DECENTRALIZED CROSS-ROBOT CONTROL SYSTEM

Team Report

Luis Suarez	495322
Laurens Hoogendijk	2963353
Yordan Bakalov	3538974
Sem Cuijpers	2108962
Soufyan Karboub	3575981
Ijsbrand Geutjes	2976528

Preface

This report is the final report of a research project to decentralize cross robot localization system, which is part of a semester seven project at Fontys Mechatronics. The goal of the research project was to design an affordable and universal localization system for robots in a warehouse.

This report is mainly written for students who will continue working on the DCRC, as well as the teachers who will assess our work. This report will start with a research part. Then the chosen idea will be worked out and tested. In the end recommendations will be made

The project was worked on in the period of September 2021 until January 2022. This was during a corona pandemic which had its influence on working on it. Despite all this the group has put significant effort into the project.

We would like to thank Pablo Negrete who was our tutor for this project and give us a lot of helpful tips when needed.

Summary

For the Decentralized Cross Robot Control (DCRC) project, research was done into possible technologies to localize and orient Automated Guided Vehicles (AGV). The project was conducted as a research project with the initial phase used to research many different potential technologies which could assist in the problem and to answer the research question: How to localize an AGV in a highly dynamic or completely empty indoor (warehouse) environments with high precision?
Many different technologies are considered, ranging from QR codes used as waypoints to sound localization technology. Ultimately a vision-based system was decided as the best solution. This system is made up of a two-part system, the first part being light based beacons which are placed on top the AGVs and the other part being a camera array mounted to the ceiling which can detect the beacons and determine the location and orientation of the beacon.

A beacon and vision system were developed to be able to validate the research and to work towards a final design and working system. The light beacon uses green LEDs in a specific pattern which makes it easy to identify and orient. Certain LEDs can be turned on or off when the beacon is initially deployed to give it a unique identifier number. The current beacon design can have up to 1024 ID labels, which can allow for a system with 1024 AGVs. The vision system is an array of cameras that look down at the beacons and can determine their location, orientation and ID label. This information can be communicated to the AGVs allowing them to always know what their location is with high accuracy, no matter the environment around them.

Table of content

Table of content tables and figures

Introduction

The fast advancement of Industry 4.0 has brought a lot of attention towards it. Many manufacturers and business owners are turning towards it for their businesses by introducing Artificial Intelligence (AI), Automated Guided Vehicle (AGV) or other robots to automate and optimize the workflow in their factories or businesses. Nowadays, 31% of all businesses around the globe have fully automated at least one function and the trend show growth in the recent future. However, there are still a lot of challenges to face and overcome for Industry 4.0 some of them considering factory automation.

One of the issues is the low flexibility of the AGV robots. Meaning the different AGV manufacturing companies have different operating systems and separate ways of communication between their products. Which leads to misunderstanding between varied brands of AGVs and them recognizing each other as obstacles. Hence, any warehouse owner would have to only buy robots and automated systems from only one manufacturer once he has bought the first robot. This could lead to other problems such as inability to buy the needed product, for example if a business owner would need an AGV to move pallets of 300kgs (about the weight of a large motorcycle) but the manufacturer only supplies up to 200kg and from 500kg to 1000kg, which means low utilization of the machine even making it more expensive.

This project aims to tackle the problem by creating a system that can localize, track, navigate and split tasks between the robots called the CoLab board. However, it is split into several assignments. This assignment is called the **Decentralized Cross-Robot Control (DCRC)** system. Forward in this document it is referred as “the project”. The assignment requires a system that can pinpoint the position and orientation of an AGV in space with precision of ±10cm (about the length of the long edge of a credit card).

The project work was split into two Research and Development. The Research phase consisted of gathering ideas and generating concepts of systems which would fit the user requirements. The Development phase was essentially designing, developing, and producing a high-level prototype. The Development part was split into two parts- Device design and Software integration

All the work done on the project was systematically documented. First, in the Research phase, Agile approach was used, after some parts of the V-model were incorporated for the design. When the Development phase started the work was split into Device design and Software integration explained in detail in the sections of this document. Conclusions and recommendations for the continuation of the project may be found at the end.

Background info

Fontys lectorate and the robotics lab are helping The Holland Robotics Logistiek (OPZuid) to solve a logistic challenge. The whole project started a few years ago and has had several iterations improving and troubleshooting. The challenge is related to having different robots interacting with each other in a highly dynamic environments such as warehouses or a large empty warehouse.
The logistic sector is increasingly looking for autonomous solutions, often in the form of Autonomous Guided Vehicles (AGV) technology. But an AGV usually does not offer the complete solution that is needed to further automate the work process. Cooperation between AGVs cannot be taken for granted, while they can be complementary.
This cooperation is often a necessity in more advanced and complicated workspaces, and communication and localization is key. With that in mind, the following question rises: *How to localize an AGV in a highly dynamic or completely empty indoor (warehouse) environments with high precision?*

This document emphasizes the localization and navigation of AGVs in highly cluttered environments with lots of different sensor data to consider or completely empty environments with no sensor data to consider.
This report shows the development of the solution by a team of engineering students.

Problem definition & Deliverables

Problem

The contemporary solutions of logistic sector that are applied to warehouses is the use of AGV. But the development of AGV is now at point that it can only recognize other AGV from the same fabricator, or else they will see it as an obstacle. This creates a big problem for the warehouse's owner because this means they only can go in business with one fabricator of AGV.
This creates the scenario that the warehouse owner only needs an AGV that can lift 100 kg, but the AGV fabricator has a machine that can lift 50 kg or 400 kg. because of this the warehouse owner needs to pick the machine from 400 kg which is far beyond what they need. Thereby they will have a fleet of AGV's that is very inefficient to their warehouse.
The warehouses owners cannot choose varied brands of AGV's working in one area, because oof the following issues:

1. Sharing information (position, speed, direction, and task) between AGVs.
2. Different ecosystems from different robot-manufactures.
3. Loss of sensor information (due to space obstacles).
4. Goods that are not in the map will conflict with the system.
5. Task divider between robots.

Deliverables

The deliverables were set up during the beginning phase of the project in consultation with the client. In the following graph are the deliverables set:

Table 1: Deliverables

Nr.		**Requirement specification:**
5.1	**Must**	Document your code and findings on a GitHub page
5.2	**Must**	Report of decisions made while carrying out the assignments' goal
5.3	**Should**	Deliver an almost market ready product

System overview

Taking into consideration the client's needs and the problems presented with the project's already existing research, the solution comes in the form of a newly designed camera-based vision system. This system consists of 2 main parts: the beacon system and the AI trained software to detect said beacon.

After various stages of research and concluding in this method to locate the robots, the tool of Nvidia inference was selected to work as the main vision recognition software. After locking in this tool, a design for the beacon was made. A PCB of strategically placed LED lights that allows to inform the direction of the AGV and the serial number/ID of it.

The beacon was design with the intention of having strong enough LEDs that could be detected form a significant distance, being from the same bright color to help the training model of the AI and 10 LEDs working each one as a bit to recognize each one of them.

The Nvidia inference vision recognition docker can use different vision recognition systems, such Google Net, Detect-Net, among others. The docker allows us to create our own dataset of the system beacon. Therefore, by taking various images we can train a model well enough to detect the beacon in different conditions.

Fontys
UNIVERSITY OF
APPLIED SCIENCES

User requirements

Research

This chapter will cover the objectives of the project and how it will be done. after talking with the client, carefully reading trough the assignment and discussion within the team the user requirements were made

1.1 Investigation stage

• Investigate the functionality current CoLab board.

• Investigate robot navigation in Open / High dynamic spaces

• Investigate the components that will be needed

1.2 Integration stage

• Improve/Implement interfaces HW/SW between the CoLab board and robots of

multiple vendors.

• Integrate the acquired decentralized robot's positions and control in a dashboard

and display each robot position and directions for users to visualize.

1.3 Testing/Validation stage

• Test/validate the system using multiple mobile robots

• Test/Validate the current system in open spaces

• Test/Validate the current system in high dynamic environments

1.4 Documentation

• Document your code and findings on a GitHub page

User requirements

Research

Nr.		Requirement specification:
1.1	**M**	Investigate the guidance functionality of the current CoLab board
1.2	**M**	AGV must be able to navigate in Open /High dynamic spaces
1.3	**M**	Investigate current Lidar solutions for robot localization
1.4	**M**	Investigate current Ultra-Wideband (UWB) solutions for robot localization
1.5	**M**	Investigate other navigation and localization system
1.6	**M**	Determine the feasibility of the localization system

Improve/Implement

Nr.		Requirement specification:
2.1	**C**	Interfaces hardware between the CoLab board and robots of multiple vendors
2.2	**S**	Create a universal mounting system for the localization system on robots already know at the project

Integrate

Nr.		Requirement specification:
3.1	C	Create an interface with data (position and direction) for the user of the system
3.2	S	Create a communication interface between localization system and CoLab board / AGV

Test/Validate.

Nr.		Requirement specification:
4.1	M	Test and validate the solution in open environment
4.2	M	Test and validate the solution in high dynamic environments
4.3	M	Test and validate the accuracy of the localization system
4.4	S	Test and validate the compatibility of the localization system with the CoLab board and AGV

Boundaries and restrictions

This chapter covers the project boundaries and restrictions, in other words what the team has to do/use and what has not.

Firstly, the project is constrained by a period of 20 weeks which cannot be extended. Furthermore, the team has only one day in the week to work on the project and have access to the AGV's only once a week-every Wednesday. This only changes the last four weeks when the project team has almost full weeks to work on the project. Budget is another thing that constrains the project- it is limited, therefore buying new sensors or other needed equipment must be done with extreme caution.

Since the project was started a few years ago this brings a set of boundaries for the team which is working on the project now. For example, such boundary is that the team now is assigned to build-on the progress previously made and not begin from nothing. Another boundary like that is the use of ROS as operation system- this is because most of the AGV manufacturers use this operating system for their robots. All the documentation from the previous work for this project has been stored in GitHub, this will not change now, thus the team is restricted to use GitHub for updates on documentation and progress reports. Another boundary is that the system will not be used for one AGV only, therefore tests with multiple devices and robots must be performed. Test results and documentations is needed.

Research

The research was the main part of the first few weeks of the project. The team had a big brainstorming session producing several ideas to find and navigate an Automated Guided Vehicle (AGV) in a closed environment with high precision. The first brainstorming session was very productive as we had more than several different approaches to the problem. The team produced a few Computers Vision related solutions such as vision on the ceiling, vision targets on the ceiling, optical sensors, lasers beacons and many more. In this chapter all the ideas are elaborated to give more information about the complexity of the problem and familiarize the reader with the problems facing each scenario.

All the ideas

Firstly, explained are solutions with discrete output such as grid-based floor, traffic lights, objects with sensors, floor mapping.

None of the solutions were workable for the DCRC project because to make a Grid on the floor a warehouse would have to stop working for at least a day to rearrange. This is too expensive and therefore not practical.

The traffic lights are another solution that requires serious infrastructure inside of the building- this requires space, money, people, materials, etc. It is not a satisfactory solution because a business might have to stop working for the installation of the system.

Sensors on the object in the warehouse was another solution the team discussed, but as some of the members had experience in working at such places many of the employees do not care about their job and probably could throw away or send away the sensors while working.

Floor mapping is the only idea that was researched deeper from all the discrete outputs. It combines with other ideas to form the full system. This will be elaborated more later.
The next part of ideas which were research into more depth were magnetic, Ultra-wide band (UWB), Bluetooth beacons, Light beacons.
Since there was already developed UWB beacon the team had to just test it. However, the precision was not high enough to meet the user requirements. The team researched in depth if the UWB beacon could be improved but the improvements were not sufficient for the purposes of the project.

While researching another idea came up- the one for inductive beacons. They work the same way as the UWB, but they have no problem going through surfaces if they are not ferromagnetic. This solution is exceptionally good for office buildings that have no ferromagnetic metals present and are not noticeably big in volume. For warehouses there would be a need for a lot of beacons which would make them resonate more, increasing the fault in measurements. The more ferromagnetic metals there are in the building, the more electromagnetic disturbance is created, resulting in higher tolerance

measurements, therefore lower precision. Since the user requirements need a system with precision up to ±10mm the magnetic beacon system would not carry out this user requirement.
The Bluetooth beacons was the next idea on the list but even without much research it was discovered it is not a good enough solution because of the limited, quite small range of the Bluetooth technology and how much obstructions obscure the signal which would lead to low precision.

The light beacon is the only idea that would be used in the project. It consists of a beacon which disperses light which could be recognized and interpreted by a controller to decide which AGV is where in the warehouse. In the following chapters it is explained in detail how this is going to be done. This chapter concentrates on the general concepts that the team produced.
Other solutions also included light- such as LIDAR, line following on the roof, lightbulb navigation, Vision targets.

All the robots already have a LIDAR sensor to navigate around on their own, but there is a drawback in this functionality, as the LIDAR sensors can be blinded by other similar sensors. Sometimes same brand AGVs count this in consideration, but if the whole point is having vehicles from various companies, the system can't rely in this type of technology.

Line following on the roof was a solution that limits the freedom and autonomy of the robots- they would only be able to follow a predetermined path and not adapt to different situations. Therefore, this idea would not be able to carry out all the user requirements and was not worth developing.

Another idea related to light was to localize the AGV around depending on the frequency of the light from the lightbulbs anywhere in the warehouse. The problem with that idea arises when the building is excessively big and ambient light has access. Ambient light might disturb the measurements of the sensors and give wrong values not just imprecise. Even if ambient light has no access to the sensors another problem arises. When the light is shining it shines in all directions therefore different frequencies would be detected by the sensor on the robot leading to misdirection.

The idea the team decided was the best from this group of ideas consists of a camera on the roof and vision targets on each AGV or any other object in the warehouse. In the following chapters it is explained in depth how this idea is realized and therefore will not be elaborated as well here. It works on the principle of Computer Vision and Machine Learning to recognize patterns of light or any other matter.

The last research topic was sonar and sound navigation from Sorama. The team dug deeper into sound vision, to give every AGV his one frequency and hang Sorama sensors on the ceiling in a matrix to cover the whole aria. Use the sound camera to detect where the AGV is and use the frequency to differentiate the AGV from each other. but the cameras are too expensive, and the technology is still not perfectly reliable. Otherwise, it is a particularly helpful solution since it can go through materials without losing valuable data or precision.

At the end of the Research Phase each of the team members researched some ideas and presented them to the team with all their pros and cons. A list of criteria was made to easily compare all the ideas and decide. This list of criteria was shown to the client in a meeting, and it was agreed that the work should continue to develop the best idea. The criteria table may be found in the Appendix as table 3.

Decision on Idea

When analyzing the above-mentioned table, the team concluded that not a single idea is a fully workable solution but some of the ideas could be implemented together and cover all the user requirements. The two ideas that make the system today are Light Navigation and Vison system from top to bottom. The concept came along in a brainstorm session and consisted of a camera on the ceiling looking down and recognizing patterns or beacons on the AGVs or any other obstacle. This way the team could split into two subgroups and develop the solution.

One group had the task to develop a visible light emitting (or reflecting) beacon which is to be placed on each AGV or any other obstacle. The beacon had to have a feature which distinguishes the front and the back this way the vision system could allocate and know the orientation of each robot in its frame. For any other obstacles, the first group decided to use reflecting tape since it is cheaper and more practical for the purpose, eventually the team opted to update this technology and will be further explained later in the report. More about the development of the beacon may be found in the Finalized technologies section of the report under Hardware.

The other half of the team had the heavy task of teaching the computer to recognize the patterns of the beacon and the vision targets. Different algorithms were used to teach the model and the team created their own dataset with the beacons to train with and test. The Vision team's progress is elaborated in detail in the Finalized technologies section of the report under Vision.

Finalized technologies

The AGVs need to find out where it is in the environment where it is placed and where needs to go. This environment can be two kinds of spaces, the first space is a large empty homogeneous warehouse, the second space is high dynamic space with a lot of changes in time to its environment.

These two kinds of spaces face multiple problems. Some of those problems are different and some correspond to each other. To create a solution for these problems, research was done about various kinds of technologies for localization.

With the findings from the research, there is a plausible technical strategy found for the localization feature. The strategy is the use of camera (Computer Vision) and a tag (Beacon). The cameras will be placed on the ceiling facing downwards to the floor and for a large surface area there will be multiple cameras placed in a matrix formation. The tags will be placed on the AGVs, facing upwards to the ceiling.

Camera

The camera will be placed in a grid formation to cover the whole space where the AGVs will be active. The problem of the use of (vision)cameras is the lack of absolute precision and is seen as a relative and not absolute. But at present the (vision)cameras have high performance and accuracy and use of software is also increased. According to the paper it states that it can be used for absolute. Therefore, this problem can be neglected in the use-case for this project.
The information that camera needs to produce for this strategy is a high precision video stream of a surface area at a distance from 20 meters height. Therefor it can check this area for activity.
When an object or AGV is under the monitored area, the X and Y coordinates can be calculated using pixel position.
The covered surface area can be increased simply by adding a second camera. This second camera will be placed in a grid formation, because then a central system can be made. This central system will combine all the pictures into a big picture that consist of multiple pictures. This system will be made of macro and micro architecture to make the system work efficient and clear.

Beacon

The beacon is a device that will be placed on top of an AGV. It has a working name " V.E.R.G.A.– Vision Enhanced Robotic Guidance Assistance ". The beacon has two main goals to the system. The aims are the following:

1. X, Y orientation of the AGV
2. Identifier of the AGV

The vision system can recognize an AGV but doesn’t know which AGV it is particularly. There are some cases of warehouses that have multiple AGVs of the same kind, because of that the vision system cannot differentiate between those AGV’s. Therefore, there needs to be an identifier placed on the AGV to differentiate between those AGVs.
There are some AGV like the Omron LD 60/90 that are almost symmetrically on the top view. This leads to the problem that the vision system doesn’t know which direction the AGV is heading in. Therefore, it can lead to some problematic situations.

For these challenges we have designed a beacon that eliminates these obstacles. The beacon is a T-shaped case that has multiple openings on the topside which let the green light illuminate through, see figure 7. Each opening is a binary bit that can be on (1) or off (0). There are in total of 10 adjustable LED, with that you can create 1024 combinations. With this capacity the system can use 1024 AGV’s simultaneously, which is well beyond the demand for a single warehouse (according to our client).

Vision

The main goal of the detection is to have a precise location of the beacon in diverse and dynamic spaces. To achieve this goal there needs to be a recognition system that's able to identify and localize the beacon. The system must work in different environmental elements like changing lighting, different angles, unknown obstacles and misplaced objects.

Therefore, the project team have designed an artificial-intelligence (AI) vision-based system. It is a camera-based vision system that can detect multiple objects, determine their position, and differentiate between them. The system can be separated into two sub systems that have different goals. This whole system works on the Nvidia Jetson Nano(2GB).
The first sub-system will use the camera feed to detect multiple objects that need to be recognized by the system. When this system detects an object, it will transfer the data to an AI model that has been trained with deep transfer learning. The model contains data from multiple beacon configurations combined with different ID labels.
The system will compare the model with the camera feed to detect and to recognize a known object from the database of the model. When the system has a match, it will send the data to the monitor and mark the object with a bounding box and ID label, see figure 5 in the appendix. The benefit of this system is that the model can always be expanded with new data and variables. This process has been made simple with the use of the step-by-step plan (see appendix).

The second sub-system will use the camera feed to localize the objects from sub-system one. This system will extract X and Y coordinates from the camera feed. To localize the objects from sub-system one, there will be a certain point within the bounding box middle point. With this target point the system can calculate the X and Y coordinates. This data will be sent to the monitor and will be placed along with a bounding box and label ID.
When the system can detect the beacon there is a possibility that the system will lose the data when the camera feed has been changed. So there needs to be a system that keeps track of the location after a successful detection of an object. This will be possible after the creation of a data log where the data of location from the object's id points are stored. To keep track of the object the system needs to compare the previous frame with the new frame. Because of this there are a certain number of possibilities that can occur during this process. The first possibility is that the object is in the same place, the second possibility is that the object has moved in a certain direction, or the third possibility is that the system isn't able to detect the object.
In the first case the ID points match exactly. In the second case the id points are slightly different and will replace the last location data with the new data. In the last case there is nothing to compare, so the system will send unknown data to the data log. By storing multiple points, it is possible that if the system misses a detection in a frame, no ID point will be terminated. This will help with fulling in the blank spots.

Hardware

PCB(V.E.R.G.A.)

The team has decided to use a light beacon and computer vision system to recognize, identify and allocate a robot no matter the state of the environment. The idea came along by looking at an LED strip and using similar technology to give an ID to each robot.

This part of the report focuses on the Light Beacon and its electrical design as well as the mechanical design. The beacon follows the schematic below and consists of 3 stationary, constant LED lights in the front to determine orientation in space, then it has 10 LEDs positioned in a straight line and adjustable by a dip switch so that there would be 2^{10} or 1024 unique IDs at the end there is one more constant LED light to help determine orientation.

To control a LED without damaging it a constant current is needed. There are two ways to carry out this task first- use a voltage regulator and a couple of resistors to keep the current constant (CCC) and connect the LED in series with it or use a voltage regulator such as a DC-DC convertor and connect an individual resistor to each LED and connect everything in parallel (CVC). The main difference comes when using a circuit which consumes more power. In such case only constant current control is recommended because when current passes through a resistor generates heat which could potentially damage the resistor or diminish its resistance enough to raise the current considerably and possibly damage the LED. Constant voltage controller is also much more unpredictable for applications with higher current consumption because it could never be known which resistor will fail first and therefore which LED. For this application, a huge current is not needed but several LEDs placed in a line connected in parallel to each other while having the same current is necessary. Another weight factor when choosing the approach was the size of the circuit and PCB that would be needed for it. Since the CCC needs a small circuit with a voltage regulator and a couple of resistors for each LED, this would increase the size of the PCB and the beacon accordingly. Therefore, the team chose to use the Constant voltage control- it uses less space and is perfect for the needed application.

When the LED control was decided components had to be chosen. The team needed an LED that would shine bright enough to be clearly visible from 10m, that is why TC5050 RGB was chosen. They have sufficient brightness for the project. Their input voltage is 3.3V and the maximum current is 25mA. They have a certain voltage drop of 2.8V and this yields a resistance of 25 Ohms to let 20mA of current through the LED. Green light was determined to be the clearest and easiest to distinguish by the vision algorithm. For this reason, only the green LED from each of the TC5050 RGBs is connected to power.

The beacon circuit also includes a voltage regulator. The team has chosen a DC-DC converter with 3.3V output capable of continuously supplying up to 1.5A of current. On the PCB which was specially designed for the project there are four pads for the voltage regulator to fit. It has an input voltage between 5.3 and 26V, 30mV output ripple and extremely high efficiency (above 90%). A capacitor is placed near the output of the buck-convertor to cut any voltage ripple.

PCB enclosure

The PCB has been enclosed in a custom, made for 3D-printing, enclosure. The enclosure is made up of 2 halves that snap together with the V.E.R.G.A. PCB in between the 2 halves. This keeps the PCB perfectly in place and protects it. The case has 4 M5 mounting points on the bottom so that it can be easily mounted to the top of any AGV. The enclosure also has a DC barrel connector on the side allowing for easy power connection with a wire run from the AGV. The two halves of the enclosure can be seen in figure 8 and 9.

The V.E.R.G.A. sits deep inside the enclosure, with only the LEDs and the DIP switched exposed. These parts are recessed for protection and to make the LEDs easier to be able to be distinguished between. The DIP switches can be easily adjusted when using a small screwdriver or similar object but not by simply reaching them with somebody's hands. This prevents the DIP switches from being accidentally bumped but still allows for easy configuration when installing the PCB. The LEDs being recessed prevents the light of one LED spilling over to another LED and potentially having the vision system detect and LED as on when it is off. It also creates a stronger contrast between the LEDs.

Testing

Testing procedure

To validate the results of the research, testing must be performed and needs to answer several questions. The main questions to be answered are, is the camera able to detect the location and orientation of the beacon and is the accuracy within the proposed requirements? To be able to conduct tests, the following testing procedure has been created.

1. Configure the Beacon with an ID number
2. Position the Beacon within view of the camera
3. Log the output of the camera

This testing procedure needs to be conducted in various environments to validate the initial research question, how to localize an AGV in a highly dynamic indoor (warehouse) environment with high precision? The AI has been trained to detect the beacon, but various environments need to be used so the dynamic aspect can be tested. The vision system needs to be validated so that it does not detect any false positives, which would cause it to send the wrong location and orientation to an AGV. Initially the testing would be conducted in an empty environment and when those tests pass, the testing can move on the dynamic environments.

Test cases

In this document the testing of the Beacons of the DCRC project. The testing aims to cover as many possible design flaws as possible. The input voltage will be tested, the current going through the LEDs and resistors, the total power draw of the circuit and condition of the beacon after a supposed operating shift in a warehouse (assumed to be between 16 and 20 hours). The testing was done individually for each resistor/LED.

Firstly, the input voltage was measured. Experimentally it was proven that 3.8V is the minimum voltage needed to keep the LEDs on while consuming 180mA (seen on the power supply). For the sake of the experiment all the tests have been performed with an input voltage of 5.7V for maximum efficiency of the DC-DC convert. The output voltage of the DC-DC convertor is 3.27V, steady DC voltage.

Secondly the resistance of each resistor was measured to make sure the value is the desired one and not too much larger or smaller. All the resistors are within the tolerance ranges and behave just as expected.

In the end the current was calculated by taking the voltage over the resistor and dividing it by the resistance. The total current of the system was measured to be 124.3mA, but it was calculated and expected to be 191.3mA. However, this does not change the performance of the system and the fact that it passes all the tests needed and satisfies all the system requirements for it.

The long operation test is done to check if the beacon could be able to work a full shift in a warehouse(16h) because it was assumed AGVs need 8h charging time. The test was concluded as successful since the LEDs nor the resistors were hot enough to change their value or function unproperly. This test also proves that the casing for the beacon could be made from plastic because plastic such as PLA has a relatively low melting point- around 50 degrees it becomes soft and pliable. With the setup of the PCB there is no danger of the board overheating and melting or deforming the casing. No heatsink is needed for the same reason.

Conclusions and Recommendations

Conclusions

The Decentralized Cross Robot Control (DCRC) project has started a few years ago but this report describes only the latest development on in. At the start of 2021-2022 school year an addition to that project was kicked off with the goal of recognizing and locating various AGVs and other objects in an empty or cluttered environment. The first part of the project was mainly research and the team came up with several ideas, some of which were implemented together to form the final solution- Computer Vision and Light Emitting beacons. From that point on the team was split into two groups- one worked on the beacons, one worked on the vision. The beacon team designed and tested a 10-bit LED beacon with green LEDs since they are the easiest to distinguish by camera out of red-green-blue; each beacon can be placed on an AGV getting power from it. That way the beacon would always be connected to the same AGV or would be changed very rarely which leads to minimal maintenance. The vision team achieved remarkable results with the available data and the self-created dataset specially for the project.

It is clear a lot of progress has been made on the project. Most of the goals were done, namely, identifying an beacon in an empty environment with little to no sensor data or completely cluttered environment with lots of obstacles such as other robots, other moving equipment, and people.

To conclude, the project was a success most of the goals were carried out and most of deliverables successfully delivered to the customer.

Recommendations

Even though the Decentralized Cross Robot Control project was a success there are still some recommendations for future continuation of the project.

- Use an excellent quality camera
- Make a big dataset of the beacons in various positions and with different camera angles. Train the model with that dataset. This can be done with the step-by-step plan vision software.
- Improve the beacon based on the results of the testing and designing previously done (recalculate resistor values, capacitor values, etc.)
- Design a mounting mechanism for the beacon casing
- Improve code for tracking the estimated location or invest in existing programs

Appendix

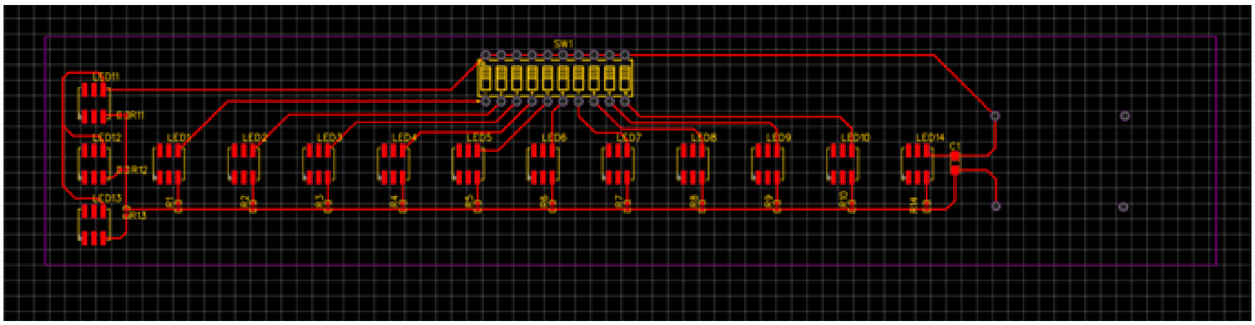

Figure 1: PCB schematic of the beacon

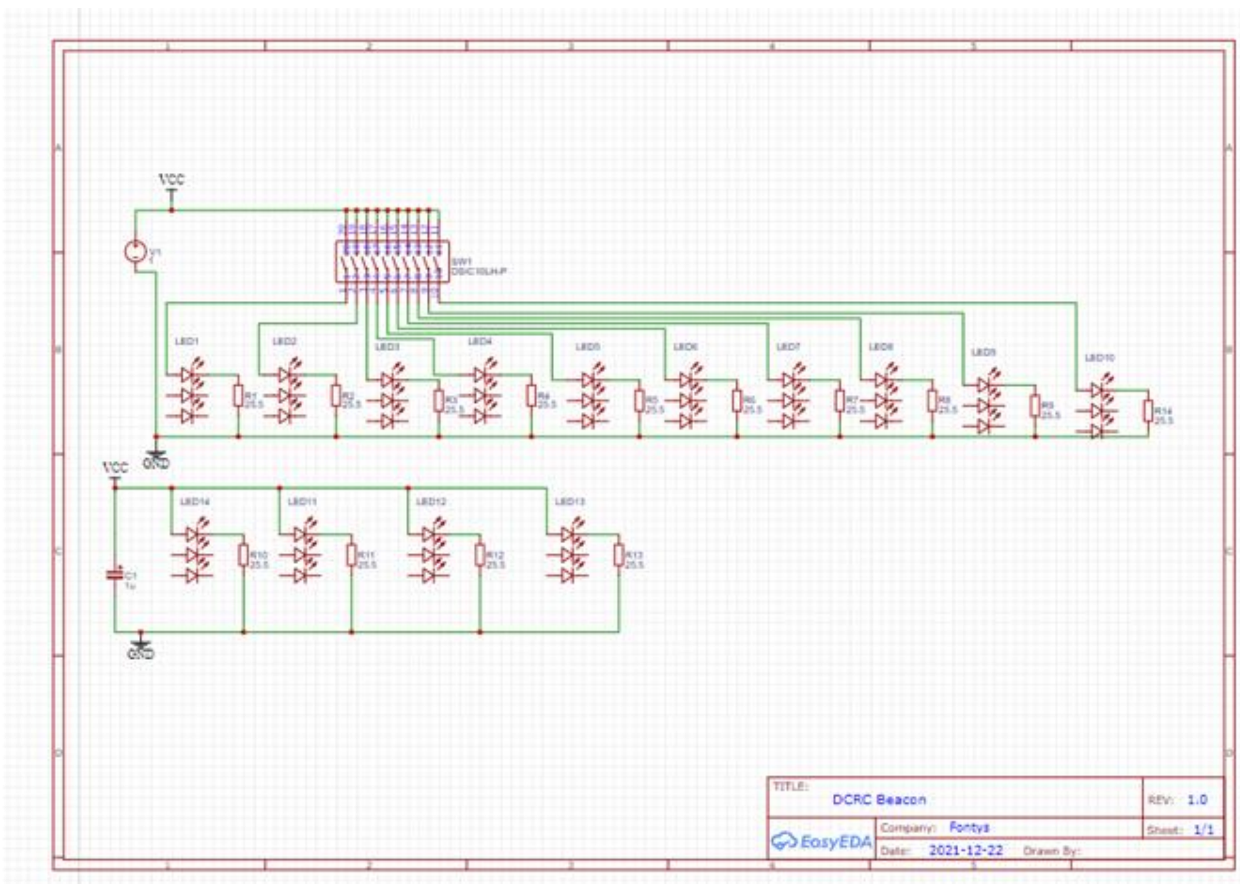

Figure 2: Electrical schematic of the beacon

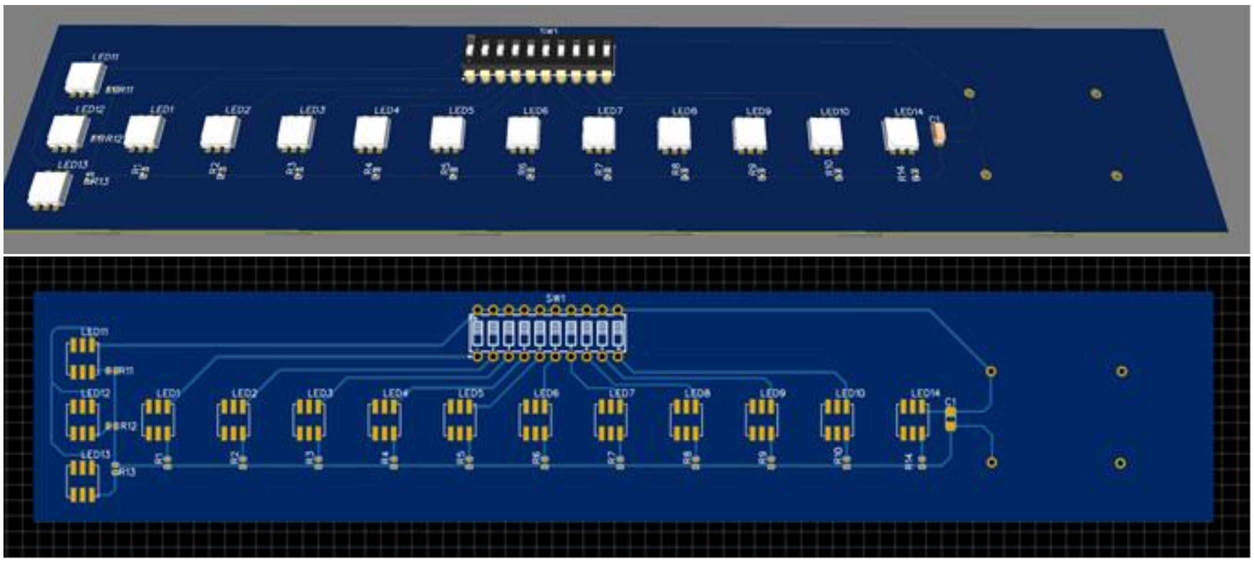

Figure 3: 3D model of the PCB from the beacon

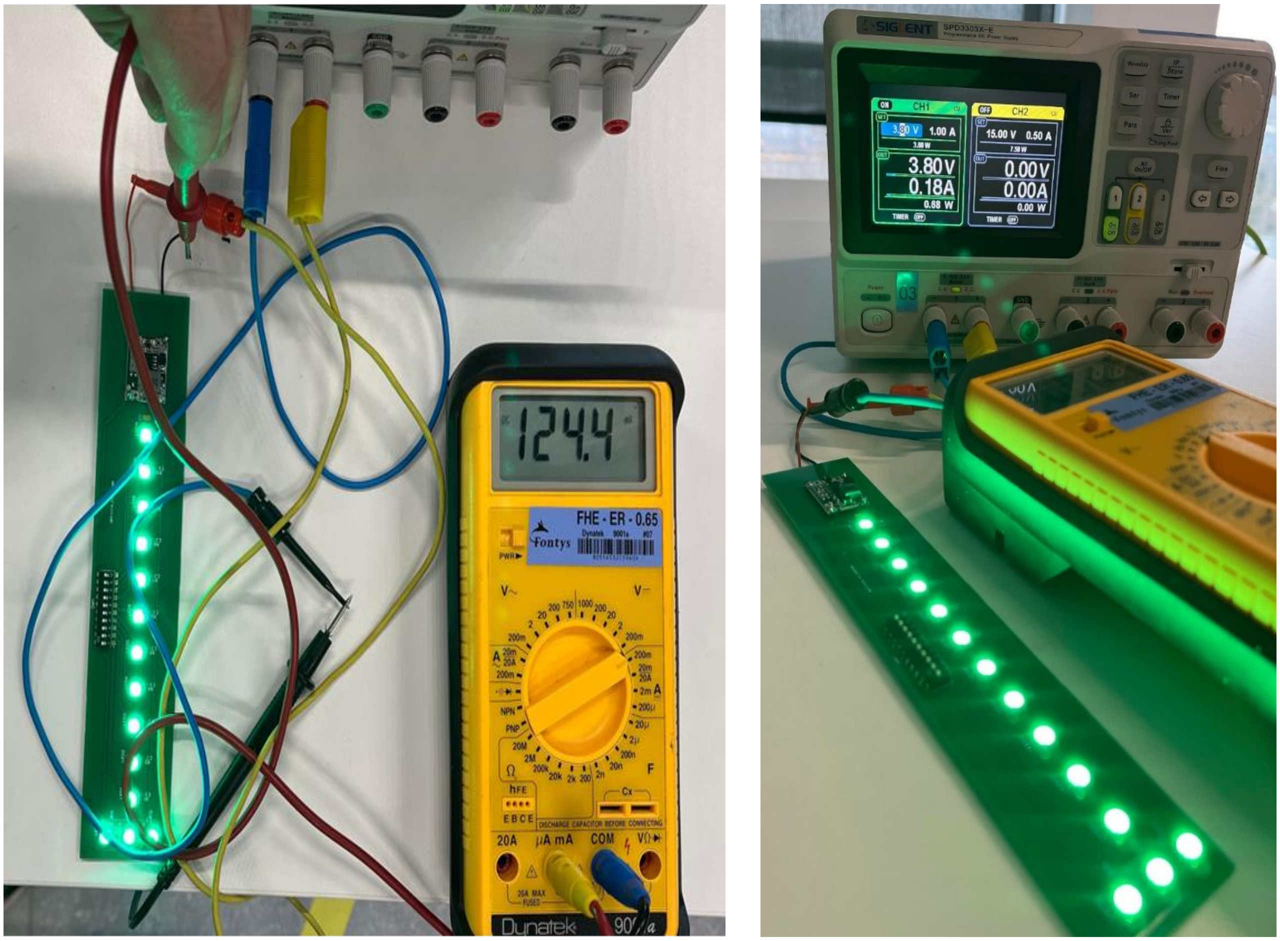

Figure 4: Testing procedure of the beacon

Table 2: Test results from the PCB

Beacon	Vin min	Vin test	V after DC-DC	V over resistor	Resistor R	I per LED
LED 1	3.8(power supply)	5.7V	3.27V	0.347V	25.8	0.0135A
LED 2	3.8(power supply)	5.7V	3.26V	0.335V	25.7	0.013A
LED 3	3.8(power supply)	5.7V	3.27V	0.353V	25.7	0.0137A
LED 4	3.8(power supply)	5.7V	3.27V	0.347V	25.7	0.0135A
LED 5	3.8(power supply)	5.7V	3.27V	0.350V	25.7	0.0136A
LED 6	3.8(power supply)	5.7V	3.27V	0.383V	25.7	0.0149A
LED 7	3.8(power supply)	5.7V	3.27V	0.345V	25.8	0.0134A
LED 8	3.8(power supply)	5.7V	3.27V	0.345V	25.8	0.0134A
LED 9	3.8(power supply)	5.7V	3.27V	0.307V	25.7	0.012A
LED 10	3.8(power supply)	5.7V	3.27V	0.359V	25.9	0.014A
LED 11	3.8(power supply)	5.7V	3.27V	0.367V	25.7	0.0143A
LED 12	3.8(power supply)	5.7V	3.27V	0.375V	25.7	0.0146A

LED 13	3.8(power supply)	5.7V	3.27V	0.339V	25.7	0.0131A
LED 14	3.8(power supply)	5.7V	3.27V	0.367V	25.7	0.0143A

Figure 5: Vision sub-system 1 live feed

Table 3: Criteria table

System name	Type localization	Type system	LoS constraints	Precision	Range of beacon	Possible Number of Beacons	Receiver-Beacon interaction	Price
A POSITIONING SYSTEM WITH NO LINE-OF-SIGHT RESTRICTIONS FOR CLUTTERED ENVIRONMENTS	Relative positioning	Magnetic ultra-low frequency Beacon (Static)- Magnetic receiver (on AGV)	No Line-of-sight constraints. Magnetic waves go through surfaces	Depends on the number of beacons (10cm for a number up to 100)	Up to 10m All directions	Thousands. Depending on the size of the warehouse and the precision needed	Each receiver must receive signal from at least 3 beacons to determine position accurately	0.049 cents of a Euro per Tag.
Camera with sound sensor	relative positioning	camera with sound location	expensive and complicated	I can't find anything about precisions, but I think	-			for one sensor and a starters kit 25000 https://www.sorama.eu

Fontys
UNIVERSITY OF
APPLIED SCIENCES

				minimal 10 cm				
QR Code	camera below the AGV which find the QR codes on the ground	QR codes on the ground and drives from QR code to QR code	The QR codes must be perfectly placed all over the area how precis it is placed on the floor the moor the precise the AGV is. and it drives from QR code to QR code.	below 1mm	every 50 cm there is a QR code	every 50 cm	1 QR code	the amazon QR code on the ground

Vision system top-bottom	camera in the roof scans for robot	Vision	If AGV is carrying something the camera can't recognize the AGV		Few cameras for scanning all AGVs			from 200 till a couple of thousand
Vision system bottom-top	camera in the robot scans roof	Vision	If AGV is carrying something large the vision system may be blocked		Every AGV needs a camera			from 200 till a couple of thousand
3d vision system	front of robot	Vision		below 1mm -- till a few cm	every AGV needs a camera			from 200 till a couple of thousand
LiDAR	relative positioning	LiDAR	only see what directly is surrounding the AGV	0,5 to 10 mm	not beacon based, onboard sensor			from $100 to a couple of thousand

Step-by-step plan vision software

- **Check:**

You must install your Nvidia Jetson Nano memory with the IMAGE file which include jetson-inference.

- **Preparations:**

Make a folder in the following address:

home/nvidia/jetson-inference/python/training/detection/ssd/data/XXXXX(make here your folder)

In that folder you need to make a TXT file with the label: labels

In there you will place every class, each row contains one class

- **Tools**

Nvidia Jetson Nano(2GB)

All the cables of the jetson Nano

PC or laptop

- **Extra material**

https://www.youtube.com/watch?v=2XMkPW_sIGg

- **Step 1:**

run inference

$ docker/run.sh

Or

$ ./inference

- **Step 2:**

Go in the folder of SSD

$ cd python/training/detection/SSD

- **Step 3:**

Run the camera capture tool to make pictures and to label you objects

$ camera-capture csi://0

Or with USB camera

$ camera-capture /dev/video0

Settings of camera capture tool:

detection

Datapath: home/nvidia/jetson-inference/python/training/detection/ssd/data/XXXXX (folder that you have made)

class labels: home/nvidia/jetson-inference/python/training/detection/ssd/data/XXXXX (folder that you have made)/labels.txt

- **Step 4:**

Start re-training your model

$ python3 train_ssd.py --data=data/XXX (folder that you have made) --model-dir=models/XXX (folder that you have made) --batch-size=2 --workers=1 --epochs=1

- **Step 5:**

Export your file with onnx

$ python3 onnx_export.py --model-dir=models/XXX (folder that you have made)

- **Step 6:**

Deploy your model to live stream

$ detectnet --model=models/XXX(folder that you have made)/ssd-mobilenet.onnx --labels=models/XXX (folder that you have made)/labels.txt --input-blob=input_0 --output-cvg=scores --output-bbox=boxes csi://0

Or with USB camera

$ detectnet --model=models/XXX(folder that you have made)/ssd-mobilenet.onnx --labels=models/XXX (folder that you have made)/labels.txt --input-blob=input_0 --output-cvg=scores --output-bbox=boxes /dev/video0

Note:

If you want to make an excysting model bigger with new data follow the step 1 to 6, using the folder of that model.

Figure 6: Complete beacon with housing

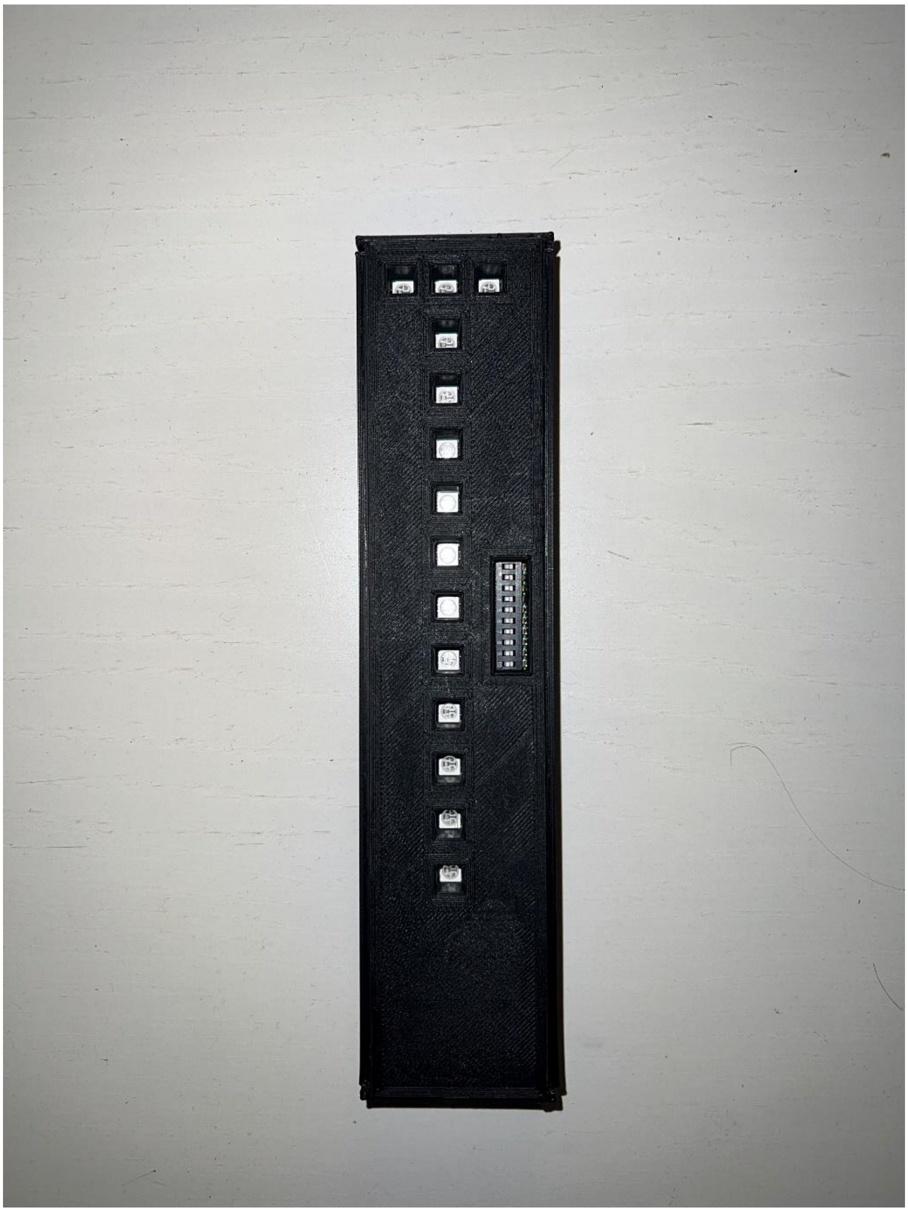

Figure 7: Complete beacon with housing

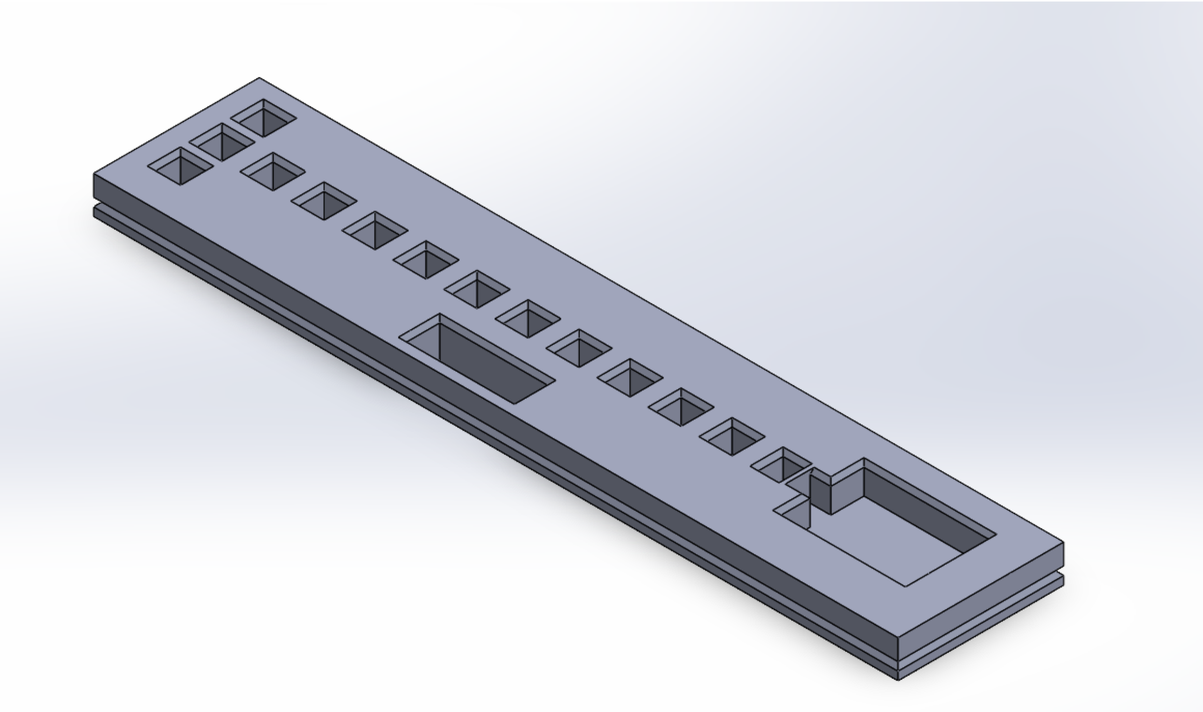

Figure 8: Top lid of the beacon housing 3D model

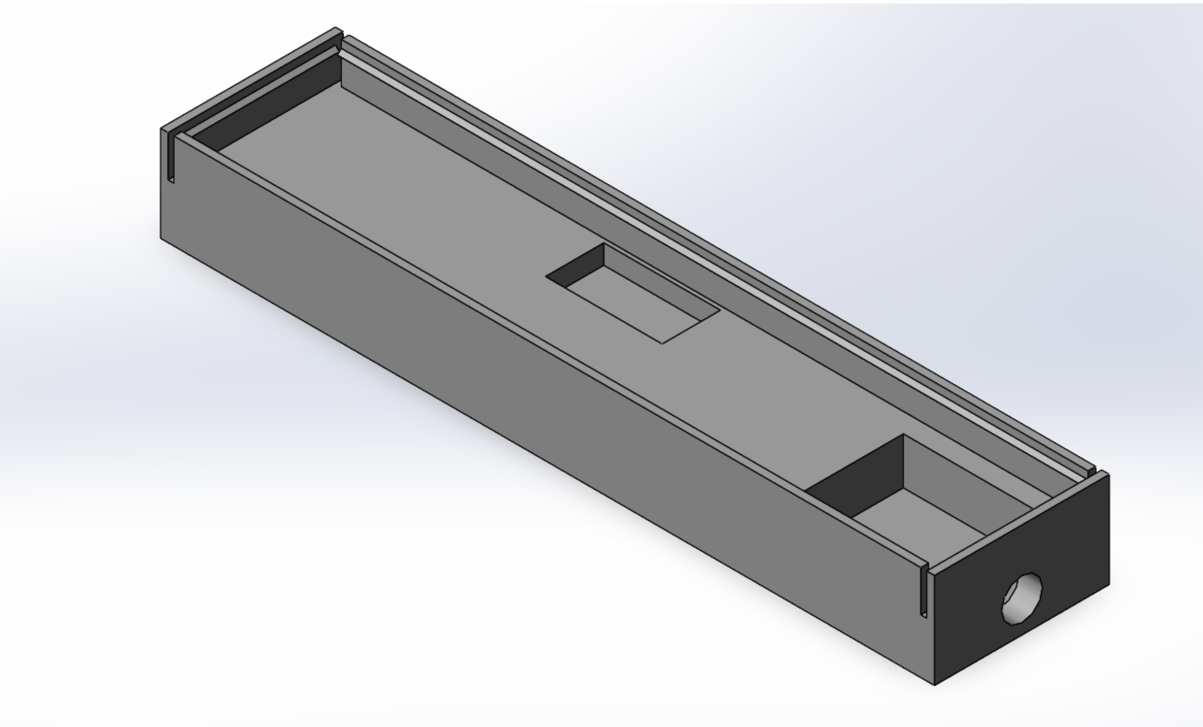

Figure 9: Bottom lid of the beacon housing 3D model