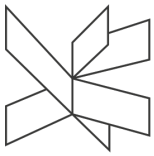

VIA University College

Large Object Detection Next to Waste Containers using IoT

PROJECT REPORT

Eimantas Sipalis 254018
Romans Kotikovs 252550

supervised by
Ib Havn

19415 characters (not including spaces)

Software Engineering, 7th semester
May 11, 2020

Document versions:

Version	Change	Date
0.1.0	Initial project report structure that follows ICT specific guidelines	2020/02/25
0.2.0	Start of versioning	2019/04/15

Contents

List of Figures

Listings

Abstract

The purpose of this project is to research and document findings of simplifying garbage collection from underground trash cans. This is a research paper rather than a development process of a “product to market”.

What are the main technical choices?

What are the results?

Used technologies: Microcontroller - ATMega2560, Ultrasonic Ranging Module HC - SR04, Servomotor, LoRaWAN, C, FreeRTOS, C# .NET Core 3.1.

Glossary

Here a list of terms used in this report can be found. It's also specified how these terms are emphasized in the text.

Terminology

Terms listed here are *italicized* when used in the text. They have the following meanings:

- *Use case* - a written description of how the task will be performed.

Acronyms, initialisms and abbreviations

The following terms are written in the same capitalization of the letters in the text as they are here.

- HAL - Hardware abstraction layer
- IoT - Internet of Things
- API - Application Programming Interface
- JSON - JavaScript Object Notation
- TCP - Transmission Control Protocol

1 Introduction

This introduction is based on the project description found in appendix **??**.

This project is conducted within the waste management industry. The stakeholder of this project is Runik Solutions that operates with Horsens commune who is the customer. It is a common problem for waste collectors, not only in Horsens, that people are leaving large and heavy objects such as household furniture, fridges, washing machines, etc. right next to the underground waste bins. This disrupts the usual waste collection workflow. Underground waste bins are one of the most advanced and used waste collection methods in Europe. That is why it is important to keep the waste collector workflow without impediments.

2 Analysis

Project expectations are described in this section.

2.1 Requirements

During the inception and elaboration phases, main requirements were established - functional requirements are expressed as use cases and non-functional requirements are stated in a list below.

2.1.1 Functional requirements

1. As a kommune's dashboard system, it shall be notified about objects placed exceeding 0.4m width and 0.3m height on the trashcan's platform (coloured in red). Items placed on top of the container do not need to be detected.
2. As a kommune's dashboard system, it shall be notified about previously detected objects that are no longer there.
3. As a system administrator, I should be able to configure time periods or the number of scans a day. (Optional, ask Ib).
4. Ask about heartbeat messages once a week

2.1.2 Non-functional requirements

1. The battery should last at least 2 years.
2. The project equipment should not interfere with the worker's usual way of cleaning out the underground containers.
3. The project should not damage the trash can's equipment (for example, drilling a hole through the trash can container's side is not allowed).
4. The delay of notifying the kommune's dashboard server after the large object detection should never exceed the 10 minute mark.

2.1.3 Use cases

Figure 1 shows the use case diagram of the system which displays the relationship between the actors and the use cases. It offers a high-level view which can be used as a basis for making sure that the system satisfies the requirements.

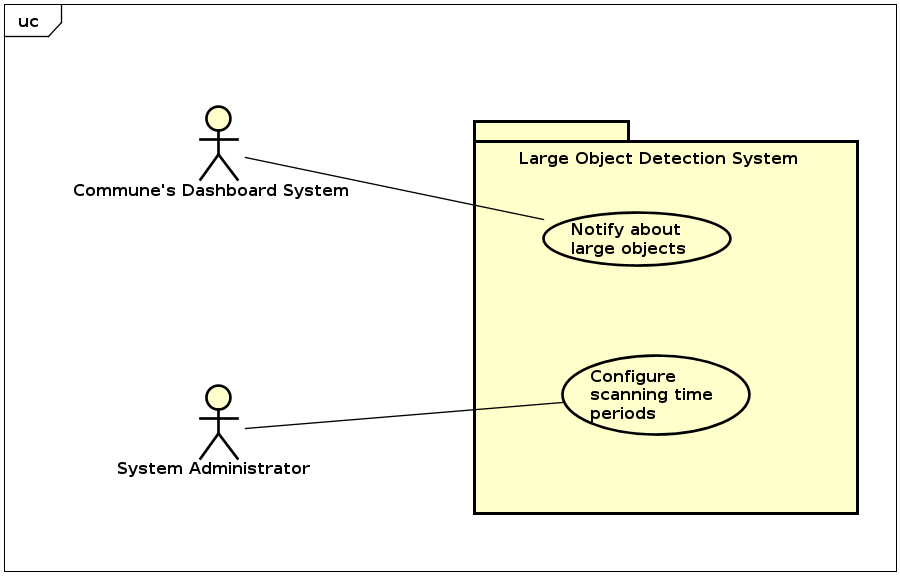

Figure 1: Use case diagram

Actor description

- **Actor** : Kommune's dashboard system
  An internal system used by the Horsens kommune.
- **Actor** : System administrator
  An external person who manages the system.

2.2 Use case descriptions

The use cases are described in more detail in this section. First a brief use case description is given and then a well detailed user description is shown in a corresponding figure.

2.2.1 Use case description: Notify about large objects

Brief use case description: A person puts a large object on the underground bin platform. System detects the object and notifies the commune dashboard system where the object has been detected. See figure 2 for a detailed use case description.

ITEM	VALUE
UseCase	Notify about large objects
Summary	System detects the object and notifies the commune dashboard system where the object ha s been detected.
Actor	Commune's Dashboard System
Precondition	Object has been places on the platform. System scans the platform.
Postcondition	Notification is sent to Commune's Dashboard System.
Base Sequence	1. Object is placed on the platform. 2. System scans for the object. 3. If system detects object, system sends notification to the Commune's Dashboard System.
Branch Sequence	1. Previously detected object is removed from the platform, send updated information to Co mmune.
Exception Sequence	1. Servo motor can't move, notify Commune.
Sub UseCase	
Note	

Figure 2: Use case description for notifying about large objects

For the purposes of the use case scenario clarification, an activity diagram is made which can be seen in figure 3

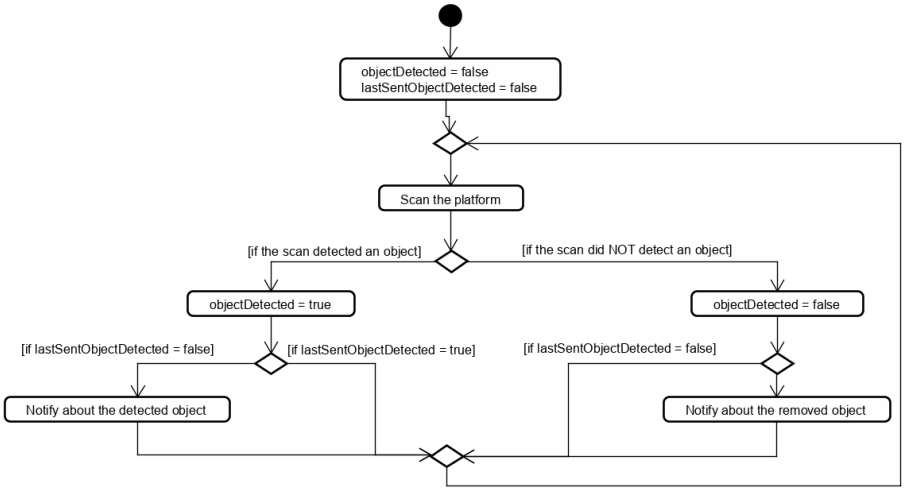

Figure 3: Use case description for notifying about large objects

2.3 Hardware and software investigation

This section server the purpose of describing the research process about the sensors and related software that was conducted to find out if the project requirements can be satisfied with today's technologies.

From the requirements the base functionality of the project can be stated - detect any large objects placed on the platform around the underground trash container. Figure 4 highlights the platform in red.

So any size-wise large (not necessarily heavy) object placed on the platform highlighted in red must be detected. There's no point in discussing the IT infrastructure of the project until a clear path to the solution of detecting a large object problem is described through the analysis process (proving that such functionality is possible). This analysis part will mostly revolve around the hardware - the sensor types and the sensor layout. While choosing the hardware it's important to consider the power consumption as the goal is for the system to operate on a battery.

Figure 4: Trash can container with platform highlighted in red

The trash cans come in batches of 4 (as far as it's been observed), however not all layouts are the same. Some are laid out in a line, others are in a square shape and sometimes they are placed on varying height level. Figures 5 and 6 indicate that.

Figure 5: Trash cans inline on different height levels

Figure 6: Trash cans in a square shape

Two main problems are raised from this analysis.

1. Sensor platform focus - Should the system optimise (reuse sensors) for a certain layout or focus on an individual trash can platform?

2. What sensors should be used?

No matter which sensor will be used in the final product it doesn't affect the sensor platform focus problem in any significant way. So the advantages, disadvantages and problems of focusing the sensors for a single platform or optimising sensors for multiple platforms will be described first.

2.3.1 Sensor platform focus

In this section the analysis of the sensor platform focus problem will be described. In the end, one approach will be chosen, with the reasons stated.

Optimising sensors for multiple platforms

The sensors can be placed in a way that allows the sensor to scan multiple platforms instead of just one. This is shown in figure 7.

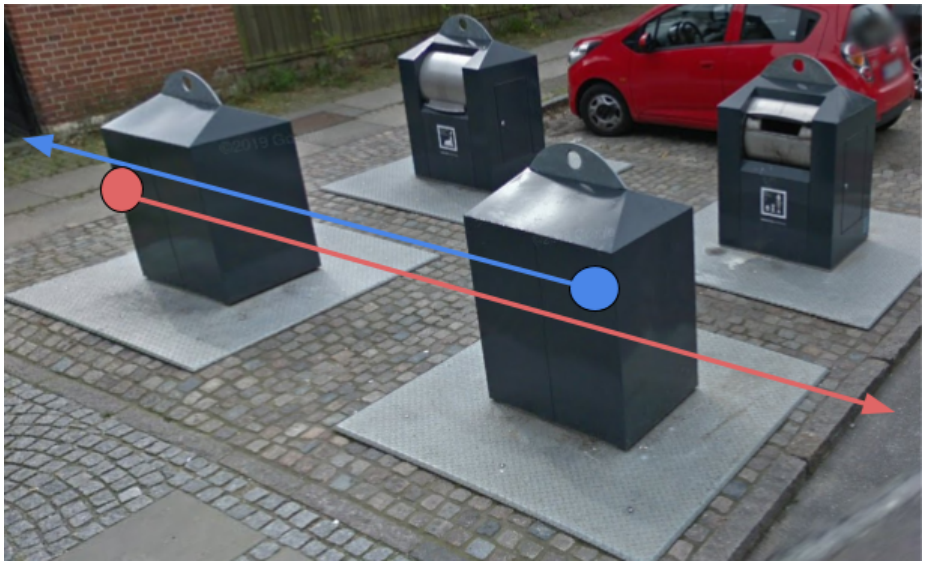

Figure 7: Sensors optimised for two trashcans

In the figure the sensors are indicated by the circles and they are facing the direction of the arrows. The sensors would measure the distance between the large object and the sensor and based on both of their measurements calculate the object's size. In the image above a single sensor set would be able to detect the large object of 2 platforms of one side. If the trash cans are in inline layout a single set of sensors would be able to detect the object on 4 different platforms. See figure 8 for an example.

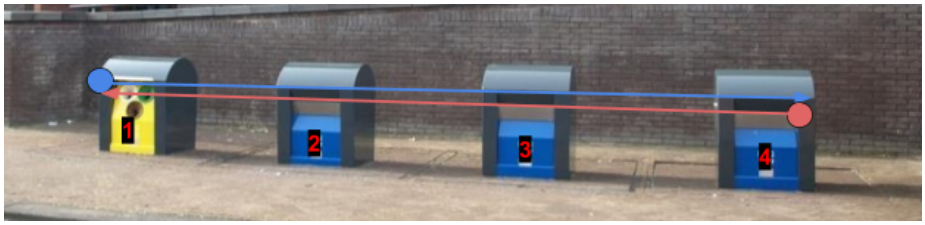

Figure 8: Sensors optimised for four trashcans

However this works only when there's only one object placed. If there's an object placed on platform 1 and another object on platform 3 (marked in figure 8) the system will think that the object is very large (which may or may not be true). Another issue is if there's an object placed in between the platforms - objects placed in between the platforms should not be detected. While it's possible to calculate that the object is not on either of the platforms the object would disallow detecting other objects that are on the platforms as indicated by figure 9. In the figure the large object is indicated by a red square. The large object is blocking sensors' lines of sight to see the other platforms. So the optimised sensor layout doesn't work greatly with multiple objects or objects in between the platforms, which is not such a common scenario, so this is not a huge disadvantage.

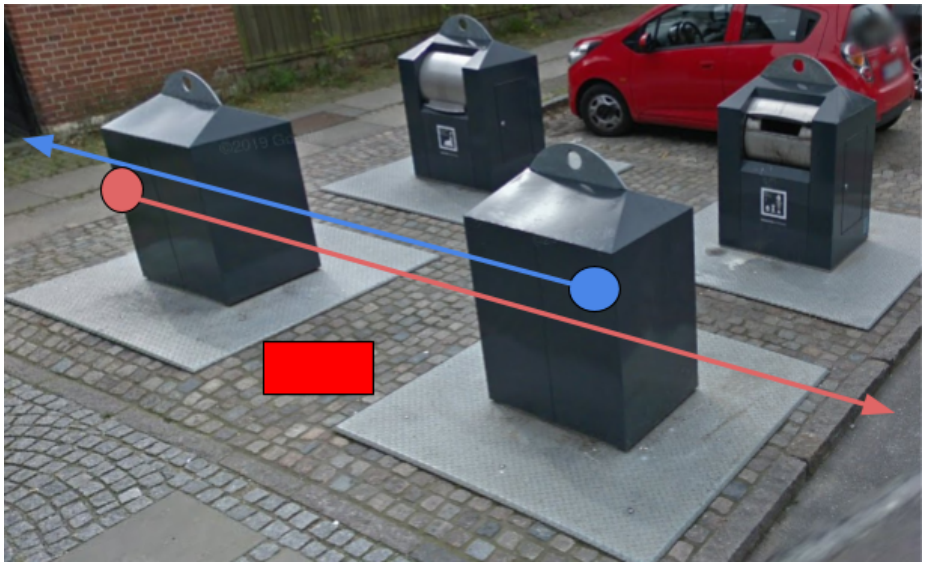

Figure 9: Object placed in between the platforms

However a big problem of this solution is scalability. There are a few different layouts and most importantly not all layouts have trash cans on the same

height level. This would cause each setup to be different - sensors have to point at different angles and sometimes it would be even impossible to set up if the incline or decline is too high. While having one sensor set for multiple platforms is cost and power efficient, the project becomes way more difficult because of all the new variables introduced by infinitely many different layouts.

Another disadvantage of optimised sensor placement is that if one of the sensors fails that's detecting for multiple platforms the detection system's functionality will be impacted for the whole trash can batch.

Focusing on an individual platform

All problems of optimised sensor layout can be solved by focusing on an individual platform:

1. Different layouts problem - doesn't exist anymore because focus is placed on a single platform so the layout of the platforms doesn't matter. Also from what was observed - an individual platform is always leveled out and not tilted. Even if there are any platforms that are tilted the platform is always going to be perpendicular to the trash can itself as shown in figure 10. This makes the solution way more scalable as it doesn't not depend on the batch layout. All the setups can be the same as the sensors can be placed on the trash cans which are always perpendicular to the platform or they can be placed on the platform itself as the platform is always in a straight line (even if the platform is tilted it's still going to be straight and not bent).
2. Multiple objects on different platforms or an object in between the platforms - is not a problem anymore since focus is placed on a single platform and the sensors are going to check only its boundaries.
3. Malfunctioning sensor - if one of the sensors fails it's going to disrupt only those platform's object detection and not the whole batche's.

Figure 10: Object placed in between the platforms

In conclusion, while using a single sensor for multiple platforms might cut down the hardware costs and increase power efficiency, it causes a lot of problems which would have to be solved individually for each different layout, which makes it extremely hard to scale the solution. Focusing on a single unit makes scalability easy - as the solution will be applicable for all different layouts. It's also likely that the savings in hardware costs would be offset by the amount of work hours needed to be done when setting up / maintaining the system because of all the different layouts. So focusing on a single platform is better for the project goals.

Note that here only the sensor layout is described, it's still possible to have the sensors focus on a single platform and have one central node that would handle the communication for the whole batch.

2.3.2 Sensor type

Since the problem for the sensor platform focus is solved, placing focus on choosing the optimal sensor type is a good idea. The following sensor types are going to be considered:

1. Ultrasonic sensor
2. LIDAR
3. Camera

4. Light array sensor

Sensor types that are the least likely to fit the project goals are described first, so they are eliminated as a possible consideration early.

Light array sensor

Light array sensors use 2 sensors (emitter and receiver) and an array of beams instead of a single beam - anything that comes in between the emitter and the receiver will be detected. This improves the detection rate by a lot compared to an ultrasonic sensor, especially for oddly shaped objects. However, its coverage area is still typically not wide enough to cover the entire side of the trashcan platform and its power consumption is way higher compared to a typical ultrasonic sensor - the Bulletin 45PVA–1LEB4–F4 has a detection width of 375mm with max power consumption of 155mA (Rockwell Automation, 2006). With these specifications the sensor already does not seem like a great choice because of the high power consumption, but more importantly because of the way it functions (requiring an emitter and a receiver) it forces the sensors to be placed in non optimal positions as indicated in figure 11.

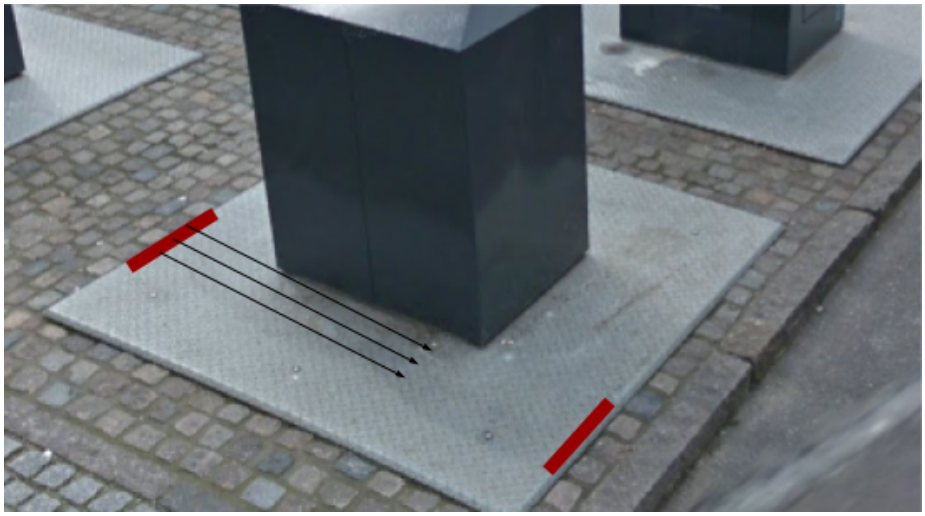

Figure 11: Light array sensor positioning

In the figure the sensors are marked in dark red. This positioning leaves the sensors very vulnerable to mechanical failures. Also it would not be easy to attach them there as they need to be elevated off the ground. In conclusion, looking at different light array sensors seems to indicate that light array sensors are more appropriate for more industrial projects like counting the number of produced objects in a production factory, where you also have access to a continuous power supply (Rockwell Automation, 2020) (Sick Sensor Intelligence, 2020).

Camera

Camera functions quite differently than other considered sensor types. The camera will be used to take images that will be analysed using computer vision to detect the object and their dimensions. However, it is not in the scope of the project to develop a computer vision solution from scratch, so a computer vision library must be used - OpenCV is one such library. From research, it seems that a good microcontroller choice would be ESP32 as it has good community support and a lot of modules available. There are quite a few projects online that are using ESP32, a camera module and computer vision to detect objects (Ayuso, 2018) (Ayuso, 2018).

However, these projects use ESP32 and the camera module to take pictures and then send them to a more powerful processing unit which does the work of computer vision. Although the mentioned projects are doing live recognition and this project needs to process only one or two images per day so the computational needs are way lower. Unfortunately even if it was possible to reduce the computational needs to a point where a battery powered ESP32 could last at least 2 years, people have not had success running OpenCV on it (albzn, 2019).

Another consideration is that the project members do not have a lot of experience working with computer vision. All in all, a camera module combined with computer vision is not an optimal choice as the computer vision libraries are not power efficient enough to run on the limited resources of the project and it's not in the scope of the project and the project members are not experienced in the computer vision area to develop or modify a computer vision library.

Lidar & Ultrasonic sensor

Lidar is a surveying method that measures the distance to a target by illuminating the target with laser light and measuring the reflected light with a sensor. Differences in laser return times and wavelengths can then be used to make digital 3-D representations of the target (Wikipedia contributors, 2020). Lidars tend to be expensive and usually sold in huge bundles of hundreds of thousands to big industries like car manufacturers. However, recently there has been development of smaller and cheaper Lidar modules that suit the needs of IoT (small module, low power usage):

1. tinyLiDAR - (MicroElectronicDesign, Inc., 2020)
2. LIDAR-Lite v3HP - (SparkFun Electronics, 2020)

LIDAR-Lite v3HP has a range of 40 meters and has field of view of around a $\frac{1}{2}$ degree - for distances lower than 1 meter the laser spread is the size of the lens and for higher distances the following formula can be used to calculate the beam diameter at a certain distance (SparkFun Electronics, 2018): Beam diameter = Distance / 100 (in whatever units the distance was measured). So even at the distance of 2 or 3 meters the beam diameter is very low - 2 or 3 centimeters. So LIDAR-Lite v3HP is not an optimal choice for the project needs.

However tinyLIDAR has a field of view of 25 degrees and the measurement distance from approximately 3 centimeters to 2 meters and a power consumption that's comparable to an ultrasonic sensor (24 mA average current during

measurement, compared to 15 mA working current of the HC-SR04 ultrasonic sensor). Even with those specifications it's still difficult to find sensor placements that would be able to cover the entire platform and not use an absurd number of sensors. As an example the sensor is placed on top of the container pointing downwards to the platform as indicated in figure 12.

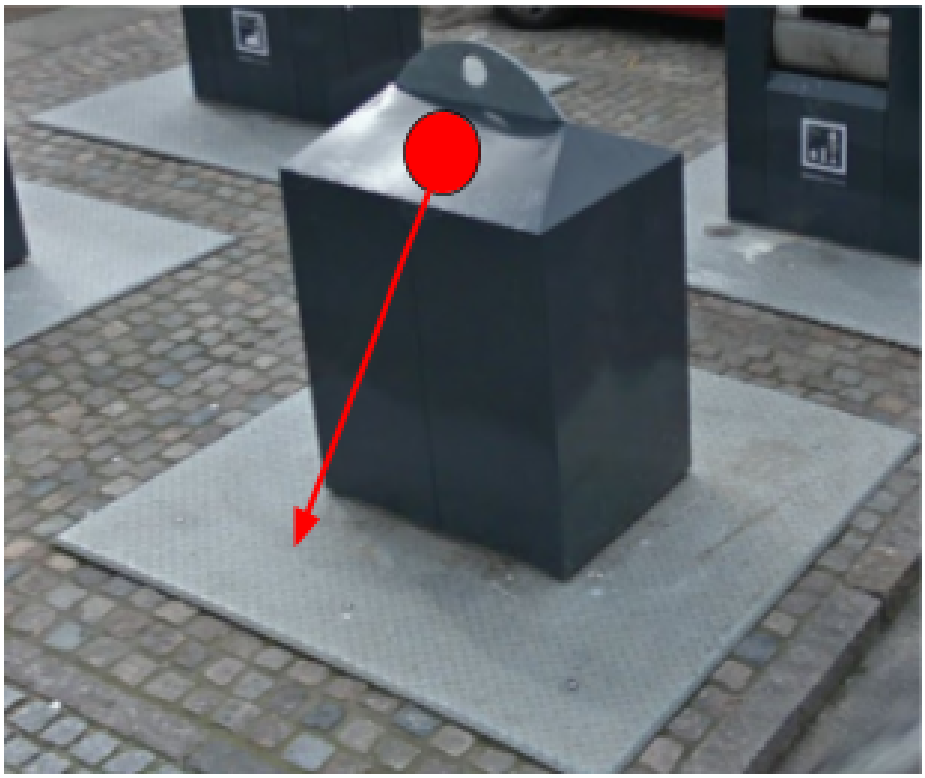

Figure 12: Single lidar positioning

Using a lidar resolution at distance calculator (georgehelser, 2018) the calculations that determine the maximum possible beam width at a certain distance can be performed (distance used is 0.5 meter as that's the distance between the sensor and a 0.3 meter height object placed on the platform) and shown in table 1

Scan field in degrees	Distance in meters	Number of spots	Beam width in meters
25	0.5	2	0.443
25	0.5	3	0.329
25	0.5	4	0.291
25	0.5	5	0.273
25	0.5	6	0.262

Table 1: Lidar resolution calculations

Even with the lowest possible number of spots chosen (2) the beam width is still only 44.3 centimeters at a 50 centimeter distance, since the platform is 160 centimeters wide, 4 of these sensors would be needed to cover one side. However if a servo motor is used for rotating the sensor more options become available. Ideally the servo motor with the sensor would be placed inside the container and the container sides would be drilled with holes. This way just one set of a servo motor and a sensor could do a full 360 degree spin and observe the whole platform. Unfortunately, drilling holes in the container sides is not allowed, so other options have to be considered. The second best option project members could think of was to place two separate motor and sensor sets on two opposite corners of the container. This allows one motor and sensor set to cover two sides of the platform. Figure 13 illustrates the sensor and motor set placed on the corner of the container and rotating and taking constant measurements to detect any objects. The motor will do a 180 degree turn to cover two sides of the platform. The same sensor and motor set is placed on the opposite corner of the container covering the other 2 sides of the platform.

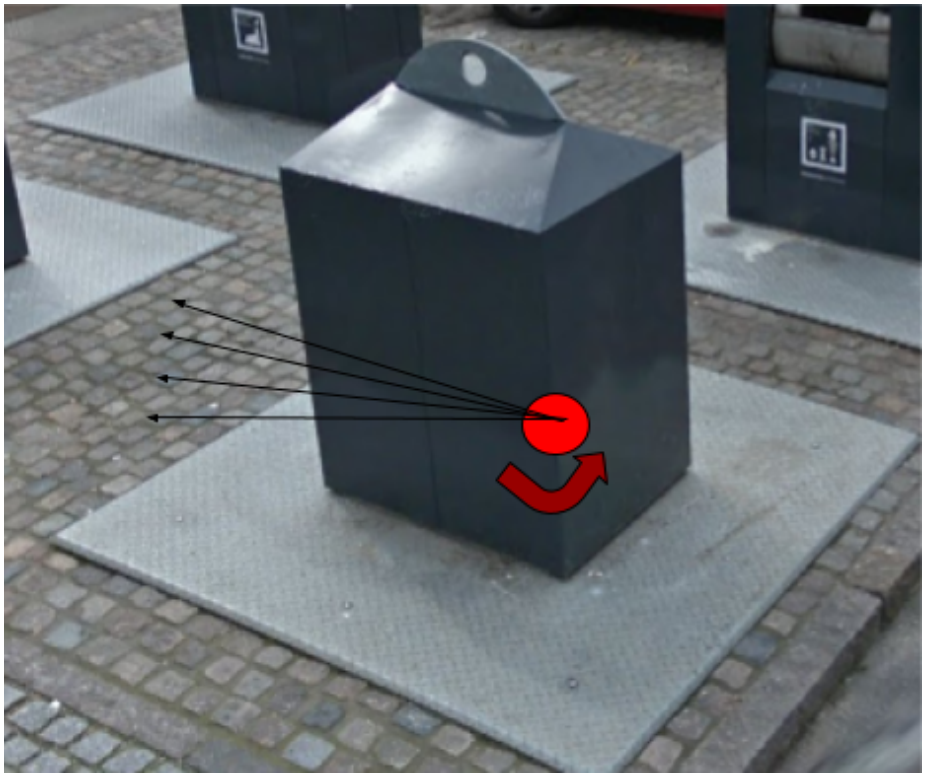

Figure 13: Rotating sensor

However, tinyLiDAR is not the only sensor that would work in this configuration, an ultrasonic sensor (specifically HC-SR04) would work really well too. On some level they can be compared as equal just with different specifications. tinyLIDAR has a higher sampling rate, bigger field of view, but it also has a higher power consumption. However they are also using different technologies - the tinyLIDAR uses infrared lasers and the HC-SR04 uses an ultrasonic sound - this difference affects how the sensors are impacted by the environment and how the emitted beams are reflected from objects. For now these differences will be ignored as this is a research project and if it turns out later that it would have made more sense to use the other sensor or perhaps a combination of both, the configuration can be updated easily as these sensors function similarly. Since the project team has access to the HC-SR04 ultrasonic sensor (tinyLIDAR has to be ordered and waited for to be shipped) and more experience working with ultrasonic sensors, the ultrasonic sensor will be the final choice.

2.4 Delimitations

Unless it has been stated otherwise, functionality is delimited due to time constraints.

The use cases for the system administrator actor were delimited. System administrator is responsible for configuring the detection system parameters, mainly the scanning times. However the need of such functionality is necessary only in production. As the purpose of this project is to find out if such detection system can be achieved, the configuration is left as a nice to have functionality.

In section 2.3 it was decided that the detection system will be focusing on an individual platform instead of trying to optimise a single sensor set for multiple platforms. On the other hand, it's possible to optimise the communication between the detection system nodes and the server. Since trash cans are placed in batches the system can be optimised by having a single "communication" node for the whole batch and a "detection" node for each trash can. The "detection" nodes would scan the platforms and send their results to the "communication" node through some low range communication protocol, after the "communication" node has collected the results of all the trash cans it will combine them into a single data set and send it out using the long-range LoRaWAN communication protocol. This way the long-range communication would be happening only once for the whole batch instead of once for every trash can in the batch. However, this kind of optimisation adds a lot of research to the project (like finding out which low-range communication protocol would be optimal for this project) and as the purpose of this project is to find out if minimum viable prototype is possible - this optimisation is out of the scope for the project.

2.5 Domain model

From the analysis above the following domain model entities are identified:

1. Trashcan's platform
2. Detection system
3. Kommune dashboard server

These entities are then modeled into a domain model diagram. Domain model diagram is shown in figure 14.

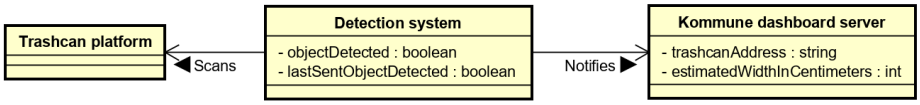

Figure 14: Domain model diagram

The domain model indicates that the detection system scans the trash can's platform for large objects and notifies kommune's dashboard server about any

detected objects (or notify about the objects that were removed that were previously alerted about) with the trash can's address and an estimated object's width, measured in centimeters.

2.6 Technology choices

As the analysis of what the system has to do has been completed and a sensor set that would be able to detect objects placed on the platform has been found, now a list of main technologies will be listed in this section with a short description and an explanation of why this technology was chosen.

1. LoRaWAN - chosen as a network protocol for communication between the detection system attached to the trash cans and the server.
    - Low power - the system is running on a battery and has to last a long time.
    - Long range - there are trash cans everywhere so long range is a necessity.
    - The project team has some experience working with LoRaWAN.
2. ATMega2560 microcontroller - chosen as the microcontroller controlling the sensors of the detection system and communicating with the server.
    - Able to interface with the chosen sensor set.
    - Has a deep sleep mode for power efficiency.
    - There's a LoRa module available for this microcontroller.
    - The project team has experience working with this microcontroller.
3. C and FreeRTOS - chosen as the software stack for the microcontroller.
    - C is a general purpose, procedural computer programming language.
    - FreeRTOS is a real-time operating system for microcontrollers.
    - Both technologies have huge communities and are a very popular and well tested choice in the embedded world.
    - Team members have experience working with both technologies and their combination.
4. C# .NET Core - chosen as the software stack for the application server
    - .NET Core is a general purpose cross-platform programming framework.
    - Teams members have experience working with it and it fits the goals of the application server well.

Perhaps these technology choices are not the most efficient ones, but as this is a research project and these technologies fit the project requirements they are the ones that will be used. If the product turns out to be a success they can be replaced with more efficient options.

3 Design

The following section describes the system design used to fulfill the functional and non functional requirements. It's split up into several subsections:

- Overall system design subsection.
- Detection system design subsection covers the design of the detection system that's attached to the trash can container and is responsible for detecting large objects.
- Application server design subsection covers the system design of the application server.
- The Middleware design subsection describes the communication between the detection system and the application server and between the application server and the Kommune's dashboard server.

References

albzn, 2019. *OpenCV on ESP32*. [Online; accessed 15-April-2020]. Available at: <https://www.reddit.com/r/esp32/comments/bheh6j/opencv_on_esp32/>.

Ayuso, G., 2018. *OpenCV and ESP32: Moving a Servo With My Face*. [Online; accessed 15-April-2020]. Available at: <https://dzone.com/articles/opencv-and-esp32-moving-a-servo-with-my-face>.

georgehelser, 2018. *LIDAR Resolution at Distance Calculator*. [Online; accessed 15-April-2020]. Available at: <https://precisionlaserscanning.com/2018/05/lidar-resolution-at-distance-calculator/>.

MicroElectronicDesign, Inc., 2020. *tinyLiDAR*. [Online; accessed 15-April-2020]. Available at: <https://microed.co/product/tinylidar/>.

Rockwell Automation, 2006. *Installation instructions PHOTOSWITCH Bulletin 45PVA Part Verification Array*. [Online; accessed 15-April-2020]. Available at: <https://literature.rockwellautomation.com/idc/groups/literature/documents/in/45pva-in001_-en-p.pdf>.

— 2020. *Light Arrays*. [Online; accessed 15-April-2020]. Available at: <https://ab.rockwellautomation.com/Sensors-Switches/Light-Arrays>.

Sick Sensor Intelligence, 2020. *Array Sensors | SICK*. [Online; accessed 15-April-2020]. Available at: <https://www.sick.com/dk/en/registration-sensors/array-sensors/c/g130174f>.

SparkFun Electronics, 2018. *LIDAR-Lite v3HP Operation Manual and Technical Specifications*. [Online; accessed 15-April-2020]. Available at: <https://cdn.sparkfun.com/assets/9/a/6/a/d/LIDAR_Lite_v3HP_Operation_Manual_and_Technical_Specifications.pdf>.

— 2020. *LIDAR-Lite v3HP - SEN-14599 - SparkFun Electronics*. [Online; accessed 15-April-2020]. Available at: <https://www.sparkfun.com/products/14599>.

Wikipedia contributors, 2020. *Lidar — Wikipedia, The Free Encyclopedia*. [Online; accessed 15-April-2020]. Available at: <https://en.wikipedia.org/wiki/Lidar>.