DEGREE PROJECT IN MECHANICAL ENGINEERING,
FIRST CYCLE, 15 CREDITS
STOCKHOLM, SWEDEN 2021

PlaCo

The plastic collecting robot

ANNIE PERSSON

JOHANNA BERGSTEN

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT

PlaCo

The plastic collecting robot

ANNIE PERSSON
JOHANNA BERGSTEN

Bachelor's Thesis at ITM
Supervisor: Nihad Subasic
Examiner: Nihad Subasic

TRITA-ITM-EX 2021-46

Abstract

The oceans are an essential global resource for all living organisms but especially for us humans. However, year after year we continue to neglect proper recycling of our waste, resulting in litter ending up in our oceans. The majority of said litter comes from single use plastic items. Through fragmentation and erosion, the plastic dissolves to smaller pieces, once they are no larger than 5 mm they are classified as micro- and nanoplastics. Little is known about these small plastic particles impact on marine life and marine environment.

As a step towards understanding this, the robot PlaCo was created. PlaCo stands for plastic collecting which is exactly what the robot does. With the help of three filters PlaCo gathers marine debris, such as plastic, from the water in which it operates. The filters have decreasing mesh size resulting in the microplastics being caught in the last one. Once emptied, the finds can be examined and logged for future referencing. With the help of a sensor, blockages of the filters can be monitored and if detected, notice will be given to the user through a LED. In order for PlaCo to move forward and for water to travel through the filters, the robot was provided with two DC motors. A microcontroller, Arduino Uno, was used to regulate PlaCo's functions.

The performance of two different sensors, an IR-sensor and an ultrasonic sensor, were investigated as well as the robots water cleaning capacity. The results indicate that the latter of the two sensors would be preferable due to its high reliability. However, the robot's water cleaning capacity could not be measured due to the chosen motors not being powerful enough. In future iterations of PlaCo, this would need to be rectified.

Keywords: Arduino, IR-sensor, marine debris, mechatronics, microplastics, ultrasonic sensor

Referat

PlaCo - Roboten som samlar upp plast i vattnet

Globalt sett är världens hav en viktig resurs för alla levande organismer men inte minst för människan. Trots detta fortsätter vi att, år efter år, försumma återvinningen av vårt avfall vilket resulterar i att skräp i stället hamnar i haven. Majoriteten av de sopor som hamnar där är resultatet av förbrukade engångsprodukter i plast. Genom sönderfall och erosion skapas allt mindre och mindre bitar av plast. Detta resulterar i att så kallade mikro- och nanoplaster skapas. De är mindre än 5 mm i bredd och om deras påverkan på det marina djurlivet och den marina miljön vet vi mycket lite om. För att minska kunskapsluckorna och för att få en bättre förståelse för deras påverkan har nu därför PlaCo konstruerats.

Med hjälp av tre sorters filter kan PlaCo samla upp marint skräp, så som plast, i vattnet där den arbetar. Filtrens finhet varierar, där det första är mycket grovt medan det sista är fint nog att klara av att samla upp mikroplaster. När PlaCo sedan töms kan mikroplasterna undersökas och dokumenteras. När en tömning behöver göras indikeras detta för användaren med hjälp utav en LED. Roboten är försedd med en sensor som läser av hur fulla filtren är. För att driva PlaCo framåt och för att underlätta filtreringsprocessen är den också försedd med två 6 V DC-motorer. Allt detta styrs med hjälp av mikrokontrollern Arduino Uno.

För att uppnå bästa tänkbara funktion hos roboten undersöktes två olika sensortyper, en IR-sensor och en ultraljudssensor. Det visade sig att ultraljudssensorn var betydligt mer pålitlig än IR-sensorn och därför valde man att använda denna. Det var även av intresse att ta reda på hur mycket vatten PlaCo kunde rena per sekund. Tyvärr skulle det visa sig att de valda motorerna inte var kraftfulla nog att driva PlaCo i vattnet. Det är därför något som behöver åtgärdas i en framtida version av PlaCo.

Nyckelord: Arduino, IR-sensor, marin nedskäpning, mekatronik, mikroplaster, ultraljudssensor

Acknowledgements

We would like to acknowledge and give thanks to the following people who have contributed to making this an enjoyable and interesting project:

Nihad Subasic, our supervisor, for his interesting and inspiring lectures.

Amir Avdic, the assistant in the lab, for being helpful and patient with our circuit.

Staffan Qvarström, for always being helpful with the choice of components, and giving us essential knowledge and material.

Fellow MF133X students, for their interesting and helpful feedback, and also for their inspiring work.

Annie Persson & Johanna Bergsten
Stockholm, May 2021

Contents

List of Figures

List of Tables

List of Abbreviations

3D 3-dimensional.

DC Direct Current.

HDPE High Density Polyethylene.

IDE Integrated Design Environment.
IR Infrared.

LDPE Low Density Polyethylene.
LED Light-Emitting Diode.

mA Milliampere.

NMPs Nano-Microplastics.

PLA Polylactic Acid.
PP Polypropylene.
PS Polystyrene.
PWM Pulse-Width Modulation.

RPM Revolutions Per Minute.

USB Universal Serial Bus.

V Volt.
VS Voltage Supply.

Chapter 1

Introduction

Oceans cover about three quarters of the Earth's surface and represent 99 percent of the living space by volume. This, along with the fact that over three billion people depend on marine and coastal biodiversity as their source of income, makes the oceans an essential global resource [1]. However, every year, 5 to 13 million tons of plastic end up in the water [2]. This causes animals to involuntarily ingest and entangle themselves in the plastic, which leads to critical injuries and deaths. More than one million seabirds, as well as 100 000 marine mammals, die as a result of plastic pollution every year. The source of the ocean litter can be traced back to the extensive usage of single-use plastic items. It is estimated that five trillion disposable plastic bags are used annually, and every minute, one million plastic bottles are purchased [1, 3].

As a step to attain a better and more sustainable future, the countries of the United Nations accepted *Agenda 2030* in the year 2015. The Agenda 2030, also known as the Sustainable Development Goals, contain 17 goals that address economic, social and environmental problems [4]. Goal number 14—*Life below water*, deals with conservation and sustainable usage of oceans, seas and marine resources. One of the targets is to, by the year 2025, prevent and significantly reduce marine pollution of all kinds, in particular, from land-based activities. This includes marine debris [1].

1.1 Background

The objective of this project is to be a part of this change. The project is set to focus on filtering water in order to remove marine debris, such as plastic. With the help of a floating vessel, three filters, with decreasing mesh size, will be collecting foreign materials to ensure that no further harm is done to marine ecosystems. The robot will also pose as a potential research instrument as one of the filters will be fine enough to collect microplastics.

Microplastics are plastic debris which measure between 0,1 mm to 5 mm in size. If the plastic particles have a size less than 0,1 mm, they are called nanoplastics. *Nano-Microplastics (NMPs)* are the result of fragmented and eroded plastic items which have been discarded or in other ways ended up in the water. NMPs are present not only in oceans and freshwater, but also in air, soil and biota. Subsequently, they are a part of the human diet [5]. However, there is very little research published on micro- and nanoplastics ubiquity as well as their impact on marine environments [6, 5]. At present, there is no sufficient data to indicate a link between NMPs and harm to marine environments and marine life. However, there is also no evidence to indicate that a constant or an increased level of NMPs in nature would not be problematic [5]. As a way to study and potentially understand the link between NMPs concentration and marine welfare, the robot of this project can be used to collect microplastics. Their presence can then later be logged and the wellbeing of aquatic organisms can be observed.

1.2 Purpose

As stated previously, this project is set to focus on cleaning water from alien materials such as plastic. The robot will, with the help of sensors, be able to tell when the filters are full. As the effectiveness of the robot is important to its function, two different types of sensors will be tested, an active Infrared (IR)-sensor and an ultrasonic sensor. It will also be of interest what quantity of water the robot will be able to filtrate.

More specifically, the purpose of this project is to investigate the following questions:

- How many liters of water can be cleaned per second?
- Can the device, in a sufficient way, tell when the filter is full?
- Which of the two sensors, the active IR-sensor or the ultrasonic sensor, is preferable for detecting a full filter?

1.3 Scope

Other than time and budget being demarcations for this project, the restrictions and limitations due to the Covid-19 pandemic must be taken into consideration.

The project will also not take the following factors into consideration:

- The effectiveness of steering and route planing as the collecting of marine debris is the main focus.
- The fluid dynamics of the robot's body due to the lack of time and knowledge.
- The robot is just a prototype which means it will not be tested in open water. It will be tested in calm waters with a limited environment.

1.4 Method

The first task is to construct the filtration mechanism and then program the sensors. The sensors should be able to sense when the filters are full and need to be emptied. Secondly, the propellers and motors can be assembled. The base of the vessel and the floating contraption will then be built around this construction as it needs to carry its weight. When this is done, a steering system will be set up. Testing of the sensors and eventually the whole robot will be carried out as a last step.

Chapter 2

Theory

The following chapter gives necessary theory concerning the components included in robots construction.

The design of the robot has taken inspiration from the Manta Trawl. The Manta Trawl is a fine-mesh net system commonly used to collect floating plastic and debris from the water's surface. Its main parts are a metal framework, usually in aluminium, with wings on either side and a long fine-mesh net tail behind it. The litter will be filtered through a mesh and can later be examined and logged. The Manta Trawl can either be left behind a tugging boat or be stationed in a river or a stream [7].

2.1 Control system

As the robot will not be towed behind a boat it needs its own control system. Said system will consist of two motors, each equipped with a propeller and a pre-programmed route.

2.1.1 Microcontroller

An Arduino UNO R3 was used to control the robot and is presented in figure 2.1 below. The Arduino UNO is an electric open-source microcontroller that is designed with both digital and analog in- and output. The microcontroller works like a computer that you can connect different electronic components to and gives the possibility to control or regulate. To upload programs to the microcontroller, the Arduino uses serial communications interfaces, like Universal Serial Bus (USB). The codes uploaded to the microcontroller are programmed using C and C++ program language in the Arduino software Integrated Design Environment (IDE)[8].

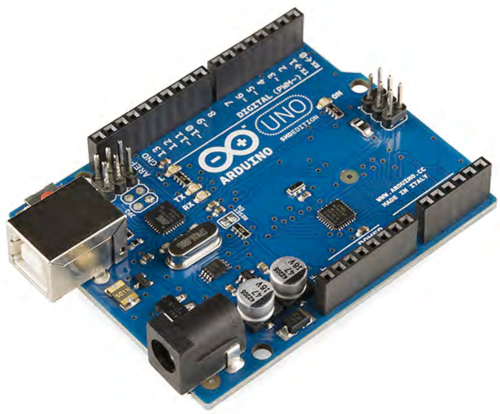


Figure 2.1. Arduino UNO, [9]

2.1.2 Steering mechanism

As stated in the scope, the core of this project is not the fluid dynamics nor the seaworthiness of the robot. Therefore, the control system will only provide the bare necessities to test the vessel's filters and sensors. An Arduino will be implemented, as part of the control system, to regulate the power output of the motors. A route will be programmed prior to the start of the operation and can be changed to fit the conditions and desired length of the route. This also eliminates the need for a "home procedure" and a base for the robot to return to when the trawling is done.

2.1.3 Motors

To be able to filtrate water, the robot needs to be moving forward. To do this, two Direct Current (DC)-motors were used. A DC-motor is an electrical motor that converts electrical energy (direct current) to mechanical energy. With a direct current flowing in a wire winding, combined with a magnetic field, a torque will be generated. This torque will be applied on the motor shaft which is converted to mechanical energy[10].

The DC-motor consists of two main components, a rotor and a stator. The rotor, or the armature, is the moving part in the motor. The armature is the part that contains rotor windings that the direct current flows through. The winding is wires that are laid in coils. When the current flows through the winding, it will create an electromagnet. The stator on the other hand is a stationary part that surrounds the rotor. It provides the magnetic circuit with a magnetic field. When the current flows through the winding in the armature, the magnetic field caused from the stator will produce two parallel, but opposite, forces on the rotor. This will make the rotor rotate and in turn generate torque. This is illustrated in a simplified illustration in figure 2.2 below[10].

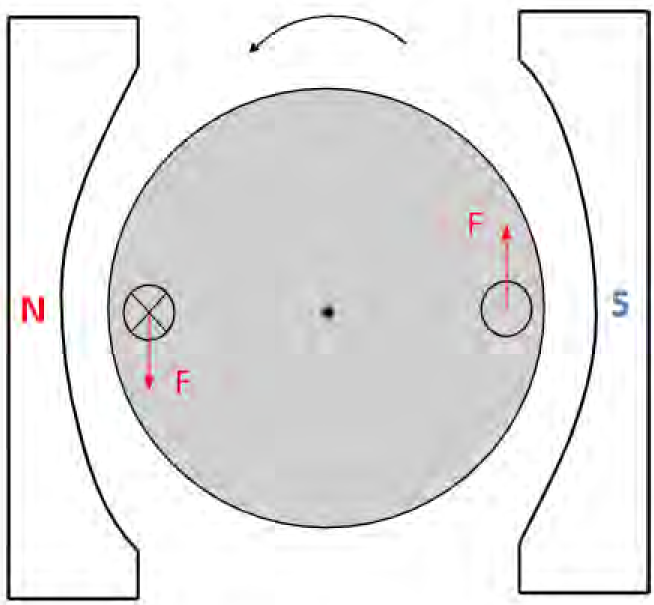


Figure 2.2. Cross section of a DC-motor (made in Power Point)

To keep the armature rotating, a commutator is attached to it. The current will then go through two brushes that are connected to the commutator. The commutator periodically reverses the current between the magnetic circuit and the armature. This makes the electromagnet switch its poles which keeps the rotor rotating. The speed of the motors can be controlled with the direct current flowing through the system[10].

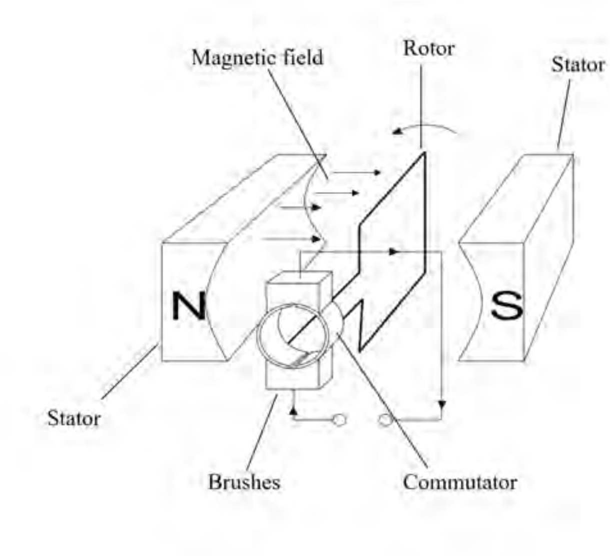


Figure 2.3. Sketch of a DC-motor, [10]

2.1.4 Power supply

In an ideal world the robot would be powered with solar panels but as these might drive up the price of the project and as the sun is a bit unreliable during the winter months. Therefore, normal AA batteries show below in figure 2.4, preferably rechargeable ones, will be used depending on the power demands of the other components.

Figure 2.4. AA-batteries, [11]

2.2 Sensors

The sensors that will be taken into consideration in this project are an IR-sensor and an ultrasonic sensor. The idea is that the sensors will detect when the filters are full or need to be changed.

2.2.1 IR-sensor

The IR- sensor is a device that detects and measures infrared radiation [12]. Infrared radiation, or infrared light as it is sometimes called, is a type of electromagnetic radiation. The wavelength of the infrared light is longer than visible light which makes infrared light invisible for the human eye. Instead, we can feel some of the radiation as heat, for example a fire [13].
The active IR-sensor both emits and detects infrared radiation, and it consists of a Light-Emitting Diode (LED) and a receiver. The LED emits an infrared light that reflects off the object which then is picked up by the receiver, which is illustrated below in figure 2.5 .

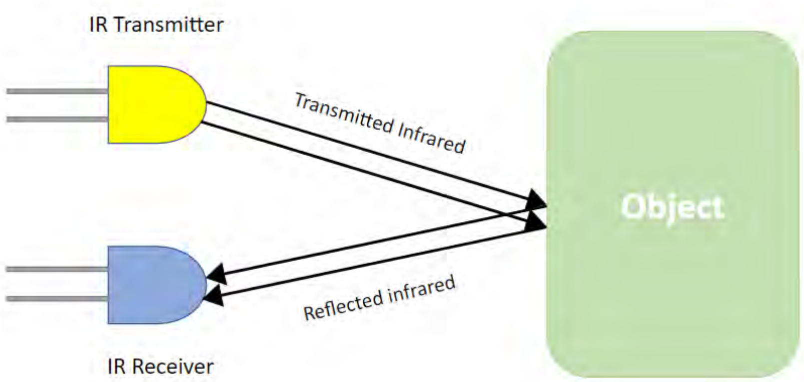


Figure 2.5. Principle of an active IR sensor (made in Power Point)

2.2.2 Ultrasonic sensor

An ultrasonic sensor is a device that can measure distance with ultrasonic waves. The main components that the sensor consists of are: a receiver, a transmitter, and a transducer. To measure the distance to an object, the transmitter emits soundwaves that will reflect on an occurring object. The emitted soundwaves are at a frequency which is higher than human hearing. The soundwaves will then reflect on the object's surface which the receiver collects, which can be seen in figure 2.1 below. The collected soundwaves are then converted to electrical signals by the transducer. Furthermore, the transducer also converts the other way around when the transmitter emits soundwaves. To then measure the distance, the time between the emission and collection of the soundwaves is measured. The distance is then calculated with the following equation[14, 15].

$$D = \frac{1}{2}Tc, \tag{2.1}$$

where D is the distance, T the time it takes between emission and collection, and c is the speed of sound.

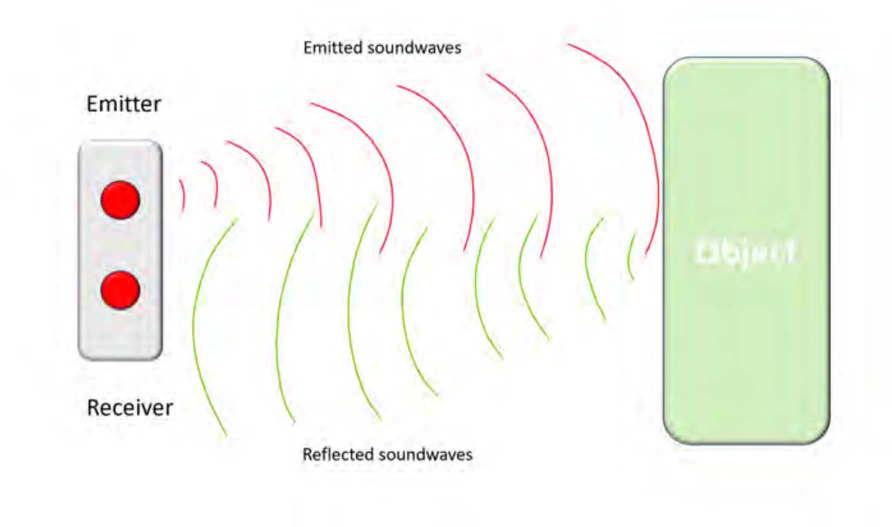


Figure 2.6. Principle of an ultrasonic sensor (made in Power Point)

2.3 Floating device

In terms of a floating device, one must consider the buoyancy. The buoyancy can be described with Archimedes principle.

"Archimedes's principle: When a body is completely or partially immersed in a fluid, the fluid exerts an upward force on the body equal to the weight of the fluid displaced by the body"

By placing a solid object in water, Archimedes's principle can be proven. First, consider the fluid to be in equilibrium hence the force on the surface of the fluid is zero. Then on the object's surface there is a pressure acting. The pressure on the object's top will be smaller than the pressure on the bottom of the object. This pressure difference is directly proportional to the depth below the surface of the fluid. The upward force acting on the object is equal to the weight of the fluid displaced by the object, this is illustrated below in figure 2.7. This is called the buoyancy force, which can be described with the following equation.

$$F_b = -\rho g V, \tag{2.2}$$

where F_b is the buoyancy force, ρ is the fluid density, g is the gravity acceleration and V is the volume of the displaced fluid[16].

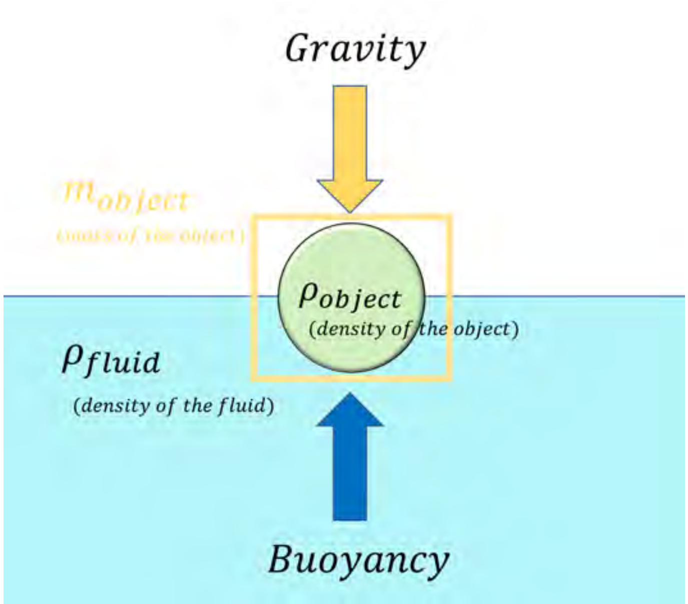


Figure 2.7. Buoyancy (made in Power Point)

Chapter 3

Prototype

This chapter will describe which components were used in the project. It will also motivate the choices of components.

3.1 Setup

This section will describe how the components are connected and how PlaCo has been constructed. In figure 3.1 the main components of the robot can be seen. To get a feel for the robot's movements and size, a simulation in Acumen was made, see Appendix A for full code.

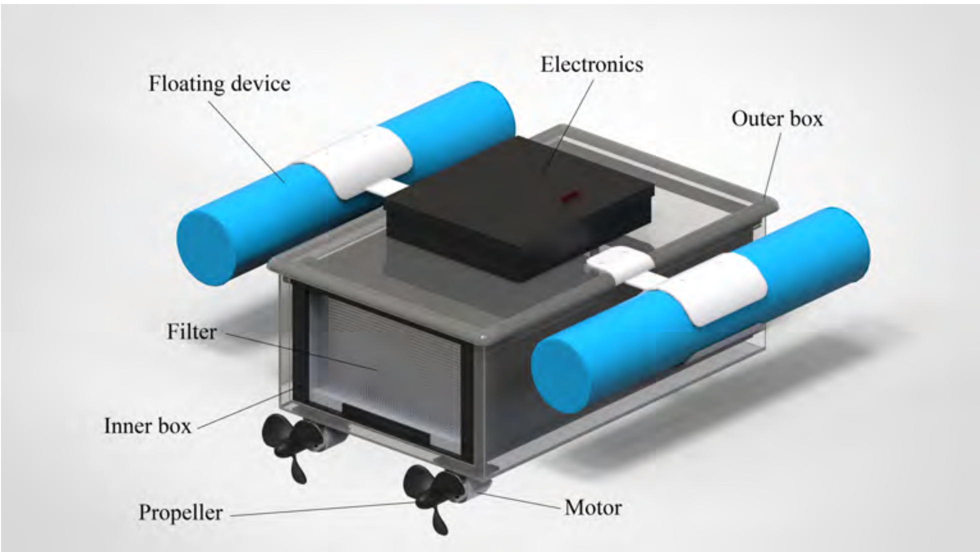


Figure 3.1. PlaCo as a whole (made in Solid Edge, rendered in KeyShot)

3.1.1 Body - inner and outer box

The outer body of the robot was made of a Polypropylene (PP) box with the following measurements.

Table 3.1. Outer box measurements

Length	39 cm
Width	26 cm
Depth	13 cm

The short sides of the box were cut open to be able to get a flow of water flowing through the robot. To easily change the filters in the robot, a smaller box was placed inside the bigger box. The smaller box was designed with the program Solid Edge and then 3-dimensional (3D)-printed with Polylactic Acid (PLA)-plastic. To keep the filters in place, the inner box has three grooves that will act like a frame for the filters.

3.1.2 Propellers

To move the robot forward, propellers were attached to the motors. Since the propellers were not the main focus of the project, they were inspired by already existing models. The propeller model was first designed in the modeling program Solid Edge and then 3D printed in PLA-plastic. The model used can be seen in figure 3.2.

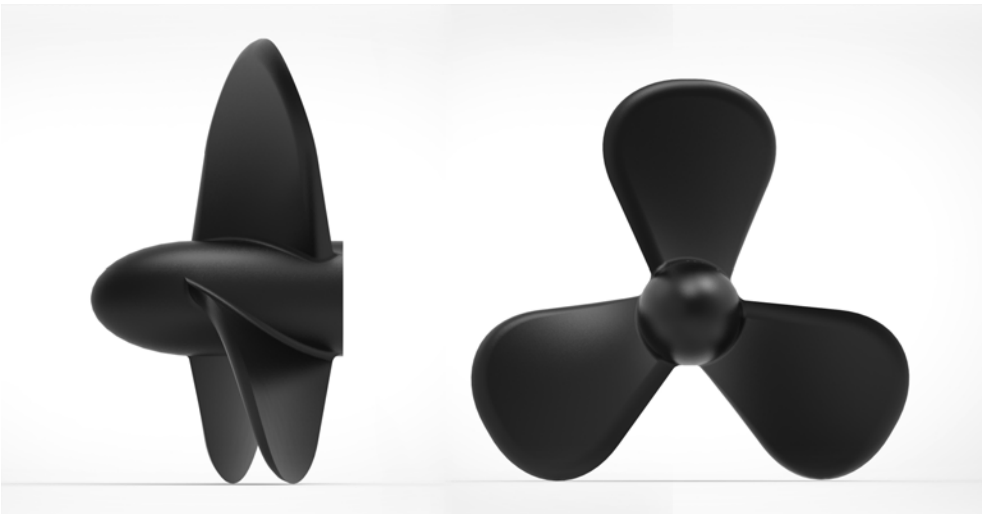


Figure 3.2. Propeller model (made in Solid Edge, rendered in KeyShot)

3.1.3 Filters

The robot's purpose was to collect plastics from the water surface, but in order to do this, the robot needed something to collect the plastic in. Therefore, three filters were designed to fulfill this purpose. The first two filters are designed as grids, and the last one as a frame where a fine mesh net was placed inside. To ensure that the mesh net would not get caught in the motors nor the rudder, the framework of the robot will be extended to include all filters.

To relieve some stress from the mesh, the first two less fine filters were deployed in the design. The first filter, the one closest to the opening, will have the largest gaps, and the second one somewhat smaller. The third and last one will have the fine mesh. With this construction the larger pieces of debris will get caught in the utmost filters and leave only the micro plastic to be collected by the fine-mesh filter. The mesh sizes will be accordingly: 30 mm for the largest, 5 mm for the middle one and 0,1 mm for smallest, illustrated below in figure 3.3.

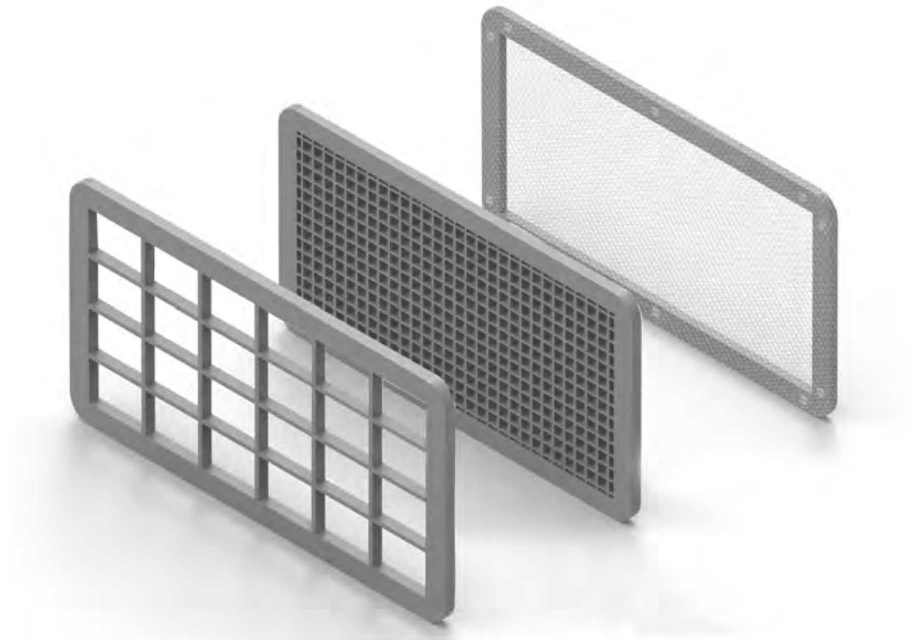


Figure 3.3. The three filters (made in Solid Edge, rendered in KeyShot and Photoshop)

3.1.4 Floating device

To help the robot stay at the surface, two foam pool noodles were attached to each side of the robot. These were made of Low Density Polyethylene (LDPE).

3.2 Electronics

The Arduino UNO is used to control the motors attached to the robot and to control the selected sensor. The Arduino was programmed to alert when the sensor is indicating that the filters are full.

3.2.1 Sensor

Before the final assembly, tests had been executed in order to determine which sensor would be preferable to utilize in the prototype. The sensor used in the prototype was a ultrasonic sensor model HC-SR04 from Elec Freaks, see figure 3.4 below. It will be placed by the first filter to collect data. The placement is at the roof of the box. It will then send a signal to the LED, which will light up when the sensor indicates that the filter is full. For more product features, see Appendix B.

Figure 3.4. Ultra sonic sensor HC-SR04, [17]

3.2.2 DC-motors

Two DC-motors were used to drive the robot forward, see figure 3.5 below. A propeller was placed at each end of the output shaft. The motors require a current of between 1340-2100 Milliampere (mA), and has a speed of 14500 Revolutions Per Minute (RPM), and a voltage of 6 Volt (V) is required for each motor.

Figure 3.5. DC-motors, [18]

However, the motors' performance must be taken into consideration as it will impact the filtration capacity of the robot. A faster motor means that a larger quantity of water can be filtered per second. DC-motors will be used in this project as they were deemed most suitable. They are both cheep as well as easy to install and run. The motors themselves will be powering 3D-printed propellers. In order to install the motors at the end of the robot, rear brackets were designed and 3D-printed. To keep water from entering the motors, a set of cases were also designed and 3D-printed. The cases were then brushed with an adhesive sealant for marine use, in case the 3D-printed parts would be porous. The casings were then finally sealed around the motors with hot glue.

3.3 Final circuit

To control all the parts, they were all connected to the Arduino which is the main component in the circuit. However, the circuit needed external power in order to work. The external power that was supplying the circuit with voltage was one 9 V battery and four 1,5 V batteries. The 9 V battery was directly connected to the Arduino and the other 1,5 V batteries were connected to supply the motors with their needed power.

In addition to that, a L9997ND dual half bridge driver, was combined into the circuit. This can be seen in Appendix C for more information. With the dual half bridge, the motors could be given a preprogrammed route. To rotate the motors in the same direction Motor1 was connected to OUT1 and Motor2 was connected to OUT2. As stated above, the motors were powered with four 1,5 V batteries, which were connected to a Voltage Supply (VS). From PIN 9 and PIN 10 on the Arduino, cables were connected to the input of IN1 and IN2 on the dual half bridge. GND was connected to the Arduino ground. The sensor has four pins, one for the power supply and one for the ground. The other two pins are Trig and Echo that receive the in- and output. These were connected to the Arduino´s PIN 4 and PIN 5. The input was then calculated and if the filter is full, the LED connected to PIN 13 lights up.

3.4 Software

The implemented software had two parts, one for controlling the route and one for detecting blockage in the filters. The part of the code used to steer the robot only relied on output Pulse-Width Modulation (PWM) signals from the Arduino to the motors. This made it possible to regulate their speed. However, the software for detecting litter was slightly more complex. This part of the software was set up after the sensors had been tested, therefore the program is catered to the ultrasonic sensor and not the IR-sensor. The LED's output was determined by the constant input from the sensor. In figure 3.6 below, one can follow the processing of said data.

There were two criteria that needed to be fulfilled for the LED to light up. The first being that the current reading of the distance between the floor and sensor needed to be in an interval of +/- 10 mm of the previous reading. This made it possible for the collected litter to move slightly between the readings but still indicate that something was stuck in the filter. It was also determined that the criteria were not passed if current reading was the same as the distance between the floor and the sensor, which was approximately 110 mm. In other terms, the sensor would not view the floor as an alien object and ignore it. The second criteria was the time perspective. In order for the litter to be viewed as stuck, a certain amount of time had to pass from first to latest detection. After some experimenting, the bar was set at circa 20 seconds, or 1000 readings. If the criteria were at any time broken, the timer would be reset and the LED would, if on, be turned off. The code can be viewed as a whole in Appendix D.

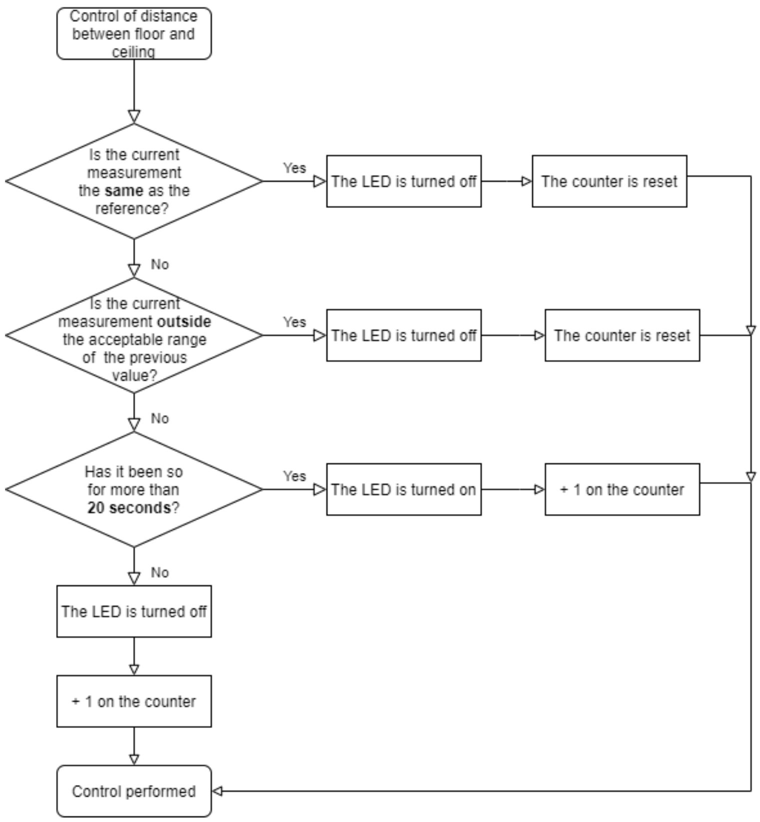


Figure 3.6. Flowchart of the software's operations for detecting blockage (made with diagram.net)

Chapter 4

Testing

To fulfil the project's purpose, several tests needed to be executed. The following chapter addresses how these were performed and how the results were measured.

4.1 Sensor testing

Firstly, the accuracy and range of the sensors needed to be assessed to ensure that they were up to par. Two IR-sensors, of the same type, and one ultrasonic sensor were tested. The tests were carried out by placing the sensors, one at a time, at the end of the table. A tape measure was then placed along the table, making a 90-degree angle with the table's edge. The maximum length of detection for this experiment was set to 1 m due to the table's size. This also gave plenty of margin for the sensor's application in the robot, where the maximum length of detection is approximately 110 mm. As it was discovered that the table reflected some of the emitted radiation once the IR-sensor was too close to its surface, resulting in the sensor giving a false positive, the sensor was held in place approximately 30 mm above the tabletop, which can be seen in figure 4.1. To certify that the test results were equivalent and comparable, the ultrasonic sensor was tested in the same conditions, seen in figure 4.2 below.

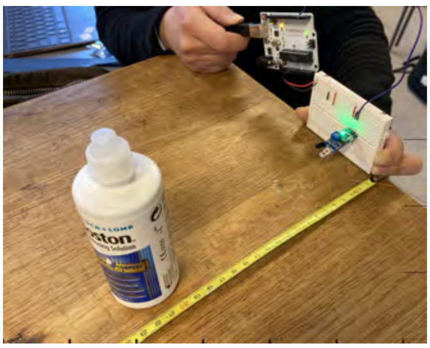


Figure 4.1. Testing conditions for the IR-sensor (picture taken by author)

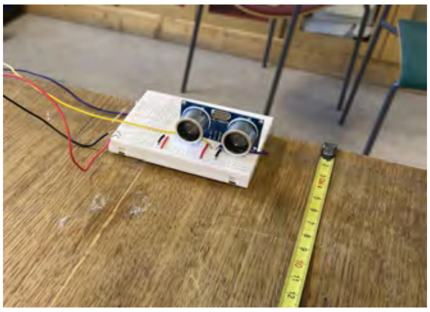


Figure 4.2. Testing conditions for the ultrasonic sensor (picture taken by author)

As the robot's main mission will be to collect plastic, it was essential that the sensor's performance did not rely too heavily on what type of plastic it collected. Therefore, the test was carried out with four different types of plastic, PLA, High Density Polyethylene (HDPE) and LDPE as well as Polystyrene (PS). Although 69,8 % of litter found in coastal environments are plastic, there are still a considerable number of other materials that end up in the water. Therefore, tests with the second to forth most found materials, organic matter (7,7 %), metal (5,9%) and paper and cardboard (5,8%), were carried out as well [19]. The different materials were represented by the following objects, in order: a part of the robot, a liquid container, a plastic bag, a round lid, one of the authors' hands (the wide side), an oval lid and a newspaper as well as carton, which can be seen in figure 4.3.

Figure 4.3. The different test objects, minus the hand, presented in the following order: round lid, liquid container, plastic bag, oval lid, newspaper, carton and robot part (picture taken by author).

The tests were carried out by placing said objects, one at the time, in front of the sensor and moving it backwards until indications were given that the detection had been lost. For the IR-sensor, this meant that only one of the lights would shine but for the ultrasonic sensor, the measured distance would not correlate to the actual distance. Once such an indication was given, the distance between the edge of the table and the object was noted. The distance between the same edge and the sensor's head was then subtracted from said distance. The range was rounded up to the nearest 5 mm and logged. This procedure was then carried out for all eight items and repeated seven times for both subtests.

Due to difficulties in waterproofing the sensors, testing of the instruments in water was not carried out as a safety precaution.

4.2 Engine testing

To ensure that PlaCo would work as a whole, the assembly was tested in the water. The designated testing area was the water of a fountain. There was some turbulence in the water but in comparison with an ocean it could be considered as still. The route used for testing the robot only involved the motors running on 75 % of the maximum speed, straightforward. However, the route could be changed to fit whatever path one might wish.

Chapter 5

Results

In this chapter, the results from the previous chapter *Testing* will be presented. For analysis and examination of the results, see the following chapter *Discussion*.

5.1 Results from sensor testing

In the table below, table 5.1, the data for the distance between the first IR-sensor's head and its last readable point can be seen. The results include readings for all objects and all experiment iterations.

Table 5.1. Table of the resulting range for IR-sensor 1.

Last readable point [mm]							
PS	455	375	185	180	165	230	205
HDPE	280	325	375	155	175	275	215
LDPE	475	465	95	190	190	385	225
PLA	235	240	140	245	355	290	310
Metal	555	425	75	420	245	395	310
Newspaper	485	455	210	210	130	305	230
Carton	665	455	245	255	165	395	350
Organic	355	545	180	170	125	240	185

In the same way the results from the second IR-sensor's readings can be seen in table 5.2.

Table 5.2. Table of the resulting range for IR-sensor 2.

	Last readable point [mm]						
PS	170	165	105	115	210	85	145
HDPE	155	220	130	170	160	105	150
LDPE	145	135	120	260	225	80	160
PLA	300	300	325	150	135	210	275
Metal	245	335	235	235	275	205	230
Newspaper	185	145	170	250	200	135	195
Carton	245	200	240	335	170	160	270
Organic	185	145	145	185	120	120	180

The ultrasonic sensor on the other hand carried out accurate readings at all times and for all eight objects, meaning its last readable point was always at 1 m hence why no table is being presented.

5.2 Results from testing of the robot as a whole

After testing PlaCo in water, the results were that the engines did not work as planned. The results of the test show that the motor´s voltage was too low to drive the PlaCo forward in water.

5.2.1 Buoyancy force

As mentioned in Chapter 2 *Theory*, one must consider the buoyancy force in order to float. By weighing the robot once it was completely assembled and then weighing the part that would be above the surface, the buoyancy force for the robot could be calculated.

In the calculation, the density of water 1000 kg/m^3 was used, because tap water is the easiest fluid to find and test in. After putting all given parameters in the equations mentioned in chapter 2 Theory, the buoyancy force could be calculated to 19.613 Newton.

Chapter 6

Discussion

This chapter will contain a discussion about the final results of this project.

6.1 Sensors

It can be concluded, in accordance with the data, that both IR-sensors were far more unstable than the ultrasonic sensor. None of the materials, in any of the readings, came close to 1 m. Several explanations can be found for this. The first being the tuning of the distance adjuster on the sensor. It was rather difficult to alter the adjuster and the sensor would still at times, even after being set to accommodate the testing conditions, give a false positive resulting in the sensor having to be tuned again. Another source of error could be the objects´ sizes and shapes, as could their colour. It was discovered that if the object was placed slightly to the side of the middle line, then the IR-sensor would not indicate a detection. The sensor was particularly sensitive if the object was round, like the liquid container. However, this was also the case with the ultrasonic sensor. It would too give a false negative if the object was not perfectly centred between the emitting and receiving sensor heads. This could potentially be problematic in future iterations of the robot. For example, if the placement of the sensor does not coincide with the collected litter, then there will be no indication that the filter is full. The ultrasonic sensor did not though, in contrast to the IR-sensor, have a colour preference. When a darker object was positioned in front of the IR-sensor, it either detected the object much closer than its lighter counterpart or not at all. This would give uncertainties in the robot's function.

The ultrasonic sensor was therefore better on two accounts with the slight problem of alignment between litter and sensor. As a result, the ultrasonic sensor would be the preferable sensor. At present, the sensor's function would be satisfactory but one or more iterations of PlaCo would be needed for its optimal function.

6.2 Motors

In the final assembly of the robot, the propellers were placed at the end of the motor shaft. The shaft was placed in a small hole in the propeller that was made to fit the measurements of the motor shaft. This was done so that the propeller would rotate with the same speed as the motor. Hot glue was then used to secure the placement. This turned out not to be the most optimal solution of securing the propellers. When programming the motors to go at full speed, the centrifugal force created from the rotating shaft also acts on the propellers, which resulted in the propellers begin pushed away from the shaft when not secured properly. The shaft of the motor has a length of 10,1 millimeters which can also be a factor to the weak securing. With a small contact surface, the centrifugal force can become too great for the mounting.

After multiple attempts trying to secure the propellers to the motors, another solution to the problem was tried. The speed of the motors was reprogrammed to go at 75 percent speed instead of 100 percent. This method made the centrifugal force less intense which resulted in that the propellers now stayed in place. However, as a consequence, this reduction affects the robot's cleaning and filtration of water because it is dependent on the speed.

After the testing of PlaCo in water, it stood clear that the question asking how much water can be cleaned could not be answered. To answer this question, a velocity of the robot in water would have been necessary.

Chapter 7

Recommendations and further work

As stated in the *Discussion* some measures are needed for optimal function of PlaCo. The most stressing of these measures is to implement a waterproof ultrasonic sensor. This would mean that the robot could function in water and PlaCo could be tested as a whole. In order to get over the problem with the sensor-litter-alignment, more sensors could be installed per filter and on all filters. This was of course the idea, but time and budget issues did not allow it. If a more sensitive ultrasonic sensor was available, it would then be a good idea to implement it, especially for the finer filters. The plastic particles there would not be large enough for the sensor used in this iteration to detect. It would only permit detection if a large build up was created which would harm the flow of water through the robot.

As stated in Chapter 5 *Results*, PlaCo did not have enough powerful motors to be able to push itself forward. The motors used in this project had a voltage of 6 V, which was too weak to overcome the water viscosity. The recommendation on future work would be to select a motor with higher voltage, perhaps even order a range of motors with different voltages. Then test which motor that meets all the requested requirements.

Another recommendation would be to find a better way to secure the propellers to the motor shaft. The motor shaft could be made longer and the securing hole in the propeller could be made deeper. In addition to that, find a better complement to hot glue.

Once the problems concerning the key functions of the robot have been fixed, one could look at implementing solar panels on PlaCo. This would be an important step toward the robot being self-sufficient and even better for the environment. If PlaCo was used as a scientific instrument for sampling, for example in a lake, then a geographical sensor could be used to implement a route.

Bibliography

[1] United Nations. *Oceans*. URL: https://www.un.org/sustainabledevelopment/oceans/. (accessed: 2021.01.29).

[2] Håll Sverige Rent. *Skräp i havet*. URL: https://hsr.se/skrap-i-havet. (accessed: 2021.04.20).

[3] Unesco. *Facts and figures on marine pollution — United Nations Educational, Scientific and Cultural Organization*. URL: http://www.unesco.org/new/en/natural-sciences/ioc-oceans/focus-areas/rio-20-ocean/blueprint-for-the-future-we-want/marine-pollution/facts-and-figures-on-marine-pollution. (accessed: 2021.01.29).

[4] United Nations. *The Sustainable Development Agenda*. URL: https://www.un.org/sustainabledevelopment/development-agenda. (accessed: 2021.01.29).

[5] Science Advice for Policy by European Academies SAPEA. *A scientific perspective on microplastics in nature and society*. Berlin, 2019. DOI: https://doi.org/10.26356/microplastics. (accessed: 2021.04.12).

[6] C. Arthur J. Baker and H. Bamford (eds). *Proceedings of the International Research Workshop on the Occurrence, E
ects, and Fate of Microplastic Ma-rine Debris*. Tacoma, Sept 9-11,2008. URL: https://marinedebris.noaa.gov/proceedings-international-research-workshop-microplastic-marine-debris. (accessed: 2021.04.27).

[7] 5 Gyres Institute. *Trawl for Plastic*. URL: https://www.5gyres.org/trawl-resources. (accessed: 2021.02.12).

[8] Arduino. *Introduction*. URL: https://www.arduino.cc/en/Guide/Introduction.(accessed: 2021.05.08).

[9] SparkFun Electronics. URL: https://www.flickr.com/photos/sparkfun/8406865680/. (accessed: 2021.04.12).

[10] H. Johansson. *Elektroteknik*. Swedish. Stockholm: Kungliga Tekniska Högskolan, 2013, pp. 287–298.

[11] Pxhere. *AA-batteri*. URL: https://pxhere.com/sv/photo/1635099?utm_content=shareClip&utm_medium=referral&utm_source=pxhere. (accessed: 2021.05.09).

[12] FierceElectronics. *What is an IR sensor?* URL: https://www.fierceelectronics.com/sensors/what-ir-sensor. (accessed: 2021.02.12).

[13] NASA Science. *Infrared Waves — Science Mission Directorate.* URL: https://science.nasa.gov/ems/07_infraredwaves. (accessed: 2021.02.12).

[14] FierceElectronics. *What is an Ultrasonic sensor?* URL: https://www.fierceelectronics.com/sensors/what-ultrasonic-sensor. (accessed: 2021.04.20).

[15] Roderick Burnett. *Understanding How Ultrasonic Sensors Work?* URL: https://www.maxbotix.com/articles/how-ultrasonic-sensors-work.htm. (accessed: 2021.04.20).

[16] R.A. Freedman H.D. Young. *University Physics with Modern Physics.* English. Fourteenth Edition. Essex, England: Pearson, 2016, pp. 400–401.

[17] Electrokit. *Avståndsmätare ultraljud HC-SR04 2 – 400cm.* URL: https://www.electrokit.com/produkt/avstandsmatare-ultraljud-hc-sr04-2-400cm/. (accessed: 2021.05.08).

[18] Electrokit. *DC-motor 6V 250mA 14500rpm.* URL: https://www.electrokit.com/en/product/dc-motor-6v-250ma-14500rpm-2/. (accessed: 2021.04.26).

[19] Håll Sverige Rent. *Vanligaste skräpet på stränder.* URL: https://skola-kommun.hsr.se/fakta/statistikportal/skrapmatningar-pa-strander/olika-fraktioner. (accessed: 2021.02.05).

Appendix A

Simulation in Acumen

```
// Simulation in Acumen

// Bachelor's Thesis in Mechatronics at ITM, KTH

//PlaCo - Plastic collecting robot

// TRITA-ITM-EX-2021:46

// Program by Annie Persson and Johanna Bergsten

// Last edit: 2021-05-25

//

// This program is used to simulate PlaCo’s movements in water with the help of
// Acumen. In the simulation the robot moves forward at a constant speed for a set
// amount of time.

// The code firstly specifies said movements before the robot’s construction is
// determined. It is built in segments. The outer and inner box were combined in
// to one box as it was deemed unnecessary for the simulated movements to have
// both. Some other simplifications of the model have also been made.

//

// Simulation of the robot’s movements, PlaCo will move straight forward, along the
// x-axis

model Main(simulator) =

initially

r = create Robot((0,0,0),white),

x1=0, x1'=0, x1''=0,

z = 0

always

  if x1<50                          // The movement will continue for 50 step units

    then x1'' = -(x1'-10)

    else if x1'>0

           then x1'' = -10

           else x1'' = 0,

    r.pos = (x1,0,0),

    z = x1


// Model of the robot

model Robot(pos,col) =

initially

_3D = (), _Plot=()

Always
```

```
_3D= ((Box                                 // Type of object, the bottom plate
         center= pos+ (0,0,0)              // Centre point
         size=(10,7,0.2)                   // Size [x,y,z]
         color=white                       // Colour
         rotation=(0,0,0))                 // Orientation in the plane

      (Box                                 // Type of object, roof
         center= pos+ (0,0,4)              // Centre point
         size=(10,7,0.2)                   // Size [x,y,z]
         color=white                       // Colour
         rotation=(0,0,0))                 // Orientation in the plane

      (Box                                 // Type of object, left sidewall
         center= pos+ (0,-3.6,2)           // Centre point
         size=(10,0.2,4.2)                 // Size [x,y,z]
         color=white                       // Colour
         rotation=(0,0,0))                 // Orientation in the plane

      (Box                                 // Type of object, right sidewall
         center= pos+ (0,3.6,2)            // Centre point
         size=(10,0.2,4.2)                 // Size [x,y,z]
         color=white                       // Colour
         rotation=(0,0,0))                 // Orientation in the plane

      (Box                                 // Type of object, first filter
         center= pos+ (3,0,2)              // Centre point
         size=(0.1,7,3.8)                  // Size [x,y,z]
         color=red                         // Colour
         rotation=(0,0,0))                 // Orientation in the plane

      (Box                                 // Type of object, second filter
         center= pos+ (0,0,2)              // Centre point
         size=(0.1,7,3.8)                  // Size [x,y,z]
         color=red                         // Colour
         rotation=(0,0,0))                 // Orientation in the plane
```

```
(Box                                    // Type of object, third filter
    center= pos+ (-4.95,0,2)            // Centre point
    size=(0.1,7,3.8)                    // Size [x,y,z]
    color=red                           // Colour
    rotation=(0,0,0))                   // Orientation in the plane

(Box                                    // Type of object, float. holder
    center= pos+ (0,0,4.2)              // Centre point
    size=(1,12,0.2)                     // Size [x,y,z]
    color=black                         // Colour
    rotation=(0,0,0))                   // Orientation in the plane

(Cylinder                               // Type of object, left floatation
                                        // device
    center= pos+ (0,-6,3.1)             // Centre point
    size=(15,1)                         // Size [length, radius]
    color=cyan                          // Colour
    rotation=(0,0,1.5))                 // Orientation in the plane

(Cylinder                               // Type of object, right floatation
                                        // device
    center= pos+ (0,6,3.1)              // Centre point
    size=(15,1)                         // Size [length, radius]
    color=cyan                          // Colour
    rotation=(0,0,1.5))                 // Orientation in the plane

(Cylinder                               // Type of object, left motor
    center= pos+ (-4.5,3,-0.60)         // Centre point
    size=(2,0.5)                        // Size [length, radius]
    color=white                         // Colour
    rotation=(0,0,1.5))                 // Orientation in the plane

 (Cylinder                              // Type of object, right motor
    center= pos+ (-4.5,-3,-0.60)        // Centre point
    size=(2,0.5)                        // Size [length, radius]
    color=white                         // Colour
    rotation=(0,0,1.5))                 // Orientation in the plane
```

```
        (Cylinder                               // Type of object, left propeller
           center= pos+ (-5.6,-3,-0.60)         // Centre point
           size=(0.2,1)                         // Size [length, radius]
           color=black                          // Colour
           rotation=(0,0,1.5))                   // Orientation in the plane

        (Cylinder                               // Type of object, right propeller
           center= pos+ (-5.6,3,-0.60)          // Centre point
           size=(0.2,1)                         // Size [length, radius]
           color=black                          // Colour
           rotation=(0,0,1.5))                  // Orientation in the plane

        (Box                                    // Type of object, house for
                                                // electronics
           center= pos+ (-2,0,4.1)              // Centre point
           size=(3,4,3)                         // Size [x,y,z]
           color=white                          // Colour
           rotation=(0,0,0))                    // Orientation in the plane
)
```

Appendix B

Ultrasonic Ranging Module HC - SR04

Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:
(1) Using IO trigger for at least 10us high level signal,
(2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
(3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.
Test distance = (high level time×velocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

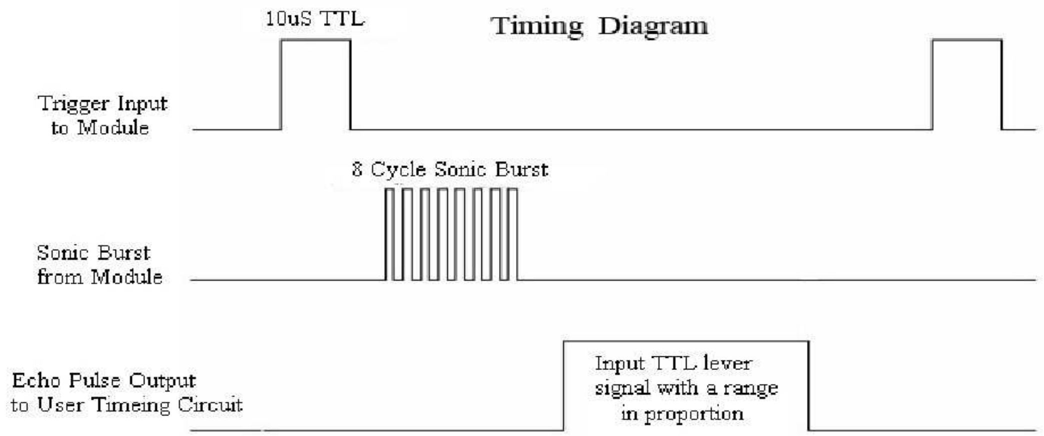
Electric Parameter

Working Voltage	**DC 5 V**
Working Current	**15mA**
Working Frequency	**40Hz**
Max Range	**4m**
Min Range	**2cm**
MeasuringAngle	**15 degree**
Trigger Input Signal	**10uS TTL pulse**
Echo Output Signal	**Input TTL lever signal and the range in proportion**
Dimension	**45*20*15mm**

Vcc Trig Echo GND

Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: uS / 58 = centimeters or uS / 148 =inch; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



Attention:

- The module is not suggested to connect directly to electric, if connected electric, the GND terminal should be connected the module first, otherwise, it will affect the normal work of the module.
- When tested objects, the range of area is not less than 0.5 square meters and the plane requests as smooth as possible, otherwise ,it will affect the results of measuring.

www.Elecfreaks.com

Appendix C

L9997ND Dual half bridge driver

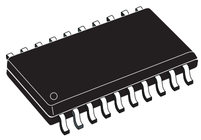
To drive the motors in the Water Trash Collecter a *L9997ND Dual half bridge* was used. In the data sheet below it´s description, electrical properties, mechanical data and more can be found.

L9997ND

DUAL HALF BRIDGE DRIVER

- HALF BRIDGE OUTPUTS WITH TYPICAL $R_{ON} = 0.7\Omega$
- OUTPUT CURRENT CAPABILITY ±1.2A
- OPERATING SUPPLY VOLTAGE RANGE 7V TO 16.5V
- SUPPLY OVERVOLTAGE PROTECTION FUNCTION FOR V_{VS} UP TO 40V
- VERY LOW QUIESCENT CURRENT IN STANDBY MODE < 1µA
- CMOS COMPATIBLE INPUTS WITH HYSTERESIS
- OUTPUT SHORT-CIRCUIT PROTECTION
- THERMAL SHUTDOWN
- REAL TIME DIAGNOSTIC: THERMAL OVERLOAD, OVERVOLTAGE

MULTIPOWER BCD TECHNOLOGY



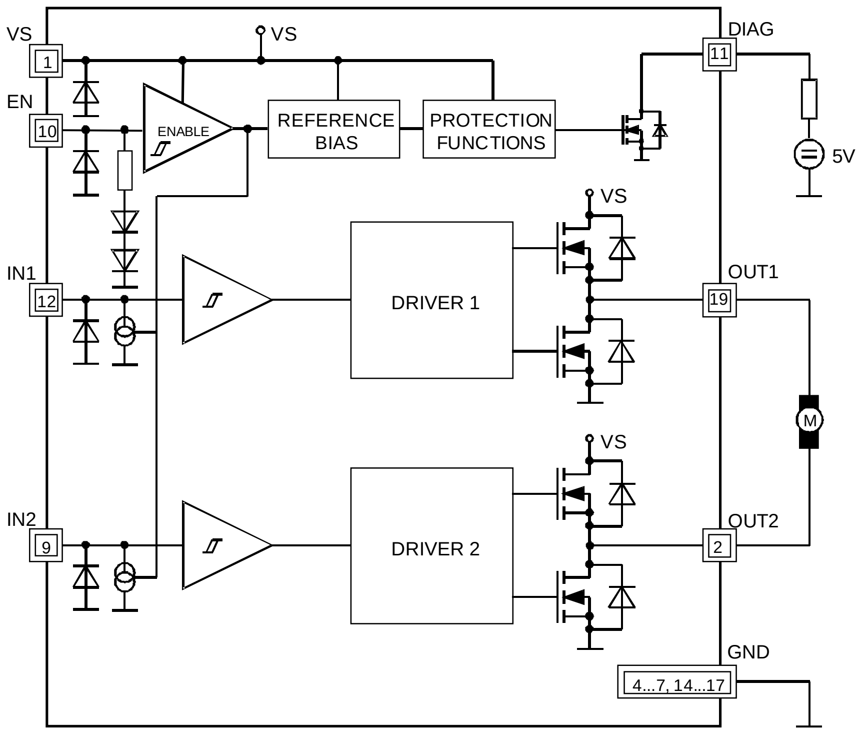
SO20 (12+4+4)

ORDERING NUMBERS: L9997ND
L9997ND013TR

DESCRIPTION

The L9997ND is a monolithic integrated driver, in BCD technology intended to drive various loads, including DC motors. The circuit is optimized for automotive electronics enviromental conditions.

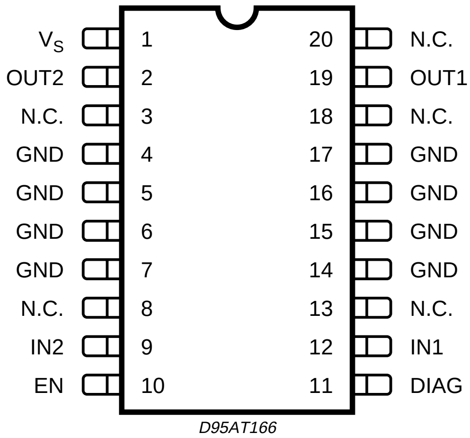
BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{VSDC}	DC Supply Voltage	-0.3 to 26	V
V_{VSP}	Supply Voltage Pulse (T < 400ms)	40	V
I_{OUT}	DC Output Current	±1.8	A
$V_{IN1,2}$	DC Input Voltage	-0.3 to 7	V
V_{EN}	Enable Input Voltage	-0.3 to 7	V
V_{DIAG}	DC Output Voltage	-0.3 to 7	V
I_{OUT}	DC Output Short-circuit Current -0.3V < V_{OUT} < V_S + 0.3V	internally limited	
I_{DIAG}	DC Sink Current -0.3V < V_{DG} < 7V	internally limited	

PIN CONNECTION (Top view)



PIN FUNCTIONS

N.	Name	Function
1	VS	Supply Voltage
2	OUT2	Channel 2: Push-Pull power output with intrinsic body diode
3, 8, 13, 18,20	NC	NC: Not Connected
4 to 7, 14 to 17	GND	Ground: signal - and power - ground, heat sink
9	IN2	Input 2: Schmitt Trigger input with hysteresis (non-inverting signal control)
10	EN	Enable: LOW or not connected on this input switches the device into standby mode and the outputs into tristate
11	DIAG	Diagnostic: Open Drain Output that switches LOW if overvoltage or overtemperature is detected
12	IN1	Input 1: Schmitt Trigger input with hysteresis (non-inverting signal control)

THERMAL DATA

Symbol	Parameter	Value	Unit
T_{jTS}	Thermal Shut-down Junction Temperature	165	°C
T_{jTSH}	Thermal Shut-down Threshold Hysteresis	25	K
$R_{th\ j\text{-}amb}$	Thermal Resistance Junction-Ambient (1)	50	K/W
$R_{th\ j\text{-}pins}$	Thermal Resistance Junction-Pins	15	K/W

(1) With 6cm^2 on board heatsink area.

ELECTRICAL CHARACTERISTICS (7V < V_S < 16.5V; -40°C < T_J < 150°C; unless otherwise specified.)

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Unit
I_{VS_SB}	Quiescent Current in Standby Mode	V_{EN} < 0.3V; V_{VS} <16.5V; T_j < 85°(*) V_{EN} = 0; V_{VS} = 14.5V; T_j = 25°C		<1 <1	90 10	µA µA
I_{VS}	Supply Current	EN = HIGH, $I_{OUT1,2}$ = 0		2	6	mA
V_{ENL}	Low Enable Voltage				1.5	V
V_{ENH}	High Enable Voltage		3.5		6	V
V_{ENthh}	Enable Threshold Hysteresis			1		V
I_{EN}	Enable Input Current	V_{EN} = 5V		85	250	µA
$V_{IN1,2L}$	Low Input Voltage				1.5	V
$V_{IN1,2H}$	High Input Voltage		3.5			V
$V_{IN1,2thh}$	Input Threshold Hysteresis			1		V
$I_{IN1,2}$	Input Bias Current	V_{IN} = 0 V_{IN} = 5V, EN = HIGH	-3 2	0 10	1 50	µA µA
$R_{ON\ OUT1,2}$	ON-Resistance to Supply or GND	I_{OUT} = ±0.8A; V_{VS} = 7V; T_j = 125°C I_{OUT} = ±0.8A; V_{VS} = 12V; T_j = 125°C I_{OUT} = ±0.8A; V_{VS} = 12V; T_j = 25°C		1.2 1.1 0.7	2.8 2.25	Ω Ω Ω
$\|I_{OUT1,2}\|$	Output Current Limitation		1.2	1.6	2.2	A
V_{DIAG}	Diagnostic Output Drop	I_{DIAG} = 0.5mA, EN = HIGH Overvoltage or Thermal Shut-down			0.6	V
V_{VSOVth}	Supply Overvoltage Threshold		17	19	21	V
t_{ONLH}	Turn on Delay Time	See Fig. 2; V_{VS} = 13.5V Measured with 93Ω load		50	150	µs
t_{ONHL}				30	150	µs
t_{OFFHL}	Turn off Delay Time			10	100	µs
t_{OFFLH}				2	20	µs
t_{dHL}	Rising Delay Time			115	250	µs
t_{dLH}	Falling Delay Time			115	250	µs
t_{rHS}	Rise Time			30	100	µs
t_{rLS}				60	150	µs
t_{fHS}	Fall Time			25	100	µs
t_{fLS}				50	150	µs

* Tested at 125°C and guaranteed by correlation

FUNCTIONAL DESCRIPTION

The L9997ND is a motor driver two half-bridge outputs, intended for driving dc motors in automotive systems. The basic function of the device is shown in the Table 1.

Table 1. Table function.

Status	EN	IN1	IN2	OUT1	OUT2	DIAG	NOTE
1	L	X	X	Tristate	Tristate	OFF	Standby Mode
2	H	H	H	SRC	SRC	OFF	Recommended for braking
3	H	H	L	SRC	SNK	OFF	
4	H	L	H	SNK	SRC	OFF	
5	H	L	L	SNK	SNK	OFF	
6	H	X	X	Tristate	Tristate	ON	Overvoltage or Overtemperature

The device is activated with enable input voltage HIGH. For enable input floating (not connected) or LOW the device is in Standby Mode. Very low quiescent current is defined for $V_{EN} < 0.3V$. When activating or disactivating the device by the enable input a wake-up time of 50µs is recommended.

For braking of the motor the status 2 is recommended. The reason for this recommendation is that the device features higher threshold for initialisation of parasitic structures than in state 5.

The inputs IN1, IN2 features internal sink current generators of 10µA, disabled in standby mode. With these input current generators the input level is forced to LOW for inputs open. In this condition the outputs are in SNK state.

The circuit features an overvoltage disable function referred to the supply voltage V_{VS}. This function assures disabling the power outputs, when the supply voltage exceeds the over voltage threshold value of 19V typ. Both outputs are forced to tristate in this condition and the diagnostic output is ON.

The thermal shut-down disables the outputs (tristate) and activates the diagnostic when the junction temperature increases above the thermal shut-down threshold temperature of min. 150°C. For the start of a heavy loaded motor, if the motor current reaches the max. value, it is necessary to respect the dynamical thermal resistance junction to ambient. The outputs OUT1 and OUT2 are protected against short circuit to GND or V_S, for supply voltages up to the overvoltage disable threshold.

The output power DMOS transistors works in linear mode for an output current less than 1.2A. Increasing the output load current (> 1.2A) the output transistor changes in the current regulation mode, see Fig.6, with the typical output current value below 2A. The SRC output power DMOS transistors requires a voltage drop ~3V to activate the current regulation. Below this voltage drop is the device also protected. The output current heat up the power DMOS transistor, the R_{DSON} increases with the junction temperature and decreases the output current. The power dissipation in this condition can activate the thermal shutdown . In the case of output disable due to thermal overload the output remains disabled untill the junction temperature decreases under the thermal enable threshold.

Permanent short circuit condition with power dissipation leading to chip overheating and activation of the thermal shut-down leads to the thermal oscillation. The junction temperature difference between the switch ON and OFF points is the thermal hysteresis of the thermal protection. This hysteresis together with the thermal impedance and ambient temperature determines the frequency of this thermal oscillation, its typical values are in the range of 10kHz.

The open drain diagnostic output needs an external pull-up resistor to a 5V supply. In systems with several L9997ND the diagnostic outputs can be connected together with a common pull-up resistor. The DIAG output current is internally limited.

Fig. 1 shows a typical application diagram for the DC motor driving. To assure the safety of the circuit in the reverse battery condition a reverse protection diode D_1 is necessary. The transient protection diode D_2 must assure that the maximal supply voltage V_{VS} during the transients at the V_{BAT} line will be limited to a value lower than the absolute maximum rating for V_{VS}.

Figure 1: Application Circuit Diagram.

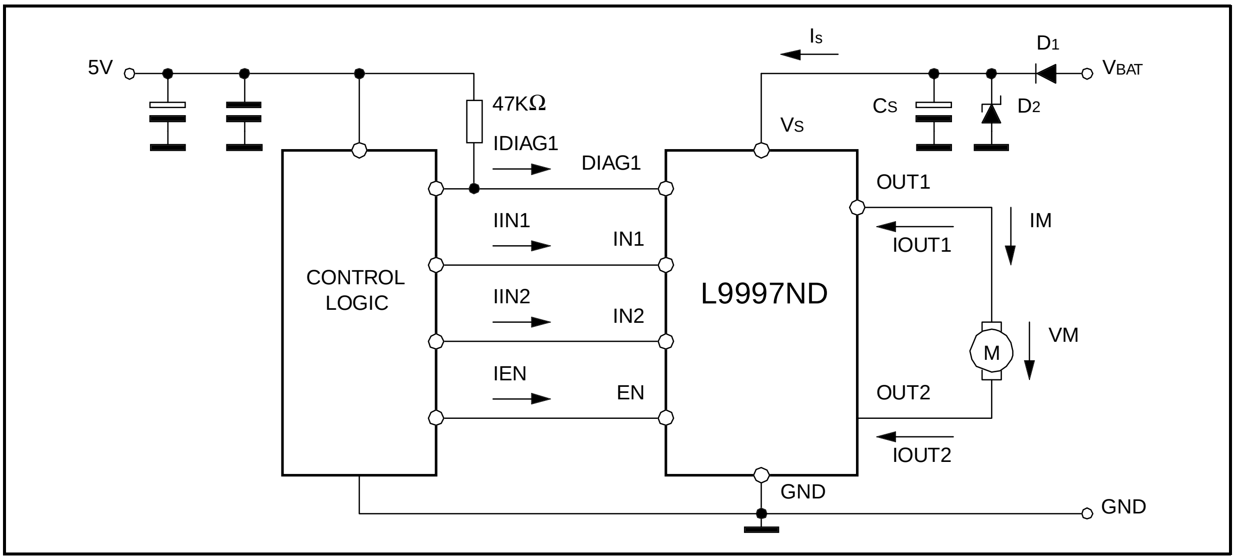


Figure 2. Timing Diagram.

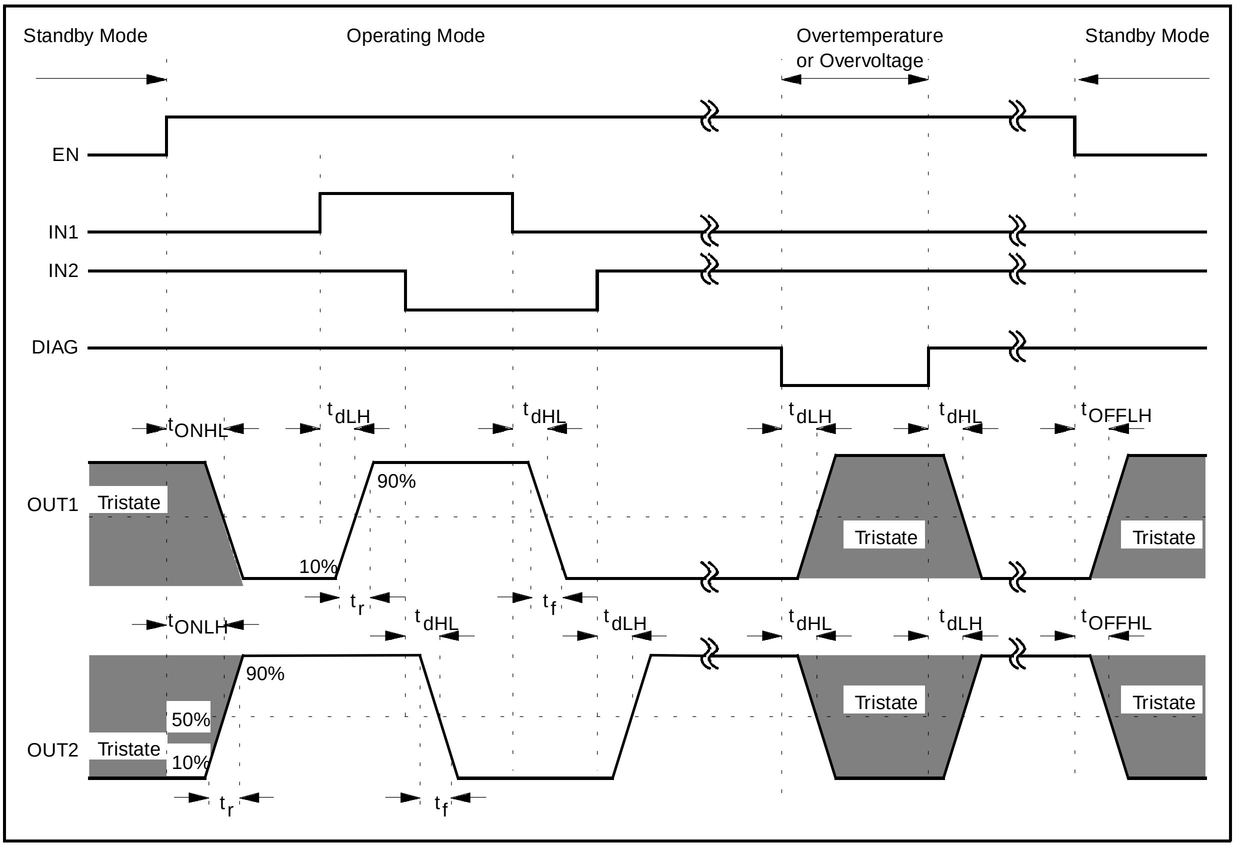


Figure 3. Typical R_{ON} - Characteristics of Source and Sink Stage

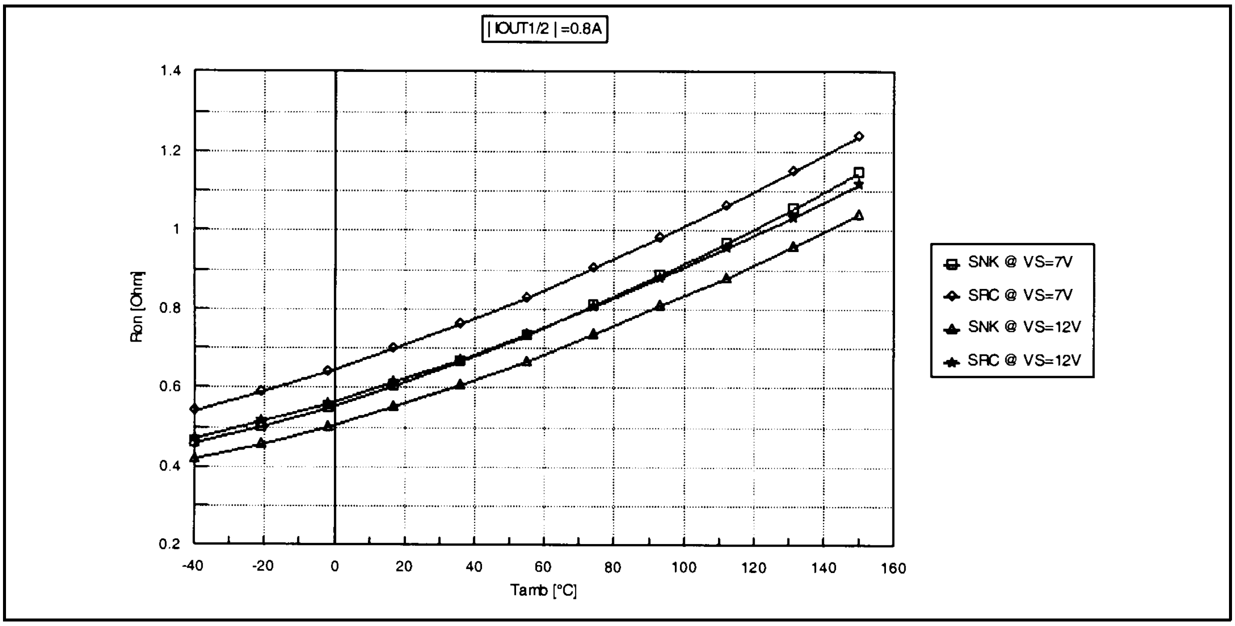


Figure 4. Quiescent current in standby mode versus supply voltage.

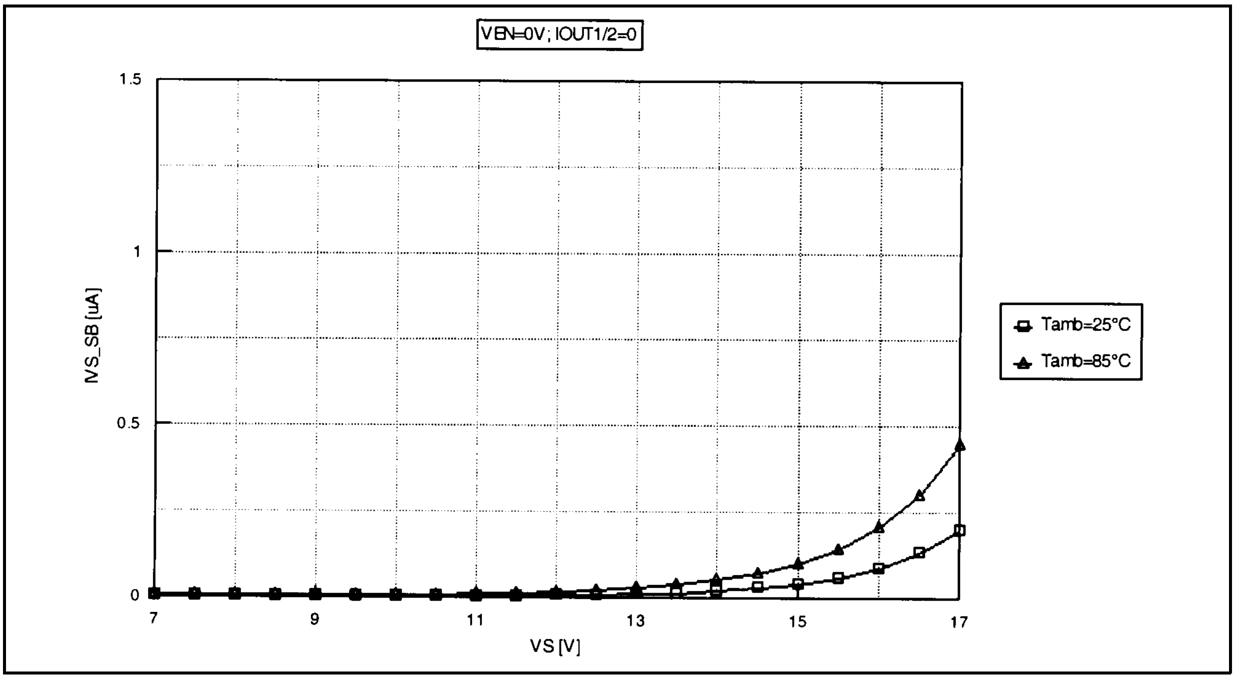


Figure 5. ON-Resistance versus supply voltage.

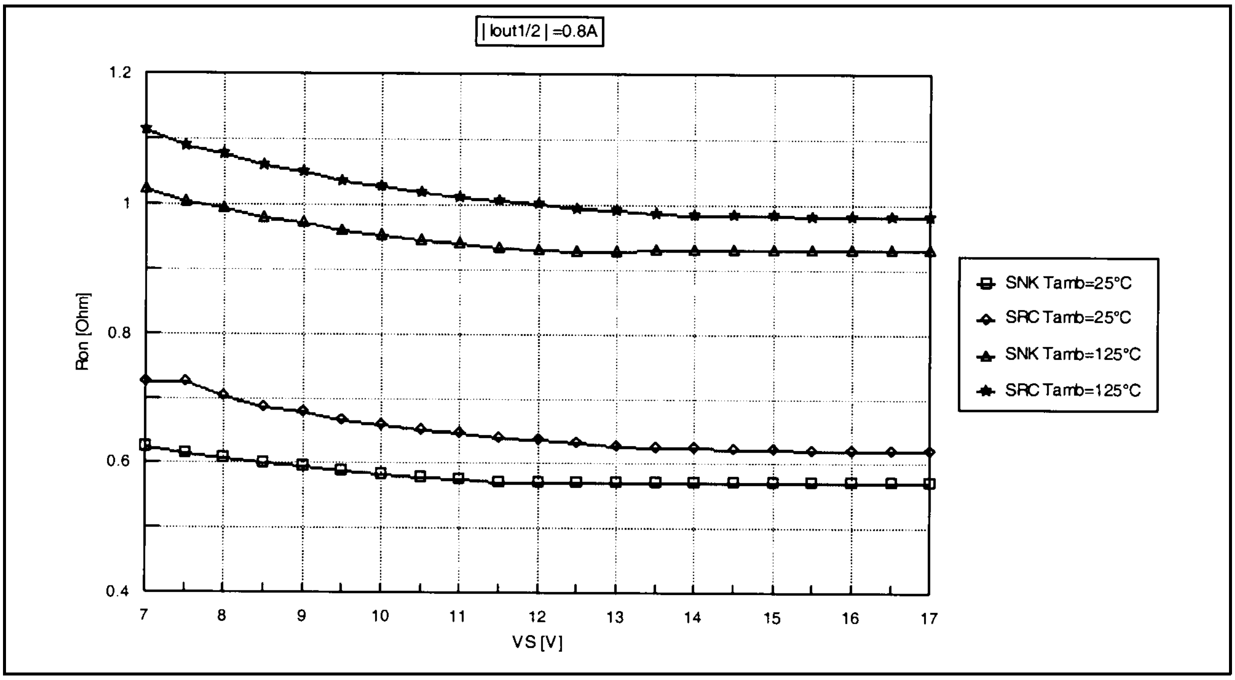


Figure 6. I_{OUT} **versus** V_{OUT} **(pulsed measurement with** T_{ON} **= 500µs,** T_{OFF} **= 500ms).**

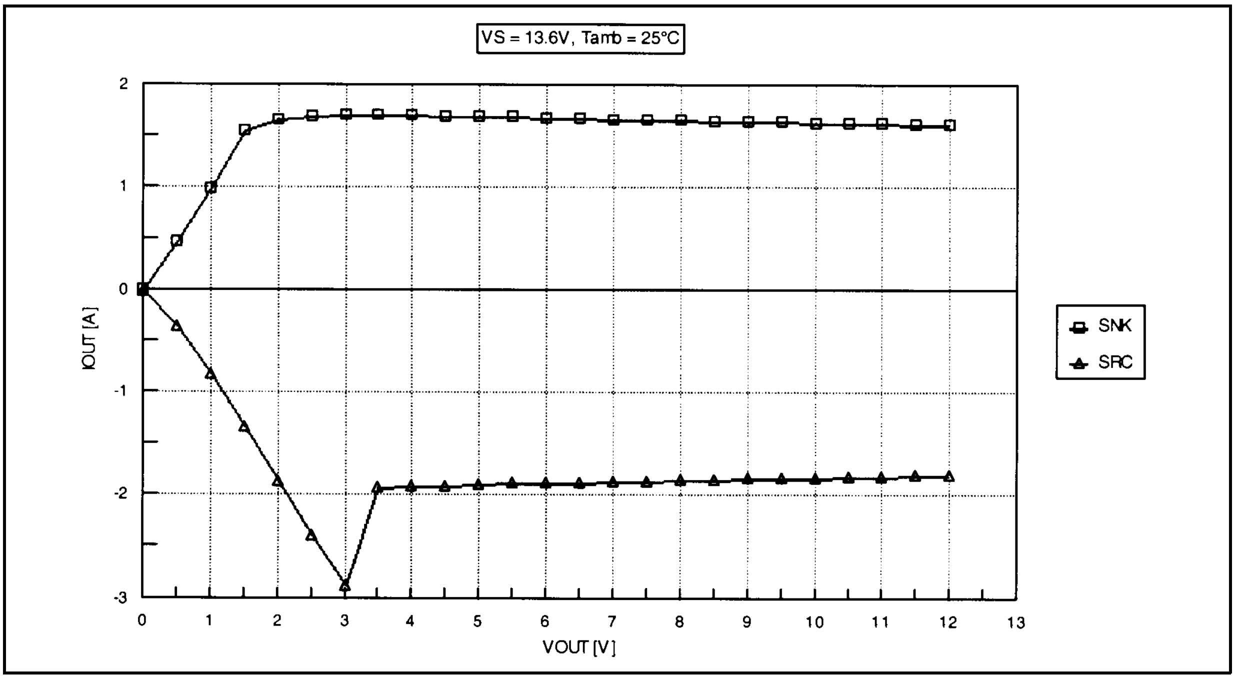
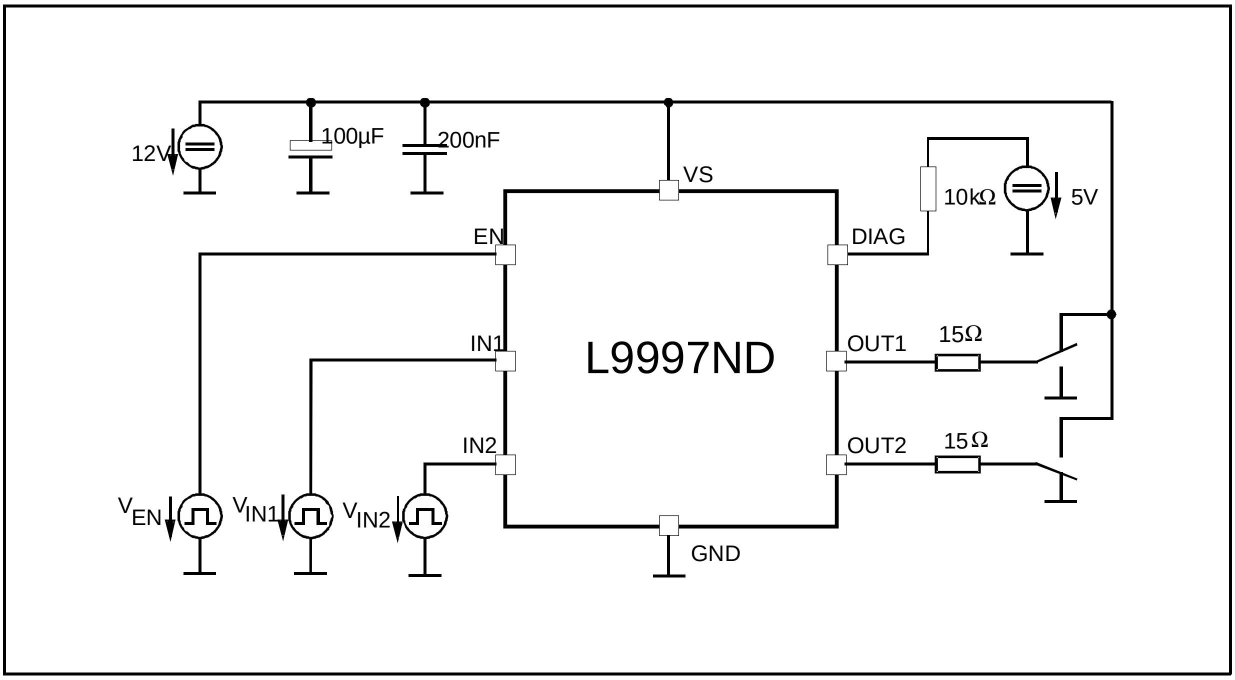
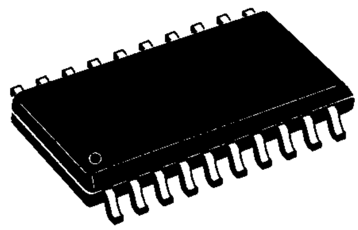


Figure 7. Test circuit.

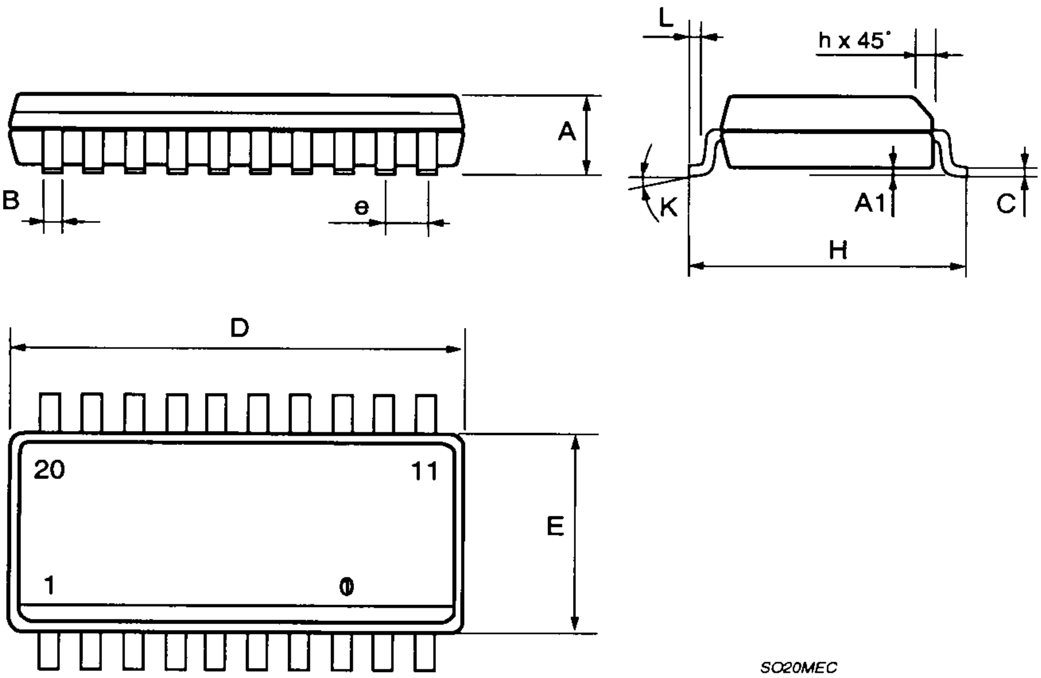


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A	2.35		2.65	0.093		0.104
A1	0.1		0.3	0.004		0.012
B	0.33		0.51	0.013		0.020
C	0.23		0.32	0.009		0.013
D	12.6		13	0.496		0.512
E	7.4		7.6	0.291		0.299
e		1.27			0.050	
H	10		10.65	0.394		0.419
h	0.25		0.75	0.010		0.030
L	0.4		1.27	0.016		0.050
K	0° (min.)8° (max.)					

OUTLINE AND MECHANICAL DATA



SO20



Please Read Carefully:

Appendix D

Arduino code

```
/*
*
* Bachelor's Thesis in Mechatronics at ITM, KTH
* PlaCo - Plastic collecting robot
* TRITA-ITM-EX-2021:46
* Program by Annie Persson and Johanna Bergsten
* Last edit: 2021-05-09
*
* This program is used for controlling PlaCo- the plastic collecting water robot.
* The robot collects marine debris such as plastic with the help of three filers.
* One of the filters is provided with an ultrasonic sensor which tells the user
* when the filter is full. This is done by lighting up a LED.
* The program also provides a route for the robot's motors through a motor driver.
* The program is uploaded to an Arduino UNO connected to the previously mentioned
* components.
* The code stores the value from the previous sensor reading to use as a reference
* for the current value. This makes it possible to determine if an object is stuck
* in the filter or not. This is done for every reading. If the current reading
* is in the set range of the previous reading, +1 will be added to a
* counter. If the criteria is met a predetermined number of times the LED will
* be turned on, indicating a full filter. If the current reading is not with in
* the range then the counter will reset and the LED, if turned on, will turn off.
*
*/

// Defining global variables
int trig = 4;     // Constant, Arduino pin connected to Ultrasonic Sensor's TRIG pin
int echo = 5;     // Constant, Arduino pin connected to Ultrasonic Sensor's ECHO pin
int LED = 13;     // Constant, Arduino pin connected to LED pin
int mot1 = 9;     // Constant, Arduino pin connected to motor 1 pin
int mot2 = 10;    // Constant, Arduino pin connected to motor 2 pin
int ref = 11;     // Constant, Distance from the sensor to the robot’s floor [cm]
int n = 1;        // Constant, +/- n = acceptable range for read values [cm]
int m = 1000;     // Constant, Number of readings for the filter to be considered
full, approximately 20 sec
int j = 0;        // Variable, Counter for number of readings within the acceptable
range
long duration;    // Variable, Duration of sound wave travel
```

```
long previousDist = 0; // Variable, Distance measurement of previous value [cm]
long currentDist = 0;  // Variable, Distance measurement of current value [cm]

void setup()
{
// Defining input/output mode for the Arduino's pins
  pinMode(trig,OUTPUT);
  pinMode(echo, INPUT);
  pinMode(LED, OUTPUT);
  pinMode(mot1,OUTPUT);
  pinMode(mot2,OUTPUT);
// Initialize serial port
  Serial.begin(9600);
}

void loop()  {
// Motor route, the robot moves straight forward at constant speed
  analogWrite(mot1,191);    // Duty cycle for motor 1 is 75% of max. speed
  analogWrite(mot2,191);    // Duty cycle for motor 2 is 75% of max. speed



// Set up to read the input from the ultrasonic sensor
// Clears the trig condition
   digitalWrite(trig, LOW);
   delayMicroseconds(2);
// Generate 10-microsecond pulse to the trig pin
   digitalWrite(trig, HIGH);
   delayMicroseconds(10);
   digitalWrite(trig, LOW);
// Reads the echo pin and calculates the distance [cm]
   duration = pulseIn(echo, HIGH);           // Returns sound wave travel time [ms]
   currentDist = duration * 0.034 / 2;       // Speed of sound wave divided by 2
(back and forth)

// Displays the distance on the Serial Monitor
   Serial.print("Distance: ");
   Serial.print(currentDist);
```

```
   Serial.println(" cm");


// Checks if the sensor detects blockage
// Checks if the current distance is not the same as the reference i.e., the
// distance between floor and sensor AND inside the range of +/- n mm of the
// previous value
   if (currentDist != ref && currentDist <= previousDist + n && currentDist >=
previousDist - n) {
          j++;                                            // Counter + 1
      if (j >= m) {          // Checking if an item have been stuck for long enough
         digitalWrite(LED,HIGH);        //...then the LED will be turned on
    } else {
         digitalWrite(LED,LOW);         //...if not, the LED will stay turned off
    }
   } else {             // If the current distance does not fulfil the criteria...
      j = 0;                            //...then the counter will be reset...
      digitalWrite(LED,LOW);            //...and the LED will be turned off
 }
   previousDist = currentDist;         // The current distance is saved as the
previous distance ahead of the next reading
}
```

TRITA-ITM-EX 2021-46