TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies
Department of Computer Systems

IA70LT

Jürgen Soom 178250IASM

SMART CAR DECK SENSOR NETWORK DEVELOPMENT FOR TALLINK MEGASTAR

Master's Thesis

Supervisor: Mairo Leier
Researcher, PhD

Tallinn 2020

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Arvutisüsteemide instituut

IA70LT

Jürgen Soom 178250IASM

TALLINK MEGASTARI TARGA AUTOTEKI SENSORVÕRGU ARENDUS

Magistritöö

Juhendaja: Mairo Leier

Teadur, PhD

Tallinn 2020

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, the literature and the work of others have been referenced. This thesis has not been presented for examination anywhere else.

Author: Jürgen Soom

2020-01-15

Abstract

The goal of this thesis is to propose a solution for smart sensor network for Tallink Megastar car deck and develop an initial prototype based on those recommendations and requirements. The developed prototype is able to detect any moving vehicles with velocities up to 40 km/h, identify their type based on the height and guide them to designated location on the car deck. Detection is based on ultrasonic sensor, that is capable of measuring every 100 ms with range up to 7.65 meters. System also featured a RGB LED, that acted as an indication, that could guide and notify the driver using different colors, based on the situation. Enclosure is designed and manufactured to protect the components from external conditions, such as moisture, water, foreign objects and having IP55 rating. Deck sensor is attached to car deck's support beam with two neodymium magnets with additional support from a attachment clamp. Communication between other sensors, subsystems and server are done via CAN Bus and Ethernet. Overall the system is competent enough to fulfill all the criterias was operate within the set conditions and limits.

This thesis is written in english and is 85 pages long, including 7 chapters, 32 figures, and 28 tables.

Annotatsioon

Käesoleva magistritöö eesmärgiks on arendada sensorvõrk, mis tuvastab ja juhatab sõidukeid Tallink Megastari autotekil. Töö keskendub esmase prototüübi arendusele, mille käigus kõigepealt uuriti turul saadavaid sarnaseid lahendusi ja nõudeid, millele seade peab vastama. Sõidukite tuvastamine põhineb ultraheli anduritel, mis suudab mõõdab iga 100 ms tagant. Antud andur valiti välja, robustsuse, madala voolutarbe ja odava hinna tõttu. Lisaks sellele lisati ka indikaator, milleks on mitmevärviline valgusdiood, millega juhatatakse autojuhte ning antakse edasi juhiseid. Kogu elektroonika jaoks disainiti ja valmistati eraldi korpus, mis kaitseb seadet erinevate keskkonna tingimuste eest ning vastab IP55 klassile. Korpus kinnitati autotekil olevatele toestus taladele kasutades selleks neodüümmagneteid ja kinnitusklambrit. Väljaarendatud süsteem suudab tuvastada autotekil liikuvaid sõidukeid keskmiselt 10 korda sekundis, vastavalt kõrgusele tuvastab nende tüübi ning juhatab sõidukid sobivasse kohta, ilma inimese abita. Kogu informatsiooni vahetus teiste sensorite, alamsüsteemidega ja laeva serveriga on lahendatud läbi CAN võrgu ja Etherneti.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 85 leheküljel, 7 peatükki, 32 joonist, 28 tabelit.

List of abbreviations and terms

AC Alternating Current

ADC Analog-to-Digital Converter

AI Artificial Intelligence

AR Augmented Reality

CAD Computer-Aided Design

CAN Controller Area Network

CRC Cyclic Redundancy Check

DC Direct Current

EMI Electromagnetic Interference

HAL Hardware Abstraction Layer

IC Integrated Circuit

IDE Integrated Development Environment

IP Ingress Protection

LDO Low-Dropout Regulator

LED Light Emitting Diode

MCU Microcontroller Unit

PCB Printer Circuit Board

PWM Pulse Width Modulation

RGB Red Green Blue

RISC Reduced Instruction Set Computer

RTC Real-Time Clock

RTOS Real-Time Operating System

THT Through-Hole Technology

USB Universal Serial Bus

Table of Contents

List of Figures

List of Tables

1 Introduction

Roll-on/Roll-off type ships and ferries date back to 1833 Scotland, when the Monkland and Kirkintilloch Railway used train ferries to move vehicles and carriages across Forth and Clyde Canal. Since then those types of ships have cemented their place in the transportation world. In past few decades, automation in different industry fields have been in a steady rise, but due to the complex nature, the shipping industry has been adopting newer technologies slower than any other field. Around less than 10% of container volume is still being handled by fully automated terminals [1].

Due to the rapid growth in transportation volumes and logistics, the demand for lower shipping costs and faster delivery times have always been in front of innovation. Combined with modern trends in automation, constant efforts have been made to improve control ports and terminals in order to make them more efficient. One example of this growing trend would be Port of Tallinn.

Every year around 10 million passengers are passing through Port of Tallinn. Automating check-ins can significantly reduce traffic in the waiting area by keeping the vehicles in constant movement without any unnecessary stops. As the vehicle drives towards the ship, a camera reads the licence plate, which then checks the collected information with the check-in system. According to that data, the system then can start guiding the vehicle to the correct lane, using Light Emitting Diode (LED) screens and gates, which are fully automated. This optimises the loading procedure, resulting in on-time departures, less dangerous situations and less waiting around. Furthermore the ability to serve more customers leads to increases in company's revenue [2]. This solution could be viewed as a premise to the smart deck project, as it has similar features, such as detecting vehicles and guiding them with LED screens. Combining smart port with smart deck, not only allows the cars to get to ship faster, but also optimizes the entire journey from start to finish, ending with the vehicle parked on the car deck at the right location much quicker.

1.1 Motivation

This thesis was proposed during the Tallink Smart Deck project to research and develop a vehicle detection and guidance solution prototype for Tallink Megastar car deck. Figure 1 part (a) depicts how the current loading and unloading system works. To this day Helmsmen and dock workers are used to guide the vehicles on car deck, which deck to choose, which lane to take and where exactly to park their vehicle. Part (b) of that same figure depicts the main goal of the smart deck system, where the electronic guiding system detects, tracks and guides the vehicles on the car deck to their designated position during the loading process without any human interaction.

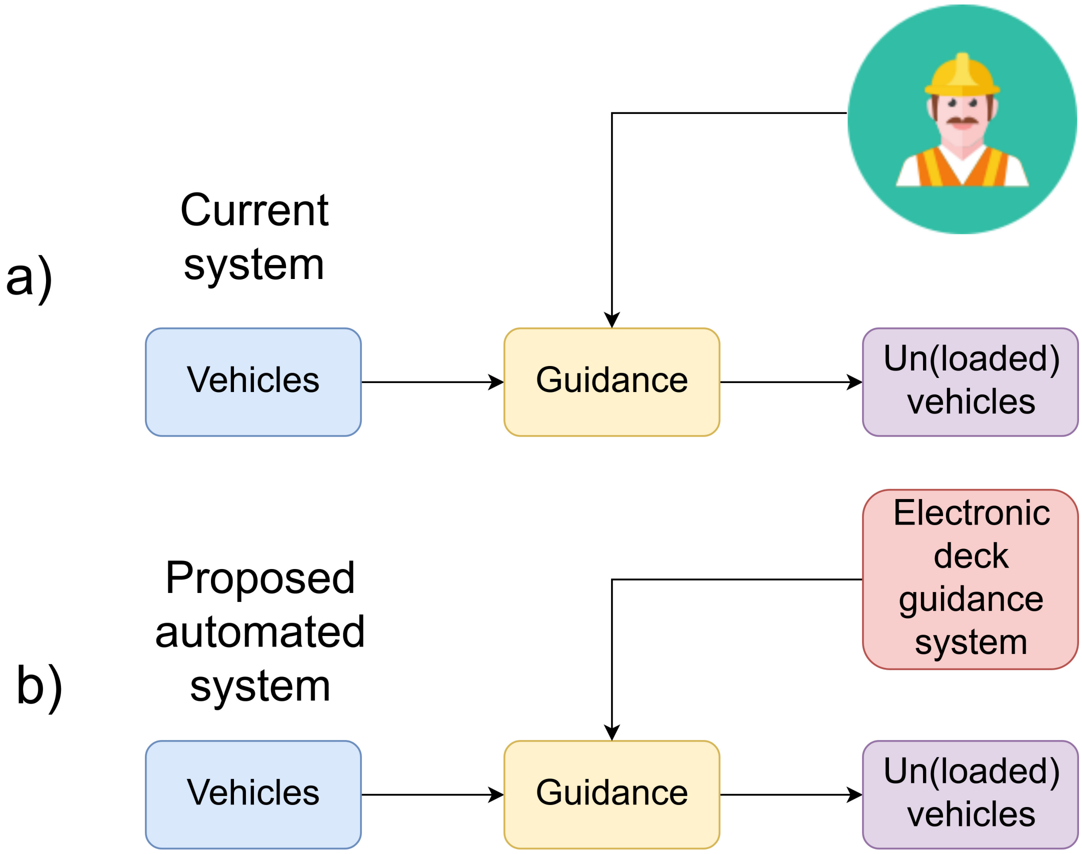


Figure 1. Current vehicle loading current system (a) vs proposed automated system (b).

It should be noted that the project is still in development phase and the final set of features may change.

1.2 Objectives

The goal of this thesis is to analyse possible solutions, develop and deploy a proposed solution for the smart deck system. In order to achieve this, the topics are divided into multiple tasks as following:

1. Analysing requirements, system's architectural solution
2. Developing deck sensor and deck controller prototypes
3. Deploying the prototype system onto ship car deck
4. Test and verify sensor data communication protocol reliability
5. Vehicle guidance and tracking algorithms

1.3 Thesis organization

The following thesis is divided into seven chapters as follows:

Chapter 1 is about introduction, motivations behind the thesis, objectives, and thesis structure.

Chapter 2 gives an overview of the theoretical background about the Tallink Smart Deck Project, challenges and criterias that must be overcome, comparisons with available systems, area of operation and conditions.

Chapter 3 gives an overview of the overall system design, subsystems and developing and manufacturing the prototype.

Chapter 4 describes the system integration with the ship.

Chapter 5 lays out the testing methodology and procedures.

Chapter 6 describes the project status and future work.

Chapter 7 concludes the thesis.

2 Background

The chapter's goal is to give an overview about theoretical backgrounds. First section gives a detailed description about the Tallink smart deck project. Then it focuses on requirements, set by Tallink. After that a brief analysis is given about state of the art solution and area of operations. Finally it touches about conditions, where the system has to operate and how to protect it from them.

2.1 Tallink smart car deck project

Tallink Smart Car Deck project is part of Smart Shipyard concept that is being developed and researched at Tallinn University of Technology in cooperation with Tallink and Estonian Maritime Academy. Development team consists over 10 active members, including people from different fields [3]. The project began in January 2019 and continues till end of 2020. The goal of the project is to develop a system that extends the smart port solution for ship's car deck so that it could decrease the loading times [4]. In the scope of this, there are total of four problems that need to be resolved:

- Automatic traffic flow handling
- Efficient cargo plan usage
- Passenger guidance to the right car deck and location of their vehicle
- Automatic ship docking solution

The thesis author's aim is to propose and develop an vehicle detection solution for the car deck. The proposed system would consist of matrix of sensors, that are placed on the support beams in order to cover all car decks and lanes, which then would be able to detect the vehicles and identify their types. Guiding vehicles to their designated positions would be solved by using indicator lights.

In addition, LED billboards are used to give directions and signals to the driver. Those are placed near the entrances, where the lanes begin. The final goal of the system, would be that the sensors would act together as a hive, which provides information in real-time about the current situation across all the car decks.

2.2 Project requirements

Before any development started, requirements, given by Tallink that must be taken into consideration when proposing a possible solution and developing system:

- Sensors should detect vehicle on any car deck, on any car lane and identify its type.
- Using a sensor, a the system must be able to detect moving vehicles on the deck, that are moving at velocity from 0 to 40 km/h. In addition, the sensor must identify the vehicle's type.
- Detecting sensor must have indicator to give the driver directions and information, where the car should go and when and where to stop. Those lights will be used to guide the driver to the exact location on the car deck.
- The system must operate on a working ship. Also it must be able to stand against any rough weather conditions, mechanical and electrical interference, that will be covered in the following chapter.
- The system's overall power consumption must be kept low as possible. As the number of required sensors might be as high as few hundred for each car deck. Furthermore, the system must be at low power mode during sea travel and switch to normal operation mode before the loading and unloading process begins.
- Nothing cannot be placed on the car deck floor.

2.3 State of the art solutions

In the following section the author will give an overview about current similar commercially available state of the art solutions in parking and also solutions that are currently researched which might give some advice and motivation when developing the smart car deck sensor network.

Comparing the proposed system to similar commercially available systems, there are no similar systems that have been made for ships. The closest available systems are automated parking garages, that can tell how many free parking spots are available and exactly where they are. Two examples are T1 mall parking building and Ülemiste City smart parking system.

Ülemiste City smart parking can be viewed as a state of the art parking facility, that uses induction based ground sensors to detect free or occupied parking spots. It also uses security cameras in collaboration with Artificial Intelligence (AI). This allows the system to constantly learn and improve, to increase its responsiveness and reliability under harsh weather conditions such as snow, fog and rain. Guiding, giving additional information and feedback to the drivers, is done by using LED screens. In addition the customer can remotely check, using Google Maps application, to check if there are any open positions in specific parking lot [5].

Compared to Ülemiste City parking, T1 mall parking system is bit less complex, as it main uses ultrasonic sensors and LED boards. The system detects if the parking place is free or not with ultrasonic sensor, that is placed right over the spot, instead of in ground. To make it easy for the driver to notice if the spot is empty or not, a Red Green Blue (RGB) LED is mounted right next to the sensor, which will glow red if it is occupied and green if it is available. Cameras and LED billboards are used to track, guide and give feedback to the driver. For example how many free parking spots are there on a certain floor. Those things together make it very comfortable and easy for the driver to find a suitable parking space [6].

Those two examples are based on a similar concept, that is used around the world in smart parking buildings. Using ultrasonic sensors shown in Figure 2 are very common in automated parking garages.

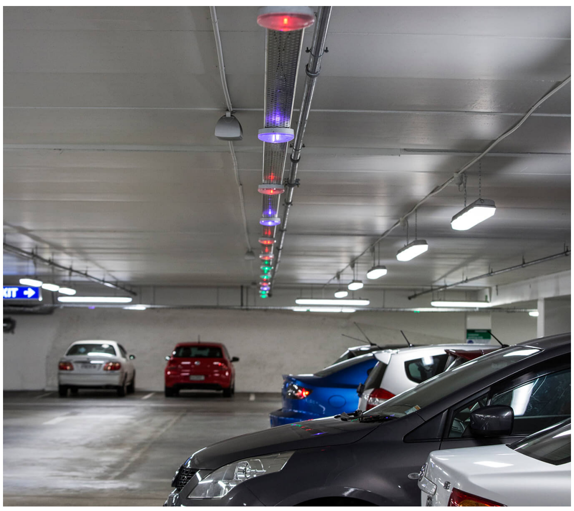


Figure 2. Ultrasonic sensor based parking lot with LED indicator [7].

Induction sensors shown in Figure 3 are also very common, usually used in outdoor parking lots, as they are very weather resistant. In addition both solutions used LED screen to guide the driver.

Figure 3. Ground based detection sensor [7].

Focusing on previous research done on detecting and tracking, there are plenty of examples. Group of researchers from University of Rochester in 2011 published a paper, where described tracking using array of ultrasonic sensors. They concluded that the solution, while being very robust, was performing very well with great accuracy. Furthermore they stated that this solution could find its way into many applications and places in the real world, such as nursing homes, stores, incarceration facilities and many more [8].

As with autonomous vehicles, ships also have been getting more and more attention, specially surface ships, thanks to rapid technological leaps in AI, communications and Augmented Reality (AR). Despite all of this, the authors still think that fully automated ships are far away and human interactions shall remain a part for years or even decades to come. Main reason for this is due to maturity of technology, regulation, laws and risks that still need to be mitigated [9]. For all of this, the author concludes that the proposed system could take some ideas from the parking building. Ultrasonic sensors could be considered one option that would be used to detect moving vehicles. Another option based on those two examples would be camera. Induction sensors are excluded because, as stated in the requirements, nothing cant be placed on the car deck floor.

2.4 Area of operation

This section will cover the details about planned sensor area of operation. In total there are twelve decks from which four of them are car decks. Amount of parking area varies from deck to deck with deck 5 having most space and deck 6 with least space for parking. Furthermore, some decks have areas in which vehicles cannot park as they are for maintenance or emergencies. It must be also taken into account that on car decks 7 and 5 there are special gaps, where vehicles can park on special conditions. Amount of parking area on each car deck is represented in Table 1. Those values presented are based on Tallink Megastar car decks.

Table 1. Tallink Megastar car decks parking area.

Deck	Parking Area	Extra Area
3	3907.05 m^2	-
5	4076.14 m^2	-
6	957.90 m^2	55.35 m^2 for motorcycles and bicycles
7	3200.47 m^2	-

2.5 Weather and environment conditions

In this section a detailed overview will be given about the working conditions, where the system must operate, what are main factors that could endanger the system's operation and how to mitigate those factors. The system must work in a outside environment, where the temperature change from -30 to + 40 °C. Main factors to consider when designing a system are as follows:

- Temperature - High temperatures can have a significant effect on how components will work and most likely will shorten their lifespan considerably, because the electronics are placed in a sealed environment without any air intake nor exhaust. Those problems can also occur when the system is in a very cold environment.
- Water and humidity - This is one of the most crucial aspects, when designing and developing a system. All the electronics must be protected against any water and humidity, which can cause electrical interference during operation or even may render the device unusable.

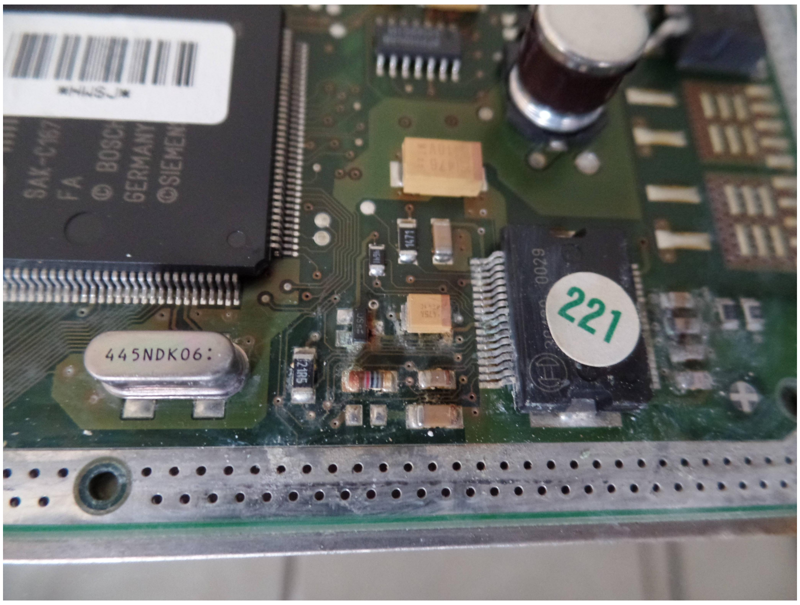


Figure 4. Water damage to the PCB and corrosion [10].

- Salty air - Highly dangerous to electronics due to Sodium chloride (NaCl), which can create chemical bonds with the surface and over time corrodes the surface as shown on Figure 5. This can result in faulty equipment, damaged components and circuits.

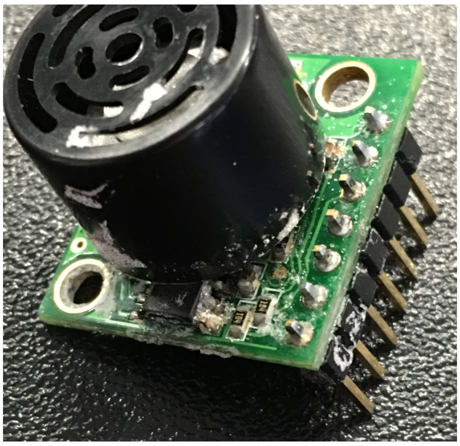


Figure 5. Salt damage to the PCB and corrosion [11].

- Physical damage - On some car decks the distance between the support beam and the vehicle is very narrow, visualized in Figure 6, could be even as low as 10 millimeters, resulting in possible situations, where the roof of the car might hit and damage the deck sensor.

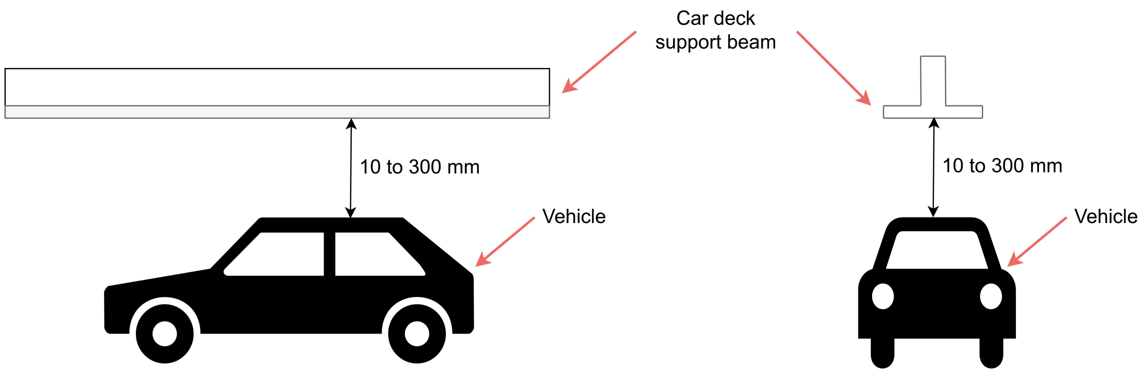


Figure 6. Distance between vehicle deck support beam from side and front view.

- Vibrations - Constant vibrations coming from the ship can leave cracks and other damages over time to the chassis or to the Printer Circuit Board (PCB).

Overall, the biggest problems here are vibrations, humidity and water. Water and humidity getting into the chassis and near electronics must be completely excluded. For example, if even a small drop of water reaches the PCB, it can cause short-circuits, which in return could render some component inoperable temporary or even irretrievably. In a lighter case, the short-circuit might only fry a component, which must be then replaced. Water caused damages must be minimized as much as possible, because if the system is offline even for short amount of time, it can cause unexpected and delays problems during loading and unloading process. In addition, longtime exposure to light humidity, can bring irreversible damages to the PCB as shown in Figure 4.

2.6 IP rating

The previous section talked about different weather conditions and how they affect system and its components, as it can result major damages if not properly protected. To have a better directions when designing a system that is protected against those factors, that were previously listed, that can occur on the ships deck. To counter those challenges, a guide or standard must be followed. For those reasons, this section, gives a brief overview about the IP ratings.

IP stands for Ingress Protection, which is sometimes referred as International Protection Rating, is used to describe enclosures effectiveness against foreign objects, such as small particles, dirt, tools and moisture. These ratings are defined in international standard EN 60529 (British BS EN 60529:1992, European IEC 60509:1989. Originally the standard was used to battle with vague marketing terms such as "waterproof" and to give the consumer more understandable and detailed explanation about device's resistance against outside factors. Usually the IP rating is presented with letters IP in front of the two digits [12].

The first digit indicates the level of protection that the enclosure provides against hazardous parts (e.g., electrical conductors, moving parts) and the ingress of solid foreign objects. Second digit indicates the protection against water and moisture. Both IP Rating digits details are brought out in more detail in Table 2 and Table 3.

Table 2. IP Rating 1st Digit [12].

Level	**Object size protected against**	**Effective against**
0	Not protected	No protection against contact and ingress of objects
1	>50mm	Any large surface of the body, such as the back of the hand, but no protection against deliberate contact with a body part.
2	>12.5mm	Fingers or similar objects.
3	>2.5mm	Tools, thick wires, etc.
4	>1mm	Most wires, screws, etc.
5	Dust Protected	Ingress of dust is not entirely prevented, but it must not enter in sufficient quantity to interfere with the satisfactory operation of the equipment; complete protection against contact.
6	Dust Tight	No ingress of dust; complete protection against contact.

Table 3. IP Rating 2nd Digit [12].

Level	**Object size protected against**	**Effective against**
0	Not protected	-
1	Dripping water	Dripping water (vertically falling drops) shall have no harmful effect.
2	Dripping water when tilted up to 15°	Vertically dripping water shall have no harmful effect when the enclosure is tilted at an angle up to 15° from its normal position.
3	Spraying water	Water falling as a spray at any angle up to 60° from the vertical shall have no harmful effect.
4	Splashing water	Water splashing against the enclosure from any direction shall have no harmful effect.
5	Water jets	Water projected by a nozzle (6.3mm) against enclosure from any direction shall have no harmful effects.
6	Powerful water jets	Water projected in powerful jets (12.5mm nozzle) against the enclosure from any direction shall have no harmful effects.
7	Immersion up to 1m	Ingress of water in harmful quantity shall not be possible when the enclosure is immersed in water under defined conditions of pressure and time (up to 1 m of submersion).
8	Immersion beyond 1m	The equipment is suitable for continuous immersion in water under conditions which shall be specified by the manufacturer.

3 System development

The chapter's goal is to give an depth overview about system development, which is divided into multiple sections. First section gives a detailed description about the system requirements. After that an analysis about the system architecture and what played role in making decision during planning. Lastly hardware related topics will be discussed, followed by software and ending with mechanics design.

3.1 System requirements

The purpose of this section is to give an overview about the necessary requirements that the deck sensor, collector and deck controller must comply with.

Deck controller

- Enclosure with IP55 rating
- Ethernet connector
- Input voltage: +7.5...35 VDC
- Communication and power from the same connector

Collector

- Enclosure with IP55 rating
- Input voltage: +7.5...35 VDC
- Communication and power from the same connector

Deck sensor

- Enclosure with IP55 rating
- Sensor with IP55 rating
- Enclosure with easy access to the electronics
- Input voltage: +7.5...35 VDC

- Communication and power from the same connector
- Attachment hook for extra stability and reinforcement

3.2 System architecture

This section's main objective is to give an overview about the system and its subsystems and components. When planning the system architecture, there are aspects that should be taken into account, which will affect some choices. Those aspects are the following:

- System reliability
- Data bandwidth
- Amount of sensors

The most important factor out of the three is the system reliability. Mitigating any problems which can cause system to crash or worse is crucial. Choosing the proper automation communication protocol, which has been used in the industry for a long time would be the best place to begin. Protocol's longevity usually means that any known issues and problems have been ironed out. Comparison of communication protocols is shown in Table 4.

Table 4. CAN Bus, Modbus and Ethernet comparison.

Feature	**CAN Bus**	**Modbus**	**Ethernet**
Maximum number of nodes	127	Up to 247*	2^8 to 2^{16}
Maximum bandwidth	1 Mbits/s	Up to 115 kbits/s	Up to 1 Gbits/s
Number of wires	2 wires	4 wires	8 wires

*Theoretically possible

Modbus and CAN-Bus are both very popular and exceptionally reliable communication protocols. At first glance Modbus theoretically supports more nodes than CAN Bus, but in reality it supports only up to 32 nodes, due to RS-485. Reason for this is rated impedance, rated for 12 Kohm. Only way to increase the maximum number of nodes is by adding an isolated repeater. The second drawback is the baud rate which can go only up to 115 kbits/s, while he maximum baud rate for CAN Bus is 1 Mbits/s, in certain conditions, which will be covered in more detail in the following chapters. Based on those parameters, the communication between the subsystems is done via CAN Bus. With the communication protocol set, the next step is to figure out how the data is sent to the ships server. The easiest and most optimal choice is to use Ethernet, due to its high bandwidth, going up to 1 Gbit/s. For this a deck controller with Ethernet capability is needed. It gets the information from the deck sensor via Controller Area Network (CAN) and transfers that data via Ethernet to the server. Early calculations estimate that even when using the maximum CAN bandwidth, bottlenecks might occur due to the amount of sensor that will be placed onto the ship's car deck. To minimize any bottlenecks and ensure that any data would not get lost or delayed, collectors must be placed between x number of sensors and the deck controller. Figure 17 visualizes the proposed system architecture and as a result the smart deck system is composed from three subsystem which are as follows:

- Deck controller - Deck controller would be responsible for exchanging information between the ship's systems and collectors, to give real time feedback about car decks.
- Collector - Collectors act as a checkpoint and avoid any bottlenecks as their main objective is to gather data from deck sensors and transmit that information to deck controller. In addition the collector controller is handling the deck sensor synchronization.
- Deck sensor - Detects vehicles, measures their height, gives directions and signals to the driver via indigator. Data collected is passed to collector, which is then transferred to the deck controller and ship's systems.

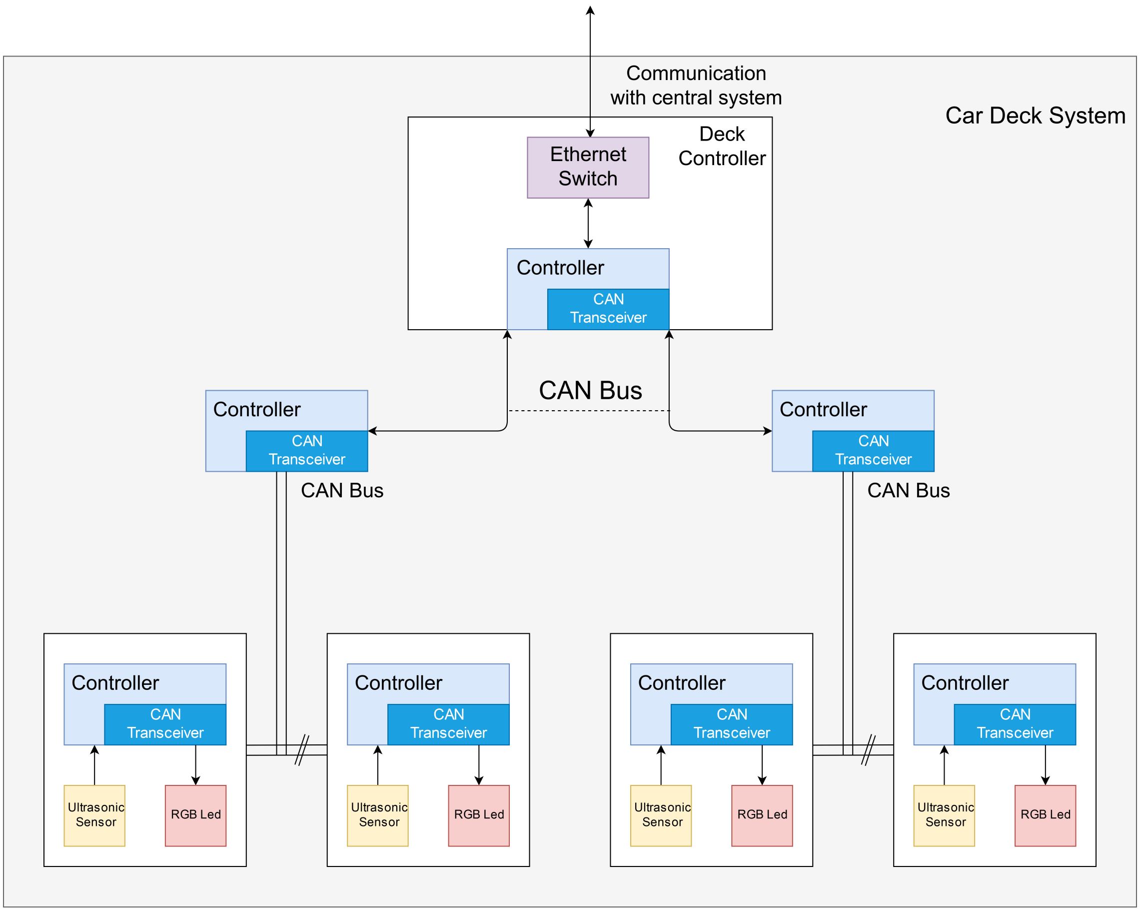


Figure 7. Car deck system architecture.

3.3 Hardware

3.3.1 Hardware architecture

This section gives an overview about the subsystems hardware architecture. Figure 8, which shows first hardware version architecture, was created to test different system components to see which of the would suite the best for the task at hand. Development kits were used because they offer all the hardware needs out of the box, which reduced the development time significantly. Initially, Li-Ion batteries were used to power the system and drive all the components, but input voltage of 3.7 V was not enough to power the development kit and the components, so a boost regulator was added to convert input voltage to 5 V. In addition it also acted as the battery charger. This would later change in Version 2, as CAT6 cables were installed on the car decks. RGB LED chosen for testing, consumed too much current, so a separate LED driver was created with NPN power transistors, that were controlled via Pulse Width Modulation (PWM), which also allowed to tests different brightness levels. Ultrasonic sensors were using Analog-to-digital converter (ADC) to get the measurement value to microcontroller. Finally a special module, which stored all the logged data to a SD Card for later analysis and allowed remote access via Wi-Fi hotspot, made the testing fast and easy, as no downtime was needed to collect any information during device operation. Deck sensor hardware Version 1 features were as follows:

- RGB LED controlling via PWM
- Multiple ultrasonic sensor for measuring
- Store sensor measurement data to SD Card
- Remotely download data from SD Card via Wi-Fi
- Battery powered

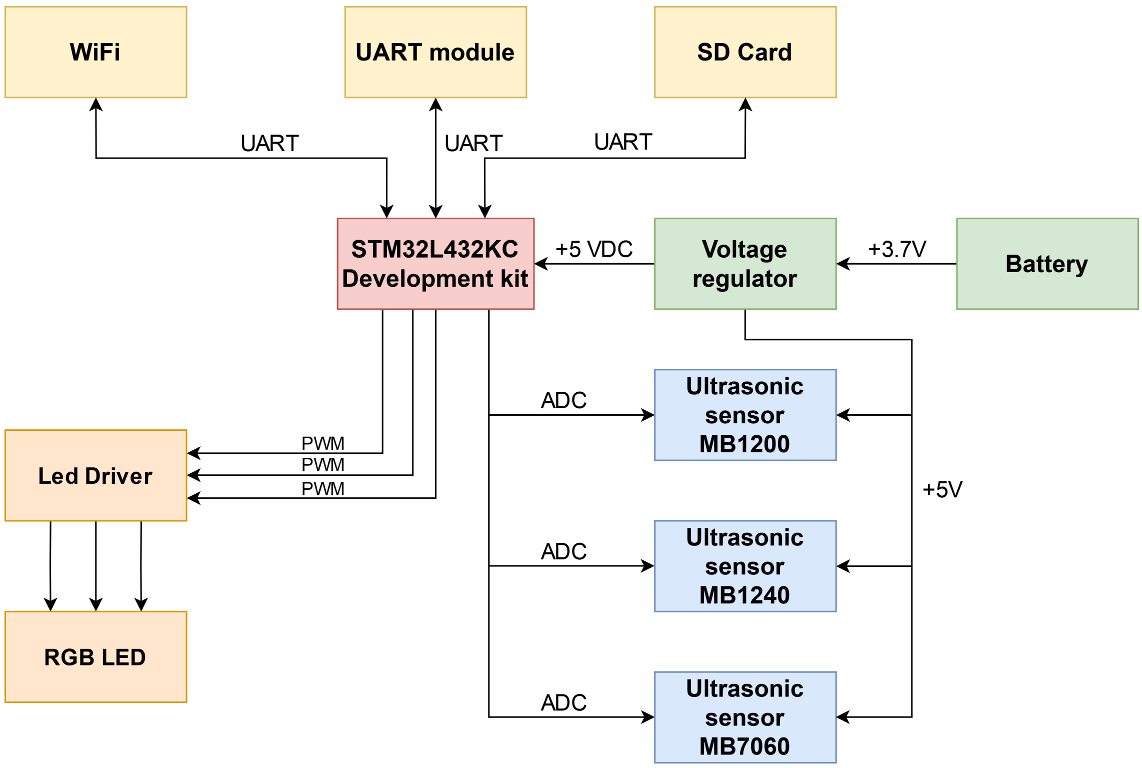


Figure 8. Deck sensor hardware architecture Version 1.

During Version 2 development, represented on Figure 9, number of changes were introduced as following:

- Battery was removed, as CAT6 cables were installed on car deck
- Single ultrasonic sensor
- CAN Bus communication
- Logging module was removed

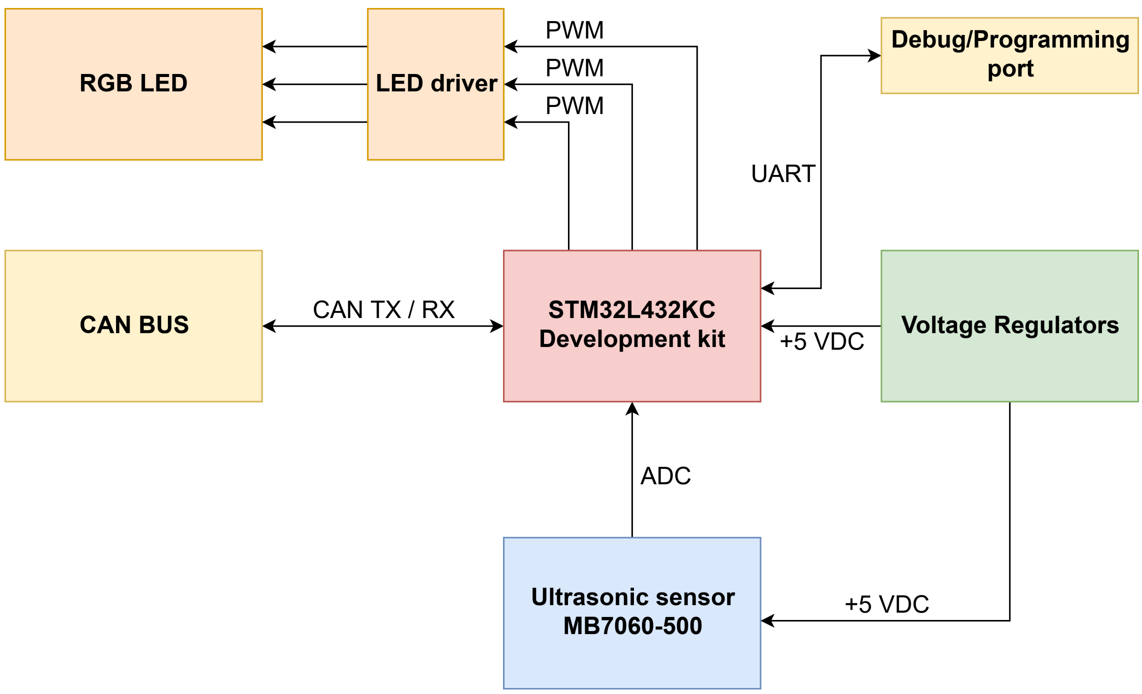


Figure 9. Deck sensor hardware architecture Version 2.

3.3.2 Hardware selection

Choosing Microcontroller Unit (MCU) for all of the subsystems was based on multiple factors:

- Microcontroller must support the following peripherals:
  - CAN - As previously stated, CAN will be used for communicating with other sensors and subsystems
  - UART - Using UART allows the developer to debug and send message via Serial to PC during testing
  - PWM - It is needed for controlling NPN transistors that are used to drive indicator LEDs
  - ADC - To monitor voltage levels and possibly interface with the sensor
- Development tools
  - Configuration tool
  - Open source IDE
- ARM core
- Personal experience

None of the chosen microcontrollers were not specialized industrial controller, meaning that they lack special features such as extreme working environment, self-tests, integrated switchable power supply. Key reasons, when it came to choosing the right microcontroller were core architecture, development tools and past experience with the microcontrollers. Reasons why ARM core was chosen for multiple reasons:

- High performance / Low power consumption ratio
- Support for multiple cores
- Real-time operating system (RTOS)
- 32 bit architecture

In addition the controllers were already available in development boards, which the thesis author was already familiar, as they were used in FS TEAM TALLINN. Having development kits available, meant that no separate PCB had to be manufactured for the MCU. Those two factors shortened the development time significantly, which would have been wasted on finding a alternative solutions and learning all the new tools.

Software development environment was provided by STMicroelectronics for free of charge, which the author was familiar with before from previous projects. For the Integrated Development Environment (IDE), System Workbench toolchain is based on Eclipse [13] which supports all of the STM32 microcontrollers and allows the developer to compile, program and visually debug the program using the ST-Link in-circuit debugger [14], which was included on the development board.

In addition STM32CubeMX [15] was also used to configure all the microcontroller peripherals, clock tree, middleware, calculate current consumption in different operating modes and conditions, using a graphical interface. Those development tools can cut down a significant amount of development time, allowing the developer focus on delivering higher quality software, instead of spending time setting up all the peripherals by hand. All controllers used in the system are 32-bit microcontrollers, using Reduced instruction set computer (RISC) ARM® Cortex®-M4 cores, which are manufactured by STMicroelectronics. The chosen microcontrollers parameters are presented in Table 5:

Table 5. Microcontrollers.

Feature	**STM32L432KC** [16]	**STM32F446RE** [17]	**STM32F767ZI** [18]
Subsystem	Deck sensor	Collector	Deck controller
Core	Cortex-M4	Cortex-M4	Cortex-M7
Flash memory	256 KB	512 KB	2 MB
SRAM	64 KB	128 KB	512 KB
Maximum CPU frequency	80 MHz	180 MHz	216 MHz
Interfaces	CAN, I^2C, SPI, UART, USB, PWM	CAN, I^2C, SPI, UART, USB, PWM	CAN, I^2C, SPI, UART, USB, PWM
Operating voltage:	+1.71...3.6 VDC	+1.71...3.6 VDC	+1.71...3.6 VDC
Package	UFQFPN32	LQFP208	LQFP144

3.3.3 CAN-Bus transceiver

Table 6 represents possible CAN transceiver choices that were considered during development.

Table 6. CAN transceiver comparison.

Feature	**TJA1051** [19]	**SN65HVD233** [20]	**NCV7341** [21]
Manufacturer	ON Semiconductor	Texas Instruments	NXP Semiconductors
Supply voltage	4.5...5V	3...3.6V	4.75...5.25V
Supply current:	60mA	6mA	80mA
Data Rate	5 Mb/s	1 Mb/s	1 Mb/s
Transceivers	1	1	1
Package	HVSON-8	SOIC-8	HVSON-8
Price	1.43€	2.77€	0,93€

Choosing CAN interface came down to reliability, as the device would be working in a harsh environment, has to be shielded from electromagnetic interference. In addition the transceiver must have a low electromagnetic emission (EME) and shielded against Electromagnetic interference (EMI). Based on those conditions the chosen transceiver was TJA1051, which is shown as ready to use module in the Figure 10. Author's previous experience also affected the choice, as they were used in Formula Student for many years, because of their reliability and performance characteristics.

Figure 10. Can Transceiver [22].

3.3.4 Voltage regulator

As the input voltage coming in is too high for some of the components, Low-dropout regulator (LDO) was used to convert input voltage into more suitable levels, that is needed to drive the microcontroller and other peripherals. Furthermore the module has to be able to output enough current that the system consumes. Power dissipation must be taken into consideration to ensure that components can operate within its defined temperature limits. If the voltage regulator operates outside the recommended thermal limits, the normal operation is severely affected, which might even damage the component itself and components connected to it. The difference between the input and output voltage is conjunction with load is energy that is dissipated by the device. To improve heat dissipation, adding a heatsink improves heat dissipation significantly, due to having a larger area for heat to dissipate. Due the systems low current consumption, no heatsink was requiered. Formula of maximum power that the device can dissipate is depicted in Equation 1:

$$PD_{MAX} = \frac{T_{JMAX} - T_A}{O_{JA}} \tag{1}$$

During search for LDO, number of candidates where considered, as presented in Table 7

Table 7. Voltage regulator comparison.

Feature	**L7805CV** [23]	**LT1086** [24]	**L4940** [25]
Manufacturer	STMicroelectronics	Analog Devices	STMicroelectronics
Min input voltage	7V	10V	6V
Max input voltage	35V	25V	30V
Output Current	1A	1.5A	1.5A
Package	TO-220, DPAK	TO-220, DPAK, TO-3	TO-220, DPAK
Price	0.44€	3.70€	1.36€

Based on the parameters, price and availability, L7805CV Integrated Circuit (IC) was chosen, due to supporting wide range on input voltages and has high output current rating, up to 1A which is more than enough, even when the system in under peak load. In addition the IC has an thermal overload protection and short circuit protection. The chosen voltage regulator is depicted in Figure 11, uses TO-220 package, which does not have the best thermal dissipation compared to DPAK package, but is easier to work with during prototyping, due to being Through-hole technology (THT) type component. In a

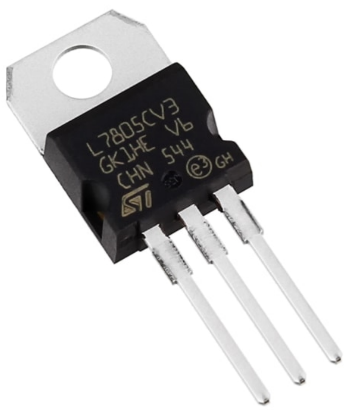


Figure 11. L7805CV regulator [26].

3.4 Car deck sensors

3.4.1 Comparison of sensors

This section will cover the comparison between different sensors, bringing out their advantages and disadvantages. In principle all the compared sensor work based on the time of flight, by outputting a signal and waiting until it is echoed back to the receiver. In the terms for detection and distance sensing, there are parameters that must be taken into consideration, such as:

- Distance: device's minimum and maximum detection range
- Update rate: usually measured in Hz, how many times in second the device can take measurements. Higher the frequency, more measurements can the sensor take
- Resolution: how accurately can the sensor measure and distinguish object from each other

Radar is an acronym for Radio Detection And Ranging and operates on time of flight basis, but instead of using ultrasonic echos to detect objects, it uses electromagnetic waves to identify an object and its parameters. Compared to ultrasonic sensor, radar operation is not affected by environmental conditions, such as temperature, humidity, dust, etc, which makes it a more suitable choice for certain applications compared to other sensors [3]. Radar operating high range could be viewed as a strong advantage, compared to other sensors, but for car deck applications, this feature does not give much weight to radar. Only thing that could manipulate the radar's readings are materials with low dielectric properties. Those materials, such as dry powders, granules, are not very good reflectors and can pass electromagnetic waves.

LiDAR stands for Light Detection And Ranging. It operates by firing laser light at very high frequency, up to 150000 pulses per second. As a result, the device creates a 3D representation of its surroundings. Not only does this give it a very precise accuracy, but also works in long range. This does not come without any drawbacks. The result of LiDAR imaging depends highly on the weather conditions, such as rain, fog and also on the time of day, that affect the reflections. This sensor would not be an ideal candidate, as it method of operation does not benefit in detecting vehicles. Furthermore the sensor would require separate system to process the data and is considerably more expensive than other sensors.

Camera is a sensor, which is based on image recognition, where image is processed to detect object and gives them parameters based on the image. With high resolution imagery, the camera is able to separate finer details that other sensors are not capable of detecting, for example color. Despite having significant advantages there are also number of drawback, that affect camera's operation, such as amount of light, weather and even lens dirtiness. Table 8 depicts sensor parameters, which are based on the ship conditions.

Table 8. Sensor comparison.

Sensor	**Range**	**Refresh rate**	**Resolution**	**Robustness**	**Cost**	**Power consumption**
Ultrasonic	7.5 m	20 Hz	1 cm	High	Very Low	15..300 mW
Radar	20 m	30 Hz	4 cm	Very High	Medium	10 W
Camera	40 m	40 Hz	5 cm	Medium	Medium*	1.25 W
LIDAR	200 m	1000 Hz	1...4 cm	High	High	6...7 W

*Local image processing

The section concludes that, despite not having huge range, very high refresh rate, the ultrasonic sensor, which will be covered in the next section, makes up those disadvantages, by being very resistant to external environmental conditions and being very cheap compared to other distance sensors and having the lowest power consumption which is very critical.

3.4.2 Ultrasonic sensor

In this section, the author will explain how the ultrasonic sensor works, why ultrasonic sensor was the best choice for detecting vehicles and what influences its accuracy during measurement. As mentioned in the beginning, as the name indicates, the ultrasonic sensors are used to measure distance by using ultrasonic sound waves and the time of flight principle. Figure 12 visualizes how the device sends out an ultrasonic pulse, which is then reflected back by the object. The bounced echo is caught by the sensor and it is then converted into an electric signal via the piezoelectric transducer. This is called as the propagation time of sound.

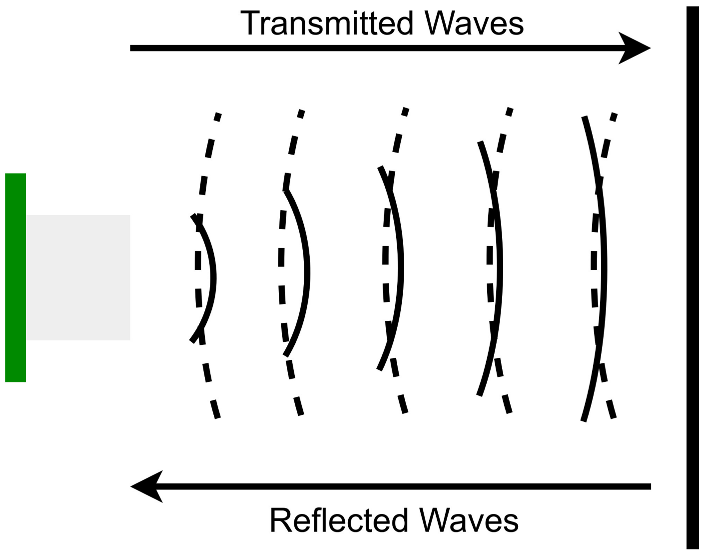


Figure 12. Ultrasonic sensor operation.

The distance is then calculated with a formula depicted Equation 2. The time lag between the emitted echo and the received echo with the help of speed of sound in air at 20 degrees room temperature, which is about 344 m per second.

$$\text{Distance} = \frac{\text{speed of sound} * \text{time of flight}}{2} \tag{2}$$

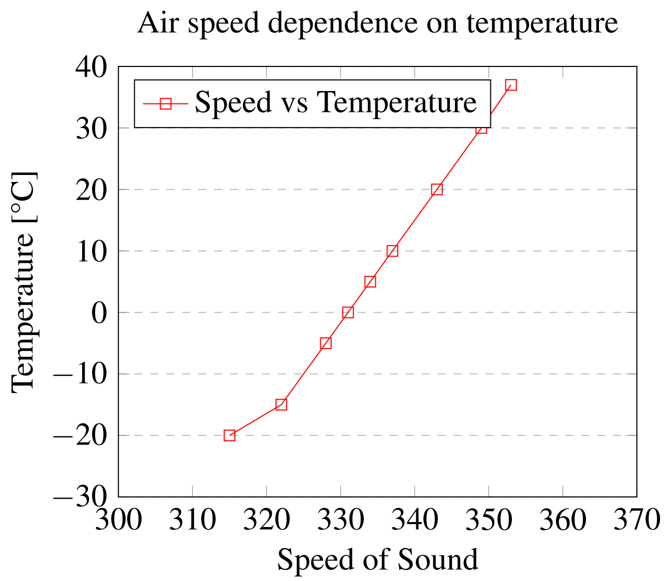
The distance equation is quite simple, the speed of sound is multiplied by the time that took from sending out the pulse and receiving it back. Finally the result is divided by 2 due to the round-trip as the distance is only half of round-trip.

Ultrasonic sound is a vibrations that uses frequency, is above the range of human hearing around 20 Hz to 20 kHz. In audio, microphones and loudspeakers that are used to send and receive ultrasonic sound are called transducers. Most ultrasonic sensor use only a single transducer for sending and receiving ultrasonic waves, which usually operate around 40 kHz to 250 kHz. This section concludes that those properties make the ultrasonic sensor very reliable and robust sensor, as it is almost immune to outside conditions. Next section will explain in more detail how some environmental conditions could affect the accuracy of the sensor.

3.4.3 Speed of sound and environment affection

As mentioned in the previous section, the ultrasonic sensor uses the speed of sound to measure the distance between an object and the sensor. Unfortunately the speed of sound can change to degree and is affect by few environmental factors. This chapter will discuss how the environment can affect sensor's detection capabilities and measurement accuracy. One of the biggest influences on speed of sound is the temperature. At room temperature around 20 °C the speed of sound is roughly around 343 meters per second. The speed of sound can viewed as a form of kinetic energy. Basic chemistry and physics explain that with higher temperature the molecules have more energy, which helps them to vibrate faster. If the temperature drops instead of rising the speed of sound also decreases. Equation 3 depicts how the speed of sound is calculated [27]. The following graph depicts the correlation between speed of sound and temperature.

$$\text{Speed of sound} = 331.3 + 0.6 * T \tag{3}$$



Other conditions such as humidity, altitude and air pressure, have little or no affect at all on the speed of sound.

3.4.4 Selecting ultrasonic sensor

Selecting an ultrasonic sensor was more complicated than expected. The criteria was, that the sensor must be rated at least IP55, which narrowed the search quite a lot. Secondly the sensor must be able to measure as many times as possible, to get as much detection readings as possible. Thirdly, the sensor's range must be at least 5 meters, so it can work on different car decks where the ceiling heights may vary. The three main possible candidates were UM30-212 [28], MB7060 [29] and xx918a3c2m12 [30]. All the chosen candidates are industrial grade ultrasonic sensors. Table 9 depicts all the chosen sensors parameters.

Table 9. Ultrasonic sensor comparison.

Parameter	**UM30-212**	**MB7060**	**XX918A3C2M12**
Working Voltage	+12...30 VDC	+3.3...5 VDC	+10...28 VDC
IP rating	IP65	IP67	IP67
Average current consumption	60mA	3mA	40mA
Beam pattern	Wide	Narrow	Medium
Beam angle	Wide	Narrow	Medium
Min Range	35cm	20cm	40cm
Max Range	600cm	765cm	500cm
Measuring Rate	8Hz	10Hz	10Hz
Interfaces	TTL	ADC, I^2C, RS232, UART	ADC, I^2C, RS232, UART

Based on the selection, the chosen sensor was MB7060, which is depicted in Figure 13, for the following reasons:

- Real-time auto calibration and noise rejection, which allows to automatically filter out any false measurement, that might otherwise trick the sensor, that a vehicle is under, but in reality there is none.

- Very narrow beam
- +3...5.5 VDC supply, which does not require a separate voltage regulator and could be powered by the microcontroller development kit.
- Very low current draw, averaging 3.4 mA
- 10Hz refresh rate, allowing the sensor to measure 10 times per second
- Multiple interfaces, such as analog, I2C, RS232, UART and ADC
- Operates at 42 KHz
- Ranging can be triggered externally or internally
- Different sizes with multiple housing options
- Ability to chain multiple sensors
- Threaded enclosure with IP67 rating

Figure 13. MB7060-500 Ultrasonic sensor [31].

3.4.5 Detecting vehicles with ultrasonic sensor

In this section an detailed overview is given how to the ultrasonic sensor will be used to detect objects. This includes its detection area, number of measurements, detection accuracy. Ultrasonic sensors will be placed on the car deck in a matrix formation. This means that each car lane will have their own row of sensors. Figure 14 visualizes how the sensors are placed from top view.

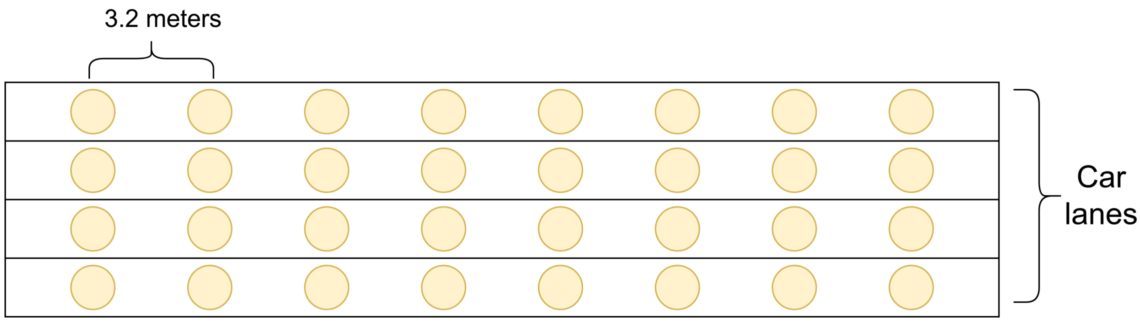


Figure 14. Deck sensors from top view.

Figure 15 depicts when the sensors are placed on the car deck, while the vehicles are moving.

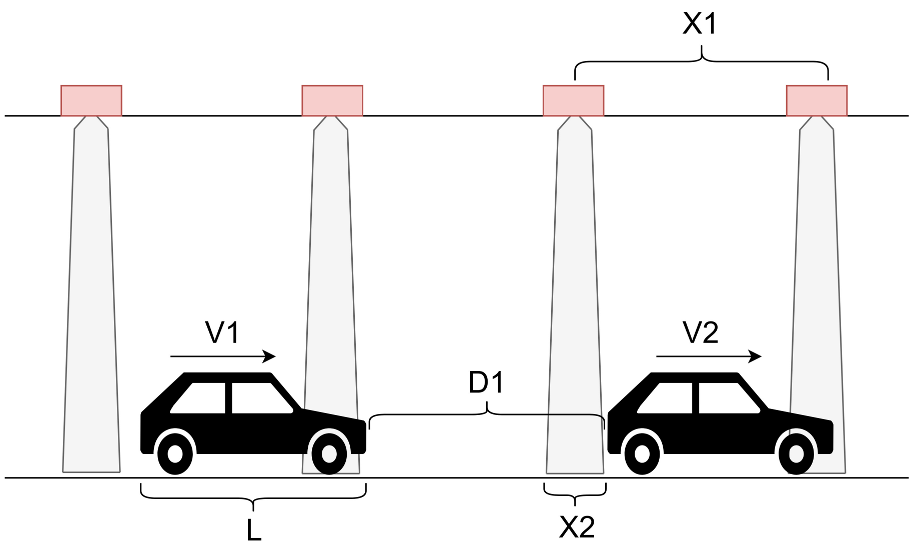


Figure 15. Detection Zones.

Detecting vehicles and number of detection measurements is a bit complicated as there are many factors to be considered:

- Vehicle velocity [V1 and V2]
- Vehicle length [L]
- Distance between vehicles [D1]
- Distance between sensors [X1]
- Detection area [X2]

- Sudden changes in velocity

Factor listed are dynamic that can change at any moment, which will affect how many times can one deck sensor detect a moving vehicles as it is moving. The faster a vehicle is driving, the less detection measurements can be done and vice versa. For example, imagine a vehicle is moving at velocity of 30 km/h, which translates into 8.33 m/s or 83 cm every 100 millisecond. The length of the vehicle is 4.5 meters. Based on the calculation if the sensor measures in every 100 milliseconds, it can detect that car around 5 times. Table 10 represents number of times the deck sensor detects a moving vehicle at different velocities.

Table 10. Number of detection vs vehicle velocity and length.

	Velocity m/s	5 km/h	10 km/h	15 km/h	20 km/h	25 km/h	30 km/h	35 km/h	40 km/h
Vehicle Length	3.2 m	23	12	8	6	5	4	3	3
	4.2 m	30	15	10	8	6	5	4	4
	4.5 m	33	16	11	8	6	5	4	4
	4.8 m	35	17	12	9	7	6	5	4
	5.1 m	37	18	12	9	7	6	5	5
	5.7 m	41	21	14	10	8	7	6	5
	12 m	87	43	29	22	17	14	12	11
	19 m	138	69	46	34	27	23	20	17

This is an ideal scenario. For example if there is a car that is following the first one with a distance that is very small, the deck sensor could detect two vehicles as one. In worst case scenario, the sensor is capable of detecting a vehicle at least 3 times, which should be more than enough to exclude false readings.

3.5 Indicators

This section will an overview about the indicators that are used to guide the passengers on the car deck. Best and most optimal choice would be to use LED lights. LED stands for light emitting diode, which is a type of diode that converts energy into visible light. They are very compact, available in many different packages, some with lenses or without them.

3.5.1 LED selection

When selecting the right LED for deck sensor that guides the drivers, the most important features to look for are the following:

- Brightness - Brighter lights are easier to notice at longer distances, as one car deck lane can be hew hundred meters long.
- Viewing angle - Higher the viewing angle the better is to see it from different angles. This is visualized in Figure 16, where a LED is placed near the support, while a car is moving towards it.

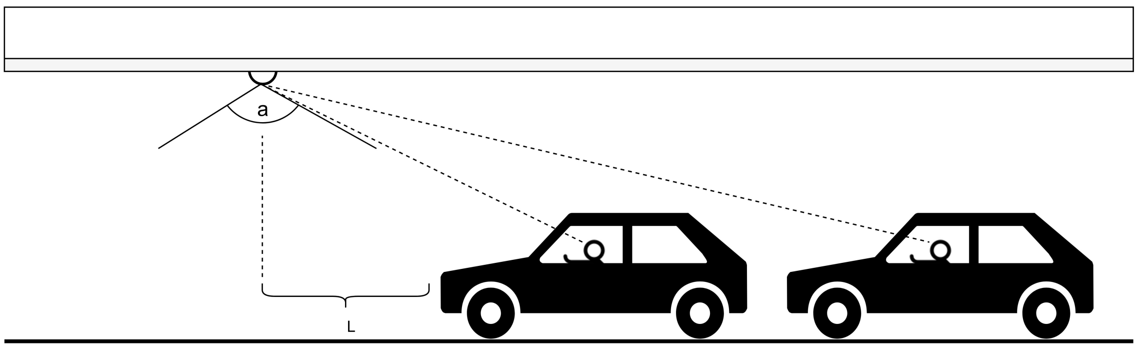


Figure 16. LED spotting distance vs LED angle.

- RGB option - Ability to cast different colors to give directions and information to the driver is mandatory. For example the colors in the list can be used to give the following information to drive or the technician. Last two colors will be displayed only during maintenance and not during loading and unloading.
  - Red - Stop / Busy
  - Green - Go / Free
  - Yellow - General Error

- Blue - Network error

- Current consumption - As the requirement states that the current consumption should be as low as possible. It should be considered that lower the amount current running through the diode, could reduce the brightness significantly.

Based on those criterias, number of possible candidates were chosen, that are brought out in Table 11 with their parameters. Picking out the best option, by only relying on just parameters would be impossible and to rather test them in different conditions, covered in later chapters.

Table 11. LED comparison.

Model	**Viewing angle**	**Current consumption**	**Brightness (mcd)**
SML-LX1610RGBW/A [32]	110	350 mA	2500/2500/800
XMLCTW-A2-0000-00C2AAAB1 [33]	130	350 mA	4500/8700/1300
SML-LX5050SIUPGUBC [34]	120	150 mA	4000/9000/2300
AAAF5051-05 [35]	120	150 mA	2300/9000/2300

3.6 Software

3.6.1 Software communication architecture

Figure represented in Appendix 1 visualizes the communication between different subsystems, which subsystem uses which algorithm and what communication protocols they use.

3.6.2 Deck sensor software architecture

This section will give an overview about the software architecture, describing what data is exchanged between different subsystems. Figure 17 depicts the deck sensor software architecture showing the overall scope of the software that needs to be written and provide a means of analysis about the systems behavior, before any software development was done.

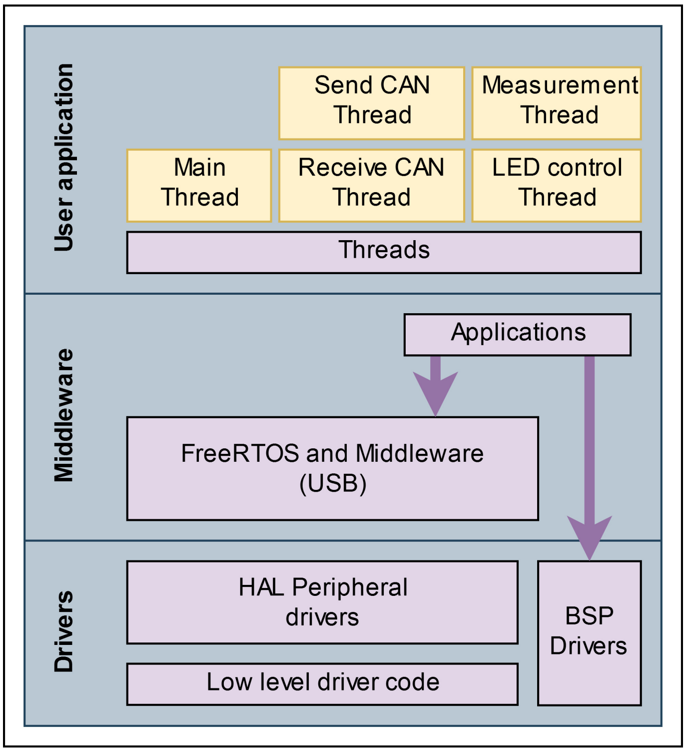


Figure 17. Deck sensor software architecture.

Before any code can was written, it should be taken into consideration which drivers, libraries and middleware to use and how the tasks should be divided.
In addition some system requirements should be also considered:

- Real time operation
- Modularity
- Running multiple tasks at the same time

Firstly microcontroller runs program in loops, meaning that the next task cannot be executed before the previous task is finished. In real time system it is unacceptable as it causes delays and problems, when necessary task could not be executed on time. Secondly a conventional processor or a microcontroller can only execute a single task at a time. Using rapid switching between different task, would make it look like as the system is running multiple tasks at the same time. Figure 18 visualizes this difference.

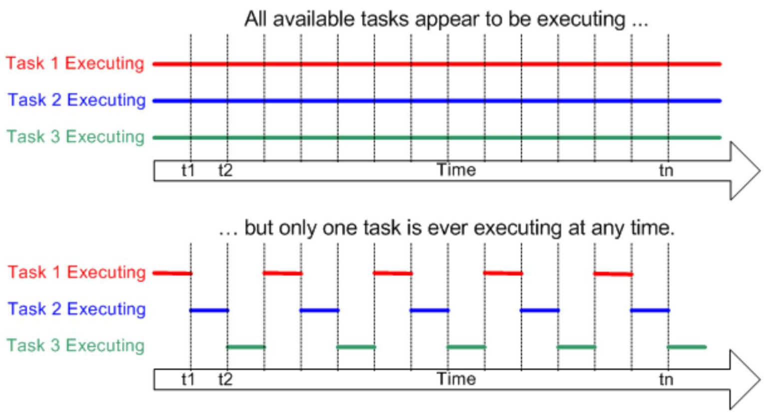


Figure 18. Multitasking vs concurrency [36].

This method has multiple perks as such:

- The multitasking and inter-task communications features of the operating system allow the complex application to be partitioned into a set of smaller and more manageable tasks.
- The partitioning can result in easier software testing, work breakdown within teams, and code reuse.
- Complex timing and sequencing details can be removed from the application code and become the responsibility of the operating system [36].

These issues can be solved by using a Real Time Operating System (RTOS) version called FreeRTOS [37], which was specifically designed for microcontrollers. As each thread or task is independent from another, it is difficult to predict, which task to call next. This is solved by using a operating system kernel, the system can decide that based on the task priority. For every task running in the deck sensor, a separate thread was created.

Hardware abstraction layer (HAL) allows the developer to write software much faster, as most of the functions that are used to control different peripherals is already available. As an result total of 5 threads were created as following:

- Main thread
- Send CAN thread
- Receive CAN thread
- Measurement thread
- LED control thread

3.6.3 Deck sensor CAN messages

In this section, an overview is given about which CAN messages are sent by the deck sensors. Each CAN message can hold maximum 8 bytes of data. Every variable is using unsigned 8-bit integer type called uint8_t.
Measurement message as depicted in Table 12, contains information about which type of sensor was used, measurement result, measurement timestamp, LED status and error notifications.

Table 12. Measurement message.

Message	**Length**	**Data type**
Message_type	4 bits	uint8_t
Sensor_type	4 bits	uint8_t
Errors	2 bits	uint8_t
Level	6 bits	uint8_t
Time	4 bytes	uint8_t
LED_status	1 byte	uint8_t
Padding	1 byte	uint8_t

Firmware and CRC message is represented in Table 13, contains data about the firmware versions, which the deck sensor uses. Each sensor has total of three firmware versions:

- Factory Firmware
- Firmware version 1
- Firmware version 2

In addition to the firmware values, Cyclic redundancy check (CRC) code is used to detect errors in raw data.

Table 13. CRC message.

Message	**Length**	**Data type**
Message_type	4 bits	uint8_t
Sensor_type	4 bits	uint8_t
FW1_CRC	2 bits	uint8_t
FW1_Ver	6 bits	uint8_t
FW2_CRC	2 bits	uint8_t
FW2_Ver	6 bits	uint8_t
FWF_CRC	6 bits	uint8_t
FWF_Ver	6 bits	uint8_t
Padding	6 bits	uint8_t

Diagnostic message represented in the Table 14 which gives feedback about the deck sensor parameters. These parameters include which hardware and firmware versions are used. In addition what type of sensor is it using and what is the incoming input voltage, that powers the deck sensor and its components.

Table 14. Diagnostic message.

Message	**Length**	**Data type**
Message_type	4 bits	uint8_t
Sensor_type	4 bits	uint8_t
HW_Ver	1 byte	uint8_t
FW_Ver	1 byte	uint8_t
Sensor	1 byte	uint8_t
Voltage	1 byte	uint8_t
Padding	3 bytes	uint8_t

Configuration message represented in the Table 15 is used to characterize how the system is operating. Whether the system in sleep mode, normal operation or in flash mode. Furthermore it can give information about the indicator, if it is switched off or showing a particular color.

Table 15. Configuration message.

Message	**Length**	**Data type**
Message_type	4 bits	uint8_t
Sensor_type	4 bits	uint8_t
LED_Brightness	1 byte	uint8_t
Sensor_Operation	1 byte	uint8_t
Padding	4 bytes	uint8_t

Clock message, which is shown in Table 16, is used to give feedback about deck sensors RTC.

Table 16. Clock message.

Message	Length	Data type
Message_type	4 bits	uint8_t
Sensor_type	4 bits	uint8_t
Time	4 bytes	uint8_t
Padding	3 bytes	uint8_t

3.6.4 Deck controller software architecture

Similar to deck sensor, the software architecture side, which is depicted in Figure 19, remains very similar, as it also uses HAL libraries and FreeRTOS. Only key difference is the amount of used threads, limited to just two. CAN transmission and CAN receive threads that are used to exchange data between the collectors, deck sensors and ship's server.

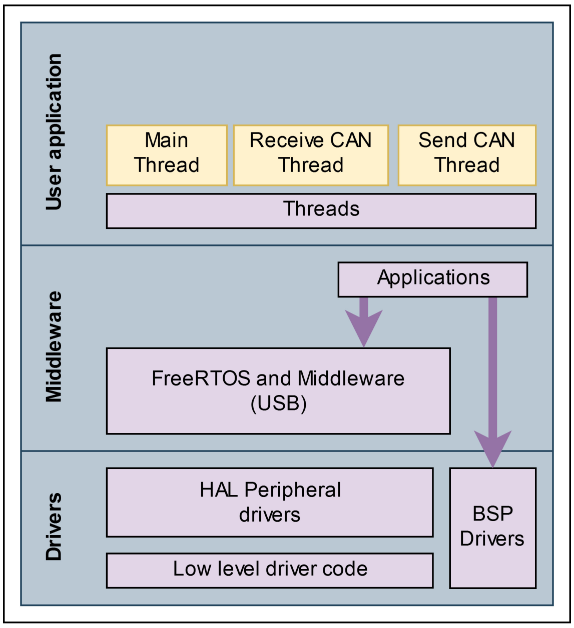


Figure 19. Deck controller software architecture.

3.6.5 Deck controller CAN messages

Messages sent out by the deck controller are similar to deck sensor's messages. Tables 17, 18, 19, 20, 21 represent what type of data is in the frame, the length and datatype of that information.

Table 17. Measurement message.

Message	**Length**	**Data type**
Message_type	4 bits	uint8_t
Errors	2 bits	uint8_t
Level	6 bits	uint8_t
Time	4 bytes	uint8_t
LED_status	1 byte	uint8_t
Padding	1 byte	uint8_t

Table 18. CRC message.

Message	**Length**	**Data type**
Message_type	4 bits	uint8_t
Sensor_type	4 bits	uint8_t
FW1_CRC	2 bits	uint8_t
FW1_Ver	6 bits	uint8_t
FW2_CRC	2 bits	uint8_t
FW2_Ver	6 bits	uint8_t
FWF_CRC	6 bits	uint8_t
FWF_Ver	6 bits	uint8_t
Padding	6 bits	uint8_t

Table 19. Diagnostic message.

Message	Length	Data type
Message_type	4 bits	uint8_t
Sensor_type	4 bits	uint8_t
HW_Ver	1 byte	uint8_t
FW_Ver	1 byte	uint8_t
Sensor	1 byte	uint8_t
Voltage	1 byte	uint8_t
Padding	3 bytes	uint8_t

Table 20. Configuration message.

Message	Length	Data type
Message_type	4 bits	uint8_t
Sensor_type	4 bits	uint8_t
LED_Brightness	1 byte	uint8_t
Sensor_Operation	1 byte	uint8_t
Padding	4 bytes	uint8_t

Table 21. Clock message.

Message	Length	Data type
Message_type	4 bits	uint8_t
Sensor_type	4 bits	uint8_t
Time	4 bytes	uint8_t
Padding	3 bytes	uint8_t

3.6.6 Sensor calibration algorithm

Figure 21 depicts the flow of the calibration algorithm. When the system is turned on, it first initializes peripherals. After that is completed, the sensor will take ten measurements. During the measurement the algorithm will calculate the average distance, if the measurement drastically changes the average height, it is then considered was a extreme value, indicating an error during measurement. The value is then removed. If all the ten measurements have been completed, the final average distance is calculated. Then results is also stored into memory, which will be used as a reference during the next re-calibration to detect any errors.

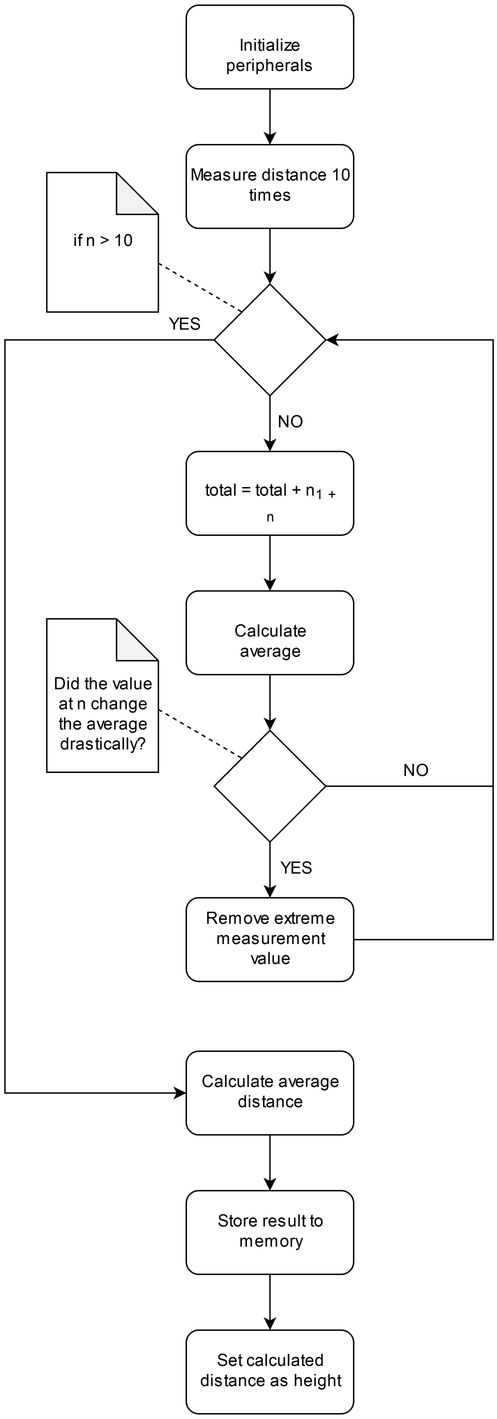


Figure 20. Sensor calibration algorithm.

3.6.7 Vehicle detection algorithm

Figure 21 portrays vehicle detection algorithm. When the system takes a measurement, it changes the RGB Led color to green if the sensor did not detect any moving vehicles. In case of detection the LED turns red. The measurement data, containing the value and timestamp, is then sent to the ship's server. Right after that it receives information from the ships systems, for example if the loading has been finished or not. If the loading has be completed, the system enters sleep mode, waking up for time to time to check if the ship has reached the port. When that condition is true, the system will exit sleep mode and continues to operate in normal mode.

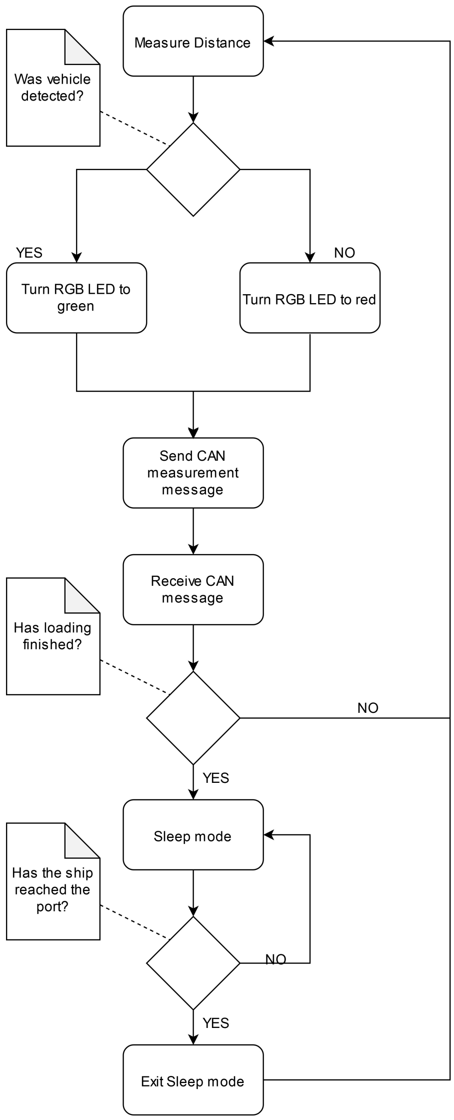


Figure 21. Vehicle Detection algorithm.

3.6.8 Sensor time synchronization algorithm

Figure 22 visualizes how the deck system is in sync with the ships systems and deck systems themselves are synchronized with each other. First the deck controller has to ask the server for a Real-time clock (RTC) value. When the value is received the deck controller set that value as its RTC. After both the server and deck controller have verified the RTC values, the deck controller send RTC values to all the collectors that are connected to it. This ensures that all the subsystems receive the same RTC value step by step. This step is then repeated between deck sensors and collector, latter sending its RTC value to the deck sensor. In case even a single sensor is not in sync with others, the synchronization being from the deck controller.

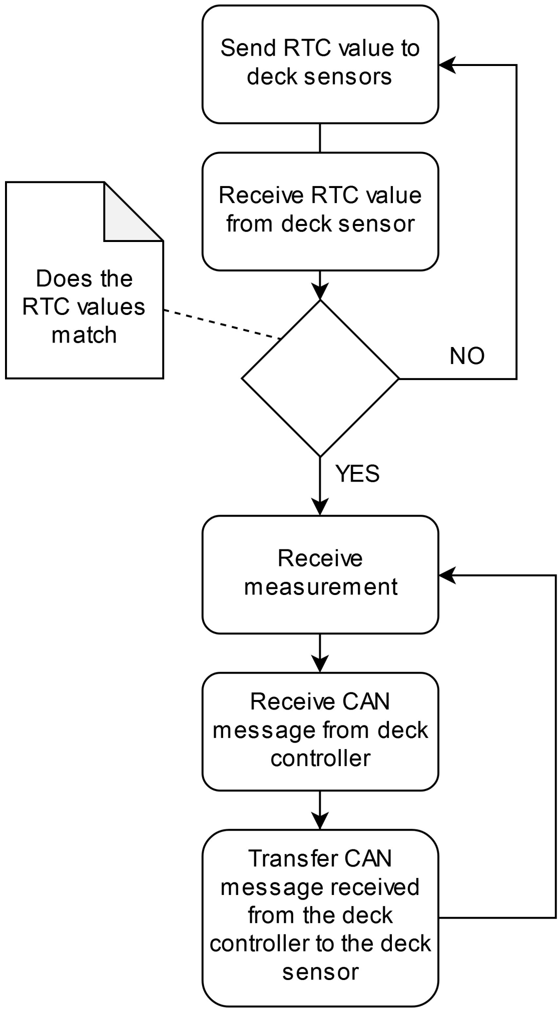


Figure 22. Sensor synchronization algorithm.

3.6.9 Vehicle guiding algorithm

Guiding algorithm, which is depicted in Figure 23 first receives measurement result from the deck sensor via CAN Bus. In case the messages contains a message, detailing that a vehicles not detected, it waits for a another message. If a vehicle was detect and the detection location was the final location The ships system are the notified. If it was a wrong vehicle, the system recalculates all the future placements and that information is then sent to the deck sensors to reroute the moving vehicles, based on the new layout.

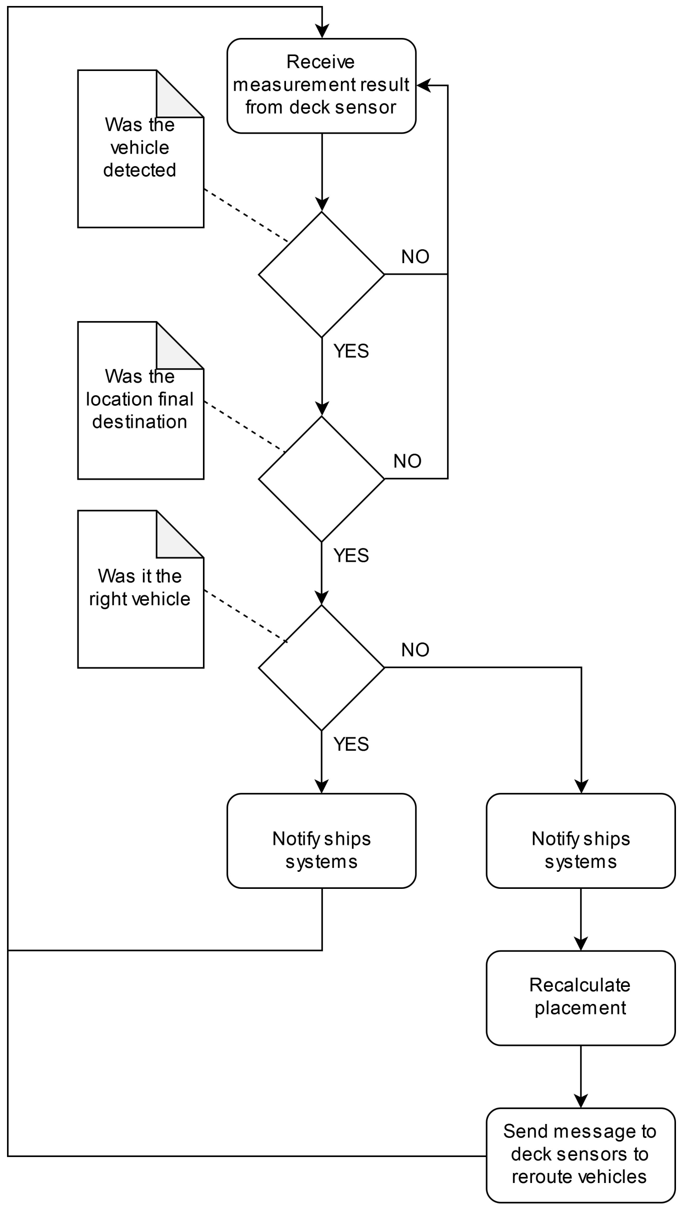


Figure 23. Vehicle guiding algorithm.

3.7 Mechanics

This section gives an overview of chassis design requirements, design choices, manufacturing options and materials. As previously stated in the last chapter, the system will be placed in rough climate environment, which might damage the components if not properly protected from external conditions. In addition to designing enclosure that can withstand temperature fluctuation, humidity, shock and more, the construction must include a bracket or method that is used to secure the device in the ship. In summary, conditions that must be met are as follows:

- IP55 rating
- Durable
- Must not conduct electricity
- Stand shocks and vibrations
- Lightweight
- Easy and cheap to produce
- Easy access to the electronics, in case of maintenance
- Locking mechanism

3.7.1 Case design

Considering all the criterias, developing only one chassis design was not possible due differences in the support beam dimension, where the deck sensors will be placed. Software used to design Computer-aided design (CAD) models of chassis was created with special software called Solid Edge [38], which not only allows the designer to create all the necessary drawings, models, but also to simulate them in a virtual environment, against different conditions, such as temperature fluctuations, forces, wind velocity etc.

As the indicator light must visible from every corner, a special cavity was designed on the bottom, shown in Figure 24 (D), so that the LED would be parallel with the bottom. This also allowed the LED to be removed with much ease during testing.

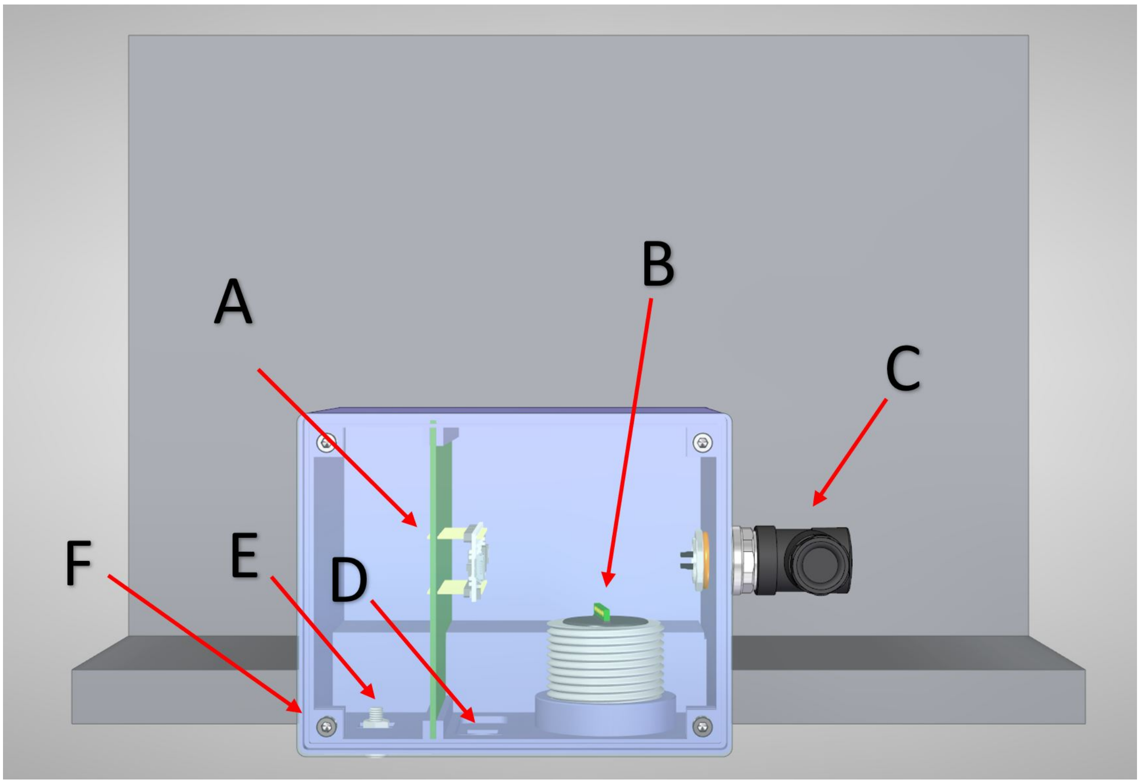


Figure 24. Deck sensor enclosure front view.

Because the sensor casing was threaded, the best option was to design are thread, depicted in Figure 24 (B), so that the sensor could be screwed in, making it leveled with bottom, while remaining watertight. Making it more secure, special piping tape was added to the thread to block any moisture getting inside the enclosure. Threading was also used to attach the connector to the side of the case,shown in Figure 24 (C).

Placing the electronics, a special slot was added to the top and bottom, which allowed the PCB to be slide into the chassis depicted in Figure 24 (A). This also made the removing of board very comfortable and fast.

To make the case as airtight as possible, silicone casket, depicted in Figure 24 (F) was added between the case and the cover. The cover was attached using four M3 bolt with nuts inside the case.

3.7.2 Attachment hook

At first the case was meant to be attached using neodymium magnets. These are rare-earth type permanent magnets, which are the most strongest magnets commercially available. Inside the enclosure, two pockets were constructed on the bottom side shown in Figure 25 (G), where the magnets are placed, leaving only few millimeters of space between the magnet and the support beam, only separated by 1.5 mm of plastic.

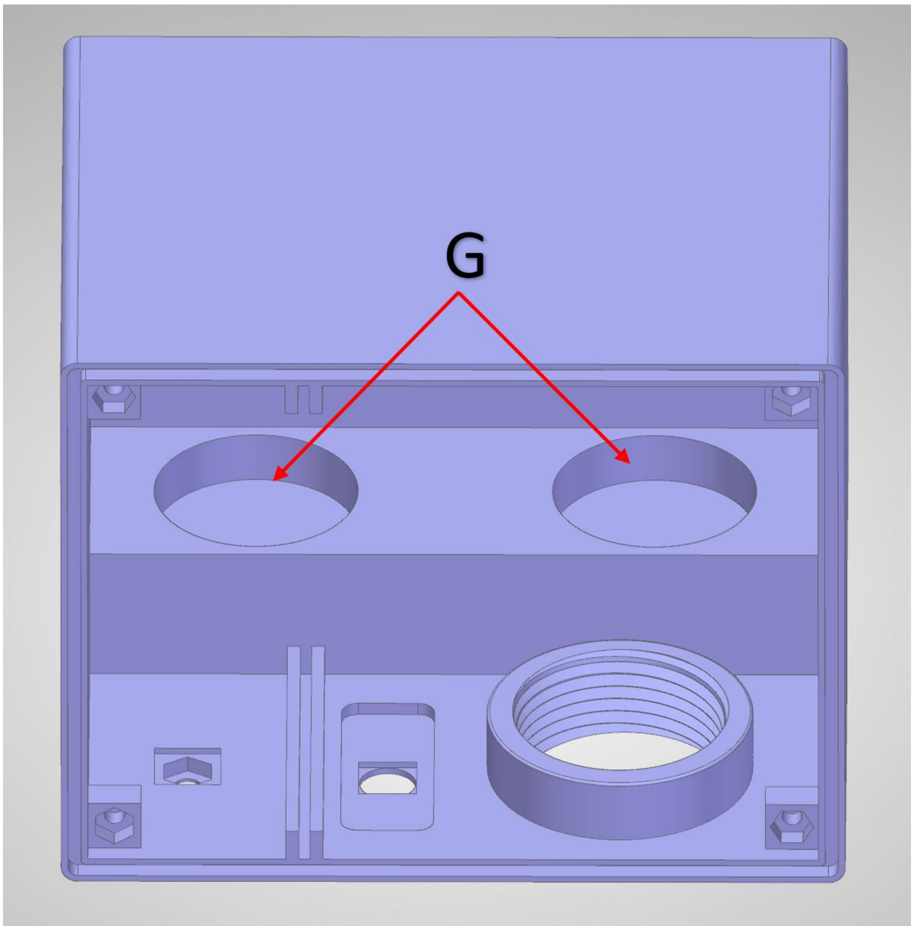


Figure 25. Deck sensor enclosure magnet slots.

The attachment hook was added to enclosure to provide more support and to ensure that the enclosure would not move in x and y axis. Hook itself was made from steel or aluminium, which has a two 90 degree bends to create a C shape attachment, which goes to the opposite end of the support beam. The dimensions of the hook vary due to the different sizes of support beams. Figure 26, shows how the other end of the hook is attached to chassis with M4 bolt, which is screwed into a nut, shown in Figure 24 (E).

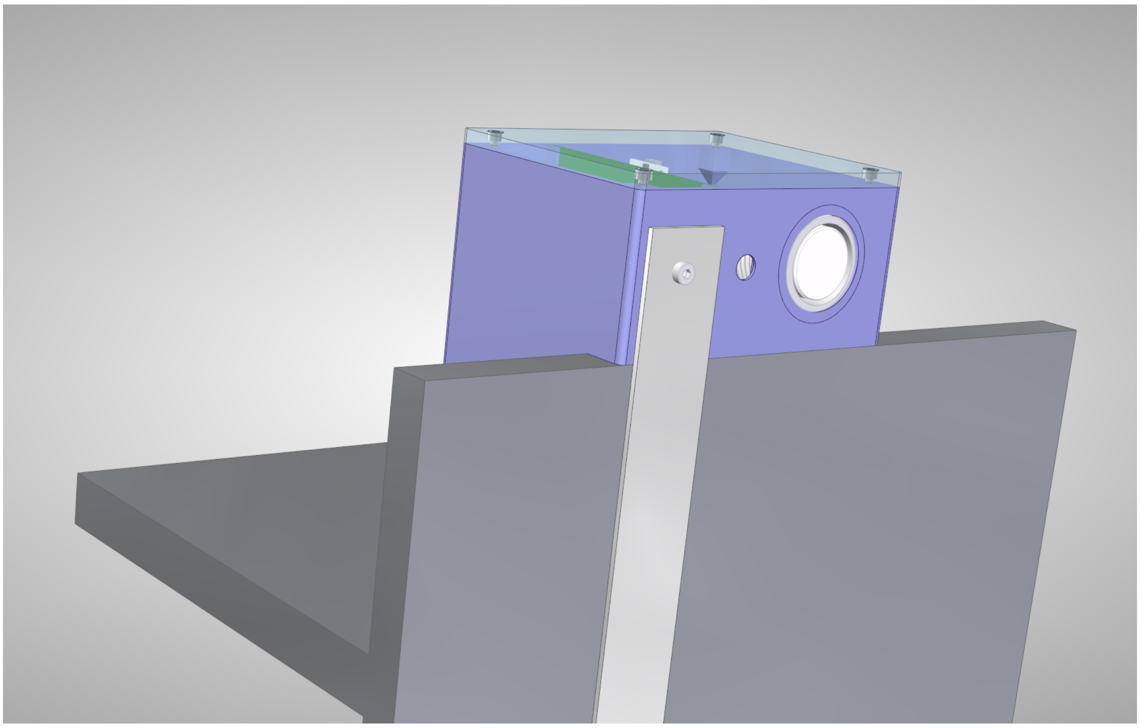


Figure 26. Deck sensor enclosure bottom view.

4 System Integration

This chapter contains descriptions how the subsystems and their components were connected together into one system and how integrating with ship's systems was handled. As previously described in the requirements, the system must cover all the car decks.

Before any sensors were placed into the shipping, layout planning had to be done. Minimum number of sensors to cover each car deck, how many deck sensors can be connected to a single collector and how many collectors could one deck controllers manage. On what CAN Bus baud-rate should the system operate without any data bottlenecks and what are the limitations when using higher bandwidth. What type of cable should be used, its characteristics and the limitations of power transmission.

Cable chosen for the connections between subsystems is CAT6, which is commonly known as Ethernet cable. Goal was to use one cable, that would carry both the CAN Bus and power. This option would save a lot of money during integration and bring down overall costs. CAT6 was the best choice due to having 8 wires. Two wires would be used for CAN Bus, at least 2 for power, leaving 4 reserved, in case of power limitations, which will be covered in future sections in more detail. Instead of connecting the deck sensors in series, the most optimal method would be to connect them in parallel. Otherwise the sensors would be depending on each other and in case of one sensor fails or gets damaged, would render the sensors next to it, inoperable. The maximum length of the cable depends on three conditions.

- CAN Bus Baud Rate
- CAN protocol
- Power delivery

4.1 Controller Area Network

This section covers basics of CAN, working principle, advantages, different frame types. CAN stands for Controller Area Network, that was developed by Bosch in 1985. Before inventing CAN-Bus, cars and other vehicles electronic systems were connected using point to point method as shown in Figure 27 [39].

As the automobile manufacturers started to add more and more electronic devices, to offer the customers more features and extras. As a result, the manufacturing costs skyrocketed, got more difficult and time-consuming, due to the amount of wiring that was added. As the sales of vehicles plummeted the car companies started to look for alternatives to point to point wiring system. The result was a CAN bus, which is multi-master, message broadcast system. Compared to other networks, such as Universal Serial Bus (USB) or Ethernet, CAN does not does not send large blocks of data. Instead it sends information from nodeA to nodeB under the supervision of a central bus master.

In a CAN network, many short messages like temperature or RPM are broadcast to the entire network, which provides for data consistency in every node of the system. In addition, the CAN provides high immunity to electrical interference, ability to self-diagnose and repair data errors. Thanks to these features, as of 1993, CAN has been considered to be an industry standard in various industries [40].

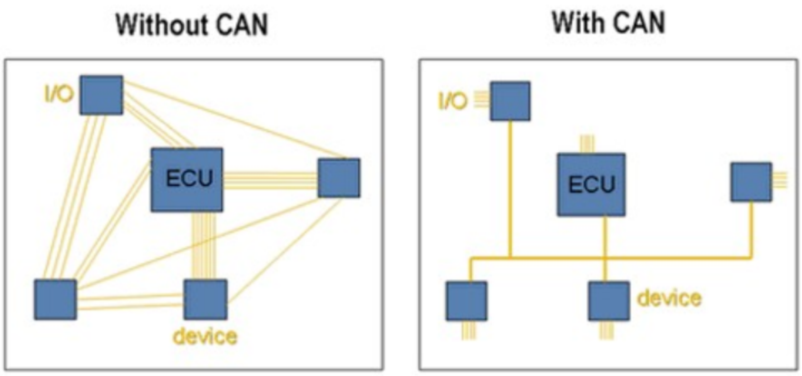


Figure 27. CAN Bus wiring vs point to point wiring [41].

Each device in the CAN Bus, has a CAN-transceiver. Every device in the CAN Bus, sees all the messages that are being exchanged between devices. Transceivers goal is separate, which messages belongs to the device and which ones do not. In addition, every message has a given priority, which is used to avoid collisions between messages. If two messages are trying to be sent at the same time, message with higher priority will be sent first and message with lower priority will have to wait until the process is finished. Standard frame is depicted in Figure 28. The packet is divided into following sectors:

- Start of frame (SOF) - The start of the message is determined by the SOF bit. Additionally, this bit is used to synchronize nodes in the network.
- Identifier - A standard 11-bit indicator determines the importance of the message. The lower the 11-bit binary code value, the higher the message importance.
- Remote transmission request (RTR) - A bit that determines whether or not it is a data frame
- Identifier extention (IDE) - A bit that distinguishes between a regular frame and an extended frame
- r0 - Reserved bit
- Data length code (DLC) - Consists of 4 bits, indicating how many bytes of information are in the data field
- Data field - A data field that can hold up to 8 bytes or 64 bits of data
- Cyclic redundancy check (CRC) - A 16-bit field containing a checksum (how many bits were transmitted) to identify bad frames
- ACK slot - Checks whether the frame has arrived or not
- End of Frame (EOF) - A 7-bit field that specifies the end of the message.
- Interframe space (IFS) - A 7-bit field containing the required time for the controller to move the message to the correct position of the received frame

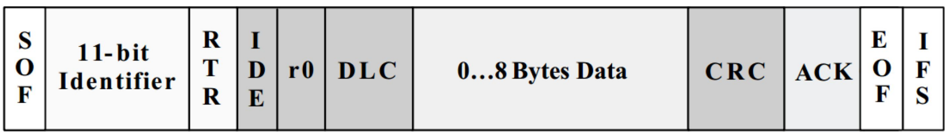


Figure 28. CAN standard frame [40].

Figure 29 depicts extended frame, which is similar to standard frame, only having two extra parts:

- Substitute remote request (SRR) - bit to replace RTR bit in extended format
- r1 - Reserved bit

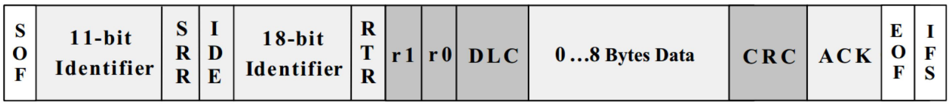


Figure 29. CAN extended frame [40].

4.1.1 Can Bus Length, Nodes and Bus Load

Baud rate describes the CAN Bus's speed of operation, how fast could it change information, is limited by the length of the cable. According to the Texas Instrument guide [40], the rule of thumb says that bus lengths over 100 meters are derived from the product of the signaling rate in Mbps and the bus length in meters, which should be less or equal to 50. This is depicted in Equation 4. In addition in Table 22 depicts maximum baud rate dependence on cable length.

$$\text{Signaling rate}(Mbps) * \text{Bus length}(m) <= 50 \qquad (4)$$

Table 22. Baud rate vs cable length.

Baud Rate	**Maximum cable length (m)**
1 Mbits/s	40
500 kbits/s	100
250 kbits/s	240
100 kbits/s	660
50 kbits/s	1000
5 kbits/s	1300

In the previous section, a standard frame is 111 bits long. It must be also considered that there are 3 bits for inter-frame spacing and bit stuffing which is dynamic and impossible to predict. Bit stuffing is a procedure that used to maintain synchronization. This is used, when there are a five consecutive bits with same polarity, a bit of opposite polarity is added. It is necessary due to the non-return to zero coding. According to the worst case scenario the 111 bit frame transforms into 135 bit frame. Based on those conditions, it is possible to calculate how many deck sensors could be connected to one

CAN Bus. For these calculation, depicted in Equation 5, worst case scenario will be used, where the frame size is 135 bits and each deck sensor sends 10 messages in second.

$$\text{Number of bits} = \text{Fr} * \text{Nmsg} \tag{5}$$

- Fr - Frame size
- Nmsg - Number of messages in a second

Based on the calculations one sensor would send 1350 bits of information in one second. If baud rate of 250 kbits/s is used and considering the worst case scenario, that would allow to connect up to 185 sensors to one bus, which would utilize the bus 100 %. Using 100 % is never possible as there are limitations tied to bus loads. The normal recommended load should be > 30%. When the bus load exceeds 70 %, network management is required.

Table 23. Can Bus Load.

Baud Rate	**Number of Nodes**	**Number of Bits in second**	**Bus load**
250 kbits/s	10	13500	5.4 %
250 kbits/s	25	33750	13.5 %
250 kbits/s	50	67500	27 %
250 kbits/s	75	101250	40.5 %
250 kbits/s	100	135000	54 %
250 kbits/s	125	168750	67.5 %
250 kbits/s	150	202500	81 %

Based on those calculations, when using 250 kbits/s the optimal choice would be 100 to 125 nodes. As mentioned before the maximum cable length is 240 meters for 250 kbits/s and distance between each node is 3.2 meters, allows to put maximum of 75 nodes total. Choosing the right protocol also plays role, as each have their own limits, shown in Table 24, how many nodes can each protocol support.

Table 24. CAN protocol maximum number of nodes.

CAN protocol	**Maximum Number of Nodes**
CANOpen [42]	127
DeviceNet [43]	64
CanKingdom [44]	254

4.2 Power delivery

This section will give an overview about the power transmission to the subsystems. When planning the cable length, determining how many sensor can be hooked to one cable, there are multiple electrical and cable parameters and conditions that play a role: wire material, wire thickness, input voltage, number of conductors, phase, distance and load. Cables are usually made from either aluminum or copper, the last being more popular on power delivery as copper is better conductor.

Wire thickness, which is usually represented in AWG (American Wire Gauge). Higher AWG value means thinner strands, reducing the wire diameter. CAT6 cables usually have wire thickness of 23 or 24 AWG. Phase means if the voltage is Alternating current (AC) or Direct current (DC). In out case, the system uses DC. Number of conductors refer to build structure of the wire itself. Whether the wire is made from solid copper or made from multiple strands of copper. The higher the count of strands, less current can the wire pass through. Final factor is length, as longer cables create more resistance. Based on the material, length and thickness, the resistance of the cable can be calculated using the formula presented in Equation 6:

$$\Omega = \frac{\rho * L}{A} \tag{6}$$

- Ω - Resistance
- ρ - Resistance
- A - Cross-section Area
- L - Length

Using this equation with another, we can calculate based on the parameters the voltage drop in the wire using Equation 7 and Equation 8.

$$V_{Drop(V)} = I_{Wire(A)} * R_{Wire(\Omega)} \tag{7}$$

$$V_{Drop(V)} = I_{Wire(A)} * 2 * L_m * R_{Wire(\Omega/km)}/1000_{m/km} \tag{8}$$

Table 25. Maximum current ratings with copper wire.

AWG	**Single Core**	**Up to 3 cores**	**4 - 6 cores**	**7 - 24 cores**	**25 - 42 cores**
24 AWG	3.5	2	1.6	1.4	1.2
22 AWG	5.0	3	2.4	2.1	1.8
20 AWG	6.0	5	4.0	3.5	3.0

Taking into consideration the wire limits, which are presented in Table 25 and using the calculations based on the input parameters. It is very critical that the voltage drop cannot drop lower than 7.5 V which is needed to voltage regulator to work properly. To consider worst case scenario, the input voltage were 12 V and 24 V, the cable used was 24 AWG and each deck sensor consumes roughly around 100 mA. With 20 deck sensors, the current consumption would be 2 A. As the calculations in Tables 26and 27 show that the one wire is not enough, but if three wires are used, giving 3 times the cable diameter, allowing to pass more current, can significantly reduce voltage drop.

Table 26. Voltage drops at 12V.

Current	**Number of wires**	**Length**	**Voltage drop**
1 A	1	32 m	5.39 V
1.5 A	1	48 m	Failed
2 A	1	64m	Failed
1 A	2	32 m	2.13
1.5 A	2	48 m	4.8 V
2 A	2	64m	8.53 V
1 A	3	32 m	1.34 V
1.5 A	3	48 m	3.02 V
2 A	3	64m	5.36 V

If the input voltage would be increased from 12 V to 24 V the results of voltage drops represented in Table 27 would be the following:

Table 27. Voltage drops at 24V.

Current	**Number of wires**	**Length**	**Voltage drop**
1 A	1	32 m	4.27 V
1.5 A	1	48 m	9.62 V
2 A	1	64m	17.10 V
1 A	2	32 m	2.13
1.5 A	2	48 m	4.8 V
2 A	2	64m	8.53 V
1 A	3	32 m	1.34 V
1.5 A	3	48 m	3.02 V
2 A	3	64m	5.36 V

It can be concluded that number of deck sensor connected to one cable is not dependant on CAN Bus limitation, but it actually is power delivery which has a biggest effect. The optimal choice is to use cables with 3 conductors with length of 48 meters, which would allow to connect 48 sensors with one CAT6 cable. Even if the wire gauge is increased from 24 AWG to 23 AWG, the difference is not considerable. In addition the maximum amps that the wires can handle is tied to its construction, mainly from how many strands in the wire made of. Wire using during the testing has 6 cores, which sets the maximum amperage to 1.6 A. It must be also noted that the cable resistance changes with the temperature as depicted in Equation 9. While the change in resistance is minuscule, the change should still be taken into consideration.

$$R = R_{ref} * [1 + \alpha * (T - T_{ref})][45] \tag{9}$$

- R - Conductor resistance at Temperate T
- R_{ref} - Conductor resistance at reference temperature T_{ref}

- α - Temperature coefficient of resistance for the conductor material, which for copper is 0.00393
- T_{ref} - Reference temperature that α is specified at for the conductor material

5 Testing and Validation

This chapter covers the testing aspects, what was tested, how the testing was handled, procedures and the results. To find nay problems, error and unknown-unknowns, the testing and its procedures have to be divided into separate phases, starting with the individual components and parts, moving into a full system step by step. Initially the individual subsystem tests were divided into three categories:

- Hardware testing
- Software testing
- Mechanical testing

After all the subsystems were separately tested, testing and validation moved on into joint testing where the subsystem were working together. Finally when the system had passed all the tests in controlled environment, the testing was carried out in a realistic environment, which was Tallink Megastar.

5.1 Hardware testing

Hardware testing main objective was to ensure that all the electrical components are operating in normally. This meant checking if all the components were connected properly, no short-circuits that can damage the the individual components, all the electrical outputs and inputs were correct. These tests were carried out by using power supply, oscilloscopes and multimeters. After all hardware tests were successfully passed, testing moved on software testing.

5.2 Software testing

During software testing, the main focus was on reliability. The system was left running for at least 24 hours and after that the measurement data was checked using specialised MATLAB script. The functionality that was to be tested is the following:

- Time of measurement is saved into files

- Length of every measurement file is around 15 minutes
- Data saved into the files, does not have any missing rows, values. Including when the battery has depleted.
- Sensor will try to connect to Raspberry Pi via WiFi and can be pinged remotely
- Readings from ultrasonic sensors stay within normal limits

Figure 30 depicts a small fraction of the logged data that was collected during software testing. Three ultrasonic sensors were set up to measure in sequence for 24 hours. A special MATLAB script was written to analyse the collected information, with each color representing a different sensor as shown in the figure. In addition a line was added, which marks the measured height from the ground as a point of reference. Y axis shows the distance measurement and X axis the time. First thing that was noticed during analysis, were constant spikes by sensor 2, which were likely to due to reflection coming from other sensors. Sensor 1 and sensor 3 also experienced some reflection, by giving a very high measurement for a moment. This was very likely because of power filtration and reflections. Overall the two sensors were quite accurate, when compared against reference measurement.

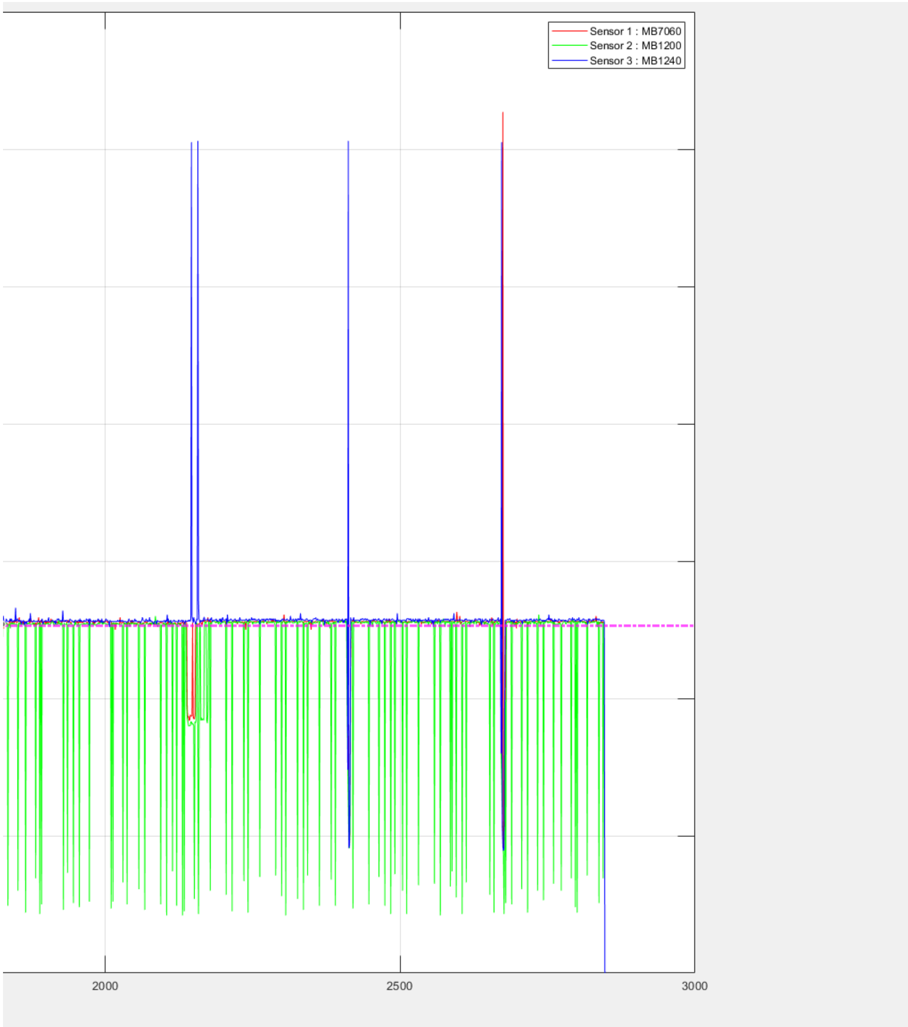


Figure 30. Ultrasonic sensor measurement data analysis.

5.3 Mechanical testing

5.3.1 IP Class testing

To ensure that the manufactured chassis matches the specifications of at least IP55 rating, a series of tests were conducted. First test simulated rain to ensure that the enclosure would be protected against any moisture. A piece of paper towel was placed inside the case and after sealing the cover, it was sprayed with a shower hose for 10 minutes from every angle. If the paper towel didn't have any spots of moisture, it was deemed waterproof. To check if case would handle moisture under pressure, it was submerged in water for 30 minutes at depth of 1 meter. For this, the case was placed in black garbage bag that was filled with water. After the 30 minutes, if the paper towel was still dry it passed the

IP rating test. During initial IP testing 2 chassis out 6 failed to achieve waterproofness, due to poor silicon seal quality. After reapplying new seals, all of the chassis passed the testing.

5.4 Subsystem testing

Before the next testing phase began, each constructed deck sensor received an unique serial number, that helped to keep track testing conditions, which versions of enclosure, hardware and software were used and the testing results, with each functionality status separately documented. Serial number digits represent values the following parameters:

- TYPE - deck sensor [DS]
- MODEL - ultrasonic type [01]
- YYMMDD - date of manufacturing
- COUNT - 4 digit counter [0001, 0002, ...]
- Enclosure version - (git commit hash)
- HW version - (git commit hash)
- SW version - (git commit hash)
- Status - [BROKEN/MANUF/TEST-RUN/TEST-FAIL/TEST-OK]

The report of the testing is depicted on Table 28:

Table 28. Testing procedure template.

Test field	**Description**	**Testing location**	**Testing method**
FUN-01	RGB LED blinking alternately blinks 100ms ON / 900ms OFF	Lab	Visual inspection
FUN-02	Chassis has mechanical attachment hook	Lab	Visual Inspection
FUN-03	CAN Bus Transmission and Receive for 24 hours without any errors	Lab	Specialized CAN software
FUN-04	Every file length is around 15 minutes	Lab	MATLAB script measurement time analysis
FUN-05	Ultrasonic sensor measurements stay in normal range	Lab	MATLAB script graph
POW-01	Test voltage regulator outputs under load	Lab	Testing with multimeter
POW-02	Test and monitor Deck sensor current consumption	Lab	Testing with multimeter
REL-01	Sensor has been tested for 24 hours while being 3-4 meters from the ground	Ship car deck	Physical Installation
REL-02	Sensor has been tested in a at least -10 degree environment for at least 24 hours	Ship car deck	Physical Installation

5.5 CAN Bus testing

This section will give an overview about CAN bus testing, that was done by simulating the CAN Bus environment using special hardware, shown in Figure 31 and software called CANDoIso [46].

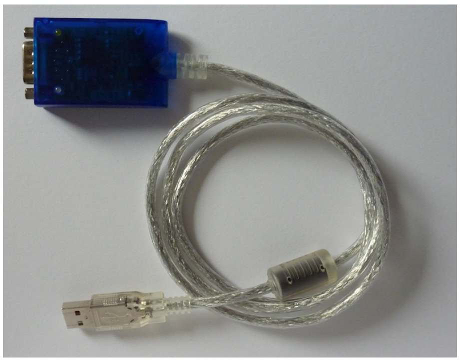


Figure 31. CANDoIso Device [47].

Software allows to send CAN messages from the PC to the device and receive message from the device as well. In addition the software can be configured to operate at different baud rates and display the contents of each frame. Figure 32 depicts how the device is connected with the system during testing. A 120 ohm resistor was added because CAN Bus needs a terminating resistor to avoid any reflections, that might occur during transmission. During testing the deck sensor was configured to send a message every 100 ms, to simulate realistic scenario, as described in the previous chapters.

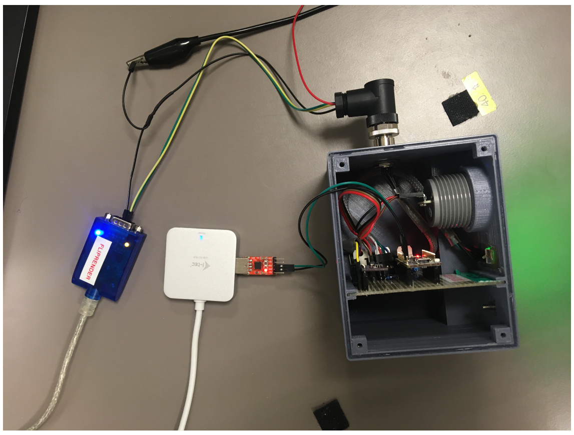


Figure 32. CAN Testing.

6 Results and future work

Total of 6 deck sensor prototypes were developed that were capable of detecting a vehicle, controlling the indicator based on detection and communicating with other subsystems via CAN Bus. Overall deck sensor the average power consumption remained under 100 mA. Ultrasonic sensor that was used, was able to measure with accuracy of ± 10 centimeters in range of 5 meters. For indicator RGB LED was attached on the bottom of the enclosure, that was easily seen even from 30 meters away. Separate power module were added to control the LED brightness via PWM was helpful, especially during testing. To change brightness levels, only few values had to be changed in the code.

Enclosure designed for hardware was manufactured using 3D printing. Chassis filled all the set requirements: lightweight, easy to produce, ability to change components quickly and most importantly being waterproof with IP55 rating.

Despite all of this, more work needs to be done, which includes deck controller and collector, which are still in prototyping phase. In addition there are many features that could be improved. Instead of just using a RGB LED as an indicator, one possible improvement would be to project a beam of light that is visible on the floor by using a reflector with lenses. This would reduce the chances of accidental blinding, if looked directly inside the LED at close range. In addition, instead of relying solely on ultrasonic sensors, adding a camera to the deck sensor, could improve vehicle detection in situations where separating vehicles from each other would be impossible with ultrasonic sensor. Some research is being conducted on that topic, which would allow image processing to be done inside microcontrollers. Finally improving detection algorithms could help in reducing the number of sensors needed for covering the whole car deck.

As the Tallink Smart Deck project continues till summer of 2020, biggest challenges are still yet to come due to the changes in requirements and system complexity.

7 Conclusion

The main objective of this master thesis was to propose an develop an initial vehicle detection and guidance solution prototype for Tallink Megastar car deck, which is capable of communicating with other sensors, subsystems and ship's systems. As a result, 6 deck sensor prototypes were created that fulfilled all given the requirements given in the beginning.

During the development, one of the biggest obstacles was the enclosure design, which has to protect against any moisture, water and have at least IP55 rating, while being lightweight, easy to manufacture and resistance to other environmental factors. Initial enclosure prototypes were manufactured using 3D printing. To make it watertight, industrial grade silicone was used to pour sealing gasket. All of the chassis passed IP55 rating tests successfully. In addition, the chassis had to be attached to car deck's support beam, for which a neodymium magnets were used. To give extra rigidity, special attachment hook was created from aluminium.

Most time in hardware development was spent on planning the layout of the board to make it modular, in case a component is needed to be changed, while being as compact as possible. In addition some features were added only for testing, such as connector for logging module, which needed separate power circuits.

Software development was fairly straightforward, HAL libraries were used to cut down development. Furthermore libraries for previous projects were used with minor modification, to be compatible with different hardware. In addition some function needed to be written from scratch.

Thesis author suggests that the designed deck sensor prototypes is a important step in making advancement in automating ships and their systems. It must also noted that biggest challenges are still ahead, as the initial concept is in constant development and might mean more changes to the current system.

References

[1] C. S. I. Ltd, *The Scope of Automation In The Shipping Industry*. [Online]. Available: https://www.cargoshippinginternational.com/the-scope-of-automation-in-the-shipping-industry (visited on 2020-01-15).

[2] B. D. Man, *Tallinn's strategy for smart port solutions*. [Online]. Available: https://eu.smartcitiescouncil.com/article/tallinns-strategy-smart-port-solutions (visited on 2020-01-15).

[3] R. Mee, *Tallink ja TalTech hakkavad koos tarka laeva arendama*. [Online]. Available: https://www.logistikauudised.ee/uudised/2019/01/16/tallink-ja-taltech-hakkavad-koos-tarka-laeva-arendama (visited on 2020-01-15).

[4] M. Leier, *Tallink smart car deck development*. [Online]. Available: https://iot.ttu.ee/projects/smart-cardeck/ (visited on 2020-01-15).

[5] Aripaev, *Nutikas lahendus parkimisprobleemile*. [Online]. Available: https://www.aripaev.ee/sisuturundus/2019/04/02/nutikas-lahendus-parkimisprobleemile (visited on 2020-01-15).

[6] J. Tomperi, *T1 Mall of Tallinn Opens!* [Online]. Available: https://www.parking-net.com/parking-news/portier/t1-mall-of-tallinn-opens (visited on 2020-01-15).

[7] S. P. Ltd, *Vehicle detection sensors*. [Online]. Available: https://www.smartparking.com/smartpark-system/smart-sensors (visited on 2020-01-15).

[8] Z. I. Roland Cheng Wendi Heinzelman, *A Motion-Tracking Ultrasonic Sensor Array for Behavioral Monitoring*. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3211111/pdf/nihms332794.pdf (visited on 2020-01-15).

[9] O. J. R. Kevin Heffner, *Enabling Technologies for Maritime Autonomous Surface Ships*. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1357/1/012021/pdf (visited on 2020-01-15).

[10] D. L. Jones, *Cleaning Water Damaged PCB*. [Online]. Available: https://www.eevblog.com/forum/repair/cleaning-water-damaged-pcb/ (visited on 2020-01-15).

[11] D. Alcox, *How Salt Water Affects Electronics*. [Online]. Available: https://www.maxbotix.com/articles/135.html (visited on 2020-01-15).

[12] T. E. Company, *What is an IP rating?* [Online]. Available: https://www.enclosurecompany.com/ip-ratings-explained.php (visited on 2020-01-15).

[13] Eclipse, *Eclipse*. [Online]. Available: https://www.eclipse.org/ (visited on 2020-01-15).

[14] STMicroelectronics, *ST-Link*. [Online]. Available: https://www.st.com/en/development-tools/st-link-v2.html (visited on 2020-01-15).

[15] ——, *STM32CubeMX*. [Online]. Available: https://www.st.com/en/development-tools/stm32cubemx.html (visited on 2020-01-15).

[16] ——, *STM32L432KC Datasheet*. [Online]. Available: https://www.st.com/resource/en/datasheet/stm32l432kc.pdf (visited on 2020-01-15).

[17] ——, *STM32F446RE Datasheet*. [Online]. Available: https://www.st.com/resource/en/datasheet/stm32f446re.pdf (visited on 2020-01-15).

[18] ——, *STM32F767ZI Datasheet*. [Online]. Available: https://www.st.com/resource/en/datasheet/stm32f767zi.pdf (visited on 2020-01-15).

[19] N. Semiconductor, *TJA1051 Datasheet*. [Online]. Available: https://www.nxp.com/docs/en/data-sheet/TJA1051.pdf (visited on 2020-01-15).

[20] T. Instruments, *SN65HVD233 Datasheet*. [Online]. Available: https://www.ti.com/lit/ds/symlink/sn65hvd233-ht.pdf (visited on 2020-01-15).

[21] O. Semiconductor, *NCV7341 Datasheet*. [Online]. Available: https://www.onsemi.com/pub/Collateral/NCV7341-D.PDF (visited on 2020-01-15).

[22] Electropeak, *TJA1051 CAN transceiver*. [Online]. Available: https://electropeak.com/pub/media/catalog/product/cache/10f519365b01716ddb90abc57de5a837/c/j/cjmcu-tja1051-can-transiever-module__-1.jpg (visited on 2020-01-15).

[23] STMicroelectronics, *L7805CV Datasheet*. [Online]. Available: https://eu.mouser.com/datasheet/2/389/l78-974043.pdf (visited on 2020-01-15).

[24] L. Technologies, *LT1086 Datasheet*. [Online]. Available: https://eu.mouser.com/datasheet/2/609/1086ffs-1268422.pdf (visited on 2020-01-15).

[25] STMicroelectronics, *L4940 Datasheet*. [Online]. Available: https://eu.mouser.com/datasheet/2/389/l4940-973918.pdf (visited on 2020-01-15).

[26] Addicore, *L7805CV regulator*. [Online]. Available: https://cdn3.volusion.com/btfzd.umflq/v/vspfiles/photos/AD252-2.jpg?v-cache=1479738161 (visited on 2020-01-15).

[27] NDT, *Temperture and the speed of sound*. [Online]. Available: https://www.nde-ed.org/EducationResources/HighSchool/Sound/tempandspeed.htm (visited on 2020-01-15).

[28] SICK, *UM30-213112 | UM30 Datasheet*. [Online]. Available: https://docs.rs-online.com/2988/0900766b816851f2.pdf (visited on 2020-01-15).

[29] Maxbotix, *MB7060 Datasheet*. [Online]. Available: https://www.maxbotix.com/documents/XL-MaxSonar-WR_Datasheet.pdf (visited on 2020-01-15).

[30] T. sensors, *xx918a3c2m12 Datasheet*. [Online]. Available: https://docs.rs-online.com/0766/0900766b81060e04.pdf (visited on 2020-01-15).

[31] Maxbotix, *MB7060-500*. [Online]. Available: https://www.components-mart.com/thumb/product/39/MB7060-500.jpg (visited on 2020-01-15).

[32] Lumex, *SML-LX1610RGBW/A Datasheet*. [Online]. Available: https://www.mouser.ee/datasheet/2/244/SML-LX1610RGBW%5C%20A-82809.pdf (visited on 2020-01-15).

[33] Cree, *XMLCTW-A2-0000-00C2AAAB1 Datasheet*. [Online]. Available: https://www.mouser.ee/datasheet/2/90/XLampXML_Color-1147431.pdf (visited on 2020-01-15).

[34] Lumex, *SML-LX5050SIUPGUBC Datasheet*. [Online]. Available: https://www.mouser.ee/datasheet/2/244/SML-LX5050SIUPGUBC-1135719.pdf (visited on 2020-01-15).

[35] Kingbright, *AAAF5051-05 Datasheet*. [Online]. Available: https://www.mouser.ee/datasheet/2/216/AAAF5051-05-1359477.pdf (visited on 2020-01-15).

[36] Amazon, *Multitasking Vs Concurrency*. [Online]. Available: https://www.freertos.org/implementation/a00004.html (visited on 2020-01-15).

[37] ——, *FreeRTOS - Real-time operating system for microcontrollers*. [Online]. Available: https://www.freertos.org/index.html (visited on 2020-01-15).

[38] Siemens, *Solid Edge*. [Online]. Available: https://solidedge.siemens.com/en/ (visited on 2020-01-15).

[39] C. Electronics, *CAN Bus Explained - A Simple Intro (2019)*. [Online]. Available: https://www.csselectronics.com/screen/page/simple-intro-to-can-bus/language/en (visited on 2020-01-15).

[40] S. Corrigan, *Introduction to the Controller Area Network (CAN)*. [Online]. Available: https://www.ti.com/lit/an/sloa101b/sloa101b.pdf (visited on 2020-01-15).

[41] N. Instruments, *Controller Area Network (CAN) Overview*. [Online]. Available: https://www.ni.com/en-us/innovations/white-papers/06/controller-area-network--can--overview.html (visited on 2020-01-15).

[42] ——, *CANOpen*. [Online]. Available: https://www.ni.com/en-us/innovations/white-papers/13/the-basics-of-canopen.html (visited on 2020-01-15).

[43] R. T. Automation, *DeviceNet*. [Online]. Available: https://www.rtautomation.com/technologies/devicenet/ (visited on 2020-01-15).

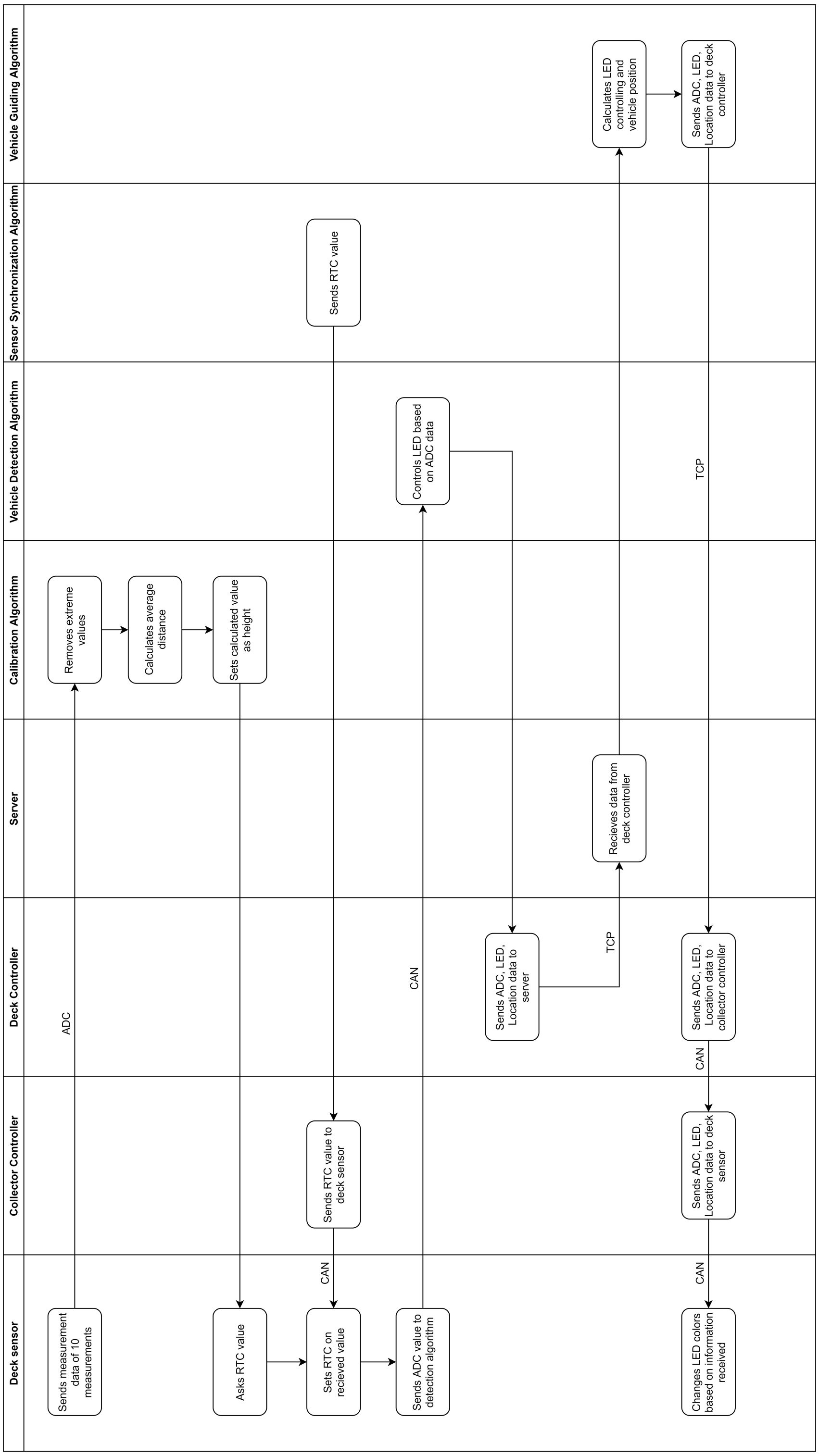
[44] L.-B. Fredriksson, *A CAN Kingdom*. [Online]. Available: https://www.kvaser.com/wp-content/uploads/2014/02/ck301p.pdf (visited on 2020-01-15).

[45] C. R. Nave, *Resistance: Temperature Coefficient*. [Online]. Available: http://hyperphysics.phy-astr.gsu.edu/hbase/electric/restmp.html (visited on 2020-01-15).

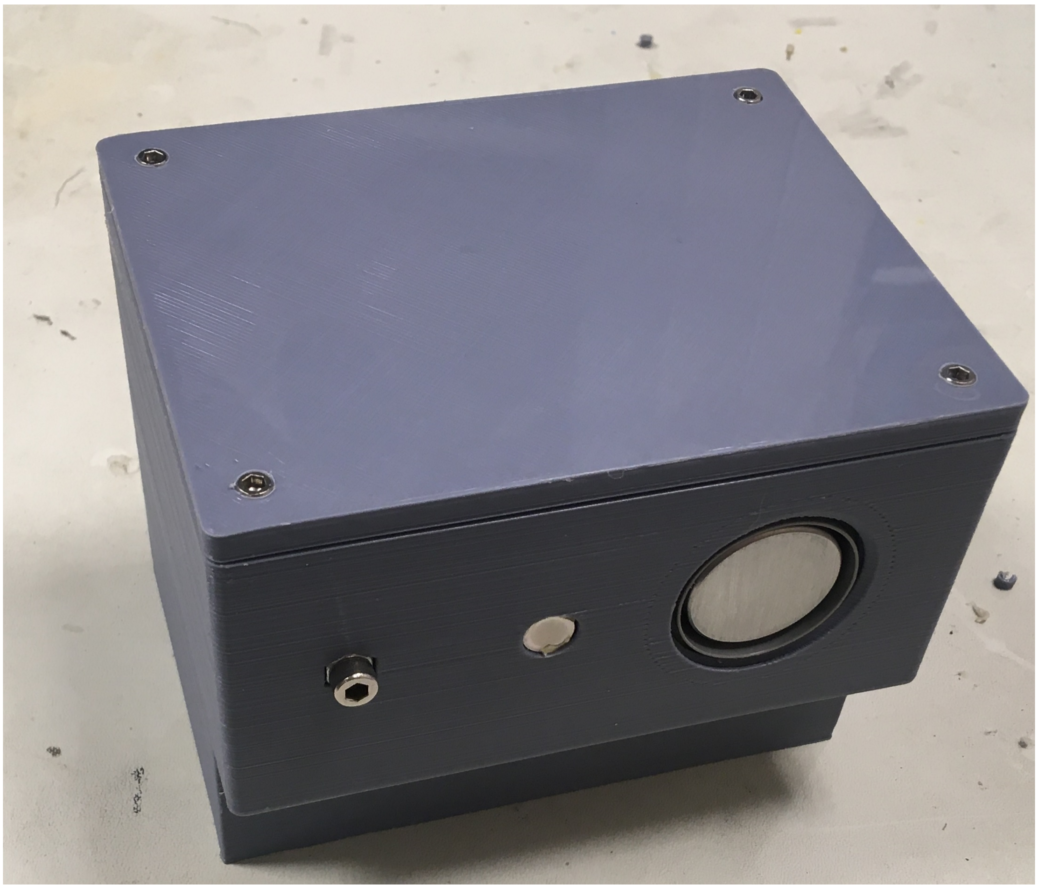
[46] Cananalyser, *Cananalyser*. [Online]. Available: http://www.cananalyser.co.uk/candoiso.html (visited on 2020-01-15).

[47] ——, *Cananalyser*. [Online]. Available: http://www.cananalyser.co.uk/CANdoTopSideFull.jpg (visited on 2020-01-15).

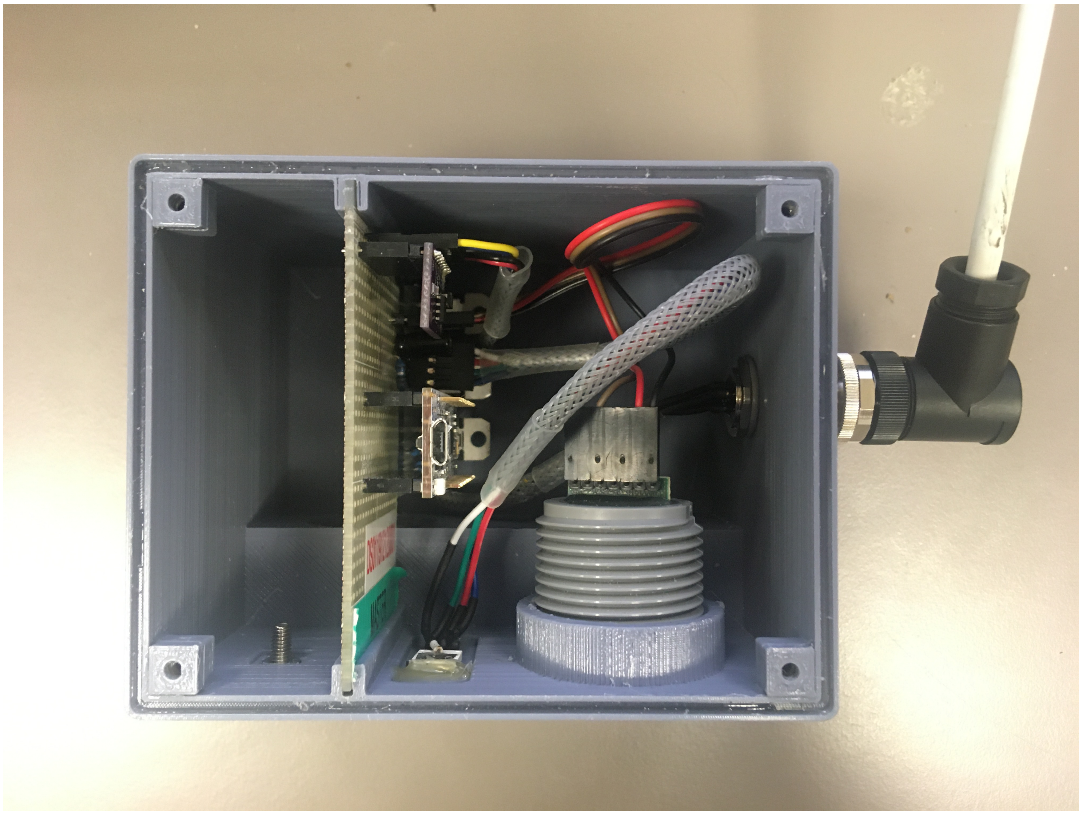
Appendix 1 – Software communication architecture



Appendix 2 – Deck sensor prototype enclosure



Appendix 3 – Deck sensor prototype insides



Appendix 4 – Source code repositories

Deck sensor and deck controller source code is available in git and the repositories are available via follwoing URLs:
Deck sensor: `git@bitbucket.org:tallink_team/deck_sensor_software.git`
Deck controller: `git@bitbucket.org:tallink_team/deck_controller_software.git`