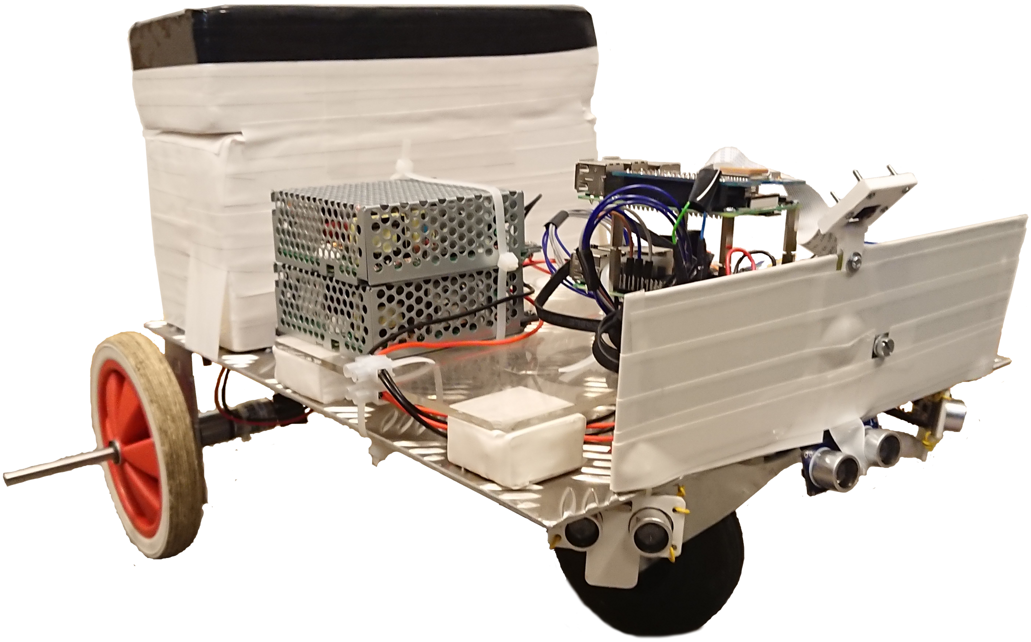
CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG



Smart robot lawn mower

Robot lawn mower without need for a boundary cable around the lawn

Jesper Mejervik Derander, Petter Andersson, Eric Wennerberg, Alex Nitsche, Erik Moen, Filip Labe

Bachelor thesis at the institution of Computer Science and Engineering

DATX02-18-05

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2018

Bachelor thesis: DATX02-18-05

Smart robot lawn mower

Robot lawn mower without need for a boundary cable around the lawn

Jesper Mejervik Derander
Petter Andersson
Eric Wennerberg
Alex Nitsche
Erik Moen
Filip Labe

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2018

Smart robot lawn mower
Robot lawn mower without need for a boundary cable around the lawn

Supervisor: Sven Knutsson, Department of Computer Science and Engineering
Examiner: Arne Linde, Department of Computer Science and Engineering

Bachelor Thesis: DATX02-18-05
Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A picture of the finished prototype

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2018

Smart robot lawn mower
Robot lawn mower without need for a boundary cable around the lawn

Jesper Mejervik Derander, Petter Andersson, Eric Wennerberg, Alex Nitsche, Erik Moen, Filip Labe

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg

Sammandrag

Robotgräsklippare som i nuläget finns tillgängliga på marknaden använder i de flesta fall en avgränsande kabel som installeras vid gräsmattans gränser och runt hinder, detta för att roboten ska stanna på gräset och att inte kollidera med några föremål. Installationen av denna kabel kan vara både kostsamt och tidkrävande samt svår att ändra på när den väl är på plats.

Resultatet av detta projekt är en prototyp som navigerar på gräsmattan utan att behöva använda en avgränsande kabel. Istället används en kombination av kamera, GPS och ultraljud sensorer. Prototypen kan navigera över ett avgränsat område och upptäcka objekt. Den kan inte upptäcka icke fysiska gränser som till exempel ej synliga tomtgränser.

Slutatsen av projectet är att teknikerna som används är användbara för att utveckla en robotgräsklippare, men att de alla behöver utvecklas. Framförallt GPS:en behöver ersättas eller förbättras för att uppnå en tillräcklig gränsavkänning.

Abstract

Robot lawn mowers currently available on the market mostly use a bounding cable that is installed at the edges of the lawn and around obstacles. This allows the robot to stay on the grass and not hit any obstacles. Installing this cable can be costly and time consuming, and it is difficult to change once in place. The purpose of this project is to evaluate if it is possible to develop a product that does not need the bounding cable.

The result of this project is a prototype that navigates without the need of a bounding cable. Instead it uses a combination of a camera, GPS, and ultrasound sensors. The prototype can sufficiently navigate on a closed area and detecting obstacles. It can not detect non physical edges for example not visible property lines.

The conclusion reached is that the techniques used in this project is useful for developing a robot lawn mower but that they all need some adjustment. Especially the GPS needs to be replaced or improved to gain the edge detection correctly.

Keywords: Robotic Lawn Mower, Navigation, GPS, Autonomous, Grass, Ultrasound.

Preface and acknowledgements

This report is part of a bachelor's project performed at Chalmers University of Technology during the spring 2018. The project is required to gain the bachelor degree and this thesis is supposed to describe the projects development and result. Four of the students in the project attends Chalmers University of Technology and two attends the University of Gothenburg.

We would like to direct a special acknowledgement to our examiner Arne Linde and Lars Norén for helping us with this project, and also Sören Kjellberg and Anders Johansson from Erasteel Kloster AB for helping us to manufacture the wheel axle and providing us with material and tools.

Contents

List of Figures

List of Tables

1
Introduction

1.1 Background

The first robot lawn mower was patented in 1969 by an American named S. Lawrence Bellinger [1]. The robot used a bounding cable to inform the robot where the borders of the lawn were placed. Within this area the robot navigated randomly across the lawn. Bellinger's robot was called 'Mowbot' and it was originally priced at $795, which would be equivalent to approximately 50 000 SEK today [2].

After the invention of Bellinger's robot several competitors launched their own robot lawn mowers, but none of the robots built at this time were successful. Some of the early robots also had problems with charging and water resistance. [3].

In 1995, the Swedish company Husqvarna launched the worlds first solar powered lawn mower [4]. The product was named 'Solarmower' and was Husqvarna's entry into the robot lawn mower market. The company continued to develop their products and between 2011 to 2015 the growth has accelerated enormously in this product category [5][6].

In today's market most robot lawn mowers uses a boundary defining cable, just as Bellinger did in 1969. This cable needs to be installed around the edges of the lawn as well as around static obstacles. This leads to a high price for installation and a bigger threshold for consumers if they would like to install it themselves. For Husqvarna's 'Automower' it takes about 2-5 hours to do the installation and setup[7]. Because of this manufacturers has been trying to develop another solution for years[8][9].

1.2 Purpose

The purpose of this project have been to build a prototype of a robot lawn mower which can navigate on an area of grass, without the need of a bounding cable installed at the edges. The finished prototype will be evaluated on how well it can stay within the bounds of the area.

1.3 Goal

The end goal has been that future robot lawn mowers may use the results of this project to help eliminate the need for a bounding cable. This would reduce installation costs and make the robot lawn mowers available to a larger market. To aid in this and make the end goal more manageable, it was divided into sub-goals:

- The prototype should be able to detect obstacles and take actions to avoid them.
- The prototype should be able to stay within the bounds of the lawn.
- The prototype should be comparable in price to commercially available lawn-mowers.

1.4 Delimitations

Since the focus of this project lies at the navigation of a robot lawn mower several delimitations regarding the prototype have been put in place.

- The prototype shall not be able to cut grass.
- A dedicated charging station shall not to be constructed.
- The area of mowing shall be limited to what existing robot lawn mowers can handle.
- The prototype shall not need to manage difficult terrain, such as large inclines and excessively uneven surfaces.

1.5 Method

This project will follow a method described in this section. The method consists of; an information gathering stage, a design stage, a construction stage, a testing and evaluation stage, and a discussion stage.

1.5.1 Information Gathering

In this stage, information from different sources will be collected. General information about robot lawn mowers will be collected as well as general information regarding systems and theories that could be of relevance later in the project. Not all information can be collected from scientific sources but all sources must be checked and deemed trustworthy.

Specific information will be gathered at later stages when needed in order to avoid wasting time on collecting information that might not be used during the project. Information collected at later stages must also be checked and deemed trustworthy.

1.5.2 Design

When sufficient information has been collected, the project enters a design stage. There will be more than one design stage due to more information being collected at later stages, that in turn can lead to other design choices being more suitable than previously assessed ones.

In the design stage design-propositions will be made using the gathered information. The propositions will be discussed before a decision is made, propositions might be altered or dropped but if a proposition is passed it will be constructed in the construction stage.

1.5.3 Construction

In the construction stage, the propositions made in the designs stage is made into reality and some general tests is carried out (i.e making sure a piece of equipment works as it should). If a design-proposition can not be realized due to unforeseen circumstances, lack of information, time constraints, or critical design flaws the proposition is dropped. After a proposition has been handled more information is to be collected and the method returns to the information gathering stage again.

Thus the construction stage is the final stage in the information-design-construct loop. This iteration process is repeated until the project is finished.

1.5.4 Evaluation and Discussion

Finally when the project is finished the project and its outcome are evaluated and discussed. This stage includes evaluation in the form of test results on the projects outcome and the project as a whole is discussed by the group, reflections are given on the process and the the result.

1.6 Current market

Husqvarna, the leading retailer of robot lawn mowers in Sweden has many different models, ranging in price from 12000 SEK up to 42000 SEK. Lets compare the cheapest, the most expensive and an average model to see what they offer. Below are three models from Husqvarna; the cheapest, the most expensive, and an average model with some relevant info about them.

Figure 1.1: Husqvarna Automower 105© (image property of Husqvarna©)

Specification	Value
Surface capacity	600 m^2
Working time on charge	65 min
Charging time	50 min
Highest sound level	61dB(A)
Surface incline capacity	<25%
Ultrasonic sensor	No
GPS-supported navigation	No
Price	12.500 kr

[10]

Table 1.1: Specifications for the Husqvarna Automower 105

Figure 1.2: Husqvarna Automower 420© (image property of Husqvarna©)

Specification	Value
Surface capacity	2200 m²
Working time on charge	75 min
Charging time	50 min
Highest sound level	58dB(A)
Surface incline capacity	<45%
Ultrasonic sensor	No
GPS-supported navigation	No
Price	23.900 kr

[11]

Table 1.2: Specifications for the Husqvarna Automower 420

Figure 1.3: Husqvarna Automower 550© (image property of Husqvarna©)

Specification	Value
Surface capacity	5000 m²
Working time on charge	270 min
Charging time	60 min
Highest sound level	61dB(A)
Surface incline capacity	<45%
Ultrasonic sensor	Yes
GPS-supported navigation	Yes
Price	41.500 kr

[12]

Table 1.3: Specifications for the Husqvarna Automower 550

Looking at the 420 and the 550 version, the only significant difference between them is the surface capacity, battery charge, and the addition of various sensors. It is reasonable to assume that the addition of ultrasonic sensors and GPS-supported navigation would not almost double the price, although it would of course contribute. It could be attributed to a more expensive battery technology and better motors. The same can be seen when comparing the 105 and 420 versions, the biggest differences is the battery, and because of that there is a difference in surface capacity. So it would seem that the main cause behind the price difference is due to the battery, and only in part due to other auxiliary systems like extra sensors and potential extra software.

1.7 Previous research

There has been academic research on robot lawn mowers, some of which is also from students at Chalmers such as these ones [13] [14].

The research group [13] made a robot lawn mover that was capable of planing its route using a local positioning system. The system they used consisted of 4 sensors that served as known points. The robot ask all of the 4 sensors for conformation and records the time it took for the signals to return. With this information, using trilateration the relative position of the robot can be determined [13, p. 31].

The other research team [14] opted for a GPS based solution. By connecting a well performing GPS receiver to their robot they managed to receive highly accurate positional data from preexisting GPS-satellites. For further accuracy the research group used whats called "Real Time Kinematic" (RTK) positioning to enhance the accuracy of the robots position.

2
Theory

Chapter two contains explanations of the theory this project is based on. If the reader is unfamiliar with some concepts regarding these subjects, this section will give a quick introduction of triangulation, trilateration, ultrasound, and GPS. Component specific information can be found in appendix A.

2.1 Trilateration

Trilateration is a geometrical method used to calculate the coordinate of a point by measuring distances between already known coordinates and the unknown point. An intuition will be given below and an in-depth explanation of trilateration can be found in [15].

Suppose that there only was one known coordinate, with a known distance to the unknown coordinate. Then the unknown coordinate would be somewhere on the circle in figure 2.1 since all the points on the circle has the same radii (the radii is the measured distance).

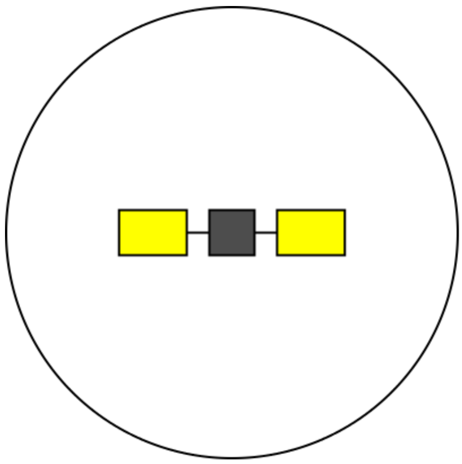


Figure 2.1: A satellite representing a known point, the unknown point is somewhere on the circle.

Suppose there are two known coordinates and their measured distances to the unknown coordinate. Then the unknown coordinate must be on one of the two intersections of the circles as seen in figure 2.2. This is because it is only on these points the distances between the unknown coordinate and the known coordinates are the same.

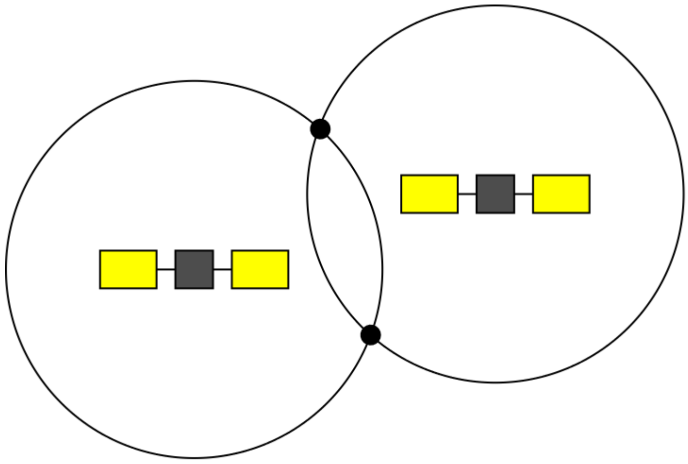


Figure 2.2: With two known points and measured distances, the unknown must be on one of the intersections of the circles.

If a third known point and a third distance is added it's possible to determine the position of the unknown point as seen in figure 2.3.

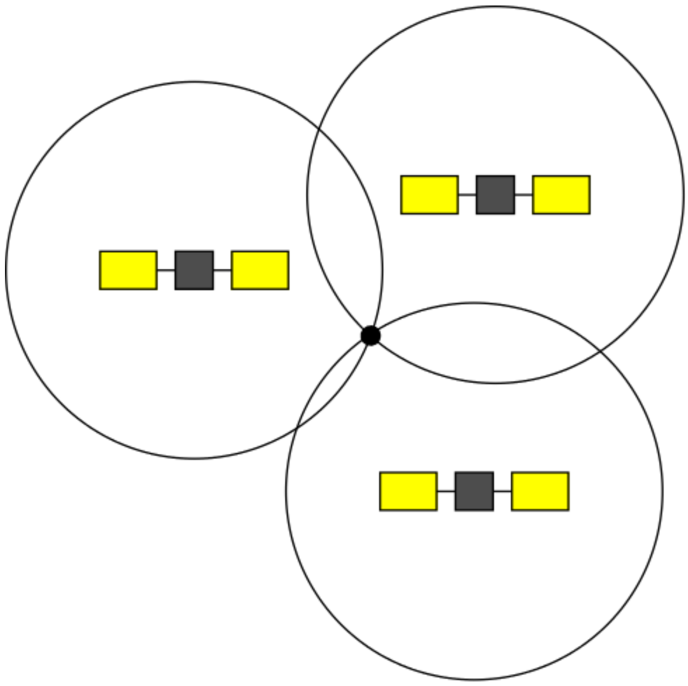


Figure 2.3: With three known points and measured distances, the unknown point can be found.

In order to calculate the coordinates of the unknown point, the well known point-distance formula is used on each of the known points to the unknown point in an equation system and solve for the unknown point's x and y coordinates, see [15] for details.

However even if there are three known points there is no guarantee that a unique coordinate will be found. Figure 2.4 shows a case with three known coordinates and their distances that yields two possible coordinates.

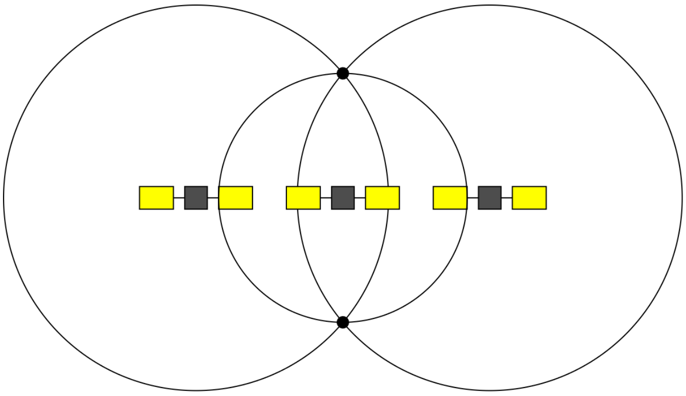


Figure 2.4: An example of an edge case with three known points

2.2 Ultrasonic sensing

Ultrasonic sensors use the properties of sound waves to measure distances. A pulse of high frequency, over 18 kHz, sound is generated. By analyzing properties of the echo generated by objects in the sound wave's path, the distance to the objects can be calculated[16]. The formula for calculating the distance is as follows:

$$d = (t * v)/2 \tag{2.1}$$

Where t[s] is the time from the unit sending out a pulse to receiving an echo, v[m/s] is the speed of sound in air and d[m] is the distance to the obstacle.

A margin of error is introduced by the fact that the formula doesn't take the following things into account; the material of the object which generates the echo, the angle the object is oriented relative to the sound wave, the temperature, and humidity of the air[16].

2.3 Global Positioning System (GPS)

A GPS unit can within a few meters in good conditions calculate its position anywhere on Earth at any time[17]. This is primarily achieved with a network of satellites orbiting around Earth. These satellites are constantly emitting high frequency radio waves, which the receiver uses to calculate its position. These signals contains information on where the satellites are positioned and the time when the signal was emitted. Using this information, the GPS unit can calculate its position on Earth using trilateration (section 2.1).

To accurately measure the position, it needs information from at least four different satellites, but this is no problem as the satellites orbits are configured in such a way that there are always at least four satellites visible from any point on Earth at any given time[18]. The accuracy of the position also depends on what the weather conditions are and what the surrounding area looks like. If the weather is cloudy or if it's very humid, the signals from the satellites can be distorted and interfered with, and this would give inaccurate readings[19]. The same problem can be found if the receiving unit is surrounded by tall and large objects such as; buildings, trees, or mountains. The signal may then bounce of such objects, which can distort the message and will make the signal take longer to arrive [19].

3
Identifying problems

In order to fulfill the purpose of the project, a set of problem-areas were identified. In this section these areas are discussed and formalized.

3.1 The Obstacle Detection Problem

Other commercially available robot lawn mowers can sense flower beds, trees and other static obstacles residing on the lawn, as marked by the bounding cable. These obstacles can be sensed very accurately, often down to mere centimeters. This paper refers to this short range sensing as micro-positioning. This could also be used to avoid more temporary obstacles, like humans and animals.

On top of the accuracy around static obstacles residing on the lawn, there is another feature that the bounding cable provides. Not all lawns have entirely physical boundaries, like fences and flowerbeds, but simply an abstract property line. The ability to sense these borders are henceforth referred to as macro-positioning.

3.2 The Navigation Problem

While a solution to the obstacle detection problem would provide information about the surrounding environment, the prototype can not automatically make decisions based on this information. And so another sub-problem exists, the navigation problem. When information has been gathered from the different sensors, the prototype must be able to interpret the data in order to effectively navigate the lawn.

3.3 The Movement Problem

The prototype has to have a motor to enable it to move around the grass. In order to choose a fitting motor there are multiple performance attributes that has to be calculated or approximated, such as required RPM of motor and maximum and minimum torque required. The RPM of the motor has to be high enough to make the prototype move in "moderate" speed, the maximum and minimum torque the motor can deliver has to be high enough to be able to power the prototype forward on flat ground and when in an "moderate" incline.

3.4 The Communication Problem

As discussed in section 3.2 the information from the sensors has to be gathered and processed, which can be handled by micro controllers. Since its likely the system will utilize several sensors it is also likely that multiple micro controllers needs to be used.

In order for several micro controllers to function as a system some form of communication between them is necessary. The amount of data that is transferred and the transfer rate between each controller has to be taken into account when deciding the medium and protocol of data transfer.

3.5 The Power Problem

If the prototype uses a wired power supply it runs the risk of cutting it in pieces or becoming entangled in obstacles. This is why the power supply has to be highly mobile in order to be considered for the robot lawn mower.

It is also desired that the power supply should have a long lifespan before it has to be replaced or undergo maintenance. The power supply should also have a sufficiently high capacity to enable a high up time, in order to cut the grass faster than the it can regrow.

4

Sensors and Navigation

4.1 Sensors

To solve the obstacle detection problem (section 3.1) entails both deciding which sensors to use, and how to clean up the data they provide. There are several alternatives to the bounding cable and many of them have been considered for the prototype. This section is focused on motivating the choice of sensors, and also go deeper into how the sensors that were chosen works to fulfill their purpose.

4.1.1 Micro-positioning

There are several systems and techniques that could acceptably solve micro-positioning. The different alternatives considered where LIDAR, camera, infrared and ultrasound.

- Light Detection and Ranging, LIDAR uses the same technology as radar but sends laser pulses instead of sound waves to measure the distance [20]. It's typically used in autonomous cars to sense their surroundings. They are however very expensive and heavy, making them unsuitable for the prototype.
- Infrared sensors does not have these flaws, its both light and inexpensive. There are however several things that affect accuracy of infrared sensors, for example the brightness of the sun, which means that the reliability of infrared sensors are unacceptable.
- Ultrasound sensors are as inexpensive and light as infrared sensors, but its sensitivity is not as dependant on external factors. It is however not always reliable on uneven and porous surfaces.
- Cameras are fairly inexpensive, but requires more computing power to be useful. It can be very accurate, depending on the software used.

With this in mind, the prototype was constructed to use both ultrasound and camera for micro-positioning. A combination of these techniques gives better accuracy than if they were to be used separately.

4.1.2 Macro-positioning

An acceptable macro-positioning solution can't be something physical that is placed on the lawn, like a fence, as this would require to much work to install and disrupt the look of the lawn. This leaves two possibilities, to bury something beneath the surface or use airborne systems. The boundary cable is an example of a buried solution, however it's what this project aimed to solve in the first place, as stated in 1.2.

This leaves only one type of system for macro positioning, namely airborne systems. Some of the airborne systems investigated are radio triangulation (both with transceivers located on the lawn or public transceivers located in the nearby area), radio trilateration (with transceivers on the lawn), and GPS.

Radio trilateration, using transceivers placed on the lawn, would give fairly high accuracy but would require the user to install equipment on the lawn. Using public radio stations would circumvent this need, however it would negatively impact the accuracy of the system due to the extra distance. Radio trilateration shares the same issues.

GPS does not require any equipment to be installed or maintained, as it depends on existing satellites (section 2.3) while still having potentially high accuracy. However the accuracy is dependant on external factors like the weather and on cloudy days the accuracy can be sub-optimal.

To achieve portability as well as ease of installation and lower costs, GPS was chosen for macro-positioning.

4.1.3 Ultrasound

The robot has three ultrasonic sensors (for component details, see A.3) mounted to the front of the robot, as seen in figure 4.1.

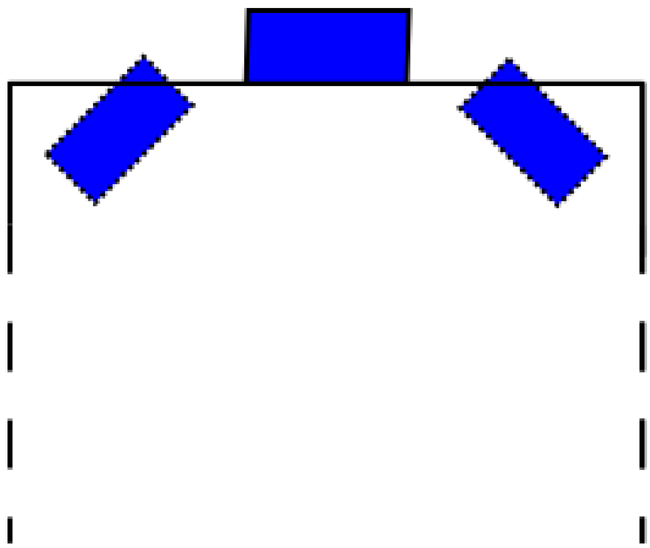


Figure 4.1: Positioning of the ultrsonic sensors

The ultrasonic sensors are controlled using an Arduino Nano (a programmable micro controller). Using the micro controller the ultrasonic sensors are configured as follows; the micro controller tells the left (figure 4.1) sensor to send out an ultrasound via rapidly toggling the sensors trig-pin to "high" and then just as quickly toggle it back to "low". The sensor then sets it's echo pin "high" for 4 milliseconds and sends out 8 short ultrasound bursts. When the sound bounces back or when the 4 millisecond timeout has passed the echo ping is set to "low" and by having the micro controller measure the duration the echo pin was "high" it's possible to calculate the distance using the speed of sound.

The output of the ultrasound sensors is a 3-tuple of boolean values corresponding to whether the left, center and right sensors is blocked or not. In case all sensors report there is nothing blocking them the output will be $(False, False, False)$

4.1.4 Camera

The robot uses a Raspberry Pi Camera Module Rev 1.3 (appendix A) to solve part of the micro-positioning problem. The camera enables the robot to detect changes in material, e.g. from a green lawn to black dirt in a flower bed. The raw images are analyzed via a average-color algorithm to decide if the material is grass or not. This information is then used to decide whether the prototype can move forward or if it has to turn.

When the picture is taken it is resized to a 14x9 pixel image, with bi-linear interpolation [21]. Both the interpolation method and size was determined by testing on several hundred images. Each pixel in the resized image corresponds to the average color in approximately 4.5 cm^2 surface patch in front of the robot. The average color is converted to a HSV (Hue Value Saturation) [22] color model and then compared to a special "grass green" color. HSV was used because its model is closely aligned with how humans perceive color and is therefore easier to work with. The values in HSV chosen to represent possible grass colors can be seen in table 4.1 and an example in figure 4.2.

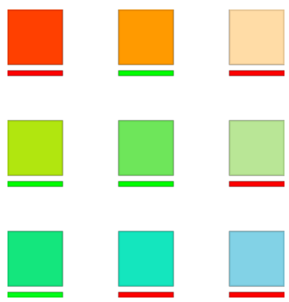


Figure 4.2: Example colors and beneath them a color indication of them being "grass green" or not.

Component	Lower bound	Upper bound
Hue	36°	150°
Saturation	35%	100%
Value	0%	100%

Table 4.1: HSV bounds for whats considered "grass green".

After comparing the colors a binary image is produced with dimensions 14x9. With a value of "1" representing the pixel is grass and "0" representing the pixel is not grass. The output from this step can be seen in figure 4.3

The binary image is then split into three vertical strips, resulting in one 4x9 (right most strip) and two 5x9 pixel strips which will be analyzed individually. In each strip, starting from the bottom of the strip, each row (4x1 and 5x1 pixels respectively) the ratio of "1"s to "0"s is calculated. If the ratio in a row is lower than a

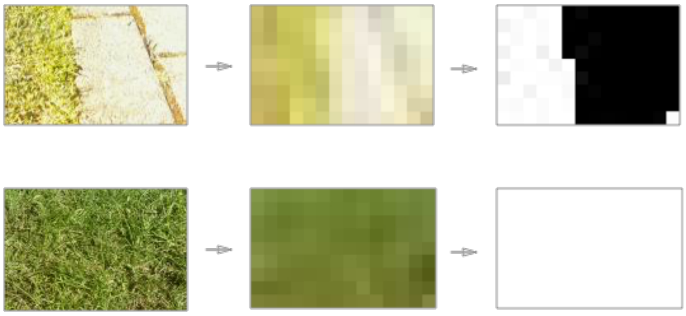


Figure 4.3: Original image (left), downsized (middle), green-analyzed(right)

size threshold, the row is not grass and therefore should not be driven on by the robot. To avoid the robot stopping too far from the edge a proximity threshold is also used. If the row index (starting from "0" at the bottom of the strip and "1" at the top) is above the proximity threshold the row is ignored. The values of the size and proximity thresholds were decided based on testing, and ended up at 0.4 and 0.2 respectively.

The output of the camera sensor is a 3-tuple of boolean values corresponding to the analysis of the 3 strips. If the area in front of the robot was clear grass, the output would be $(True, True, True)$. But if there was asphalt in the left side of the image, it would return $(False, True, True)$.

Algorithm 1 Camera algorithm

```
threshold ← 0.4
proximity ← 0.2
for row r in image, starting from the bottom do
    greens ← 0
    for binr do
        if isOkColor(b) then
            greens ++
        end if
    end for
    if (greens/r.length) < threshold && (r.index/image.height) < proximity then
        return false
    end if
end for
return true
```

4.1.5 GPS

To detect and create artificial borders, a GPS unit for the Raspberry PI is used. Borders are considered artificial when there is no clear physical difference between two areas, but things like land ownership can create borders regardless.

This project aimed to solve this with GPS. When the prototype is installed, the borders of the intended area on which it will operate on is mapped out and stored on it. While the prototype is running it will continuously take measurements to check what its coordinates are and if they are within the given area. If the prototype is near the border, it will recognize this and take actions to prevent it from leaving the area.

To accomplish this the prototype has been outfitted with the Adafruit Ultimate GPS HAT (appendix A.5) for the Raspberry PI computer. It continuously receives data from satellites within range, calculates where the unit is, what speed and direction it's going, then it transmits it to the Raspberry PI over UART (Universal Asynchronous Serial Transceiver, a hardware device used for asynchronous serial communication between components).

In order for the GPS unit too work properly and handle the data correctly, it must first get a fix. The prototype gets a fix when it has gathered enough initial data from visible satellites to be able to measure its position. The time to get a fix varies depending on what the surrounding area looks like. If the unit is located in a large open area, the time to get a fix is much lower than if it was surrounded by tall buildings. The time to get a fix and more importantly the accuracy of the data also depends on the weather conditions, various noise and interference, and the current configuration of the satellites. In optimal conditions the GPS unit can get a fix in under 30 seconds but in extreme cases it could take more than 30 minutes to get a fix.

In order to determine whether or not the prototype is within the intended borders of the lawn, an algorithm is used which can determine if a given point is within a polygon[23]. The installing the robot and the lawn is mapped out is done through an android application which gives a user the ability to control the robot and set up each node of the polygon, this is described in section 5.2. When the prototype wants to know if it's close to the edge of the lawn, it reads its current coordinates and run these through the algorithm.

This is done on a Raspberry PI that is separate from the one that handles the navigation. The Raspberry PI with the GPS continually sends the prototype's latitude and longitude, as well as if it has determined if it will hit a border. The output of the GPS system is $(Latitude, Longitude, True/False)$.

4.2 Navigation

Based on the data the robot collects from its various sensors, it has to make decisions to avoid obstacles and also make sure it can cover the entire area its supposed to cut.

4.2.1 Navigation method

Lawn mowers currently on the market uses random walk to make navigation decisions on the lawn. Random walk means that the robot, when encountering an obstacle, will spin around its own axis a randomized amount of degrees and will then continue moving forward until a new obstacle is encountered.

This seems like it would not result in acceptable coverage of the lawn, but as it turns out, given enough time it will actually cover the entire lawn. To test this a simple simulation was made in the beginning of the project, shown in figure 4.4. In the simulation the grass grows at 2.5% of the acceptable height every $10,000$ seconds which means that the lawn would grow too tall in approximately 5 days. These metrics are based on how fast grass grows during mid summer. The simulated lawn mower moves at 1 m/s. The simulations shows that a $20,000$ m^2 lawn can be adequately mowed while utilizing random walk as a navigation method.

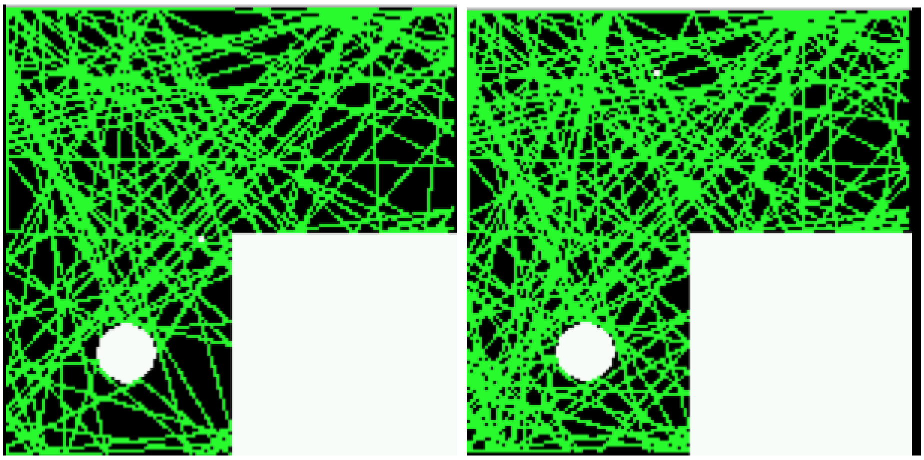


Figure 4.4: Screen-shots from simulation of random walk.

4.2.2 Algorithm

When using random walk there is only one more decision that needs to be made, whether to turn or not. This decision is in turn dependant on all the different sensors. The output of them (as described in sections 4.1.4 4.1.3 and 4.1.5) needs to be coalesce into one verdict. The GPS sensor is treated differently from the other two sensors since if GPS says it can't to further it means that the output of the other sensor readings is irrelevant. The ultrasound and the camera are treated equally.

The readings from all sensors is converted to floats, which means that ($True$, $True$, $False$) would be converted to $(1.0, 1.0, 1.0)$. Readings from the camera and ultrasound are received with a frequency of approximately 35 readings per second. Both the camera and the ultrasound suffers from misreadings from time to time, taking this into account a sliding time window is used. The last 10 readings from both of these sensors is averaged to give consistent reading. This averaging will result in a 3-tuple of floats between -1 and 1. The minimum value in the tuple (as explained in section 4.1.3 4.1.4 the tuple represents Left, Middle ,and Right) is chosen as the final output from the camera and ultrasound. The minimum value is the relevant one since its doesn't matter where in front of the prototype the obstacle is. Readings from the GPS are received at a frequency of approximately 1 per second and do not suffer from spurious misreadings so only the last value is converted to a float and used as final output. How these values are used to make a navigation decision is described in algorithm 2.

Algorithm 2 Navigation algorithm

```
gps ← float[−1, 1]
ultrasound ← float[−1, 1]
camera ← float[−1, 1]
decision ← continue
if gps < .8 then
    decision ← turn_around
else if ultrasound < 0.0 then
    decision ← turn_around
else if camera < 0.0 then
    decision ← turn_around
end if
return decision
```

5

Auxiliary Systems

Although the focus of the project is the hardware and software described in 4, there are several auxiliary systems that has to be in place in a functioning prototype. This chapter will elaborate on these systems.

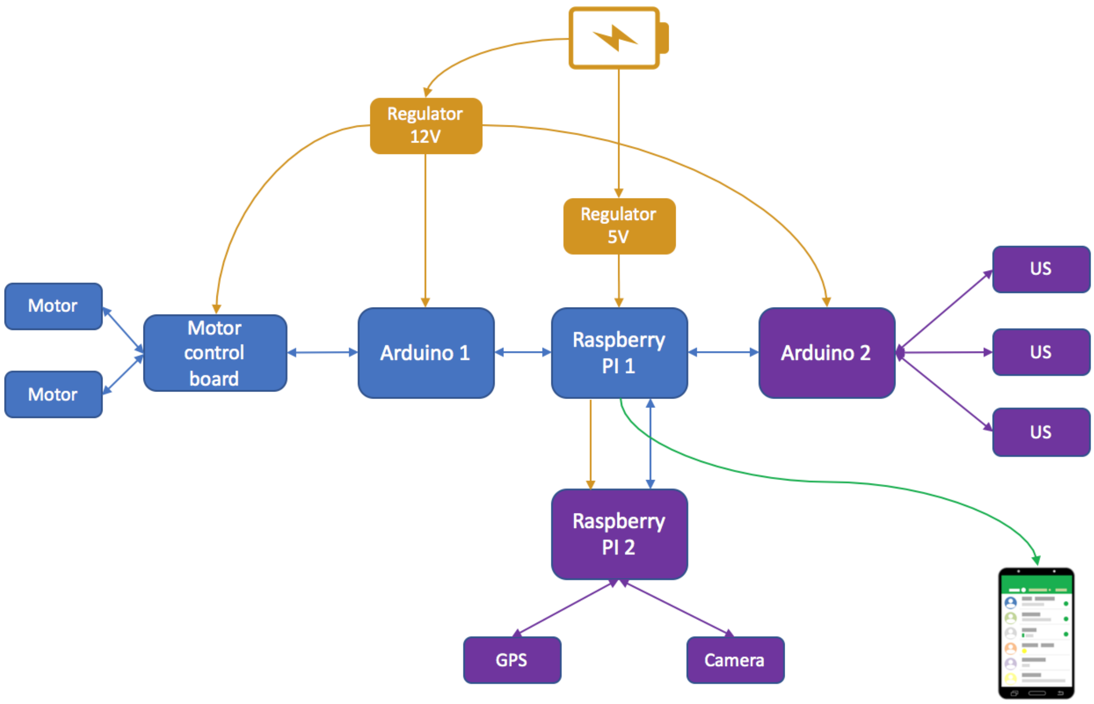


Figure 5.1: Overview of the components in the entire system

5.1 Movement

As concluded in 3.3, a motor with enough performance to match the requirements is needed, the performance requirements has to be calculated in order to choose a fitting motor. Also there has to be a system that can enable steering of the motor. This section aims to motivate the choice of motors, wheels and control system, and also further explain the functionalities of these components.

5.1.1 Motor and wheels

In order to choose a fitting motor for the project a simple mathematical model of the prototype was calculated. The weight of the prototype, the desired velocity and the radius of the wheels were all approximated in the beginning of the project (Table 5.1). Frictional force was calculated by using a grass coefficient μ [24].The mathematical model only takes into account the force of attraction of the robot and the frictional force at the grass, which makes it a highly simplified model of the actual physical forces acting on it, and therefore the model can only be used as a rough estimate of the actual performance needed by the motor.

Variable	Symbol	Value	Unit
Velocity	v	1	$[m/s]$
Weight	m	$4-6$	$[Kg]$
Wheel radius	R	0.075	$[m]$
Grass coefficient	μ	0.35	$-$
Earth gravity	g	9.82	$[m/s^2]$

Table 5.1: Estimated specifications of the robot lawn mower

$$F_g = m * g \qquad [N] \tag{5.1}$$

Formula 5.1: Force of attraction

$$F_f = F_g * \mu \qquad [N] \tag{5.2}$$

Formula 5.2: Friction force at the grass

$$\tau = F_f * R \qquad [Nm] \tag{5.3}$$

Formula 5.3: Desired torque

$$n = \frac{v * 60}{2 * R * \pi} \qquad [RPM] \tag{5.4}$$

Formula 5.4: Desired speed

When analyzing the specification that was calculated and approximated in 3.3 the conclusion was that a 12V DC motor with high torque and low RPM would be preferable. The choice of using a 12V motor were because some of the other components was designed to operate at 12V and therefore it would be easier to supply the components with power since they would be running at the same voltage and it would require less voltage managing components for the individual components to be able to work properly.

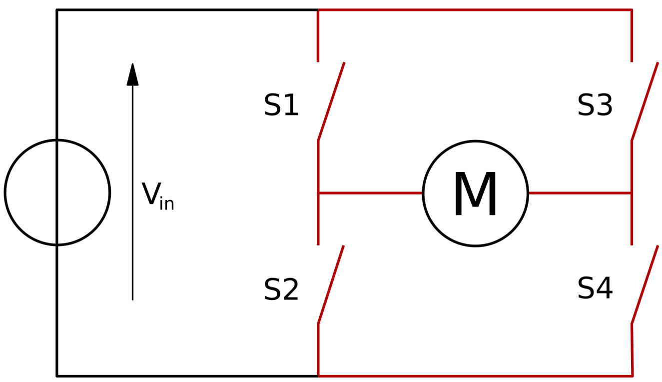
A planetary geared 12V DC motor was chosen to be the mechanical energy provider to the prototype. The motor can deliver high torque and enough speed while being a relatively small and cheap, which was preferable since the budget of the project was limited (See [25] for motor specifications).

The rear-wheels that were chosen to go with the motors was two hobby-wheels with flat rubber tires [26], and a radius of 70 mm. As for the front-wheel, a wheel that can rotate 360° was chosen [27] to enable the prototype to turn in narrow spaces, the radius of the front wheel are 50 mm.

5.1.2 Control board

The motor controller board that was used is the dual H-bridge L298 chip [28]. The L298 chip is mounted on a circuit with a heat sink, 5V regulator, output- and input pins. The inputs are four digital pins which are used to control the direction of the two DC motors and the last two pins are analogue pins which are used to set the demanded voltage to the motors.

A dual H-bridge is what enables the L298 to be able to run the motor both forwards and backwards. A dual H-bridge regulates in which way the current is flowing through the circuit and into the motor by using electrical switches. Depending on which direction the motors should rotate, the switches in the dual H-bridge can be changed (which are done with the digital pins). Doing this the current will flow the opposite way into the DC motor and the polarity will change and make the motor rotate the opposite direction [29].

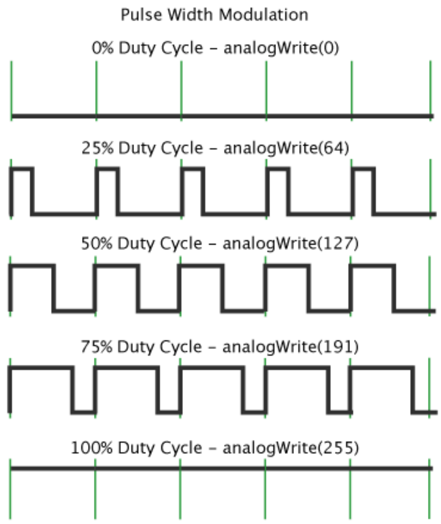


[30]

Figure 5.2: The H-bridge with switches S1-S4 used to change the polarity of the current into motor, M

As for the voltage regulation of the motors, which are done with two analogue pins, a PWM (Pulse modulating signal) signal is used to set the input signal. The PWM signal regulates the duty-cycle of the voltage by switching the power on and off. By alternating the duty-cycle (relation between on and off time) the average voltage can be regulated and it can then control the voltage that the motors are being powered with. By controlling the voltage fed to the DC motors the torque and speed of the motor can be regulated by simply changing the value of the analogue PWM signal.

A lower value of the analogue PWM signal will correspond to a lower voltage fed to the motors, and a higher value to a higher voltage fed to the motors [31].

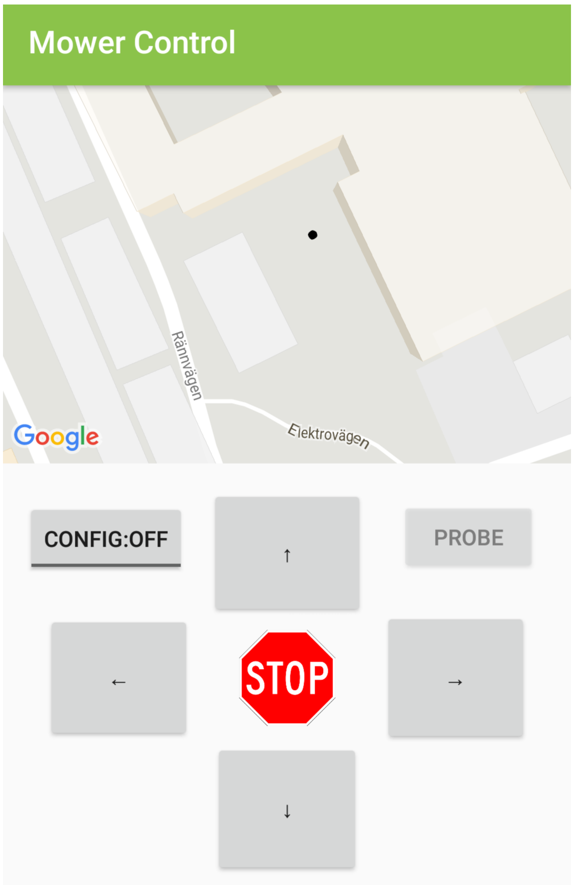


[32]

Figure 5.3: The duty-cycles corresponding to different analog values.

5.2 Application

In order to configure the GPS and setup the intended area the prototype should be contained by, an Android application is used. This area is configured by giving the user total control over the prototype, who then sets up each corner of the area through a probe action.



Map data ©2018 Google

Figure 5.4: Interface the android application

The black dot on the map in figure 5.4 shows the current location of the robot. In order to begin the configuration process the config button has to be pressed, then the user has full control over the prototype. The user then drives the prototype around the lawn and periodically probes the current coordinates, which are stored on the prototype. These points form the area that the prototype is not allowed to leave.

5.3 Exchange of data

The robot has several components that need to communicate with each other, thus the robot has to handle the exchange of data between the components. In total there are six pairs of components that need to communicate with each other to some extent. The connections can be found in table 5.2.

No	Component	Direction	Other Component	Serial/Parallel
1	App	Sends to	RPI (Navigation)	Serial
2	RPI (Sensors)	Sends to	RPI (Navigation)	Serial
3	RPI (Navigation)	Sends to	RPI (Sensors)	Serial
4	RPI (Navigation)	Sends to	Arduino (Engine)	Parallel
5	Arduino (Ultrasound)	Sends to	RPI (Navigation)	Parallel
6	Arduino (Engine)	Sends to	Control board	Parallel

Table 5.2: Components that need to communicate with each other, the direction of the data and the interface type between them are shown in this table.

Connection number 1 in table 5.2 is a serial 802.11 (HTTP 1.0) connection between the app and the navigation Raspberry PI, the HTTP connection uses port 8085. The app acts as client and the Raspberry acts as the server for the HTTP connection. The app can send the following commands:

- /config/position – Sends a coordinate-tuple (latitude , longitude) position.
- /config/on – Sets the Raspberry PI to configure mode
- /config/probe – Makes the robot store its current position
- /config/off – Turns off configure mode
- /config/forward – Makes the robot go forwards

Connection number 2 and 3 are a serial 802.3 (HTTP 1.0) connection on port 8080 between the two Raspberry Pis. Here the communication goes both ways instead of just one-way as in connection 1. The sensor PI can send the following requests to the navigation PI:

- /gps – The navigation PI returns:
  - A coordinate-tuple (latitude , longitude)
  - A boolean representing it has been configured
  - A boolean representing it inside the area (according to the GPS)
- /camera – The navigation PI returns:
  - Three booleans, each representing that the robot can isn't blocked to the left, middle, or right side of the camera's field of view

Connection number 4 consists of a three bit wide parallel interface between the navigation Raspberry PI and the Arduino controlling the engine. A bit combination represents a specific movement command (table 5.3).

Connection number 5 consists of a three bit wide parallel interface between the navigation Raspberry PI and the Arduino ultrasound. Each of the bits represent an ultrasonic sensor and indicates that an obstacle is in front of the sensor.

Bit combination	Command
000	Stop
001	Move Forwards
010	Move Backwards
011	Turn Right
100	Turn Left

Table 5.3: The bit combination represents the command the Arduino receives, note that the combinations: '101', '110', and '111' is unused.

Finally connection number 6 in the table consists of a six bit wide parallel interface that the Arduino responsible for the engine uses to communicate with the control board from section 5.1.2.

5.4 Power

This section will cover the calculations of the total power consumption, the process of choosing a battery and voltage regulators that is needed for the project.

5.4.1 Power Consumption

To calculate the amount of time the battery could power the components of the prototype it was necessary to first calculate the power consumption of each component and then summarize it in order to obtain a value for the total power consumption of the prototype.

The values of the currents used to calculate the power consumption of each component was taken from the technical specifications of each component and serves as an approximation of the actual power consumption which fluctuates between max values and average values.

$$P = U * I \qquad [W] \tag{5.5}$$

Formula 5.5: Calculated electrical power by using the voltage (U) and current (I)

Component	Supplied voltage	Max Current	Avg Current	Max Power	Avg Power
Raspberry PI	5 V	2.5 A	0.40 A	12.5 W	2 W
Arduino	12 V	0.90 A	0.3 A	10.8 W	3.6 W
Motor	12 V	4.90 A	0.60 A	58.8 W	7.2 W

[33] [34]

Table 5.4: Power consumption of the components

Component	Quantity	Total Max Power	Total Avg Power
Raspberry PI	2	25 W	4 W
Arduino	2	21.6 W	3.6 W
Motor	2	117.6 W	14.4 W
Total		164.2 W	22 W

Table 5.5: Summarized power consumption of the components

5.4.2 Supply

In section 3.5 it was concluded that the power supply have to be mobile and have a long lifespan. The motors and the Arduino is running on 12V and therefor the battery was sought to have a minimum of 12V nominal voltage. With this in mind a LiPo battery with four battery cells, nominal voltage of 14.8V and a charge capacity of 5200 mAh was used. (See A.7).

Even though 12V nominal voltage should be enough to voltage for the motors and Arduino, the battery could drop down under the nominal voltage. That is why it was chosen to buy a battery with higher than 12V nominal voltage and use a DC/DC step-down circuit to ensure steady 12V fed to the motors.

The downside of LiPo batteries is that they are very sensitive of how they are discharged and how they are charged. Every cell of the battery has to be charged evenly with a balancing circuit and the discharging also has to be somewhat balanced. Because if one of the cells are being discharged to it's critical level the power drawn from the entire battery has to stop [35].

$$E = P * t \qquad [Wh] \tag{5.6}$$

Formula 5.6: Calculated electrical energy by using electrical power (P) and time (t)

$$t = \frac{E}{P} \qquad [h] \tag{5.7}$$

Formula 5.7: Solved formula 5.6 for time

Battery voltage	Charge capacity	Total energy	Hour(s) Min	Hour(s) Max
14.8 V	5.2 Ah	77 Wh	0.468 h	3.5 h

Table 5.6: Amount of time the battery should be able to power the components

If the battery is discharged to it is critical level the battery will take damage. The critical voltage of LiPo batteries are approximately 3V per cell [36]. Therefor a voltage monitoring device had to be applied to the batteries to ensure that the voltage never goes below 3V per cell to establish safe discharging of every battery cell.

5.4.3 DC/DC step-down circuit

As discussed in Section 5.4.2 a DC/DC step-down circuit was used in order for the motors and Arduinos to be fed with steady 12V supply. The Raspberry PIs also need power, but they need 5V to enable them to work properly. That is why the project used two different DC/DC step-down circuits.

One of the step-down circuits delivered steady 12V output and the other one delivered steady 5V output (See Section A.8). Steady voltage is preferred in order to manage speed and torque from the motors. If the voltage is constant it is possible to have set speed- and torque variables that controls the motor, if it wouldn't be a constant voltage, then it had to be dynamically regulated variables since the voltage would change over time due to discharging of the battery.

6
Results

In this chapter data on how the robots different sensors and systems performed are presented, as well as what tests were made and how they were performed.

6.1 Ultrasonic sensors

Testing of the ultrasound sensors was first done in an optimal environment, with no interference, and only flat and solid surfaces. The prototype drove towards a wall, 20 times straight on and 20 times in a 45° angle, stopping as soon as it detected the wall. When approaching head on, the prototype stopped at an average distance of 6 cm and a median of 6 cm. When approaching at a 45° angle it stopped at an average distance of 12.8 cm and a median of 13 cm. This difference was due to the fact that when the prototype is driving towards the wall, all of the momentum is directed perpendicular to the wall. On the other hand when approaching at an angle, some of the prototype's velocity is directed parallel to the wall. In other words, the prototype's stopping distance is the same, but the final distance depends on the angle of approach.

The results when tested with a wall are excellent, but in reality it will not consistently be this good. When tested in the field, often when bushes and flowers were in the prototype's path it failed to stop in time, and sometimes it did not stop at all. With larger and more solid objects, like tree trunks and large rocks, the ultrasonic sensors worked as intended and the prototype could stop at a distance of around 6 cm. The same that was observed when tested against a flat wall.

6.2 Camera

The testing of the camera during development was done on a set of 782 images that had been taken in various conditions and scenarios. The images were obtained by manually driving the prototype around the lawn, constantly taking and storing images. The images were then manually labeled with the expected result from the camera algorithm. These labels were compared with the output of the camera algorithm when executed with the same set of images. The images used for testing can be found in the Github repository A.14.

The accuracy was measured in two ways, overall accuracy and rate of false positives. The true accuracy is when the algorithm behaves as desired, it correctly recognizes when it can or cannot continue. This is the most important measure, the better the algorithm is at this, the better it performs in a real situation. But as it did not manage to achieve 100% accuracy, there needed to be an evaluation of when and why the algorithm failed. Either the label said the image was OK and the algorithm disagreed (false negative), or the label said the image was not OK and the algorithm disagreed (false positive). The cases when the algorithm gives a false negative is not as important as when it gives a false positive, because it is better for the algorithm to try to avoid obstacles that are non-existent, than to fail to recognize obstacles that are in the path of the prototype. So the secondary way that the camera algorithm was evaluated was the percentage of failed tests that were false positives.

The final iteration of the algorithm managed to achieve a 95.4% accuracy and a false positive rate of 41.6% on the image set. When testing the camera in the field it performed remarkably well. When approaching a clear division between grass and a stone walk-path head on, the prototype stopped at an average distance of 6.75 cm from the target. Testing darker colored obstacles was done by recording the distance the prototype stopped at when approaching a section filled with large, partly buried rocks. This time the prototype stopped at an average distance of 12.8 cm. This difference is due to the fact that the rocks were surrounded by an area not filled entirely with grass, compared to the first case where the grass was planted right to the edge.

6.3 GPS

To test the GPS and determine whether it could produce good enough results, it was intended to mark out an rectangular area using GPS coordinates, store them on the prototype and then drive it towards an edge of the area. The prototype was instructed to stop completely when it detected it was going out of bounds, so the distance between the prototype and the edge could be measured.

It was discovered that when doing two measurements at the exact same place, the GPS gave back two different results. The GPS had an accuracy of around 10 meters, making it difficult to mark out the area to the degree of accuracy needed. An idea to solve this problem was to manually enter the coordinates of the area using a map, but this was quickly scrapped as the problem with the GPS would still be present. Even if the testing area was manually marked out with a satisfactory accuracy, the GPS would still give too imprecise results when trying to determine where the prototype was located within this area.

The GPS also had trouble setting up and maintaining a fix. It needs a fix in order to give readable results. When the GPS did get a fix it took long time to establish it, sometimes up to 30 minutes, and the fix was frequently lost. So it was very difficult to even begin testing.

In conclusion, testing the GPS was difficult due to factors such as problems getting a fix and difficulties setting up a proper testing area. The results that were obtained when the GPS could be tested showed a lack of accuracy and reliability of the measurements.

6.4 Cost of the prototype

An important part in comparing the success and merit of the project is the cost to construct the robot and how it compares to other robot lawnmowers. There are many variables that could be considered when estimating the cost, such as the cost of labor, the cost of materials, the cost of development and eventual cost of installation. It is difficult to accurately determine what the cost of development and what the cost of installation would be, but cost of materials and parts are easy to summarize as it is known how much each part cost. Presented below is the cost of the parts used on the prototype.

Amount	Part	Cost
2	Raspberry Pi	648 SEK
2	Arduino Nano	138 SEK
2	Back wheels	60 SEK
1	Rotating wheel	60 SEK
2	Motors	738 SEK
1	Control circuit	89 SEK
3	Ultrasound sensors	105 SEK
1	Camera	211 SEK
1	GPS unit	420 SEK
1	Lipo battery	495 SEK
1	12 v DC-DC converter	230 SEK
1	5 v DC-DC converter	230 SEK
18	Sum	3424 SEK

Table 6.1: The parts used for the prototype

6.5 Combined

To evaluate the overall performance of the prototype's construction a some of test were made:

- The velocity of the prototype was measured on a flat lawn and was approximately 0.2 m/s. This velocity is during continuous driving forward and does not include a start and stop distance.
- The maximum inclination for the prototype was evaluated to approximately 12 degrees on a lawn, and a further inclination than this would make the wheels of the prototype lose traction and spin.
- Battery time was tested during a shorter time and then calculated to be approximately 3 hours.

The goal with the project was that the robot could stay inside a dedicated lawn area without the need of a buried bounding cable. With a physically bounded area the robot can stay inside those bounds, but in an unbounded area it can not. This is because of the inaccurate result of the GPS that was supposed to handle the unbounded lawn areas.

7
Discussion

To determine the success of the project and its usefulness, we will in this chapter examine the main parts of the prototype and evaluate them against the purpose and goal of the project. Then we will discuss the future potential of the project and what areas can be improved upon.

7.1 Evaluation of the robot

7.1.1 Ultrasound

The prototype uses the ultrasound sensor to with a relatively small margin detect physical obstacles in its path. On average it will detect object from a distance between 6−13cm (see 6.1). Ultrasound sensors are very good at detecting medium to large solid objects, such as wood or stone. It is also better when the obstacle does not move, because this makes it easier to detect it. If an object just appears briefly in front of the sensors (e.g as an animal running by) there is a chance that the sensors would detect the obstacle too slow or maybe not at all. But this is not a huge problem, because when this situation occurs and the prototype fails to stop the obstacle is not there anymore and it is safe to continue.

The downside with ultrasound is that it is not that good at sensing fabrics and objects that deflects or absorbs sound waves, for example bushes and flowers. Most times this was not an issue because bushes and flowerbeds are not often planted with grass leading all the way up to the stems. Usually there is some dirt or some other material as a boundary which the camera can detect, and help cover for this weakness of the ultrasound. This is a strategy that works well for the robot, the camera and ultrasound complements each other and can succeed when the other fails.

We also noticed that sometimes when driving down a hill, the ultrasound would indicate that there was something in the way when in reality there was nothing there. This happens because the angle of the slope made it so that the ultrasound would detect the ground at the foot of the hill and interpret this as a physical object. It does not happen every time and is not a serious problem, but it is something to consider when deciding if ultrasound should be used. If the lawn is bumpy, it could negatively affect the performance of the prototype as the ultrasound is frequently giving incorrect readings. To sum up, we found that used correctly and in the right environment, ultrasound sensors can be an effective tool in enabling a robot lawn mower to effectively navigate on a lawn and avoid obstacles on it.

7.1.2 Camera

Using a camera on the prototype worked really well, as indicated by a 95.4% success rate in the tests we ran (section 6.2). It did however have trouble with dark and bright areas, such as in shade or sunshine, and when approaching a change in material from an acute angle it is slow to react. Lastly it does not handle surfaces that are green but not grass very well. This has to do with how the prototype determines when a given color is green or not. Because we manually set the thresholds for hue, value and saturation, it is inevitable that some edge cases where missed.

Say that there is a green tarp on top of the lawn where the prototype is driving. It would be unable to tell the difference between the tarp and the grass, so it would continue to drive over the tarp.

7.1.3 GPS

The way we used the GPS-reciever proved to be too inaccurate, at least for the purpose of this project, namely to detect the boundaries of the lawn that might not have a physical presence. Being too inaccurate in this context could lead to permanent damage to the mower or the environment.

It should be mentioned that the Adafruit GPS HAT was implemented without an external antenna or auxiliary system like RTK(Real Time Kinematics). Based on previous research from section 1.7, using either of these might have improved the performance of the GPS HAT but due to time constraints this was not an option.

Our solution is in many cases not ideal. A lawn with many entities obstructing a clear view of the sky will make the GPS-reciever "go blind". This prevents any potential user from having many trees for example. Even moderately tall buildings could have an impact on the signal, which is something that might have played a big role in the difficulty of testing the device. This would rule out utilization of this sub-system in sub-urban areas, something that lawn mowers guided by cable can handle with ease.

Since this sub-system prevents the prototype from mimicking the behaviour of robot lawn mowers on the current market, improvements are needed for it to become viable. This sub-system could potentially be replaced with trilateration (section 2.1) over radio, a solution which was discussed as a potential candidate for the macro-location. It was however scrapped in favor of the relatively ease of use and implementation for the Adafruit GPS HAT.

7.1.4 Driving characteristics

The handling and driving of the robot was not optimal. The motors used was not strong enough to power the robot up hills with efficiency, it had some problems going up steep hills. This might be due to the fact that the simplified mathematical model that was used to find benchmarks of the motor requirements did not give results that were precise enough.

Lack of traction also caused problems for the movement of the prototype. Grass does not give a lot of traction, especially in wet conditions, and the tires that were bought was not the best for this type of outdoor job. In order to have higher traction the tires has to have better treading so they can dig down into the ground and gain traction. The second improvement in order to increase traction is the weight distribution. Since the robot is rear wheel-driven, most of the weight should be placed as far back as possible on the robot in order for the wheels to have the highest amount of traction. Due to poor weight distribution and component placement on the chassis, an extra weight had to be placed on the chassis in the back to have sufficient traction which also increased the total weight of the prototype, making the workload for the motors even higher.

Even though the prototype is not able to drive optimally in all types of terrain, it performed well enough in good conditions. When the lawn was flat and even, with short and dry grass the robot was able to navigate in a satisfactory fashion. In order for the prototype to become a functioning robot lawn mower, it has to have stronger motors and wheels with better traction.

7.1.5 Cost analysis

If we compare the cost of building the prototype listed in section 6.4 with the cost of the cheapest model presented in the study of the current market (1.6), which costs 12.500 SEK , the robot that we constructed is significantly cheaper. So it seems that our prototype is much cheaper than the products on the market. But considering that our prototype is missing some crucial parts, such as a charging station, the ability to cut grass, a proper visual design and a insufficient edge detection. We need to include all these in the pricing as well as other cost like customer support, development and etc.

But in the end, the construction cost of the robot is well within reasonable amounts and the budget of this project. Should the project be replicated or the results used in other projects, extra costs could occur when making a finished product, but ultimately it won't cost considerably more than other similar robots. So it seems that at least in terms of cost, the idea behind the project was reasonable and money would not directly be a limiting factor in the viability of using the methods tested in this project on a commercially viable product.

7.2 Evaluation of the method

The method presented in 1.5 was over all a success in the project. There was however one part in particular that was not fully realized in the construction phase of the GPS. After realizing that the GPS would not be usable there was no new proposal to solve macro-positioning. This can be attributed to lack of time.

Another issue that we faced, that in part could be attributed to the method, is the fact that when entering the construction phase there was a long delay before the ordered parts arrived. This lead to downtime that was not expected. By being able to order parts earlier, or being able to do other things while waiting, this could have been avoided.

The method allows time for colleagues in the project to gather information and learn new techniques. If a similar project was attempted, we would recommend using the same method.

7.3 Can it be built upon/improved

Another sensor area that could possibly yield better results with more development is the GPS and the algorithm to go with it. First of all, an obvious way to improve this system is to increase the accuracy and reliability of the readings. This could be done in different ways, one is to use a RTK station as an other research group did [14]. This is more expensive and maybe too cumbersome to actually use in a finished product, but it could be worth investigating. With better inputs, the way the prototype uses this data could also be improved.

We feel that the ultrasound has performed to a acceptable degree, and could essentially be used as is. What could be investigated is how different configurations and amount of ultrasound sensors affect performance. Additionally, if the prototype were fully surrounded by ultrasound sensors, not just the front, it could be used to better map where an obstacle is located, and how to avoid it.

An area we definitely can see where our design can be greatly and beneficially improved upon is the camera system. It could be worth trying to use machine learning to improve performance, as it is difficult to hard-code what should be considered grass or not. If you train a neural network to recognize this instead, it could produce better results and would avoid trying to manually define what is grass and not, which could differ greatly from lawn to lawn.

Lastly, the movement of the prototype could be improved by another set of wheels. The ones we used did not have satisfactory traction.

7.4 Social and ethical aspects

The most obvious risk with robot lawn mowers is the rotating knives they are equipped with. The blades could cause injuries to a person or an animal, or damage an object on the lawn. To launch the prototype on the market there needs to be systems implemented in the software to prevent accidents from happening. Since we have not installed the blades in our prototype it may also be needed to re-design the product to make it more secure to its surroundings. In future projects when the blades is installed all this needs to be taken into consideration.

It is important that the the prototype can ensure that it will stay inside its dedicated work-space. If the lawn is located close to a road it could cause considerable problems to the traffic if it would wander off, and a neighbor could also find it disturbing if the robot is found on their land.

Except for the safety aspects, a more autonomous lawn mower may result in a wider use on the market and therefore a lower demand for garden services provided by humans, since the mowing part is one of the working tasks a gardener has. The problem with automating tedious task and eliminating jobs is not unique for our product, and is widely discussed all over the world since autonomous products is becoming more and more established on the market. Although the physical work will decrease, the development, maintenance and repairs of these autonomous products will increase and this will lead to new jobs.

Since the robot uses a camera to navigate, this may also create an ethical issues with the camera taking pictures of sensitive scenes when navigating. In our implementation the camera does not save the pictures and it is also directed down at the ground and does not capture any sensitive information. However if all robot lawn-mowers was equipped with a camera this could become a integrity problem like the one seen with drones. This is also a problem for a future development group to analyze and examine how the consumers feels about this.

Animals who get hurt by robot lawn mowers has been a subject on the Swedish news. In an article from Expressen[37] an animal handler describes an increase of animals getting injured by the robots. The victims is mainly pets like dogs and cats. A solution that is presented in the article is to only use the robot lawn mower during the night. Another article [38] described how hedgehogs is specially vulnerable to the robot lawn mowers because of their defence strategies which makes the hedgehog roll into a ball. The robots can not identify the hedgehog and their sharp spikes can not defend the hedgehog against the robot. In this article it is advised to run the robot lawn mower during the day since the hedgehog is active during the night. This shows a conflict, when the robot is best to use. But since our prototype uses ultrasound sensors it can detect hedgehogs and also detect other animals such as dogs and cats. In terms of animal injuries our robot is an improvement and solves a problem for many pet owners.

If we take all of these ethical issues into consideration, our prototype has improved on other robot lawn mowers in some areas with regards to ethical aspects, like avoiding animals, but some other areas like privacy could become a problem if not handled correctly. But we think that the the negative aspects are small enough, and the potential positive aspects outweigh these with a large margin to warrant further development in the subject of smarter robot lawnmowers.

7.5 Should the project be repeated

As stated in section 7.2, there are functionalities that are absent on the final prototype. However, just because this project did not turn out perfect does not mean that it was without merit and not worth expanding upon. Quite the opposite, we believe that there is definite value in repeating and improving upon on what we have done.

If the project were to be repeated we advice against building from scratch as it was the most time consuming part of our project. Lots of things were surprisingly time consuming such as; constructing a chassis, constructing the wheel-axle, constructing attachments for the sensors, and also attach all the components to the prototype. We recommend acquiring a base unit that already has motors and wheels so more time can be dedicated to improving the sensors and the algorithms.

We also recommend dedicating time for planning early in the projects lifespan. By doing the majority of the planning early the risk of having to waste time on redesigning, applying additional components, or implementing new systems. Thus by taking some extra time early on for planning more time will be available later for improving the sensors and the algorithms.

We think a smarter and simpler way to use robot lawn mowers definitely has a place on the market, and further research on robot lawn mowers needs to be carried out.

8 Conclusion

After developing a prototype robot lawn mower and investigating the value and viability of a robot that does not rely on a guiding cable to be installed, we conclude that there is great potential in expanding this area. The camera and ultrasound systems works well for a prototype, and we can see them being used in a fully developed product with great results. Especially the camera is a relatively unused technique among robot lawn mowers, which can improve the future product with the possibility to detect flowerbeds and other object that ultrasound cannot detect. We use a relatively easy algorithm and it still has a very high accuracy, so improvements on the camera algorithm can definitely make the prototype even better. Combined with the ultrasound sensors, the prototype is relatively good at detecting obstacles in its path.

Although a GPS would likely be sufficient as a macro-positioning system, the current implementation requires considerably more development before being used successfully. Alternatively another method could have been used to handle the problem with non visual borders and this is something we recommend another group to investigate.

The project has developed successfully even though all the goals were not met. We have concluded that we should have focused more on the optimization of the sensors and the navigation rather than building the prototype. So our proposal if someone would continue or repeat or work would be to allocate more time to this part. Even though the constructing part was a bit time consuming it was a educative process which gained our members in knowledge.

References

[1] Self-propelled random motion lawnmower. by S. Lawrence Bellinger. US3698523A. Oct. 17, 1972. URL: `https://patents.google.com/patent/US3698523` (visited on 03/16/2018).

[2] Modern Mechanix. *Ad from Mowbot Inc.* [Online]. URL: `http://blog.modernmechanix.com/mowbot/` (visited on 03/21/2018).

[3] Probotics. *Robomower model history: 1999-2015.* [Online]. URL: `https://www.probotics.com/robomower/model-history.htm` (visited on 05/11/2018).

[4] Husqvarna. *Tidsaxel som visar Husqvarnas historia [Time axel that shows the history of Husqvarna].* [Online]. URL: `http://www.husqvarnagroup.com/sv/om/historia/tidslinje` (visited on 03/21/2018).

[5] Bloomberg. *Rise of the Lawn-Cutting Machines.* [Online]. URL: `https://www.bloomberg.com/news/articles/2012-10-25/rise-of-the-lawn-cutting-machines` (visited on 03/21/2018).

[6] HUI Research. *Rekordstarkt köpintresse för robotgräsklippare [Record strong purchase intrest for robot lawn mowers].* [Online]. URL: `http://www.hui.se/nyheter/rekordstarkt-kopintresse-for-robotgrasklippare` (visited on 03/21/2018).

[7] Husqvarna. *Vanliga frågor Husqvarna [Frequently asked questions].* [Online]. URL: `https://www.husqvarna.com/se/grasmatta-tradgard/nar-du-arbetar/underhall/robotgrasklippare-vanliga-fragor/` (visited on 01/31/2018).

[8] Digital Trends. *iRobot's robot lawn mower might be called Terra.* [Online]. URL: `https://www.digitaltrends.com/home/irobot-rlm-the-robotic-lawn-mower-gets-fcc-approval/` (visited on 05/04/2018).

[9] SmartMow. *SmartMow 2 Red.* [Online]. URL: `http://smartmow.com/portfolio-item/smartmow-2-red/` (visited on 05/04/2018).

[10] Husqvarna. *Specifications for Automower 105*. [Online]. URL: https://www.husqvarna.com/se/produkter/robotgrasklippare/%5C%5Cautomower-105/967645421/ (visited on 03/14/2018).

[11] Husqvarna. *Specifications for Automower 420*. [Online]. URL: https://www.husqvarna.com/se/produkter/robotgrasklippare/%5C%5Cautomower-420/967673121/ (visited on 03/14/2018).

[12] Husqvarna. *Specifications for Automower 550*. [Online]. URL: https://www.husqvarna.com/se/produkter/robotgrasklippare/%5C%5Cautomower-550/967650221// (visited on 03/14/2018).

[13] A.Gustafsson et al. *Robotgräsklippare med ruttplanering*. 2017.

[14] M.Baerveldt et al. *Satellitnavigerad robotgräsklippare med RTK*. 2016.

[15] F. Thomas and L. Ros. "Revisiting trilateration for robot localization". In: *IEEE Transactions on Robotics* 21.1 (Feb. 2005), pp. 93–101. ISSN: 1552-3098. DOI: 10.1109/TRO.2004.833793.

[16] M.Kelemen et.al. "Distance Measurement via Using of Ultrasonic Sensor". In: *Journal of Automation and Control* 3.3 (2015). [Online], pp. 71–74. DOI: 10.12691/automation-3-3-6. URL: http://pubs.sciepub.com/automation/3/3/6 (visited on 02/23/2018).

[17] [Online]. URL: https://www.gps.gov/systems/gps/performance/accuracy/ (visited on 04/23/2018).

[18] [Online]. URL: https://www.gps.gov/systems/gps/space/ (visited on 04/23/2018).

[19] Marshall Brain and Tom Harris. *How GPS Receivers Work*. Sept. 25, 2006. URL: https://electronics.howstuffworks.com/gadgets/travel/gps.htm (visited on 04/15/2018).

[20] Ann Neal. *LiDAR vs. RADAR*. [Online]. URL: https://www.sensorsmag.com/components/lidar-vs-radar (visited on 05/14/2018).

[21] *Nearest Neighbor, Bilinear, and Bicubic Interpolation Methods*. [Online]. URL: https://www.mathworks.com/help/vision/ug/interpolation-methods.html (visited on 05/14/2018).

[22] Alvy Smith. *Color Gamut Transform Pairs*. Vol. 12. Aug. 1978, pp. 12–19.

[23] W. Randolph Franklin. *PNPOLY - Point Inclusion in Polygon Test*. [Online]. URL: https://wrf.ecse.rpi.edu//Research/Short_Notes/pnpoly.html (visited on 04/27/2018).

[24] Engineering ToolBox. *Friction and Friction Coefficients*. [Online]. URL: https://www.engineeringtoolbox.com/friction-coefficients-d_778.html (visited on 03/22/2018).

[25] Robotzone. *Specifications DC Planetary Gear Brush Motor*. [Online]. URL: https://www.servocity.com/files/index/download/id/1455645044/ (visited on 02/16/2018).

[26] Clas Ohlson. *Hobbyhjul med gummibana*. [Online]. URL: https://www.clasohlson.com/se/Hobbyhjul-med-gummibana/30-8304 (visited on 05/11/2018).

[27] Clas Ohlson. *Vagnshjul Ø 100 mm*. [Online]. URL: https://www.clasohlson.com/se/Vagnshjul%20%C3%98%20100%20mm/Pr309723000 (visited on 05/11/2018).

[28] Sparkfun. *L298*. [Online]. URL: https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf (visited on 05/08/2018).

[29] Andras Tantos. *H-Bridges- the Basics*. [Online]. URL: http://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/ (visited on 02/22/2018).

[30] The arduino cc team. *File:Pwm5steps.gif*. [Online]. URL: https://commons.wikimedia.org/wiki/File:Pwm_5steps.gif (visited on 04/17/2018).

[31] Dejan Nedelkovski. *Arduino DC Motor Control Tutorial - L298N|PWM|H-Bridge*. [Online]. URL: https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/ (visited on 02/22/2018).

[32] Cyril BUTTAY. *File:H bridge.svg*. [Online]. URL: https://commons.wikimedia.org/wiki/File:H_bridge.svg (visited on 04/17/2018).

[33] Arduino. *Arduino Nano productpage*. [Online]. URL: https://store.arduino.cc/usa/arduino-nano (visited on 03/27/2018).

[34] Raspberry Pi. *WHAT ARE THE POWER REQUIREMENTS?* [Online]. URL: https://www.raspberrypi.org/help/faqs/#powerReqs (visited on 04/13/2018).

[35] F. Baronti et al. "Hierarchical platform for monitoring, managing and charge balancing of LiPo batteries". In: *2011 IEEE Vehicle Power and Propulsion Conference*. Sept. 2011, pp. 1–6. DOI: 10.1109/VPPC.2011.6043074.

[36] Brian Schneider. *A Guide to Understanding LiPo Batteries*. [Online]. URL: https://rogershobbycenter.com/lipoguide/ (visited on 03/29/2018).

[37] Gustaf Nilsson. *Robotgräsklipparen – en fälla för ditt husdjur [Robot lawn mower - a trap for your pet]*. [Online]. URL: https://www.expressen.se/gt/robotgrasklipparen--en-falla-for-ditt-husdjur/ (visited on 04/20/2018).

[38] Johan Dernelius. *Robotgräsklipparen – igelkottens värsta fiende [Robot lawn mower - the hedgehog's worst enemy]*. [Online]. URL: https://www.svt.se/nyheter/lokalt/skane/robotgrasklipparen-igelkottens-varsta-fiende (visited on 04/20/2018).

[39] Arduino. *Arduino Nano*. [Online]. URL: https://www.arduino.cc/en/Guide/ArduinoNano (visited on 03/27/2018).

[40] Raspberry Pi. *RASPBERRY PI 3 IS OUT NOW! SPECS, BENCHMARKS & MORE*. [Online]. URL: https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/ (visited on 03/29/2018).

[41] ElecFreaks. *Ultrasonic Ranging Module HC - SR04*. [PDF]. ElecFreaks, 2011. URL: http://www.micropik.com/PDF/HCSR04.pdf (visited on 03/23/2018).

[42] Raspberry Pi. *Camera specifications*. [Online]. URL: https://www.raspberrypi.org/documentation/hardware///camera/README.md (visited on 03/12/2018).

[43] [Online]. URL: https://learn.adafruit.com/adafruit-ultimate-gps-hat-for-raspberry-pi/overview (visited on 04/24/2018).

[44] Conrad. *RC Batterpack (LiPo) 14.8 V 5200 mAh Antal celler: 4 40C Gens ace Stick XT90*. [Online]. URL: https://www.conrad.se/RC-Batteripack-%5C%28LiPo%5C%29-14.8-V-5200-mAh-Antal-celler:-4-40-C-Gens-ace-Stick-XT90.htm?websale8=conrad-swe&pi=1594978&ci=SHOP_AREA_211999_1209075 (visited on 04/10/2018).

[45] MEAN WELL. *25W Single Output DC-DC converter*. [PDF]. MEAN WELL, 2012. URL: https://www.elfa.se/Web/Downloads/_t/ds/sd25_eng_tds.pdf (visited on 04/10/2018).

A

Miscellaneous

A.1 Arduino Nano

The developed robot lawn mower uses one Arduino Nano to help control the electric motors and another one for controlling the ultrasonic sensors. Arduino Nano is one of Arduino's smallest models with few ports but in a compact format, it features almost the same specs and connections as Arduino Uno [39]. The board is as all of Arduinos products open-source based. The Nano is based on the ATmega328P single-chip microcontroller, the same as the Arduino Uno uses. It also uses Arduinos own development environment, Arduino Software(IDE) which is a modified variant of C and C++. It offers a lot of libraries for a wide range of different applications.

Microcontroller	ATmega328
Architecture	AVR
Operating Voltage	5 V
Flash Memory	32 KB of which 2 KB used by bootloader
SRAM	2 KB
Clock Speed	16 MHz
Analog IN Pins	8
EEPROM	1 KB
DC Current per I/O Pins	40 mA (I/O Pins)
Input Voltage	7-12 V
Digital I/O Pins	22 (6 of which are PWM)
PWM Output	6
Power Consumption	19 mA
PCB Size	18 × 45 mm
Weight	7 g

[33]

Table A.1: Arduino Nano specifications

A.2 Raspberry PI

The developed robot lawn mower uses two Raspberry PIs (RPi), one for sensor processing and one for decision making. Both RPis are model 3B, which were released 2 years ago. Its the first RPi model that features a built in wireless networking chip.

As with all RPi models, 3B can run a full ARM operating system, making it an excellent all purpose-chip.

System on Chip	Broadcom BCM2837
CPU	4× ARM Cortex-A53, 1.2GHz
GPU	Broadcom VideoCore IV
RAM	1GB LPDDR2 (900 MHz)
Networking	10/100 Ethernet, 2.4GHz 802.11n wireless
Storage	microSD
GPIO	40-pin header, populated
Ports	HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI) Display Serial Interface (DSI)

[40]

Table A.2: Raspberry Pi 3B specifications

A.3 Ultrasonic Ranging Module HC-SR04

The Ultrasonic Ranging Module utilizes the physics of mechanical waves, in this case ultrasound to determine the distance from the emitter to another surface. The module includes one ultrasonic transmitters, one receiver and a control circuit. Some of the technical data of the module can bee seen in table A.3.

Working Voltage	5V
Working Current	15mA
Working Frequency	40kHz
Max Range	4m
Min Range	2cm
Measuring Angle	15 degrees
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimensions	45 × 20 × 15 mm

[41]

Table A.3: Technical values of the URM

The module is relatively simple, it requires four pins to operate. Apart from the voltage in and ground pins (Vcc and Gnd respectively) it has a trigger pin and a echo pin. When the trigger pin is set to high, the module sends out eight ultrasonic sound bursts at 40kHz. When the sound bounces back to the module the Echo pin becomes set to high. By measuring the time between these events it's possible to calculate a distance using the speed of sound.

A.4 Raspberry Pi Camera Module Rev 1.3

The camera used in this project is the Raspberry Pi camera module Rev 1.3. It is a camera specifically developed for the Raspberry PI, and it comes with a great deal of software support.

Size	25 × 24 × 9 mm	
Still resolution	5 Megapixels	
Video modes	1080p30, 720p60 and 640 × 480p60/90	
Sensor	OmniVision OV5647	[42]
Sensitivity	680 mV/lux-sec	
Horizontal field of view	53.50 +/- 0.13 degrees	
Vertical field of view	41.41 +/- 0.11 degrees	

Table A.4: Technical Values of the RPI camera v 1.3

A.5 Adafruit Ultimate GPS HAT

Sensitivity	- 165 dBm	
Update speed	10 hz	
Channels	66	[43]
Current	20mA	
Altitude limit	32 km	

Table A.5: Technical Values of the Adafruit Ultimate GPS HAT

A.6 DC Motor

Operation voltage	6-12 V	
Rated voltage	12 V	
Max stall current	4.9 A	
Rated-load current	0.56 A	
Rated load	6 kgf-cm	
No-load speed	32 RPM	[25]
Rated-load speed	28 RPM	
Gear type	Planetary	
Gear ratio	1/370	
Weight	100 g	

Table A.6: DC motor specifications

A.7 Battery specifications

Battery type	LiPo	
Number of cells	4	
Connector type	XT90	
C rating	40 C	[44]
Nominal voltage	14.8 V	
Battery capacity	5200 mAh	
Weight	443 g	

Table A.7: Battery specification of LiPo battery pack

A.8 DC/DC converters

Input DC voltage	9.2-18 V	
Output DC voltage	11-16 V	
Rated current	2.1 A	[45]
Rated power	25.2 W	
Weight	380 g	

Table A.8: 12V DC/DC step-down converter SD-25A-12

Input DC voltage	9.2-18 V	
Output DC voltage	4.5-5.5 V	
Rated current	5 A	[45]
Rated power	25 W	
Weight	380 g	

Table A.9: 5V DC/DC step-down converter SD-25A-5

A.9 Camera mount schematic

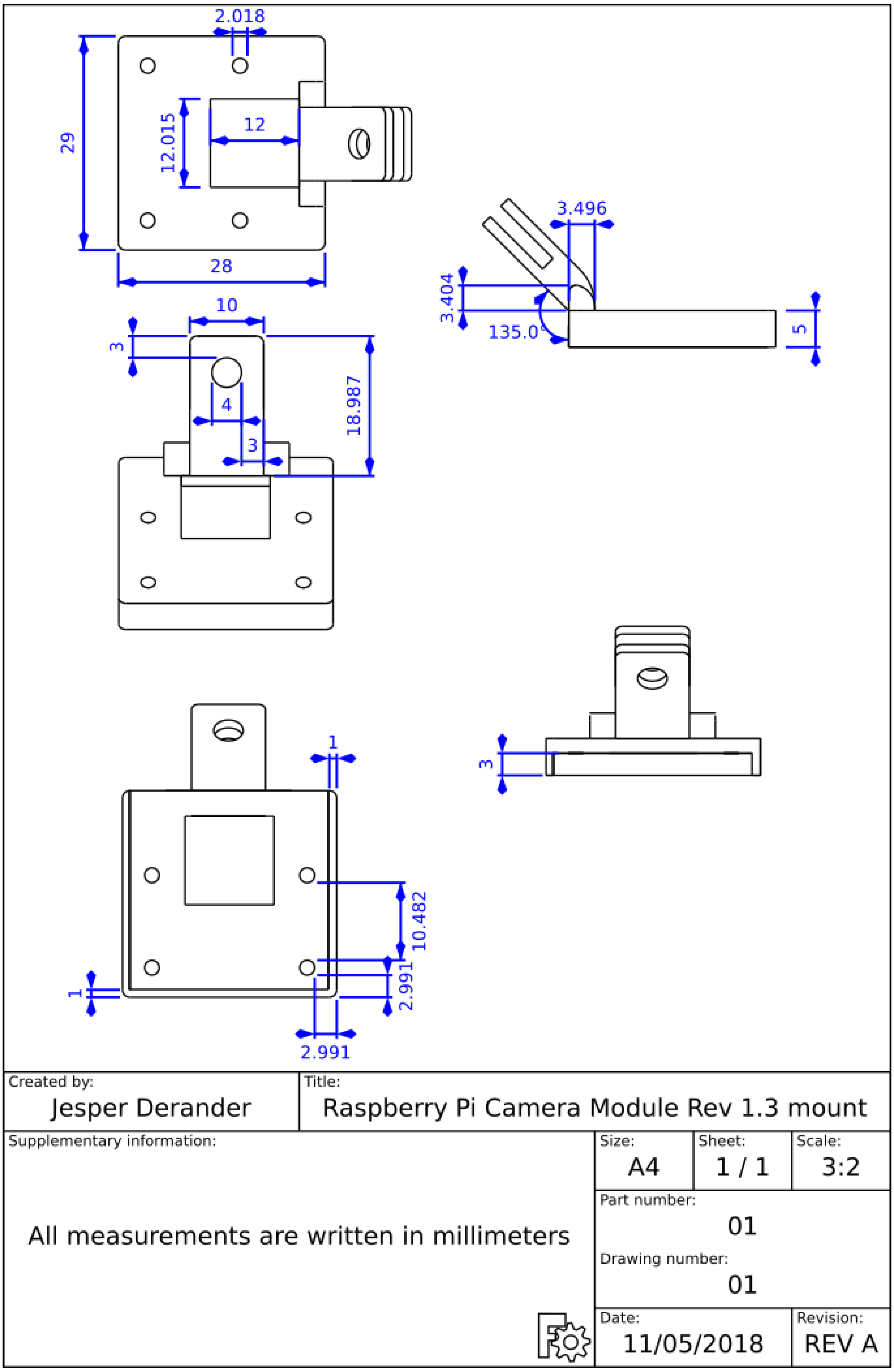


Figure A.1: A schematic of the raspberry pi camera module rev 1.3 mount

A.10 Ultrasound bracket schematic

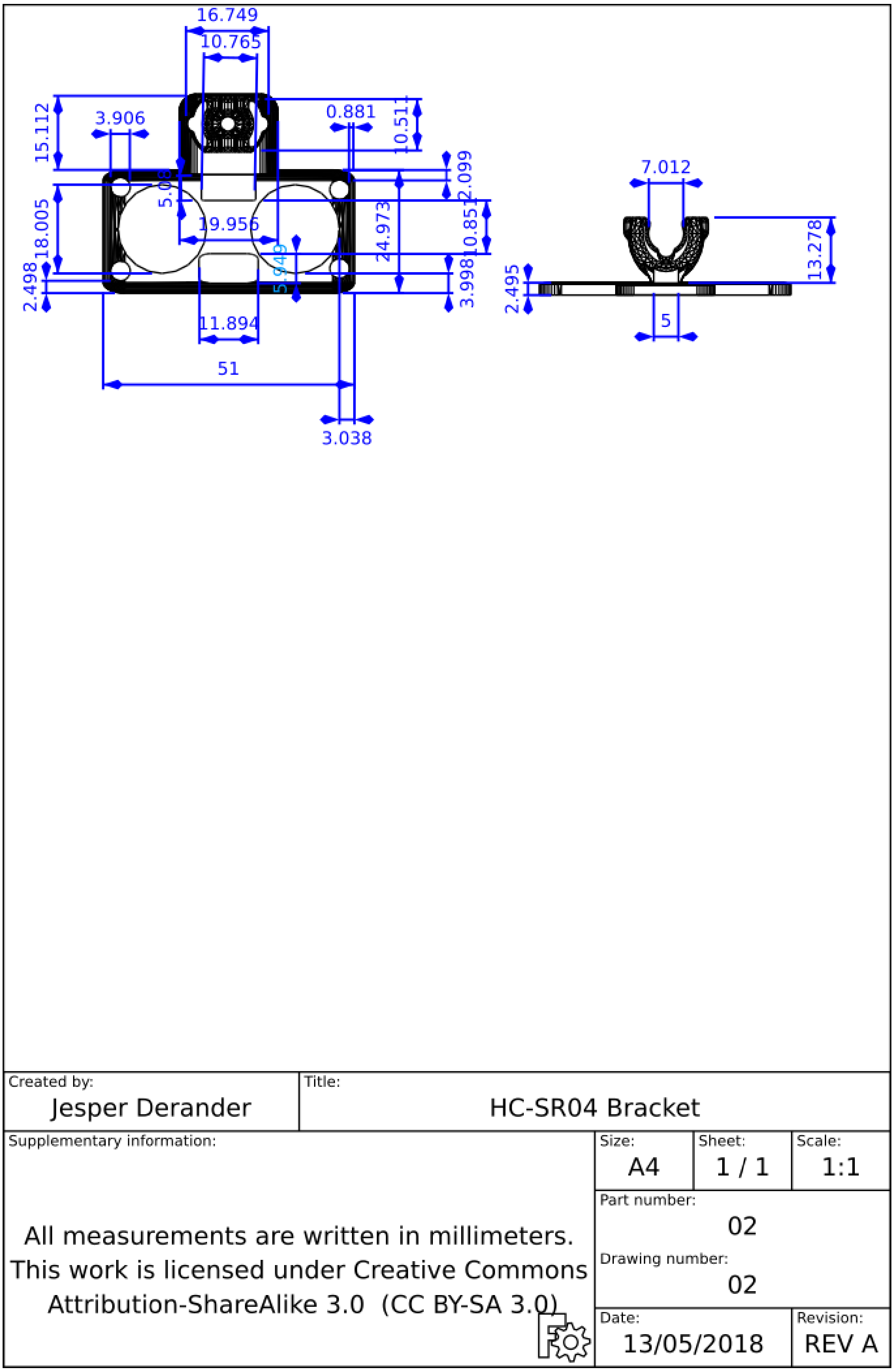


Figure A.2: A schematic of the brackets used for the front mounted ultrasound sensors

A.11 Ultrasound bracket stand schematic

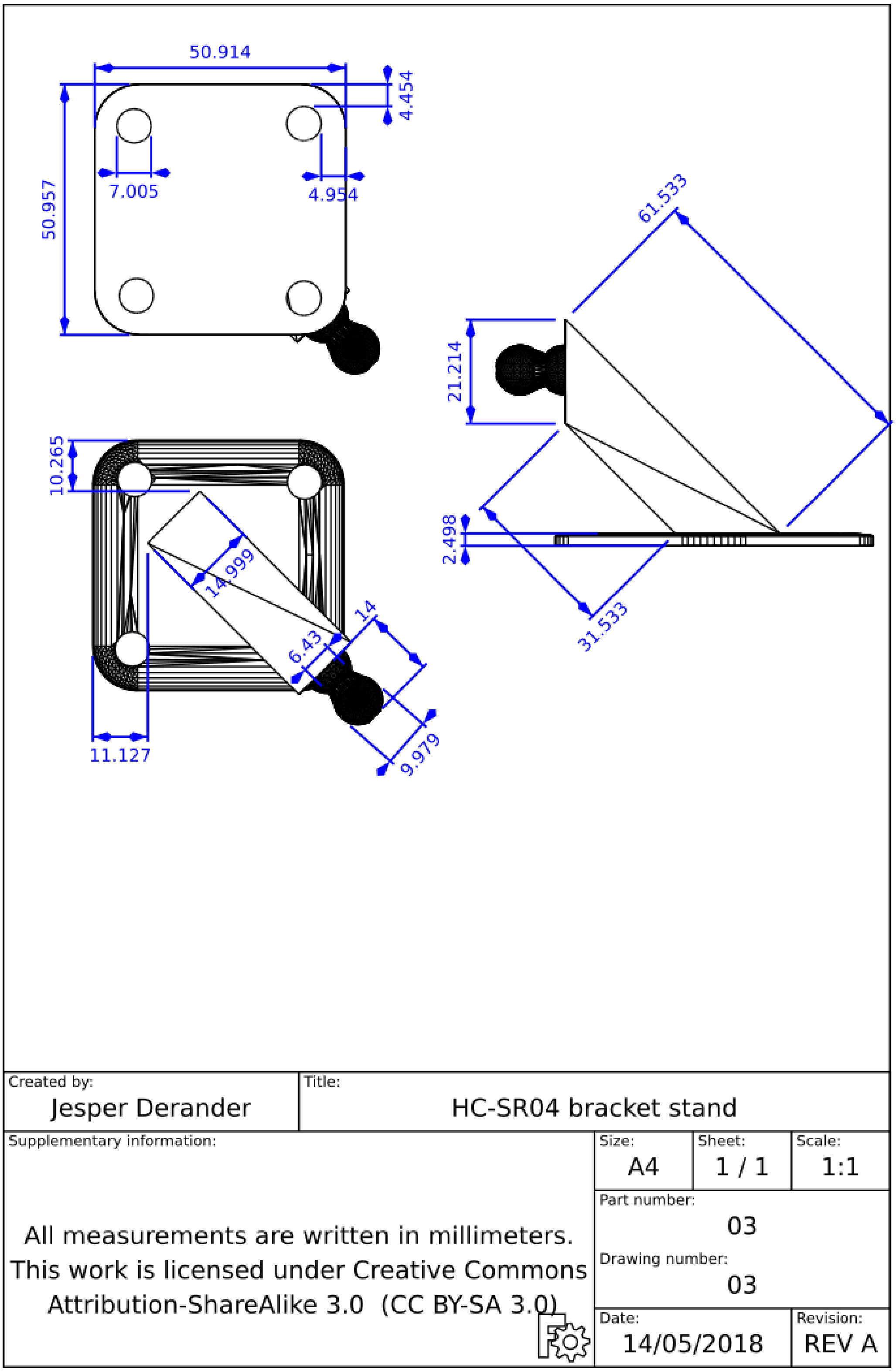


Figure A.3: A schematic of the stands that snap into the brackets for the HC-SR04 sensors

A.12 Chassis schematic

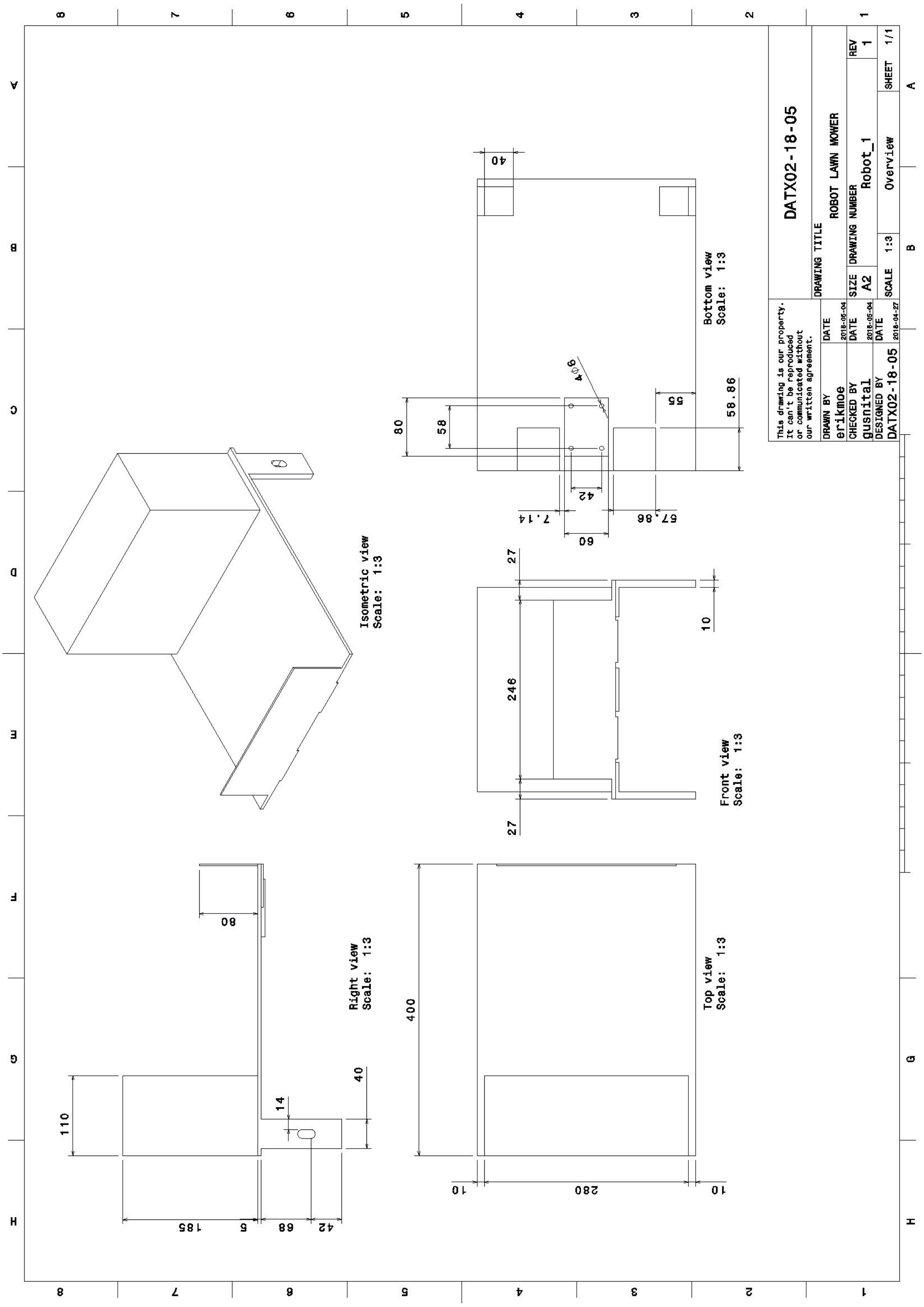


Figure A.4: A schematic overview of the prototype

A.13 Circuit layout

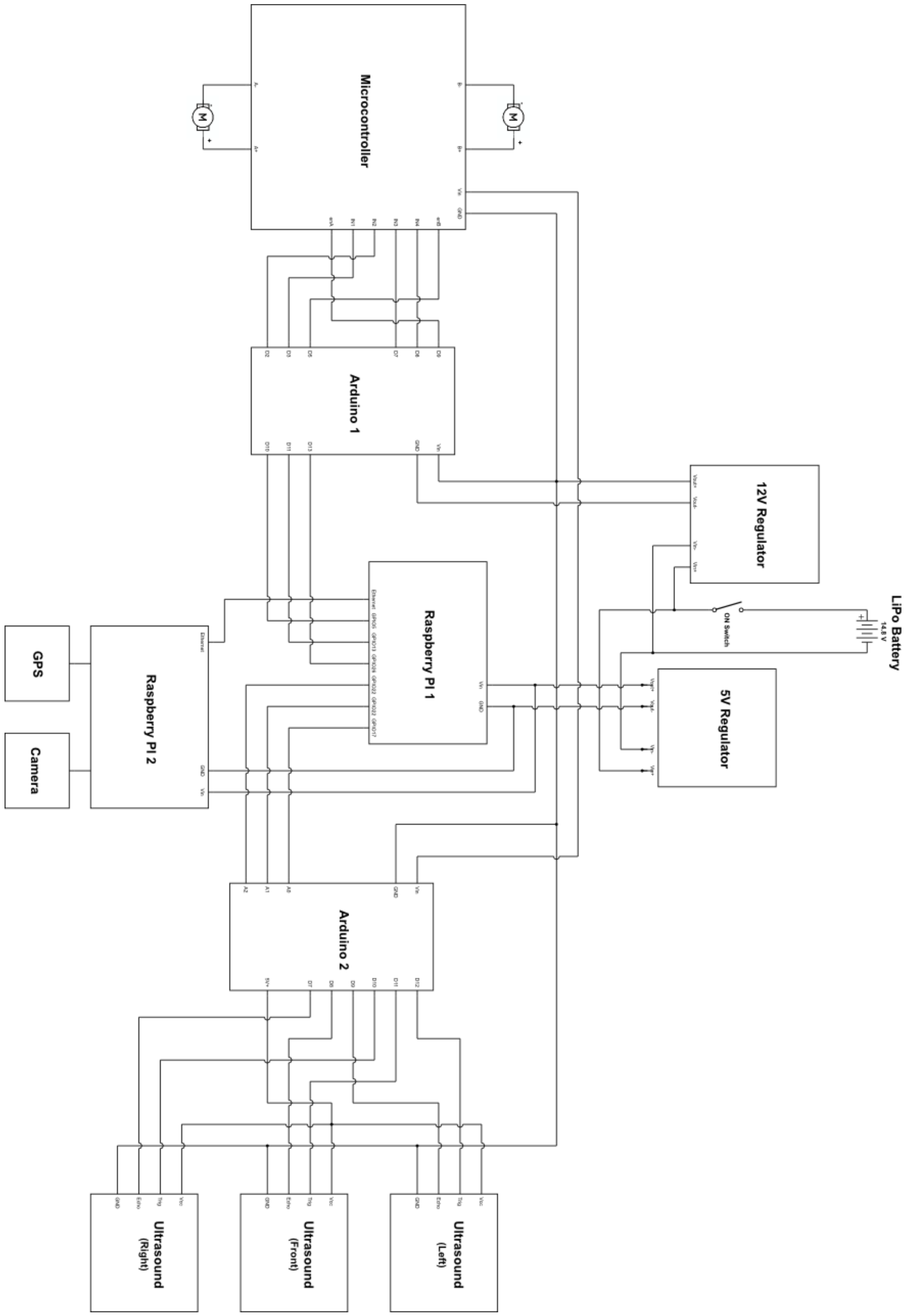


Figure A.5: Electrical schematic of the robot lawn mower

A.14 GitHub repository

The code that were used in this project can be found in this GitHub repository: https://github.com/ericwenn/robot-lawn-mower