory, be used to match the air temperature of two thermal systems by having two radiators on inside each system. This set up would take energy from the air of the hotter system and move it to the cooler system. The energy transfer would happen without the air from one of the systems coming in contact with the other allowing one or both of the systems to be enclosed (air tight). This method requires pumps, a suitable liquid, and tubing/pipes. It also usually requires radiators and fans as well.

Having a bidirectional system would allow us to only implement one system, removing the need to make two separate systems work together.

Reversible heat pump Heat pumps work in a somewhat similar fashion to a closed loop liquid cooling system, except that that the heat pumps can make the temperature difference larger as opposed to smaller. It does this by utilizing pressure to evaporate and liquefy a solution. When the solution evaporates it absorbs energy from its environment. When the solution condenses into a liquid it emits energy. The heat pump utilizes this to transfer energy from one system to another. Utilizing heat pumps is the most common way to make large temperature differences. Some common examples are: refrigerators, freezers, and air conditioning units. The biggest downside to heat pumps is that even the smallest units are quite large. They are reasonably power efficient however.

Thermoelectric Thermoelectric Cooling (TEC) devices are by far the newest technology mentioned here. Thermoelectric units work by using the Peltier effect to transfer heat from one side to the other by applying a voltage to the terminals [97]. The units are named after the Peltier effect and are therefore called Peltier devices. Peltier devices have the same functionality as heat pumps but in a much smaller form factor [98]. Besides the much smaller size, Peltier units also do not have any moving parts (steady-state). This also means that the orientation of the units does not matter. Another property that Peltier units have is that heat is transferred very quickly. This means that the temperature can be very precisely controlled [99]. All that is needed to change the direction of the thermal flux is to reverse the polarity of the unit. This could be as simple as swapping the leads. Peltier units are ideal when size and orientation are the biggest priorities. The largest downside to Peltier units is that they are not very power efficient, although this is getting better as the technology progresses.

Choice After careful consideration we decided the key criteria for the thermal systems would be:

- Compact size
- Lightweight
- Rapid cooling and heating
- Not being affected by physical displacement (vibrations / movements)

Thoughtfully weighing out the properties of the thermal solutions, we decided to choose the option that fit our key criteria the best: thermoelectric cooling. TECs leave a small footprint and are ideal for precise temperature control.

4.7.2 Designing the TEC regulator

In order to control the TECs an adjustable voltage regulator is needed. There are many was to go about designing an adjustable voltage regulator; the most pertinent being switch mode power supplies and resistive voltage regulators.

The first design of the voltage regulator was a was that of a switching power supply. Switching power supplies have high efficiency and provide a stable DC output voltage that is ideal when controlling TECs [100]. The type of switch mode power supply of that would be most ideal for this operation would be a buck converter. The buck converter works by having all of the current being supplied to the load (TEC) being run through an inductor. The inductor is resistant to change in current, in other words, the current though the inductor changes faster than the voltage at the input. The change in current, along with the inductance, generates a voltage across the inductor.

$$V_L = L\frac{di}{dt} \tag{4.27}$$

where V_L is the voltage across the inductor, L is the inductance, and $\frac{di}{dt}$ is the change in current over time. This creates a voltage drop across the load without affecting the average power drawn from the voltage source. This is why the buck converter has such high efficiency. Efficiency is defined as ratio between the power of the output and the power being supplied from the source.

$$\eta = \frac{P_{out}}{P_{in}} \tag{4.28}$$

Where η is the efficiency, P_{out} is output power, and P_{in} is input power.

The first regulator design incorporated feedback into the design. Fig.4.54 displays the schematic for the first iteration of the voltage regulator.

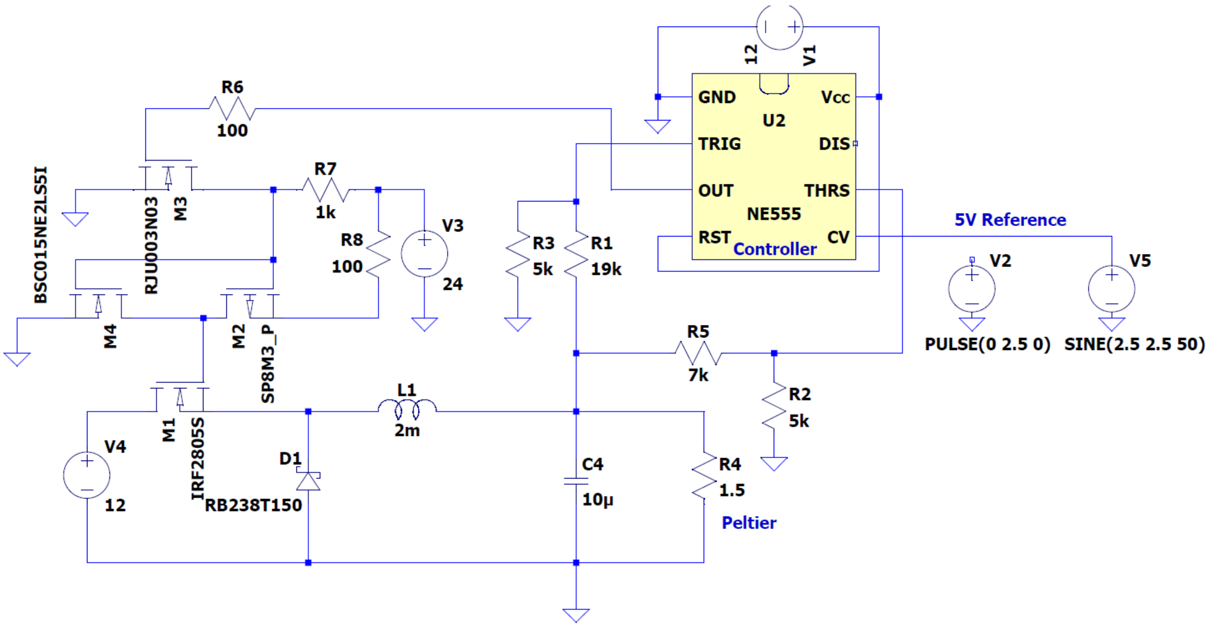


Figure 4.54: Switch mode power supply schematic

The buck converter circuit can be broken down into three main parts. Those parts being: the basic buck converter layout at the bottom, the gate driver for the main switching MOSFET, and the feedback controller. The high power buck converting section at the bottom starts with the voltage source (V4), the 12V battery. The output of the battery is controlled by the MOSFET (M1). The MOSFET acts as a voltage controlled switch that opens and blocks off access to the battery, creating the alternating current needed for the inductor (L1) to do its job.

When the MOSFET is switched off the inductor will force the current to flow, this creates a negative electrical potential (voltage) at the left side of the inductor. This voltage can become very large, and will make the inductor work as a rapidly declining power source. The magnitude of the voltage is determined by how much energy is being stored in the magnetic field of the inductor.

In order to reduce this voltage a flyback-diode is placed before the coil in order to create a complete circuit for the current to pass though when the MOSFET is closed. Schottky diodes are typically used for this purpose because they can transfer a lot of power and they have a lower forward voltage drop than typical diodes. Fig.4.55 shows the current coming from the diode and the MOSFET.

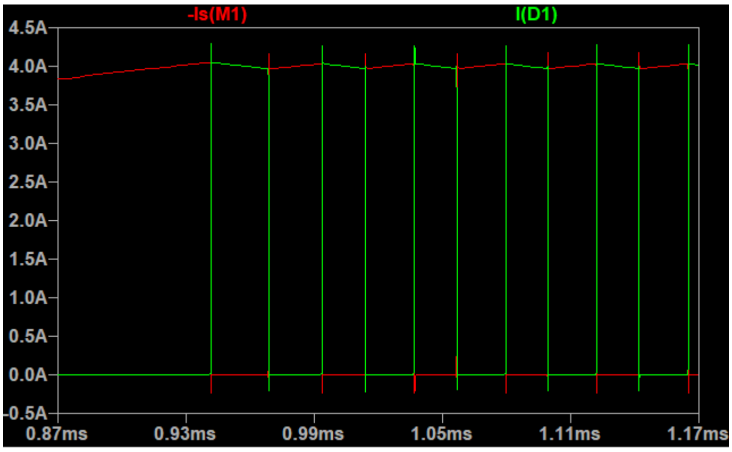


Figure 4.55: Simulated buck converter flyback and switching current

After the inductor there is a capacitor (C4) that helps smooth out the output voltage. This capacitor is not essential and could be removed if needed. The resistor (R4) represents the Peltier element. It has a very low resistance which means it draws a large amount of current (about 8A max).

The voltage across the Peltier element is monitored my the feedback controller (U2). The controller is made from a 555-timer (NE555). 555-timers are one of the most commonly used integrated circuits (ICs) available. They are typically used to generate all sorts of square waves and delayed pulses. However, if the timer is configured correctly it can also be used for many other applications.

The output of the timer is determined by three of the inputs. Those inputs being: trigger (TRIG), threshold (THRS), and control voltage (CV). When the voltage at the trigger input becomes lower than half of the control voltage the output becomes high i.e. Vcc (12V). The output becomes low (0V) when the voltage at the threshold input becomes higher than the control voltage.

Using this information we can make a pulse generator for the control of the power switching MOSFET. The resistor pair R5 and R2 is a voltage divider that scales down the output voltage of the regulator to make it the same scale as the 5V reference voltage going into the control input.

$$V_{THRS} = \frac{5k\Omega}{5k\Omega + 12k\Omega} V_{out} = \frac{5}{12} V_{out} \tag{4.29}$$

Much of the same is true for the resistor pair R1 and R3. They scale down the output voltage by another half compared to the other pair.

$$V_{TRIG} = \frac{5k\Omega}{5k\Omega + 19k\Omega} V_{out} = \frac{5}{24} V_{out} = \frac{1}{2} \cdot \frac{5}{12} V_{out} \tag{4.30}$$

By using this set-up the output voltage of the timer becomes determined by whether the voltage across the Peltier element is lower or higher than the voltage at the control input. The control voltage then becomes a reference voltage that can be used to control the output voltage of the regulator.

The output of the 555-timer is then connected to the gate of the MOSFET M3, through a resistor (R6). Everything the resistor R6 and the MOSFET M1 is the gate driver for M1. What it does is that it amplifies the output voltage of the 555-timer from 12V up to 24V. This is required because in order to fully open M1 a voltage of at least 5V–ideally around 10V–is needed between the gate and the source of the MOSFET (V_{GS}). The gate is the flat part of the MOSFET that points upward in the schematic. The source is the part that is connected to the inductor and the diode. When M1 is fully open the voltage at the source becomes 12V. Therefore the gate voltage must be at least 17V in order to keep the MOSFET open.

This design creates a closed loop system that adjusts itself in order to make the output voltage of the regulator equal to the scaled up version of the reference. The 555-timer acts as very fast comparator that can switch the input voltage on and off over one hundred thousand times per second (f>100kHz). Fig.4.56 displays the plots of the simulated output voltage across the load and the switching input of the regulator. We can see that when the current is cut off by the MOSFET, the current starts to come from the diode. These two currents are added together at the junction where they meet and becomes a stable direct current.

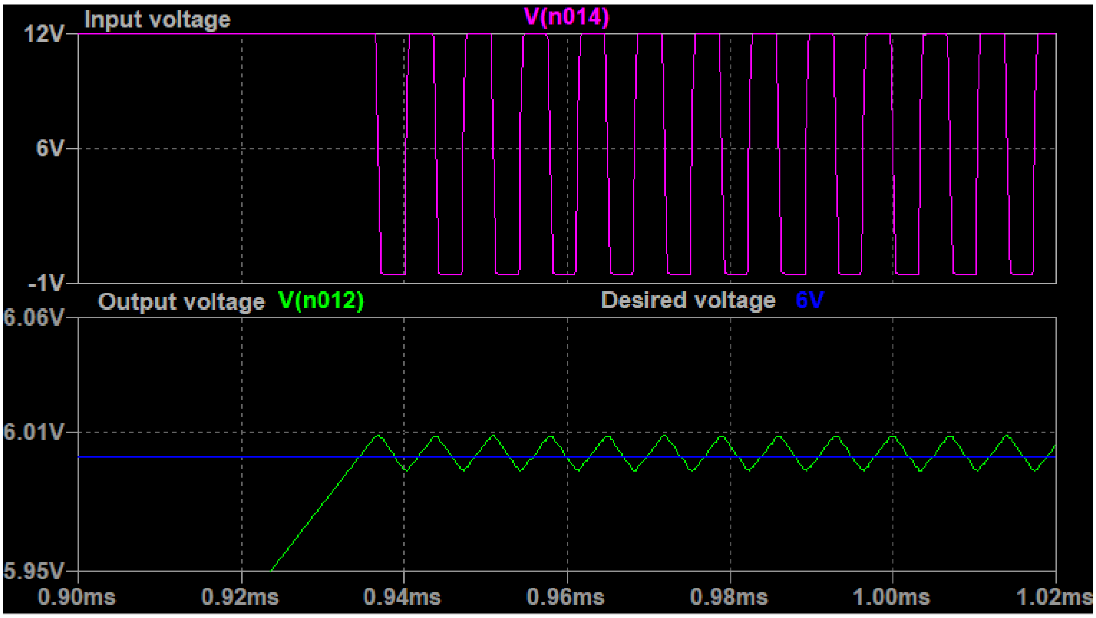


Figure 4.56: Simulated buck converter output and switching input voltage

In this plot we can see that the output voltage (green) rises much more slowly than the input voltage (pink). When the output voltage surpasses the desired voltage (blue), the input voltage is then switched off by the 555-timer. When the output voltage dips below the desired voltage, the input is then switched on again. This fast and continuous switching of the input creates the very stable output voltage with minimal power losses.

This design lacks one key function that the final regulator requires. In order for the TEC element be able to both cool and heat, it needs to be able to change the direction of the current through the Peltier element i.e. reverse the polarity. This can be achieved by adapting the design into an H-bridge layout. H-bridges are normally used to control the speed and direction of motors. They are also a core part of the design of high voltage power inverters, like the one used to run the computer running the Argos software. The H-bridge schematic can be seen in Fig.4.57.

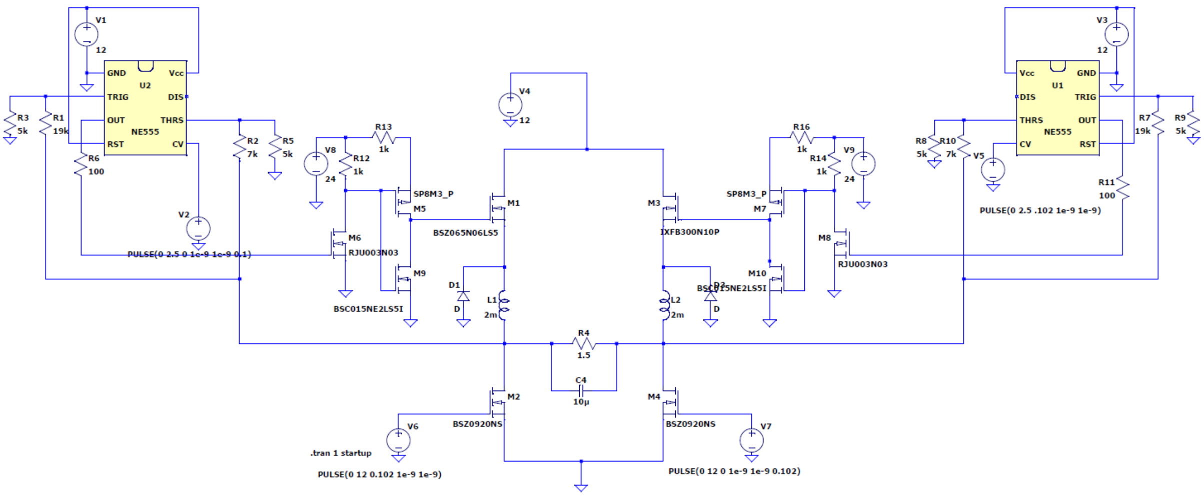


Figure 4.57: H-bridge switching regulator schematic, "The Eagle"

The H-bridge design is very similar to previous. It is essentially the same design with a mirrored copy of itself and two additional MOSFETs. The reason it is called an H-bridge is because the MOSFETs are arranged in such a way that creates the form of an H when the load is connected between the MOSFETS at the middle. It works by switching one of the top MOSFETS while keeping the opposing bottom MOSFET open, while keeping the two others closed. For example if M1 is switching, then M2 and M3 are closed while M4 is open. The two valid paths are highlighted in green and red in Fig.4.58.

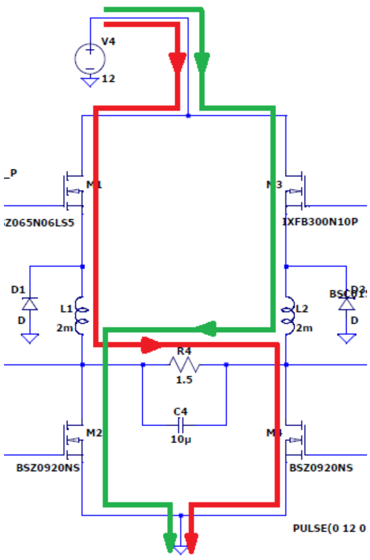


Figure 4.58: H-bridge Paths

4.8 Modeling the Software

Starting work on the software we wanted early on to map out the features of the system, which are visible to the user. We looked at the UML diagrams from Argos in both 2016 and 2018. We decided to start with designing a new use case diagram, which is based on the current state of the system. Then add the use cases which are regarded as new features of the system. While designing the diagrams, we wanted to keep a clear separation between the work that has already been completed as opposed to what we will actually be working on. Because of this we decided to use color coding, through the outer circle in each use case. this system will follow through all our software diagrams. The explanation for the four different colors are shown in Figure 4.59. While "Previously Implemented" and "Will Be Implemented" describe the pre existing use cases which does not require a contribution from us, and anything which we are adding from scratch. The other two categories are meant to indicate anything where at least some amount of work has already been done. We distinguish between "Implemented, but broken" meaning that it is a feature which has previously been completed and completed, but some change has caused the functionality to not function as intended. We also have the category of "Partially Previously Implemented" this is meant to, simply specify that some work has gone into the feature, and it may even in some cases work to an extent. However, there is still some work to be done on the feature.

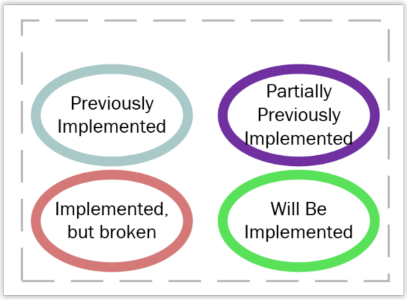


Figure 4.59: Argos 3 use case Legend

We also considered to color code each use case we would work on, this would potentially make it easier to follow from use cases and all the way to the final software architecture. However, because of the number of use cases we would be involved in, finding easily discernible color proved to be a bit of a challenge. Concluding that attempting to color code each individual use case, we chose instead to use numbering as shown in Fig. 4.60. While each use case has a name, such as "View Insta Live", the main method of referring to the use cases will be by calling them by their code like "UC1". In addition to this we will keep referring to each use case by both code and full name in every consecutive diagram which builds on it.

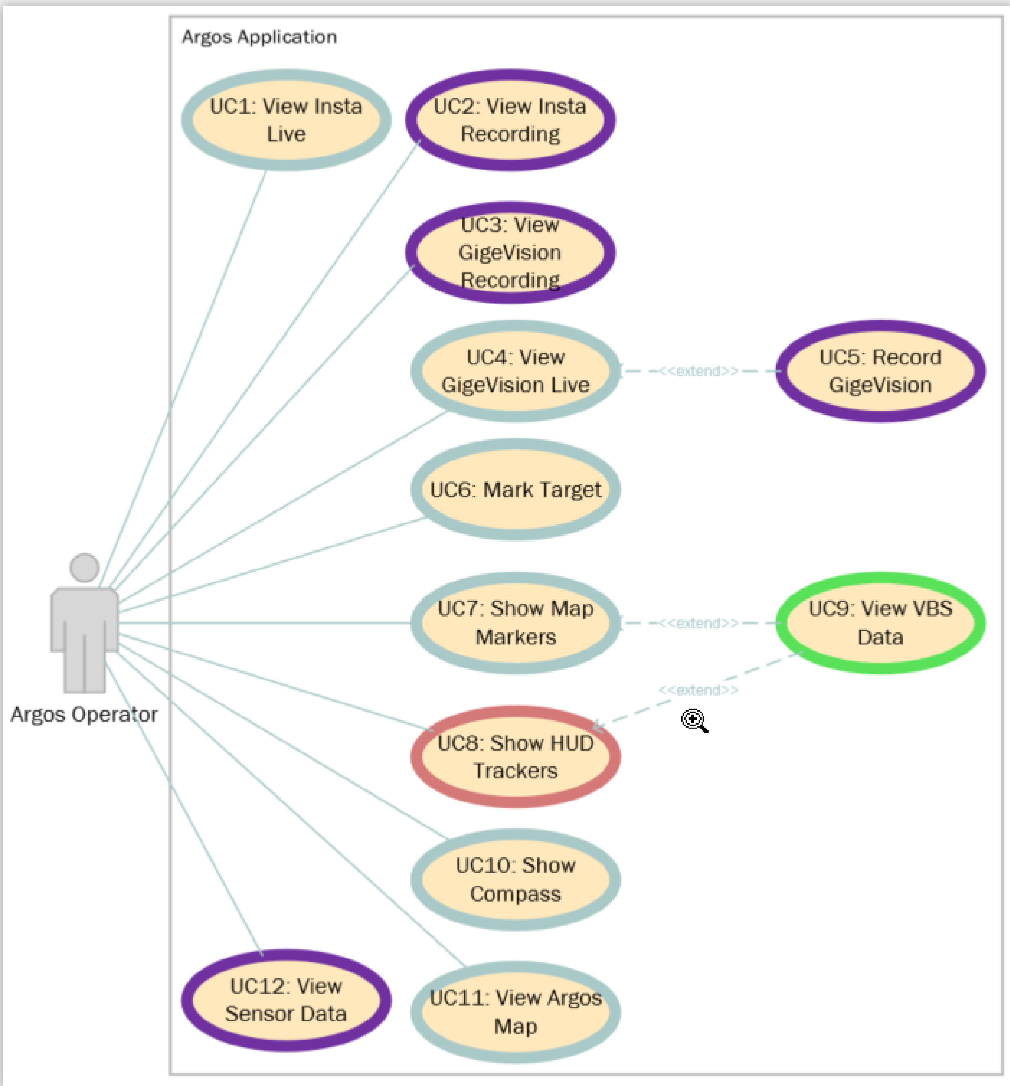


Figure 4.60: Argos 3 use case diagram

4.9 Modeling Software Communication With Battle Management System

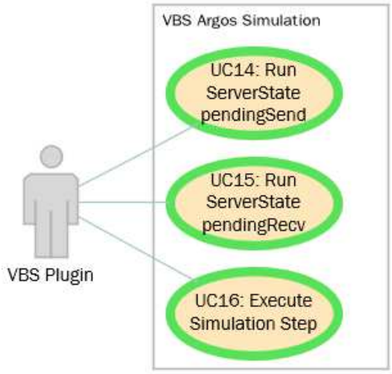


Figure 4.61: VBS use case

Fig. 4.61 use case shows how we plan to model the plugin to Virtual Battlespace (VBS). It will have different server states that will control the operations. The planed server states are pendingSend and pendingRecv. In addition will the plugin do some operations on each simulation step.

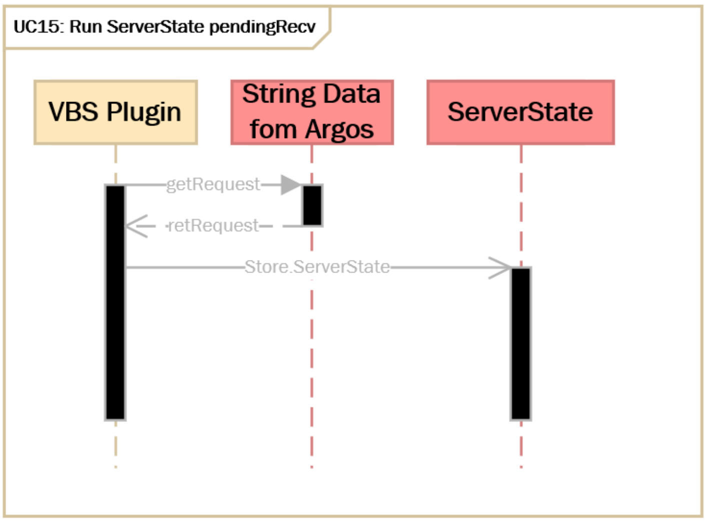


Figure 4.62: VBS sequence diagram Run ServerState pendingRecv

Fig. 4.62 shows that the gets a request from Argos and stores the request in ServerState. This is the first operation that runs after Argos and VBS connects.

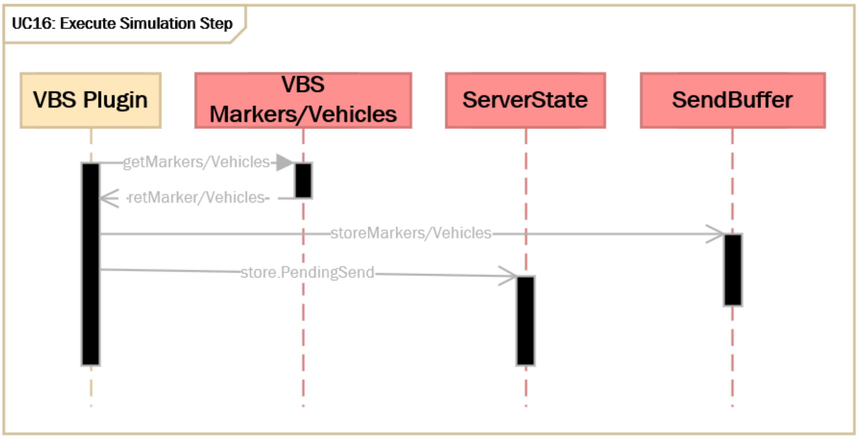


Figure 4.63: VBS sequence diagram Execute Simulation Step

Fig. 4.63 shows what happens in each simulation step. The plugin gets information about markers and vehicles and store them in sendBuffer. Afterwards it store pendingSend in ServerState. This happens every time the VBS plugin receives a request.

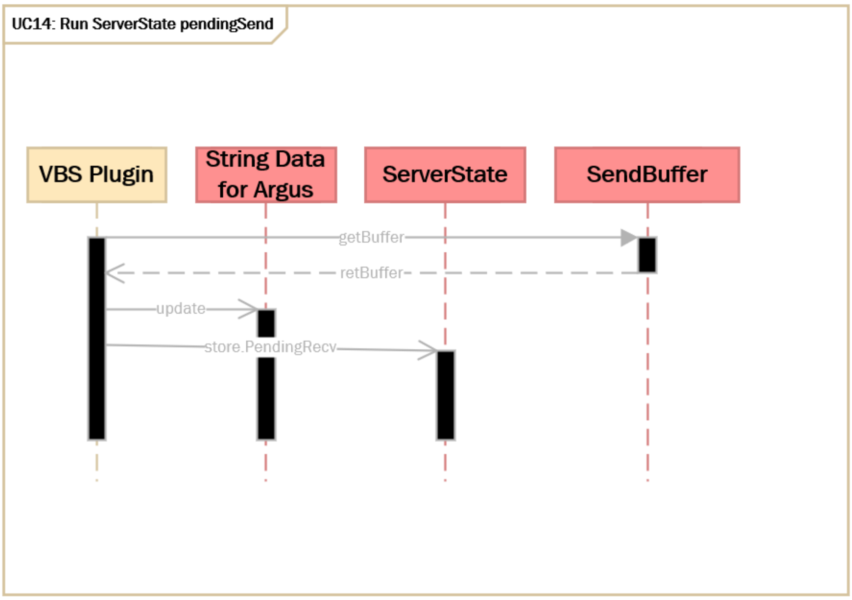


Figure 4.64: VBS sequence diagram Run ServerState pendingSend

Fig. 4.64 shows how the VBS plugin is sending the information. The VBS plugin starts by getting the information from SendBuffer, and then updating Argos with the information. Afterwards it stores pendingRecv in ServerState.

VBS is military focused version of a realistic game called Arma3, this game is also created by Bohemia Interactive and has an active community [107]. Using Arma3 as a substitute for VBS, gives us access to more information. Using Arma3 works because it is built in a similar way as VBS with a lot of similarities in scripting. It is also easier to find approximate information on VBS by focusing on Arma3 research. After a conversation with our employer we concluded that to use Arma3 as a proxy for VBS3 will give us an easier time learning and using VBS.

To reduce the size of the dll file that is the VBS plugin, we are looking to use the Winsock library [110]. This is a library that is already a part of Windows and is used for socket programming.

When creating the demonstration mission, we will have to locate it south of the university. This is because of the area that the VBS map cover. The map covers a large area south-east of the university and starts just south of the university. In the demonstration mission we will need to have different types of vehicles and markers spread out in the mission area.

From the perspective of Argos we were able to determine ahead of time that the data we wanted to store about an internal Argos target, when compared to the data we would receive from VBS were very similar, with mostly the targets from VBS having some additional data, such as type and subtype. Based on this we were able to use C++ class inheritance, in combination with overriding functions, to alsmost agnostically achieve everything we would want with both, as hinted by Fig. 4.65.

```
if (!SelExt)
    target = Player->GetBuffer()->GetTarget(targetID);
else
    target = Player->GetExtBuffer()->GetTarget(extTargetID);
```

Figure 4.65: VBS one object for external and internal targets

4.10 Modeling Target Tracking Bug Fixing in Software

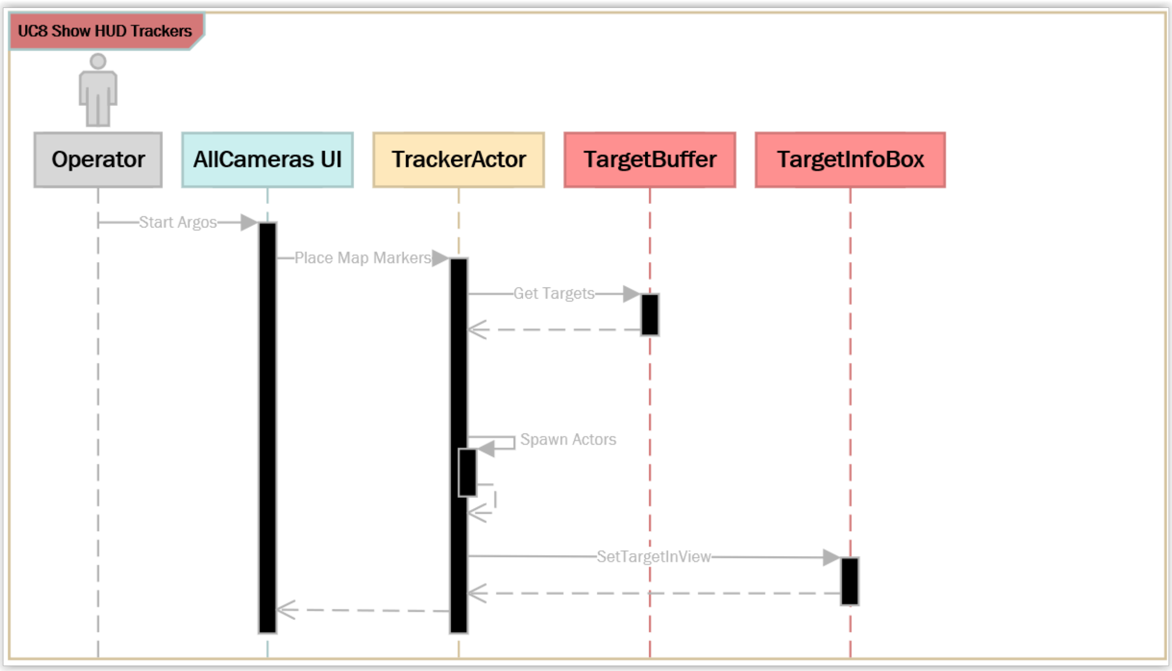


Figure 4.66: Use Case 8 Sequence Diagram

By using Fig. 4.66, we found out that to resolve the issues with target tracking, we would need to change The user interface (UI) in order to accommodate for the modified scaling. The changes would effectively mean scaling down the UI in order to place it within the camera screens in Unreal Engine space. We could also know that to re introduce the functionality of targets scaling based on distance. As well as appearing in the correct position we would have to change the controller, modifying how it calculates where to place the HUD markers. To make this change we would need the position of the Argos vehicle, fortunately the first element stored in TargetBuffer was constantly updated with the current GPS position of the vehicle. In addition to modifying the UI, and controller, we also decided to modify how TargetInfoBox stores its data. Since it contains both the data to display about a target, as well as how to format the text to be shown in the UI, we had to modify this in order to make the text formatting more easily readable. Fortunately to be able to scale the HUD markers by their distance to the Argos vehicle, all we need is the distance, which each target was already supposed to store internally.

4.11 Modeling GigeVision Recording and Playback Re-Implementation in Software

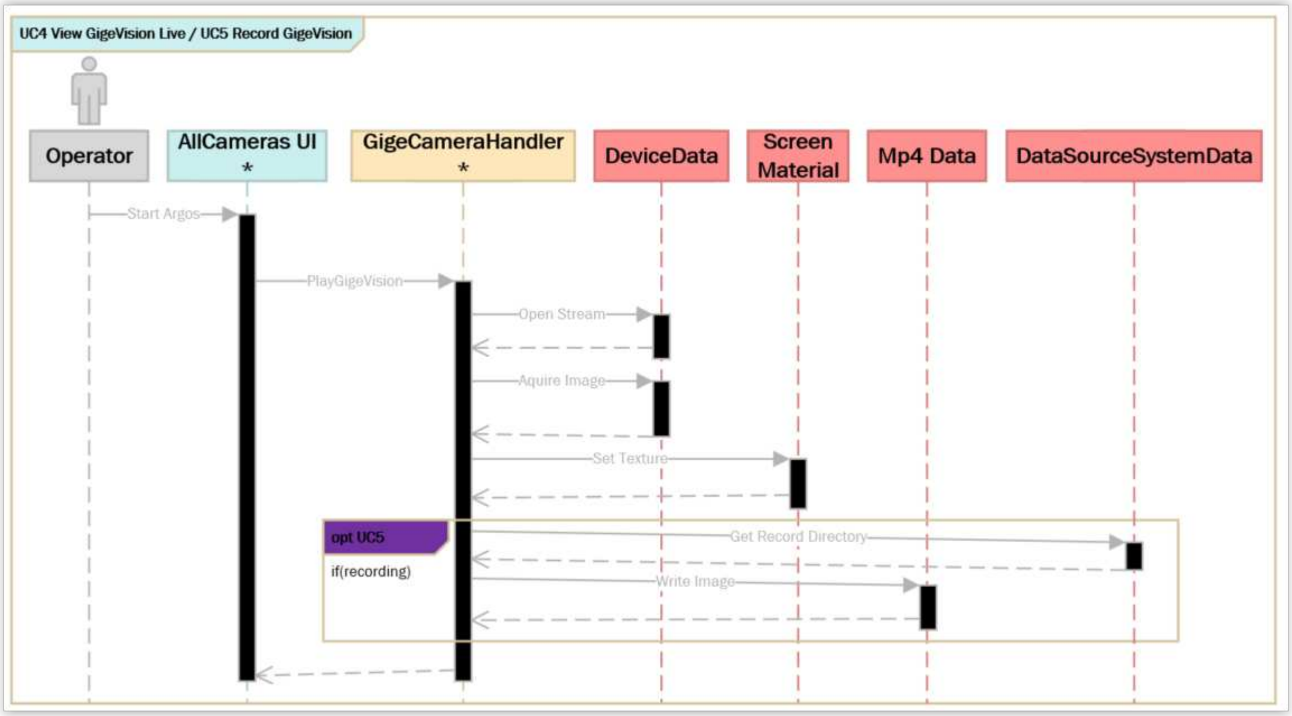


Figure 4.67: Use Case 4 and 5 Sequence Diagram

Before we could deal with how the application determines where to save the recordings, we had to find some way to actually start a recording. Fortunately the UI which sends the signal to start recordings, has indirect access to all four GigeVision cameras through the Camera Switcher. Which is responsible for spawning everything related to the four GigeVision cameras as well as the Insta 360 camera into the world.

4.12 Modeling Software Changes Accommodating Changes in Electrical System

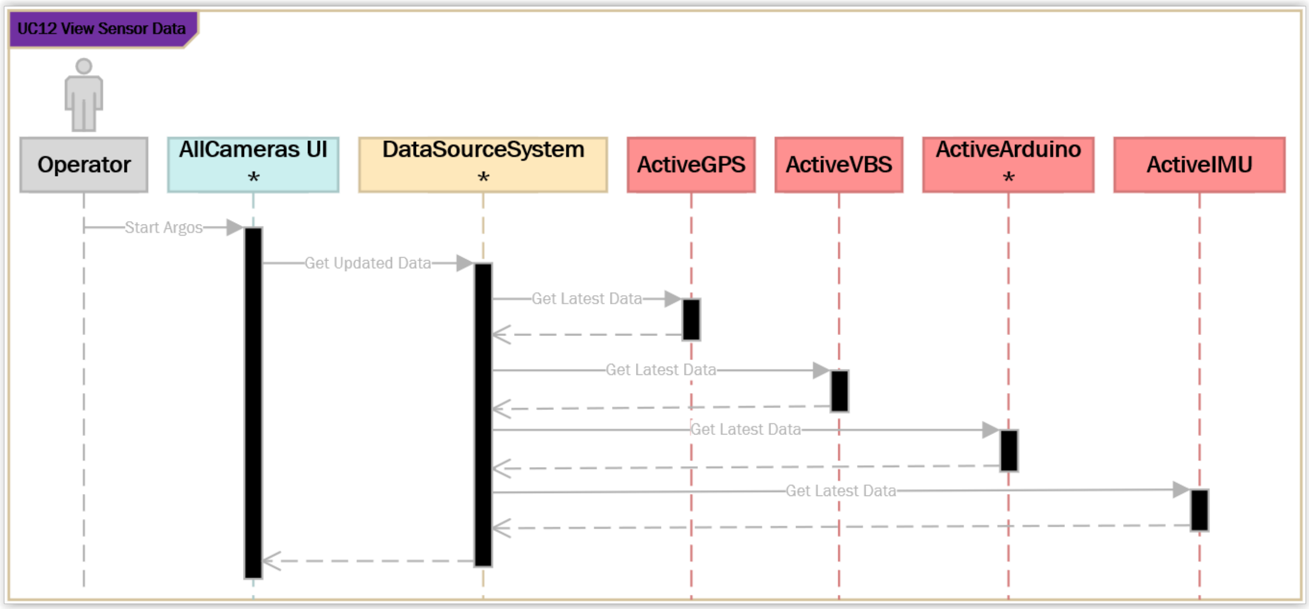


Figure 4.68: Use Case 12 Sequence Diagram

Since the data we receive from the electrical system, would change as a result of this project, indicated by the asterisk in the ActiveArduino box in Fig. 4.68. We would have to also modify the controlling code and the user interface (UI), represented as DataSourceSystem and AllCameras UI in the sequence diagram. The changes in the controller would be to parse the data coming from the arduino, and verifying it is valid data. It would also include the changes necessary to detect any values outside of the accepted range. Changing the UI allowed us to both display the values during normal operation for manual observation by the operator, as well as to display explicit warnings if the measured values reaches critical levels. Since an object already existed in the UI, to show status data related to the GPS, such as current heading, velocity and position, we also had the option to extend this object as opposed to adding our own.

4.13 Official Argos Website Update and Modeling Improvements

One of the color coding systems used on the website, has been to distinguish between, the three different disciplines represented at the project, as well as the project manager as shown in Fig. 4.69.

Figure 4.69: projectargos.net team border colors

While working on the website, it was discovered that because of the very manual process of having to manually set the hexadecimal color code for every single team member, at some point some of the colors had been replaced by different colors. As a result, while working out a easier to use solution, we also had to go back to previous years, to ensure that the same color system is used everywhere. The way we came up with in order to make color coding was through the use of CSS classes as defined in Fig. 4.70.

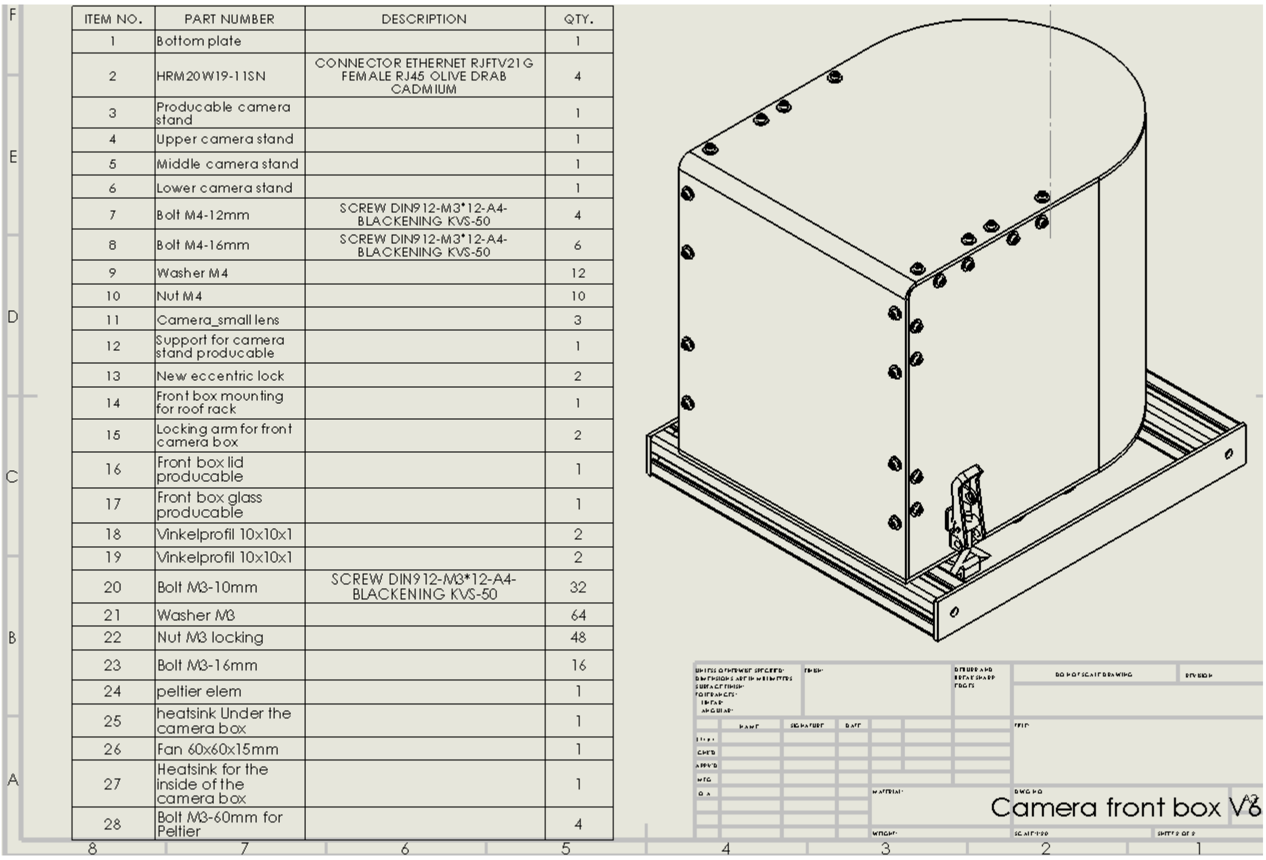
```css
.team-border.EL
{
  background: #F7AF27;
}

.team-border.HW
{
  background: #F32630;
}

.team-border.SW
{
  background: #18988F;
}

.team-border.PM
{
  background: #2852A5;
}
```

Figure 4.70: projectargos.net team border CSS definition

5 Implementation and Production

5.1 Front Camera Box Production and Implementation

For the front camera box we had some changes due to the inaccuracy of manual labour. We had to improvise and use a 3D printed part to make the aluminium lid levelled with the plexiglas so the camera box would be fully sealed, we ended up using used extruded acrylic polymer as the glass on the front camera box.[63] As of now the plexiglas has a small crack that is only hindered from growing by TEC7 sealant. This is something that has to be addressed and we hope we have time for before we present our project. You can see the parts used in the entire front camera box in Fig. 5.1 and even though we ended up using a lot of machine screws, washers and nuts you don't need any tools to open the camera box or to adjust the camera brightness. The front camera box is about one fourth the size of the legacy system and a lot easier to mount to the roof of a car because of the roof rack.



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	Bottom plate		1
2	HRM20W19-11SN	CONNECTOR ETHERNET RJFTV21G FEMALE RJ45 OLIVE DRAB CADMIUM	4
3	Producable camera stand		1
4	Upper camera stand		1
5	Middle camera stand		1
6	Lower camera stand		1
7	Bolt M4-12mm	SCREW DIN912-M3*12-A4-BLACKENING KVS-50	4
8	Bolt M4-16mm	SCREW DIN912-M3*12-A4-BLACKENING KVS-50	6
9	Washer M4		12
10	Nut M4		10
11	Camera_small lens		3
12	Support for camera stand producable		1
13	New eccentric lock		2
14	Front box mounting for roof rack		1
15	Locking arm for front camera box		2
16	Front box lid producable		1
17	Front box glass producable		1
18	Vinkelprofil 10x10x1		2
19	Vinkelprofil 10x10x1		2
20	Bolt M3-10mm	SCREW DIN912-M3*12-A4-BLACKENING KVS-50	32
21	Washer M3		64
22	Nut M3 locking		48
23	Bolt M3-16mm		16
24	peltier elem		1
25	heatsink Under the camera box		1
26	Fan 60x60x15mm		1
27	Heatsink for the inside of the camera box		1
28	Bolt M3-60mm for Peltier		4

Figure 5.1: Bill of material (BOM) front camera box

The front camera box functions as revised earlier by having eccentric locks that keeps the lid clamped to the bottom part on a list to make it waterproof. This design makes it easy to retrieve anything within the box or change the light exposure which is correspondent to requirement F.3.1 this requirement has been tested and accepted see Table B.15.

For production of the camera boxes we had to make everything ourselves. This was a choice made because we wanted a iterative design process where we could make changes on the fly. Because of tight time restraints we didn't order machined parts as these tends to have a long delivery time and the more accurate you want it the more expensive the parts are to produce. Production was a complicated matter as we are not trained in manual labour and production. But we used a jigsaw to cut the aluminium pieces to get our desired design. We also used belt sanders to smoothen the edges and create a more accurate piece. We used a metal plate cutter to get as accurate dimensions as possible beforehand we bended the aluminium extrusions with a plate bending machine to get our desired angle.

We started of by making the camera rig for the front camera box. We could not make this out of a single metal sheet because of the camera focal length. This caused the earlier explained design choice to create bracket inserts and fasten them with machine screws. All of these parts were used by using the metal plate cutter, bending machine and drill. The holes for the cameras were lines up using a laser cut part to get them as accurate as possible and as explained earlier the cameras are mounted upside down for easier retrieval.

To bend the plexi we had to create a mold to bend the plexi around inside an oven. It is utmost importance that the plexi has slow cooling so it doesn't warp and cause tension. Because of this we are limited to not being able to open the oven when forming the plexi while making sure the plexi forms around our mold. The mold was created by laser cutting plywood scaled 4mm smaller than the outer arc of the plexi. This is to get the correct dimension when the plexi is formed. We then used clamps to clamp a 1mm aluminium sheet metal to the plywood to finish the mold in our required design. We had to make sure everything the mold was made out of could handle up to 150°C. In the end we put an aluminium foil over to cover all the impurities in the aluminium sheet as to not get these impurities in the camera view, you can see the plexiglas is warped at the edges in Fig. 5.2. Our mold is not however perfect, we did not get the exact right arc and length we needed. Regardless we adapted and changed the design of the bottom plate and the aluminium lid.

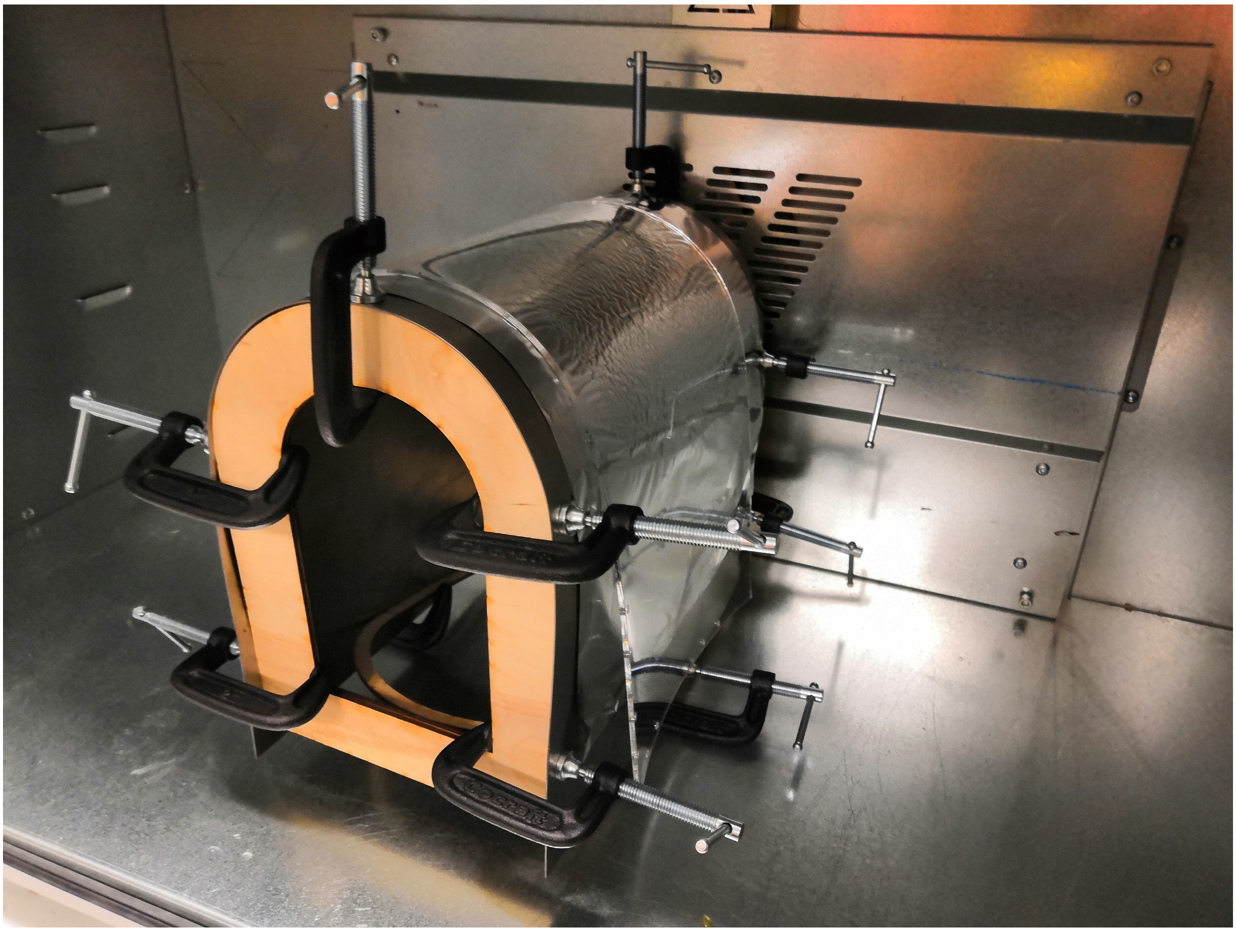


Figure 5.2: Plexiglas mold in oven, warping at the edges of the plexiglas

The aluminium lid for the front camera box is produced by first using the metal plate cutter to get the desired length and width of the lid. Then we use the bending machine and bend the plate 90°, the last step is to use the Jigsaw to create the arc with the radius of 100mm see Fig. 5.3 we used a laser cut plywood part to draw the arc to know where to cut. We bought a 10x10mm angle bracket to fasten the aluminium lid to the plexiglas with machine screws, washers to lessen the pressure on the glass and minimize galvanic corrosion and lock nuts to avoid using loctite. We were advised by one of our supervisors that they had experienced plexiglas would crack when subjected to loctite we did a test on this and did not experience any deteriorating in the plexiglas but not wanting to take any chances we rather went for lock nuts.

Figure 5.3: Front camera box lid jigsaw cutting

The bottom plate was one of the last things produced because it's the easiest part to change if a redesign is needed. We used the plate cutter as previous to get the correct dimensions and then we used a jigsaw to cut the arc. Again like previous we had laser cut plywood to get the form of the parts and marked using the plywood. For the placing of the component inside the camera box there was some changes under production. We ended up placing two connectors on the left and two on the right side. this was done because of the roof mounting and the eccentric lock used more space of the bottom plate than expected. It was also a wish that the peltier was connected to the camera stand using it as a heatpipe for temperature regulation. The outline and holes to the heatpipe and peltier was created by drilling a big enough hole in each of the edges using the jigsaw to cut out the form. As it's hard to get a perfect straight line with the jigsaw we used the belt sanders to compensate for any inaccuracies.

In the end everything was painted using bengalack, this is because it can be used on a untreated aluminium see Fig.5.4.

Figure 5.4: Spray painting the bottom plate of the front camera box

5.2 Rear Camera Box Production and Implementation

The rear camera box top shell was made out of carbon fiber composite using Hexcel carbon fiber in composition with Ampreg 21 epoxy system. This gives a shell strong enough to withstand the air pressure from the wind. The casting of the top shell was done in Krags composite lab and it's equipment. We printed several 3d models of the top shell to get an idea of the size and positioning of the camera as it is 185°field of view. We also needed a mould for casting. we coated the mould with wax so it's possible to remove the mould after casting. The casting was done on the red mould seen in Fig. 5.5.

Figure 5.5

The shape of the mould was tricky to cast around. The front camera hole was the main issue, where we had a hard time getting the fiber to stick to the mould. The area around hole was too small, and fiber did not want to stick to the surface. We managed to get it to stick and ended up with the result seen in Fig. 5.6

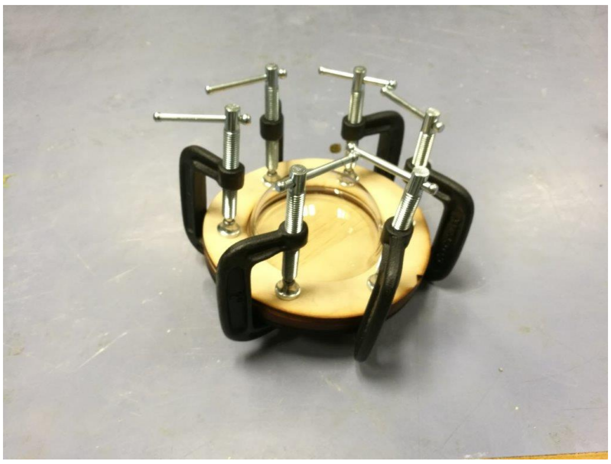
Figure 5.6: Successful casting of the top shell of the rear camera box

For the locking mechanism between the top shell and the bottom plate we decided to go for eccentric locks. This gives the option to lock the top shell in place, but also gives the option to access the camera in a short time period as requirement F.3.1 requires. This requirement was tested and accepted as seen in Table B.15. We decided to mount the camera upside down. This makes it easier to mount and dismount the camera if ever needed. To make the the box water resistant, we sealed every bolt hole with silicone, and equipped the bottom plate with a rubber gasket to clamp down on. We found this was successful after testing as seen in Table B.17. We decided to use our idea with a glass dome as our solution and we decided to blow mould the acrylic dome ourself as plastic domes are expensive. We decided to use the same glass as on the front camera box, plexiglas[63]. Several tries were attempted as seen in Fig. 5.7.

Figure 5.7: First try of blow moulding plastic dome testing if it was possible to blow mould ourselves. Did not achieve sufficient pressure, as mould was not air tight. Did not achieve sufficient size

Several attempts were done to gather experience and we decided to refine the moulding method. We came up with a design for the blow moulding contraption seen Fig. 5.8a and did several more attempts including our final successful attempt seen in Fig. 5.8b.

(a) Design of blow mould contraption

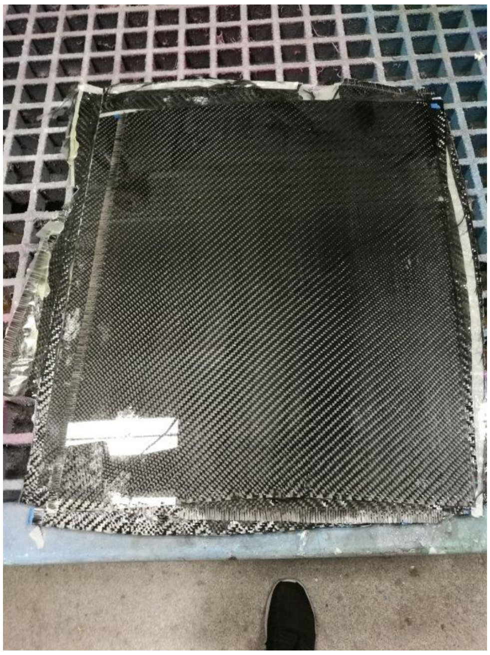


(b) Successful blow mould of plastic dome using air pressure

Figure 5.8: Successful testing of the new blow mould contraption

To keep the view of the glass clear during rain, we decided to use water repellent coating on the glass. This was an option we decided on after we ran successful tests of increasing contact angle as seen in Table B.21. This makes the water form into droplets and roll off easier from the glass. This is the same method they use on wind shields on cars. This fulfils part of requirement F.3.2 that says that the cameras shall not be affected by the environment.

The rear camera box required a great deal of post processing including post curing, sanding and varnishing. During the sanding, weaknesses in the top shell cast were discovered. Holes appeared in front of the casting after sanding. To repair this, we cast a carbon fiber composite plate and cut/milled repair pieces and support for the eccentric lock mount as seen in Fig. 5.9b. We coated the pieces with adhesive and managed to glue the parts together with a reasonable result.



(a) Carbon fiber plate used for cutting repair and support pieces

(b) Repair and support pieces ready for adhesive to fix the rear box

Figure 5.9

The bottom structure of the box is made out of aluminium EN AW 5052. The bottom part of the rear box was produced in the University facilities, just like the front camera box. A thermoelectric peltier element is used for heating/cooling with a heat sink both inside and outside the box. This ensures that the cameras have the required operational temperature. The cable connection will also remain the same as the legacy system.

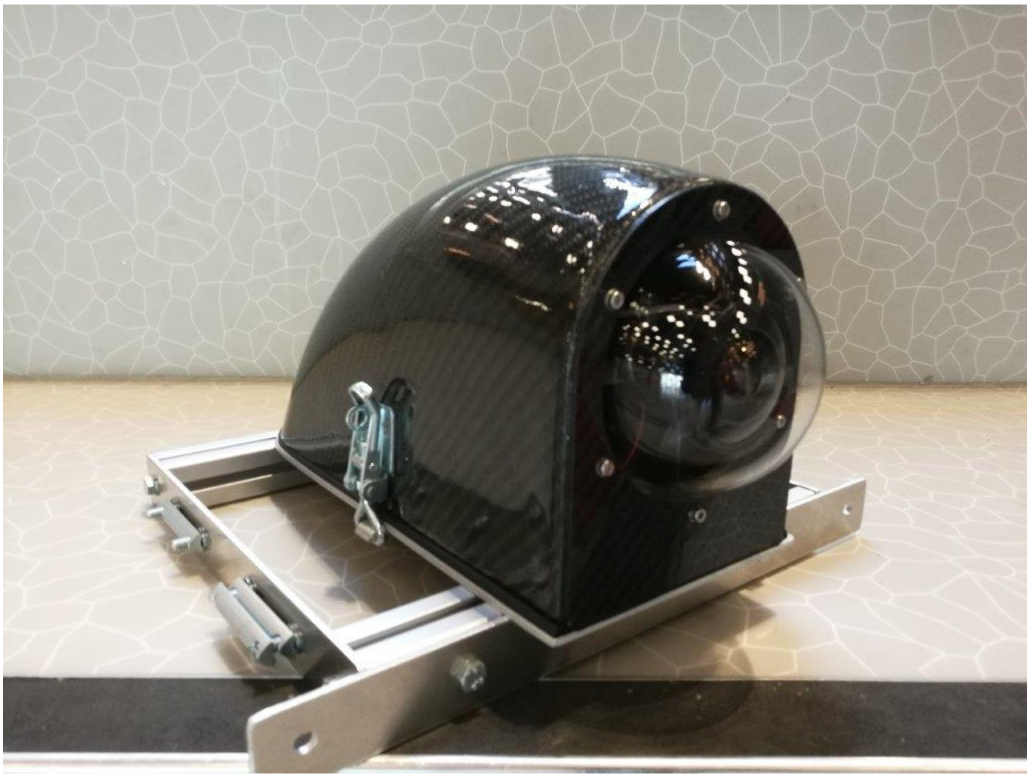


Figure 5.10: Produced rear camera box for Argos

5.3 Mounting System Implementation and Production

After receiving the aluminium profile system order, we immediately began producing the roof mounting system. The aluminium profiles needed to be cut in correct lengths and fitted. The profile came in lengths of 2m. This was achieved in the university laboratory facilities. The system uses t-slot nuts, and corresponding slotted aluminium profiles. Angle brackets were used in corners to create a frame easy enough to lift on top of a car. In our design we also decided to go with a modified eccentric lock. The roof rack of the car has rectangular crossbars which our idea was to lock in on with a hook. We actually modified stainless steel butcher hooks. We cut off one side, and threaded them to m6 threads to mount in the eccentric lock. After initial mount, we found out that the friction between the roof rack of the car and the locking hooks were too low and had a risk of sliding. We decided to coat the hook using heat-shrink tubing as seen in Fig. 5.11.

Figure 5.11: Hooks for mounting between roof rack and our mounting system

The rubber of the heat-shrink tubing increased the friction between the roof mounting hooks and the roof rack crossbars. During our mounting test we found that the friction was sufficient. Having threaded hooks makes the locking mechanism flexible but sturdy and gives us the ability to adjust the tightness against the roof rack of the car. Just turning the hook by one round will tighten or loosen the hold against the roof rack of the car. On the eccentric locks there are safety locks to prevent the mechanism from opening during use. Using a spring pin or locking pin will prevent unwanted opening of the eccentric lock. The eccentric locks can be mounted directly on the aluminium profile system as seen in Fig. 5.12 locking it in place.

Figure 5.12: Eccentric lock mounted on the roof mounting system

To lock the camera boxes in place, we created frames to mount the camera boxes on that can slide into the slots of the aluminium profiles. We decided to lock the frames in place with an aluminium plate across the tip of the mounting frame. Using hand screws we can secure the frames in place reducing the movement of the camera boxes and securing them in place. Using the t-slot nuts we ordered we can slide the boxes into the aluminium slot on the mounting rack. Since time was short we decided to design our own hand screw. Threading the aluminium profiles as m6, as they are designed for, we were able to use the m6 bolts we already had. We printed a handle and glued them together as seen in Fig. 5.13.

Figure 5.13: Fastener for camera box frames

Initially our idea was have the aluminium securing plates as a separate part. During testing we saw that we were able to mount the securing plate on the camera box frame, which gave us less loose parts which is desirable. Successful test report for mounting can be seen in Table B.15

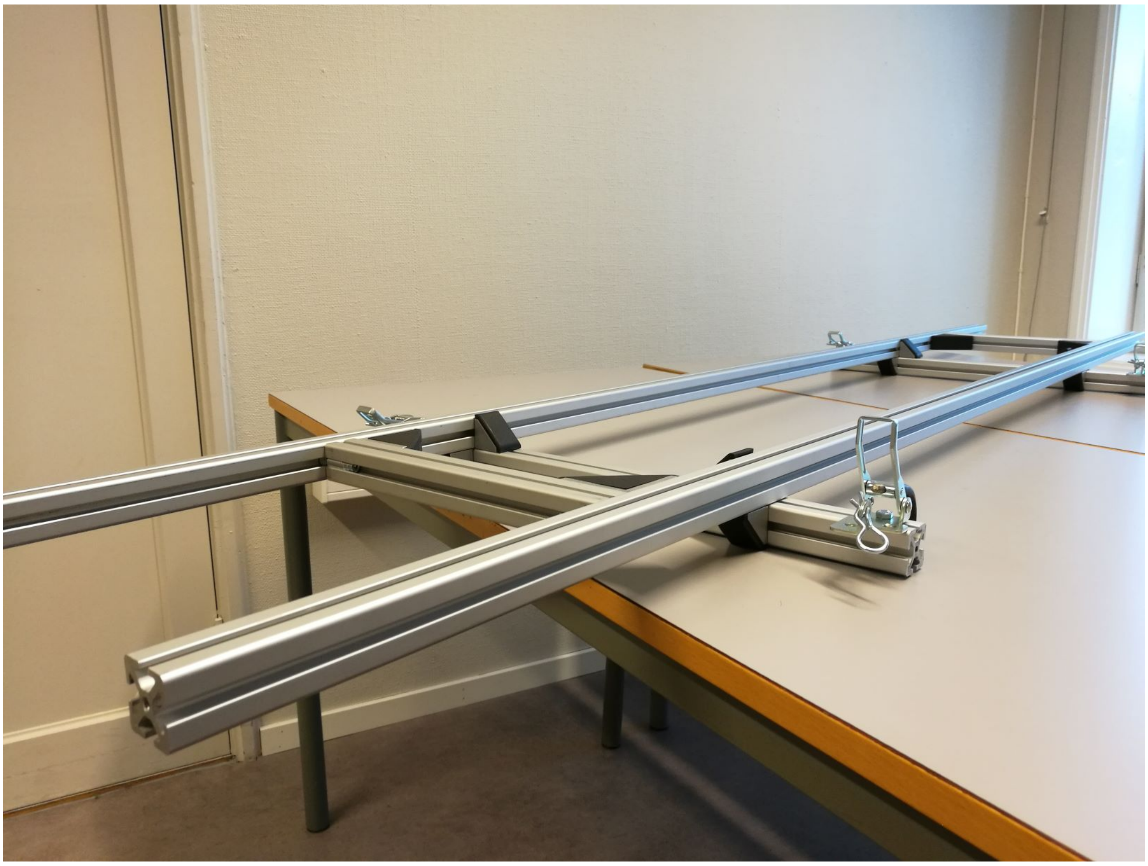


Figure 5.14: Produced roof mounting system for Argos

5.4 Power System

We presented the power supply options to our customer during a meeting to uncover thoughts and preferences. During the meeting, it was pointed out that modularity of the system was still a very high priority, since the system will still be undergoing development after this project. The customer also pointed out that ease of mounting the system in a car should be a high priority. Due to these priorities, it was agreed upon that continuing with the power inverter solution, would be the best way forward. After providing the new weighting for our options in the decision matrix seen in Fig. 4.50, this option came out at top.

In difference from the current system we received from the earlier projects, this solution will not draw any direct power from the battery power lines. This will remove the needs for filtering the input of the inverter, which would otherwise be necessary, due to the noise made by the inverter itself.

After uncovering measurements from the power usage of the system as a whole, a decision was made that a 3000 watt inverter would not be necessary and just add additional weight. A decision was made that a 1500 watt inverter would be more than sufficient, since the highest watt usage of the system, was measured to be 600 watts. Even though a 800 watt inverter would be sufficient, the modularity was still important and the extra power could be necessary later in development. The diagram for the complete electrical system is shown in Fig. 5.15.

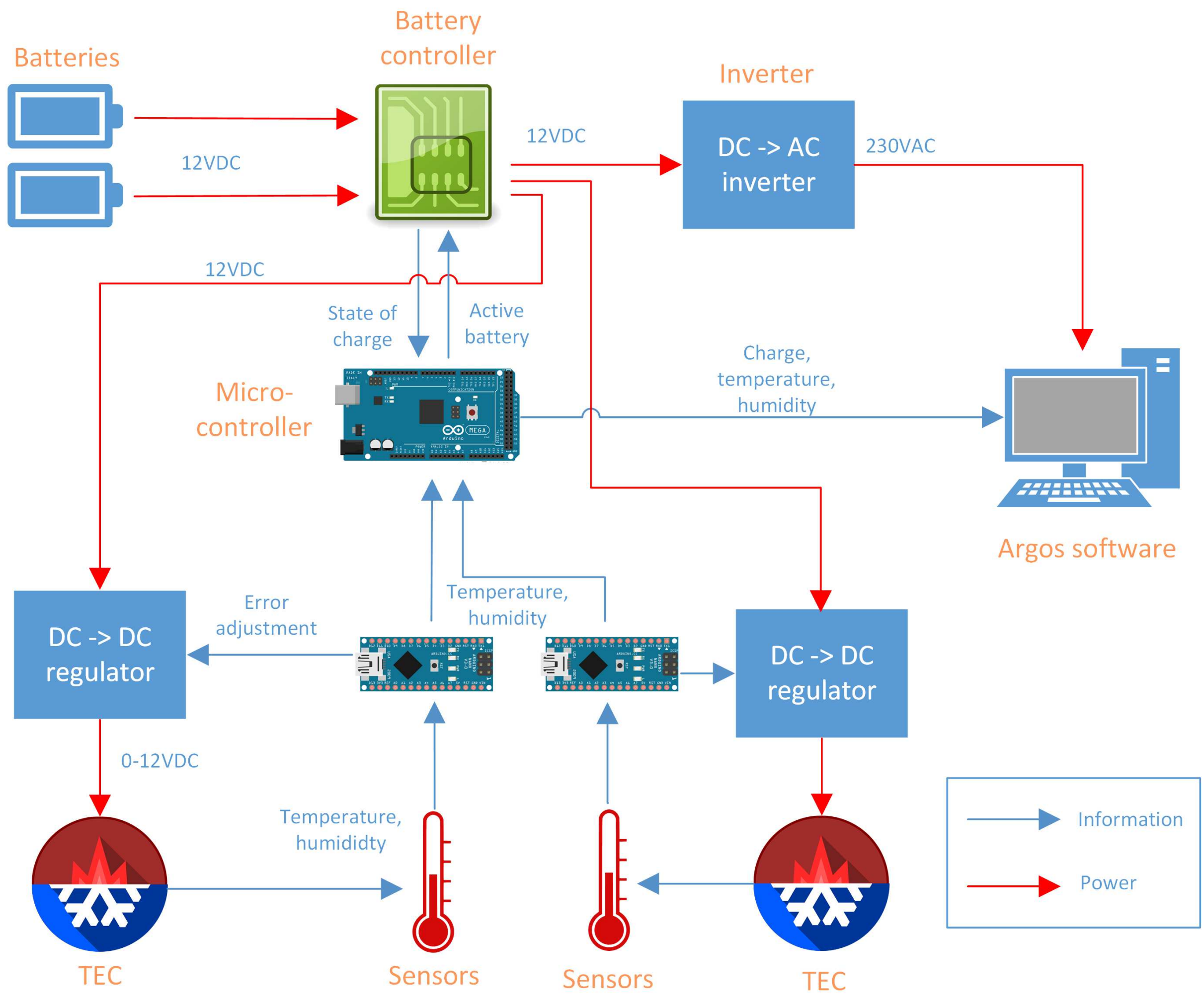


Figure 5.15: Electrical system overview

5.5 Battery Monitoring & Controller Circuit

Designing a circuit can be very intricate, considering there are so many variables that can affect the outcome. We had already decided that a voltmeter and a coulomb counter would be a good approach for measuring the state of charge. Additionally, we had a pretty good idea on how to implement said things. An operation amplifier has extremely many useful features that we wanted to take use of and therefore found it fit for our circuitry. To begin with, we used a differential amplifier for both the voltmeter and the coulomb counter. Only difference was that the coulomb counter would integrate the output, while the voltmeter would just output the difference.

The first iteration In the first iteration we focused on designing the battery monitoring circuit. The battery monitoring circuit is divided into two parts, voltmeter and Coulomb counter. Even though these circuits will be working towards the same purpose, estimating the battery state of charge, they preform two different tasks, one measuring voltage and one measuring charge over time.

To begin with we created a simulation of the circuit in LTSpice to be sure that our calculations for the ideal circuit were correct. One of the obstacles we had to consider, were that the Arduino could only read voltages between 0 and 5 volts.

The voltmeter, as stated earlier, would have a differential amplifier. One of the inputs would have the positive 12 volt battery pole, directly coupled into a sensing resistor. The other input would be coupled into a voltage reference, where we chose to use a Zener-diode. The reference voltage would be 10 volts, so that the voltage read by the Arduino, would be between 10 and 15 volts. Since a battery fluctuates from 12 to 13 volts, depending if its fully charged or not, this would allow the sensing to be perfectly centred.

The Coulomb counter would have very similar set up as the voltmeter as mentioned earlier. In difference with the voltmeter, the output of the Coulomb counter would be a integral of the input. As with the voltmeter, the circuit was first designed in LTSpice to create a simulation of the Coulomb counter and as shown in Fig. 5.16, the output increased, depending on the voltage drop across the sensing resistor.

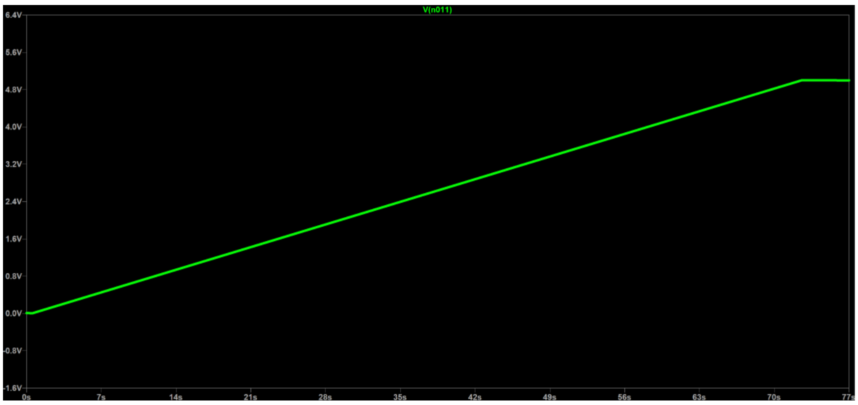


Figure 5.16: A simulation of Vout in Fig. 4.52

Having the batteries parallel coupled should be avoided at all cost, which is why hot-swapping would become a part of the circuit. With this in mind and a wish for taking use of MOSFETs for the hot-swapping, a battery controller Integrated Circuit (IC) would also be necessary. The IC would allow us to have both of the batteries active, even though not preferred, but still not allow current to flow from one battery to the other. We found a IC named LTC4359 that preformed these functions and simulated the circuit in LTSpice successfully [87].

After simulating the circuit, we ordered the parts necessary to preform tests on the circuit. Even though this sounds simple, most of the job is to read and understand the datasheets, so the components with correct values are bought. An example of this would be the IRBF7430PBF [88], which was rated for as much as 195 amperes, which would be more than sufficient for the battery controller.

After receiving the components, we began to mount the components and wire the circuit on a prototype circuit board, as shown in Fig. 5.17. This would allow us to preform tests on the circuitry. Producing a circuit for testing on a prototype board, reviled itself to not be the brightest idea. If a component either were defect, or the circuit didn't work, the component would have to be de-soldered to be replaced. This became very time consuming.

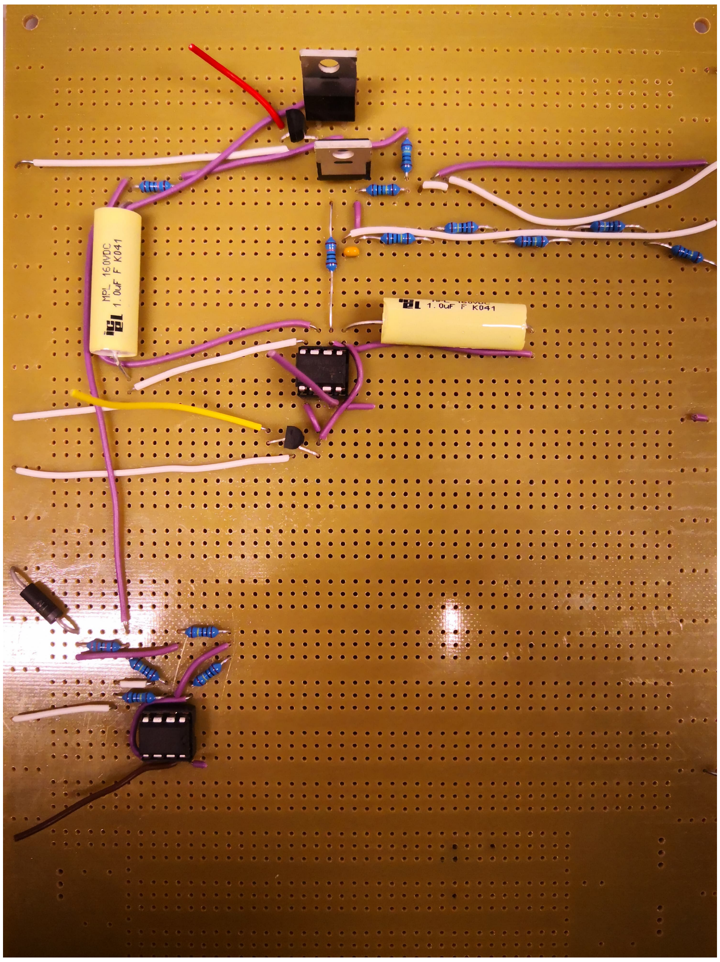


Figure 5.17: Picture of the first prototype circuit with components

After testing the circuit, a few problems reviled itself. The first problem was the readings of the voltmeter was a bit inaccurate. The second problem that we uncovered, were that the Coulomb counter did not register the small voltage drop across the MOSFET.

A second test, which resulted in a third problem, was preformed on the MOSFETs that whether they would tolerate the massive current. By connecting the batteries to the drain and the inverter to the source on the MOSFET, we were able to turn on the power. We induced a heavy load to the circuit and the MOSFET exploded.

The second iteration In the second iteration we began to uncover how the problems found in the first iteration could be solved.

The solution to the second problem was easy to uncover. The tolerance of the Zener-diode was $\pm$10%. This would result in the voltage reference providing different voltages, depending on the Zener-diode. To prevent this from becoming a problem, the Zener-diode was replaced by a voltage reference IC. A voltage reference IC provides a much more stable output. In addition to a stable output, we chose a voltage reference IC with the possibility to trim the output. This would allow us to adjust the voltage reference, depending on what output we required.

The solution to the first problem was much worse to solve and can be divided into two categories. The value of resistance and the sensitivity of the operation amplifier. When we researched the operation amplifier, we discovered that the operation amplifier has a input resistance of 1000 Giga Ohm. This should be close to infinity, but with a resistance of 6.8 Mega Ohm on the input could affect the behaviour of the operation amplifier. To work around the problem we learned about the instrumentation amplifier, which amplifies the signal difference. The outputs of the instrumentation amplifier is then fed into a differential amplifier.

Even though an instrumentation amplifier would solve the problem caused by the resistors, the operation amplifiers would require a voltage source higher that 12 volts, which is the sensing voltage. The amplifier increases the voltage difference, but with the same voltage level as the input. If the operation amplifiers has a lower voltage that the input, the operation amplifier would reach its saturation point. Also, if the output were to exceed 5 volts, the Arduino could no longer measure the signal.

To work around the saturation, but still keep the voltage less than 5 volts, we voltage divided the circuit between the instrumentation amplifier and the differential amplifier. Even though this wasn't the most idea solution, it would resolve the problem.

For the problem with the sensitivity of the operation amplifier, the voltage difference necessary to provide an output was 15 millivolts [89]. We had to find a component with lower voltage difference needed to register difference. We found the OPA177 operation amplifier that were able to sense differences as small as 10 microvolts [90]. Since we would always have a difference of 1 to 10 millivolts, this would be sufficient.

For the third problem, provided by the single MOSFET exploding when running 120 amperes trough the component, we turned to one of our professors to understand what we had overlooked in the datasheet. The professor pointed out that the power dissipation at 375 watts could be a limitation. Even though as shown in Eq. 5.1, the statement does not make sense. We searched the datasheet for answers, but could not find an answer that made sense.

$$P_D = I_{DS}^2 * R_D S = 120A^2 * 1.2mOhm = 17.28W \tag{5.1}$$

To try to see if there was a way to work around the problem, we first tested another time to see if there was a defect component. To later try 4 MOSFETs in parallel. Which solved the problem partially, since there was now huge amounts of head dissipating from the components. To prevent to much overheating we added an additional MOSFET to the switching circuit, so that it became 5 MOSFETs in parallel as shown in Fig. 5.18, which made the problem more manageable.

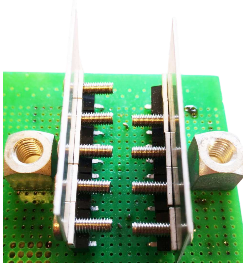


Figure 5.18: Picture of 2x5 MOSFETs on a prototyping board for testing

Further into the development we tried to find a way for the Arduino to read the output as easy as possible, and we began to discuss a component one of our group members had knowledge of, the 555-timer integrated circuit. The 555 timer has many different applications, but we use it for extending a pulse outputted by an operation amplifier driven into saturation, which then resets the integrator. This way we do not depend on the speed of the Arduino to reset the circuit, but only depend on it to read a pulse, given by the 555-timer.

After the massive changes in the second iteration we simulated all the different circuits in LTSpice separately, before simulating the whole circuit as one. The changes made were simulated and produced a positive outcome.

Since time began to become a problem, we also began to produce a Printed Circuit Board (PBC). Both to reduce time for testing and learn the process of creating one. If we were to do any mistakes on the first PCB we also would have time to redesign and make a new, before the end of the project.

The difference when producing a PCB instead of a simulation is finding the correct footprint. A footprint is the placement on the circuit board combined with drill holes and solder mask to hold the component in place. We had datasheets available but not all of them gave the exact footprint, but rather the measurements. To convert the measurements into a footprint, very precise placement and drawing is necessary. Producing a footprint is very time consuming, which made the process long and tedious. Its also easy to make mistakes during this process, since there are so many dimensions that has to be correct Otherwise the components wont fit or the pins will not be traced correctly.

When the footprints were completed, the drawing of the schematic began. The drawing is pretty similar to what one would do in a simulation program, but one also has to connect all the signals correctly.

When the signals are connected and the footprints are finished, all that remains is place the components on a board and trace the wires. Fig. 5.19 shows how all the signals are tangled when entering the board designing.

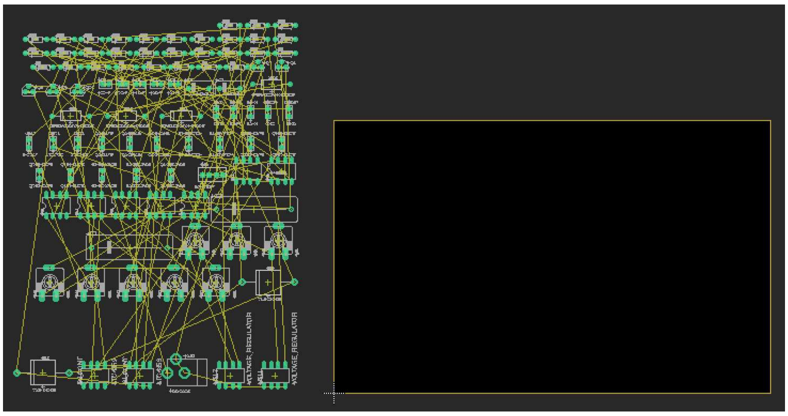


Figure 5.19: The first PCB before placement of components

To go from all the components being tangled, to having a placement as smooth as possible, requires both time and skill to do it professionally. A few crossovers with the signals (yellow lines) in not possible to avoid, but the fewer the better. In Fig. 5.20 the placement of the components is done and much less tangled.

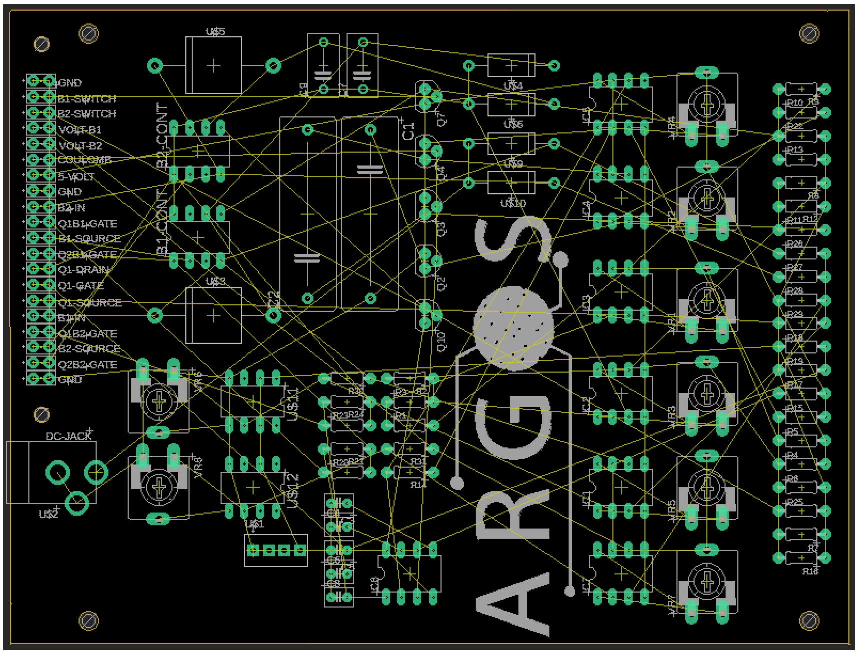


Figure 5.20: The first PCB with finished placement of components

When tracing the wires, the current that will flow trough the circuit, has to be taken into consideration. Normally a width of 8 mills is used for signals. 8 mills will allow ca 1 ampere to run trough without causing to much heat. If a higher current is needed, a lager trace will be needed. When the larger traces are set, the auto-routing can take place. With the auto-routing, a lot of time is saved due to the computer does most of the work. The auto routing can also be set to larger width, but this will increase the boards size. The last step is to do adjustments to the traces. The finished board schematic is shown in Fig. 5.21.

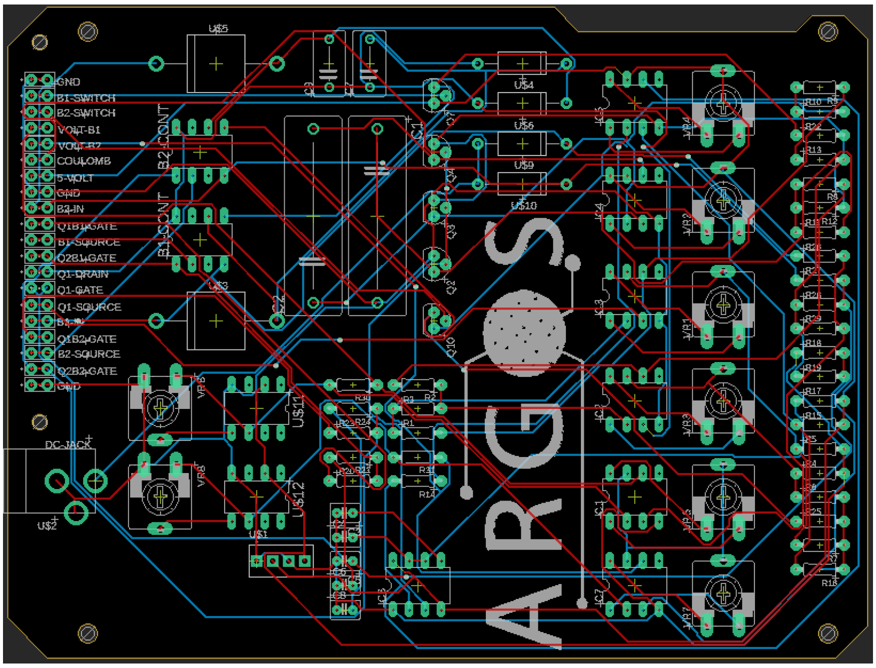


Figure 5.21: The first PCB fully traced

Most PCB producers only require a Gerber file to make the circuit board. The files takes only minutes to extract. The file contains each of the layers including measurements

When the board is produced, only the placement of component on the physical board remains. If the pre-work with the footprints were done correctly, it will only take a couple of hours, depending on how many components on the board. The finished product is shown in Fig. 5.22.

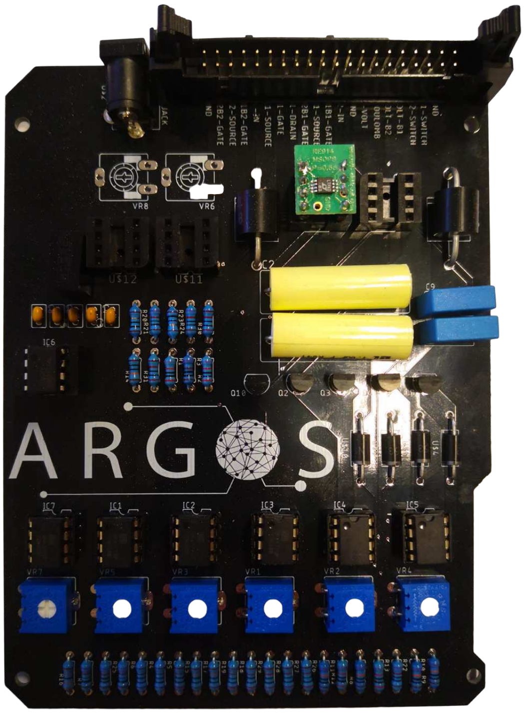


Figure 5.22: Picture of the first PCB with components

When the components were mounted, a testing of the circuit was necessary. Even if the circuit and PCB had been simulated and double checked, there could be defects in the product. A mistake was found, that one of the signals were missed and had to be connected by wire. We connected the circuit to the necessary power and signal lines and turned on the power. Many of the components began to smoke, which usually indicates something has gone very wrong. The problem was that we only had components for this one test and if we were to do another, we still had to wait for a new order.

The third iteration A search for indications of what caused the components to smoke began. While checking datasheets we uncovered a problem with the battery controller IC that weren't noticed earlier. The IC was only rated for currents up to 20 amperes. This alone would not cause the circuit to smoke, since we only ran a few milliamperes trough the circuit. With this knowledge, a redesign of the battery controller would be necessary, but the search for the cause, continued.

With knowledge of the battery controller needing a redesign, we decided use relays. This would save us both time and problems with heat dissipation. We carefully considered the necessary specialisations for the relays before deciding on the HRC-150 300A automotive relays [91]. The relay allows continuous current at 150 amperes.

Trying to find the cause of the problem with the PCB we began testing of all the separate circuits using breadboards.

Beginning with the voltmeter, the circuit was wired and connected to a supply voltage. As expected the voltmeter supplied the voltage between 1 to 4 volts depending on the input. However, there was a zone where the operation amplifier did not react, 0-1 volts and 4-5 volts. This could be resolved, but still didn't cause the problem of the total circuit.

We wired the Coulomb counter on another breadboard and ran a similar test. And as expected, a pulse was sent by the 555-timer every so often. As the voltmeter, this circuit had some problems when dealing with voltages around the 5 volt threshold. This caused the reset line to become stuck at a certain voltage level, which caused the Coulomb counter to not output as intended. While this is a serious complication, this could not cause the the problem with the PCB.

We began to scout the schematic and the footprints. After many hours of searching, a mistake was spotted. The rectifier diodes that was part of the surge protection for the battery controller IC, was connected to the wrong pins. This caused the battery controller to receive towards the IC ground connection.

The fourth and final iteration To solve the problem with the Coulomb counter, we tested various resistors and small changes to the circuit. We uncovered that the instrumentation amplifier worked extremely well. The fault was with the output of the differential amplifier, at some points it would not reach the full supply voltage. We began researching why this problem occurred and found a problem with single supply voltage for operational amplifiers. The single supply voltage produces a shift in some operational amplifiers. This causes a certain voltage zone that will not register [92]. A choice was made that the whole circuit could just be ran at 15 volts. This would of course cause problems for the Arduino, which is why a voltage divider of the output signal was necessary. In Fig. 5.23 a measurement of the Coulomb counter integrators output and the input to the 555-timer is shown. The green signal is inverted and extended by the 555-timer, which outputs a signal to both the Arduino and the reset line.

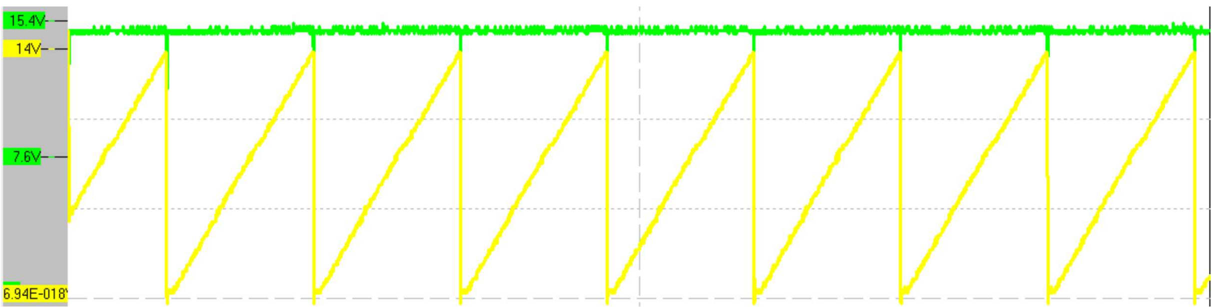


Figure 5.23: Measurement of the Coulomb counter integrator in yellow and signal into 555-timer in green

With the same knowledge, the voltmeter was adjusted to have 15 volts as well. To control that the output voltage would not reach above a certain level, a adjustment to the voltage reference was made. The voltage reference was trimmed up to 11.5 volts. To also increase the measuring field, a amplification was added to the differential amplifier as shown in Eq. 5.2. The voltage level between 12 and 14 volts, would be measured between 1 to 5 volts. The final circuit is shown in Fig. 5.24.

$$A_G = \frac{R2}{R1} = \frac{20K}{10K} = 2 \tag{5.2}$$

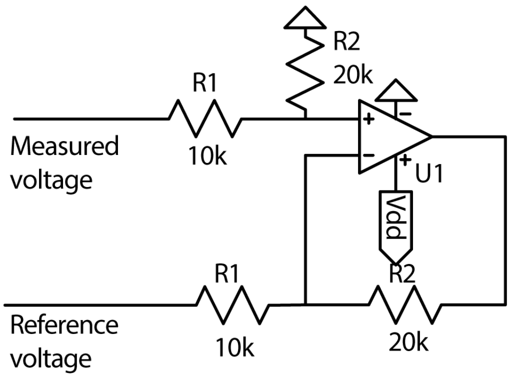


Figure 5.24: Volt meter circuit

To avoid shorting the batteries when using the relays was now a problem, since we no longer took use of the LTC4953 integrated circuit. To work around the problem, we implemented two diodes to the controller that was rated for 2x80 ampere. By parallel coupling these we would accomplish 160 amperes and should be sufficient. The diodes would solve the issue with current passing from one battery to the other, the diodes introduce a voltage drop of 0.7 volts. The voltage drop could cause a problem due to the internal resistance of the battery. If the batteries are under large loads and high currents are being drawn, the voltage drops slightly. The voltage drop is not necessary a problem alone, but combined with the drop across the battery it might. The DC/AC inverter would have problems supplying the necessary voltage to run the computer. When taking this into consideration, a loss of 0.7 volts becomes a huge loss. This caused us to only have the current run trough the diodes, while a switch is necessary. The final circuit for the battery switching without the controls is shown in Fig. 5.25.

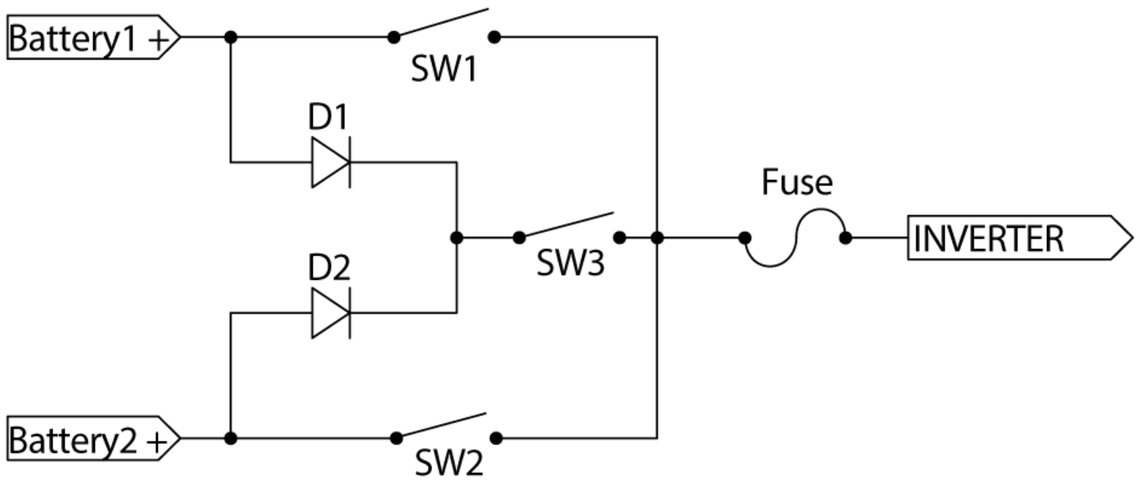


Figure 5.25: Circuit drawing of the relay overview for the battery controller

When designing the new PCB we avoided as many external wires as possible. The Arduino that would be controlling the circuit, was placed upside-down on the circuit board. In difference with the first PCB, traces that required more current than others, were enlarged and a ground plane was added. The ground plane creates a trace as large as possible, which allows trace calculations to be less important. Most PCBs today uses this feature. As lessons were learned form the first PCB construction, a triple and quadruple check of the signals and traces was done. The final traced PCB is shown in Fig. 5.26.

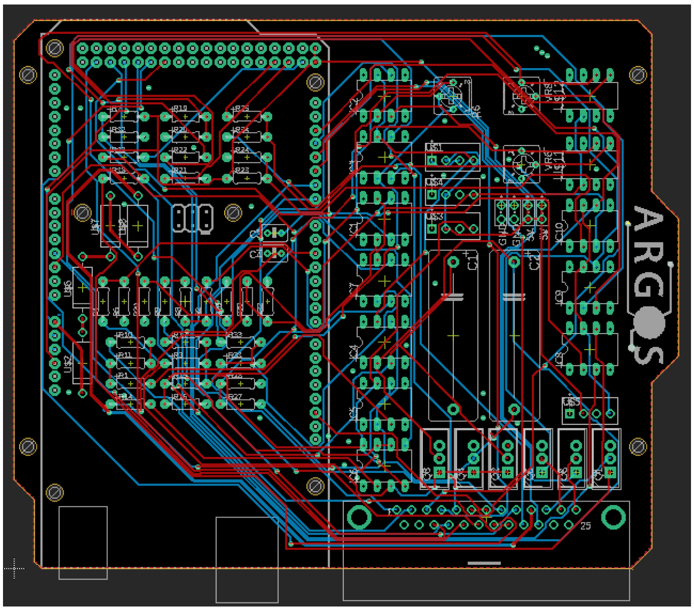


Figure 5.26: The second PCB fully traced

With a small time frame left to complete the project, the components were soldered onto the board and all the smaller parts of the board was tested separately. The test were successful and only the complete test of the system remained. The finished board with components and the Arduino mounted is shown in Fig. 5.27.

Figure 5.27: Picture of the second PCB with components

5.5.1 Controlling and Monitoring the Power System

To operate the battery monitoring and control circuit, an Arduino was coded to preform the necessary actions. Since the switching and start-up of the controller, were to happen automatic, many variables had to be accounted for.

Monitoring and calculating the state of charge Since the Coulomb counter sent a pulse whenever the differential amplifier had reached the maximum capacity, an interrupt in the Arduino code made it very easy to register. By registering how many pulses that would be registered, a estimation of how much charge had depleted from the battery. However, we had to know how much charge the battery had when it was connected.

The voltmeter in the battery monitoring gave a signal the fluctuated from 1 to 5 volts, depending on the charge of the battery. By reading this signal with a analogue input, we were able to calculate the voltage on the battery. Since we knew the voltage on the battery, we would be able to calculate how many pulses was remaining on the battery. When we had an estimation of how many pulses remained, it became possible to estimate the charge at any time.

Switching batteries When switching from one battery to the other, it was very important that the batteries would never be directly connected. So to switch from one to the other, a half-step had to be introduced. During the half-step, neither of the batteries would be directly connected, but both the batteries, would have current flowing trough the diodes. This could cause a small voltage drop of 0.7 volts, but only for a very short time.

To switch between the different stated, several cases were defined. The cases decide whether the relays would be active or not. By defining each of the outputs in every case, the batteries would never be allowed to be fully active at any time.

When the switching from one battery to the other was activated. The code would memorise what the starting case was, before switching from one step to the other after a defined time. In Fig. 5.28 a illustration is showing the output for the different cases. C0 is when the power is connected and neither of the batteries are on. When the power button is pushed, the Arduino checks which battery has the most charge, and activates it.

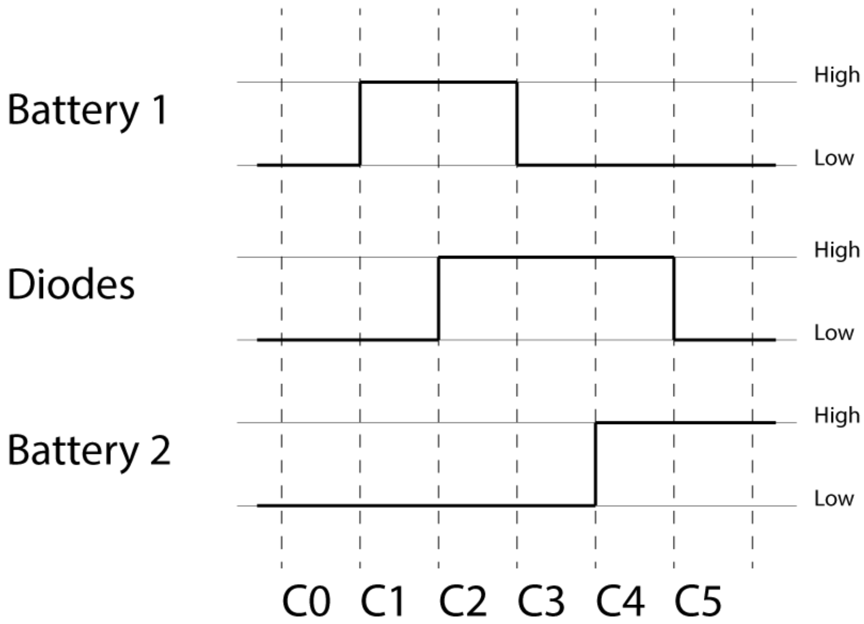


Figure 5.28: Illustration of the different cases

When switching from battery 1 to battery 2, the Arduino would remember C1 as the starting case, before beginning the switch. After a certain amount of time, the case would go from C1 to C2 before allowing more time to pass. When the set amount of time has passed, a switch from case C2 to C3 is done, before waiting again and moving from C3 to C4. If it were to switch from battery 2 to battery 1, the case would move from C4 to C3 to C2 and finally C1.

5.6 Building the Temperature Controller

5.6.1 Voltage Regulator Design Process

The design of the voltage regulator has gone through six notable iterations.

Iteration 1 The first iteration of the voltage regulator was based on the design shown earlier in Section 4.7.2 Fig.4.54. The simulated circuit assumed that the we could produce an adjustable DC voltage reference from the Arduino; this turned out to be incorrect. Most Arduinos use pulse width modulation (PWM) to supply analogue power. PWM does lower the average voltage, but the voltage taken at a random instance is always binary (high or low). In order to have a constant adjustable voltage a digital to analogue converter (DAC) is needed. The Arduino Due has two built-in 12 bit DACs. The challenging part about using these DACs is that they can only provide a voltage that is between 0.55V and 2.75V. A circuit was made to condition the signal to move the range to 0V-5V. This was done using op-amps. However, the conditioning circuit was unstable and was placed on hold, at least temporarily. It was replaced by using a potentiometer to divide 12V supply voltage down to the desired voltage. The resistors into the 555-timer's trigger and threshold inputs were changed to accommodate for the change. The gate driver of the N-channel power switching MOSFET had to be changed from the initial plan.

In order reduce the current when testing, two large 6Ω power resistors were used in series as the test load. The desired reference voltage was set to 0V at the beginning when everything was connected correctly and the circuit was turned on. When the reference was raised the output was also raised by the same amount. However, we quickly discovered that the MOSFET became extremely hot while it was running. This was not a problem when the reference was either completely open or closed. Power is lost in the MOSFET when it switches from low to high. This power loss in the transistor becomes heat. The switching time of the MOSFET should ideally be a few hundred nano seconds. Since the MOSFET is switching one hundred thousand times per second we hypothesised that the reason the MOSFET became so hot was because the switching time being too long.

Iteration 2 In an attempt to reduce the switching time of the circuit we decided to replace the N-type MOSFET with a P-type MOSFET, as shown in Fig. 5.29. The P-type MOSFET can be configured in such a way that it only requires 12V to open the gate, meaning that much fewer components are needed in order to drive the gate. In order to increase the heat dissipation the P-type MOSFET was connected to a cooling block. When switching on the power of this iteration we discovered that the time to open became much improved, but off time became much longer. This caused the output to become unstable.

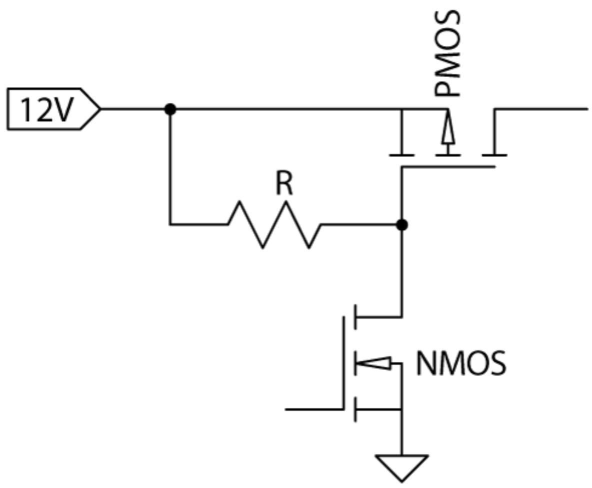


Figure 5.29: Iteration 2 changes

Iteration 3 In the third iteration the resistor shown in Fig. 5.29 was replaced with an N-type MOSFET in order to reduce the time it takes to close the MOSFET, as shown in Fig. 5.30. This did reduce the off time, but it also caused brief short circuiting during switching. This happened because the transistors used to open the gate were not in synch which caused a brief period of time where both of the MOSFETs where open.

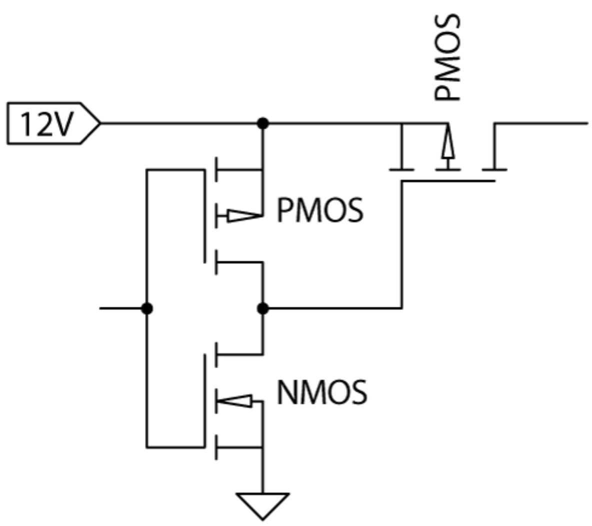


Figure 5.30: Iteration 3 changes

Iteration 4 The fourth iteration scrapped the whole switch mode power supply idea. It was instead based on a more traditional amplifier design. The circuit is shown in Fig. 5.31.

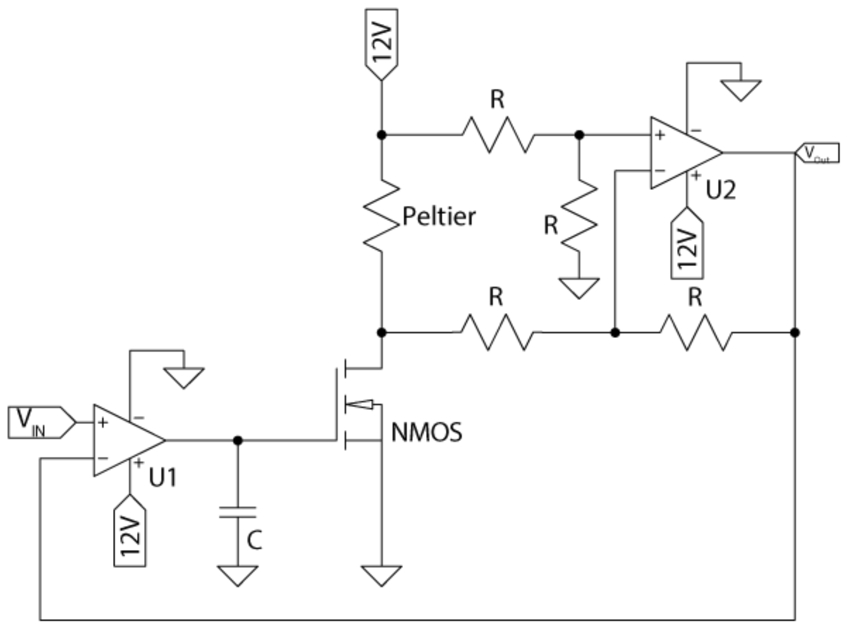


Figure 5.31: Iteration 4

This design uses feedback from U2 into U1 to open the mosfet just enogh so that the voltage accross the peltier element is the same as V_{in}. U2 is a differential amplifier where the output is the same as the voltage across the Peltier element.

$$V_{out} = 12 - V_{DS} \tag{5.3}$$

Where V_{DS} is the drain to source voltage of the MOSFET. The output voltage is connected to the inverting input of U1. U1 essentially adjusts its output so that the inverting (-) and non-inverting (+) inputs become equivalent. The capacitor (C) is added to the gate of the MOSFET in order to reduce the speed of which the op-amp (U1) can change. The speed needs to be reduced because the linear region of the MOSFET is very narrow. The gate voltage will increase and decrease too quickly without the capacitor leading to an unstable output voltage. The power loss in the MOSFET is

$$P_m = \frac{(V_{cc} - V_o)V_o}{R_L} \tag{5.4}$$

Where P_m is the power loss in the MOSFET, V_{cc} is the voltage supply (12V), V_o is the Voltage across the load, and R_L is the load resistance (1.5Ω). This leads to a much more predictable and stable power loss which reaches its maximum at the half way point (6V), as shown in the plot in Fig. 5.32.

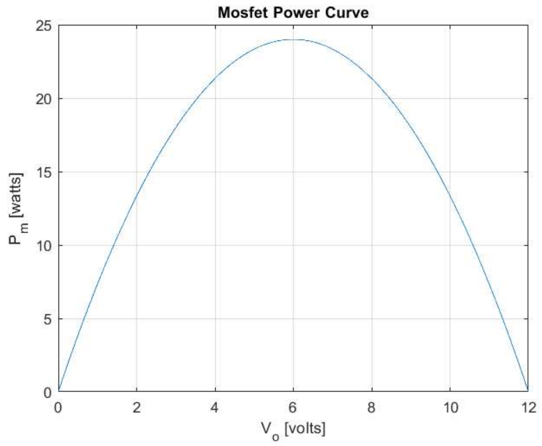


Figure 5.32: MOSFET power curve

Iteration 5 In the fifth iteration we removed the capacitor at the gate of the MOSFET and added a comparator (U3), as shown in Fig. 5.33. Hysteresis was introduced to the comparator to increase the resistance to noise. In short, hysteresis means in this case is that the required input voltage to change states is dependant on the current state. A 5V PWM signal is sent into the comparator which then amplifies the input signal into the gate driving op-amp (U1).

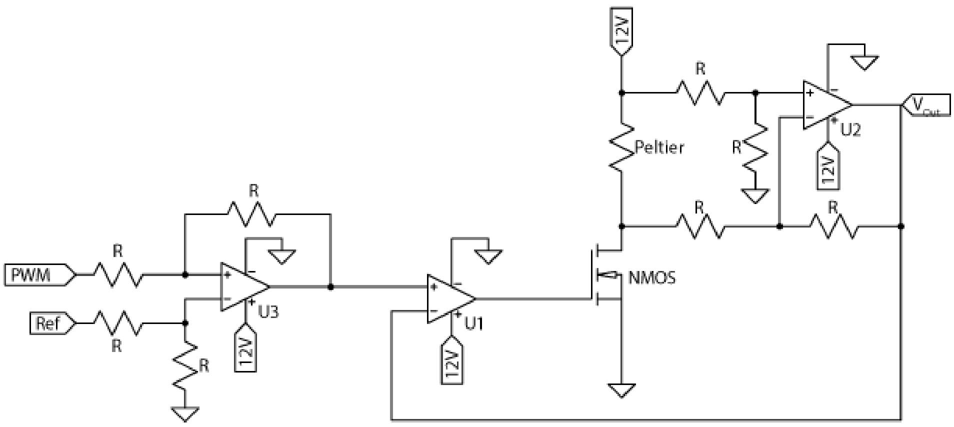


Figure 5.33: Iteration 5

Iteration 6 Now that the cooling can easily be regulated, we still needed to have the ability to heat. To do this in a simple way we used two relays set up such that the orientation of the load becomes switched when a signal is sent to the relay, as seen in Fig. 5.34.

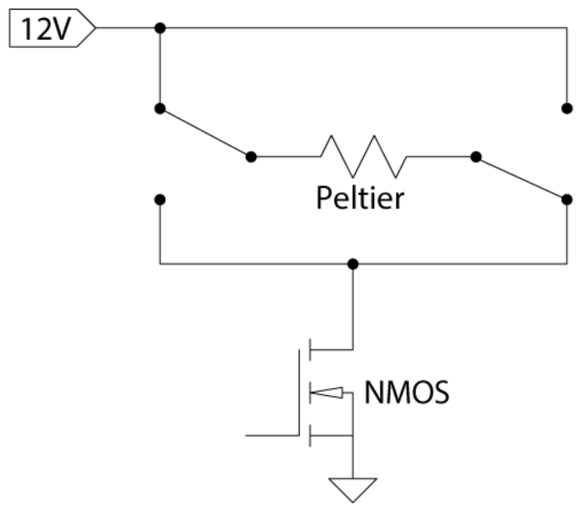


Figure 5.34: Iteration 6 changes

5.6.2 Sensor Integration

Considering the –fairly wide– temperature range that the cameras are rated for (5°C to 45°C), the sensors do not have to be extremely accurate. The most important features the sensors is that it needs to be able to be set up quickly and easily interface with the microcontroller (Arduino Nano) used to control the temperature. The speed of which the temperature can be regulated accurately will ultimately be determined by the response time of the temperature sensor. The rate of change in temperature caused by the TEC must be slower than what the sensor can pick up. If it is larger then the actual temperature will begin oscillate. For example if the temperature needs to be increased, the controller will believe that the temperature is lower than what it actually is; leading to the controller increasing the temperature even more than what it should. The sensor temperature will always lag behind the true temperature. When the controller believes that the desired temperature is reached it will turn off the TEC, but the temperature that the sensor reads will continue to rise even though the TEC is off. This leads to the TEC being off for too long and will cause the temperature to fall too far. Which in turn causes the TEC to raise the temperature again once the controller notices. This cycle is what causes the temperature oscillations.

With this in mind we decided to use a temperature sensor that is quick to respond to change in temperature. A sensor which also has a library for Arduino that is simple to set up and utilize. The temperature sensor we chose was the DS18B20+. The DS18B20+ is a digital thermometer with a configurable resolution of 9 to 12 bits [101]. It has a wide temperature range (-55°C to 125°C) that our intended operating range (10°C to 40°C) fits nicely in the middle of. The sensor has a library for Arduino that is called DallasTemperature [102] that talks to the sensors and converts the bits received from the sensor into degrees Celsius. The DallasTemperature library uses the OneWire library [103] which handles the 1-wire serial interface with the sensor.

The system also requires a humidity sensor; although it is not critical to the operation of the system. Measuring relative humidity can quickly become a complex affair. The transducers necessary to measure humidity are capacitive, which is much more difficult to measure than resistive transducers. Transducers are devices that convert one form of energy to another. The humidity sensors have a very low sensitivity of 30fF/%RH [105] ($3 \cdot 10^{-14}$).

Considering all of the diffculties around humidity sensing, and the non-criticality of the sensor, we chose to use a humidity sensor module that –much like the temperature sensor– is simple to interface with. The sensor we decided on using was the DHT22. This can measure the relative humidity from 0% to 100% with an accuracy of $\pm$2% [104]. The DHT22 has an easy to use library for Arduino made by the producers of the sensor –Adafruit– called DHT-sensor-library.

5.7 Serial Communication

The communication between the battery controller, temperature controller and the Argos software is all done over serial. More specifically using the RS-232 serial data communication standard. The RS-232 is built into Arduinos as the standard communication method. RS-232 requires three wires RX, TX, and ground. Since we have to communicate with two Arduinos inside the camera boxes and the Argos software we decided to use an Arduino Mega. The Arduino Mega has four serial ports, where one is also used for USB serial communication. The initial plan was to use one Arduino Mega as the communication central for all of the other Arduinos. However we decided to incorporate the communication code into the battery controller Arduino in order to reduce the amount of unnecessary complexity.

The Arduino Nanos inside the temperature controllers measures both the temperature and the humidity inside of the camera boxes. The the values for each of them are stored as floating point variables (floats) inside the Arduino. The transmission sequence is displayed in Fig.5.35. To transmit the floats from the Nano to the Mega a byte type pointer is created to the address of the floats. First the type of information that is being sent is transmitted. This is either an 'h' for humidity or a 't' for temperature. The next byte sent indicates which camera box the information stems from. Indicating this in the transmission means that the cables connecting the Nanos to the Mega can be connected to either serial ports. This makes it so that there is one less thing to think about when setting up the entire Argos system, ultimately making the process faster and easier. The pointer is created earlier is then used to transmit the four exact bytes that the floats are stored as inside of the Nano. The last byte to be sent is a 'd' to indicate that the transmission is done.

On the receiving end of the serial communication, the Arduino Mega has to do more than the Nano. The Mega has to distinguish between data coming from the front and rear camera boxes. It has to interpret the data correctly. It also needs to make sure that the data received is correct and valid. The beginning of the receiving function is shown in Fig.5.36. When the Arduino Mega has received six or more bytes it will start the data

```
void sendData(float info, byte type) {
  byte* data = (byte*)&info;
  Serial.write(type);
  #ifdef front
  Serial.write('f');
  #endif
  #ifdef back
  Serial.write('b');
  #endif
  Serial.write(data, 4);
  Serial.write('d');
}
```

Figure 5.35: Information transmission function for Arduino Nano

interpretation process. It starts off by reading all of the bytes received until it reaches a 'd' and places the read bytes into a buffer. This buffer is then used to recreate the floating point variable sent from the Nano using a float pointer. The Mega then uses the first two bytes inside the buffer to determine the type and origin of the data.

An *if* statement is used to ensure that the data is atleast within the ranges of what the sensors can read. Finally, when updating the data stored on the Mega a digital first order IIR filter is used to slow down the change in the stored temperature and humidity. The filter reduces the impact of stray readings that are much smaller or larger than the current value. Reference [106] contains much more information about digital filters, if one is inclined to read more about it.

```
void retriveInfoS1() {
  static byte s1_buff[6] = {0};
  if (Serial1.available() >= 6) {
    Serial1.readBytesUntil('d', s1_buff, 6);
    float* data = (float*)&s1_buff[2];

    if (s1_buff[0] == 't' && s1_buff[1] == 'f') {
      if (*data <= 150 && *data >= -50) {
        temp.front = (*data) * k + (1 - k) * temp.front;
        Serial1.write('k');
      }
      return;
    }
    if (s1_buff[0] == 'h' && s1_buff[1] == 'f') {
```

Figure 5.36: Part of the information receiving function for Arduino Mega

All of the data that the Argos software requires –the temperatures, the humidities, and the battery charges– is sent in a longer but simpler way than what the Nanos do. The Mega simply sends an indicator of what the data is and then sends the floating point data as a string that is readable by a human.

5.8 Implementing Software Models

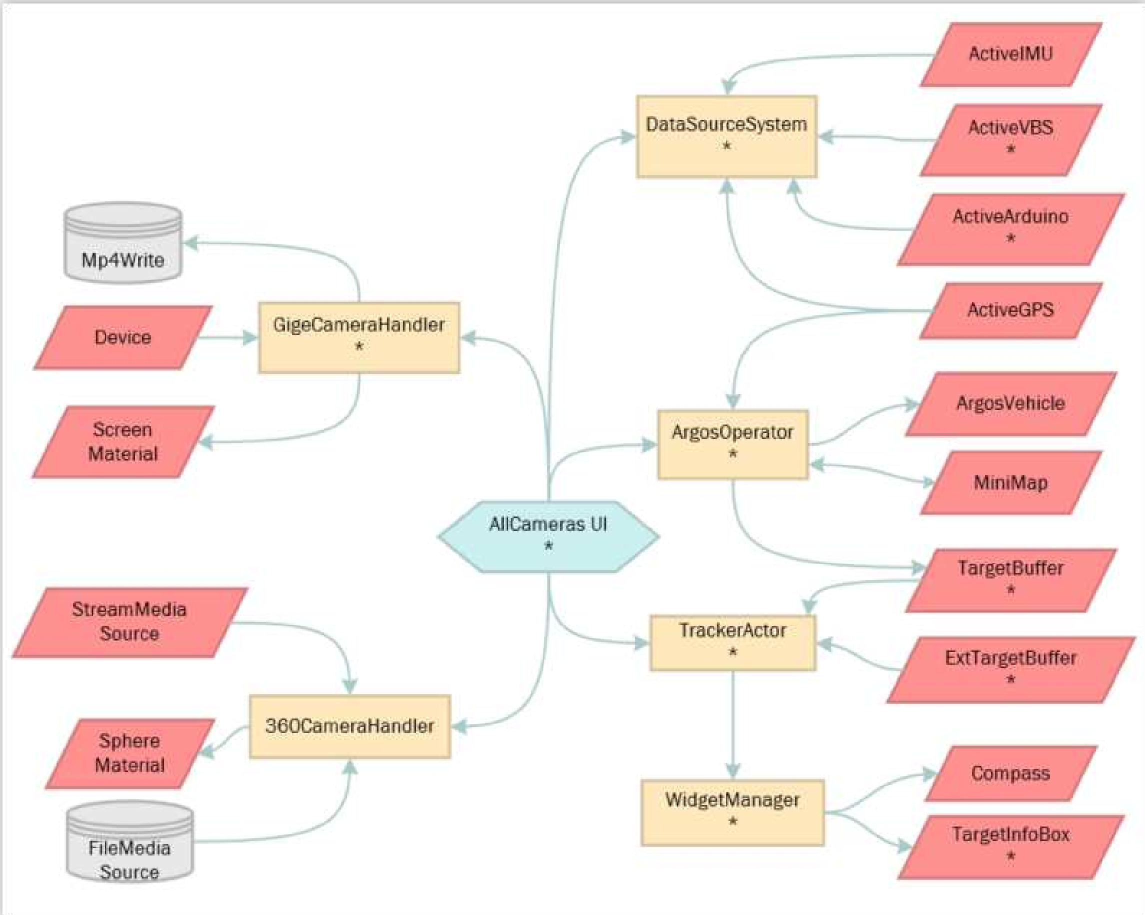


Figure 5.37: Complete Software Architecture

In order to get a complete view of the Argos Application and how much of we had to modify to resolve all of the software problems, we designed a complete software architecture diagrams. The diagram builds on the more specific sequence diagrams for each use case, shown in Fig. 5.37. The Architecture diagrams allows use to more easily see that we actually had to work on all but one controller, and making changes to 5 of the 16 data objects. The diagram also, even though it is an abstraction, illustrates how complex the application truly is.

5.9 Implementing Communication with Battle Management System

When we worked on the Virtual Battlespace (VBS) plugin we started by looking into the functions of the software. We used the similarities to Arma 3 as a stepping off point to get an understanding of the basics. After testing the Arma functions to get information from the player character, we had some realizations. We found out that although the there are similarities between VBS and Arma, the nature of the software is so different that most of the functions used in the first-person shooter game doesn't work for the simulation software that is VBS.

We started to do some extensive research into the functions that's made for VBS[109]. We found out that all coding could be written in sqf files located in the mission folder. We also found out that there are a vast number of functions and a lot of them have no descriptions. Based on the description that do exist, their seems to be many functions that do the same thing. A realisation appeared after many hours of testing that there are some small but important differences between the functions. With better understanding of how to work with the VBS, we tried to get information about other units and vehicles and display them on the screen. This was more challenging than first anticipated because of all the possible information that is connected to each vehicle and unit.

After testing most of the functions that we were planning to use, we started to look at the plugin template that were in a VBS system folder. This template had some comments, but they didn't describe how to use what was already there. We tested the ExecuteCommand template function with some functions that already had been tested directly in VBS. In the beginning we only got crashes. After further testing and reading, we found out that there was a sqf call function that could be used. This made it possible to call a sqf file from the C++ plugin code. Later we found out that everything could be done in the same scope by writing inside a string and use backslash for line separation.

while working on the sqf functions in the plugin, we started looking for each function that would return the information that we wanted. For example, when we are getting the coordinates, we used getPos to get the in-simulation coordinates and then we used the posToCoord function to get the real coordinates that we can use in Argos. This is shown in Fig. 5.38 from line 262 to 264. We also format all the information into a standard formatted string, so it is easy to parse the information afterwards. Shown from line 268 to 270. After many tests we found out that the last called variable is the return value of the ExecuteCommand function

```
259    ExecuteCommand("allVs = ''; \
260    { \
261        _vName = _x; \
262        _vPos = getPos _x; \
263        _realVPos = posToCoord[[(_vPos select 0), \
264            (_vPos select 1)], 'LL']; \
265        _objType = 'Vehicle'; \
266        _vType = objectType _x; \
267        _team = side _x; \
268        _vInfo = format['%1,%2,%3,%4,%5,200,%6;', _objType, \
269            _vName, _vType, (_realVPos select 0), \
270            (_realVPos select 1), _team]; \
271        allVs = allVs + _vInfo; \
272    } forEach allVehicles; \
273    hint allVs; \
274    allVs;", sendBuffer, sendBufferLn - 1);
```

Figure 5.38: Script for VBS in plugin

When all the sqf code were close being finished, we started to work in the connection between the Argos software and the VBS plugin. We started to look at the WinSock library. We choose to use this library because it is part of windows, so we didn't need to download a third-party library. Without any knowledge about any other library before we started, so this was the most practical choice. Because we don't want the plugin to be larger than necessary, we concluded that a windows library was the best choice. We looked at the WinSock example server code as a starting point and worked with it to make it work as wanted[110]. After we had tested the server separately form the VBS plugin, we merged it into the plugin. When we started the software, we realised that we needed to run the server on a different thread, because everything froze as soon as the plugin started. We used the thread library which is a part of C++ to solve this problem[108].

After some problems with de-syncing between Argos and the VBS plugin server, we made it possible to shut down and restart the server without needing to restart VBS. The server starts on the first simulation step and a Boolean is set to true. This Boolean indicates if the server is running or not. If this Boolean is false the server isn't running, and it will start on the next simulation step. If the client disconnects from the server, it will set the Boolean to false and restart the server. The server can be shut down as well, by pressing a button that is pre-determined in the mission. When this button is pressed serverExit will be stored in server State. This will shut down the server and the user can press a separate button to start the server again. When the server receives "getV" or "getM" it will return a list of vehicles or markers respectively. This list does also contain name, object type, subtype and coordinates. For vehicles it also returns the team it is a part of. The teams are split into three and the team names is decided by the VBS software.

Thanks to how data data is formatted by the VBS plugin before sending to Argos, it becomes very practical fort Argos to split the recieved string with commas as a delimiter as shown in Fig. 5.39.

```
if (InData.ParseIntoArray(TSArray, delim) < 6)
    return false; // ParseData deals with logging
float Latt = FCString::Atof(TSArray[3].GetCharArray().GetData());
float Longt = FCString::Atof(TSArray[4].GetCharArray().GetData());
float Altt = FCString::Atof(TSArray[5].GetCharArray().GetData());
```

Figure 5.39: Code for parsing string received from VBS

Since one of the main goals for Argos being able to communicate with VBS is for demonstration purposes, it was very important for Argos to reliably record the data received from VBS. Thanks to a pre existing DataSource system, all we needed to do was to deal with the logic of formatting the target data into a single string, this cam be achieved as shown in Fig. 5.40.

```
return FString::Printf(TEXT("%d\t%d\t%s\n"),
    static_cast<int>(this->Timestamp.ToUnixTimestamp()),
    this->Timestamp.GetMillisecond(),
    *Buffer->ToLogLine());
```

Figure 5.40: Script for writing VBS targets to recording

Thanks to being able to format the data, however we want when writing the log, we are able know exactly how the data is formatted when reading from the same log later, as shown in Fig. 5.41

```
TArray<FString> TSAr;
InString.ParseIntoArrayWS(TSAr);
tmp = TSAr[2];
this->Buffer = new FExtTargetBuffer(tmp);
```

Figure 5.41: Code for reading VBS target log during playback

5.10 Implementing Target Tracking Bug Fixes in Software

Resolving the issue of the heads up display (HUD) trackers appearing outside of the camera screens, this was a matter of modifying a single value, representing the distance from the operator at which they should be placed. As long as the radius of the camera screens never change again, this solution should be fine. However, in the event that they are changed, this value would need to be updated again. A potentially better solution, would be to read the radius of the camera screens when the markers are spawned, in order to determine both distance from the player, as well as the possible scale-range .

To be able to get the HUD targets to appear in the correct location, we needed to calculate their position relative to the position of the Argos vehicle. By getting the position of the target, as well as the Argos vehicle, we are able to calculate the targets relative position through vector subtraction as shown in Fig. 5.42.

```
FVector TLoc = this->Buffer->ReadBuffer[I].GetCartesian();
FVector VeLoc = this->GPSTarget->GetCartesian();
FVector ReLoc = TLoc - VeLoc;
FRotator rot = ReLoc.Rotation();
rot.Yaw -= GPSTarget->GetBearing();
FVector RoLoc = rot.Vector() * ReLoc.Size();
```

Figure 5.42: Calculating Relative Rotation and Location

Once we have the targets position relative to the vehicle, we then have to account for the vehicles current direction, the reason we have to do this is we want to simulate the operator looking forward to be relative to forward, compared to the vehicle. This means we need to subtract the current bearing of the vehicle, as shown in Fig. 5.42.

```
float dist = ReLoc.Size() / 100;
float scale = FMath::GetMappedRangeValueClamped(FVector2D(100.0f, 5000.0f),
                                                 FVector2D(0.2f, 0.08f),
                                                 dist);
DiamondMarkers[I]->SetActorScale3D(FVector(scale, scale, scale));
```

Figure 5.43: Scaling HUD Trackers

In order to scale the HUD trackers to their distance from the Argos Vehicle, we are able to use the targets position relative to the vehicle as shown in Fig. 5.43. Because the map used in Argos is scaled, when we take the length of the vector, we also have to divide it by 100 in order to get the distance in meters. By using the GetMappedRangeValueClamped function, which is provided by Unreal Engine, we are able to get a scale value, which is proportional to the distance within a specific range. This means that the scale will not become any smaller than when the distance reaches 5 kilometers. The scale also will not become any bigger, once the vehicle is less than 100 meters away from the target.

While all of the issues related to the target tracking, only required small modifications to resolve, we had a challenge with the changes causing them having been largely undocumented. Especially when some of the issues already had code, which at a first glance seemed to solve the problem, it turned out that deprecated values and functions were often used. It also turned out that our only good method for testing the HUD trackers position relative to the GPS was to either play back old recordings with GPS data, or to bring the GPS and drive around. This is because Argos does not use any form of compass besides the direction we get from the GPS. This problem turned into a bigger challenge since the only recording we had with both GPS data and a video recording was not able to play back at the intended 30 frames per second. Meaning that the playback of the GPS would run a lot faster than the video, making it very difficult to compare the two.

5.11 Re-Implementing and Improving GigeVision Recording and Playback in Software

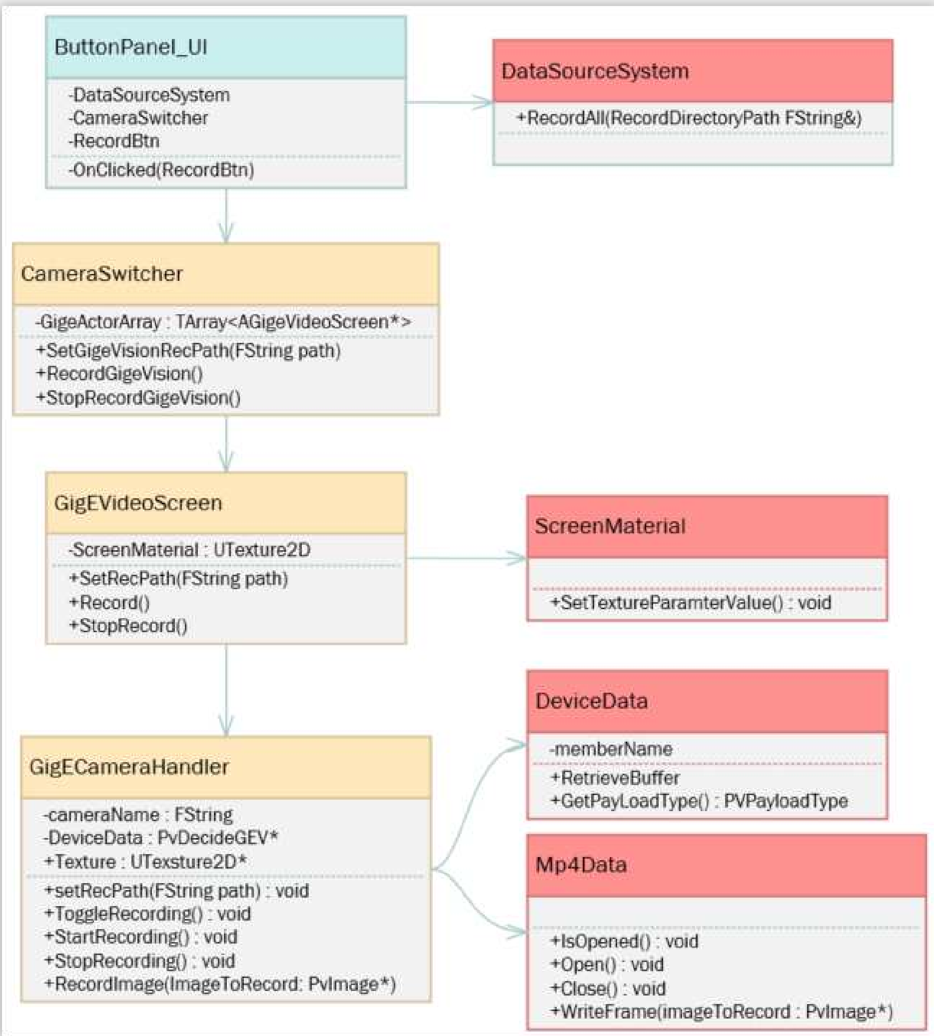


Figure 5.44: GigE Class Diagram for Recording

By utilizing the connection between the user interface (UI) and the camera controller mentioned in Section 4.11 we are able to start and stop recordings by calling a function inside of the Camera Switcher.

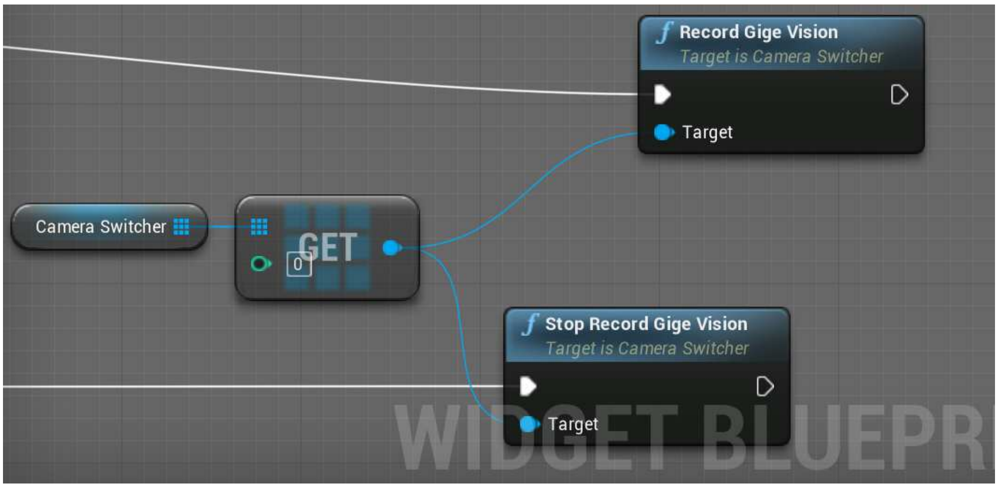


Figure 5.45: Blueprint for GigeVision Start/Stop Recording

In Fig. 5.45 we can see a mixture of white and blue lines, the white lines can be seen as order of operation, once a box has bee executed, it may send a signal though the white line telling another box to execute. To compare this to more traditional code, this may be compared to going to the next line of code after a line has been executed. The blue lines are effectively telling Unreal Engine, whose "Record Gige Vision" we wish to call. The reason we need this is that there could in theory be other objects which has the same function signature. Unfortunately, the UI does not store a direct reference to the Camera Switcher object, instead it stores an array of them. Fortunately we can assume there will only actually be one Camera Switcher object, hence the "Get" box, this effectively gets the first object in the array.

Having re-enabled the functionality to record the images from the GigeVision cameras, we could go on to dynamically storing the recordings, so that they don't overwrite each other. Discussing potential solutions for this, we could create our own folders, or simply adjusting the name of the recorded files. The issue with doing anything like this specifically for GigeVision recordings is that it would cause more work to later play back the recordings. Instead of trying to implement something like that, we chose to instead use the recording system that is already in use for other data. When a recording is started the Record All function is called in the DataSource System object, this will give out a record directory path, which is the path to a folder named after the current time and date. By adding the functionality to set GigeVisions record path through code, we were able to set the recording path after the DataSource System has generated the path, but before we have actually started recording the images.

5.12 Implementing Communication With New Electrical System

Since the software and electrical engineers were able to discuss how the data were to be formatted when sent from the electrical system to the computer. We planed how to format the data in order for parsing the data to be as simple as possible. One factor in the plan is for example that we were able to agree which order the different values, were to be sent, meaning we were able to more confidently know exactly where in the received string any specific value would be as illustrated in Fig. 5.46.

```
int ArgumentCount = sscanf_s(
    StringCast<ANSICHAR, TCHAR>(*LastReading).Get(),
    "tf: %f, tb: %f, hf: %f, hb: %f, b1: %f, b2: %f",
    &tempF,
    &tempB,
    &humF,
    &humB,
    &bat1,
    &bat2);
```

Figure 5.46: Code for parsing data from arduino

Once we had the most recent data from the arduino stored in the application, we are able to use Unreal Engine blueprints to set the most recent values once every frame in the UI object as shown in Fig. 5.47.

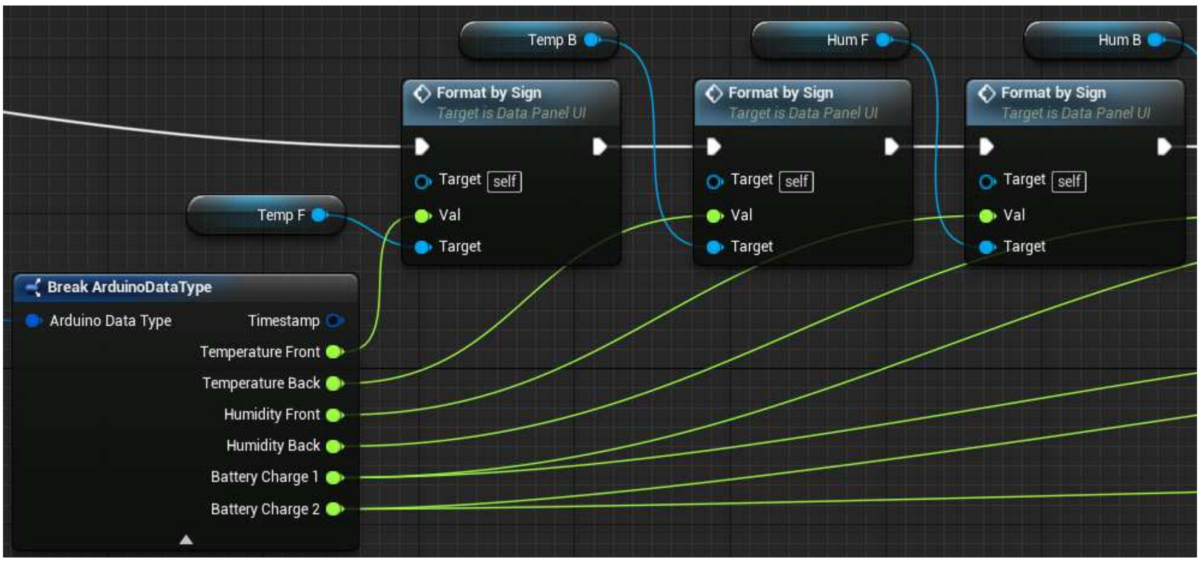


Figure 5.47: Unreal Engine blueprint for setting values to UI object

By making the data panel larger, and filling it with default values, simulating the data we would receive from the arduino, we were able to verify that we could use the same data panel for both GPS and arduino data.

Figure 5.48: Warning message for surpassing accepted thresholds

We had to display a warning, illustrated in Fig. 5.48. The warning would appear when the temperature in the camera boxes goes outside the accepted range, or when the current battery charge in all connected batteries goes under the determined shut down point as shown in Fig. 5.49. If the warning shows up, with the current implementation we decided to leave it up to the user to safely stop driving, and shut down the system.

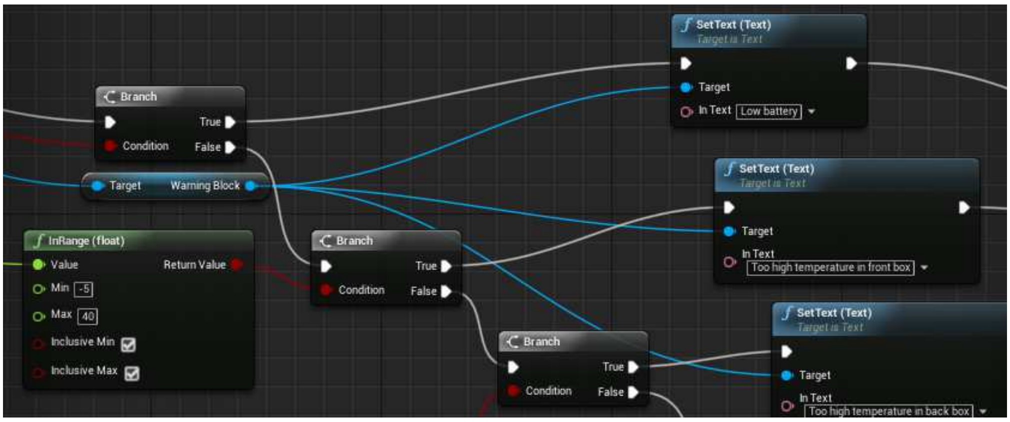


Figure 5.49: Unreal Engine blueprint for generating warning message

Since we wanted to be able to show all data we running playback of a recording, we also needed to define how the sensor data shall be formatted when logging it to a file, as shown in Fig. 5.50.

```
return FString::Printf(TEXT("%d\t%d\t%f\t%f\t%f\t%f\t%f\t%f\n"),
        static_cast<int>(this->Timestamp.ToUnixTimestamp()),
        this->Timestamp.GetMillisecond(),
        TemperatureFront,
        TemperatureBack,
        HumidityFront,
        HumidityBack,
        BatteryCharge1,
        BatteryCharge2);
```

Figure 5.50: Code for writing arduino data to log file

As well as defining how to parse the data from the log during playback, shown in Fig. 5.51.

```
int ArgumentCount = sscanf_s(
        StringCast<ANSICHAR, TCHAR>(*InString).Get(),
        "%d %d %f %f %f %f %f %f",
        &UnixTimestamp,
        &Milliseconds,
        &TemperatureFront,
        &TemperatureBack,
        &HumidityFront,
        &HumidityBack,
        &BatteryCharge1,
        &BatteryCharge2);
```

Figure 5.51: Code for parsing recorded arduino data during playback

5.13 Implementing Official Argos Website Improvements

```
<div class="team-member team-member-name">Vegar Skogen</div>
<div class="team-border PM"></div>
<div class="team-member-title" style="text-align:center;">Prosjektleder</div>
```

Figure 5.52: HTML using CSS subclass

By using the CSS class introduced in Section 4.13, we are able for a single team member to easily specify a subclass, such as PM for the project manager as shown in Fig. 5.52. To guarantee that the issue of lacking year logos would not become an issue anytime soon, we also decided to use one common font, Yantranamov [113]. As well as creating images for both bachelor (yellow), summer (blue) and current (green) for every year since 2015, and all the way to 2030. In the event that project Argos were to keep going for more than the next 11 years, we also decided to keep the adobe illustrator files as available to any future groups. This decision could also become relevant if Argos ever were to be done as anything other than summer or bachelor projects, as it would allow other types of groups to create their own colors.

6 Challenges

6.1 General Challenges

At the start of the project we experienced a circuit break causing us to lose power to our room. It happened a handful of times and after 6 weeks from the first occurrence, an additional power outlet was installed with a new fuse. However, this minor inconvenience made it difficult to test all the equipment that we needed, when we started our project.

When we were assigned a room, we made sure to ask for the largest room available. This was because of all the equipment and system parts that we inherited from previous Argos project. Even though our room is the largest available for bachelor projects, it was still too small for all the equipment. It was a challenge to manage the equipment and space needed for the three disciplines. The lack of space made it hard to work on the electrical and mechanical system parts. Because of the size and weight, it was impractical to move everything down to the workshop, since we could not secure them over night.

When we were planning the documentation and deciding on a methodology, we realised how difficult it was to communicate these differences and deciding on a working method.

Lack of organised information from the university was also a challenge. It was hard to find the right information about the equipment and components that we could use from the university. This included information about labs and workshops being locked during the whole Easter week and not just the holiday. As we were in a critical production phase of our project at the time, this was information that would have been useful to know. Missing equipment in the workshop was also a problem, since some users do not deliver equipment back to its designated area.

There was a lot of sickness and allergies after Easter. This combined with the limited project time, made us somewhat stressed. During the last two weeks of the project there was industrial work outside, which made it challenging to concentrate. Since there was no ventilation either, we were forced to keep the windows open.

We learned that communicating across the disciplines was challenging because of the knowledge gap. We used a lot of time trying to explain and discussing ideas with each other. This became extremely apparent at the end of the project, as software and mechanical were going to help electro with producing a component.

Throughout Argos, various logos and illustrations has been made, including the year-logos mentioned in 5.13. Since there was a lack of documentation, a lot had to be designed from scratch. This is also present with other aspects of previous Argos groups, such as the use of fonts. Presumably some Argos group decided to use one specific font for all marketing material. However, since the decisions for styles had not been documented, we were forced to make decisions, and make most of it from scratch.

6.2 Mechanical Challenges

6.2.1 Roof Mounting Challenges

The time was restricted and as we progressed we noticed that making a carbon fiber roof plate would take too much time. Our plan was to create a large plate but this caused production inconveniences. We would have to borrow the autoclave at KDA, since the equipment on school campus isn't large enough. There was option to wet cast the plate, but it is a long and tedious process we didn't have time for. With inspiration from aluminium profile system we changed our design. We found a company called Rollco that sold aluminium profiles we could use instead of the large plate. Unaware that Rollco was in the middle of changing supplier, which caused our order to be wrong. They were struggling with internal difficulties. Our order arrived with two different profile systems that did not fit together. The t-slot nuts were fitted to a different aluminium profile slot as seen in Fig. 6.1. Dealing with an incorrect shipment meant contacting Rollco and straighten out the situation. After about a week, they replied they would send us new supplies free of charge since the error was on their side. However this postponed our production start by more than 2 weeks. This gave us less time to implement and test than we would like to for the roof mounting solution.

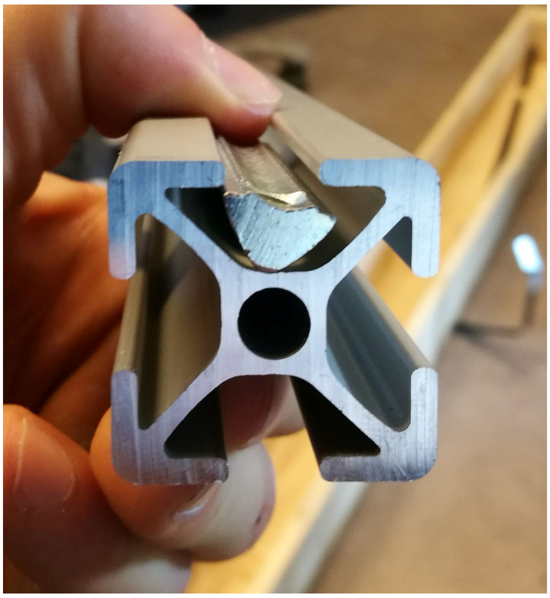


Figure 6.1: Incorrect t-nut for the aluminium profile system we received

As the design developed, the computer files became so complex that the simulations was hard to execute. We had to simplify the analysis at first. After many failed attempts of trying the whole system, we finally got one simulation to work. The analysis took an hour to compute and the computer was struggling after because of RAM usage.

6.2.2 AutoCAD Challenges

At the start of our project we had some problems with SolidWorks becoming corrupt on one of our members computer. After extensive troubleshooting we could not find the source of this problem. We even tried to do a clean install of the software but we ended up doing a reinstall of windows which was our last resort. We also have had problems with getting our analysis through in SolidWorks with regular crashes see Fig. 6.2.

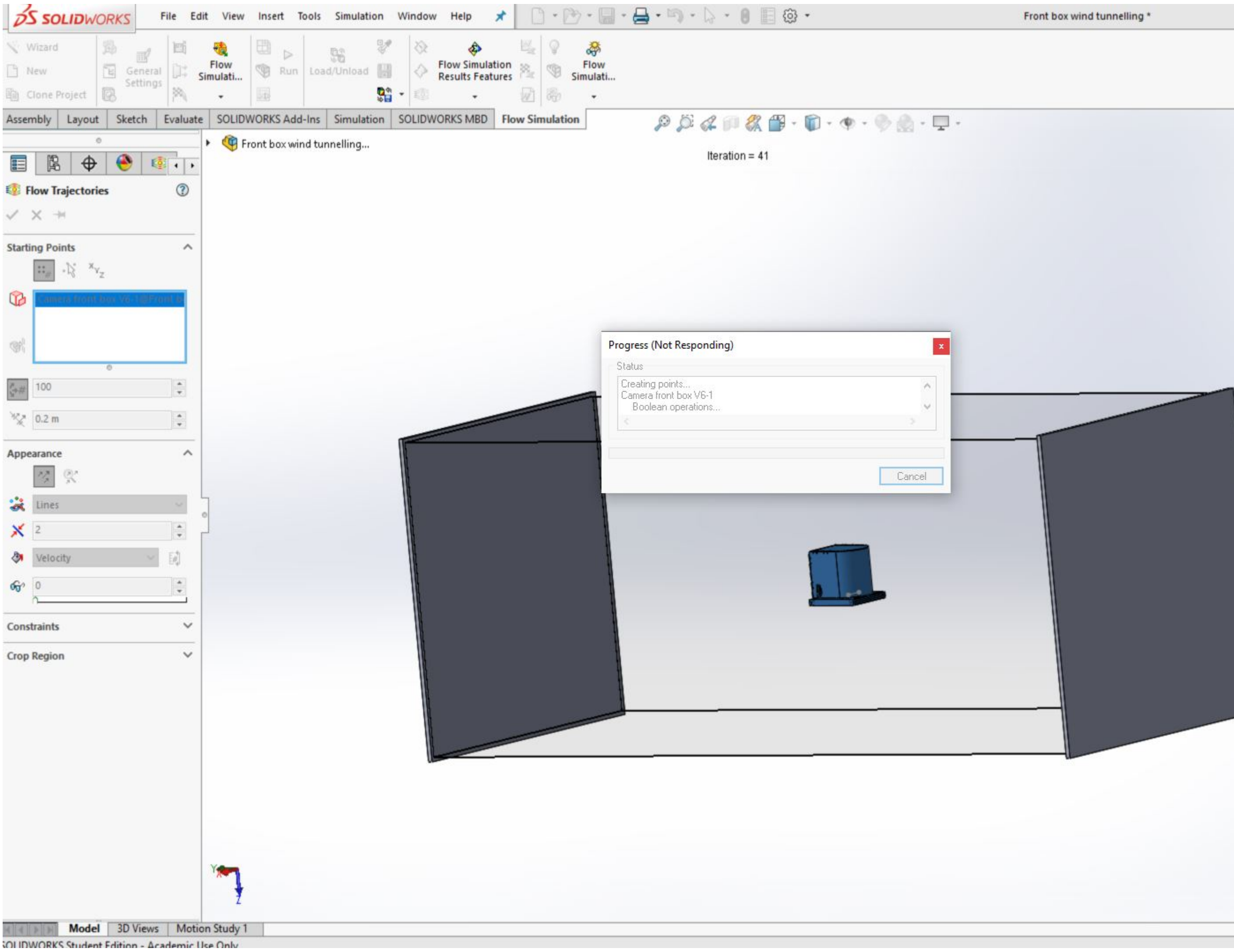


Figure 6.2: Computer error from SolidWorks

6.2.3 Front Camera Box Challenges

The camera lenses has a barrel distortion which causes a distinction between our measured AOV and the stated AOV by the datasheet. This caused us to go though a lot of testing, trial and error before the design of the front camera box was finished. This caused some delay in the production and design of the front camera box.

We started production by creating the metal bracket for the front facing cameras.

Our first attempt was failed because we bended the sheet aluminium a little further than 90°causing the cameras to be angled downwards see Fig6.3.

Figure 6.3: First production of camera stand

Creating a mold for the plexiglas for the front camera box was probably the most significant challenge of the production phase of the front camera box. Since this has an arc with a radius of 100mm we looked for a pipe with this dimension. We could not find one that can handle up to 150°Celsius so we rather decided to create the mold out of laser cut plywood. We bended a 1mm aluminium sheet around it and used clamps holding it together and finished with an aluminium foil to have a smooth surface see Fig. 6.4.

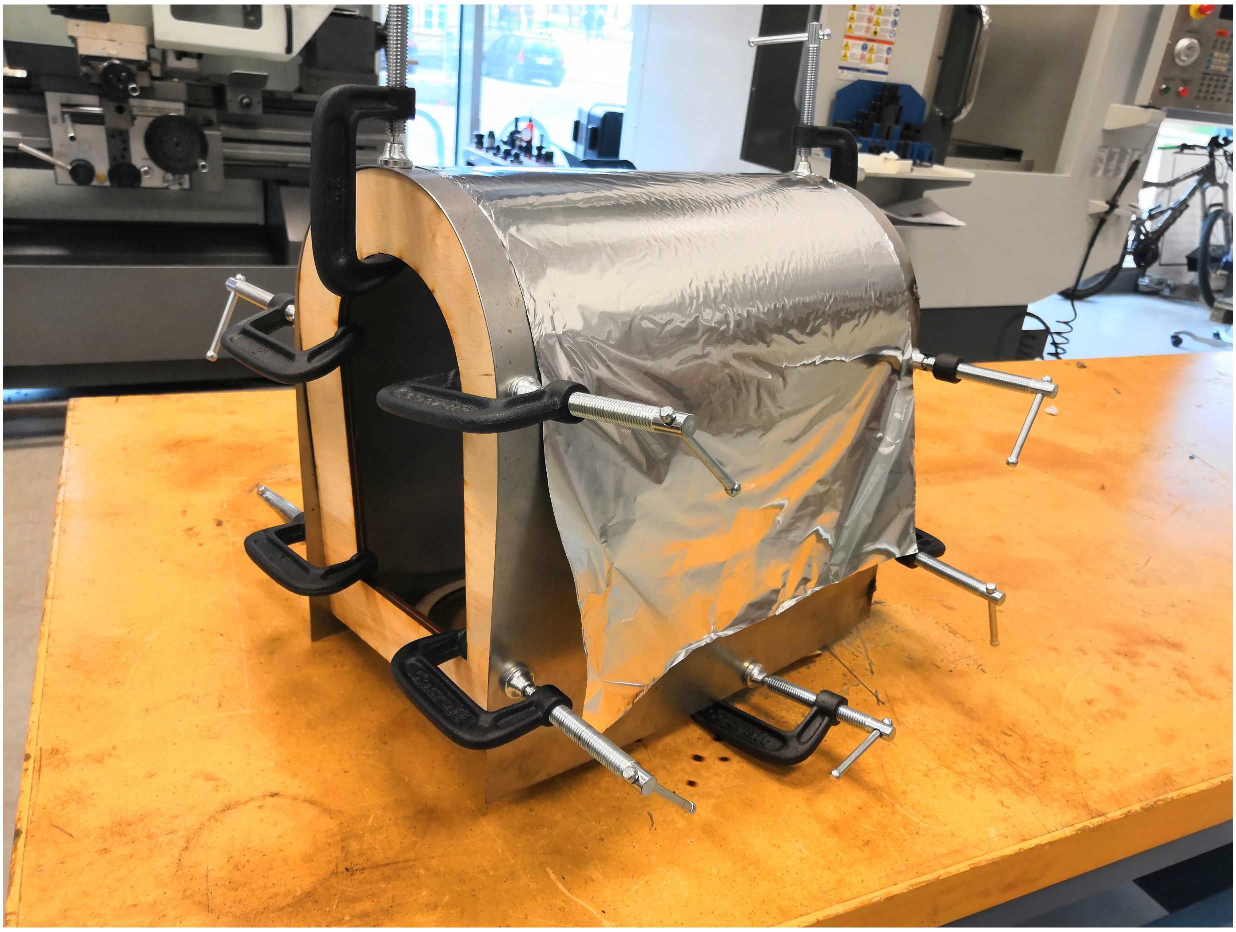


Figure 6.4: Plexiglas mold

With this design we can put the whole mold in the oven at 150°Celsius so the plexi can bend around the mold. Because of the high temperature requirement we had a lot of restrictions with how to bind the mold together. We could either use a specific two component glue or clamps. The clamps had to be full covered metal so it could withstand the temperature. When we put the plexiglas in the oven we noticed though we had an even mold surface, you can see smallest impurities in the surface reflected in the plexi. However luckily this has no affect on the camera view as the impurities in the glass is out of focus. We did some test bending before we did the final one. In our first test bending we did not use enough heat so the plexiglas failed and broke see Fig. 6.5.

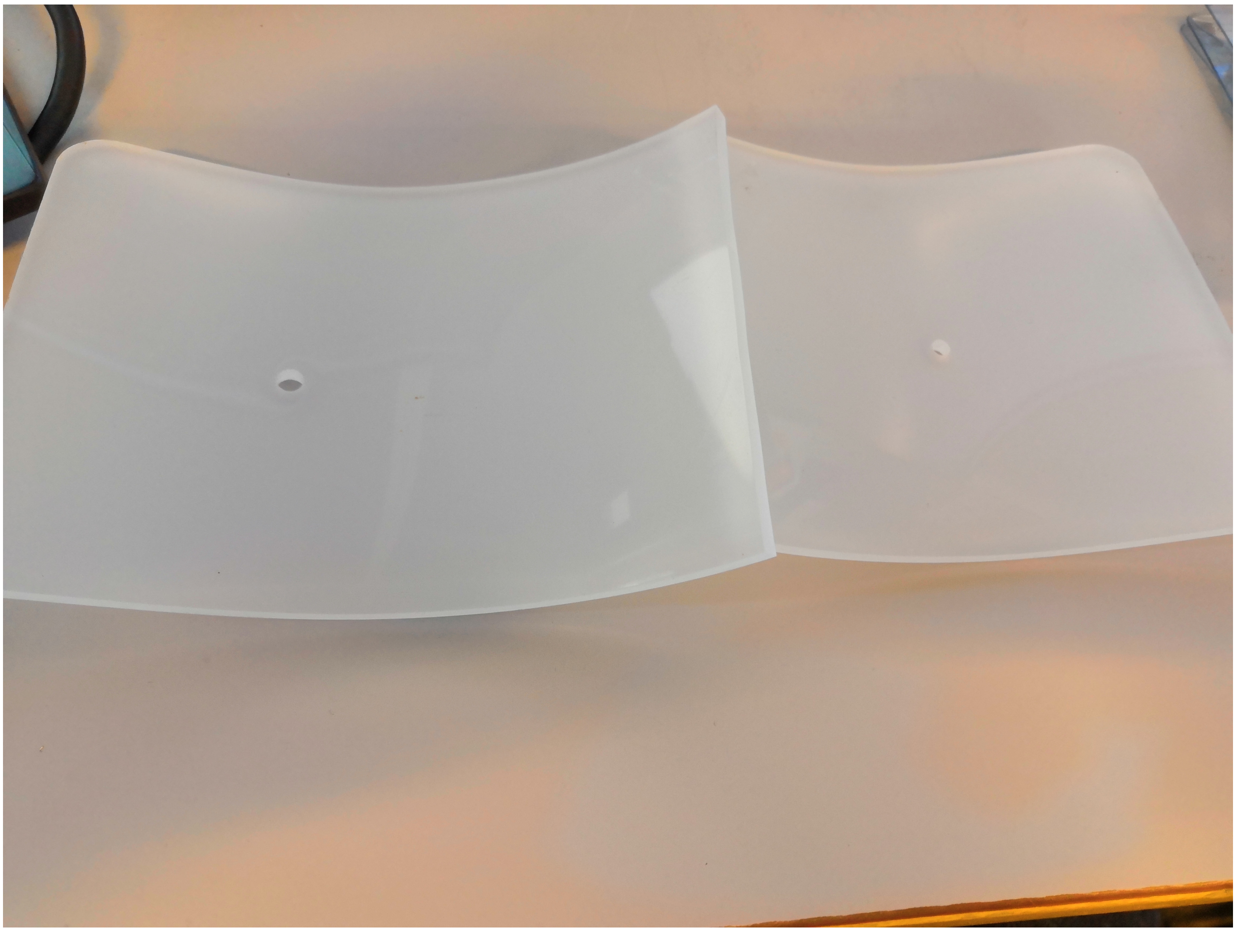


Figure 6.5: Opal white plexiglas first bending test

Aligning the holes to fit the parts and drilling holes for the machine screws proved to be impractical, because our clamping brackets for the drill press was smaller than our part. We had to improvise and use external clamps to use the drill and some places we used a hand drill. Since we produced everything with manual labour and created all the parts we needed ourself. Since we had loose tolerances and all the parts didn't always fit together and we had a lot of post processing to do on all our parts. The aluminium lid ended up a little shorter than the plexiglas after production, so we had to design a 3D print that built up to the back wall to level the plexiglas and aluminium see Fig. 6.6. With this design our thought was to use TEC7 to seal the list to the lid of the front box.

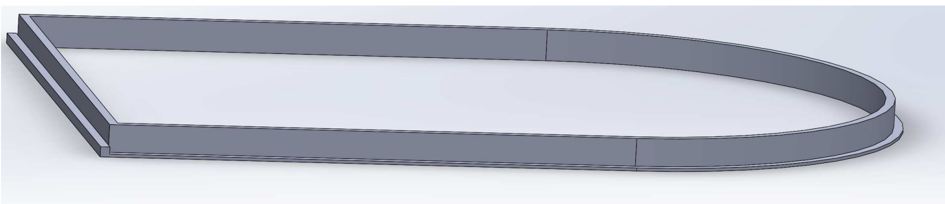


Figure 6.6: 3D printed list for front box

When it was time for final assembly the only thing missing was the eccentric locks from the front box. To mount these we had to drill the last two holes in the plexiglas.

When we drilled the last holes in the plexi the plexi cracked see Fig. 6.7. When we continued to drill the crack grew a little bit. We have tried to use tec7 to seal the crack and stop it from growing and it seems to have worked as a temporary solution. We have ordered new plexiglas as our second mold did not have our desired shape see Fig. 6.8.

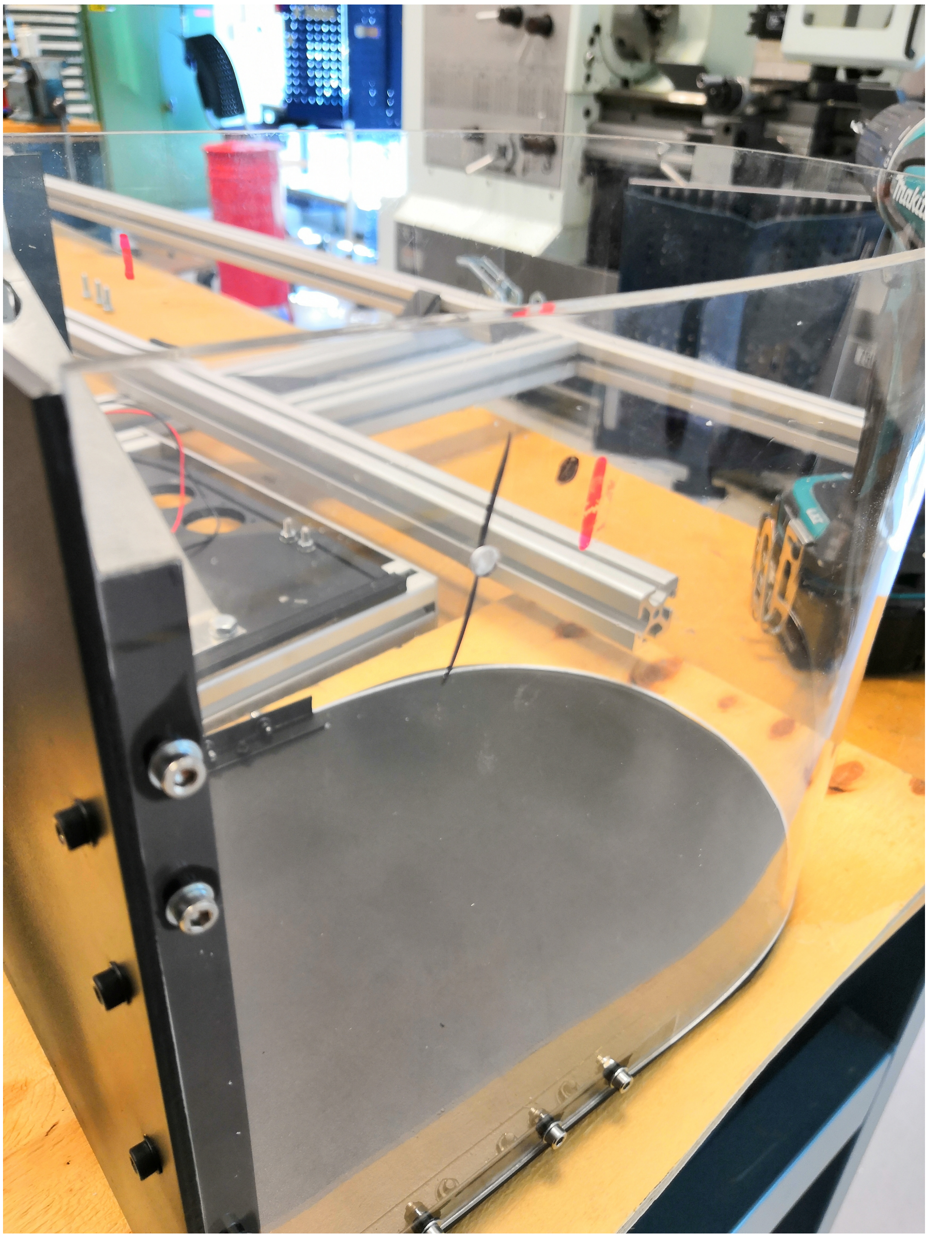


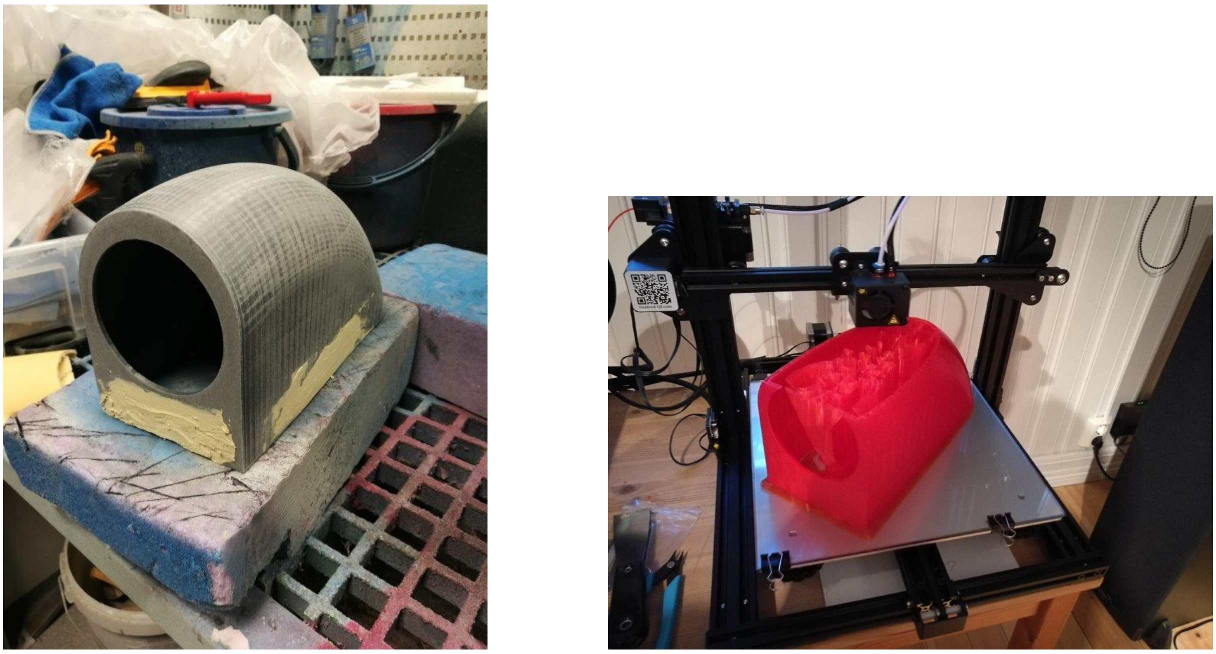
Figure 6.7: Plexiglas crack

Figure 6.8: Second plexiglas bending warping

Since our plexiglas cracked and the new didn't arrive in time for project end we will hand over the project with the cracked plexiglas we recommend to exchange it for a new one. All the 2D drawing will accompany the project so it's easy to create a new bended plexiglas.

6.2.4 Rear Camera Box Challenges

The design of the rear camera box was pretty straight forward, but the challenges began when production started. Creating part of the rear camera box in carbon fiber created uncertainty because of lack of experience. Lab engineers were unavailable for guidance due to busy weeks and travelling when we started production. Problems started when we started 3d printing our mould for the top shell. It was large a large model which run the risk of warping our mould during printing. One print took around 2 days to finish to a failed print was quite frustrating. We 3d printed several moulds and tried fixing with spackle as seen in 6.9a. We were advised to not use spackle on the mould, because it is hard to remove after casting is done. The third printing attempt as seen in 6.9b, was successful.



(a) First 3d printed mould with spackle to straighten the warping

(b) Third 3d print attempt for mould which was successful

Figure 6.9: 3d printed moulds for casting

Due to inexperience we also ran into a problem while casting. The carbon fibre did not want to stick to the front area of the box because of the small surface area. After post processing weaknesses in the cast appeared. The front surface of the rear camera box became so thin that holes appeared as seen in Fig. 6.10. We dealt with this problem by casting a carbon fiber composite plate and created supports.

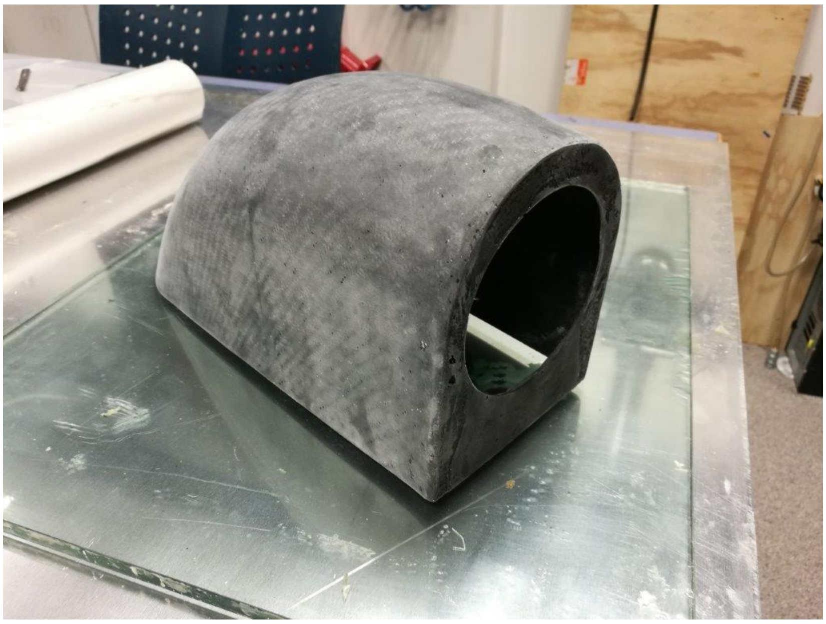


Figure 6.10: Rear camera box after post processing sanding. Holes appeared in the front area of the box

Problems occured while blow moulding the glass dome for the rear camera box. Several attempts were done where many attempts failed. Failures included a combination of incorrect temperature, pressure and moulding method. Blow moulding design was required for a successful attempt. Several domes became crooked due to variable pressure as seen in Fig. 6.11a. During one of the first attempts we even managed to get a vastly misshapen dome due to incorrect temperature and pressure as seen in Fig. 6.11b.

(a) Several domes of varying sizes and crooked shapes

(b) Misshaped dome due to temperature and pressure. Air bubbles has started to form in the material

Figure 6.11: Plastic domes from several blow moulding attempts

6.2.5 IMU Vibration Attenuation Challenges

For designing the vibration attenuation for the IMU we learned that we had no access to a vibration simulation device. Therefore we unfortunately could not simulate which type of vibrations that made the IMU drift. Which means all our testing and effort had to be trial and error without any scientific measurements. There has also been an uncertainty because when in the office we get better results with the magnetometer turned turned off, while driving we get the best result when it's turned on.

6.3 Electrical Challenges

6.3.1 Simulation and Reality

Simulating and doing calculations is very different from operating with real components. When simulating a circuit, the components is often considered ideal, which means there are no tolerances and offsets. When operating with real components, the offsets and tolerances are always given in a datasheet, but requires much more calculations and preparations. Several times through the project, differences were uncovered and had to be accounted for. Some of these required a full redesign of the circuit, while others only required a replacement of the component.

When searching for the correct components, a huge challenge was to take all the small important variables into consideration. When presented with a datasheet, its very easy to overlook small details that could decide whether the circuit will succeed or fail. We uncovered this problem several times throughout the project. However, it was not necessarily the main cause of the problem itself, but forced us to redesign the circuit.

Since the battery controller would control the flow of the current to the inverter, the system had to be rated for very high currents at 150 amperes. Most components are not rated for such currents, which makes it very challenging to find the correct components for the circuit. When a component with the necessary requirements is found, there are still a lot of variables that could affect the outcome and cause it to overheat or explode. In addition, when designing a circuit that would revolve around the high currents, precautions has to be taken to avoid later problems. Several times throughout the project, components or circuitry broke, due to larger currents were introduced to the circuit

6.3.2 Thermoelectric Challenges

Designing the TEC has been a constant stream of challenges. Most of them stemming from the voltage regulator. The first hurdle to overcome was to create a reference voltage from the Arduino. Many different methods for doing this was theorised and tested. In the end what we ended up doing was to go the complete opposite direction of what we originally intended. What started out as being a slow analogue signal became a very fast binary signal in the end.

Overheating MOSFETs was the the largest challenge when it came to designing the TEC regulator. The heat stemming from the linear region of the MOSFET can only be overcome by finding the right balance of the following things: switching states faster, switching less frequently, reducing the current, or adding heat sinks. The solution that time allowed us to come up with did all of these things.

6.3.3 Printed Circuit Boards

When we produced Printed Circuit Boards (PCB) a lot of challenges was uncovered during the process. In difference with simulation, the footprint of the components has to be considered. Otherwise the component will not fit on the board when placing components. Routing the signals provides also challenges. Because if a signal is overlooked, it could cause the whole circuit to malfunction. Even if the signal is correctly placed, many problems can rise while tracing the wires. A normal problem is to underestimate the current that will flow through the wire, which could result in the PCB overheating. This also depends on the thickness of the copper trace on the circuit board. Components can also become damaged while being soldered to the board. It is very hard to notice if has become damaged before testing the whole circuit. A component that has been burnt or has faults could actually cause severe damage to the circuit. While testing the final PCB for the battery controller, a faulty pot meter was found. Luckily, it was not able to cause any damage before being replaced.

6.4 Software Challenges

6.4.1 Working with Legacy Code

At the very beginning of the project it was determined that we work with the previously implemented software, rather than starting from scratch. This meant thoughout the project, that we had to read and understand someone elses code. It also made the challenge even worse, that a lot of code had not been sufficiently commented, or otherwise documented. All of this meant that we ended up spending as much time attempting to figure out how the code was supposed to work, as figuring why it did not work. As a result of the lack of documentation we inherited, we also lost one piece of help which we had hoped to get. We had hoped that we would receive up to date software, as well as system architecture diagrams. However lacking this, we had no better method of understanding the code, than to read the code itself. It also meant that when designing new diagrams, they had to be based entirely of the current source code, and our understanding of it, which at times turned out to be wrong. Thanks to having spend a lot of time, simply understanding the code, and designing new diagrams from scratch, we did not get to spend nearly as much time solving problems as we would have proffered.

6.4.2 Lack of Documentation for Virtual Battle Space

The biggest software problem to solve, was without a doubt implementing communication between Argos and Virtual Battle Space (VBS). This is related to the fact the VBS is primarily targeted at the military, and military contractors, it also allows the developer, Bohemia Interactive to charge a premium to teach developers how to develop VBS plugins. Unfortunately our employer was not willing to pay that premium.

This meant that the only way we had to learn, was through the three plugin examples bundled with VBS, however even they were commented to the absolute minimum. As a result of this lack of documentation, our only way of actually figuring out how to develop the plugins was through a combination of guessing and intuition with how it made sense that this would work. The perfect example of this becoming an issue was with the ExecuteCommand function used in the VBS plugin. When originally developing the plugin we had simply assumed that VBS would run its plugins in different threads, which would require ExecuteCommand to be a threadsafe function. However based on the issues we encountered, our only theory is that VBS actually runs each plugin consecutively, which would mean that from the prespective of VBS, there would be no need for ExecuteCommand to be threadsafe.

When we started to use the ExecuteCommand function, there was a challenge to figure out how to use it. the only comment about it explained that we needed to use it for the VBS coding, but nothing about how to use it. Without any information, we had no other option than to continue guessing and testing. After some time we figured out how to get the return value, which was the char parameter of the ExecuteCommand function.

Many of the functions used in VBS programming also lack descriptions, which made it difficult to find out which function to use to get specific information about different types of objects. This challenge came from the fact that there is different function with similar names that do almost the same thing. There is even possible to get different information with the same function, with just a slightly different syntax. For example, with the use of the function called name we only got the name of the driver of a vehicle when we tried to get the vehicle name, because objects return the name when reading off directly.

When we tried to use different function to see what worked we got a lot of error messages. This was yet another challenge, because of how generic all the error messages was. Each error message ether complained about a missing symbol or just pointed to a part of the code. Most of the time when it complained about a missing symbol, the symbol was ether not necessary or already there. when the error pointed to the code, the place had often nothing to do with the error.

6.4.3 Limited VBS map of Kongsberg

While we did receive a map of Kongsberg which was compatible with VBS, it turned out that the maps northernmost point was just south of the university campus. Since our main intent was to show the connection of something moving around in VBS, and the targets moving around in Argos as a result. In order to correctly demonstrate this connection, we became limited in only being able to simulate targets south of the university. As a common pattern, this issue was not documented anywhere , meaning we also had to experiment, in order to know where the VBS map actually stopped.

7 Argos Final Product

The car mounting system is created by connecting aluminium profiles together. The system is held in place on the roof rack of a car by using eccentric locks with hooks, fitted with a shrink tube to increase the friction. To mount the aluminium rack to the roof of a car, lift the rack on top of the car, center it and customize the width of the hook to fit the width of the roof rack. Make sure it is tightened. As a side note it is easier to mount the camera front box on the roof mounting system, before you center the mounting system by pulling it to one side of the car. Fig.7.1 shows how the entire system is connected.

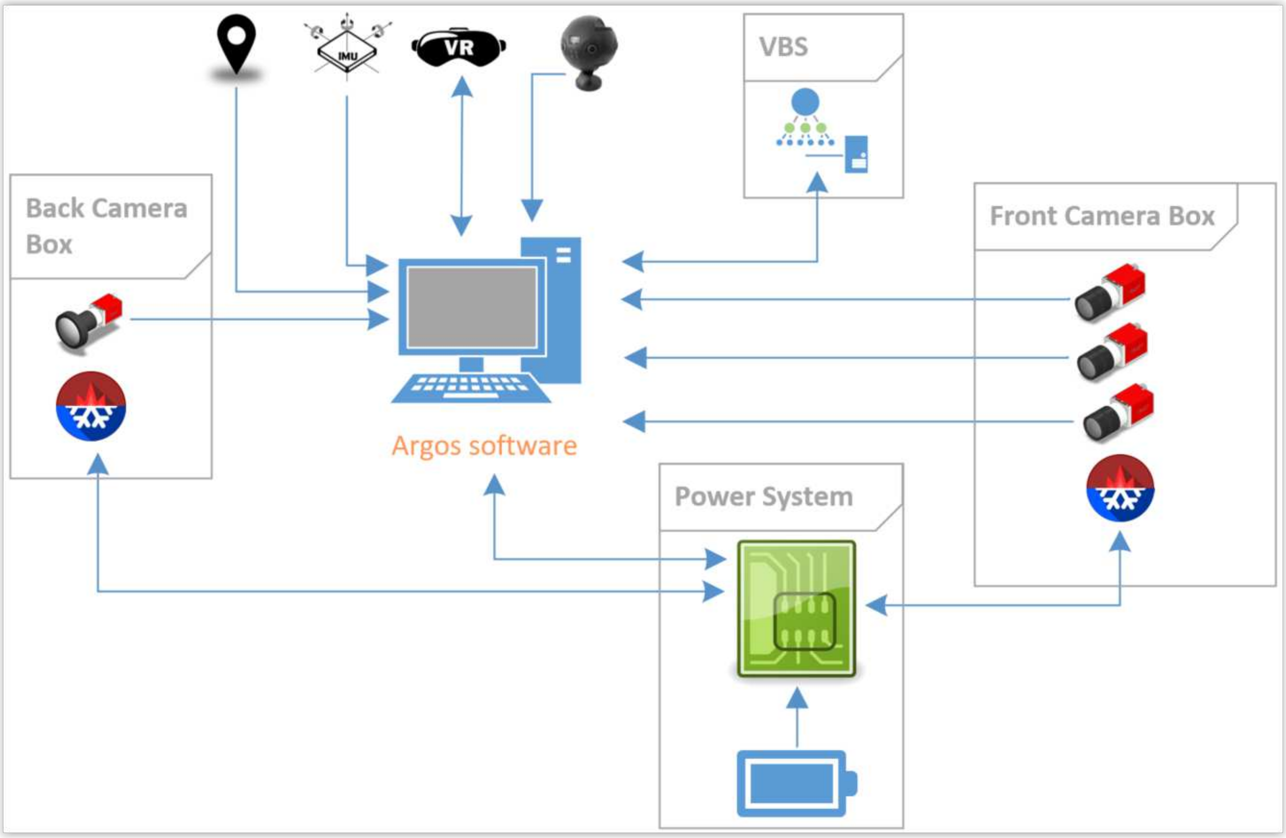


Figure 7.1: Argos 3 Complete System Architecture

The camera boxes are designed to slide into the slots of the roof mounting system. When installing the rear camera box, it is suggested to stand behind the car. While the front camera box is suggested to install from the side. After the camera boxes are installed, a locking bolt can be screwed into place to secure the camera boxes. If one should wish to change the light sensitivity on the cameras, unlocking the eccentric locks and lifting the lid, should give easy access. Remember to secure the lock with the spring pin afterwards. The boxes are equipped with cameras that provides a 360°view and temperature regulator. A micro controller with temperature and humidity sensors, reads the environment in the camera box. A thermoelectric Peltier element controls the environment, inside the box. A power cable connector is placed under each box together with the necessary Ethernet connectors.

The power system has been redesigned to give stable power to all the components and has implemented automatic battery hot swapping. The power cable for the camera boxes are connected directly to the power system providing power and signals. To connect the batteries to the power system, hook up the correct polarity to each of the dedicated pins. To activate the batteries and allow usage, the activation switch must be flipped. When the desired batteries are activated, push the power button. The battery controller will select the battery with highest charge. If both batteries are activated, the power button can also force a battery switch. The power system will send all information provided by the camera box and the battery status to the Argos software.

The data collected from camera boxes by the power system is sent to the PC and processed by Argos. The IMU and GPS will provide position and movement to Argos while using the system. The information gained will be sent to the VR goggles. This provides more insight in the whole system making sure everything functions correctly. The IMU can be mounted either to the power supply or the pc inside the car and its cable will be directly connected to the PC. Argos will also receive information from VBS.

From VBS we can acquire information, positions and simulate different scenarios. VBS communicates directly with Argos and shows where the simulated positions is in comparison with the position of Argos.

8 Conclusion

Redesigning the physical parts of Argos was essential to get a more streamlined and convenient testing of Argos software. Our job was to make the mechanical and electrical parts easier to handle and require as little maintenance as possible. Our employer wanted future interns and students to be able to focus on developing the Argos software only. The mechanical parts were made more streamlined and lightweight. The time it takes to mount, dismount and access the cameras for testing has significantly decreased. We also succeeded making the whole process more manageable if there's only one tester.

The vibration attenuation for the IMU was redesigned to work as a permanent solution regardless that we couldn't run simulations for it. We spent a lot of time designing and running simulations before starting production. We learned a lot about how long production takes and about production methods. Learning to do better 3D designs for production. The Plexiglas for the front box is currently cracked because of a late glass delivery. It is advised to address this as soon as possible. This has caused us to not being able to run a waterproof test on the front camera box. We also advise to buy more durable locking bolts to secure the camera boxes in place. In hindsight we do not regret doing production by ourself but we would've had better time if we sent the parts to production.

The power system that would provide the power for the computer was very well researched in the beginning of the project. By researching the different options, we were able to establish a good solution that worked very well for its purpose. The true sine wave provides an exceptional output for any AC driven systems. However, if the system were to go beyond the proof of concept state, either a full DC system would save space and most likely less problems. If a full DC system would be possible, a recommendation for the future is the use of NUC. At the time of the project, the systems were not fully developed, but will improve over time.

In the beginning, a preference for the use of transistors to control the batteries was a feature we really wanted, to avoid moving parts. With the time limit on the project, we had to move away from the idea and use relays. There are many heavy-duty transistors on the market and should be considered if a redesign of the battery controller is required. A fuse between the relays from battery 1 and battery 2 may be considered in the future, which would work as a last resort, if the battery controller were to malfunction.

As we were developing the battery monitoring circuit, a lot of the mistakes were made because we did not consider the limits of the components. If we were to develop a circuit in the future, the problems it can cause will be a lot less underestimated. If had the knowledge of designing circuits from scratch, we believe a lot more work would have been to research better and focusing on the exact values.

The TEC was much more effective than anticipated, which caused us to use a lot more time on the system than necessary. A lot of the focus was towards the effectiveness of the TEC but could have been used on the whole circuit. Since the TECs had such high-power ratings, creating a switching regulator for the circuit was a lot more complicated than anticipated. The complication could have been avoided, or at least made less of a problem, if we had chosen a TEC with lower power ratings. Feedback have solved many of the complications.

Through the entire project we have noticed various issues with the software. The most severe problem that we noticed was that during playback of an Insta 360 recording. If we attempted to quit the application through the menu, the application was almost guaranteed to cause a memory leak. In the cases where we did not forcefully kill the application within seconds of the memory leak starting, the computer would almost always go through a blue screen of death. It was a part of the plan for us to resolve this, however when we finally finished the Virtual Battlespace (VBS) implementation, we were unable to trigger the memory leak. We are not sure if we managed to unknowingly resolve the issue, or if there was some other factor necessary for the memory leak to begin. While we can't get the memory leak to occur, it should be of the highest priority for the upcoming summer interns to discover what may be causing the leak so that it can be resolved.

While we were able to re-implement the functionality to record the stream from the GigeVision cameras. We unfortunately did not have enough time to re-implement the ability to play back the recordings for later use. This is not a very high priority functionality, as it is the intention since Argos 2.0 to mostly use Insta 360 recordings for playback. However, the external supervisor has expressed a wish for the functionality to return.

We also became aware of how tedious it is to synchronize a recording from the Insta 360 to that of other data sources. The best solution for synchronization as of now is to manually add placeholder data or remove data from the data recordings. While manual synchronization works, it is a very tedious process as such we would recommend exploring either the possibility to record the Insta 360 stream received in the Elgato capture car. Alternatively, if the Insta 360 records with a timestamp, to read and use this for synchronization.

While working on the VBS we managed to find a solution to the requirement, but we also wanted to add an additional feature. We wanted to add an object to represent the Argos vehicle in the simulated mission, with some information from Argos. Because of time restraints, we weren't able to do add this feature. With the connection between Argos and VBS already done, it shouldn't be too difficult to add Argos to the mission. The biggest challenge will be to convert the information from a variable in C++ to a VBS variable.

Looking back at the project it has been incredibly fun and educational. We feel honoured to have been assigned the project by Kongsberg Defence & Aerospace, a world class company.

References

[1] Gangjun Yang, Guoding Chen, Suihuai Yu and Jianjie Chu, "The method of agile industrial design," 2009 IEEE 10th International Conference on Computer-Aided Industrial Design & Conceptual Design, Wenzhou, 2009, pp. 53-56. Available:
https://ieeexplore.ieee.org/document/5374892

[2] Kent Beck et al, *Manifesto for Agile Software Development*, 2001. Available:
https://agilemanifesto.org/ [Accessed March 2019]

[3] Ajan Reddy, *The Scrumban [R]Evolution: Getting the Most Out of Agile, Scrum, and Lean Kanban*, Addison-Wesley Professional, 2015

[4] M. Ikonen, E. Pirinen, F. Fagerholm, P. Kettunen and P. Abrahamsson, "On the Impact of Kanban on Software Project Work: An Empirical Case Study Investigation," 2011 16th IEEE International Conference on Engineering of Complex Computer Systems, Las Vegas, NV, 2011, pp. 305-314. Available:
https://ieeexplore.ieee.org/document/5773404

[5] N. Oza, F. Fagerholm and J. Münch, "How does Kanban impact communication and collaboration in software engineering teams?," 2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), San Francisco, CA, 2013, pp. 125-128. Available:
https://ieeexplore.ieee.org/document/6614747

[6] R. Juric, "Extreme programming and its development practices," ITI 2000. Proceedings of the 22nd International Conference on Information Technology Interfaces (Cat. No.00EX411), Pula, Croatia, 2000, pp. 97-104. Available:
https://ieeexplore.ieee.org/document/915842

[7] P. King and C. Smith, "Technical lessons Learned Turning the Agile Dials to Eleven!," Agile 2008 Conference, Toronto, ON, 2008, pp. 233-238.
https://ieeexplore.ieee.org/document/4599483

[8] C. Doshi and D. Doshi, "A Peek into an Agile Infected Culture," 2009 Agile Conference, Chicago, IL, 2009, pp. 84-89. Available:
https://ieeexplore.ieee.org/document/5261104

[9] M. Molhanec, "Agile Product Design - A modern approach to quality improvement," Proceedings of the 36th International Spring Seminar on Electronics Technology, Alba Iulia, 2013, pp. 178-182. Available:
https://ieeexplore.ieee.org/document/6648238

[10] M. Molhanec, "Extreme manufacturing — Agility to greater productivity and quality," Proceedings of the 2014 37th International Spring Seminar on Electronics Technology, Dresden, 2014, pp. 164-169. Available:
https://ieeexplore.ieee.org/document/6887585

[11] G. Yang, S. Yu, J. Chu and G. Chen, "Industrial Design Oriented Agility Evaluation Based on CTRS Model," 2010 International Conference on System Science, Engineering Design and Manufacturing Informatization, Yichang, 2010, pp. 99-102. Available:
https://ieeexplore.ieee.org/document/5640150

[12] Gangjun Yang, Suihuai Yu, Guoding Chen and Jianjie Chu, "Agile industrial design management based on Scrum," 2010 IEEE 11th International Conference on Computer-Aided Industrial Design & Conceptual Design 1, Yiwu, 2010, pp. 889-891. Available:
https://ieeexplore.ieee.org/document/5681880

[13] D. Fox, J. Sillito and F. Maurer, "Agile Methods and User-Centered Design: How These Two Methodologies are Being Successfully Integrated in Industry," Agile 2008 Conference, Toronto, ON, 2008, pp. 63-72. Available:
https://ieeexplore.ieee.org/document/4599453

[14] A. Mundra, S. Misra and C. A. Dhawale, "Practical Scrum-Scrum Team: Way to Produce Successful and Quality Software," 2013 13th International Conference on Computational Science and Its Applications, Ho Chi Minh City, 2013, pp. 119-123. Available:
https://ieeexplore.ieee.org/document/6681108

[15] A. Srivastava, S. Bhardwaj and S. Saraswat, "SCRUM model for agile methodology," 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, 2017, pp. 864-869. Available:
https://ieeexplore.ieee.org/document/8229928

[16] F. u. Rehman, B. Maqbool, M. Q. Riaz, U. Qamar and M. Abbas, "Scrum Software Maintenance Model: Efficient Software Maintenance in Agile Methodology," 2018 21st Saudi Computer Society National Computer Conference (NCC), Riyadh, 2018, pp. 1-5. Available:
https://ieeexplore.ieee.org/document/8593152

[17] R. Turner, R. Madachy, D. Ingold and J. A. Lane, "Modeling kanban processes in systems engineering," 2012 International Conference on Software and System Process (ICSSP), Zurich, 2012, pp. 23-27. Available:
https://ieeexplore.ieee.org/document/6225976

[18] M. O. Ahmad, J. Markkula and M. Oivo, "Kanban in software development: A systematic literature review," 2013 39th Euromicro Conference on Software Engineering and Advanced Applications, Santander, 2013, pp. 9-16. Available:
https://ieeexplore.ieee.org/document/6619482

[19] L. Bo and J. Xiaomin, "Research on the Construction of Industrial Design Knowledge Agile Management System," 2015 Sixth International Conference on Intelligent Systems Design and Engineering Applications (ISDEA), Guiyang, 2015, pp. 672-675. Available:
https://ieeexplore.ieee.org/document/7462708

[20] Z. Xiaosong and L. Fei, "Agile Design System Development for Switch Cabinet," 2007 2nd IEEE Conference on Industrial Electronics and Applications, Harbin, 2007, pp. 1678-1680. Available:
https://ieeexplore.ieee.org/document/4318695

[21] United States Department of Defense, *Risk Management Guide For DOD Acquisition*,6th ed., Aug. 2016. Available:
https://www.acq.osd.mil/damir/documents/DAES_2006_RISK_GUIDE.pdf

[22] United States Department of Defence, *Department of Defense Risk, Issue, and Opportunity Management Guide for Defense Acquisition Programs*, US DoD, January 2017
https://www.acq.osd.mil/se/docs/2017-rio.pdf#page=24

[23] IEEE Power & Energy Society, "Guidelines for Test Cases", testing guidelines, *IEEE Power & Energy Society*. Available:
http://sites.ieee.org/pes-tccwg/tips-for-making-a-test-case/

[24] Shakti P, "Identifying effective test scenarios & designing great test cases - Points to remember", *Trigent*, 20. January 2016. Available:
https://blog.trigent.com/points-to-remember-while-identifying-effective-t

[25] Vincent from Thule, "Re: Ticket (#202679876): Thule fastclick" Personal email (21.02.19).

[26] Rollco, "Aluminium Profile System", aluminium profile catalogue. Available:
https://www.rollco.no/storage/6A9E6C24D4B1EC765E45149EC5DA32A589B6B070FF4
8673bbd32ac74d19bae259a1c454ac09/pdf/media/
bf3c085a9c4c49a8980690a00682f1ad/Aluminium_Profiles_New%
20range%202019.pdf

[27] Ø. Vollen, Eds., *Statikk og fasthetslære*, Kristiansand, NKI Forlaget, 2006, pp.260-305.

[28] W. C. Young and R. G. Budynas, *Roark's Formulas for Stress and Strain*, 7th ed., New York, McGraw-Hill, 2002, p-817.

[29] B. Gjevik, *Innføring i Fluidmekanikk*, pp.107-119 and p.164, Mathematical Institute, University of Oslo, June 2017. Available: http://folk.uio.no/atlej/Mek4230/Mek4230_kompendium.pdf

[30] Gooch J.W. (2011) Galvanic Corrosion. In: Gooch J.W. (eds) Encyclopedic Dictionary of Polymers. Springer, New York, NY Available: https://doi.org/10.1007/978-1-4419-6247-8_5400 [last visited 05.03.19]

[31] P.A. Schweitzer, *Paint and coatings: applications and corrosion resistance*, New York, NY, CRC Press, 2005.

[32] P. Varshney and S.S. Mohapatra, "Fabrication of Durable and Renegerable Superhydrophobic Coatings on Metallic Surfaces for Potential Industrial Applications", 2017 8th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT), Cape Town, 2017, pp. 16-20. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7917427&isnumber=7917396 last visited 11.03.19

[33] Kowa Lenses, "Camera Lens", LM6JC Datasheet. Available: https://www.kowa-lenses.com/media/pdf/1c/9e/b1/LM6JC.pdf

[34] Stemmer Imaging, "Technical Manual", Mako G-223 manual. Available: https://www.stemmer-imaging.com/media/uploads/cameras/avt/92/92330-Allied-Vision-Mako-GigE-User-Manual_lQh4MXM.pdf

[35] T. Dobbert, *Matchmoving: The Invisible Art of Camera Tracking*, 2nd ed., Indianapolis, Indiana, John Wiley & Sons, 2013, p.116.

[36] J. Xu and S. Zhuang, "Measurement of Lens Focal Length With Hartmann–Shack Wavefront Sensor Based on 4F System", *Optik*, Vol.126, Issue 13, July 2015, pp.1303-1306. Available: https://www.sciencedirect.com/science/article/pii/S0030402614006767

[37] J. Kandula, G. S. Kumar and B. Bhasker, "Experimental analysis on drag coefficient reduction techniques", 2016 4th International Symposium on Environmental Friendly Energies and Applications (EFEA), pp. 1-5. Available: https://ieeexplore.ieee.org/document/7748771

[38] Zhu Hui, Yu Hao and Yang Zhigang, "Numerical analysis on effect of back/front windshied and hood angle on automotive aerodynamic drag,"

IET International Conference on Information Science and Control Engineering 2012 (ICISCE 2012), Shenzhen, 2012, pp. 1-5. Available: https://ieeexplore.ieee.org/document/6755661

[39] S. Li, J. Zhang, Y. Liu and T. Hu, "Aerodynamic Drag Reduction Design of Van Body Truck by Numerical Simulation Method", 2011 Second International Conference on Digital Manufacturing & Automation, Zhangjiajie, Hunan, 2011, pp. 168-171. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6051911&isnumber=6051823

[40] G. Yu and D. Liu, "Numerical analysis on the relationship between aerodynamic drag coefficient and 3d molding of rear view mirror," 2011 IEEE 2nd International Conference on Computing, Control and Industrial Engineering, Wuhan, 2011, pp. 117-121. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6007971&isnumber=6007941

[41] Haddar M., Romdhane L., Louati J., Ben Amara (eds), *Design and Modelling of Mechanical Systems. Lecture Notes in Mechanical Engineering*, Berlin, Heidelberg, Springer, 2013, pp.89-96.

[42] M. Sadri, D. Younesian, and E. Esmailzadeh, "Nonlinear harmonic vibration and stability analysis of a cantilever beam carrying an intermediate lumped mass", *Nonlinear Dynamics*, May 2016, Vol.84, Issue 3, pp.1667–1682.

[43] M.N. Hamdan, M.H.F. Dado, "LARGE AMPLITUDE FREE VIBRATIONS OF A UNIFORM CANTILEVER BEAM CARRYING AN INTERMEDIATE LUMPED MASS AND ROTARY INERTIA", *Journal of Sound and Vibration*, Vol.206, Issue 2, 1997, pp.151-168. Available: https://doi.org/10.1006/jsvi.1997.1081

[44] F. Hammer, S. Egger-Lampl, and S.Q. Möller, "Quality-of-user-experience: a position paper", *Quality and User Experience*, 2018. Available: https://doi.org/10.1007/s41233-018-0022-0

[45] T.K. Das, P.Ghosh, and N.C. Das, "Preparation, development, outcomes, and application versatility of carbon fiber-based polymer composites: a review", *Advanced Composites and Hybrid Materials*, 2019, pp.1-20. Available: https://doi.org/10.1007/s42114-018-0072-z

[46] Shekar, K.C., Priya, M.S., Subramanian, P.K. et al., "Processing, structure and flexural strength of CNT and carbon fibre reinforced, epoxy-matrix hybrid composite", *Bulletin of Material Science*, Vol. 37, Issue 3,

May 2014, pp.597–602. Available:
https://doi.org/10.1007/s12034-014-0703-0

[47] E. Håkansson, J. Hoffman, P. Predecki and M. Kumosa "The role of corrosion product deposition in galvanic corrosion of aluminum/carbonsystems", *Corrosion Science*, vol. 114, Jan., pp. 10-16, 2006

[48] M. Gavelec, J. Ramik, K. Zimmermann, "Decision Making and Optimization", Switzerland, Springer International Publishing, 2015

[49] D.D. Frey, P.M. Herder, Y. Wijnia, et al., "The Pugh Controlled Convergence method: model-based evaluation and implications for design theory", *Research in Engineering Design*, Vol.20, Issue 1, March 2009, pp 41–58. Available:
https://doi.org/10.1007/s00163-008-0056-z

[50] J. Wu, Y.W. Mai, and A.F. Yee, "Fracture toughness and fracture mechanisms of polybutylene-terephthalate/polycarbonate/ impact-modifier blends", *Journal of Materials Science*, Vol.29, Issue 17, Jan. 1994, pp.4510–4522. Available:
https://doi.org/10.1007/BF00376274

[51] G. Allen,, D.C.W. Morley, and T.J. Williams, "The impact strength of polycarbonate", *Journal of Materials Science*, Vol.8, Issue 10, Oct. 1973, pp.1449–1452. Available:
https://doi.org/10.1007/BF00551669

[52] S.D. Maurya, S.K. Kurmvanshi, S. Mohanty, et al., "Synthesis and characterization of crosslinked transparent poly(ester-urethane-acrylate) containing methyl methacrylate", *Macromolecular Research*, Vol.25, Issue 9, Sept. 2017, pp.871–881. Available:
https://doi.org/10.1007/s13233-017-5105-x

[53] SABIC Innovative Plastics, "PRODUCT DATASHEET LEXAN™ 9030 and LEXAN™ 9030 TG SHEET", plastic sheet datasheet, 2018. Available:
http://sfs.sabic.eu/wp-content/uploads/resource_pdf/1525361061-35841449-LEXAN_9030_AND_LEXAN_9030_TG_DATASHEET_2018.pdf last visited 28.02.19

[54] Evonik Industries AG, "Technical information PLEXIGLAS®", plastic sheet datasheet, 2013. Available:
https://www.plexiglas.net/sites/lists/pm/documentsap/211-1-plexiglas-gs-xt-en.pdf [last visited 28.02.19]

[55] N.P. Bansal and R.H. Doremus, *Handbook of Glass Properties*, London, Academic Press INC., 1986, p.34. Available:

https://books.google.no/books?hl=en&lr=&id=yacgBQAAQBAJ&oi=fnd&pg=PP1&dq=silica+glass+properties&ots=D9Id3E0Hni&sig=jDz1KjOWkgxjjaHMewcHKojGyeg&redir_esc=y#v=onepage&q=density&f=false [last visited 28.02.19]

[56] SABIC Innovative Plastics, "PRODUCT DATASHEET LEXAN™ MARGARD™ MRX Sheet", plastic sheet datasheet. Available: https://astrup.no/content/download/4618/16126/version/1/file/PC+Lexan+MRX.pdf [last visited 28.02.19]

[57] X. Dong, Y. Zhang, J. Liu, and G. Hu, "A fisheye image barrel distortion correction method of the straight slope constraint", 2015 8th International Congress on Image and Signal Processing (CISP), 16 Oct. 2015. Available:
https://ieeexplore.ieee.org/document/7407870

[58] P.G. Awade, R. Bodhula, and N. Chopade , "Implementation of barrel distortion correction on DSP in real time", 2016 International Conference on Computing Communication Control and automation (ICCUBEA), 13 Aug. 2016. Available:
https://ieeexplore.ieee.org/document/7860047

[59] L. Shuhua, Z. Jun, "GPU-Based Barrel Distortion Correction for Acceleration", 2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing, 15 Nov. 2013. Available:
https://ieeexplore.ieee.org/document/6832003

[60] Fujifilm, "Fujinon CCTV LENS: FE185C086HA-1" Fish eye lens datatsheet Available:
https://www.fujifilm.com/products/optical_devices/pdf/cctv/fa/fisheye/fe185c086ha-1.pdf

[61] S. Kalpakjianand and S. R. Schmid, "MANUfACTURING ENGINEERING AND TECHNOLOGY", 7th ed., Singapore, Pearson Education South Asia Pte Ltd, 2014,p-97.

[62] L. Calabrese, E. Proverbio, G. Di Bella et al. "Assessment of ageing effect on the mechanical behaviour of steel/aluminium self-piercing riveted joint", in International Journal of Mechanical and Materials Engineering, vol. 10, no. 9, 2015 Available:
https://doi.org/10.1186/s40712-015-0034-y

[63] Evonik Industries, "Technical Information PLEXIGLAS® XT, UV transmitting" acrylic polymer datasheet, 2015. Available:
https://www.plexon.no/Portals/0/adam/Content/Cml-v9AoCkyaKiW-CnFTwQ/Link/232-6-PLEXIGLAS-XT-UV-transmitting-en.pdf last visited 16.05.19

[64] H. D. Young and R. A. Freedman, "University Physics", 12th ed. San Francisco, CA, USA, Pearson Education, 2008, p-456

[65] K. E. Sandvold et. al., "Gyldendals formelsamling i matematikk", 9th ed., Norway, 2008, p-6

[66] Fabio Riccardi, *Design and analysis of elastomer vibration dampener for IMU*, 22. July 2015
https://www.researchgate.net/publication/280245288_DESIGN_AND_ANALYSIS_OF_ELASTOMERIC_VIBRATIONS_DAMPERS_FOR_IMU_ON-BOARD_RUAV

[67] F. N. Murrieta-Rico et al., "Resolution improvement of accelerometers measurement for drones in agricultural applications," IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, 2016, pp. 1037-1042. https://ieeexplore.ieee.org/document/7793466

[68] D. Shmilovitz, "On the definition of total harmonic distortion and its effect on measurement interpretation", *IEEE Transactions on Power Delivery*, Volume: 20, Issue: 1, Jan. 2005. Available:
https://ieeexplore.ieee.org/document/1375138

[69] T.M. Blooming and D.J. Carnovale, "Application of IEEE STD 519-1992 Harmonic Limits", Conference Record of 2006 Annual Pulp and Paper Industry Technical Conference, August 2006. Available:
https://ieeexplore.ieee.org/document/1673767

[70] R. Nowakowski and B. King, "Challenges of designing high-frequency, high-input-voltage DC/DC converter", Texas Instruments Incorporated, 2011. Available:
http://www.ti.com/lit/an/slyt415/slyt415.pdf

[71] B. M Hasaneen and Adel A. Elbaset Mohammed,
"Design and Simulation of DC/DC Boost Converter", 2008 12th International Middle-East Power System Conference, 15 July 2008. Available:
https://ieeexplore.ieee.org/document/4562340?arnumber=4562340

[72] Subhash Chander, Pramod Agarwal, and Indra Gupta, "Design, Modeling and Simulation of DC-DC Converter", 2010 Conference Proceedings IPEC, 20 January 2011. Available:
https://ieeexplore.ieee.org/document/5697039

[73] S.Vlahinic, D. Brnobic, and N. Stojkovic, "Indices for Harmonic Distortion Monitoring of Power Distribution Systems", *IEEE Transactions on Instrumentation and Measurement*, Volume: 58, Issue: 5, May 2009. Available:
https://ieeexplore.ieee.org/document/4745761

[74] S.B. Kjaer, J.K. Pedersen, and F. Blaabjerg, "Power Inverter Topologies for Photovoltaic Modules – A Review", Conference Record of the 2002 IEEE Industry Applications Conference. 37th IAS Annual Meeting, 10 December 2002. Available:
https://ieeexplore.ieee.org/document/1042648

[75] J. Dixon and L. Moran, "High-Level Multistep Inverter Optimization Using a Minimum Number of Power Transistors", *IEEE Transactions on Power Electronics*, Volume: 21, Issue: 2, March 2006. Available:
https://ieeexplore.ieee.org/document/1603664

[76] T.C. Hsia and D.A. Landgrebe, "On a Method for Estimating Power Spectra", *IEEE Transactions on Instrumentation and Measurement*, Volume: 16, Issue: 3, Sept. 1967. Available:
https://ieeexplore.ieee.org/document/4313631

[77] C.S Moo, K.S. Ng and Y.C. Hseih, "Parallel Operation of Battery Power Modules", *IEEE Transactions on energy conversion*, Volume: 23, Issue: 2, June. 2008. Available:
https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4458231

[78] M.D. Apperley and M.M. Alahmari, "Tracking battery state-of-charge in a continuous use off-grid electricity system", Department of Computer Science, The University of Waikato, October 2013
https://www.researchgate.net/publication/260944095_Tracking_battery_state-of-charge_in_a_continuous_use_off-grid_electricity_system_Working_Paper_042013_Department_of_Computer_Science_The_University_of_Waikato

[79] Young & Freedman, *University Physics With Modern Physics*, 14th ed. p.818, Pearson Education Inc, 2016. ISBN 13: 978-0-321-97361-0

[80] W.K. Chan, *Fundamentals of Circuits and Filters*, 3rd ed., CRC Press, p.16.3. ISBN 10: 1-42-005887-8

[81] G.F Franklin, J.D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 7th ed., Pearson Education Limited, Essex, England, 2015, p.66. ISBN 10: 1-29-206890-6

[82] R. Xiong, J. Cao, Q. Yu, H. He, and F. Sun, "Critical Review on the Battery State of Charge Estimation Methods for Electric Vehicles", *IEEE Access*, Vol.6, December 2017, pp.1832-1843.
Available: https://ieeexplore.ieee.org/document/8168251

[83] B.Homan, G.J.M. Smit, R.P. van Leeuwen, M.V. ten Kortenaar, and B.V. Ten, "A comprehensive model for battery State of Charge prediction", 2017 IEEE Manchester PowerTech, 22 June 2017. Available:
https://ieeexplore.ieee.org/document/7980943

[84] S.Pang, J.Farrell, and Jie Du, M. Barth, "Battery state-of-charge estimation", Proceedings of the 2001 American Control Conference, 25-27 June 2001. Available:
https://ieeexplore.ieee.org/document/945964/

[85] Y. Hu and S. Yurkovich, "Battery state of charge estimation in automotive applications using LPV techniques", Proceedings of the 2010 American Control Conference, 29 July 2010. Available:
https://ieeexplore.ieee.org/document/5530734/

[86] A. Nugroho, E. Rijant, F.D Wijaya, and P. Nugroho, "Battery State of Charge Estimation by Using a Combination of Coulomb Counting and Dynamic Model With Adjusted Gain", International Conference on Sustainable Energy Engineering and Application (ICSEEA), 14 January 2016, Available:
https://ieeexplore.ieee.org/document/7380745

[87] Linear Technology Corporation, "Ideal Diode Controller with Reverse Input Protection ", LTC4359, datasheet, 2012.

[88] International Rectifier, "HEXFET Power MOSFET", IRBF7430PbF, datasheet, 2012.

[89] Intersil, "Dual/Quad, Low Power CMOS Operational Amplifiers", ICL7621, datasheet, 1999.

[90] Texas Instruments, "Precision Operational Amplifier", OPA177, datasheet, 2000.

[91] Tyco Electronics Corporation, "High Current Relay 150", 1393315-2, datasheet, 2014.

[92] C. Kitchin, "Demystifying single-supply op-amp", Analog Devices Inc., March 21, 2002., Available:
https://www.ieee.li/pdf/essay/single_supply_op_amp_design.pdf

[93] C.C.P. Neftali, C.C.G. Marx, and S.T.J. Alfonso, "Fast Response Temperature Model Based on Thermal Data for Joule Heating Process Using Exposed Joint Thermocouple Method", 2015 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), 6 Nov. 2015. Available:
https://ieeexplore.ieee.org/document/7395114

[94] Y.C. Lee, H.L. Chen, C.H.Chen, and H.Y. Lin, "Infrared heating and roller-based contact printing technology for the fabrication of micro/nanostructures", 2009 4th IEEE International Conference on Nano/Micro Engineered and Molecular Systems, 8 Jan. 2009. Available:
https://ieeexplore.ieee.org/document/5068694

[95] C. Phondon and P. Intani, "High power induction heating", 2014 17th International Conference on Electrical Machines and Systems (ICEMS), 25 Oct. 2014. Available:
https://ieeexplore.ieee.org/document/7014079

[96] O. Lucia, P. Maussion, E.J. Dede, and J.M. Burdío, "Induction Heating Technology and Its Applications: Past Developments, Current Technology,and Future Challenges", *IEEE Transactions on Industrial Electronics*, Vol.61, No.5, May 2014. Available:
https://ieeexplore.ieee.org/document/6595059

[97] W.C. White, "Some experiments with peltier effect", *Electrical Engineering*, Vol.70 , Issue: 7 , July 1951. Available:
https://ieeexplore.ieee.org/document/6436699

[98] A. Jose, A. D'souza, S. Dandekar, J. Karamchandani, and P. Kulkarni, "Air Conditioner using Peltier Module", 2015 International Conference on Technologies for Sustainable Development (ICTSD), 6 Feb. 2015. Available:
https://ieeexplore.ieee.org/document/7095879

[99] G. Engelmann, M. Laumen, K. Oberdieck, and R.W. De Doncke, "Peltier module based temperature control system for power semiconductor characterization", 2016 IEEE International Power Electronics and Motion Control Conference (PEMC), 28 Sept. 2016. Available:
https://ieeexplore.ieee.org/document/7752123

[100] M. G. J. Fry, "A Switch Mode Power Supply" INTELEC '82 - International Telecommunications Energy Conference, Washington, DC, USA, 1982, pp. 277-284. Available:
https://ieeexplore.ieee.org/document/4793743

[101] Maxim Integrated Products, Inc., "Programmable Resolution 1-Wire Digital Thermometer", DS18B20, datasheet, 2018.

[102] M. Burton, T. Newsome, G. Barros, and R.Tillaart, "DallasTemperature", Arduino library, Jan. 2019. Available:
https://github.com/milesburton/Arduino-Temperature-Control-Library

[103] P. Stoffregen, "OneWire", Arduino library, Jan. 2019. Available:
https://github.com/PaulStoffregen/OneWire

[104] Adafruit, "Digital relative humidity & temperature sensor", AM2303/DHT22, datasheet.

[105] U. Kang and K. D. Wise, "A high-speed capacitive humidity sensor with on-chip thermal reset," in IEEE Transactions on Electron Devices, vol. 47, no. 4, pp. 702-710, April 2000. Available:
https://ieeexplore.ieee.org/document/830983/

[106] L. Litwin, "FIR and IIR digital filters," in IEEE Potentials, vol. 19, no. 4, pp. 28-31, Oct.-Nov. 2000. Available:
https://ieeexplore.ieee.org/document/877863

[107] Robert Purchese, "Bohemia's war: the story of the company behind Arma and DayZ", *Eurogamer*, 30.04.2014. [Online], Available:
https://www.eurogamer.net/articles/2014-04-29-bohemias-war-the-story-of-t

[108] "std::thread", *cplusplus.com*, [Online], Available:
http://www.cplusplus.com/reference/thread/thread/

[109] "VBS Scripting Reference", *BISimulation*, 29.04.2019. [Online], Available:
https://sqf.bisimulations.com/display/SQF/VBS+Scripting+Reference?fbclid=IwAR3SzJwUJ0GFvB-diXmc9_OpYylCBJIX0sEnDNhNV-wb_PTcWh2w4Plb-Xs

[110] Jhon Kennedy, Michael Satran, "Windows Sockets 2", *Microsoft*, 31.05.2018. [Online], Available:
https://docs.microsoft.com/en-us/windows/desktop/winsock/windows-sockets-start-page-2

[111] "prosjekt", *projectargos.net*, [Online], Available:
http://www.projectargos.net/prosjekt/

[112] "UE4Duino", *github.com*, [Online], Available:
https://github.com/RVillani/UE4Duino/

[113] "Yantranamov", *fonts.google.com*, [Online], Available:
https://fonts.google.com/specimen/Yantramanav

Appendices

A List of Requirements

Originally we received a baseline requirement specification from the customer, by breaking those down into smaller more specific requirements, and multiple meetings we were able to achieve a better understanding of the customers true requirements for the changes we were expected to implement. All requirements are divided into 3 different categories, Functional, Non-functional, and Constraints. This categorization of the requirements was initially made by the customer, partially based on the two previous bachelor projects. Each requirement has a unique ID, which also includes the type in the form of F (Functional), NF (Non-functional), and C (Constraint) as illustrated in Table A.1.

Table A.1: Requirement Classification Explanation

Classification	Explanation
F	Functional Requirement
NF	Non-Functional Requirement
C	System of Project Constraint

Predicting that we would wish to add, or modify some requirements though the project, it was also necessary for each requirement to have a status, we specify this though the statuses Accepted, Pending Acceptance, Pending Change, Replaced. The different statuses are explained and illustrated in Table A.2.

Table A.2: Requirement State Explanation

State	Explanation
Accepted	The requirement has been accepted by the customer
Pending Acceptance	The new requirement is awaiting acceptance from customer
Pending Change	The requirement has been proposed modified and awaiting acceptance
Replaced	The requirement has been replaced by other requirements
Removed	The requirement was turned down, or removed for other reasons

Since the customer regarded certain requirements to be less important than other requirements, it was also necessary for all requirements to have a priority, we decided to only keep three levels of priority, A, B, and C, as shown in Table A.3.

Table A.3: Priority Explanation

Priority	Explanation
A	The requirement must be achieved
B	The requirement should be achieved
C	The requirement should be achieved, but is not a priority

Knowing that some requirements would contain use cases to the software, we were also able to predict that we wanted requirements to be traceable to use cases if the requirement is connected to one. Knowing this this ahead of time we were able to design a template for requirements as shown in Table A.4.

Table A.4: Requirement table explanation

[Class.X]	[Date Added]	[Requirement Title]	[State]	[Priority]
	[Origin]	[Use Case ID] (Functional Software Only)	[Test ID]	
	Description	[Requirement Description]		
	Comments	[Requirement Comments]		

A.1 Functional Requirements

Table A.5: Argos 3 functional requirements

F.1.1	17.01.19	BMS simulation	Accepted	A
	KDA			
	Description	The system shall have a BMS simulation that simulates and sends data to Argos		
	Comments			
F.1.1.1	04.02.19	BMS to Argos communication	Accepted	A
	Group			
	Description	The BMS shall be able to send data to Argos		
	Comments			
F.1.1.2	04.02.19	BMS simulation	Accepted	A
	Group	UC6		
	Description	The BMS shall simulate real life scenarios for Argos		
	Comments			

F.1.1.2.1	04.02.19	BMS simulation scenario	Accepted	A
	Group	UC6		
	Description	The BMS must have a predefined scenario		
	Comments			
F.1.1.2.1.1	04.02.19	BMS NPCs	Accepted	A
	Group	UC14		
	Description	The simulation must have NPCs, simulating allies, enemies and civilians		
	Comments			
F.1.1.2.1.2	04.02.19	BMS map	Accepted	A
	Group	UC6		
	Description	The simulation scenario must use a map of Kongsberg		
	Comments			
F.1.2	17.01.19	Display BMS data	Accepted	A
	KDA	UC14		
	Description	The system shall be able to display BMS information overlay from an external data link		
	Comments			
F.1.2.1	04.02.19	Display BMS data	Accepted	A
	Group	UC4		
	Description	The Argos Application shall have a OSM map of Kongsberg		
	Comments	This is already fulfilled, also somewhat wrong, consider removing		
F.1.2.2	04.02.19	Receive communication	Accepted	A
	Group			
	Description	The Argos Application shall be able to receive data from BMS		
	Comments			
F.1.3	17.01.19	Argos visible in BMS	Accepted	A
	KDA	UC15		
	Description	The Argos vehicle/system shall be accessible/visible in the BMS		
	Comments			
F.1.3.1	04.02.19	Argos to BMS communication	Accepted	A
	Group			
	Description	The argos system shall be able to send data to the BMS		
	Comments			

F.1.3.2	04.02.19	BMS recieve data from Argos	Accepted	A
	Group	UC15		
	Description	The BMS shall be able to spawn and update 3D models based on data from Argos		
	Comments			
F.1.4	04.02.19	Argos, BMS data communication	Accepted	A
	Group			
	Description	The Argos application and BMS shall be capable of two way communication		
	Comments			
F.1.4.1	04.02.19	Argos, BMS data protocol	Accepted	A
	Group			
	Description	The Argos application and BMS shall have to common protocol for communication		
	Comments	Reconsider this, as it's easier to parse data in Argos than BMS		
F.1.4.2	04.02.19	Argos, BMS data link	Accepted	A
	Group			
	Description	The Argos application and BMS shall be connected with a data link		
	Comments			
F.2.1	17.01.19	Power source system	Accepted	A
	KDA			
	Description	The power source system for Argos in a car shall be redesigned and supply a stable power and voltage to the Argos system		
	Comments			
F.2.1.1	04.02.19	Power source interference	Accepted	A
	Group			
	Description	The power source system shall generate minimal interference		
	Comments			
F.2.1.2	04.02.19	Power source run-time	Accepted	A
	Group			
	Description	The power source system shall allow for a long run-time (1 hour)		
	Comments			

F.2.1.2.1	04.02.19	Power source power consumption	Accepted	A
	Group			
	Description	The power source system shall inflict lowest possible power consumption		
	Comments			
F.2.1.2.1.1	04.02.19	Power source efficiency	Accepted	A
	Group			
	Description	The power source system shall maintain a high efficiency (xx% or more)		
	Comments			
F.2.1.2.2	04.02.19	Power source energy capacity	Accepted	A
	Group			
	Description	The power source system shall have a high energy storage capacity		
	Comments			
F.2.1.2.3	04.02.19	Power source hot-plug	Accepted	B
	Group			
	Description	The power source system shall allow for adding more energy to the system without interruption of the system		
	Comments			
F.2.1.3	04.02.19	Power source current tolerance	Accepted	A
	Group			
	Description	The power source system must tolerate large changes in current		
	Comments			
F.2.2	17.01.19	Power source mounting	Accepted	A
	KDA			
	Description	The power source system shall be easy to mount and dismount in a vehicle		
	Comments			
F.2.2.1	04.02.19	Power source external cables	Accepted	A
	Group			
	Description	The power source system shall have as few external cables as possible		
	Comments			

F.2.2.2	04.02.19	Power source vehicle modification	Accepted	A
	Group			
	Description	The power source system shall not require any modifications to the vehicle it is to be used in		
	Comments			
F.2.2.3	04.02.19	Power source fastening	Accepted	A
	Group			
	Description	The power source system shall be securely fastened to deal with emergency stops		
	Comments			
F.2.3	17.01.19	Power source data system	Accepted	A
	KDA			
	Description	The power source system shall supply status information that can be accessed or viewed in the VR goggles		
	Comments			
F.2.3.1	04.02.19	PC from power source communication	Accepted	A
	Group			
	Description	The PC must be able to receive data from the power source system		
	Comments			
F.2.3.2	04.02.19	Supply status information	Accepted	A
	Group			
	Description	The power source system must be able to transmit status information to the PC		
	Comments			
F.2.3.2.1	04.02.19	Acquire humidity	Accepted	A
	Group			
	Description	The power source system shall acquire the level of humidity surrounding the cameras		
	Comments			
F.2.3.2.2	04.02.19	Acquire temperature	Accepted	A
	Group			
	Description	The power source system shall acquire the temperature surrounding the cameras		
	Comments			

F.2.3.2.3	04.02.19	Acquire time left	Accepted	A
	Group			
	Description	The power source system shall acquire the remaining run-time		
	Comments			
F.2.3.2.4	04.02.19	Acquire charge left	Accepted	A
	Group			
	Description	The power source system shall acquire the remaining charge		
	Comments			
F.3.1	17.01.19	Camera accessibility	Accepted	A
	KDA			
	Description	The camera system shall be redesigned so that it is possible to access the cameras in a short time period (3 minutes)		
	Comments			
F.3.1.1	01.02.19	Camera access multiple use	Accepted	A
	Group			
	Description	The camera cover mounting shall be multiple use		
	Comments	E.g. no silicone that needs to be replaced each time		
F.3.1.2	01.02.19	Camera cover tool-less	Accepted	A
	Group			
	Description	The camera system shall require no other equipment to gain access to the cameras		
	Comments	E.g. don't need screwdrivers to get access		
F.3.1.3	01.02.19	Simpler connections	Accepted	A
	Group			
	Description	The camera system shall require no manual disconnection of wires to gain access to the cameras		
	Comments			
F.3.2	01.02.19	Camera environment protection	Accepted	A
	KDA			
	Description	The cameras shall not be affected by the environment		
	Comments			

F.3.2.1	17.01.19	Camera rain protection	Accepted	A
	KDA			
	Description	The camera box and its contents shall not take damage if exposed to rain		
	Comments	It does not need to be water-proof		
F.3.2.2	17.01.19	Clear camera view	Accepted	A
	KDA			
	Description	The cameras shall always have clear view		
	Comments			
F.3.2.3	17.01.19	Camera environment regulation	Accepted	A
	KDA			
	Description	Regulate the environment of the cameras so they can operate within -10°C and 30°C external air temperature		
	Comments			
F.3.2.3.1	01.02.19	Camera environment regulator	Accepted	A
	Group			
	Description	The camera system shall have a regulator		
	Comments			
F.3.2.3.2	01.02.19	Camera environment thermal transfer	Accepted	A
	Group			
	Description	The camera system shall be able to transfer thermal energy		
	Comments			
F.3.2.3.3	01.02.19	Camera environment humidity measurement	Accepted	A
	Group			
	Description	The camera system shall be able to measure humidity		
	Comments			
F.3.3	17.01.19	Camera system mounting	Accepted	A
	Group			
	Description	The camera system shall take no longer than 5 minutes to mount/dismount on a car		
	Comments			

F.4.1	22.01.19	Vehicle IMU movement compensation	Accepted	A
	KDA			
	Description	The system shall use an IMU (Inertial Measurement Unit) align the system when used in a vehicle		
	Comments			
F.4.2	22.01.19	IMU stabilization	Accepted	A
	KDA			
	Description	The IMU shall be stabilized in order to prevent drifting when used in a vehicle		
	Comments			
F.4.2.1	22.01.19	IMU movement filter	Accepted	A
	Group			
	Description	The IMU must eliminate specific movements		
	Comments			
F.4.2.1.1	04.02.19	IMU high frequency	Accepted	A
	Group			
	Description	The IMU shall eliminate movements over 50Hz		
	Comments			
F.4.2.2	04.02.19	IMU Mounting	Accepted	A
	Group			
	Description	The IMU must must be securely mounted		
	Comments			
F.5	17.01.19	Insta 360 Mounting	Accepted	B
	KDA			
	Description	The Insta360 camera can be mounted on a vehicle		
	Comments			

A.2 Non-Functional Requirements

Table A.6: Argos 3 non functional requirements

NF.1	17.01.19	English Language	Accepted	A
	KDA			
	Description	Information and messages must appear in English		
	Comments			
NF.2	17.01.19	Accepted VR goggles	Accepted	A
	KDA			
	Description	The VR goggles supported shall be the Oculus Rift CV1 and HTC Vive		
	Comments			
NF.3	17.01.19	Accepted VR Refresh Rate	Accepted	A
	KDA			
	Description	The VR goggles must have refresh rate higher than 75 Hz		
	Comments			
NF.4	17.01.19	Accepted VR Resoulution	Accepted	A
	KDA			
	Description	The VR goggles must have at least 960x1080 resolution per eye		
	Comments			
NF.5	17.01.19	Accepted VR Field of View	Accepted	A
	KDA			
	Description	The VR googles must have at least 100 degrees field of view		
	Comments			
NF.6	17.01.19	Maximum Accepted Latency	Accepted	A
	KDA			
	Description	Glass to glass latency must be less than 75 ms		
	Comments	This only applies to the Mako cameras		
NF.7	17.01.19	Minimum Network Throughput	Accepted	A
	KDA			
	Description	Network must have a minimum throughput of 512 MB/s		
	Comments			

NF.8	17.01.19	Minimum Camera Framerate	Accepted	A
	KDA			
	Description	Cameras must be able to capture at least 50 FPS		
	Comments	49.5 for Mako cameras, because of resolution		
NF.9	17.01.19	Runnable on a laptop	Accepted	B
	KDA			
	Description	The system shall be able to run on a powerful laptop. The laptop shall be able to run recordings and live video from one camera		
	Comments			
NF.10	17.01.19	Argos Website	Accepted	A
	KDA			
	Description	The Argos web site (www.projectargos.net) shall be updated		
	Comments			
NF.11	17.01.19	Document Argos PC Setup Procedure	Accepted	A
	KDA			
	Description	The installation and setup of Windows 10 for the Argos computers shall be documented		
	Comments	Partially done by Bachelor and Summer teams of 2018		
NF.12	17.01.19	Support Doxygen	Accepted	A
	KDA			
	Description	The source code shall support doxygen documentation		
	Comments			
NF.13	17.01.19	System Startup Time	Accepted	B
	KDA			
	Description	System start-up should take maximum 20 seconds		
	Comments			
NF.14	17.01.19	Clicks to Show Live Video	Accepted	B
	KDA			
	Description	Showing live video in VR goggles should take at most 5 clicks		
	Comments			

NF.15	17.01.19	Learning Key Functionality	Accepted	B
	KDA			
	Description	It should be possible to learn key functionality in 10 minutes		
	Comments			
NF.16	17.01.19	Mastering All Functionality	Accepted	B
	KDA			
	Description	It should be possible to master all functionality in less than 1 day		
	Comments			
NF.17	17.01.19	Support for 360 Camera	Accepted	A
	KDA			
	Description	It should be possible to use a 360 degree camera		
	Comments			

A.3 Constraints

Table A.7: Argos 3 constraints

C.1	17.01.19	System Availability	Accepted	A
	KDA			
	Description	The system must not be available to the public		
	Comments			
C.2	17.01.19	Accepted Programming Language	Accepted	A
	KDA			
	Description	Programming language shall be C++		
	Comments			
C.3	17.01.19	Accepted Operating System	Accepted	A
	KDA			
	Description	OS shall be Microsoft Windows 10 64-bit version		
	Comments			
C.3	17.01.19	Accepted Software Version Control	Accepted	A
	KDA			
	Description	Software Version Control system shall be used. It can be either Mercurial (Hg) or GIT		
	Comments			
C.4	17.01.19	Accepted Integrated Development Environment	Accepted	B
	KDA			
	Description	IDE shall be Microsoft Visual Studio		
	Comments			

B Test Report

B.1 Test Report Explanation

For every test that will be preformed, it has to be rated either accepted, inconclusive or failed as seen in Table B.1. What rating the test will receive depends if it has met the criteria necessary to approve the test as accepted, or if there was any unforeseen factors that made the test inconclusive. If it did not pass the criteria and there were no unforeseen factors, the test will be rated as failed.

State	Explanation
Accepted	The test was successful and has passed the criteria
Inconclusive	Results were unclear and test needs to be repeated
Failed	The test did not pass the criteria

Table B.1: Test result explanation

Every test will have a priority rating as seen in Table B.2. The priority depends only on what rating the tested requirement has.

Priority	Explanation
A	The tested requirement has priority level A
B	The tested requirement has priority level B
C	The tested requirement has priority level C

Table B.2: Priority Explanation

In Table B.3 and Table B.4 one can see a short explanation of the tables that will be used for each test.

Table B.3: Testing explanation

Test ID	[Test ID]	**Priority**	[Priority level]
Requirements tested	[Requirements that are to be tested]		
Test description	[Description of how the test will be preformed]		
Resources	[Tools and resources necessary to preform the test]		
Test steps	[Necessary steps to preform the test]		
Acceptance Criteria	[Criteria for the test to pass]		

Table B.4: Test report explanation

Test ID	[Test ID]	**Date**	[Date]
Test Number	[Test number]	**Result**	[Conclusion]
Testers Name	[Testers names]		
Comment	[Comments on the test]		

B.2 IMU Testing

Table B.5: Testing IMU

Test ID	T.1	**Priority**	A
Requirements tested	F4.1, F4.2, F4.2.1		
Test description	Driving with the IMU mounted in a car		
Resources	IMU, Oculus rift, predator laptop, car, two team members		
Test steps	1. Mount the IMU inside a car 2. Connect pc and VR goggles to IMU 3. Drive in a straight line 4. Take a slow turn 5. Make a sharp turn 6. High acceleration 7. High de-acceleration 8. Driving over speed bumps		
Acceptance Criteria	• The map should always stay in front of the Argos operator • The Argos operator should not notice any drift		

Table B.6: Test report IMU attenuation

Test ID	T.1.1	**Date**	28.02.19
Test Number	1	**Result**	Inconclusive
Testers Name	Marco Tande, Vegar Skogen & Jan Olav Møen		
Comment	Noticed some drifting in the VR, unsure if it's because of being held in place in a car seat or drift, have to do more tests		

Table B.7: Test report IMU attenuation

Test ID	T.1.2	**Date**	15.03.19
Test Number	2	**Result**	Inconclusive
Testers Name	Marco Tande & Vegar Skogen		
Comment	Mounted IMU in a better position with less variables, wrong cable connection point unsure if this caused drifting		

Table B.8: Test report IMU attenuation

Test ID	T.1.3	**Date**	15.03.19
Test Number	3	**Result**	Failed
Testers Name	Marco Tande & Vegar Skogen		
Comment	The test showed that our design failed, we need to design a new dampening		

Table B.9: Test report IMU attenuation

Test ID	T.1.4	**Date**	01.04.19
Test Number	4	**Result**	Failed
Testers Name	Marco Tande & Vegar Skogen		
Comment	Tested with our new design for vibration dampening, test failed as we had a slight drifting over time while driving		

Table B.10: Test report IMU attenuation

Test ID	T.1.5	**Date**	02.04.19
Test Number	5	**Result**	Inconclusive
Testers Name	Marco Tande & Vegar Skogen		
Comment	Ran a test with the same test setup as the previous, but we turned on the magnetometer on the IMU. Had some slight drifting in the start and then it was considerably improved results. We assume the drifting at the start was because the IMU was initializing, need to run another test		

Table B.11: Test report IMU attenuation

Test ID	T.1.6	**Date**	04.04.19
Test Number	6	**Result**	Accepted
Testers Name	Marco Tande & Vegar Skogen		
Comment	Testing with magnetometer and with IMU stack instead of a single IMU, this test was accepted because we had no drifting at all and the dampening was efficient enough. We only have to do some optimization to the design as it isn't designed for an IMU stack but for a single IMU		

Table B.12: Test report IMU attenuation

Test ID	T.1.7	**Date**	09.04.19
Test Number	7	**Result**	Accepted
Testers Name	Marco Tande & Vegar Skogen		
Comment	Testing with a new design designed for use with an IMU stack, the IMU stack was much more stable and this design was accepted		

Table B.13: Test report IMU attenuation in a fossil fuel car

Test ID	T.1.8	**Date**	24.04.19
Test Number	8	**Result**	Accepted
Testers Name	Jan Olav Møen & Vegar Skogen		
Comment	Tested with magnetometer, IMU stack and the new vibration attenuation solution in a fossil fuel car, had no drifting except in start up which we consider as initializing of the IMU and is accepted		

B.3 Camera Mounting Testing

Table B.14: Testing camera mounting solution

Test ID	T.2	**Priority**	A
Requirements tested	F.3.1, F.3.1.1, F.3.1.2, F.3.1.3, F.3.3		
Test description	Mounting cameras on the car		
Resources	Camera boxes, car mounting solution, two team members		
Test steps	1. Mount camera box on car 2. Open camera boxes mounted on the car 3. Change the light sensitivity on the cameras 4. Close the camera boxes mounted on the car		
Acceptance Criteria	• Mounting of the camera box on the car should take less than 5 minutes and not require any tools • You should be able to open the camera boxes, change the light sensitivity of the cameras and close the boxes again within 3 minutes and without using any tools • The camera boxes should be securely mounted		

Table B.15: Test report camera mounting and accessibility

Test ID	T.2	**Date**	16.05.19
Test Number	1	**Result**	Accepted
Testers Name	Marco Tande & Kenneth Halvorsen		
Comment	Tested mounting of camera boxes on Argos car, tested light sensitivity switch. Test did not exceed 3 minutes		

B.4 Camera Protection Testing

Table B.16: Testing camera protection

Test ID	T.3	**Priority**	A
Requirements tested	F3.2, F3.2.1, F3.2.2		
Test description	Take the camera box outside in the rain or under a garden hose		
Resources	Camera boxes, mounting solution, water, water repellent, isopropyl alcohol		
Test steps	1. Take the camera boxes outside make sure the cameras not are mounted inside it yet 2. You need water to spray on the camera boxes 3. Inspect that no water got inside the boxes 4. Put the cameras inside the box 5. Connect the cameras to the Argos VR system and pc 6. Spray water on the windscreen		
Acceptance Criteria	• The box should not become wet inside • The cameras should have a clear view always and not take damage		

Table B.17: Test report camera protection

Test ID	T.3.1	**Date**	19.05.19
Test Number	1	**Result**	Accepted
Testers Name	Kenneth Halvorsen		
Comment	Poured water over back camera box for 1 minute. Water did not get inside box		

Table B.18: Test report camera protection

Test ID	T.3.2	**Date**	21.05.19
Test Number	2	**Result**	Accepted
Testers Name	Kenneth Halvorsen & Vegar Skogen		
Comment	Tested visibility of camera while spraying water on windscreen after repellent treatment. Camera had clear view and did not take damage		

B.5 Camera Glass Testing

Table B.19: Testing water repellent

Test ID	T.4	Priority	A
Requirements tested	F.3.2.2		
Test description	Testing to see if the water repellent works on a specific glass material		
Resources	Desired glass material, water repellent, isopropyl alcohol		
Test steps	1. Wash glass with isopopyl alcohol 2. Take a drop of water on the glass and check the wetting angle of the water drop 3. Wash the glass with isopropyl alcohol 4. Rub the glass with water repellent coating 5. Take a water drop on the glass, inspect and check the wetting angle 6. Evaluate results		
Acceptance Criteria	• The wetting angle should be higher when using the water repellent		

Table B.20: Test report water repellent

Test ID	T.4.1	Date	11.03.19
Test Number	1	Result	Accepted
Testers Name	Kenneth Halvorsen & Marco Tande		
Comment	Contact angle increased significantly using cheap brand water repellent		

Table B.21: Test report water repellent

Test ID	T.4.2	**Date**	11.03.19
Test Number	2	**Result**	Accepted
Testers Name	Kenneth Halvorsen		
Comment	Contact angle increased significantly using expensive brand water repellent		

B.6 Battery Monitoring Testing

Table B.22: Testing battery monitoring

Test ID	T.5	**Priority**	A
Requirements tested	F.2.3, F.2.3.1, F.2.3.2, F.2.3.2.3, F.2.3.2.4, F.2.1.3		
Test description	Drain the battery and monitor the results with a battery monitor circuit		
Resources	Battery monitoring circuit, batteries, Arduino, computer, load for circuit		
Test steps	1. Connect the battery monitoring circuit to the batteries and a load 2. Connect the Arduino to the circuit 3. Turn everything on and slowly increase the load on the circuit 4. Observe measurements done 5. Run test multiple times to observe consistency 6. Evaluate results		
Acceptance Criteria	• The circuit shall withstand the high currents and not get destroyed in the process • The measurements shall be observable on the computer • The measurements shall be consistent		

Table B.23: Test report battery monitoring

Test ID	T.5	**Date**	28.03.2019
Test Number	1	**Result**	Failed
Testers Name	Emil Hultin & Vetle Tørjesen		
Comment	Measurements were not correct		

Table B.24: Test report battery monitoring

Test ID	T.5	**Date**	03.05.2019
Test Number	2	**Result**	Failed
Testers Name	Emil Hultin & Vetle Tørjesen		
Comment	Measurements were not correct		

Table B.25: Test report battery monitoring

Test ID	T.5	**Date**	09.05.2019
Test Number	3	**Result**	Conclusive
Testers Name	Emil Hultin & Vetle Tørjesen		
Comment	Measurements were correct and consistent		

B.7 Power Source Mounting Testing

Table B.26: Testing power source system mounting, dismounting and emergency stops

Test ID	T.6	**Priority**	A
Requirements tested	F.2.2, F.2.2.2, F.2.2.3		
Test description	Mount the power source system in a car and preform emergency stop then dismount the system		
Resources	Power source system, mounting equipment		
Test steps	1. Mount the power source system in the vehicle and fasten it 2. Fasten a security strap 3. Speed the car up to 40 km/h 4. Check the surroundings if it is safe to preform an emergency break and alert the passengers 5. Preform a emergency break in a safe environment 6. Dismount the power source system from the car		
Acceptance Criteria	• The power source system shall be easy to mount/dismount • The power source system shall stay in place while preforming emergency break		

Table B.27: Test report power source system mounting

Test ID	T.6	**Date**	
Test Number	1	**Result**	Inconclusive
Testers Name	Emil Hultin & Vetle Tørjesen		
Comment	The project ended before a test could be performed		

B.8 Power Source Interference Testing

Table B.28: Testing interference on power source system

Test ID	T.7	**Priority**	A
Requirements tested	F.2.1, F.2.1.1		
Test description	Measure interference on input and output of the power system		
Resources	Power source system, oscilloscope		
Test steps	1. Connect the power source system to batteries and a load circuit 2. Turn on the power supply and measure first without load in the circuit 3. measure with high load 4. Evaluate Results		
Acceptance Criteria	• Interference and noise should be as low as possible		

Table B.29: Test report power interference

Test ID	T.7	**Date**	16.05.1029
Test Number	1	**Result**	Conclusive
Testers Name	Emil Hultin, Vele Tørjesen & Vegar Skogen		
Comment	The interference from the new inverter was much lower than the old. $V_P P$ at 0.7V		

B.9 Power Source Runtime Testing

Table B.30: Testing run-time for power source system

Test ID	T.8	**Priority**	A
Requirements tested	F.2.1.2, F.2.1.2.2		
Test description	Measure run-time for power system with load		
Resources	Whole system, stop watch		
Test steps	1. Rig up the whole system with fully charged batteries 2. Turn on the whole system and run it for as long as possible while timing it with a stop watch 3. Evaluate Results		
Acceptance Criteria	• The system should stay on longer than one hour		

Table B.31: Test report power run-time

Test ID	T.8	**Date**	
Test Number	1	**Result**	Inconclusive
Testers Name	Emil Hultin & Vetle Tørjesen		
Comment	Project ended before a test could be done		

B.10 Power Source Efficiency Testing

Table B.32: Testing efficiency for power source system

Test ID	T.9	**Priority**	A
Requirements tested	F.2.1.2.1, F.2.1.2.1.1		
Test description	Measure watt usage for power system		
Resources	Power source system, multimeter		
Test steps	1. Connect up the power source system without load 2. turn the power and measure voltage and current usage 3. Calculate watt usage 4. Evaluate results		
Acceptance Criteria	• The system should use as little wattage as possible		

Table B.33: Test report power efficiency

Test ID	T.9	**Date**	05.02.2019
Test Number	1	**Result**	Inconclusive
Testers Name	Vetle Tørjesen, Emil Hultin & Vegar Skogen		
Comment	We measured the watt usage of the computer with a watt-meter. At peak performance the watt output was 441 watts		

B.11 Hot-Swap Testing

Table B.34: Testing hot-swapping batteries

Test ID	T.10	**Priority**	B
Requirements tested	F.2.1.2.3		
Test description	Hot-swap batteries without interruption of the system		
Resources	Power source system, batteries, battery hot-swap circuit		
Test steps	1. Connect up the power source system without load 2. Turn the power on 3. Send signal to change to a different battery 4. Replace the battery not in use 5. Switch back to the newly swapped battery 6. Evaluate results		
Acceptance Criteria	• The system shall not turn off while swapping • The battery shall be measured correctly charge percentage • The circuit shall not be destroyed in the process		

Table B.35: Test report battery hot-swap

Test ID	T.10	**Date**	16.05.2019
Test Number	1	**Result**	Inconclusive
Testers Name	Emil Hultin & Vetle Tørjesen		
Comment	We wired the relays and ran a switching test, where every activated as it should, but without batteries connected.		

B.12 HUD Marking Testing

Table B.36: Testing HUD Markers

Test ID	T.11	**Priority**	A
Requirements tested	F.1.5		
Test description	Display Tracking Markers		
Resources	GPS, Argos Computer		
Test steps	1. Launch the application 2. Enable tracking 3. Look for HUD trackers 4. Look at HUD trackers 5. Close the application		
Acceptance Criteria	• The markers shall become visible shortly after enabling marking • The markers shall be rotated towards the operator • The markers shall be positioned correctly relative to each other • The markers shall be scaled dynamically by their distance from the operator • The markers shall be positioned correctly relative to the compass • Information about the marker shall be displayed when looked at • The Information about the marker shall be correct		

Table B.37: Test report HUD Markers

Test ID	T.11	**Date**	27.02.19
Test Number	1	**Result**	Failed
Testers Name	Vegar Skogen		
Comment	Failed at step 3, the markers did not become visible		

Table B.38: Test report HUD Markers

Test ID	T.11	**Date**	28.02.19
Test Number	2	**Result**	Failed
Testers Name	Vegar Skogen		
Comment	Failed at step 4, the information about the markers did not appear		

Table B.39: Test report HUD Markers second test

Test ID	T.11	**Date**	29.02.19
Test Number	3	**Result**	Failed
Testers Name	Vegar Skogen		
Comment	Failed at step 4, the information about the marker was incorrect		

Table B.40: Test report HUD Markers third test

Test ID	T.11	**Date**	5.03.19
Test Number	4	**Result**	Failed
Testers Name	Vegar Skogen		
Comment	Failed at step 4, the markers appear in the wrong positions relative to the compass		

B.13 Camera View Angle Testing

Table B.41: Testing view angle

Test ID	T.13	**Priority**	A
Requirements tested	F.1.8.2		
Test description	Testing view angle for cameras		
Resources	Manufactured test model, cameras, Argos pc, VR glasses		
Test steps	1. Mount the cameras to the new rig 2. Connect the cameras to the Argos pc with Ethernet 3. Open Argos application and steam VR 4. Inspect the picture 5. Evaluate results		
Acceptance Criteria	• The camera feed should be fluid without any blind zones or overlaps		

Table B.42: Test view angle

Test ID	T.13.1	**Date**	27.02.19
Test Number	1	**Result**	Failed
Testers Name	Marco Tande & Vegar Skogen		
Comment	Testing the view angle in the former front camera box, blind spot		

Table B.43: Test view angle 2

Test ID	T.13.2	**Date**	11.03.19
Test Number	2	**Result**	Failed
Testers Name	Marco Tande & Vegar Skogen		
Comment	Testing the angle of view for new solution 81.9°, too much overlapping		

Table B.44: Test view angle 3

Test ID	T.13.3	**Date**	13.03.19
Test Number	3	**Result**	Failed
Testers Name	Marco Tande & Vegar Skogen		
Comment	Increased the angle of view from 81.9°to 95°, too much overlapping		

Table B.45: Test view angle 4

Test ID	T.13.4	**Date**	14.03.19
Test Number	4	**Result**	Inconclusive
Testers Name	Marco Tande & Vegar Skogen		
Comment	Increased angle of view to 102.9°, it was overlapping higher in the seam and dead spot lower		

Table B.46: Test view angle 5

Test ID	T.13.5	**Date**	25.03.19
Test Number	5	**Result**	Accepted
Testers Name	Marco Tande & Vegar Skogen		
Comment	Went down from 102.9°to 98°, Since we had too much overlapping at 95°and both overlapping and blind spot in the middle at 102.9°. We have a small blind spot for about 500mm our from the camera view and then overlapping, this is accepted		

B.14 VBS Simulation

Table B.48: Test BMS Simulation

Test ID	T.14.1	**Date**	02.04.19
Test Number	1	**Result**	Failed
Testers Name	Jan Olav Møen & Vegar Skogen		
Comment	VBS stopped responding with generic error		

Table B.47: Test BMS Simulation

Test ID	T.14	**Priority**	A
Requirements tested	F.1.1, F.1.2, F.1.4		
Test description	Test that the VBS Plugin simulates and sends data to Argos, and that the data is correct. this will also test if Argos displays it correctly.		
Resources	Car PC, Argos PC		
Test steps	1. Move plugin to folder 2. start VBS3 3. start VBS mission 4. start Argos		
Acceptance Criteria	• VBS should print information about markers and vehicles • Argos should receive information about markers and vehicles • Argos should display the markers and vehicles • the loop should continue until we exit one of the applications		

Table B.49: Test BMS Simulation

Test ID	T.14.2	**Date**	03.04.19
Test Number	2	**Result**	Failed
Testers Name	Jan Olav Møen & Vegar Skogen		
Comment	VBS stopped responding with the same generic error		

Table B.50: Test BMS Simulation

Test ID	T.14.3	**Date**	05.04.19
Test Number	3	**Result**	Failed
Testers Name	Jan Olav Møen & Vegar Skogen		
Comment	worked for a time, but VBS and Argos communication went out of sync		

Table B.51: Test BMS Simulation

Test ID	T.14.4	**Date**	09.04.19
Test Number	4	**Result**	Accepted
Testers Name	Jan Olav Møen & Vegar Skogen		
Comment	worked this time, but more data will should be sent. Need new test after code change.		

Table B.52: Test BMS Simulation

Test ID	T.14.5	**Date**	24.04.19
Test Number	5	**Result**	Failed
Testers Name	Jan Olav Møen & Vegar Skogen		
Comment	worked for a time, but VBS and Argos communication went out of sync		

Table B.53: Test BMS Simulation

Test ID	T.14.6	**Date**	03.05.19
Test Number	6	**Result**	Accepted
Testers Name	Jan Olav Møen & Vegar Skogen		
Comment	worked, all data sent successfully and no de-sync problems		

Table B.54: Test Power Source Data System

Test ID	T.15	**Priority**	A
Requirements tested	F.2.3		
Test description	Test that Argos receives and displays battery percentage, temperature and humidity.		
Resources	Argos PC		
Test steps	1. connect the Arduino 2. start Argos		
Acceptance Criteria	• Argos should receive information about battery, temperature and humidity • Argos should display the information		

B.15 Power Source Data System

Table B.55: Test Power Source Data System

Test ID	T.15.1	**Date**	08.05.19
Test Number	1	**Result**	Failed
Testers Name	Jan Olav Møen		
Comment	Argos received, but didn't display the information		

Table B.56: Test Power Source Data System

Test ID	T.15.1	**Date**	09.05.19
Test Number	2	**Result**	Accepted
Testers Name	Jan Olav Møen		
Comment	Argos display the information		

C List of Risks

C.1 Hardware Risk

Table C.1: Hardware risks

ID	Risk	Probability	Severity	Risk level
RH.1	GPS failure	2	1	2
Mitigation	We have more GPS's to replace any failed GPS units			
RH.2	Development PC failure	1	1	1
Mitigation	All files are backed up on all computers and on KDA's server			
RH.3	Argos PC 1 failure	2	2	4
Mitigation	All files are backed up on all computers and on KDA's server and we have 2 other Argos pc's we can use			
RH.4	Argos PC car failure	3	3	9
Mitigation	We have a second pc capable of running VR			
RH.5	Argos laptop failure	3	2	6
Mitigation	All files are backed up on all computers and on KDA's server and we have 2 other Argos pc's we can use			
RH.6	Car failure	1	2	2
Mitigation	We have multiple cars at our disposal			
RH.7	Office network failure	1	4	4
Mitigation	We have several switches at our disposal			
RH.8	Camera failure	1	5	5
Mitigation	Making sure the requirements for the camera system are fulfilled and ordering repair or buying new ones			
RH.9	Mechanical connection failure	2	2	4
Mitigation	Running tests, simulations and prototyping			
RH.10	Power failure while testing in car	3	5	15
Mitigation	All files are backed up and someone without VR goggles are sitting in the passenger seat			
RH.11	Battery failure	1	4	4
Mitigation	Implementing fuses and having spare batteries			
RH.12	Camera box failure	1	3	3
Mitigation	Simulations and calculations			

RH.13	Water proof failure	3	5	15
Mitigation	Testing without the cameras inside to check for water-proofing and camera placement			
RH.14	Camera overheating	4	5	20
Mitigation	Temperature sensors, warning system if the temperature reaches a certain level and active cooling			

C.2 Software Risk

Table C.2: Software risks

ID	Risk	Probability	Severity	Risk level
RS.1	Hg fail	4	2	8
Mitigation	We have backup and we can correct some corruption or create new repository			
RS.2	Too big changes in architecture to cope for software	3	4	12
Mitigation	Continuous conversation between hardware and software			
RS.3	Takes too long to implement BMS	3	4	12
Mitigation	Continuous prioritisation with costumer			
RS.4	BMS license	2	2	4
Mitigation	We will get other tasks from Alexander, he's trying his best to get a license			
RS.5	Serial in Unreal Engine	1	5	5
Mitigation	There are libraries to do it			
RS.6	Pre existing problems in software	3	2	6
Mitigation	Planned a long time to maintain all code			
RS.7	Official Argos website becomes corrupted	2	1	2
Mitigation	We have a backup			

C.3 Human Error Risk

Table C.3: Human Error risks

ID	Risk	Probability	Severity	Risk level
RE.1	Absent group member	4	2	8
Mitigation	We can have meetings over skype or other means			
RE.2	Supervisor is absent	2	4	8
Mitigation	We have regular meeting and ways to contact them			
RE.3	Important files deleted by accident	2	2	4
Mitigation	Google drive makes sure we can view past logs			
RE.4	Damage equipment	2	4	8
Mitigation	We try to have back up equipment and most equipment is easily replaceable			
RE.5	Someone fails to meet the deadline	3	4	12
Mitigation	We have meeting each morning telling how far we've come and what we need help with			

C.4 Project Management Risk

Table C.4: Project Management risks

ID	Risk	Probability	Severity	Risk level
RP.1	Delay in project plan	3	3	9
Mitigation	Because we are agile we should see delays early, being prepared for delayed parts and doing research on shipping before we order			
RP.2	We need to spend more than our budget allows	2	5	10
Mitigation	Plan what we need to buy, we use mainly off the shelf products			
RP.3	Delivery time	4	2	8
Mitigation	We work on other tasks in the meantime			
RP.4	Power outage in the bachelor room	3	2	6
Mitigation	Try to not use more electricity than we need			
RP.5	Disagreements between group members	3	2	6
Mitigation	We talk about everything and the majority decides, Alexander is the final arbiter			

C.5 Human Injury Risk

Table C.5: Human Injury risks

ID	Risk	Probability	Severity	Risk level
RI.1	VR goggles failure	2	3	6
Mitigation	We don't drive with VR on a public road and you always have one person without VR goggles beside you			
RI.2	Argos car PC failure	3	3	9
Mitigation	We don't drive with VR on a public road and you always have one person without VR goggles beside you			
RI.3	Camera failure	2	3	6
Mitigation	We don't drive with VR on a public road and you always have one person without VR goggles beside you			
RI.4	Electro shock from power system	2	3	6
Mitigation	Properly isolated wiring, not working with the power turned on			
RI.5	Camera box mounting system fails while driving	2	5	10
Mitigation	Simulation and testing			
RI.6	Cable disconnect and hit pedestrians	1	4	4
Mitigation	Cables are securely mounted			
RI.7	Sudden lag in camera feed	2	3	6
Mitigation	We don't drive with VR on a public road and you always have one person without VR goggles beside you			
RI.8	PC or power system in car isn't rigidly mounted	2	4	8
Mitigation	Check that it is fastened securely			
RI.9	Injury while working with materials	3	3	9
Mitigation	Follow HMS standards and USN workshop rules			
RI.10	Burn injury using soldering iron	4	2	8
Mitigation	Being careful when handling a soldering iron			

D Project Budget

This year Argos got a budget of 40.000,- NOK shared for all the disciplines. It was established early on by our employer that if we need a larger budget we could come to an agreement. We started of the project by making an estimate of what we assumed we were going to use see Fig. D.1.

ARGOS 3 BUGDET	Ting	Stykkpris	Antall	Total
Elektro	Inverter	10 000,00 kr	1	10 000,00 kr
	Annet Elektro	10 000,00 kr	1	10 000,00 kr
		0,00 kr		0,00 kr
Administrativt	T-skjorte	165,00 kr	7	1 155,00 kr
	Genser	300,00 kr	7	2 100,00 kr
		0,00 kr		0,00 kr
				0,00 kr
Maskin overslag	Plast glass	349,00 kr	2	698,00 kr
	mutter- og skivesortiment	129,00 kr	1	129,00 kr
	skruer/bolter	20,00 kr	15	300,00 kr
	Carbon fiber	0,00 kr		0,00 kr
	Aluminium extrution	170,00 kr	3	510,00 kr
	Alu plater	407,00 kr	2	814,00 kr
	Vindu rens	20,00 kr	1	20,00 kr
	Rain X water repellent for plastic	300,00 kr	1	300,00 kr
	Filament for 3d print	250,00 kr	2	500,00 kr
	Takfeste	229,00 kr	4	916,00 kr
	Gummiknotter	26,00 kr	1	26,00 kr
	Gummilister/tetningslister	200,00 kr	1	200,00 kr
	Buffer	5 000,00 kr	1	5 000,00 kr
		0,00 kr		0,00 kr
		0,00 kr		0,00 kr
				0,00 kr
	Total	27 565,00 kr		32 668,00 kr
	Gjenstående			7 332,00 kr
	4 413,00 kr			

Figure D.1: Argos estimated budget

In the last weeks of our project we exceeded our budget. After a meeting with our employer we agreed it was more important to finish the project, so we were allowed to exceed the budget see Fig. D.2.

Piquet	385,00 kr	7	2 695,00 kr
Hettejakke	645,00 kr	7	4 515,00 kr
Elektro 1	5 920,90 kr	1	5 920,90 kr
Elektro 2	7 116,00 kr	1	7 116,00 kr
Elektro 3	1 515,92 kr	1	1 515,92 kr
Elektro 4	737,80 kr	1	737,80 kr
Elektro 5	624,87 kr	1	624,87 kr
Elektro 6	1 809,41 kr	1	1 809,41 kr
Elektro 8	819,22 kr	1	819,22 kr
Elektro Biltema	752,80 kr	1	752,80 kr
PCB1	961,30 kr	1	961,30 kr
PCB2	712,47 kr	1	712,47 kr
Rain X	298,00 kr	1	298,00 kr
Rollco	5 935,00 kr	1	5 935,00 kr
Biltema 1	536,80 kr	1	536,80 kr
Clas ohlson 1	391,00 kr	1	391,00 kr
Elektro 7	1 700,00 kr	1	1 700,00 kr
Plexiglas 1	653,00 kr	1	653,00 kr
Plexiglas 2	850,00 kr	1	850,00 kr
Handlet på Jernia	325,61 kr	1	325,61 kr
Clas ohlson 2 & MaxBo	198,80 kr	1	198,80 kr
AL plater	1 442,00 kr	1	1 442,00 kr
PCB camerabox	872,38 kr	1	872,38 kr
Elektro 9	1 114,32 kr	1	1 114,32 kr
		Total Brukt	42 497,60 kr
		El Brukt	24 657,39 kr
		HW Brukt	7 813,80 kr
		Gjenstående	−2 497,60 kr
		ink. mva	53 122,00 kr

Figure D.2: Argos budget

D.1 Project Work Log

Fig. D.3 shows a pie charts containing all the group members work hours up to 23.May.

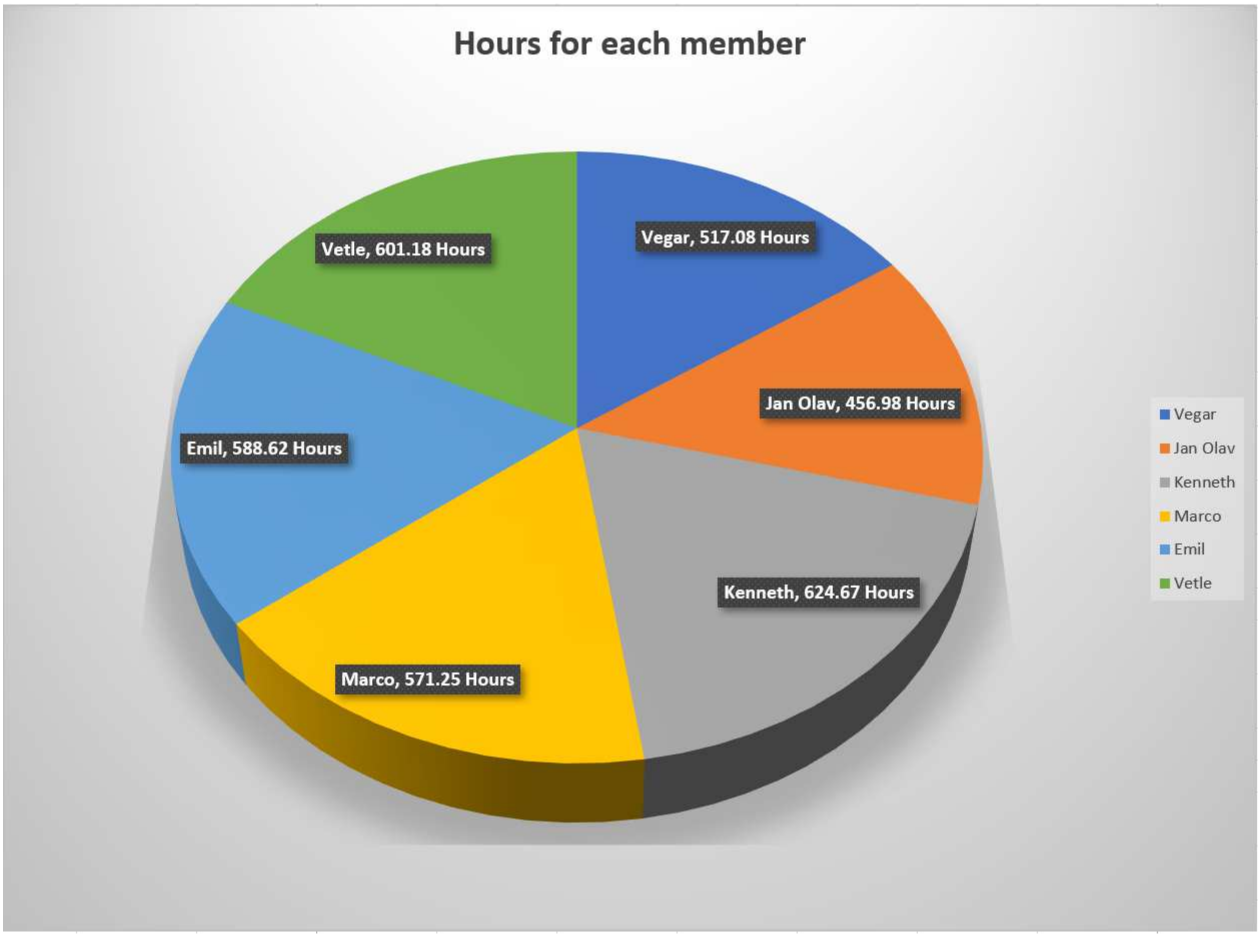
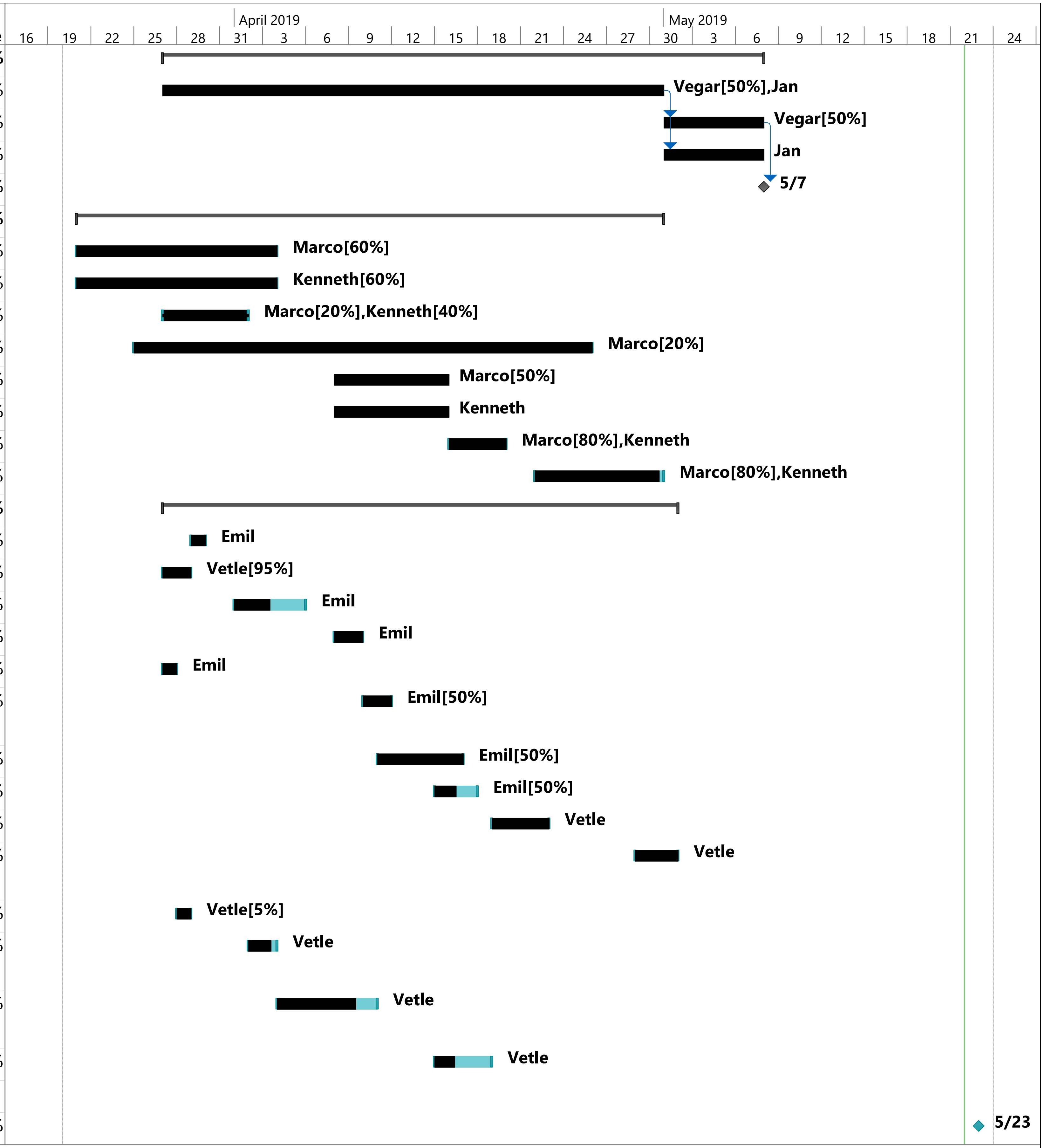


Figure D.3: Work log pie chart for all group members

E Project Plan

ID		Task Name	Duration	Start	Finish	Resource Names	% Complete
1	✓	**Software**	**30 days**	**Wed 3/27/19**	**Tue 5/7/19**		**100%**
2	✓	Collect, Send & Receive Data	5 wks	Wed 3/27/19	Tue 4/30/19	Vegar[50%	100%
3	✓	Memory Leak	1 wk	Wed 5/1/19	Tue 5/7/19	Vegar[50%	100%
4	✓	Sensor Data HUD	1 wk	Wed 5/1/19	Tue 5/7/19	Jan	100%
5	✓	GigeVision Playback	0 days	Tue 5/7/19	Tue 5/7/19	Vegar	100%
6		**Mechanical**	**29 days**	**Thu 3/21/19**	**Tue 4/30/19**		**99%**
7	✓	Finish Design Front Box	10 days	Thu 3/21/19	Wed 4/3/19	Marco[60	100%
8	✓	Finish Design Back Box	10 days	Thu 3/21/19	Wed 4/3/19	Kenneth[6	100%
9	✓	Finish Design Roof Mounting	4 days	Wed 3/27/19	Mon 4/1/19	Marco[20	100%
10	✓	IMU Stabilization	24 days	Mon 3/25/19	Thu 4/25/19	Marco[20	100%
11	✓	Production Front Box	6 days	Mon 4/8/19	Mon 4/15/19	Marco[50	100%
12	✓	Production Back Box	6 days	Mon 4/8/19	Mon 4/15/19	Kenneth	100%
13	✓	Production Roof Mount	4 days	Tue 4/16/19	Fri 4/19/19	Marco[80	100%
14		Final Assembly Testing	7 days	Mon 4/22/19	Tue 4/30/19	Marco[80	99%
15		**Electrical**	**26 days**	**Wed 3/27/19**	**Wed 5/1/19**		**76%**
16	✓	Produce BCCB	1 day	Fri 3/29/19	Fri 3/29/19	Emil	100%
17	✓	Program BCCB	2 days	Wed 3/27/19	Thu 3/28/19	Vetle[95%	100%
18		Test BCCB	5 days	Mon 4/1/19	Fri 4/5/19	Emil	50%
19	✓	Design Hot Swap Curcuit	2 days	Mon 4/8/19	Tue 4/9/19	Emil	100%
20	✓	Program Hot Swap	1 day	Wed 3/27/19	Wed 3/27/19	Emil	100%
21	✓	Integrate Hot Swap Into BCCB	2 days	Wed 4/10/19	Thu 4/11/19	Emil[50%]	100%
22	✓	Produce complete BCCB	4 days	Thu 4/11/19	Tue 4/16/19	Emil[50%]	100%
23		Test Complete BCCB	3 days	Mon 4/15/19	Wed 4/17/19	Emil[50%]	50%
24	✓	Camera Enclosure Monitoring	2 days	Fri 4/19/19	Mon 4/22/19	Vetle	100%
25	✓	Finish Design Temperature Regulation	3 days	Mon 4/29/19	Wed 5/1/19	Vetle	100%
26	✓	Order Components	1 day	Thu 3/28/19	Thu 3/28/19	Vetle[5%]	100%
27		Produce Temperature Regulator	2 days	Tue 4/2/19	Wed 4/3/19	Vetle	80%
28		Program Temperature Regulation	5 days	Thu 4/4/19	Wed 4/10/19	Vetle	70%
29		Test Temperature Regulation	4 days	Mon 4/15/19	Thu 4/18/19	Vetle	30%
30							
31		Final Delivery	0 days	Thu 5/23/19	Thu 5/23/19	Vegar,Vet	0%



Project: Argos3 new plan
Date: Wed 5/22/19

Task		Project Summary		Manual Task		Start-only		Deadline	
Split		Inactive Task		Duration-only		Finish-only		Progress	
Milestone		Inactive Milestone		Manual Summary Rollup		External Tasks		Manual Progress	
Summary		Inactive Summary		Manual Summary		External Milestone			

F 2D Blueprints

Following are the blueprints for the mechanical design for project Argos the bachelor student created in 2019.

F.1 Front Camera Box

In the following section are the blueprints for the design of the front camera box.

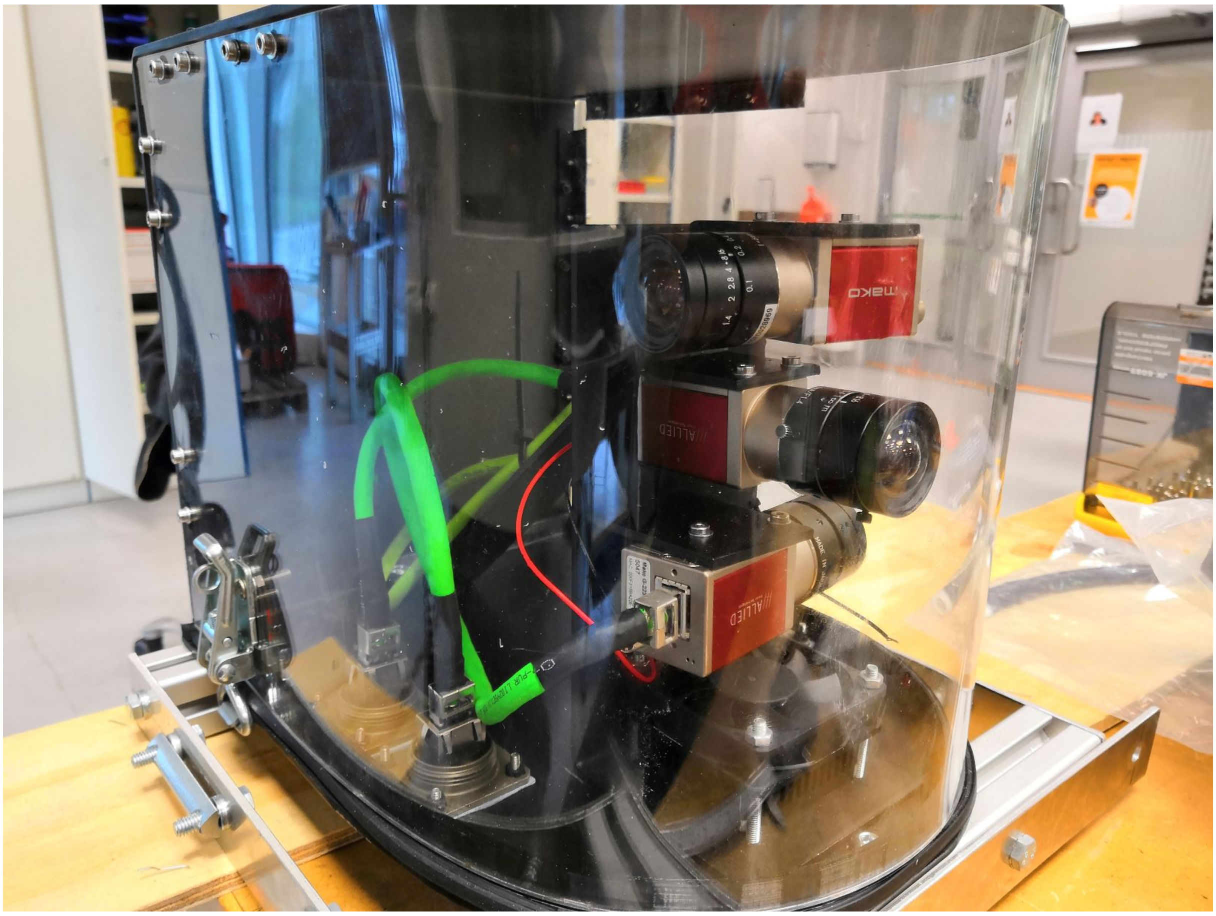
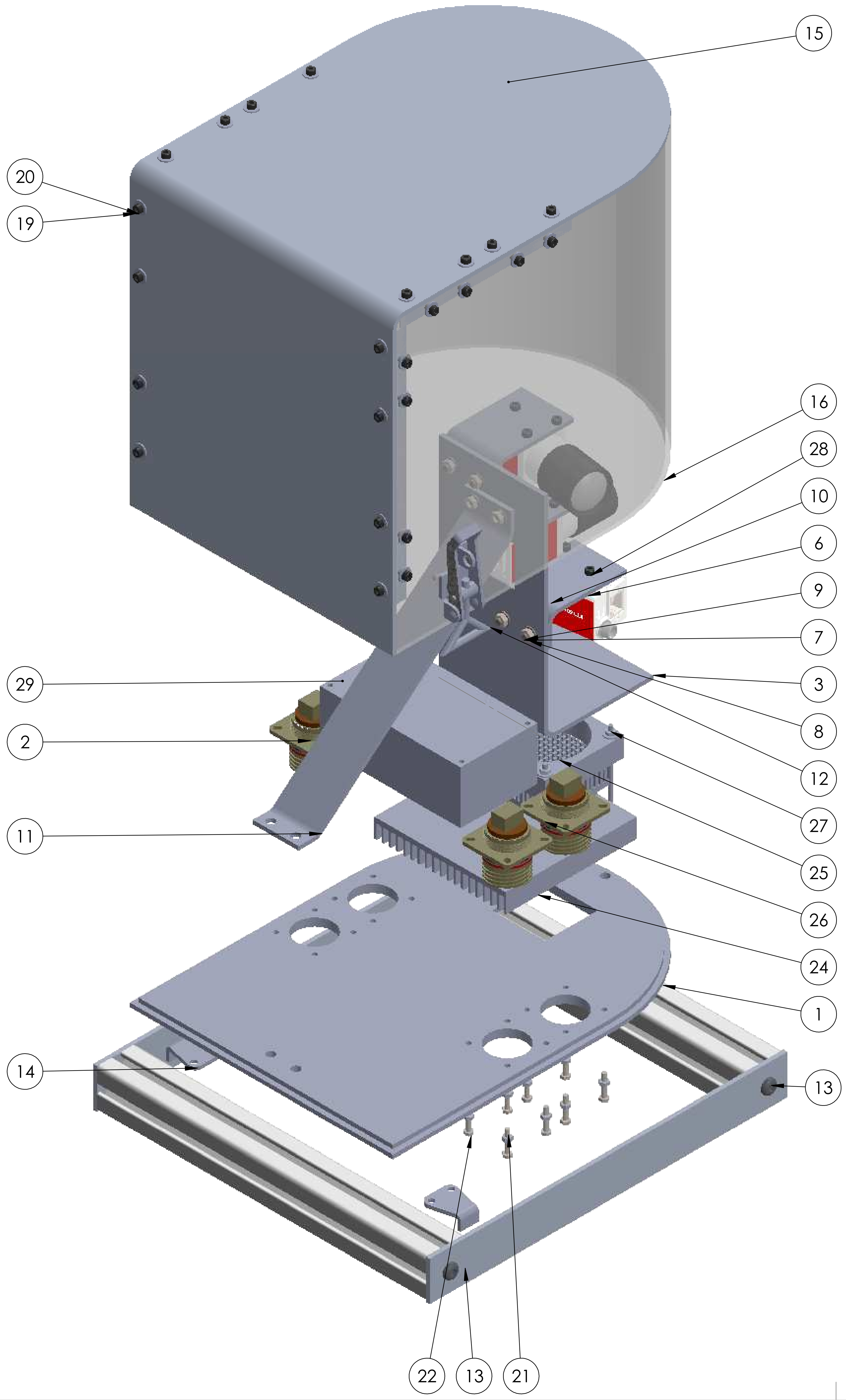


Figure F.1: Produced front camera box for the Argos system



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	Bottom plate		1
2	HRM20W19-11SN	CONNECTOR ETHERNET RJFTV21G FEMALE RJ45 OLIVE DRAB CADMIUM	4
3	Producable camera stand		1
4	Upper camera stand		1
5	Middle camera stand		1
6	Lower camera stand		1
7	Bolt M4-12mm	SCREW DIN912-M3*12-A4-BLACKENING KVS-50	10
8	Washer M4		12
9	Nut M4		10
10	Camera_small lens		3
11	Support for camera stand producable		1
12	New eccentric lock	31-1464	2
13	Front box mounting for roof rack		1
14	Locking arm for front camera box		2
15	Front box lid producable		1
16	Front box glass producable		1
17	Vinkelprofil 10x10x1		2
18	Vinkelprofil 10x10x1		2
19	Bolt M3-10mm	SCREW DIN912-M3*12-A4-BLACKENING KVS-50	32
20	Washer M3		64
21	Nut M3 locking		48
22	Bolt M3-16mm		16
23	peltier elem		1
24	heatsink Under the camera box		1
25	Fan 60x60x15mm		1
26	Heatsink for the inside of the camera box		1
27	Bolt M3-60mm for Peltier		4
28	Bolt M3-6mm	SCREW DIN912-M3*12-A4-BLACKENING KVS-50	9
29	circuit placeholder		1

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

MATERIAL:

TITLE:

DWG NO.

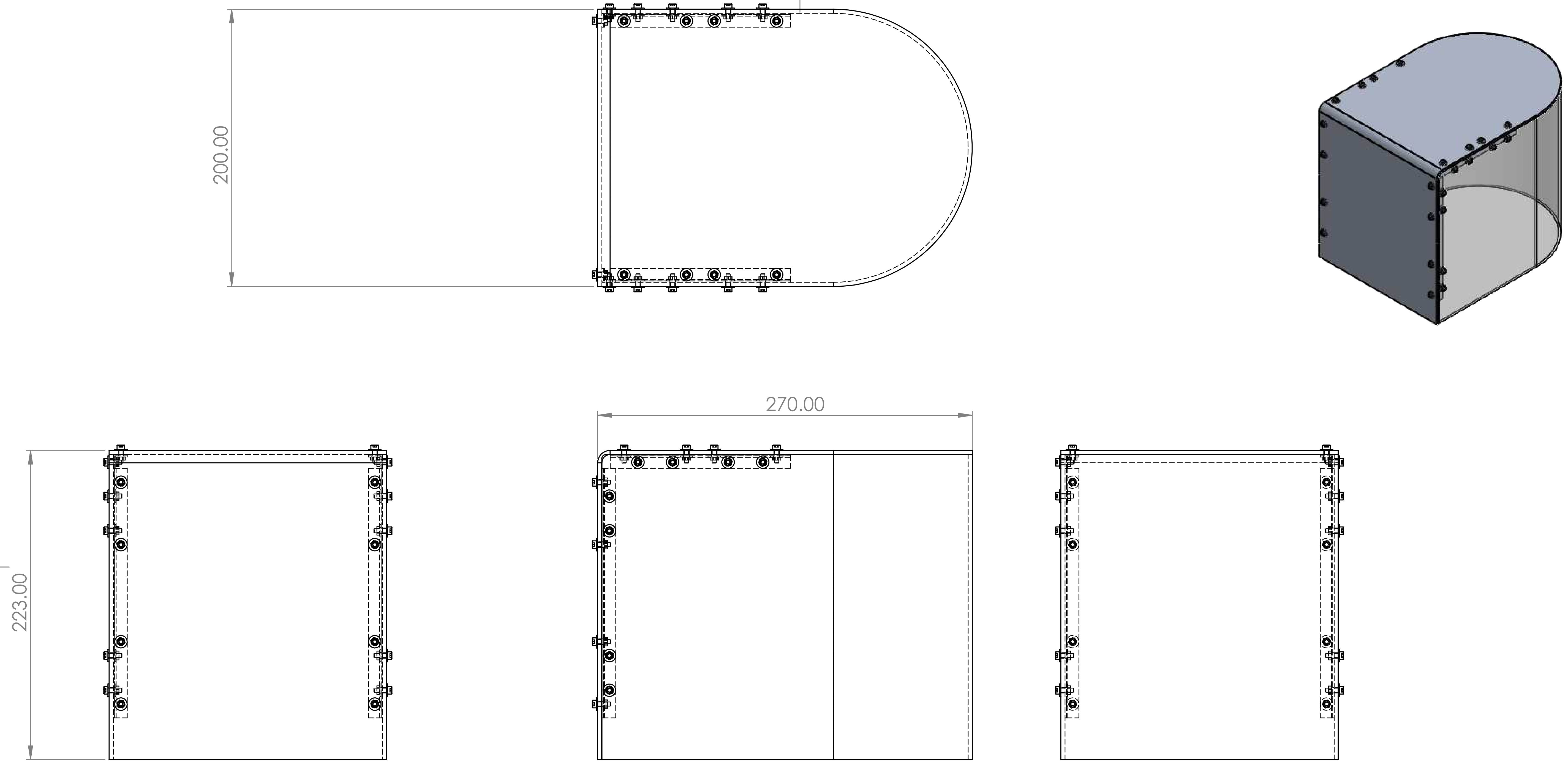
Camera front box V6

A2

WEIGHT:

SCALE:1:2

SHEET 2 OF 3



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	Front box lid producable		1
2	Front box glass producable		1
3	Vinkelprofil 10x10x1		2
4	Vinkelprofil 10x10x1		2
5	Bolt M3-10mm	SCREW DIN912-M3*12-A4-BLACKENING KVS-50	32
6	Washer M3		64
7	Nut M3 locking		32

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

MATERIAL:

DWG NO.

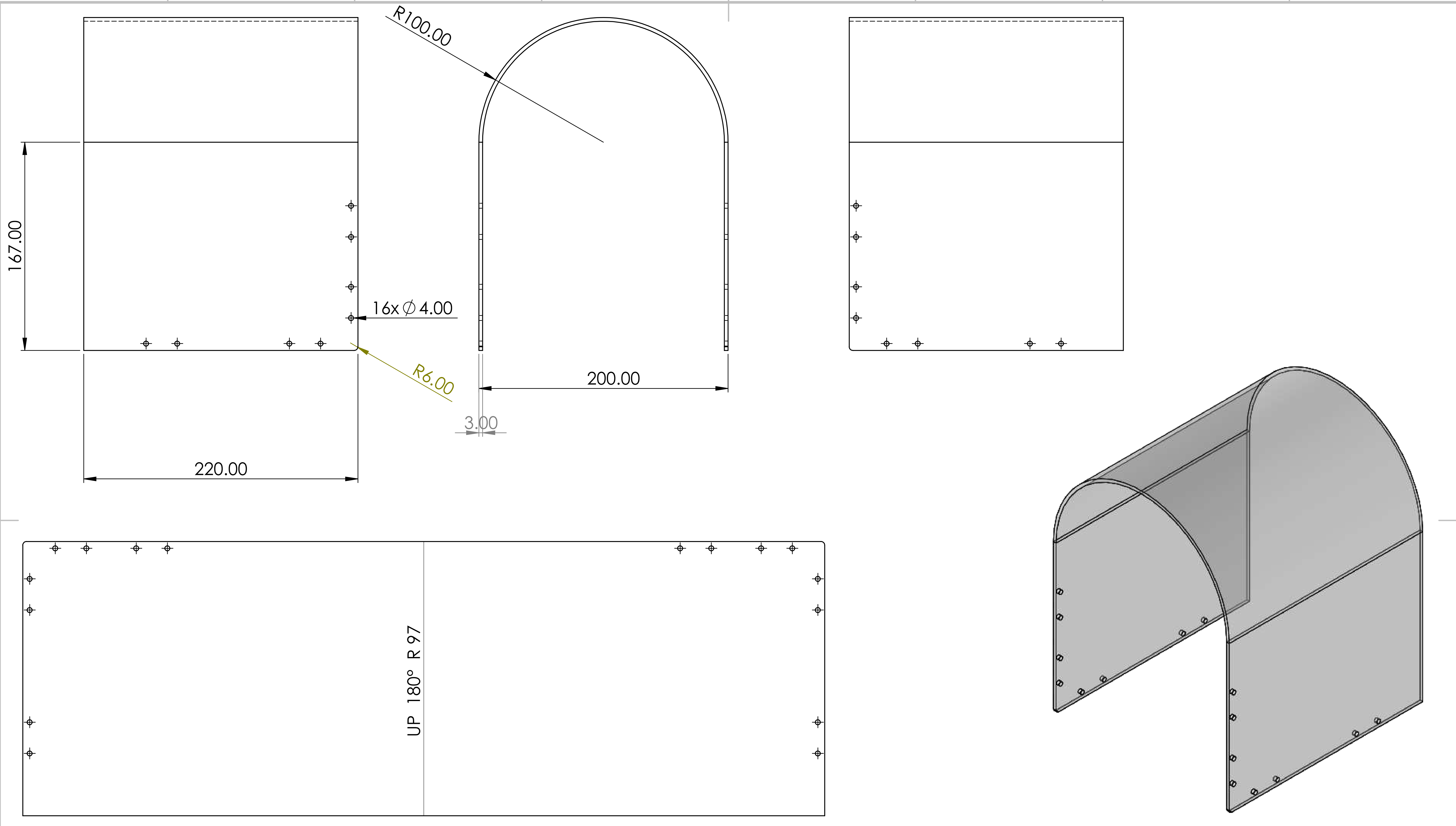
Lid for front box

A3

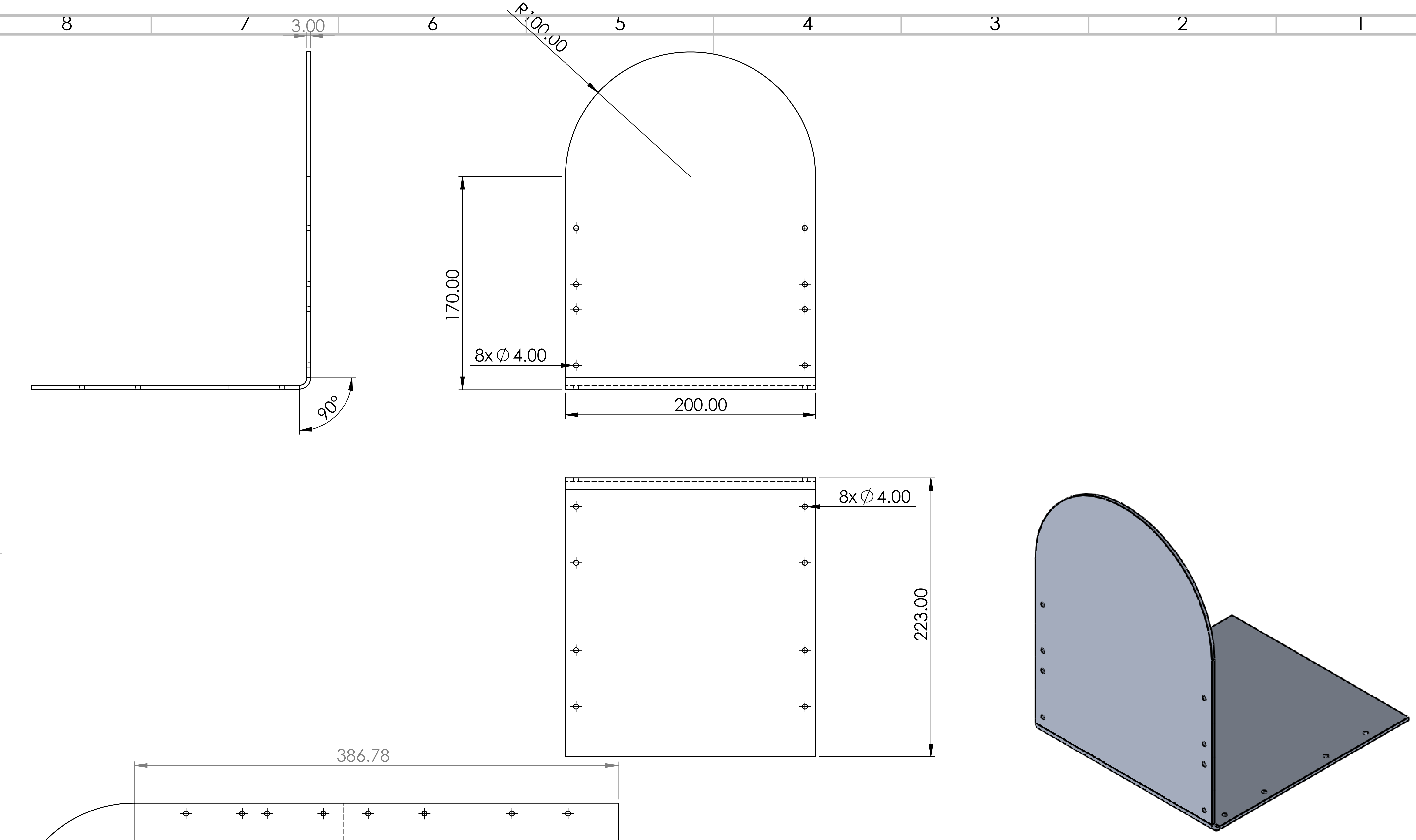
WEIGHT:

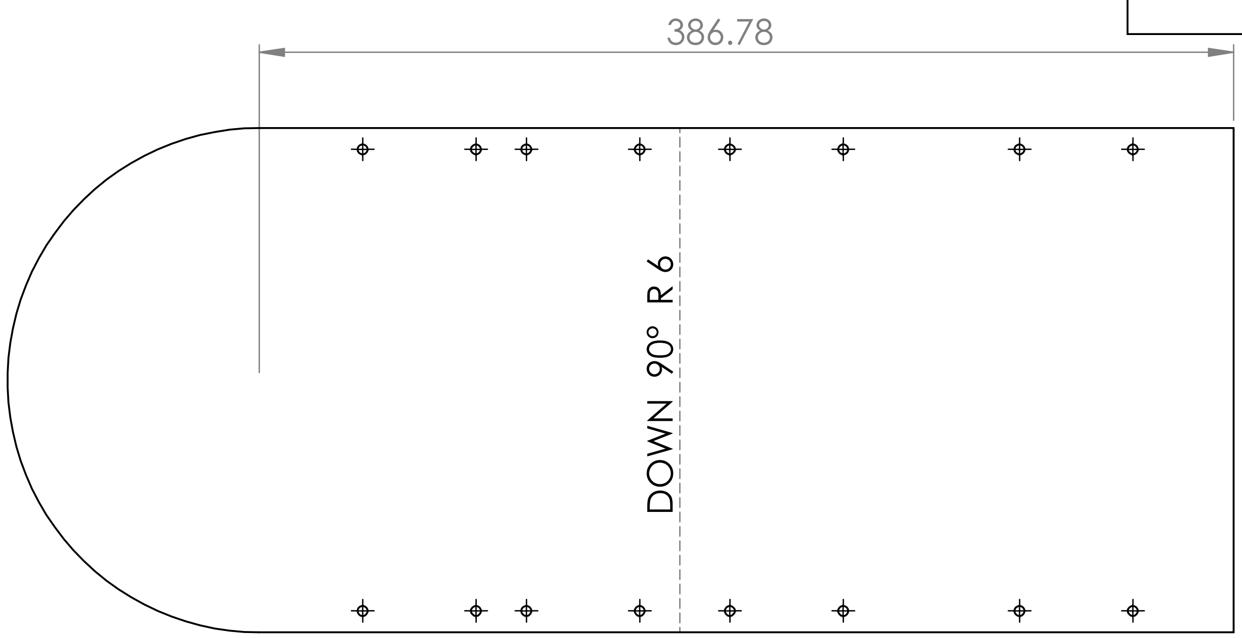
SCALE:1:3

SHEET 1 OF 6

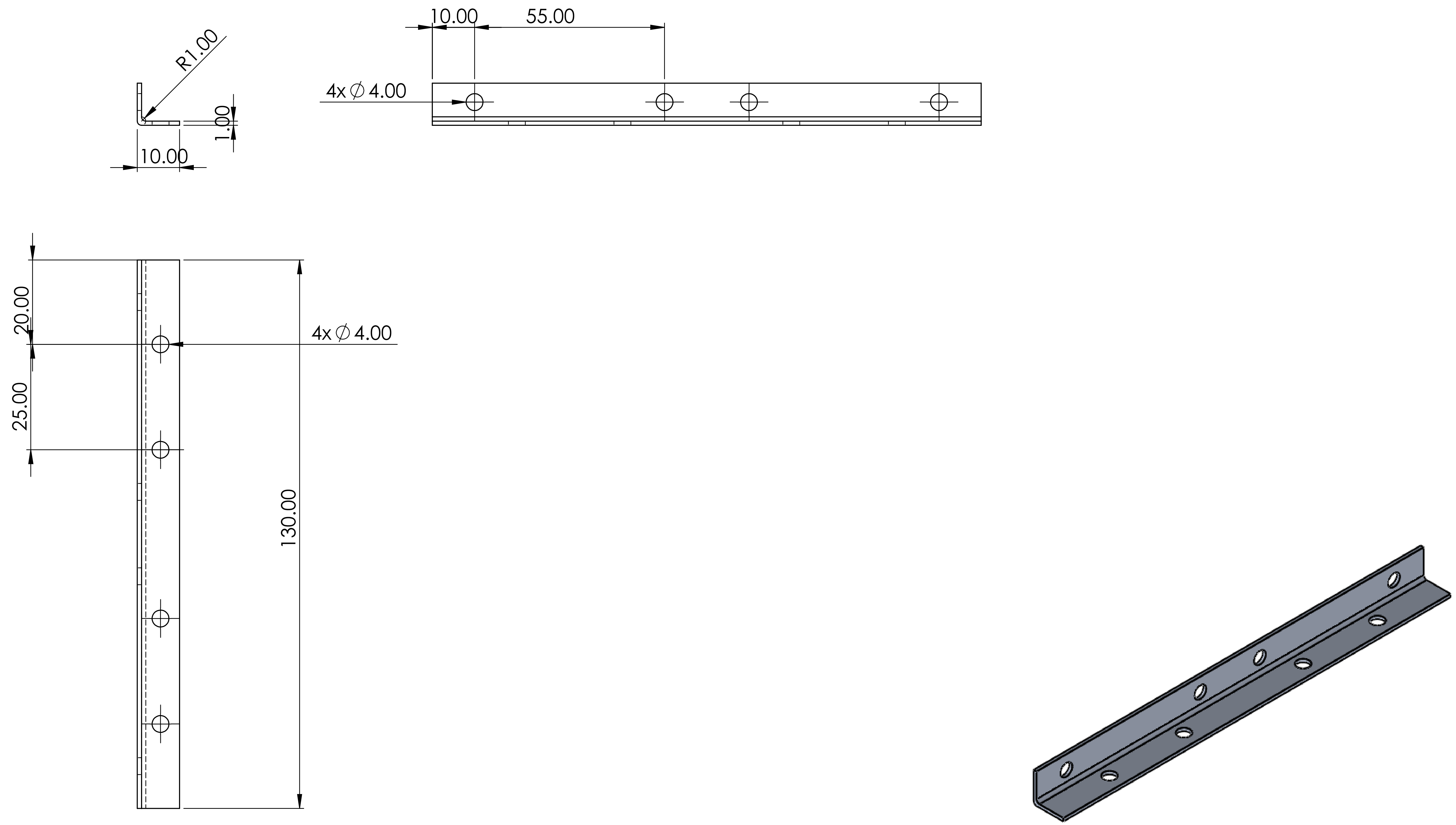


UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:		DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
	NAME	SIGNATURE	DATE	TITLE:	
DRAWN					
CHK'D					
APPV'D					
MFG					
Q.A				MATERIAL:	DWG NO. Glass front box	A3
				WEIGHT:	SCALE:1:3	SHEET 2 OF 6

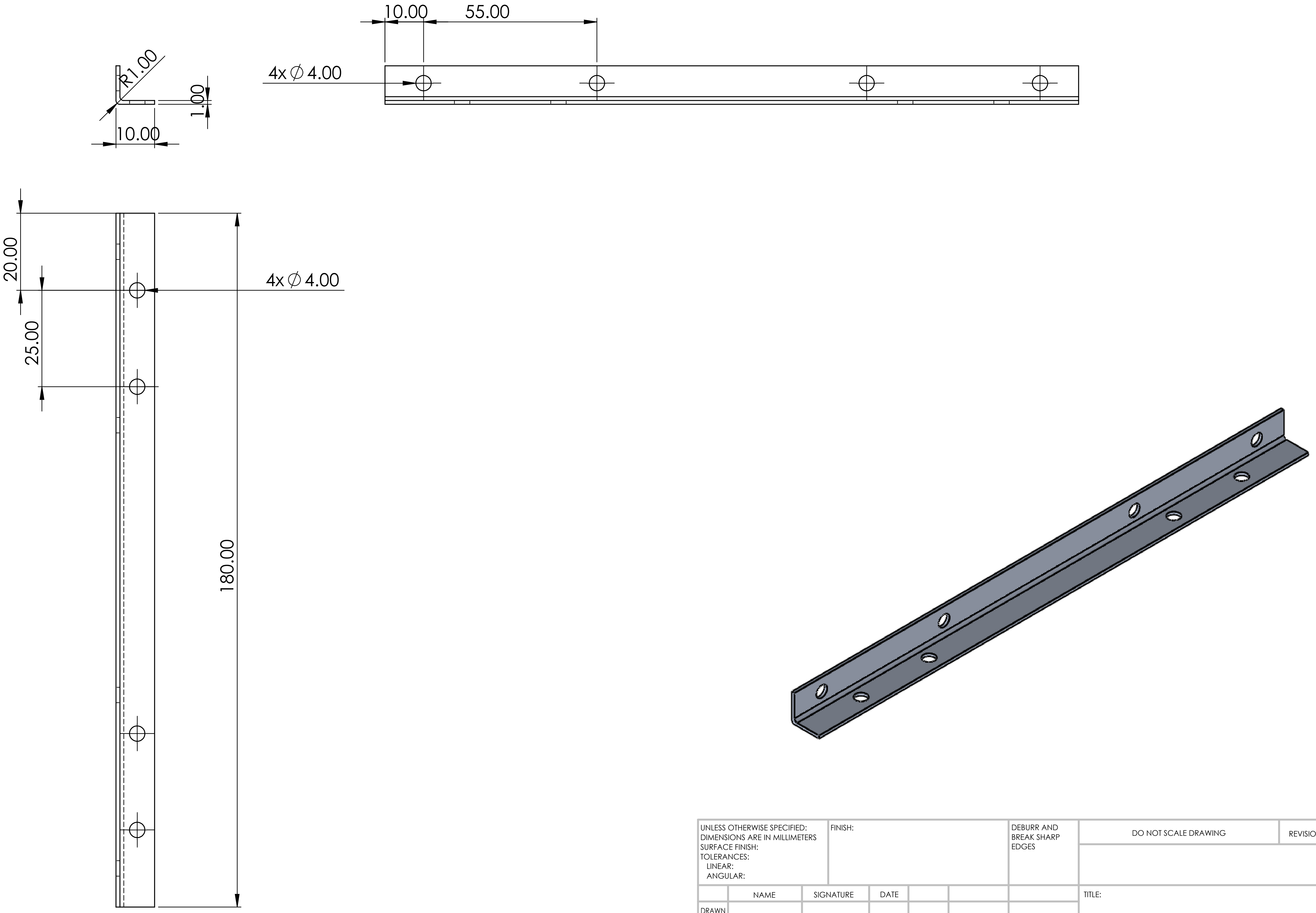




UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:		DEBURR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING	REVISION
	NAME	SIGNATURE	DATE		TITLE:	
DRAWN						
CHK'D						
APPV'D						
MFG						
Q.A				MATERIAL:	DWG NO. Lid for front box	A3
				WEIGHT:	SCALE:1:3	SHEET 3 OF 6



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:			FINISH:			DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
	NAME	SIGNATURE	DATE				TITLE:	
DRAWN								
CHK'D								
APPV'D								
MFG								
Q.A				MATERIAL:			DWG NO. Angle bracket 130mm	A3
				WEIGHT:			SCALE:1:1	SHEET 4 OF 6



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

MATERIAL:

DWG NO.

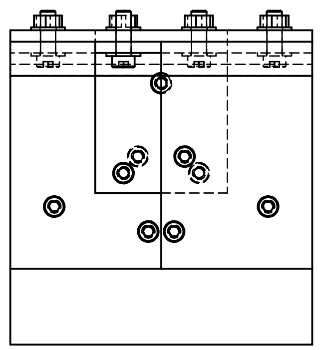
Angle bracket 180mm

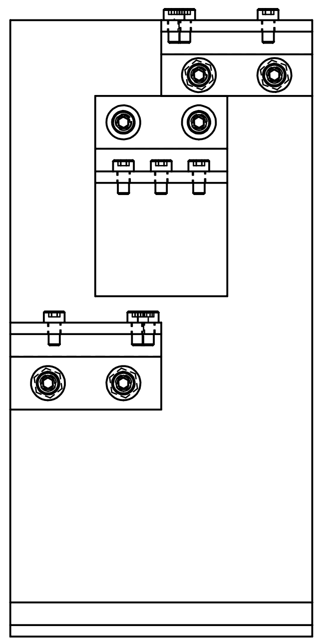
A3

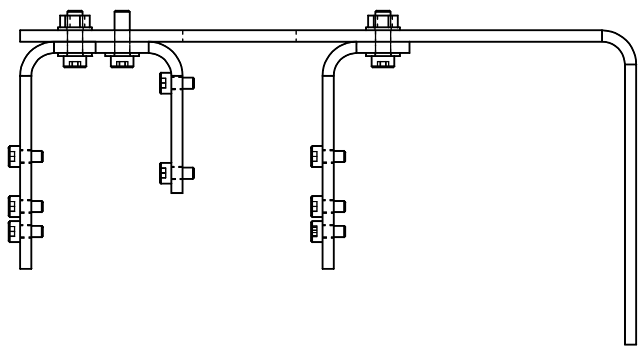
WEIGHT:

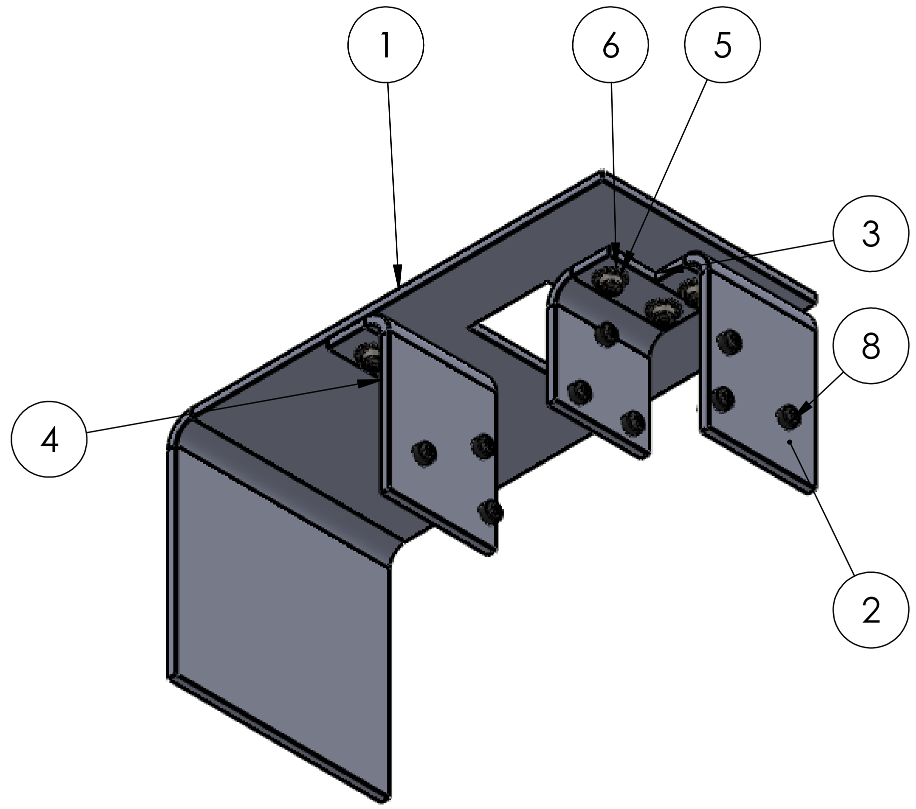
SCALE:1:1

SHEET 5 OF 6









ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	Producable camera stand		1
2	Upper camera stand		1
3	Middle camera stand		1
4	Lower camera stand		1
5	Bolt M4-12mm	SCREW DIN912-M3*12-A4-BLACKENING KVS-50	6
6	Washer M4		10
7	Nut M4		4
8	Bolt M3-6mm	SCREW DIN912-M3*12-A4-BLACKENING KVS-50	9

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

MATERIAL:

DWG NO.

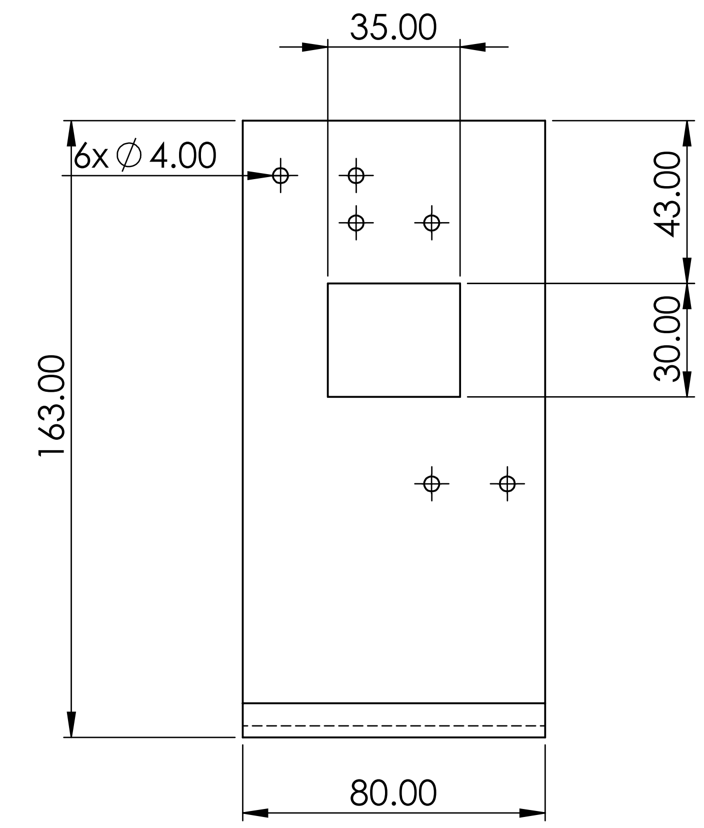
Producable camera stand

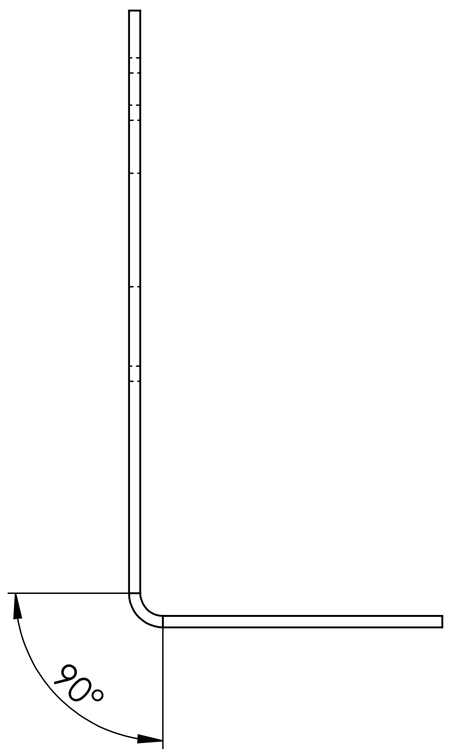
A3

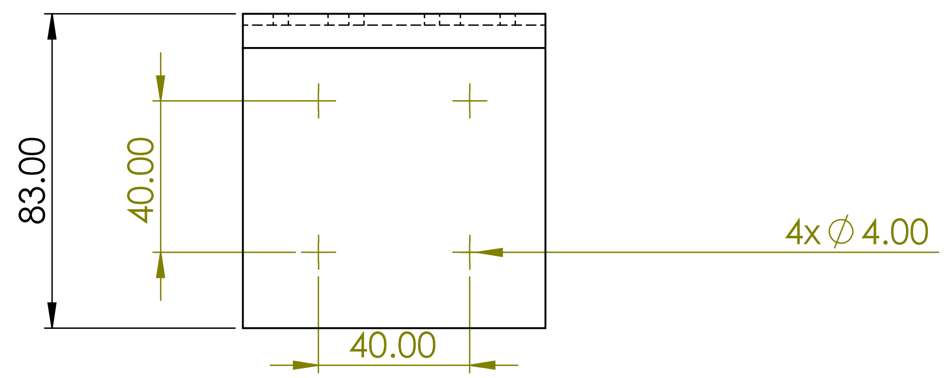
WEIGHT:

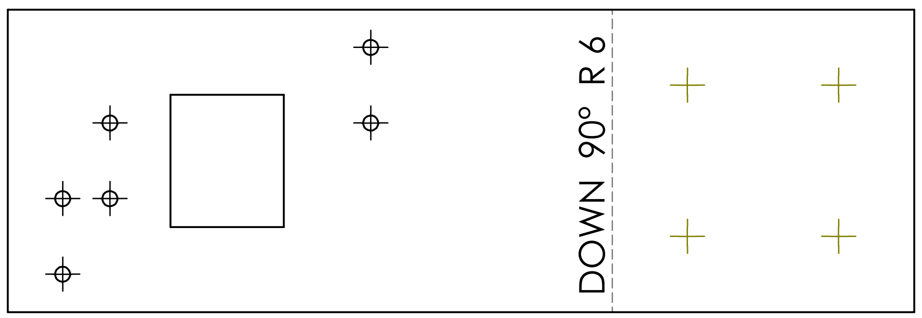
SCALE:1:2

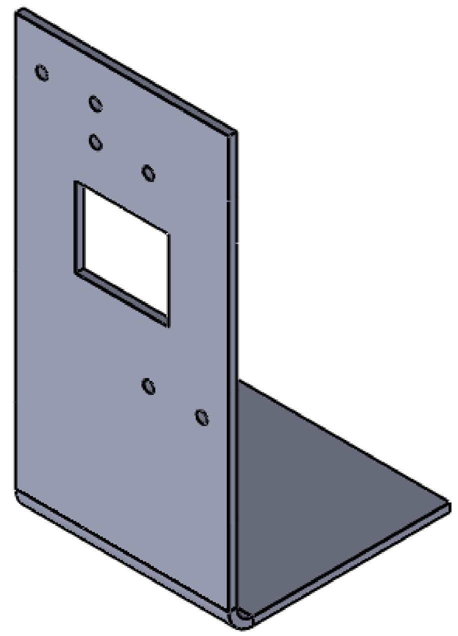
SHEET 1 OF 4











UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

MATERIAL:

DWG NO.

Producable camera stand

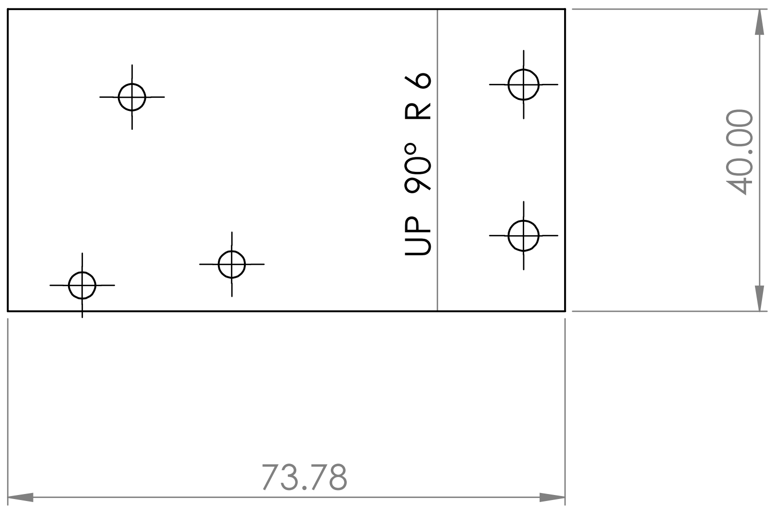
A3

WEIGHT:

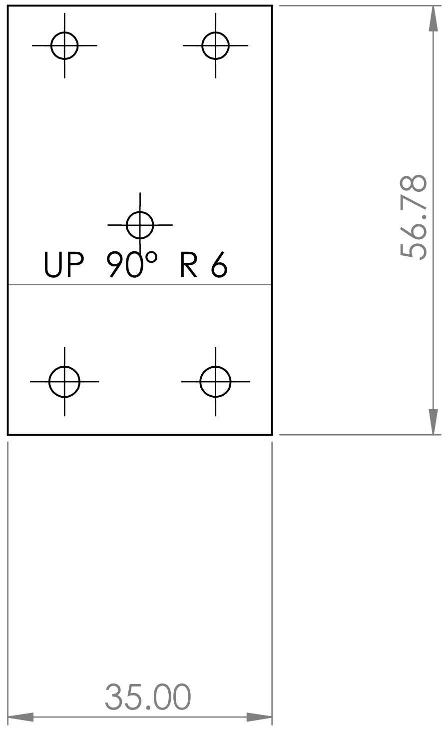
SCALE:1:2

SHEET 2 OF 4

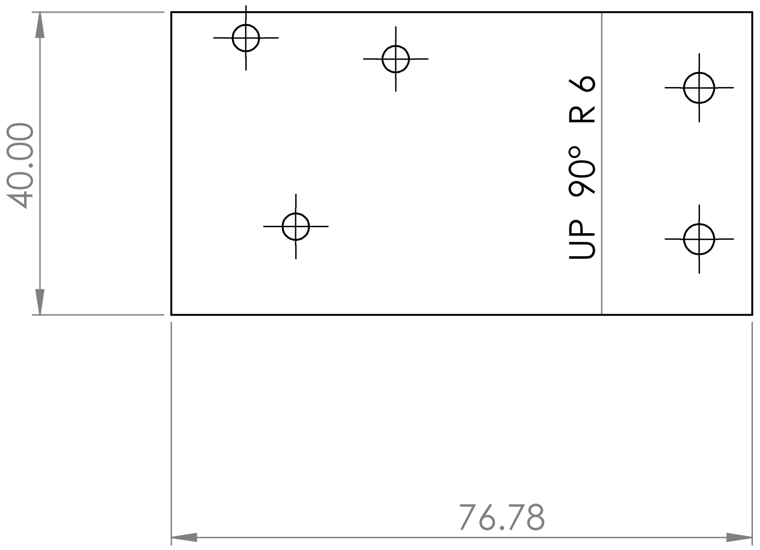
Upper camera stand



Middle camera stand



Lower camera stand



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

MATERIAL:

DWG NO.

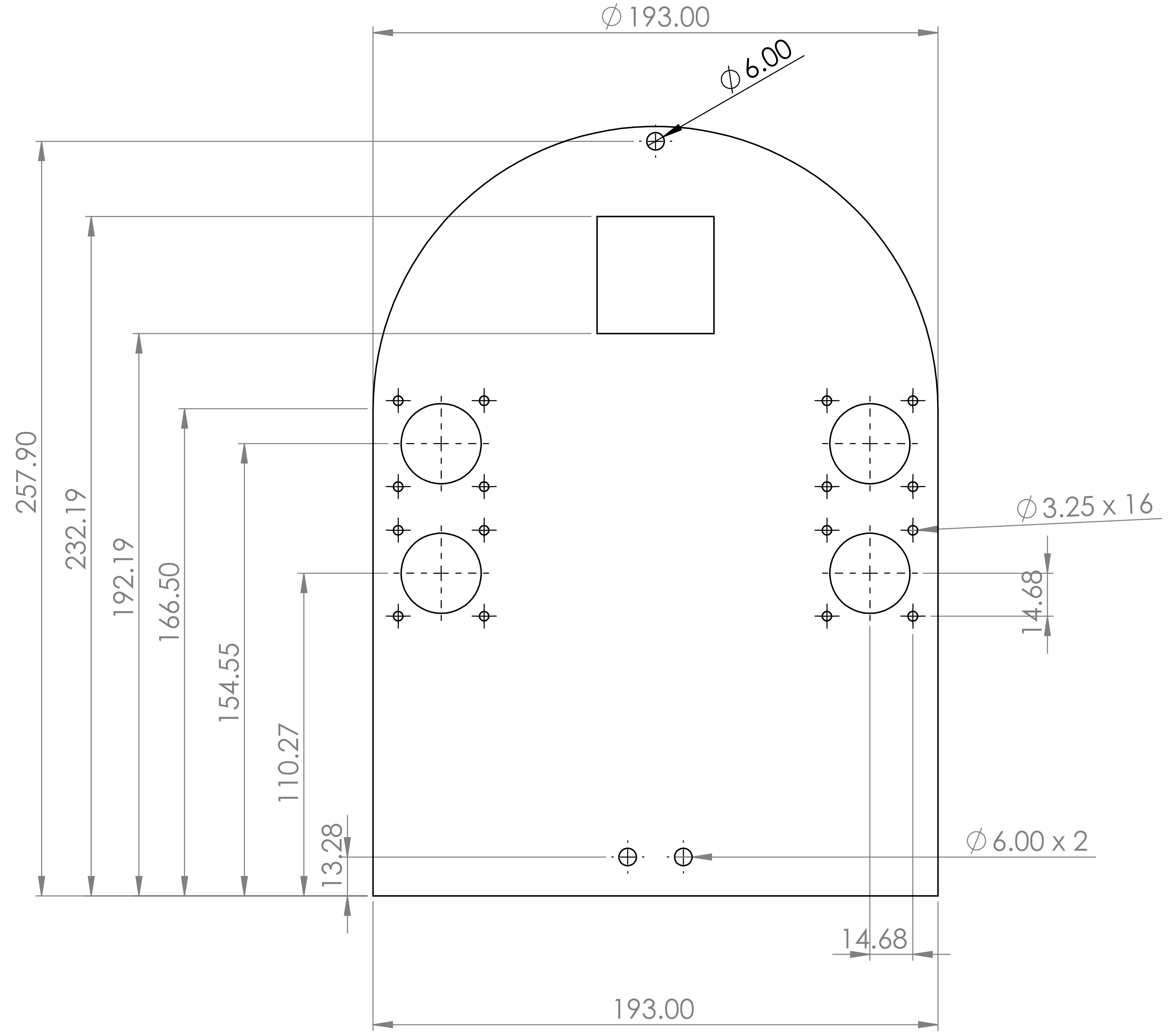
Producable camera stand

A3

WEIGHT:

SCALE:1:1

SHEET 3 OF 4



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: None
TOLERANCES: None
LINEAR:
ANGULAR:

FINISH:

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

MATERIAL: Aluminium

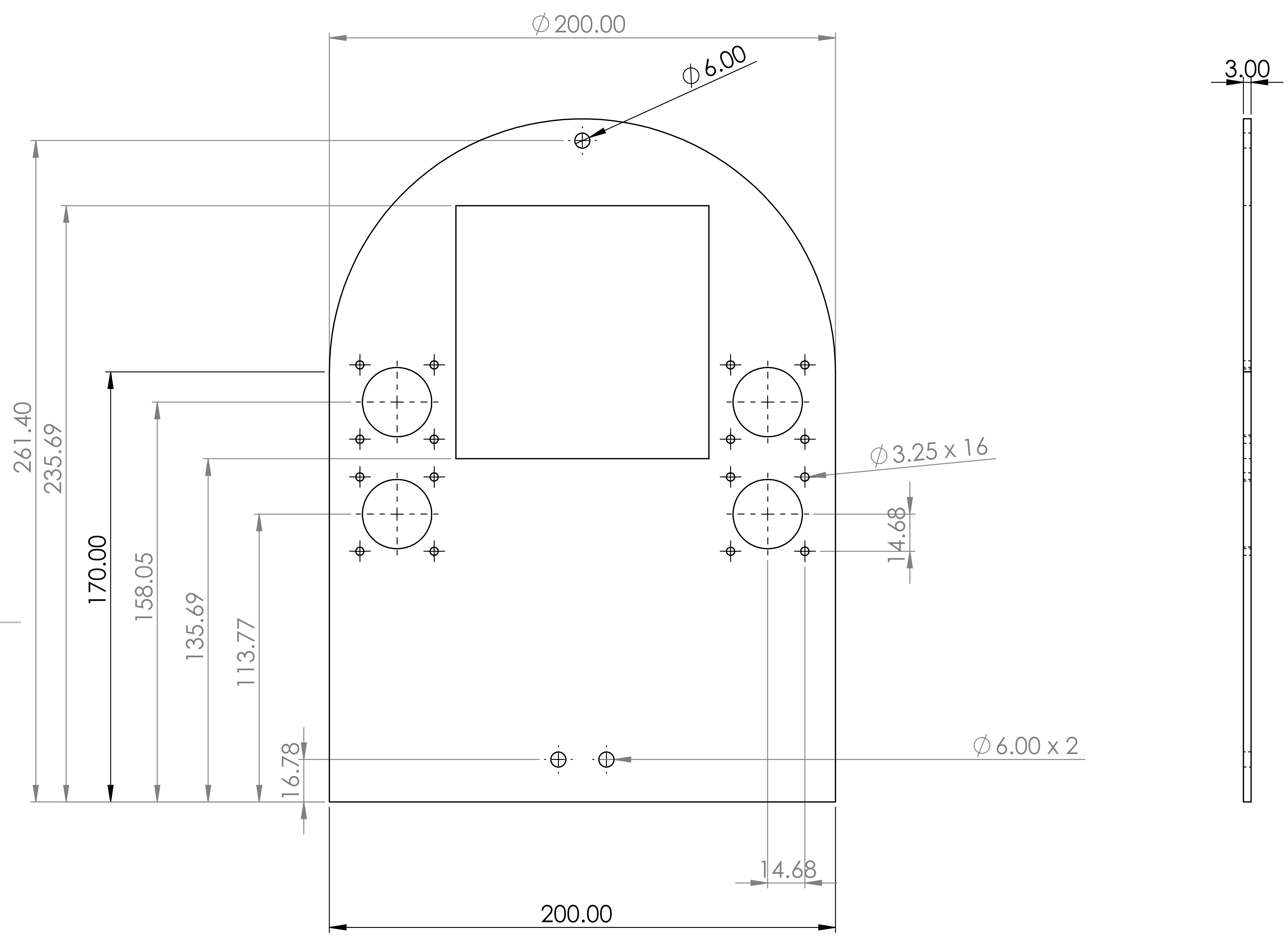
DWG NO. Bottom plate top

A3

WEIGHT:

SCALE:1:2

SHEET 1 OF 1



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: None
TOLERANCES: None
LINEAR:
ANGULAR:

FINISH:

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

MATERIAL: Aluminium

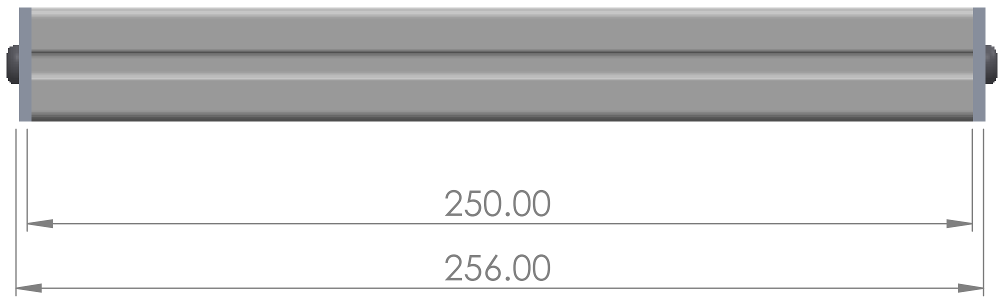
DWG NO. Bottom plate bottom

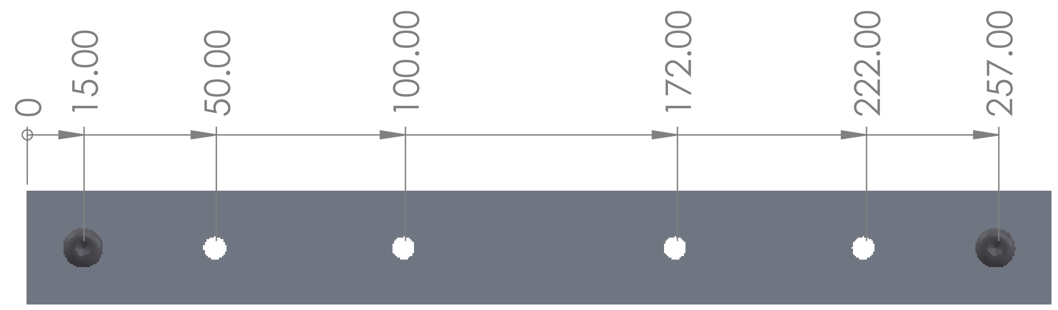
A3

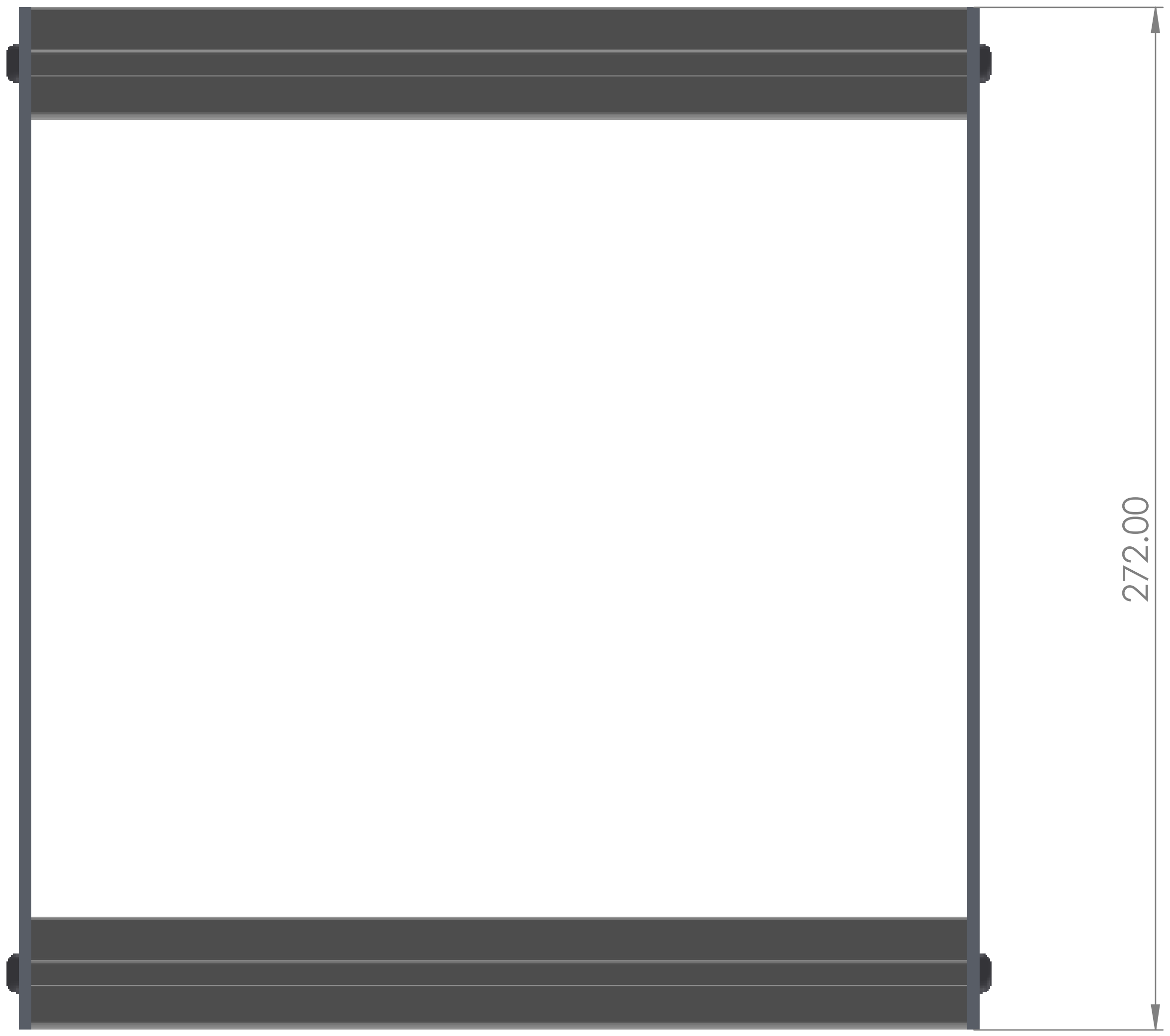
WEIGHT:

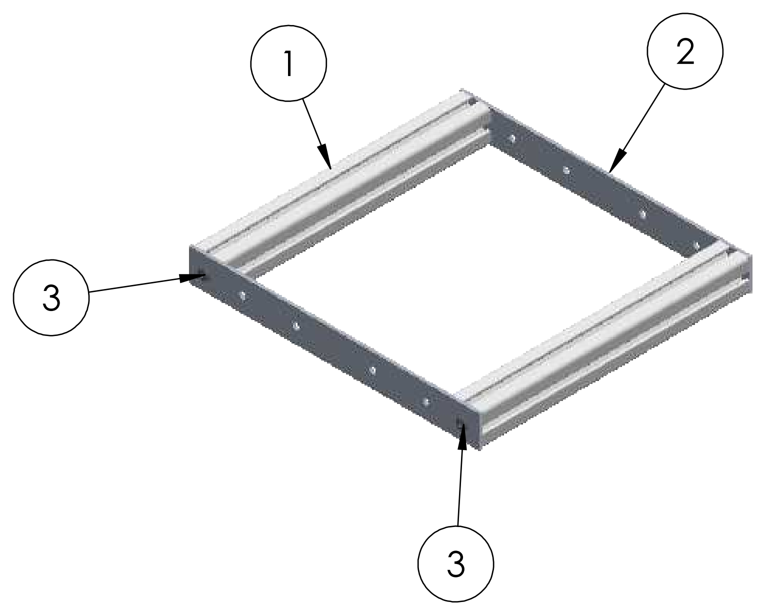
SCALE:1:2

SHEET 1 OF 1









ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	30x30 alu profile 250mm		2
2	Fastener for front box mounting		2
3	Bolt m6x20		4

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

MATERIAL:

DWG NO.

Front box mounting for roof ra

A3

WEIGHT:

SCALE:1:2

SHEET 1 OF 1

F.2 Rear Camera Box

In the following section are the blueprints for the design of the rear camera box.

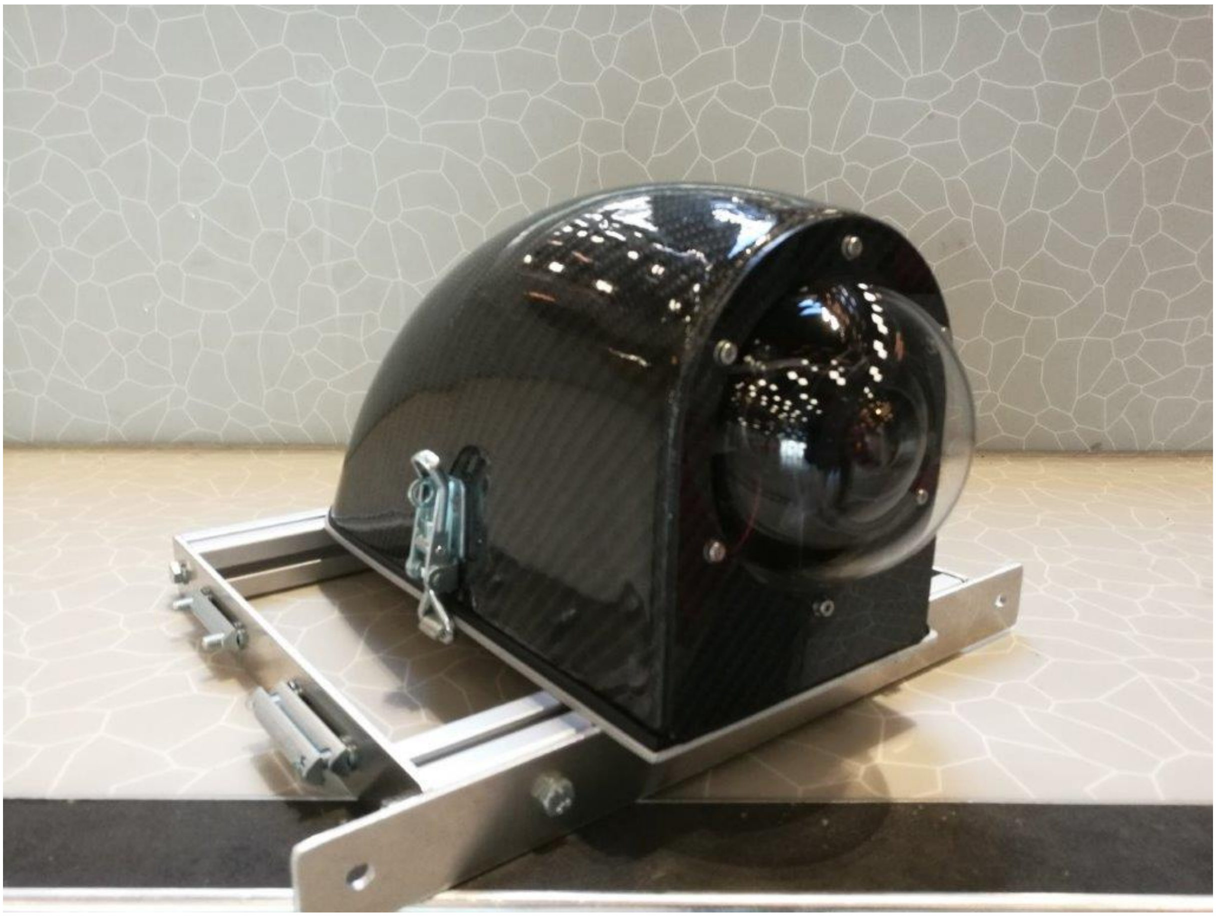
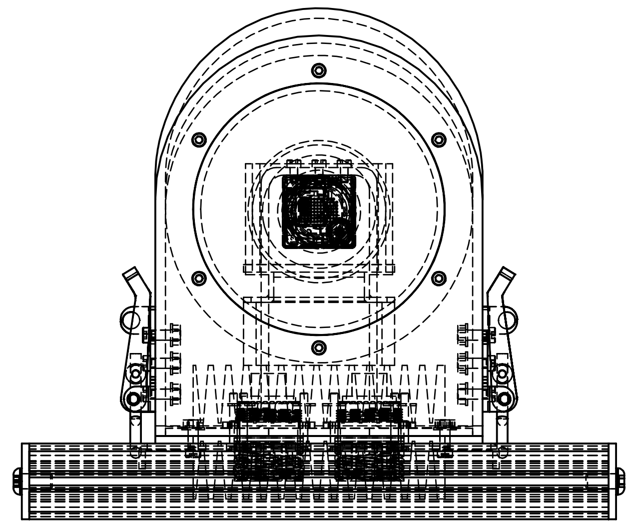
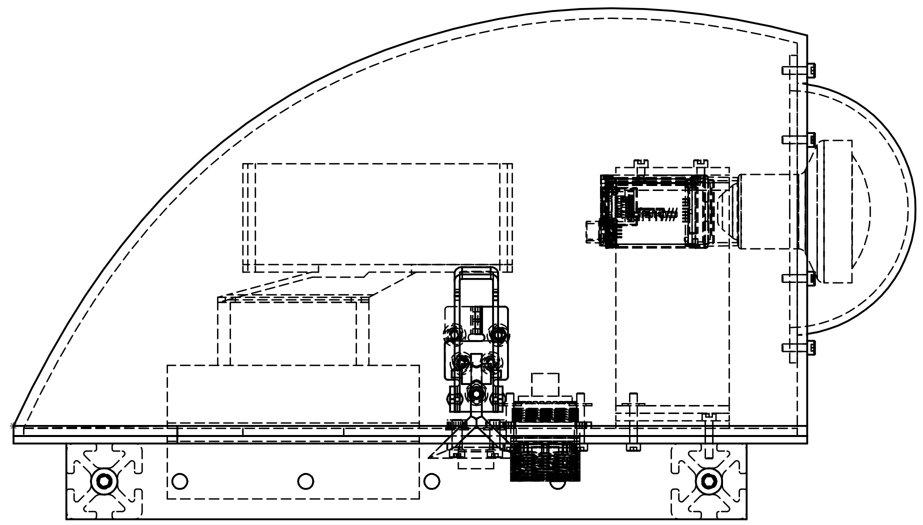
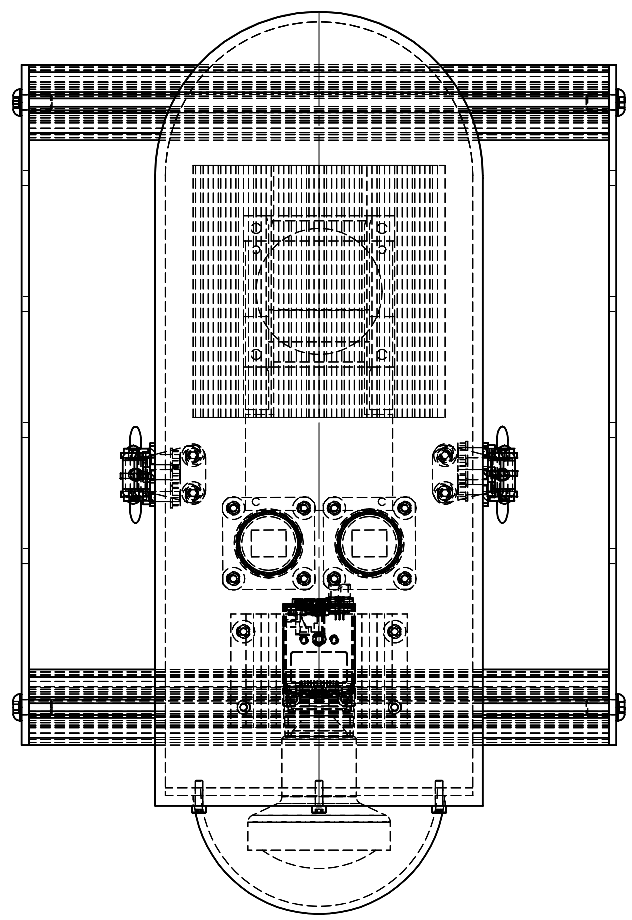
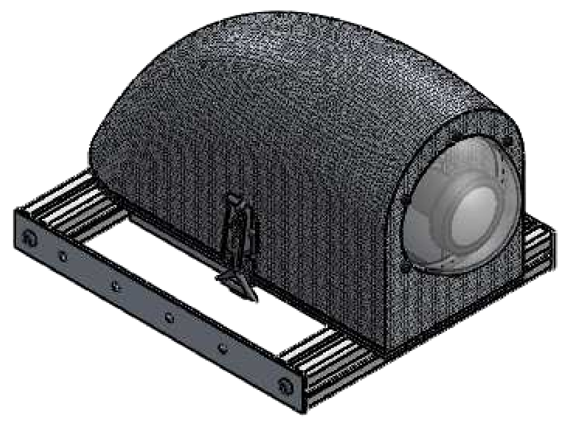


Figure F.2: Produced rear camera box for the Argos system







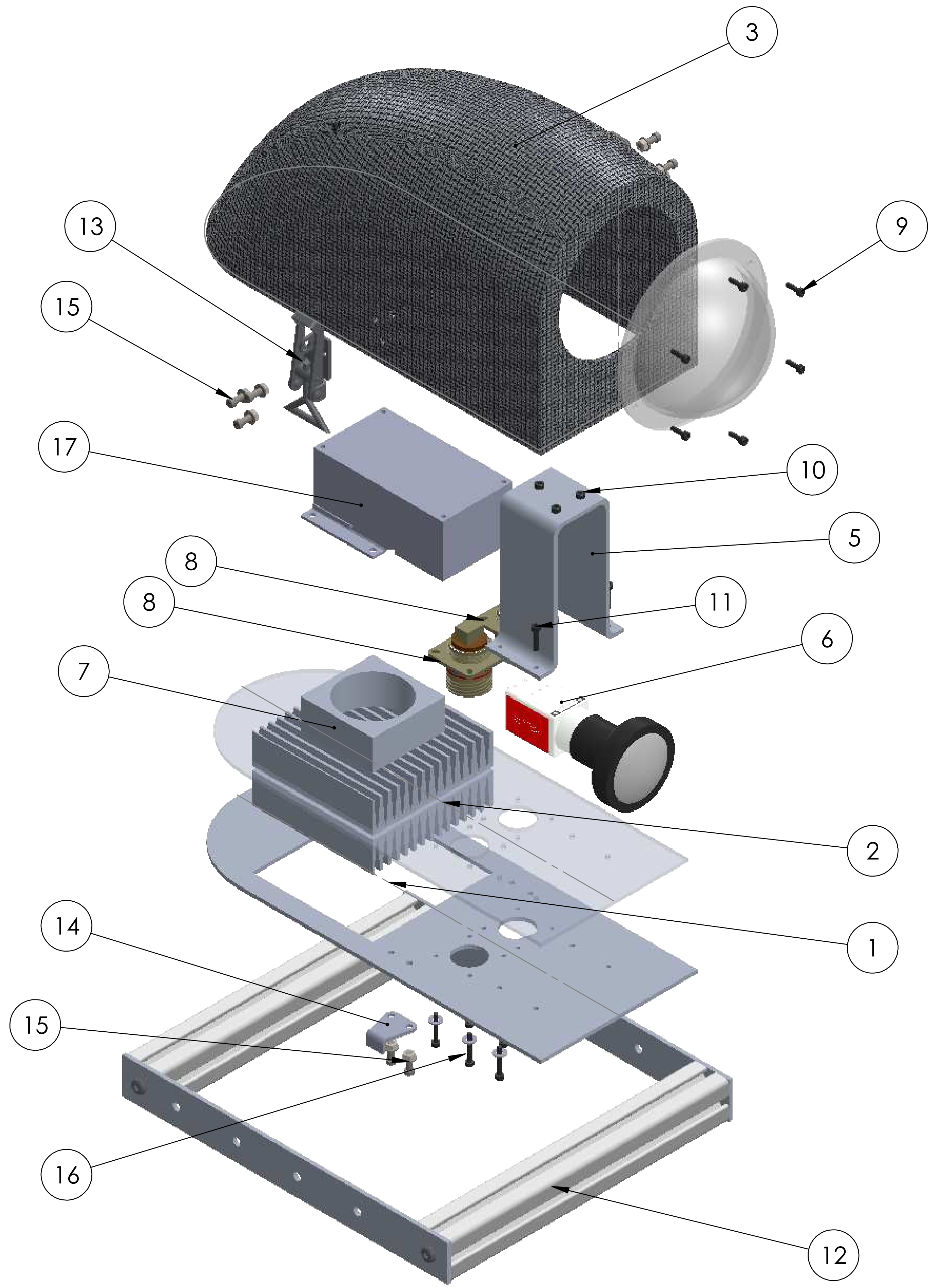


UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION

	NAME	SIGNATURE	DATE	TITLE:
DRAWN				
CHK'D				
APPV'D				
MFG				
Q.A				MATERIAL:

DWG NO. Argos back camera box A3

WEIGHT: SCALE:1:10 SHEET 1 OF 1



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	bottom floor bot	EN AW 5052	1
2	bottom floor top	EN AW 5052	1
3	Top shell	Carbon Fiber Composite, Hexcel 200 g/m2 fiber, Ampreg 21 epoxy system	1
4	glass dome	Plexiglas XT 3mm	1
5	camera mount	EN AW 5052	1
6	Fish eye assembly		1
7	peltier		1
8	HRM20W19-11SN	CONNECTOR ETHERNET RJFTV21G FEMALE RJ45 OLIVE DRAB CADMIUM	2
9	Bolt M3-10mm	SCREW DIN912-M3*12-A4-BLACKENING KVS-50	6
10	Bolt M3-6mm	SCREW DIN912-M3*12-A4-BLACKENING KVS-50	3
11	Bolt M3-15mm	SCREW DIN912-M3*12-A4-BLACKENING KVS-50	2
12	Rack mounting 30x30	EN AW 5052	1
13	New eccentric lock	31-1464	2
14	Locking arm	31-1464	2
15	M4 bolt, washers, nuts	DIN 966	10
16	m3 bolt, washer nuts	DIN 912	10
17	Circuit for electronics	PLA	1

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

MATERIAL:

DWG NO.

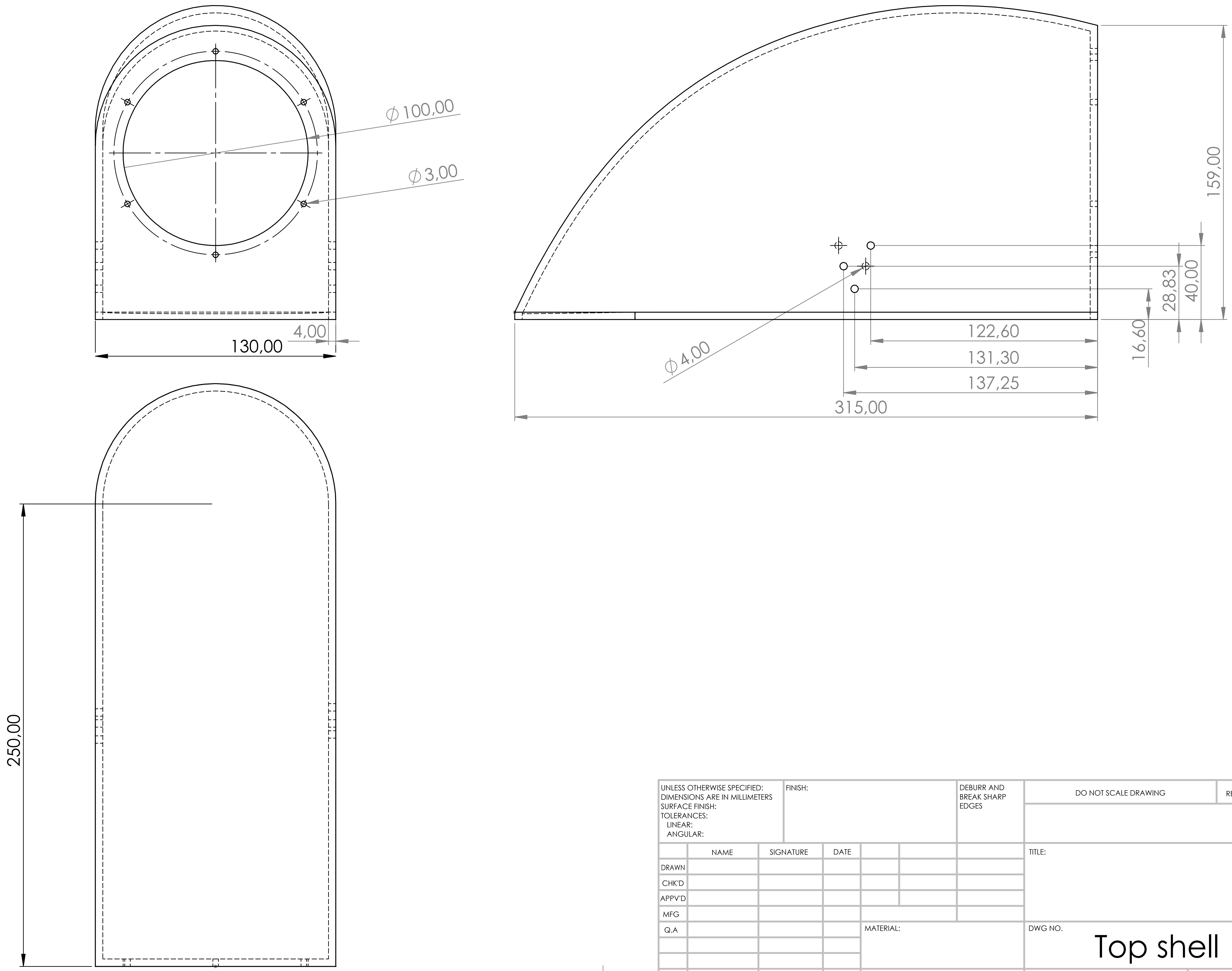
Argos back camera box exploc

A3

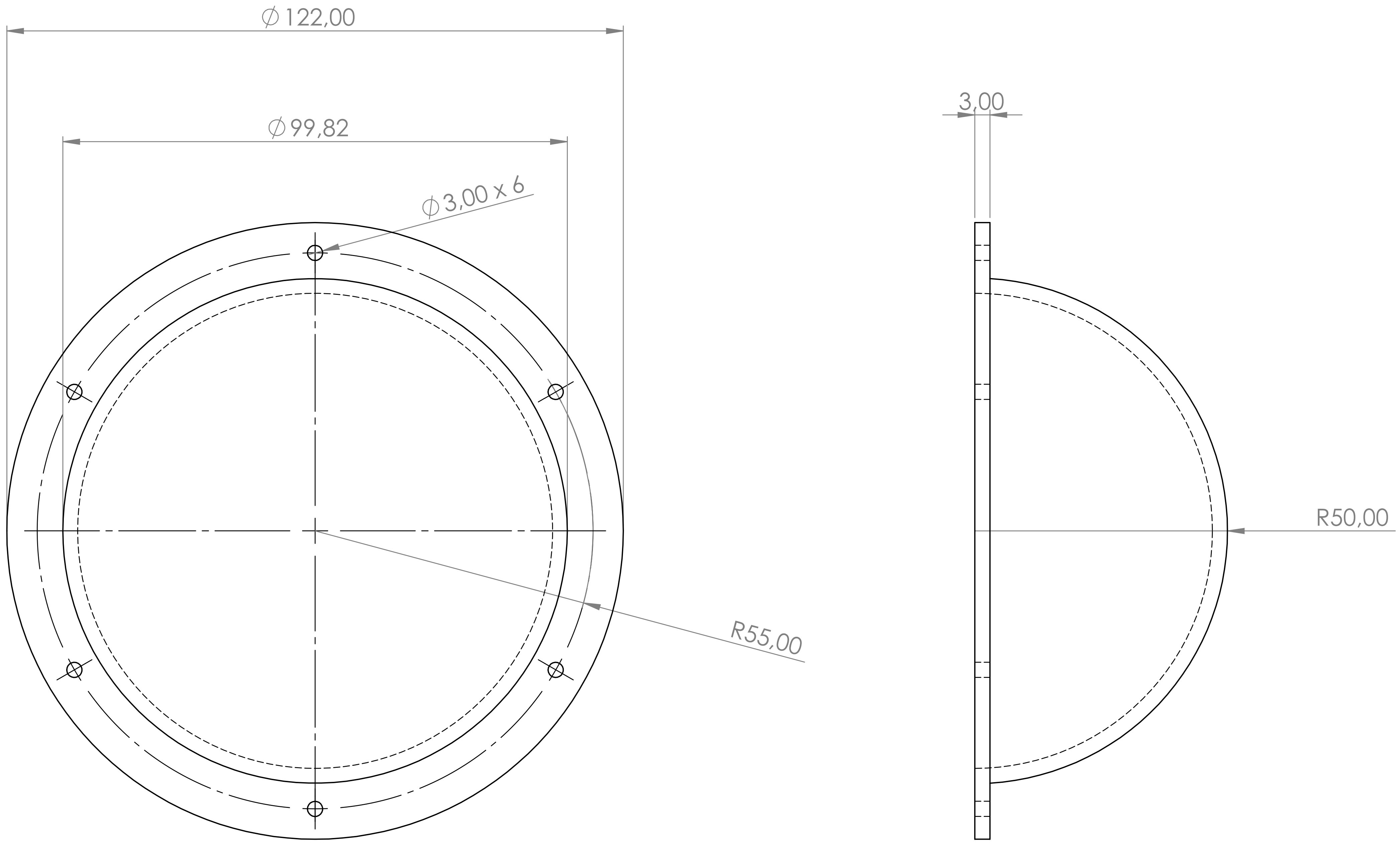
WEIGHT:

SCALE:1:10

SHEET 1 OF 1



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
	NAME	SIGNATURE	DATE	TITLE:
DRAWN				
CHK'D				
APPV'D				
MFG				
Q.A			MATERIAL:	DWG NO. Top shell — A3
			WEIGHT:	SCALE:1:2 — SHEET 1 OF 1



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

MATERIAL:

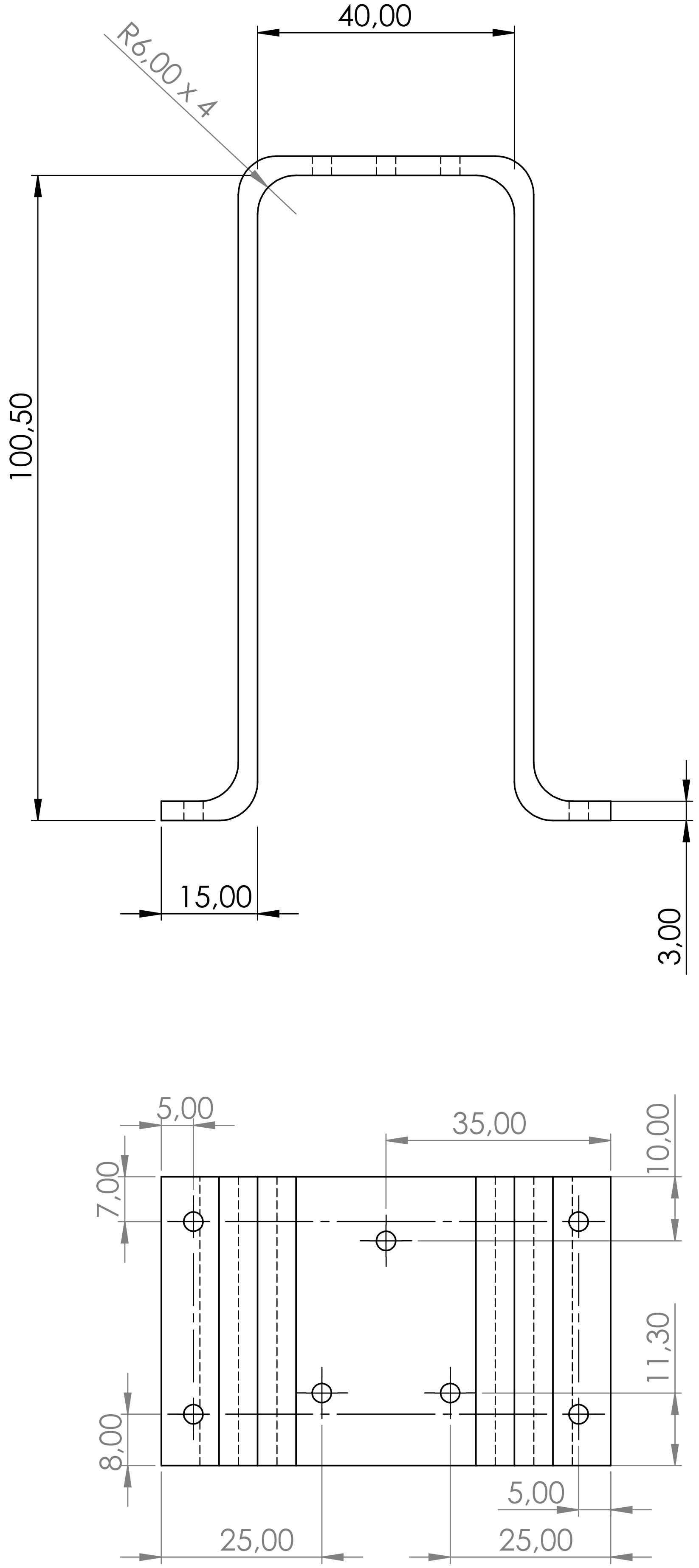
DWG NO. glass dome

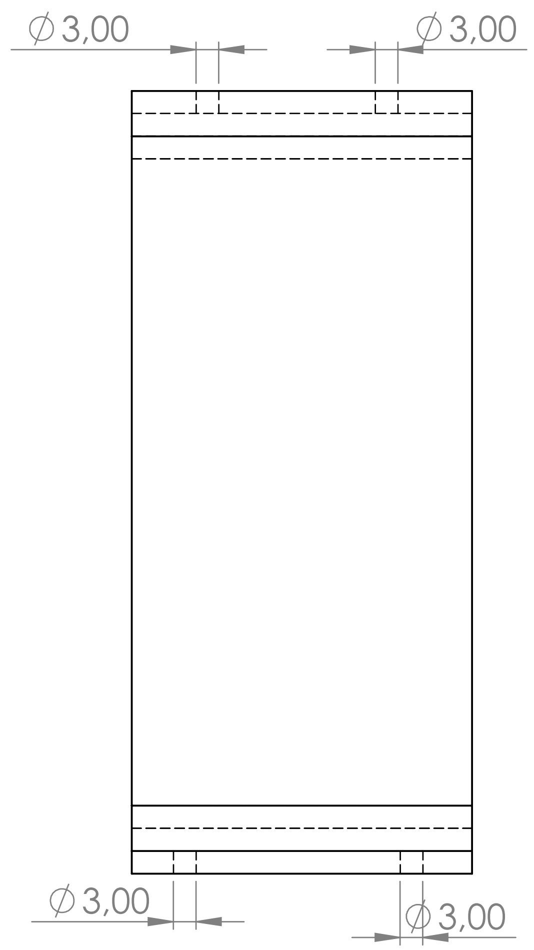
A3

WEIGHT:

SCALE:1:2

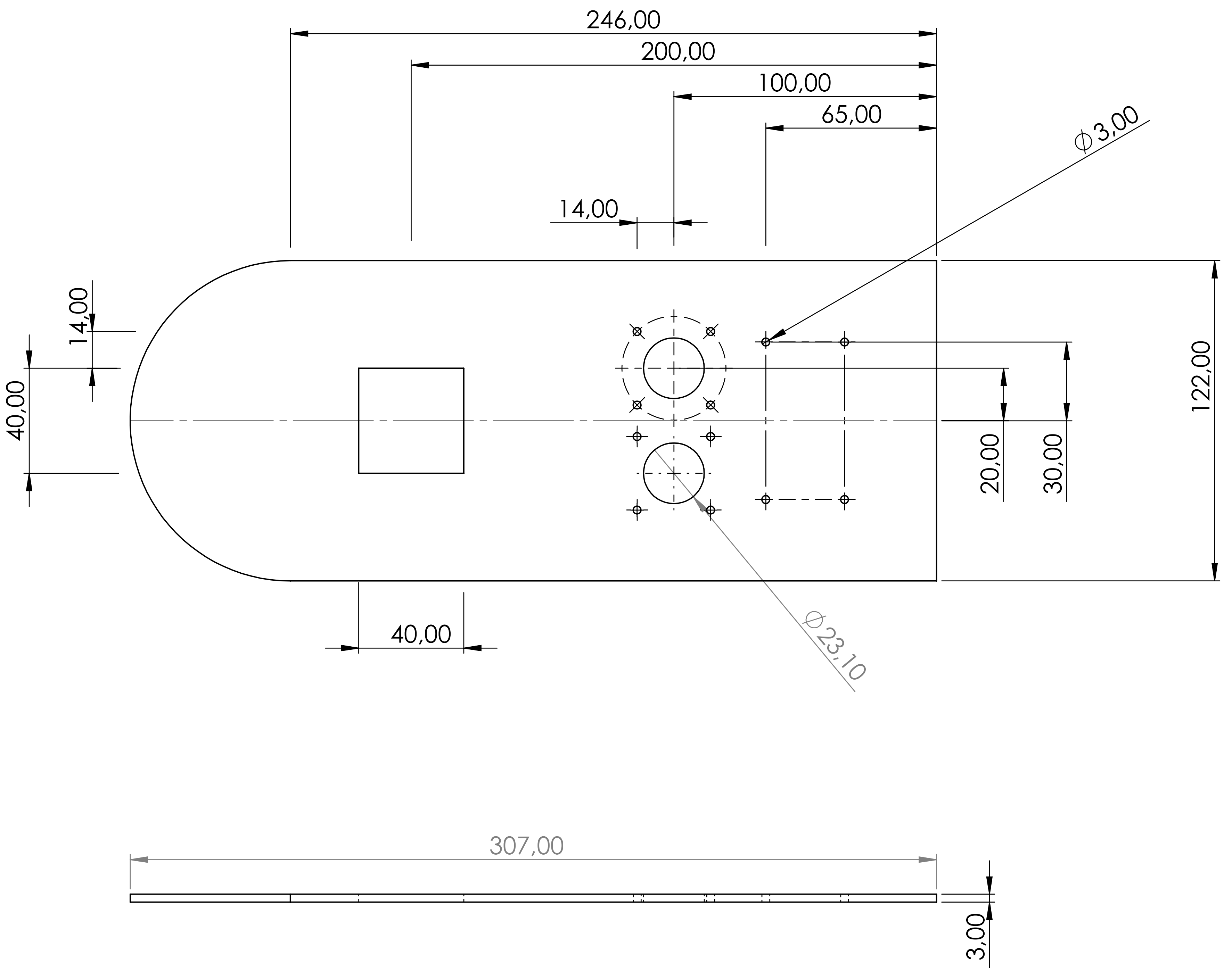
SHEET 1 OF 1



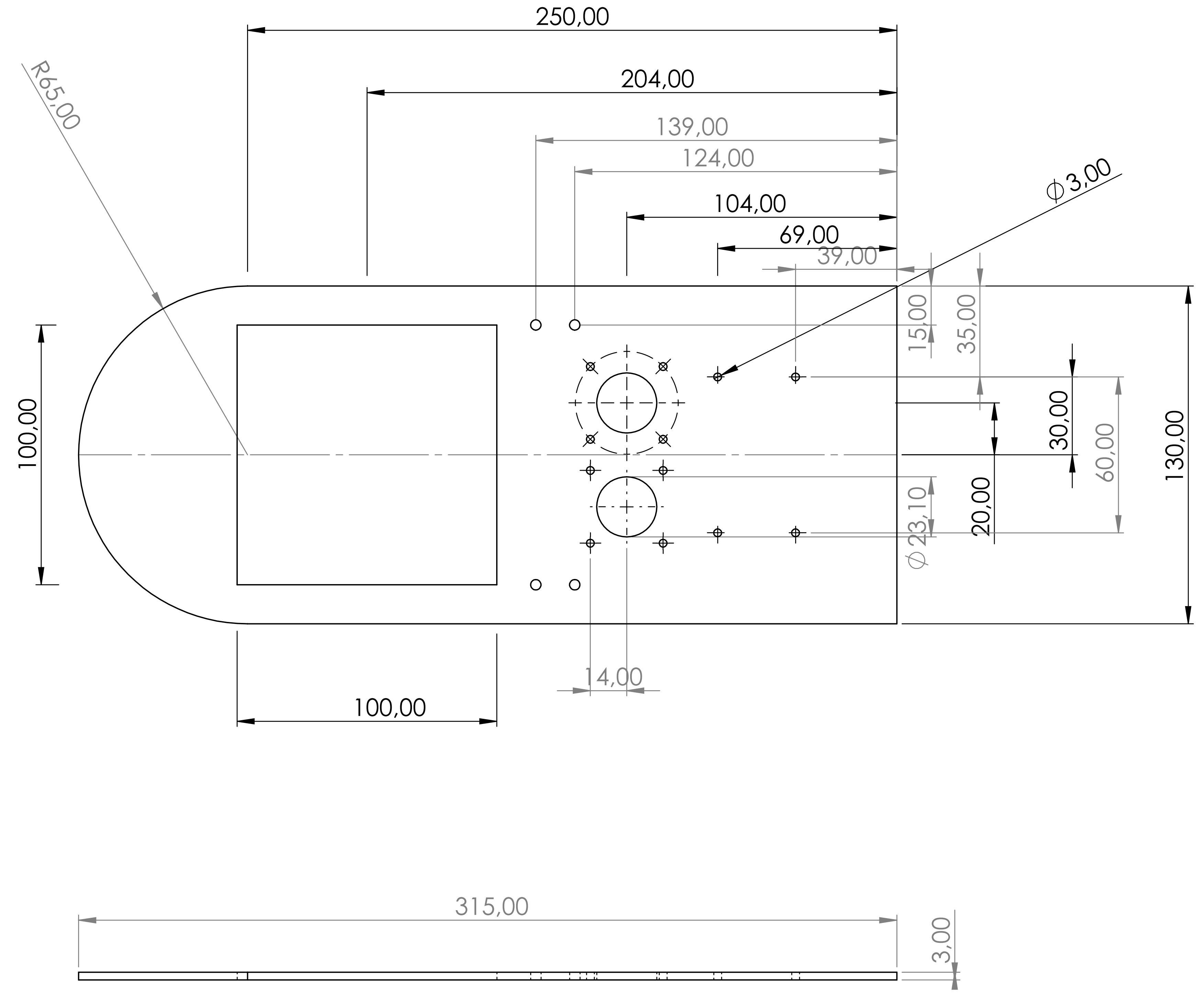


UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION

	NAME	SIGNATURE	DATE			TITLE:
DRAWN						
CHK'D						
APPV'D						
MFG						
Q.A				MATERIAL: Aluminium		DWG NO. camera mount — A3
				WEIGHT:		SCALE:1:1 — SHEET 1 OF 1

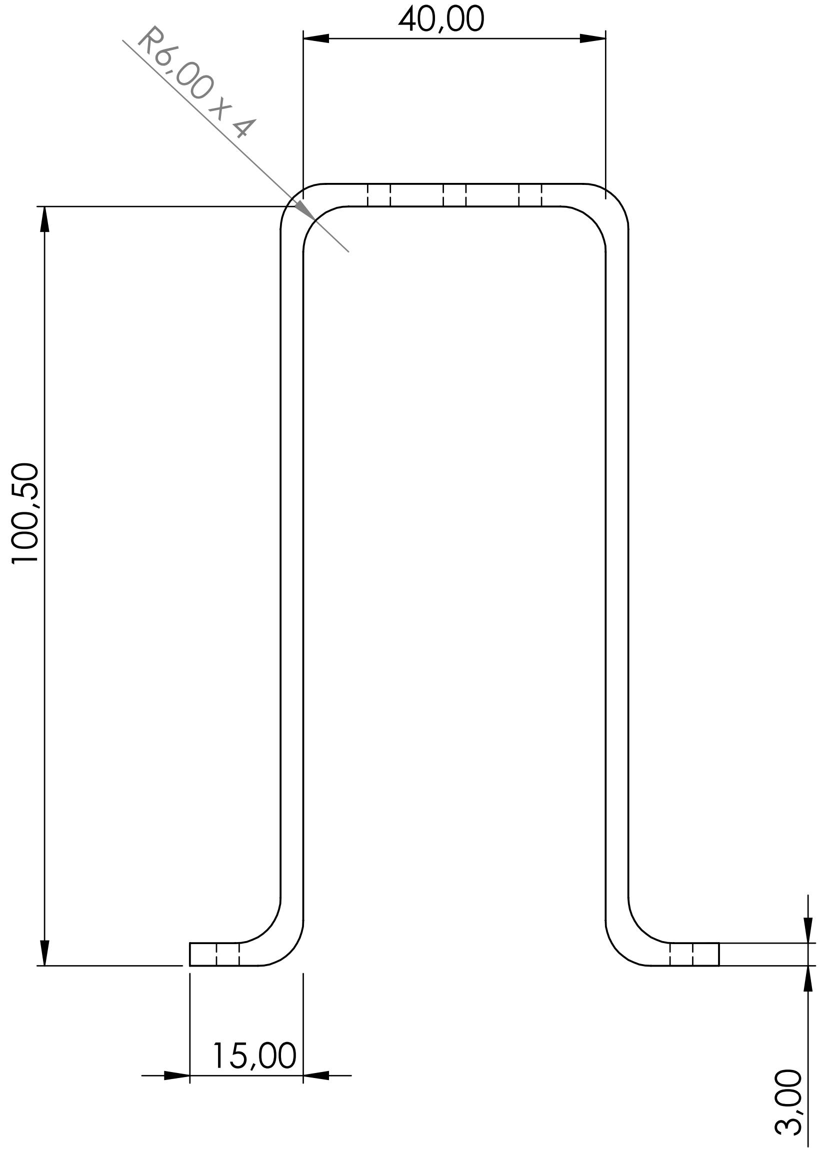


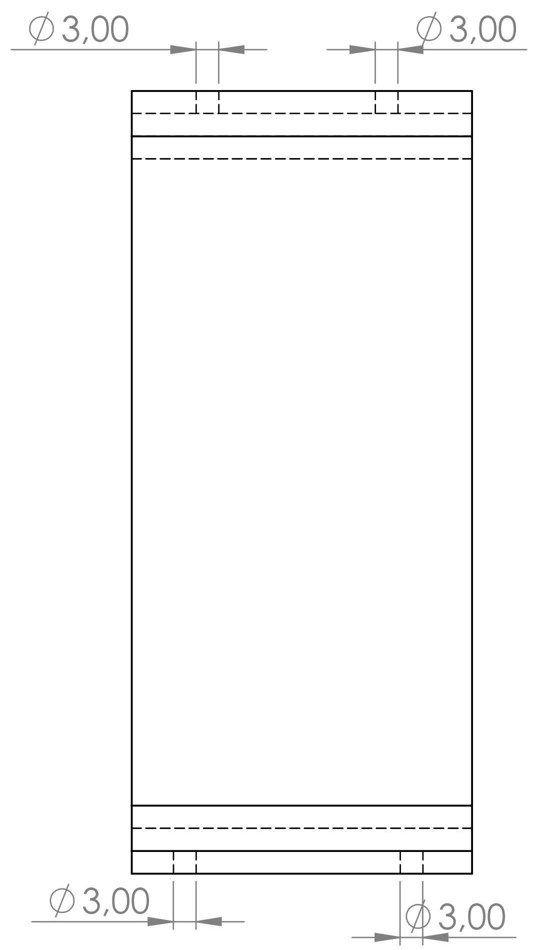
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:		DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION	
	NAME	SIGNATURE	DATE		TITLE:	
DRAWN						
CHK'D						
APPV'D						
MFG						
Q.A				MATERIAL: Aluminium	DWG NO. bottom floor top	A3
				WEIGHT:	SCALE:1:2	SHEET 1 OF 1

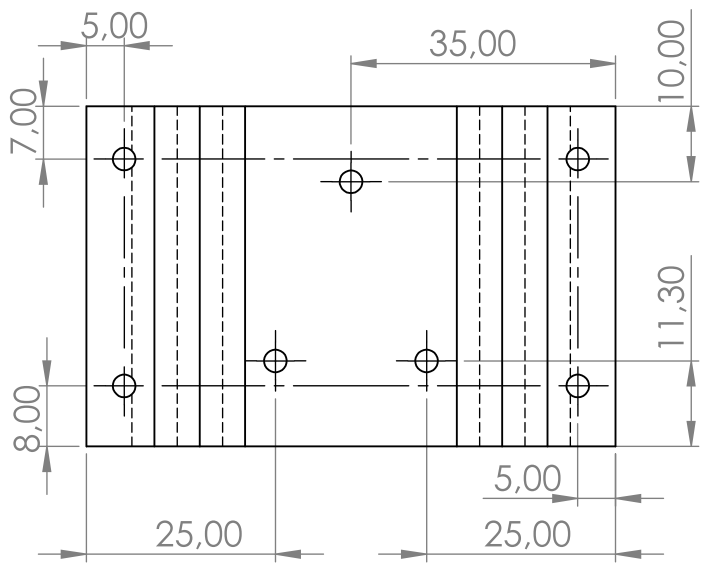


UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION

	NAME	SIGNATURE	DATE		TITLE:
DRAWN					
CHK'D					
APPV'D					
MFG					
Q.A				MATERIAL: Aluminium	DWG NO. bottom floor bot A3
				WEIGHT:	SCALE:1:2 SHEET 1 OF 1







UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

MATERIAL:
Aluminium

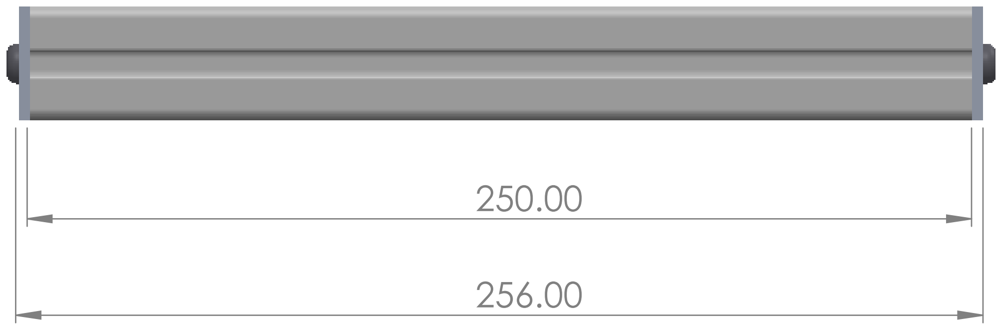
WEIGHT:

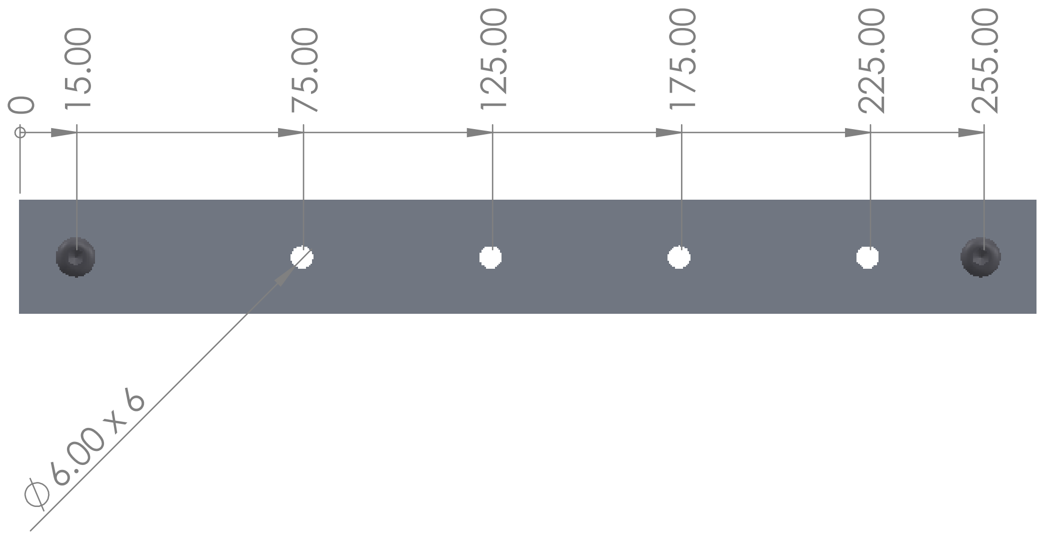
DWG NO.
camera mount

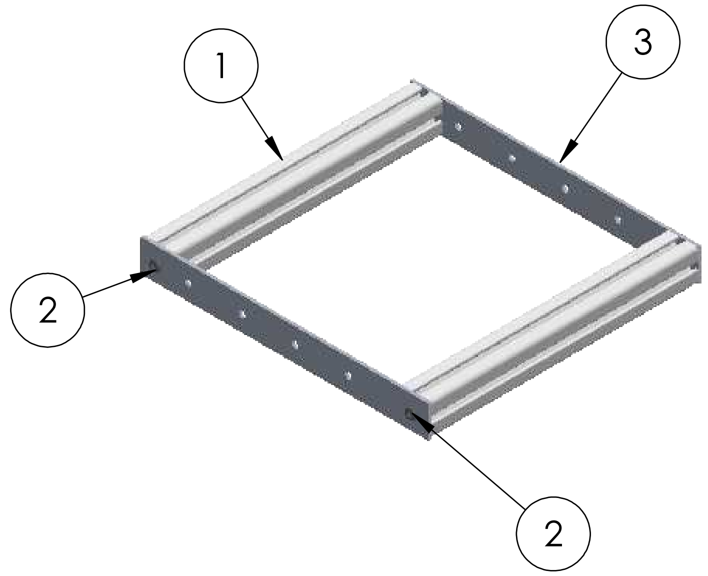
A3

SCALE:1:1

SHEET 1 OF 1







ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	30x30 alu profile 250mm	RollCo 30x30 Aluminium profile, P3010, 250mm	2
2	Bolt m6x20		4
3	sliderhold		2

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

MATERIAL:

DWG NO.

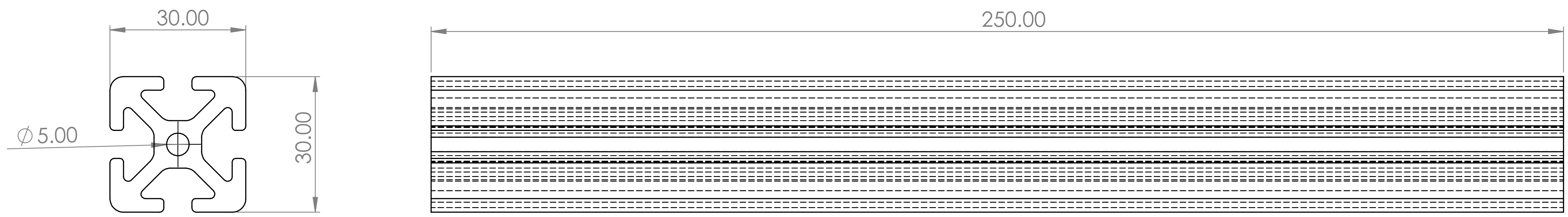
Rack mounting back box draw

A3

WEIGHT:

SCALE:1:2

SHEET 1 OF 1



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

MATERIAL: Aluminium

WEIGHT:

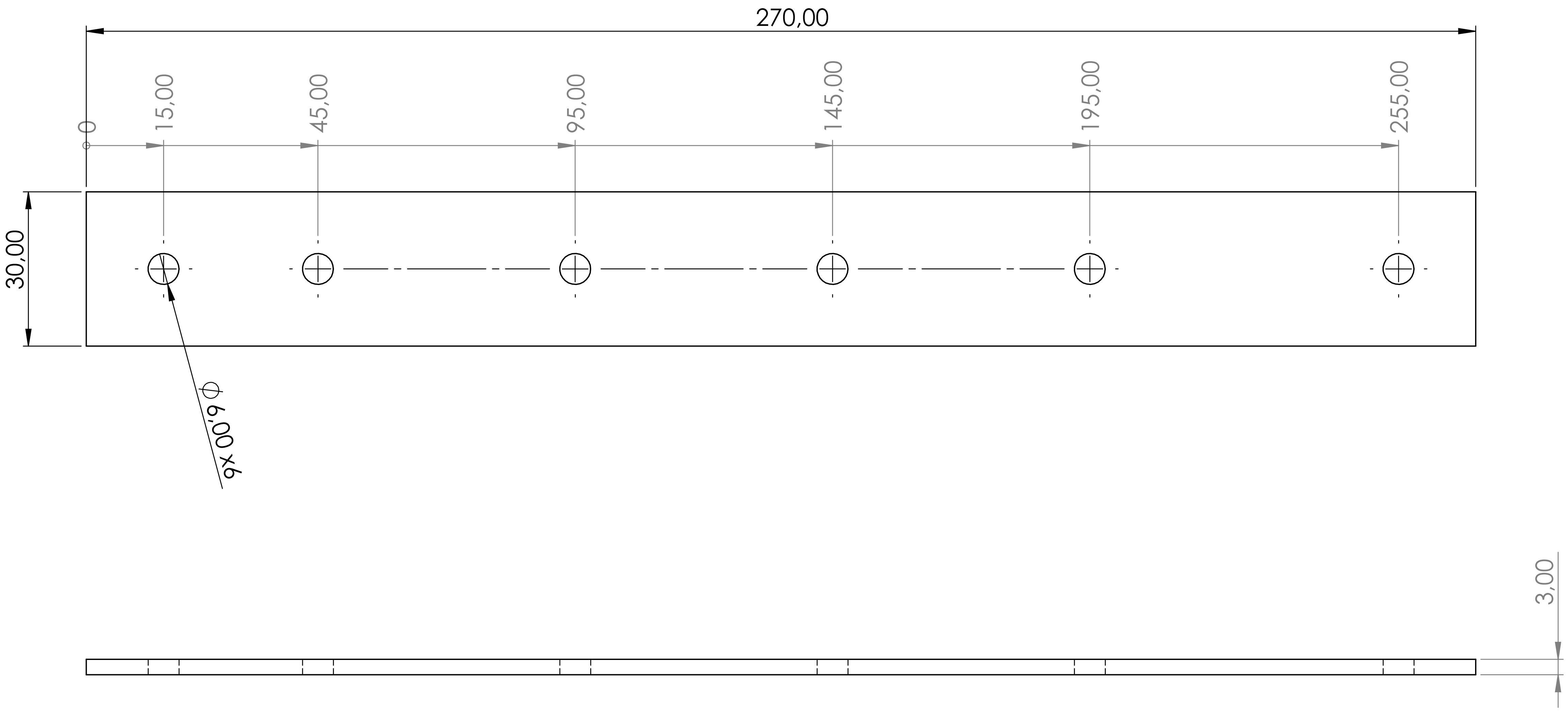
TITLE: RollCo 30x30 Aluminium profile, P3010, 250mm bottom box frame

DWG NO.

A3

SCALE:1:2

SHEET 1 OF 1



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:			DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
	NAME	SIGNATURE	DATE		TITLE:	
DRAWN						
CHK'D						
APPV'D						
MFG						
Q.A				MATERIAL:	DWG NO. sliderhold	A3
				WEIGHT:	SCALE:1:2	SHEET 1 OF 1

F.3 Mounting System

In the following section are the blueprints for the design of the roof mounting system.

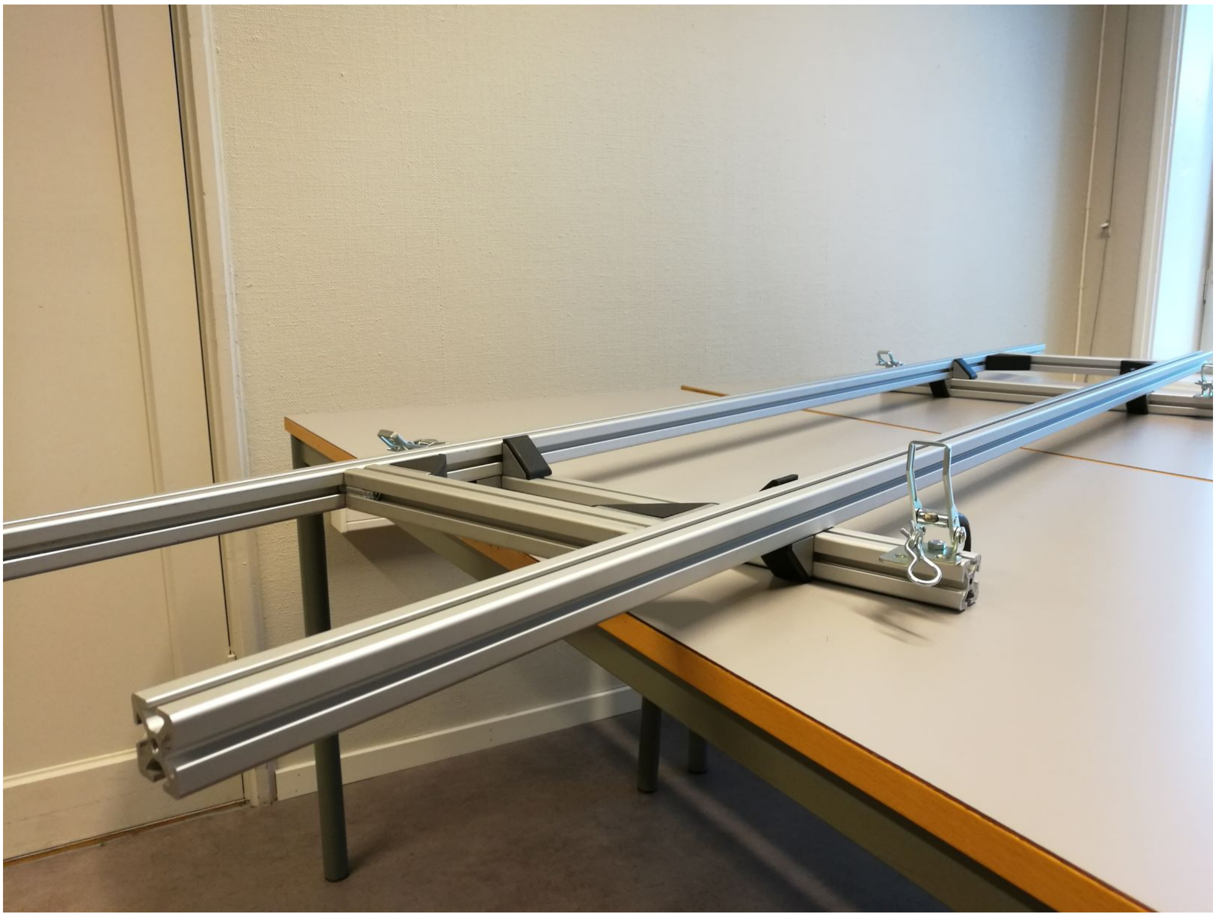
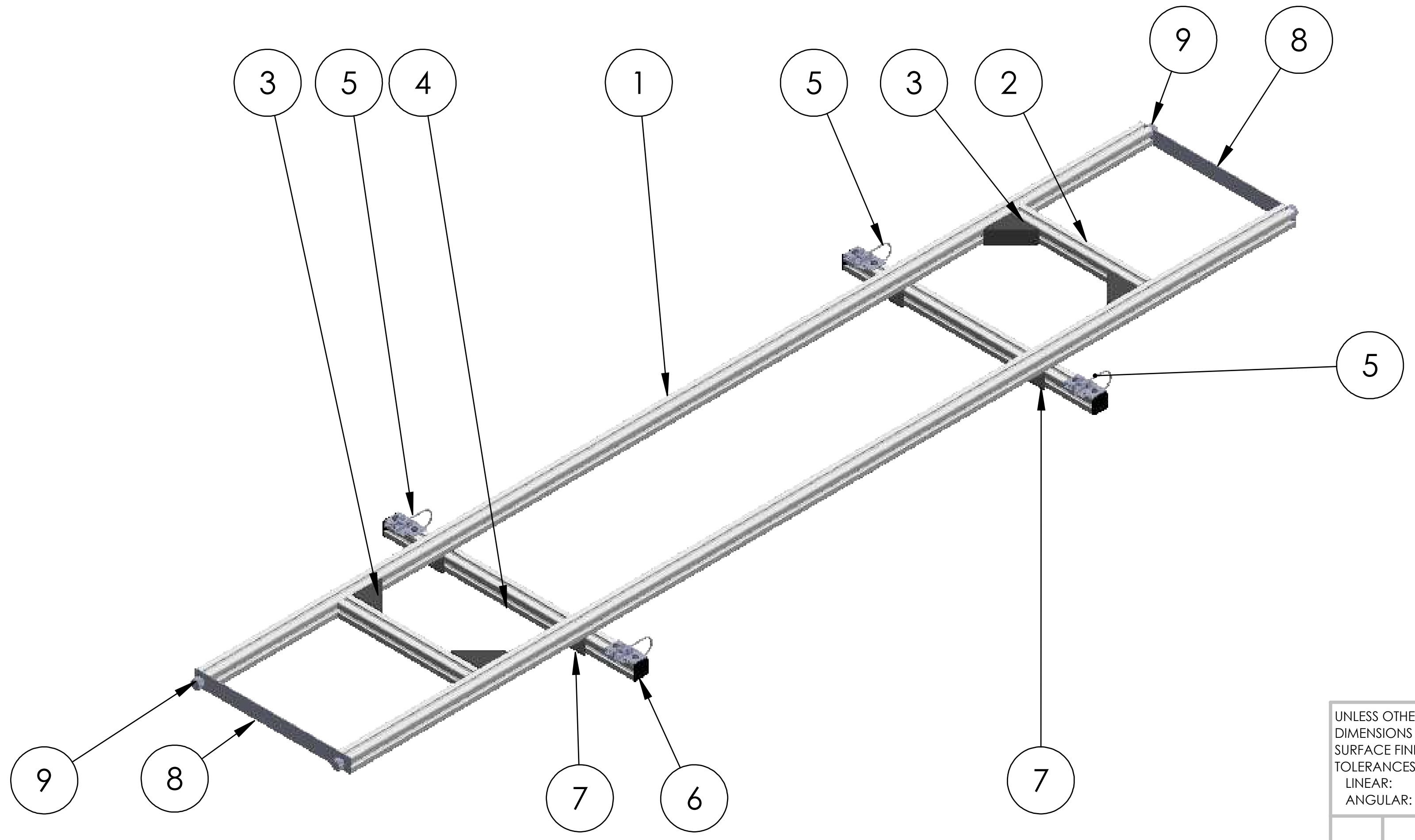


Figure F.3: Produced roof mounting system for the Argos system

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	30x30 alu profile 2000mm	RollCo 30x30 Aluminium Profile, PP3010, 2000mm	2
2	30x30 alu profile 270mm	RollCo 30x30 Aluminium Profile, P3010, 270mm	2
3	PZ3074-Rollco	60x60 fastening set	4
4	30x30 alu profile 527mm	RollCo 30x30 Aluminium Profile, P3010, 527mm	2
5	roof lock system	Eccentric lock with hook	4
6	PA3010-Rollco	Plastic caps	4
7	PZ3070-Rollco	30x30 fastening set	4
8	cameraholder		2
9	Fastening knob		4



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

MATERIAL:

DWG NO.

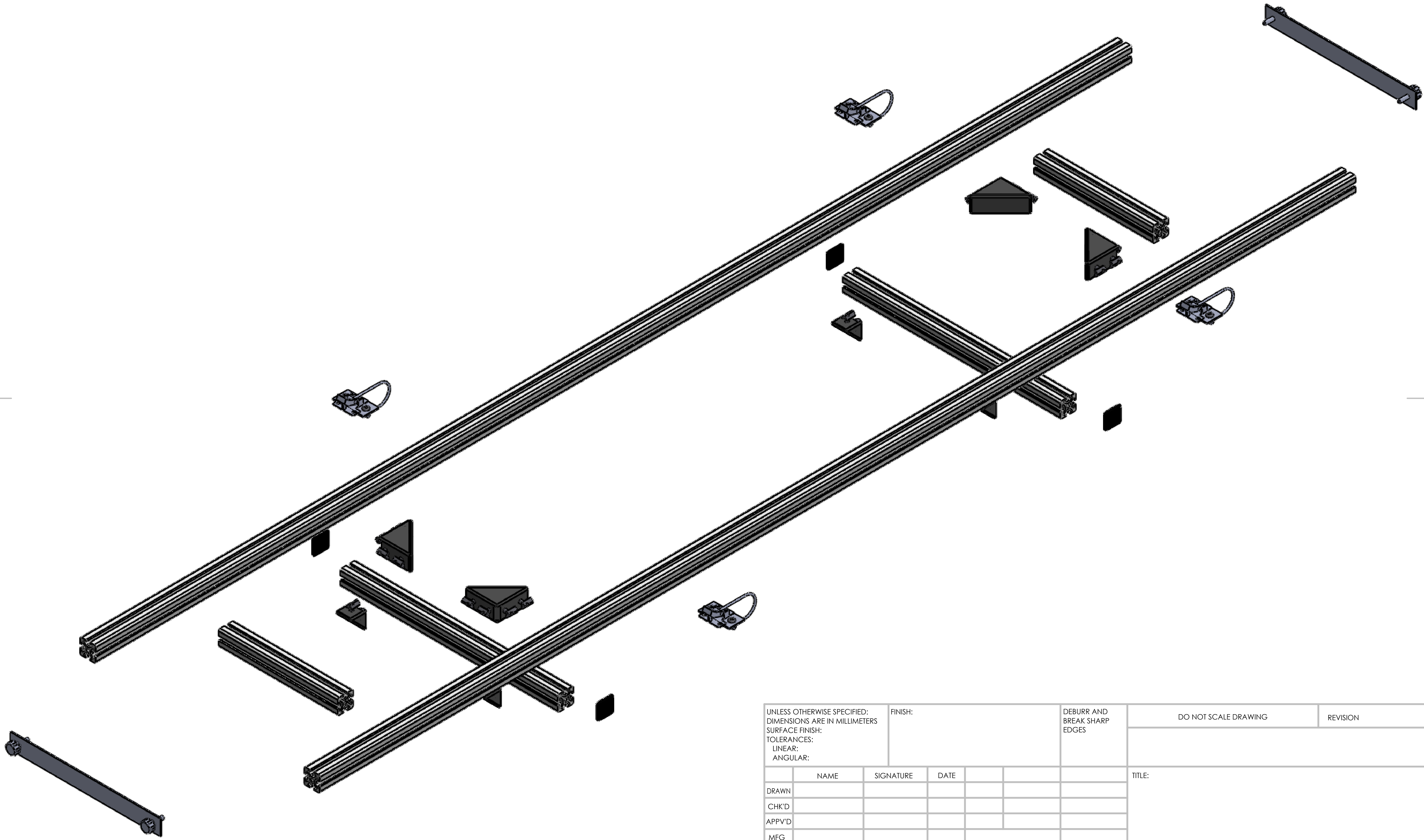
Argos roof rack bars

A3

WEIGHT:

SCALE:1:10

SHEET 1 OF 2



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:	DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

MATERIAL:

DWG NO.

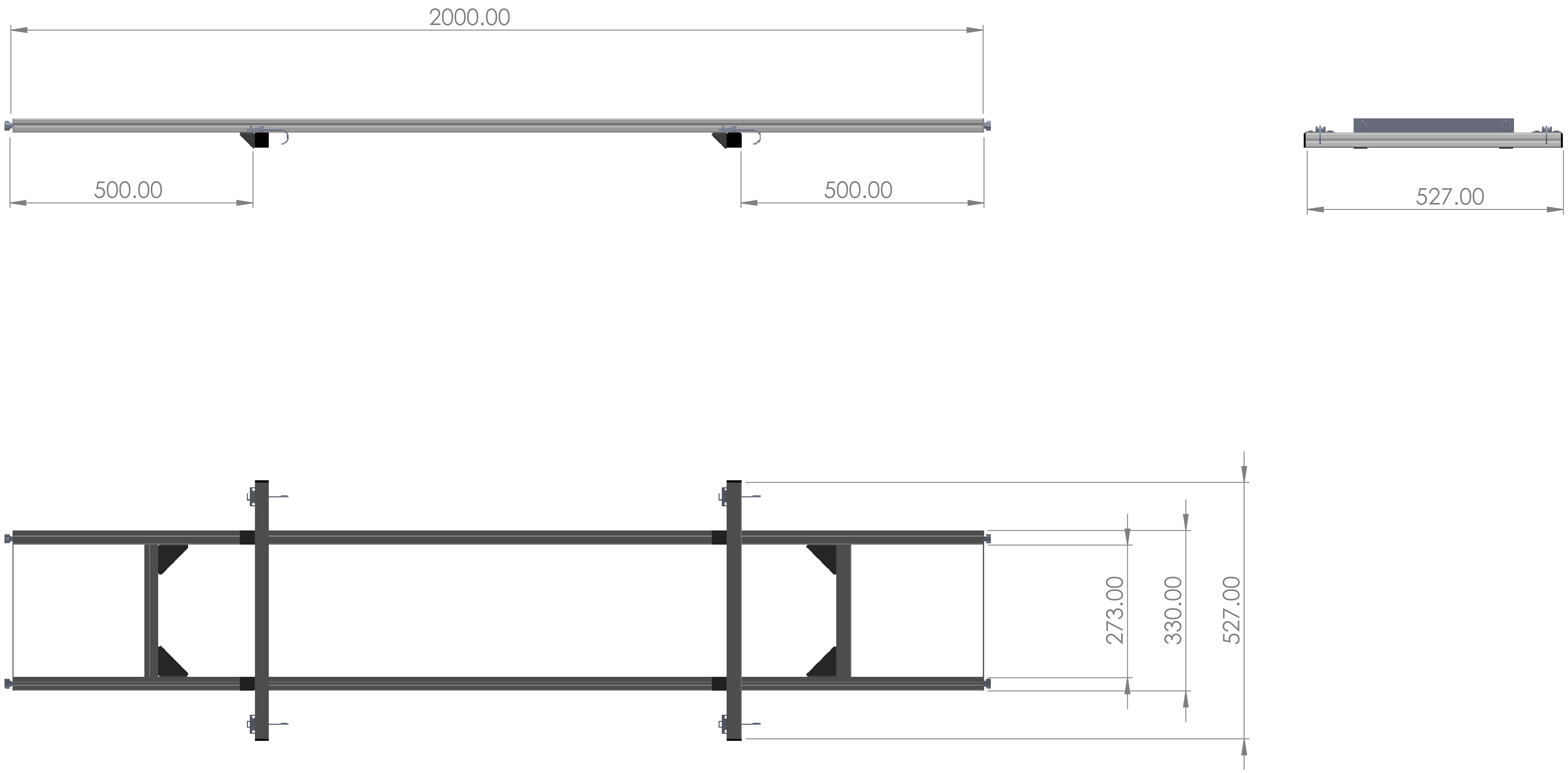
Roof rack exploded

A3

WEIGHT:

SCALE:1:20

SHEET 1 OF 1



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

MATERIAL:

DWG NO.

Roof Rack dimensions

A3

WEIGHT:

SCALE:1:10

SHEET 2 OF 2

G Circuit Schematics

In the following section are the circuit schematics for the hardware of the Argos System.

G.1 First PCB For Battery Control

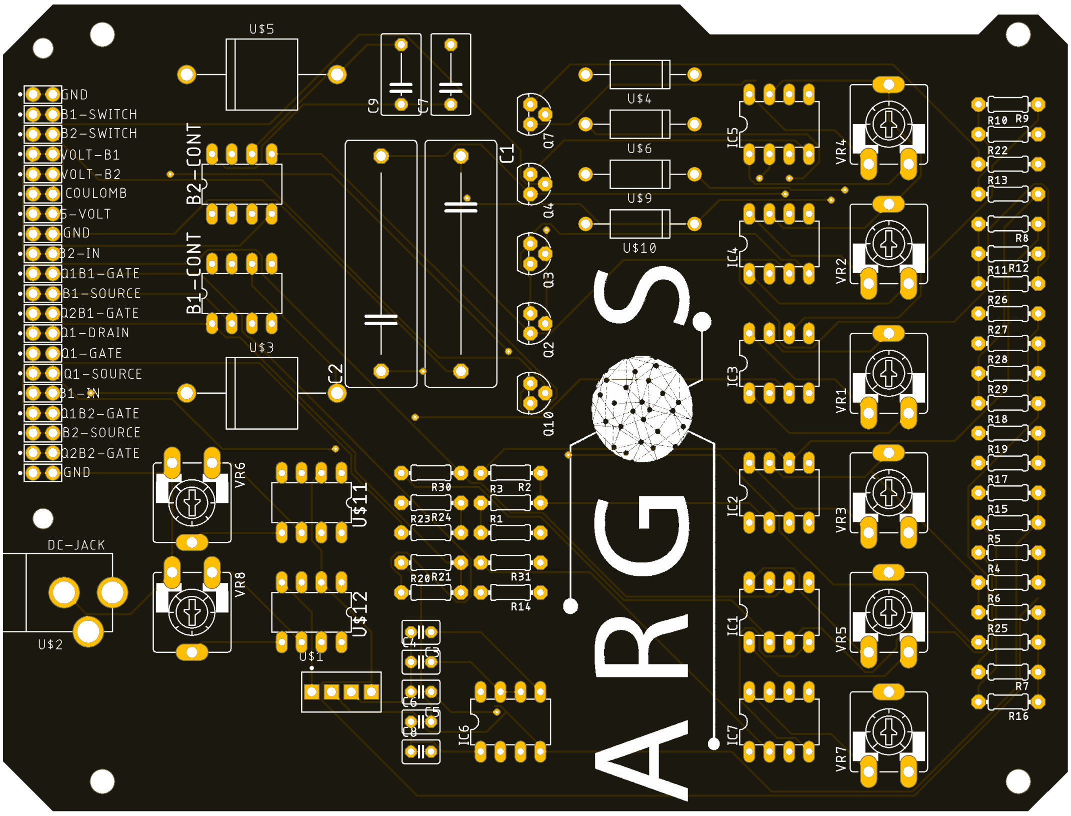
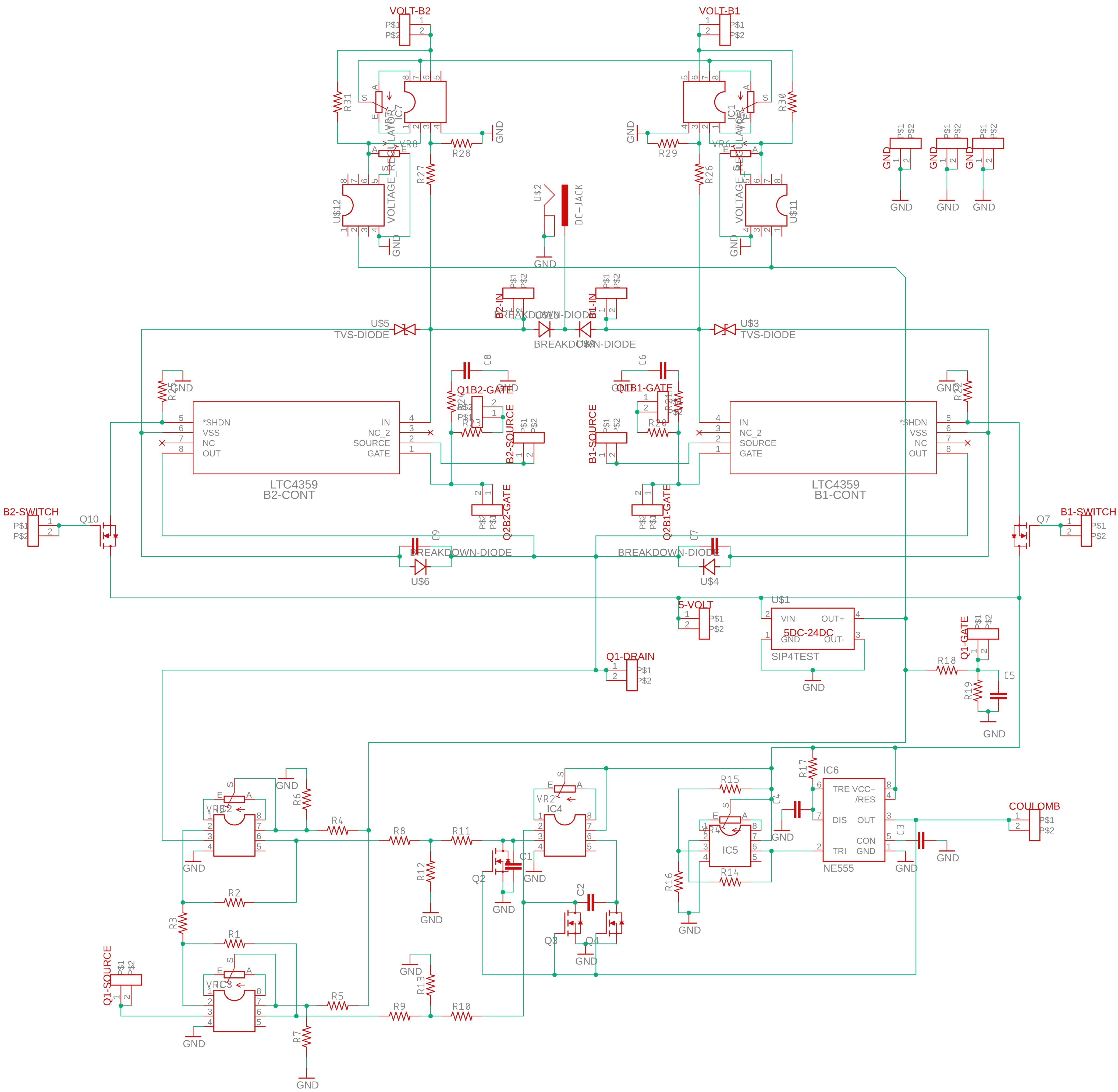


Figure G.1: The first battery controller PCB


VOLT-B2
VOLT-B1
R31
R30
IC7
IC1
R28
R29
VR8
VR6
R27
R26
U$12
U$11
VOLTAGE_REGULATOR
GND
U$2
DC-JACK
B2-IN
B1-IN
BREAKDOWN-DIODE
U$5
U$3
TVS-DIODE
C8
C6
Q1B2-GATE
Q1B1-GATE
R25
R22
R24
R23
R21
R20
B2-SOURCE
B1-SOURCE
*SHDN
VSS
NC
OUT
IN
NC_2
SOURCE
GATE
LTC4359
B2-CONT
B1-CONT
Q2B2-GATE
Q2B1-GATE
B2-SWITCH
B1-SWITCH
Q10
Q7
C9
C7
U$6
U$4
5-VOLT
U$1
VIN
OUT+
OUT-
5DC-24DC
SIP4TEST
Q1-DRAIN
Q1-GATE
R18
R19
C5
R6
R4
R8
R11
R2
R3
R1
R5
R9
R10
R12
R13
R7
VR2
VR3
IC2
IC3
IC4
C1
C2
Q2
Q3
Q4
R15
R17
IC6
C4
R14
R16
IC5
TRE VCC+
/RES
DIS
OUT
CON
TRI
GND
NE555
C3
COULOMB
Q1-SOURCE
P$1
P$2

G.2 Final PCB For Battery Control

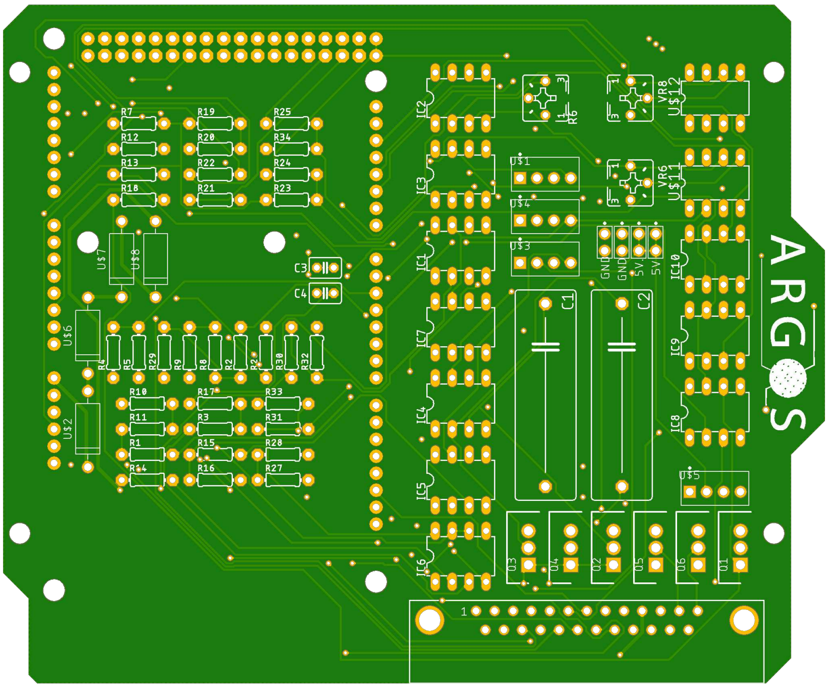
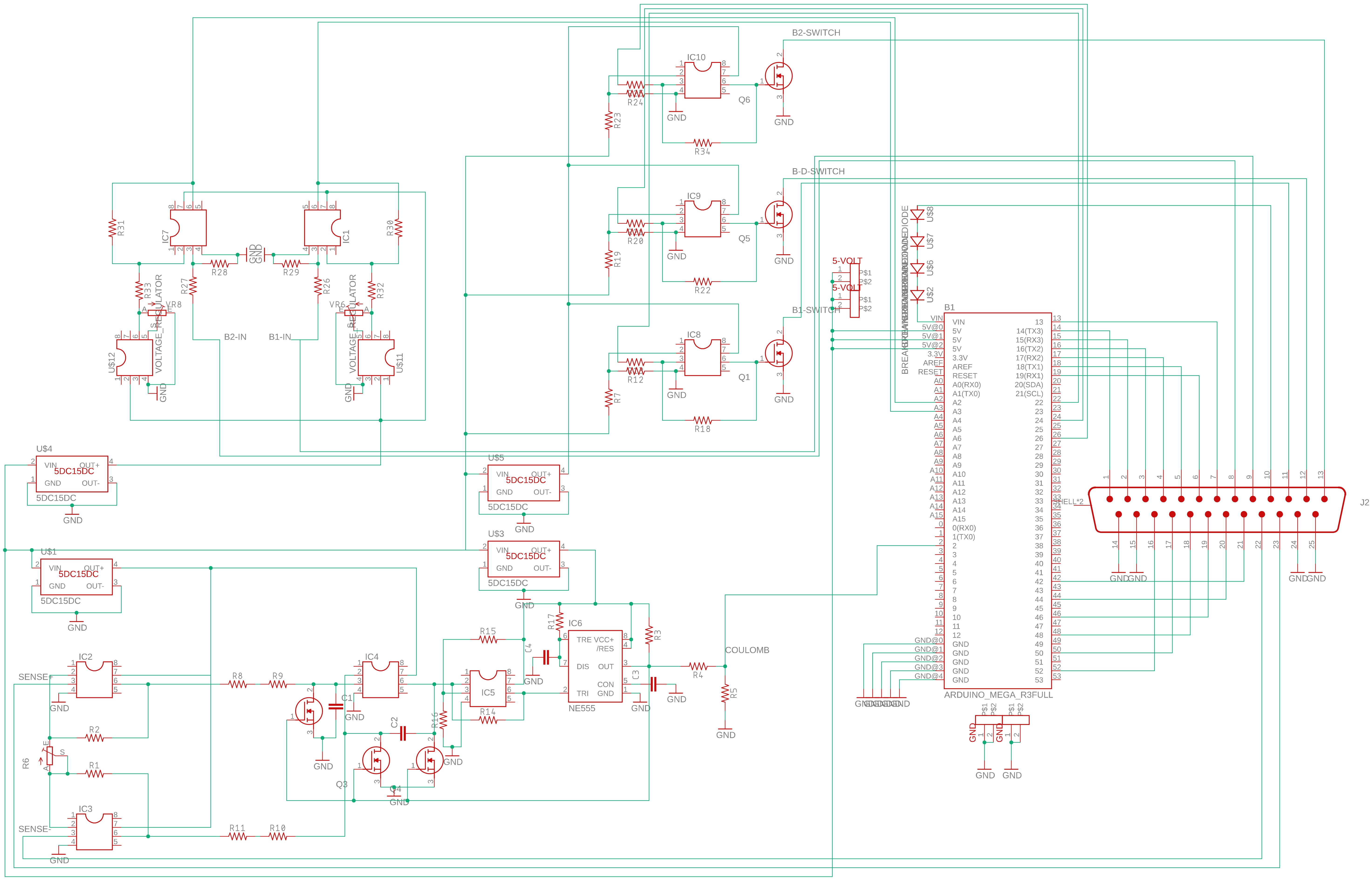
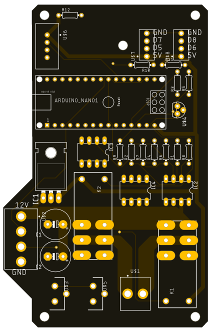


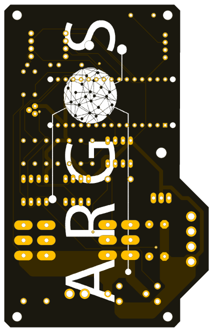
Figure G.2: The final battery controller PCB


B2-SWITCH
B-D-SWITCH
B1-SWITCH
B2-IN
B1-IN
5-VOLT
COULOMB
SENSE+
SENSE-
NE555
ARDUINO_MEGA_R3FULL
5DC15DC
VOLTAGE_REGULATOR
J2

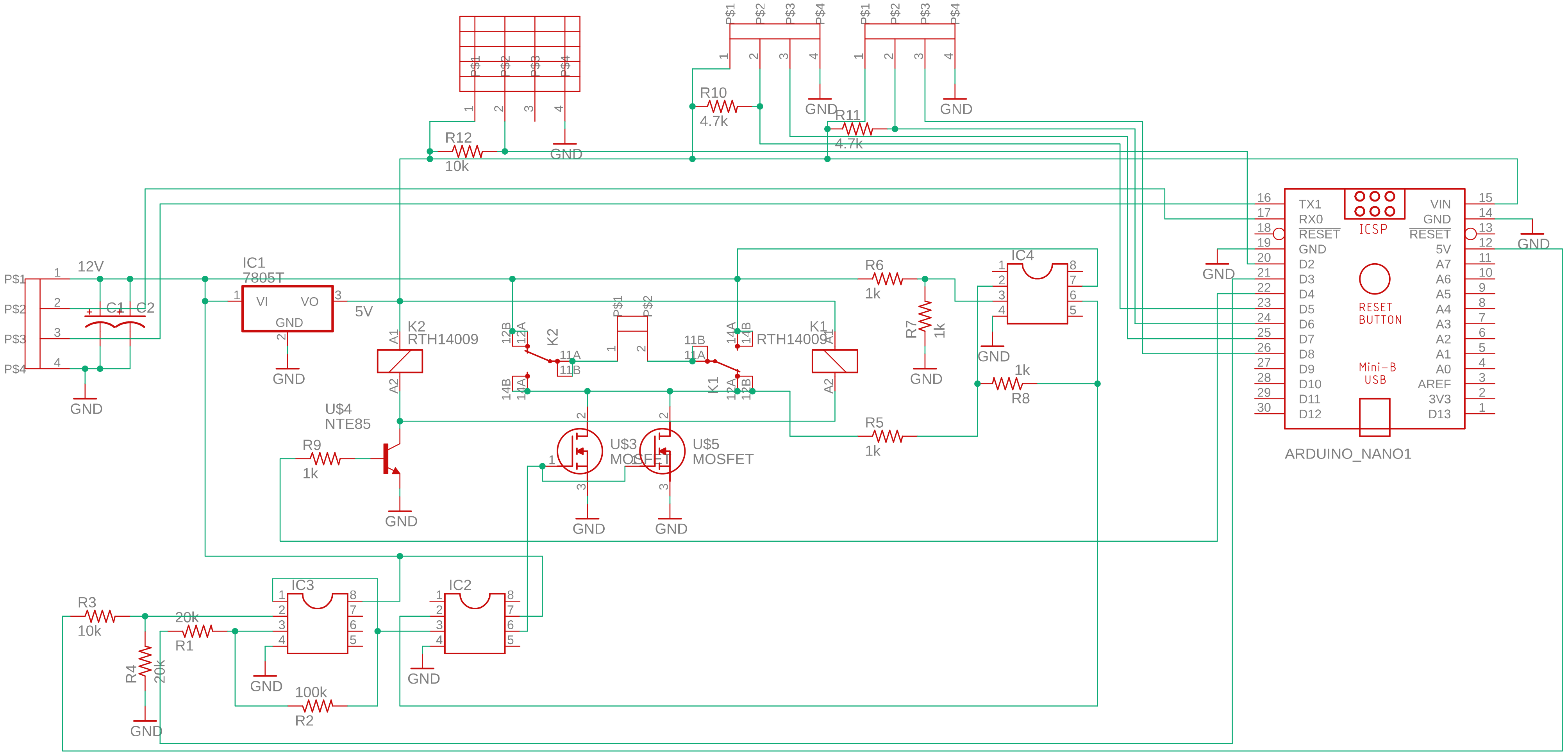
G.3 Temperature Controller PCB



(a) Top-side of the Temperature controller PCB



(b) Bottom-side of the Temperature controller PCB



12V
C1
C2
IC1
7805T
VI
VO
GND
5V
R12
10k
R10
4.7k
R11
4.7k
GND
K2
RTH14009
K1
RTH14009
R6
1k
R7
1k
IC4
R8
1k
R5
1k
U$4
NTE85
R9
1k
U$3
MOSFET
U$5
MOSFET
IC3
IC2
R3
10k
R1
20k
R4
20k
R2
100k
ARDUINO_NANO1
TX1
RX0
RESET
ICSP
RESET BUTTON
Mini-B USB
VIN
AREF
3V3