EXAMENSARBETE INOM TEKNIK,
GRUNDNIVÅ, 15 HP
STOCKHOLM, SVERIGE 2019

ROBOTIC BALL RETRIEVING

A METHOD TO DETECT, COLLECT AND RETRIEVE A BALL

ELLEN SEVERINSSON

JULIA MALACHOWSKA

KTH
SKOLAN FÖR INDUSTRIELL TEKNIK OCH MANAGEMENT

ROBOTIC BALL RETRIEVING

A METHOD TO DETECT, COLLECT AND RETRIEVE A BALL

ELLEN SEVERINSSON
JULIA MALACHOWSKA

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

TRITA-ITM-EX 2019:52

Abstract

The aim of the presented thesis is to examine the potential of an autonomous football handling machine. It has been noted that the number of machines with the purpose of autonomously handling footballs, tennis balls, golf balls etc. are increasing but that no such machine is yet well established on the market. The performance of a robot that can both find, collect and retrieve a ball to a fix position is being examined. An area where the machine is intended to fill a purpose is for example penalty or free kick practice within football, where a single player stands relatively far away from goal and the ball does not necessarily bounce back to the player after scoring. Such a robot could make practice more efficient by relieving the player from the repetitive task of running back and forth to the goal, hence providing him/her with the flexibility of practicing alone.

The results of the study show that ultrasound is not an effective technique for detecting a football, because of many reasons. One of the reasons is that the ultrasonic sensor doesn't measure distance accurately enough when dealing with spherical surfaces. However, since the developed demonstrator is tested in a fixed setting, its success rate is high. With a few adjustment, the system as a whole shows high potential. The main recommendation for future work is to replace the ultrasonic sensor with a camera, and implementation of feedback control, for the steering of the robots movements.

Keywords: ball collector, Arduino, ultrasonic sensor, mechatronics, sports equipment

Referat

Bollkalle

Syftet med det presenterade arbetet är att undersöka potentialen hos en autonom maskin för hantering av fotbollar. Det har noterats att antalet maskiner med syfte att autonomt hantera fotbollar, tennisbollar, golfbollar och dylikt. har ökat, men att ingen sådan maskin ännu är väletablerad på marknaden. Prestandan hos en robot som både kan hitta, plocka upp och återbringa en boll till en bestämd plats undersöks. Ett område där roboten ämnas användas är till exempel vid träning av straffar eller frisparkar inom fotboll, där en ensam spelaren står relativt långt bort från målet och bollen inte nödvändigtvis studsar tillbaka till spelaren efter att bollen placerats i mål. En dylik robot skulle möjliggöra mer effektiv träning genom att bespara spelaren den repetativa uppgiften att springa fram och tillbaka till mål, och således delge denne med flexibiliteten av att kunna träna ensam.

Studiens resultat visar att ultraljud inte är en effektiv teknik för att detektera en boll, av flera anledningar. En av anledningarna är att ultraljudssensorn inte har tillräcklig noggrannhet för att mäta avstånd till sfäriska ytor. Eftersom den framtagna prototypen är testad i en riggad situation är antalet lyckade försök trots detta hög. Med ett antal justeringar uppvisar systemet som helhet god potential. De huvudsakliga rekommendationerna för framtida utveckling av produkten är att byta ut ultraljudssensorn mot en kamera, samt att införa feedback-reglering för styrning av robotens rörelser.

Nyckelord: bollkalle, Arduino, ultraljudssensor, mekatronik, sportutrustning

Acknowledgements

Many people have contributed in making this project possible and joyful. We would like to devote an extra gesture of gratitude towards:

Nihad Subasic, our supervisor, for directing the work in the right direction and constantly contributing with an encouraging spirit.

Sresht Iyer and Seshagopalan Thorapalli, the assistants in the lab, for your invaluable help and patience.

Ulf Gustavsson, head of the metal work shop, for being kind enough to make time to manufacture a crucial part for the demonstrator for us.

Staffan Qvarnström for being helpful at all times.

Ellen Severinsson, Julia Malachowska
Stockholm, May 2019

Contents

List of Figures

List of Abbreviations

$3D$	3-dimension
CAD	Computer-aided design
cm	Centimeters
DC	Direct current
IDE	Integrated Design Environment
KTH	KTH Royal Institute of Technology, Stockholm
PLA	Polylactic acid
PWM	Pulse Width Modulation
RPM	Revolutions per minute
USB	Universal Serial Bus
V	Volt

Chapter 1

Introduction

This chapter serves as an introduction and includes the background studies, purpose, scope, and methods used to achieve the results of this project, as well as a description of the scenario for which the developed robot is intended for.

1.1 Background

Sport psychology is a thoroughly researched area, where the psychology of a trained football player practicing penalties and free kicks have been proven to be particularly interesting. Having the perfect free kick at practice is not a warrant for success when the audience, and pressure, is present. A conclusion is that the quality of practice is just as important as the quantity, or amount of repetitions, to get it right when it matters [1].

How both quality and quantity could be granted when a player is practicing alone has been the main focus for this project, which aims at designing a robot that provides a football player with the flexibility of practicing alone. The robot should have the sole purpose to effectively retrieve the ball back to the player after the player has shot the ball at the goal. This would let the player to focus less on the task of running back and forth between the goal and the penalty spot to collect the ball. The player would have more time on preparing for one day scoring in a championship final if this robot was available.

When it comes to ball retrieving, and more specifically the tasks to find, collect, and return a ball to an appointed position, a significant number of robotic projects have already been made. Yet no solution has been superior enough to establish itself and become dominant on a market [2], thus making it an interesting focus for study.

Therefore, the focus of this thesis is the effectiveness of one particular proof-of-concept design for a robotic ball retriever.

1.2 Purpose

The presented project was carried out with the aim to design an easy-to-use robotic vehicle able to retrieve a ball to a player with high efficiency. With this aim in mind, the following research question was formulated:

- *With what success rate can the robot developed within the presented project perform while carrying out the following three functions when a ball is placed at different distances from the robot?*
  - *Detecting a scaled down football with ultrasound*
  - *Collecting a scaled down football with a lifting-mechanism*
  - *Returning a scaled down football to a player in a projectile movement*

1.3 Scope

The scope of the thesis is:

- The demonstrator is not a finished product of a real size robot. It is a prototype using a table tennis ball as a scaled down football for proof-of-concept.
- The demonstrator is programmed to react to one ball and one player only, and will always consider the object closest to it as the object it is supposed to collect. A goal net is not considered in its surroundings, neither is the possibility of a ball landing on the robot with a force that could destroy it.
- The demonstrator is designed for an indoor environment with no occurring weather, such as rain or wind, and with flat ground.

1.4 The scenario of this project

A football player is practicing penalties alone. The ball retrieving robot is located near the goal. The player kicks the ball towards the goal, see Figure 1.1, and after scoring, it will stay on the ground. It is fair to assume that the ball will in most cases either end up in between the player and the goal or in the goal, by other means, very close to where the robot is located.

The ball retrieving robot is able to detect where the ball is positioned and drives to it, see Figure 1.2, collects it and then begins searching for the player. When the player's position is detected, the ball is shot back to the player, see Figure 1.3. When the ball has returned, the player can once again put the ball on the penalty spot and again practice scoring.

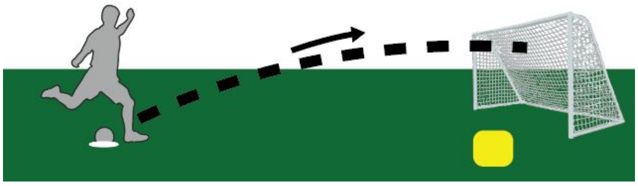


Figure 1.1: *A football player kicks a ball towards the goal. The yellow square represents the robot [3].*

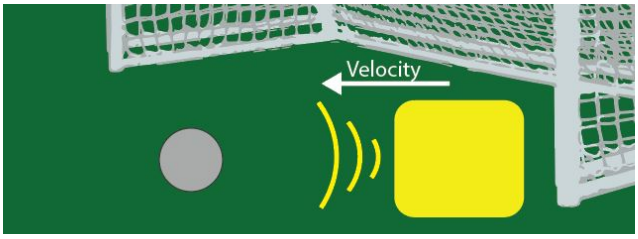


Figure 1.2: *The robot detects the ball and drives towards it. The yellow square represents the robot [3].*

Figure 1.3: *The robot collects the ball, detects the player and returns the ball back to the penalty spot. The yellow square represents the robot [3].*

1.5 General method

After the purpose of the project had been stated, see section 1.2 Purpose, the work continued in accordance with established design methodology [4]. At the outset, all required functions were formulated. The abilities to move, detect a ball or a player within a pre-set distance, collect a ball, and then launch the ball to an appointed position where the player was located, were fundamental.

After researching and brainstorming, it was decided that the robot should be built as a four-wheel vehicle with direct current (DC) motors connected to the rear wheels for moving. Eventually, after comparing ideas, the construction ended up as the following: For detecting objects, the vehicle was equipped with an ultrasonic sensor. For collecting objects, a stepper motor connected to a lifting scoop was attached, and for launching, a rotating wheel with a paddle that forces the ball into a projectile movement towards the player.

Prior to manufacturing every needed part for the demonstrator and assembling the robot, a computer aided design (CAD) model was created using the program *Solid Edge* [5]. This model is presented in Figure 1.4.

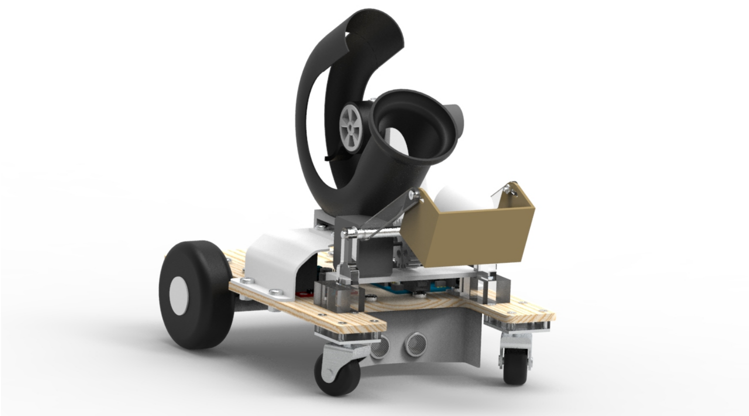


Figure 1.4: *The demonstrator as a CAD-model [3].*

Using the CAD model, a real demonstrator was built out of laser-cut plywood and 3D-printed polyactic acid (PLA) plastic. Finding the correct measurements was an iterative process from the first to the final version of the robot. When the final demonstrator was assembled, tests were carried out and results were noted. Lastly, improvements were documented for further work.

Chapter 2

Theory

This chapter covers the theoretical foundation of the project, as well as previous work.

2.1 Previous work

Work with a similar purpose as this project has been created before. Each one resulting in constructions with different designs and solutions. Needless to say, there are many ways to solve problems involved in making a ball retrieving robot. Some of the previously developed solutions are presented in this section. For a list of the previous work referred to, see Appendix B.

2.1.1 Methods for detecting an object

Detection of an external object can be solved by different technologies. Two common components used for such function are the vision sensor and the distance sensor.

In a previous made project, containing a golf ball collecting robot, a camera was used with the results that the used camera, called Pixi, worked only in light and clean surroundings and had sometimes a hard time detecting white golf balls. The used camera was able to detect objects only after colour and not shape [6].

An earlier robotic vehicle used an ultrasonic sensor to measure distance, avoid collisions and regulate the velocity during movement [7].

2.1.2 Methods for collecting an object

Methods for collecting objects are many. While looking at earlier established ball collecting robots, most use one of two technologies: a rotary blade or some sort of lifting arm, both illustrated in Figure 2.1.

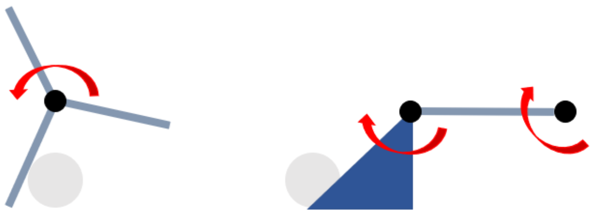


Figure 2.1: *Two examples of collecting mechanisms used by existing ball collecting robots: a rotary blade and a lifting arm or scoop [3].*

In a project made in the year of 2012 at Cornell University, a rotary blade is used for picking up tennis balls. The conclusion is made that it is an effective technology used for collecting balls [2].

In a project made in the year of 2015 at BIRLA Institute of Technology and Science, an arm was used to pick up balls. The project concluded that a common error that could occur with and arm was that the arm would knock off the ball whenever it missed it, which increased the time taken for collecting the ball. When the collecting was successful, the arm managed to put the ball on the robot smoothly [8].

2.1.3 Methods for returning a ball

Using a fly wheel to launch a ball is commonly used in automatic ball launcher machines, such as pitching machines used for baseball practice, or automatic ball launchers for dogs [9]. The technology of launching balls using one or two rotating wheels was patented already in the early 20th century [10].

Assuming that the angle between the ground and the fly wheel is constant, the distance at which the ball can be shot can still be regulated by varying the rotating speed with which the wheels are rotating.

2.2 Components

This chapter contains brief descriptions of the key components used in the project.

2.2.1 Microcontroller

A microcontroller is a small computer used for connecting other components such as a sensor or a motor through programmable input and output pins. In other words, a program with a set of instructions can be uploaded to the memory which enables control over the electrical signals received by or sent to other connected devices. The pins can read both analog and digital signals, and can output digital signals, both constant electrical waves and square waves known as pulse width modulation

(PWM) [11]. A well known example of a type of microcontroller is the Arduino UNO, shown in Figure 2.2.

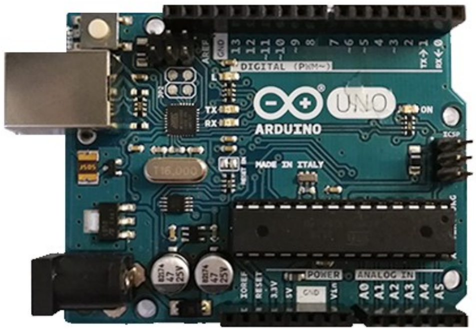


Figure 2.2: *The Arduino UNO used in this project [3].*

The hardware of an Arduino UNO includes a processor, a memory, several input and output pins, a universal serial bus (USB) plug, a plug for external power supply and a reset button. The software, code written in the programming languages *C* and *C++*, is uploaded via the USB plug. The software is developed in a specific integrated design environment (IDE), The Arduino IDE, and is the set of instructions used to control whatever the microcontroller is being used for [12].

2.2.2 Ultrasonic sensor

Ultrasonic sensors are often used for distance measuring [13]. An ultrasonic sensor sends out an ultrasonic wave via its transmitter, and registers the time it takes until the wave has been reflected back to its receiver, see Figure 2.3. If the sensor is used to measure distances in air, the speed with which the wave travels is known and will be equal to the speed of sound, a constant. An object can be located on a straight line in front of the sensor or at an angle for the signal to be received [14].

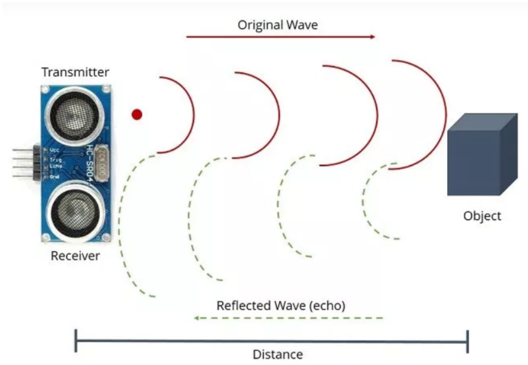


Figure 2.3: *The working principle of an ultrasonic sensor [15].*

The texture of the object does not influence the time measuring, as long as it does not absorb sound waves [16]. Because of the nature of wave reflection, the best possible accuracy is obtained when an ultra wave is reflected onto large, flat surfaces rather than small, round ones. Since the only parameter the sensor operates with is time, it does not output a lot of data for the user to process [17].

Since the time it takes for a transmitted wave to reflect back to the receiver is registered by the sensor, and the velocity of the wave is known as the speed of sound in air, the distance between the sensor and an obstacle can be calculated with Equation 2.1.

$$distance = \frac{time \cdot velocity}{2} \tag{2.1}$$

2.2.3 Motor control

Controlling an electrical motor can be done in terms of speed and direction of rotation. An H-bridge is a component that can be used to change direction of the rotation of an electrical motor by switching the polarity applied across the motor terminals. The working principle of the component is shown in Figure 2.4. When the circuit is connected to switch 1 and 2 the motor rotates in one direction and by changing the circuit to switch 3 and 4, the motor rotates in the opposite direction [18].

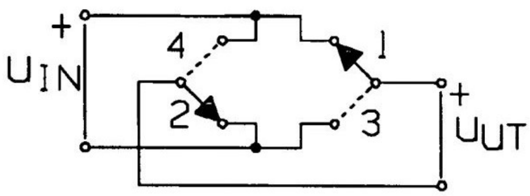


Figure 2.4: *The basic circuit structure of an H-bridge [18].*

The rotational speed of a DC motor is directly proportional to the applied voltage across its terminals. By switching transistors in an H-bridge between ON and OFF states with a certain frequency, the voltage applied to the motor terminals will be in the form of a square wave called PWM signal. In this case, the voltage that is applied across the motor terminals over a period of time is then the average voltage over that cycle. In other words, the speed can be controlled by changing the voltage applied across the motor with PWM signals at different duty cycles. 100 % duty cycle means full speed. This function can be implemented with H-bridges with a built in transistor using PWM signals from a microcontroller [19].

2.2.4 Stepper motor

Stepper motors are often used in precision work. They have excellent position control and hence can be used for precise control application. Additionally, stepper motors have very good holding torque which makes them an ideal choice for robotic applications. Stepper motors are also considered to have a long life time in comparison to normal DC or servo motors. The motor runs on direct current and is an electrical motor. The motor works due to current flowing through its stator coils creating a magnetic field that aligns a pole pair of the rotor with the energized stator coil creating a movement in steps.

Many different types of stepper motors exist and two popular ways to classify them are bipolar and unipolar stepper motors. A unipolar motor can shift rotational direction without any external component by changing polarization in different coils inside the motor. A bipolar motor on the other hand needs a motor control, such as an H-bridge, to be able to change the rotational direction. However, the advantage of a bipolar motor is that it produces more torque than a unipolar one of the same size [20].

Chapter 3

Demonstrator

This chapter contains all the information about the built demonstrator and its hardware, software and electronics. The results achieved by testing the demonstrator is also presented.

3.1 Problem formulation

A demonstrator was built to enable testing and answer the stated research question. in order to succeed testing, the demonstrator had to fulfill the following steps:

1. Detect a scaled down football within a restricted area
2. Collect found ball
3. Detect the player within a restricted area
4. Retrieve the ball to the player

3.2 Find a scaled down football

The ability to find a ball, defined as being able to locate the ball and move towards it, was fundamental for the robot in order to work as an autonomous ball collector.

The ability of interaction with the surroundings was enabled by an ultrasonic sensor. An ultrasonic sensor can, without a lot of data processing, be used to find the distance between the robot and an object, such as a ball or a player. The ability of moving towards a ball was implemented by giving the robot wheels. The front wheels were set to be cart wheels since those could rotate freely. Two DC motors and two motor drivers, one component respectively, were connected to each back wheel for steering. The motor drivers enabled control of the current through the DC motors, which let the robot to switch state between breaking, driving, forward and backwards, and turning.

The DC motors were chosen as model *327305* from *maxon motor*, see Figure 3.1. This motor is able to rotate with various revolutions per minute (RPM) depending on fed current which makes it a good choice for a robotic car that needs to be able to drive with different speed. The maximum allowed voltage is 12 volts (V). The used motor drivers were model L997ND from *ST*, see Figure 3.2. The data sheets for the used motor and motor driver can be found in Appendix A.

Figure 3.1: *The used DC motor for driving [3]*

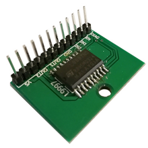


Figure 3.2: *The used motor control for the DC motors [3]*

The used ultrasonic sensor was of the model *HC-SR04* from ElecFreaks, see Figure 3.3 for a picture and Appendix A for details. This ultrasonic sensor is one of the most affordable distance sensors and is able to determine the distance to an object that is 2 to 400 centimeters in its path. The limited scope is 15 degrees. The robot had to be able to determine the distance, not what kind of object it was. Since the only central objects within the scope, see section 1.3 Scope, is a ball and a player, the robot will know by default that the first detected object is a ball and the second a player. Therefore, because of to the low price, low data usage and wide finding range, is the chosen sensor considered to meet the requirements for this project.

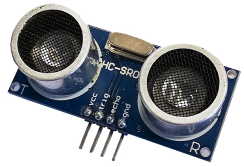


Figure 3.3: *The ultrasonic sensor used for determine distance to objects [3]*

The sensor was placed at a fixed position on the vehicle on the same height as the radius of the ball, so that the waves could be reflected at the thickest part of the ball. If the robot moved, the vision scope of the sensor changed as well. It is obvious that an object farther away from the ultrasonic sensor is at an higher risk to be missed by the sound waves if the robot is rotating too fast. Even having that the ultrasonic sensor sending out ultra waves as often as possible while rotating as slow as possible, the sensor might miss the object. Observe the process of searching for a ball by rotating in Figure 3.4.

Figure 3.4: *The angle θ depends on how fast the robot rotates between every sent ultrasound wave.*

3.3 Collect a scaled down football

In order for the robot to retrieve a ball to a player it must first gather the ball. This function was provided by an arm and a scoop. For this application a stepper motor was considered to be a good choice of motor since the movement is repetitive and must be precisely controlled. Also, the scoop needs to be able to be held at a static position. The chosen stepper motor was from *Tamagawa* model *TS3214N61*, see Figure 3.5.

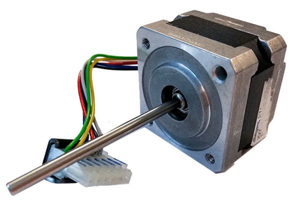


Figure 3.5: *Used stepper motor [3]*

Since the chosen stepper motor was used as a bipolar motor, an H-bridge had to be connected to guarantee being able to lift the scoop up and down. The used motor driver is the *L298N* dual H-bridge model from *Quinqi*, see Figure 3.6.

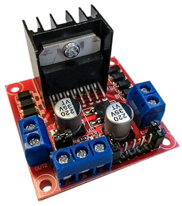


Figure 3.6: *The motor driver used for the stepper motor had a built in H-bridge [3]*

The decision to use a scoop instead of a rotary blade as the collecting mechanism, was made early in the designing process. Collecting a ball with rotary blades is easy to implement if the robot should only gather a ball in a basket. Since the aim in this project was to transfer the ball to another mechanism for launching, a scoop was considered to be a better choice. Nonetheless could the ball be lifted to a high level in a controlled manner but also be kept in the scoop without being lost in the process.

3.4 Retrieve the ball to the player

The mechanism enabling retrieving the ball to the player was decided to be a rotating wheel with a paddle, see the yellow component in Figure 3.7. This would work as a human throwing a ball since the paddle would force the ball to move inside a tube and then into a projectile movement when the tube ends. The mechanism allows the ball to be launched in the same direction as the vehicle is directed, which is the same direction as the ultrasonic sensor. Many different mechanisms could have been applied to this robot to retrieve the ball. However, since a table tennis ball is a light-weight ball and small, there was no need to apply high forces during launching. A wheel with a paddle was therefore considered enough for this project.

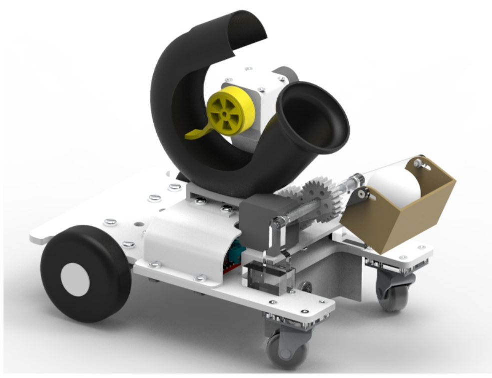


Figure 3.7: *The yellow component is a wheel with a paddle that launches the ball [3].*

The RPM of the motor shaft holding the paddle-wheel was adjusted according to the distance between the player and the ultrasonic sensor. The only signal allowing control of the motor was the voltage supplied via the Arduino and motor controller. Therefore, the relationship between the voltage and the distance on which the ball landed after being launched, was controlled. After observing where the ball would land, applying different voltages on the motor terminals, and plotting the collected data, it was clear that a linear relationship could be applied. Knowing that the distance would be zero if there was no voltage, a constant was calculated according to the linear relationship identified as Equation 3.1 shows.

$$constant = \frac{distance}{voltage} \tag{3.1}$$

3.5 Electronics

The control of every electric component went through the Arduino. However, since some components needed more power than the Arduino could supply, an external power source was used. The components that needed an external power supply of 12 V were the motor drivers and the stepper motor. All of the components were connected to a common ground. Figure 3.8 shows the connections made between the electrical components.

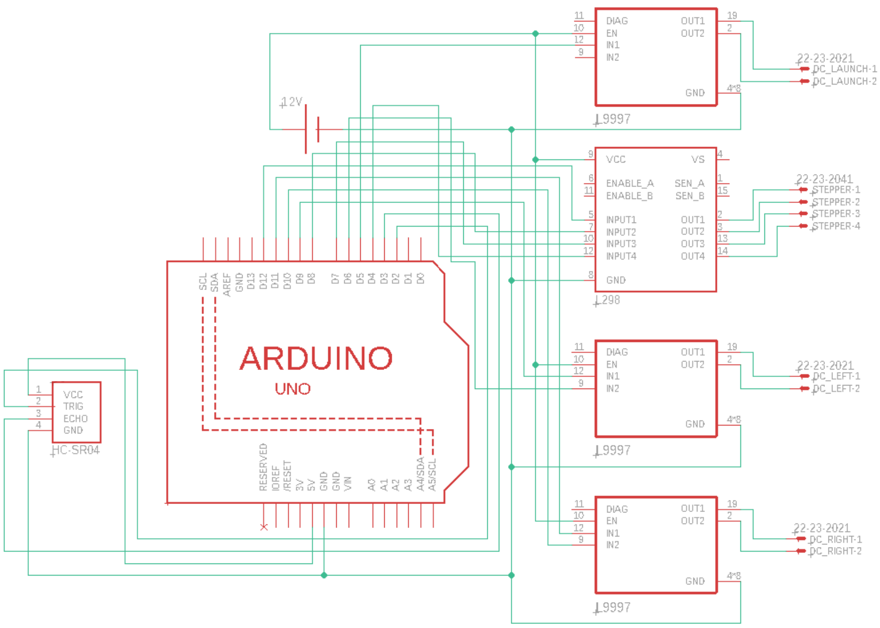


Figure 3.8: *The connections between the electrical components in principal, figure made in EAGLE [21].*

3.6 Software

The program controlling the movement of the robot repeatedly used the information from the distance sensor in order to decide what sequence of actions to execute. It had to consist of the following instructions:

1. Drive straight until the ball is reached
2. Collect the ball with the scoop
3. Rotate on the spot until the player is found
4. Return the ball by rotating a wheel with a paddle

The Figure 3.9 shows the logical flow of instructions. The full program is found in Appendix C.1.

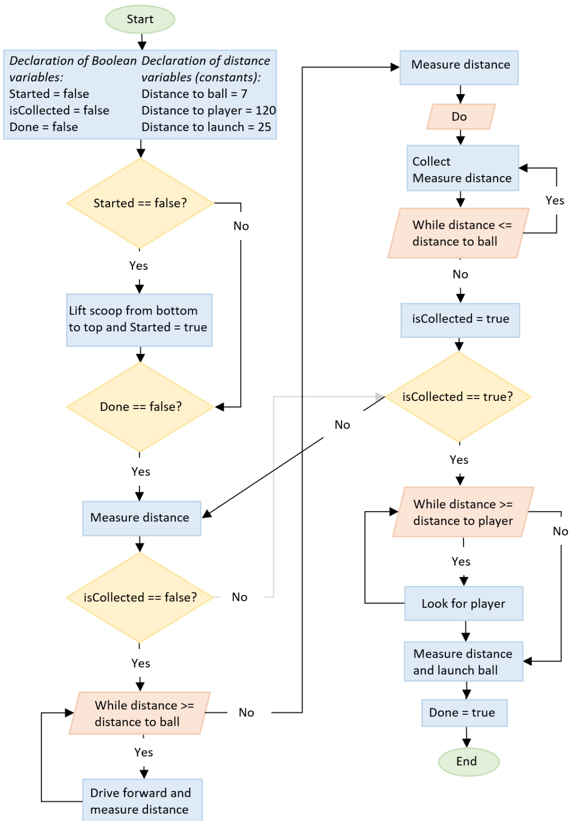


Figure 3.9: *Flowchart of the code used to run the demonstrator [22]*

The structure of the program consisted of one main loop, and within the main loop three different functions were called. The functions that were defined outside the loop were called *sensor*, *collect* and *launch*. The *sensor* function was called every time the distance was to be measured by the sensor. The functions *collect* and *launch* were for collecting and launching respectively. Some Boolean variables were initiated before the main loop, that switched value as the tasks had been executed, so that it would not enter the same function twice. After a sequence where the robot had detected the ball, collected the ball, and returned the ball to the player, it was programmed to not execute any more actions.

3.7 Experiments

This chapter covers everything about the testing of the demonstrator and how the results were evaluated.

3.7.1 The experimental setup

The experimental setup for testing the demonstrator's ability to find, collect and return a ball was built to match the theoretical scenario stated in section 1.4. The scope of the project stated in section 1.3 was also taken into consideration.

Testing data was gathered for each of the three functions independently, since every function in itself formed a self-sufficient subsystem regarding both hardware and software. Data over the ability to return the ball to the player could only be gathered if both tests for detecting and collecting were successful. With other words, testing always followed the three following steps:

$$\text{Detect ball} \rightarrow \text{Collect ball} \rightarrow \text{Return ball}$$

Figure 3.10 shows how every test started, the robot was positioned, with its wheel directed forward. A table tennis ball was positioned straight in front of the robot and the player, a blue cardboard box, stood still on a specific spot, angled towards the robot ball and the ball.

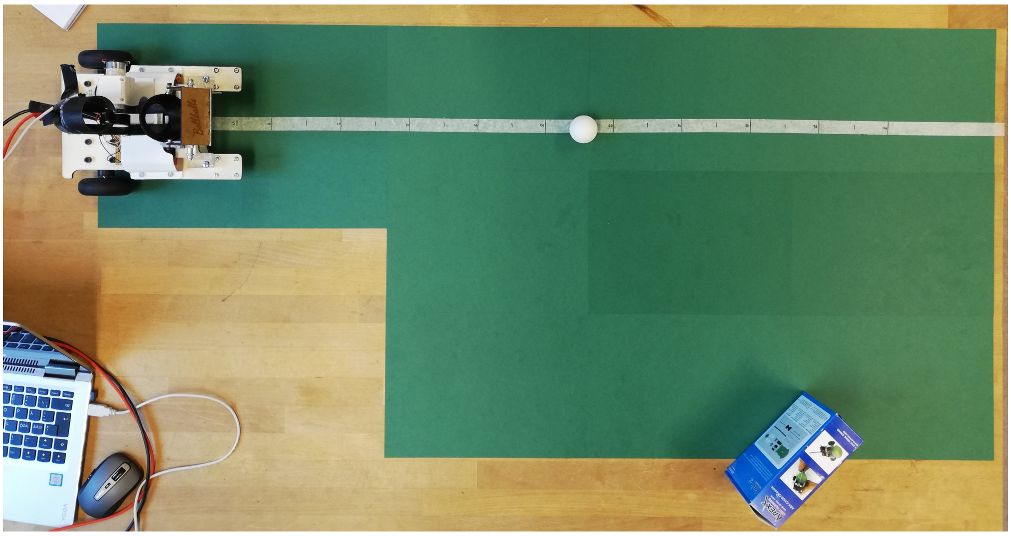


Figure 3.10: *The experimental setting and placement of ball and player, player symbolized by the blue box.*

Data over the success rate for detecting, collecting, returning, was collected over three different distances between the robot and the table tennis ball. Every distance: 20 cm, 45 cm and 60 cm, were tested 20 times each.

The ball was placed on a straight line in front of the robot, instead of an angle, as shown in the experimental setup. This is because the reliability of a single ultrasonic sensor detecting a small, round surface is relatively low. The ball was therefore positioned in front of the robot with the hope of a high finding rate.

The player on the other hand was located at an angle from the sensor since the player, symbolised by a cardboard box in the testing, had a large, flat surface. This means the sensor should have a relatively high detecting rate while detecting the player. The player was positioned at the same spot at all times. Since the ball was placed at three different positions, and since the robot collects the ball at those positions, different distances between the player and the robot during returning was tested nevertheless.

Detecting the ball was considered to be successful if the robot was able to start driving at the starting position and stop driving just in front of the ball without pushing the ball, or omitting it. The ball would then end up right in front of the sensor.

Every time the robot succeeded detecting the ball, data over the success rate for collecting could be gathered. The testing of collecting was considered successful if the robot was able to lift the ball from the floor and keep the ball in its scoop.

Every time the robot succeeded detecting and collecting the ball, data over the success rate for returning the ball could be gathered. The test of returning was considered successful if the ball being launched landed in front of the player, which meant the sensor had been able to detect the player and the launching motor rotated with a correct speed. It would be strongly unwanted for the ball to land behind the player in a real life situation.

3.7.2 Findings and evaluation

The results of the experiments is presented in Table 3.1. The percentile success rates presented are based on the frequency of success for each function, based on the definition of success in section 3.7. Distance refers to the distance between the sensor on the robot and the ball in the starting position.

Success rates			
Distance	Detecting	Collecting	Returning
30 cm	95%	100%	47%
45 cm	90%	94%	100%
60 cm	85%	94%	93%

Table 3.1: *Statistic results of the tests at different distances between robot and ball*

When the ball was placed 30 cm in front of the sensor, it properly stopped in front of the ball 19 of 20 cases. When it had detected the object, it succeeded collecting the ball every time, 19 out of 19 times. Returning succeeded in 9 out of those 19 times.

When the ball was placed 45 cm from the robot, it succeeded detecting the ball 18 of 20 times. Collecting was then carried out with a success 17 of those 18 times, and returning 17 of those 17 times, hence the 100% success rate.

When the ball was placed 60 cm from the robot, it succeeded detecting the ball 17 of 20 times. Collecting was then carried out with a success 16 of those 17 times, and returning 15 of those 16 times.

Since the tests of each function were independent of one another, the probability for the robot to succeed to go through all three functions in one sequence could be calculated as the product of each of the individual success frequencies [23], according to formula 3.2. The results of this calculation for each distance is presented in Table 3.2.

$$P_{succeed\,with\,all\,three functions} = P_{detecting} \cdot P_{collecting} \cdot P_{returning} \tag{3.2}$$

Probability of success	
Distance	Probability
30 cm distance	45%
45 cm distance	85%
60 cm distance	74%

Table 3.2: *The probabilities for the robot to succeed with all 3 tasks with the ball placed on different distances.*

As seen in Table 3.1, the most successful subsystem of the robot was the collecting mechanism. The lowest success rate was performed by the returning system. The observation is made that the highest probability of the robot succeeding with performing the full sequence of three functions is when the robot has the least distance to the player after collecting the ball.

Chapter 4

Discussion and conclusions

This chapter contains a discussion about the results that is followed by a conclusion.

4.1 Discussion

The statistical tests made on the demonstrator shows that high performance rates can be accomplished with the robot developed in the presented project. Several problems did arise and even though the demonstrator performed well in most cases, it sometimes failed. The discussion in this section is divided in the separate test results for clarity.

Detection

In order for detecting objects to work as intended it was crucial that the robot's movement allowed to be controlled to enable driving straight forward when programmed to drive straight forward. It was also crucial that the ultrasonic sensor gave the right information about what distance it was between the sensor and the object it was intended to detect. Both of these areas sometimes proved to be problematic.

There were two main observations made on why the robot sometimes had a hard time driving straight. Firstly, the DC motors were not reliable while driving in the sense that applied voltage made the two motors rotate with different RPM. To fix this problem, a constant was introduced in the software to give one motor less voltage than the other so both motors rotated with more alike RPM with the given voltage. It is probable that a better performance could be carried out if more iterations over the constant were made.

Secondly, the front wheels could rotate freely and were not controlled by the software. The cart wheels could therefore make the robot turn unreliably in the beginning of the tests if they were not put straight forward by the tester. Since the robot only could drive straight forward if the cart wheels were pointed in the wished direction, the human factor could have a big part as of why the robot sometimes had a hard time driving straight. Whenever the ball did not end up right in front of the sensor, the sensor had a hard time detecting the ball, which means that driving straight played a big role as to if the sensor found the ball or not.

The ability of determining the distance to objects, could also have effect on the results. It was noticed that when the ball was very close to the sensor, it could not provide this information and instead said that it was very far. This is probably due to the fact that the chosen sensor is only able to find objects further away than 2 cm. With this fact in mind, the robot sometimes drove towards the ball and then begun to think the ball was further away that it actually was.

It was also noticed that the sensor sometimes had a hard time detecting the ball if it was further away, even if it was in the allowed view of the chosen sensor. This is probably because of the size and the shape of the ball. The ball is small and round and that is an object ultrasonic sensor sometimes have a hard time detecting. However, due to the experiment setup where the ball was always in front of the sensor, the sensor never had to look for the ball specifically. The ball could be positioned far away but the sensor had only a large role in the detecting when the ball was detected closely to the robot at which it should stop driving.

Collecting

The collecting mechanism had the most satisfactory performance and worked almost every time. One factor could be that it was controlled mainly by a stepper motor, which is known to be precise. Another reason for the results could be that the collecting function was developed early in the project and because of this a lot of time could be used to work on iterating both software and hardware related to the collecting mechanism.

There could be many factors that play a role in the success for collecting. One factor could be that the table tennis ball was grinded with sand paper to make the friction between the ball and the scoop greater. Another factor could be that the scoop collected the ball by pushing the ball towards the robot and therefore prohibited the ball from rolling away.

Retrieving

The returning mechanism failed most often out of the three functions and the probable reason could be that it was dependent on both DC motors steering the robot and the ultrasonic sensor. Different distances gave very different results, which could mean the distance between the robot and the player plays a big role when it comes to accuracy. The robot failed returning the ball every second time during the test when the player was farthest away.

Since the robot was rotating while searching for the player, the value of the delay after detecting the player could potentially have a major impact on the success rate as well. The delay value decided where the robot would place itself in front of the player after detection. A greater delay would mean the robot would rotate a bit longer and the ball could perhaps hit the player more often rather than miss and shoot to the side of the player.

It should be said that the launching mechanism, the wheel with a paddle, worked well during every try, which means that the low success rate for the retrieving mechanism often depended on how well the ultrasonic sensor detected the player and how fast the robot stopped rotating. In order to reflect the results over the retrieving mechanism more justly the test of retrieving could have been divided in two parts: 1) detecting the player, and 2) launching the ball.

Overall discussion

There seemed to be a strong correlation between how often the robot succeeded with its task and at what distance the robot had to locate the player. The results showed clearly that the closer the player was to the robot, the better the overall performance was. The highest overall success rate was performed at 45 cm, which is were the player was closest to the robot. The second highest success rate was observed when the ball was located at 60 cm from the robot, were the player would be located second closest to the robot for shooting. It had already been stated that the sensor worked better the closer to it the objects were, which serves as a intuitive reason for explaining the achieved result.

The huge simplification of the scenario is probably taking too few outcomes into consideration for the robot to be considered as useful even in a scaled up version. For example, the situation where the ball bounces further away than the restricted area close to goal, or where there are more people active on the field than the one practicing shots at goal, would need to be prepared for and would require the code to be considerably expanded, since at the moment, the code is very precisely adapted to a very restricted scenario. Further studies would need to be done in order to prove whether or not an automated solution is effective when it comes to penalty practice.

Regarding the developed ball retrieving robot's competitive advantages comparing it to other robots with the same functions, it's main strength is that the picking-up mechanism, aka the scope and it's arm, uses only one motor, thus minimizing the number of motors needed to run this particular sub-system. The foundation in developing such a simple mechanism was iterations and testing.

4.2 Conclusions

The answer to the research question stated in section Purpose 1.2 is presented in the following:

- *With what success rate can the robot developed within the presented project perform while carrying out three functions: detecting, collecting and returning a scaled down football, at the following distances between the ball and the robot?*
  - At 30 cm distance: 45 %
  - At 45 cm distance: 85 %
  - At 60 cm distance: 74 %

The different subsystems of the robot plays different roles in the results, as seen in Table 3.1. The function *detecting* had the highest success rate at 30 cm, second highest at 45 cm and least at 60 cm. The reasons for this could be that the robot sometimes had a hard time driving straight when the ball was located far away and that the ultrasonic sensor could not perform at a full potential with a relatively small, spherical object, causing the ultra waves to bounce in different directions.

The function *collecting* had the highest success rate at 30 cm, second highest at 45 cm and 60 cm. Overall was the rate the same, differentiating from 100% or 94% success rate. The reason for this could be that the collecting function had nothing to do with the distance between the ball and the robot. If the robot had succeeded detecting the ball, which meant the ball would end up very close to the sensor, the collecting function would succeed.

The function *returning* had the highest success rate at 45 cm, second highest at 60 cm and least at 30 cm. At 30 cm, the returning mechanism failed every second time and this could be explained by the fact that the player was furthest away from the robot when it had collected a ball 30 cm from its starting position.

In conclusion, the developed robot has the ability to work as a ball retrieving robot within the scope of this project. Future work is however required in order for the robot to work with a higher success rate.

Chapter 5

Recommendations and future work

Considering how straight-forward the mechanisms developed and used were, both from a working principle point of view and a manufacturing point of view, the success rate can be considered satisfactory. However, in order to make the robot work conveniently in other circumstances than the experimental setting many changes and improvements would need to be done, not to mention the scaling up required to be adapted to a football.

A camera is strongly recommended to be added so that the robot can detect objects better and for distinguishing objects from each other. For the motors driving the vehicle, it would be strongly suggested implement feedback in order to have greater control of the steering, not allowing deviation from the desired path because of voltages of output signals or motors not being precise. The stepper motor which is currently used to lift the scoop with the ball is working at the edge of its potential as it is, and would hence need to be exchanged to be able to lift a heavier load. The 3D-printed gears which worked well for a small load would probably not work as well for a heavier load, why metal gears are recommended. Concerning the launching mechanism, it is currently neglecting the forces since the ping pong ball is so light weight. In the scaled up version, a launching mechanism handling high forces, for example a fly wheel, could with advantage be implemented.

Bibliography

[1] James Barraclough. *Practising penalties – Necessary or pointless?* (Online) https://believeperform.com/performance/practising-penalties-necessary-or-pointless/, Accessed 2019-03-20, 2011.

[2] Peter Kung Charles Cheung. *The Autonomous Tennis Ball Picker*. Cornell University, Dept. of Electrical and Computer Enginnering, NY, 2012.

[3] Julia Malachowska Ellen Severinsson. Own photography/figure, 2019.

[4] David G. Ullman. *The Mechanical Design Process*. McGraw-Hill Education, 2010.

[5] Solid edge st10. Siemens, 2018.

[6] Fredrika Kringberg Marcus Olssen. *POKI, Robotic ball collector on golf ranges*. KTH Royal Institute of Technology, Dept. of Machine Design, Stockholm, 2016.

[7] Simon Tönnes Joakim Storfeldt. *Effects of Using Multiple Sensors to Guide an Atonomous Vehicle*. KTH Royal Institute of Technology, Dept. of Machine Design, Stockholm, 2017.

[8] Parakh M. Gupta. *Magneto Ball collector and shooter*. Birla Institute of Technology and Science, Hyderabad, India, (in dialogue with Parakh M. Gupta made in May of 2019), 2016.

[9] Dogviously.com. *Best Automatic Ball Launchers For Dogs*. (Online) https://www.dogviously.com/best-automatic-ball-launchers/, Accessed: 2019-04-23, 2019.

[10] Bartley N. Marty. *Automatic base ball pitching machine*. United States Patent Office, 1916.

[11] Hans-Petter Halvorsen. *Introduction to Arduino*. Telemark Univerity College, Faculty of Technology, Dept. of Electrical Engineering, Information Technology and Cybernetics, Telemark, Norway, 2013.

[12] Yusuf Abdullahi Badamasi. *The working principle of an Arduino*. 11th International Conference on Electronics, Computer and Computation (ICECCO), Abuja, Nigeria, 2014.

[13] Tarek Mohammad. *Using Ultrasonic and Infrared Sensors for Distance Measurement.* World Academy of Science, Engineering and Technology, Hong Kong, China, 2009.

[14] William Hamlet and William Kusewich. *Using The Ultrasonic Sensor to Determine Location.* NYU Tandon School of Engineering, Dept. of Mechatronic Design, New York City, 2007.

[15] Andras Tantos. *H-brigdes - the basics.* (Online) http://www.modularcircuits.com/blog/articles/h-bridge-secrets/hbridges-the-basics/, Accessed: 2019-04-14, 2011.

[16] Keyence Corporation. *What is an ultrasonic sensor?* (Online) https://www.keyence.co.uk/ss/products/sensor/sensorbasics/ultrasonic/info/, Accessed: 2019-03-15, 2019.

[17] James Jeffery Trobaugh. *Winning Design! LEGO MINDSTORMS NXT Design Patterns for Fun and Competition.* Springer Science+Business Media, 2017.

[18] Hans Johansson. *Elektroteknik.* KTH Royal Institute of Technology, Dept. of Machine Design, Stockholm, Sweden, 2013.

[19] Electronics Tutorials. *DC Motors.* (Online) https://www.electronics-tutorials.ws/io/io$_7.html, Accessed: 2019-05-08, 2017.$

[20] Reston Condit. *Stepping motors fundamentals.* Microchip Technology Inc., University of Iowa, Iowa, 2004.

[21] Eagle software, ver. 9.3.0.

[22] Kjell Bäckman. *Structured programming with C++.* Dept. of Economy and Informatics, University West, Trollhättan, Sweden, 2012.

[23] prof. Tom Britton prof. Sven Erick Alm. *Stokastik, sannolikhetsteori och statistikteori med tillämpningar.* Liber AB, 2014.

Appendix A

Appendix A

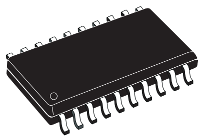
A.1 Electrical components datasheets

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



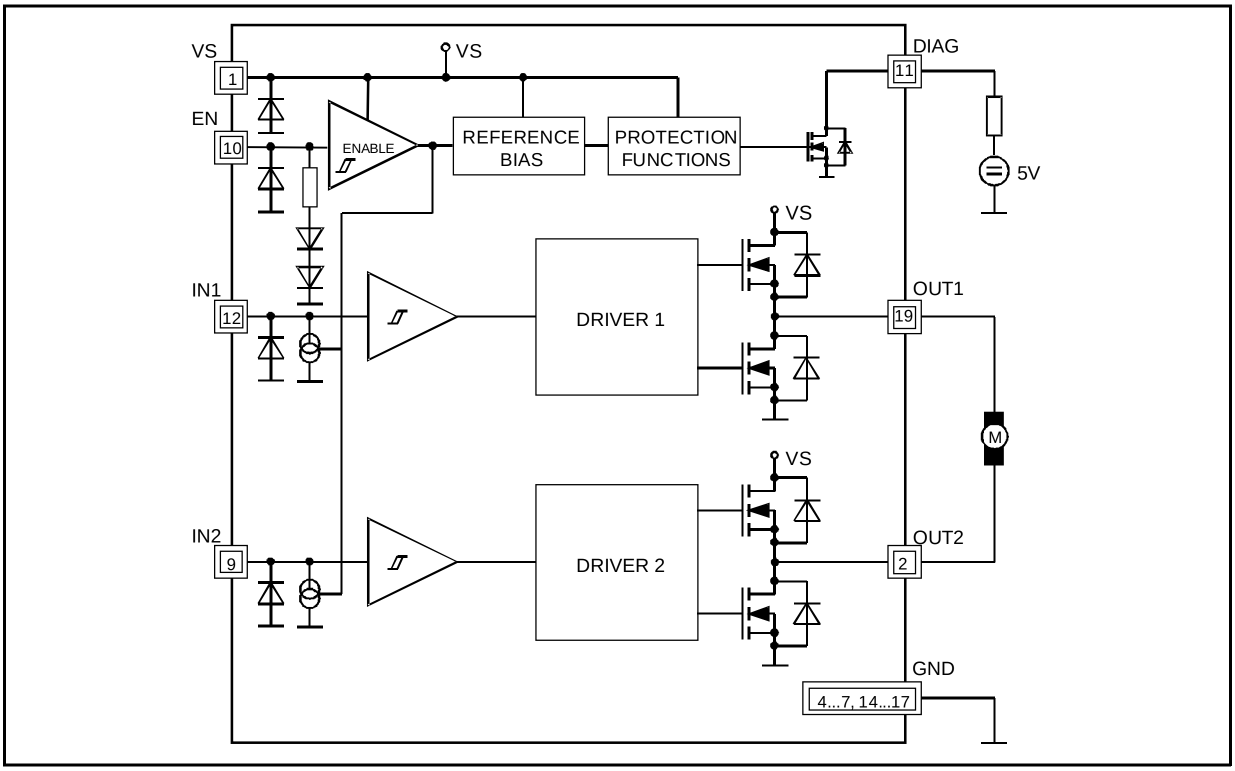
SO20 (12+4+4)

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.

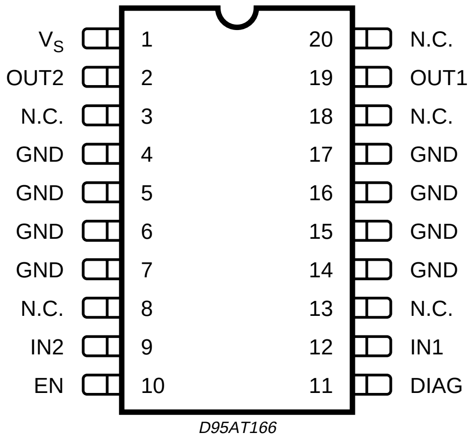
BLOCK DIAGRAM



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
$\lvert I_{OUT1,2}\rvert$	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.

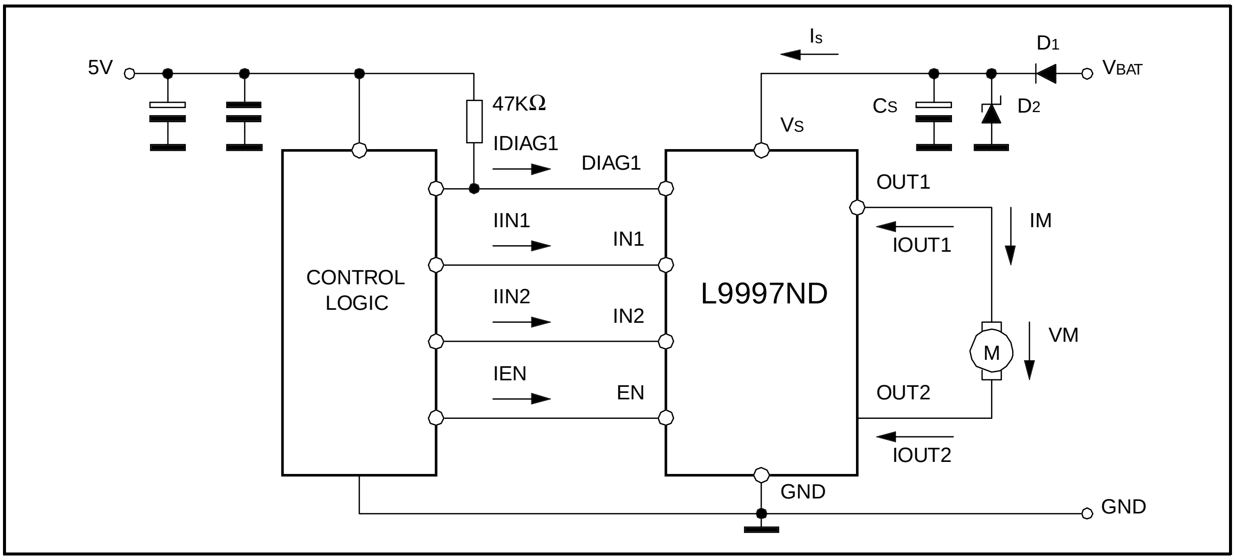


Figure 2. Timing Diagram.

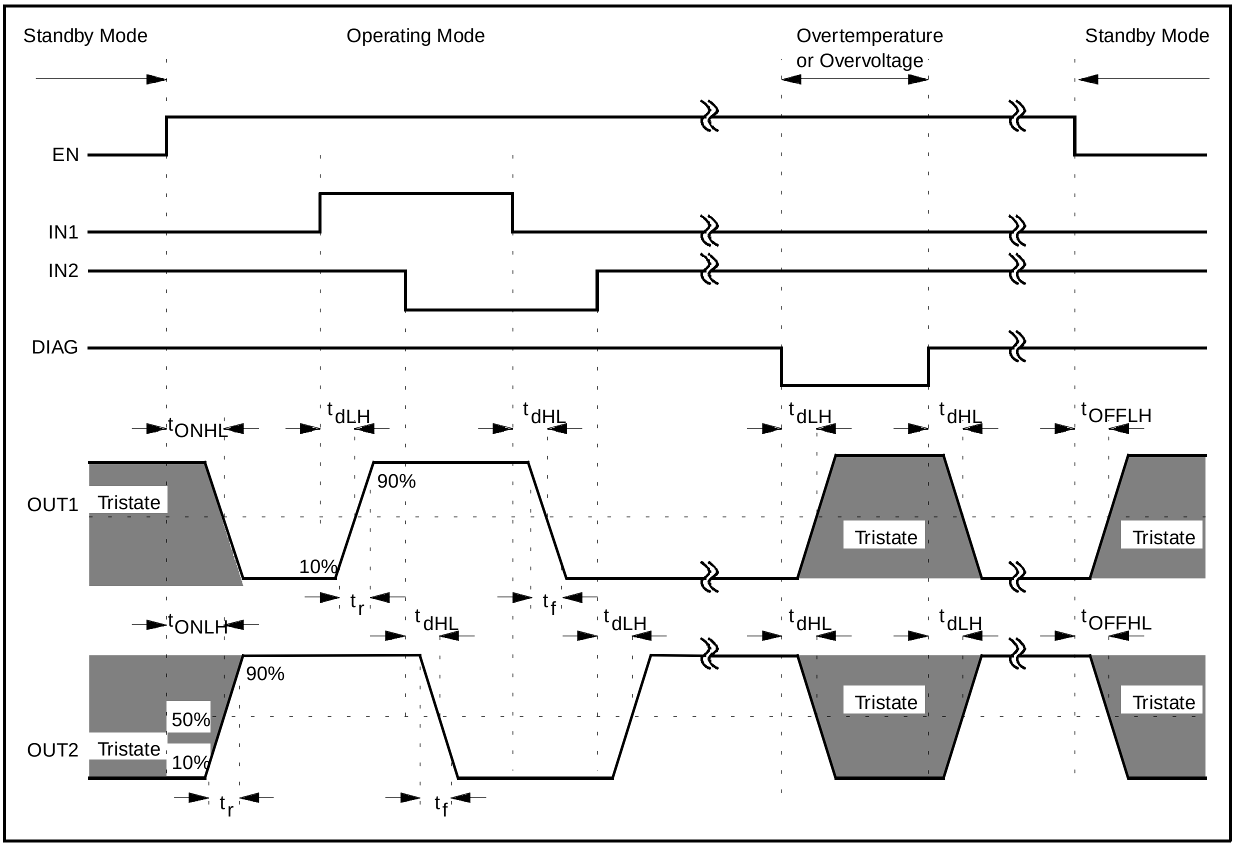


Figure 3. Typical R_{ON} - Characteristics of Source and Sink Stage

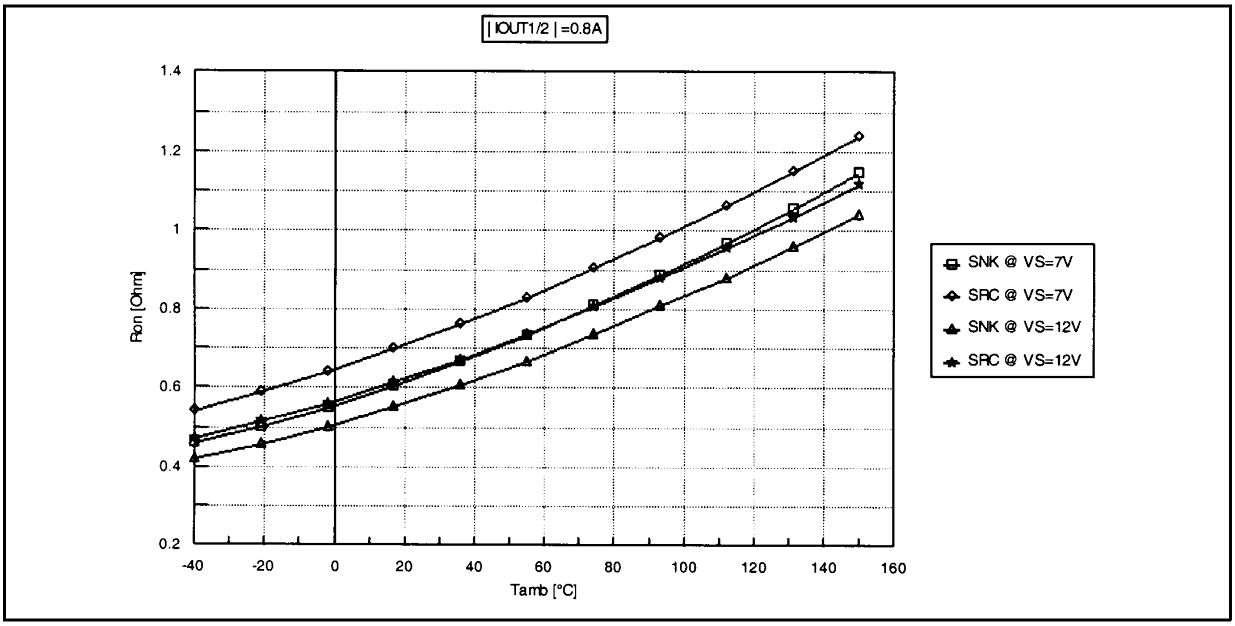


Figure 4. Quiescent current in standby mode versus supply voltage.

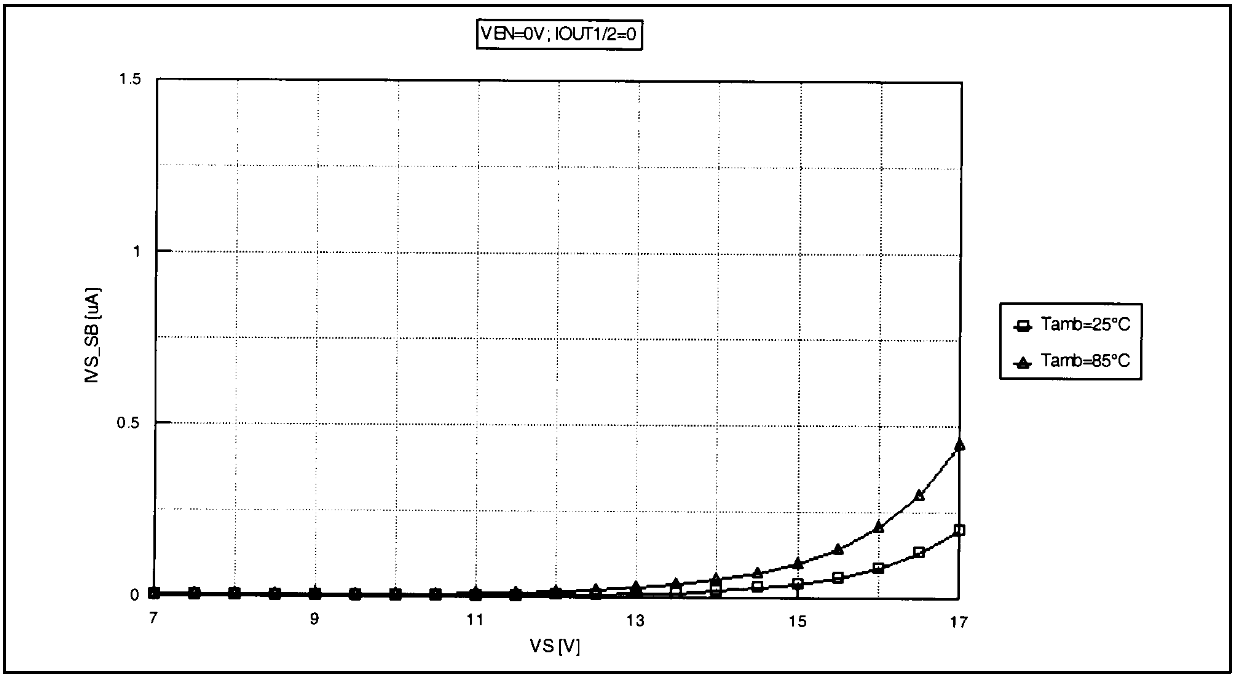


Figure 5. ON-Resistance versus supply voltage.

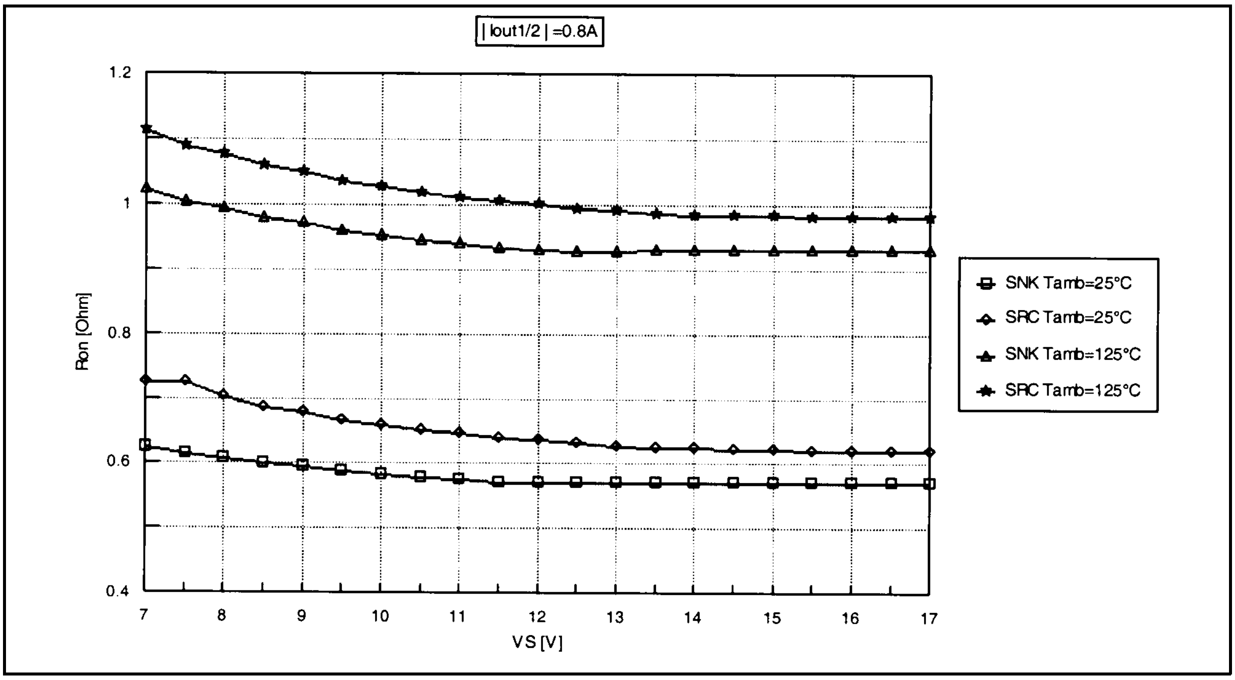


Figure 6. I_{OUT} versus V_{OUT} (pulsed measurement with T_{ON} = 500µs, T_{OFF} = 500ms).

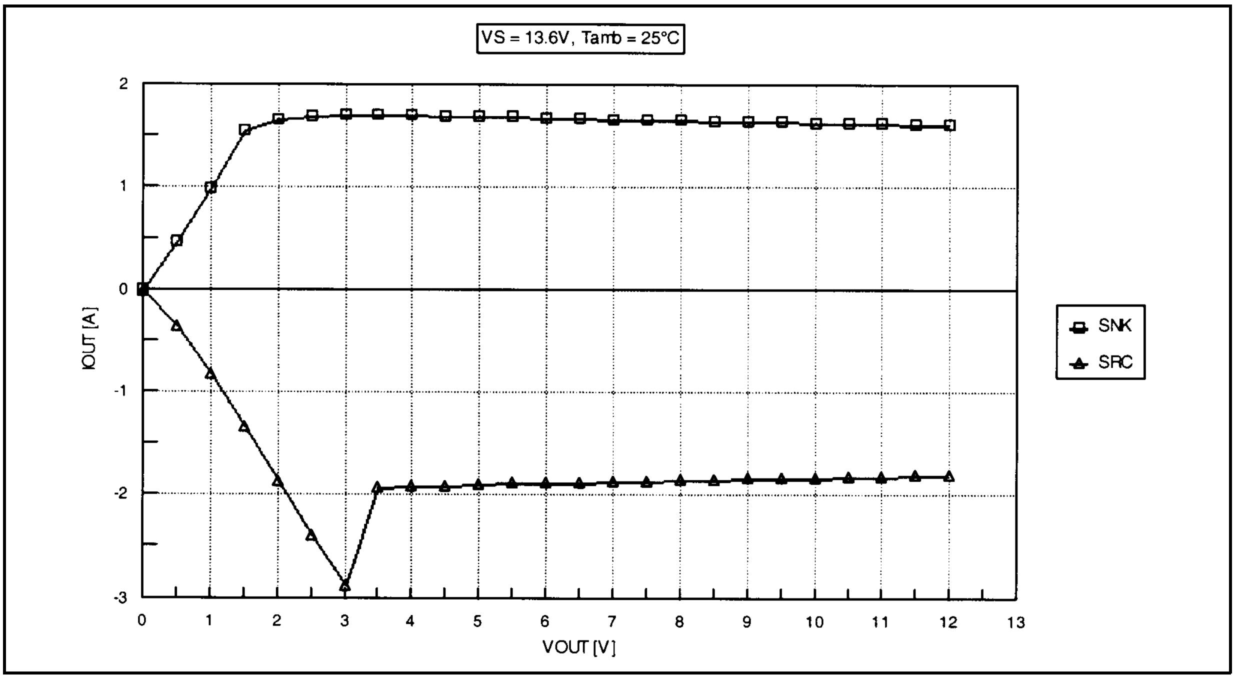
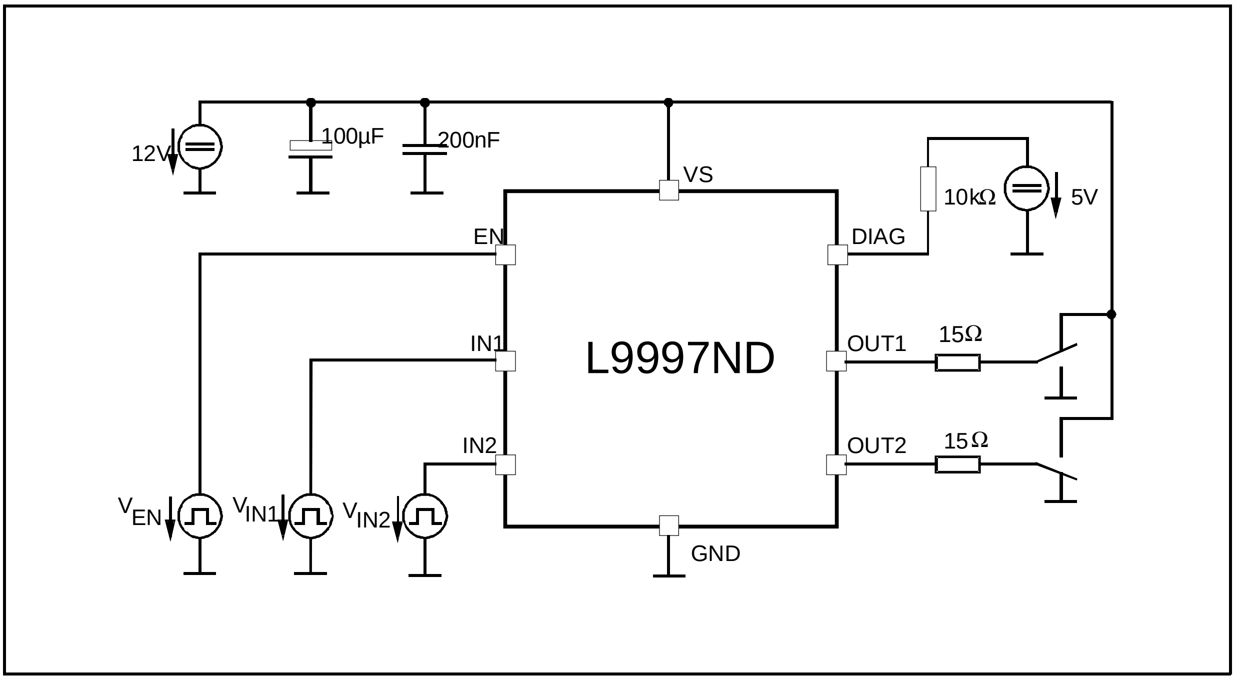
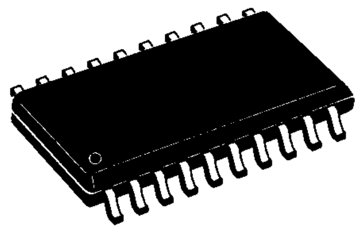


Figure 7. Test circuit.

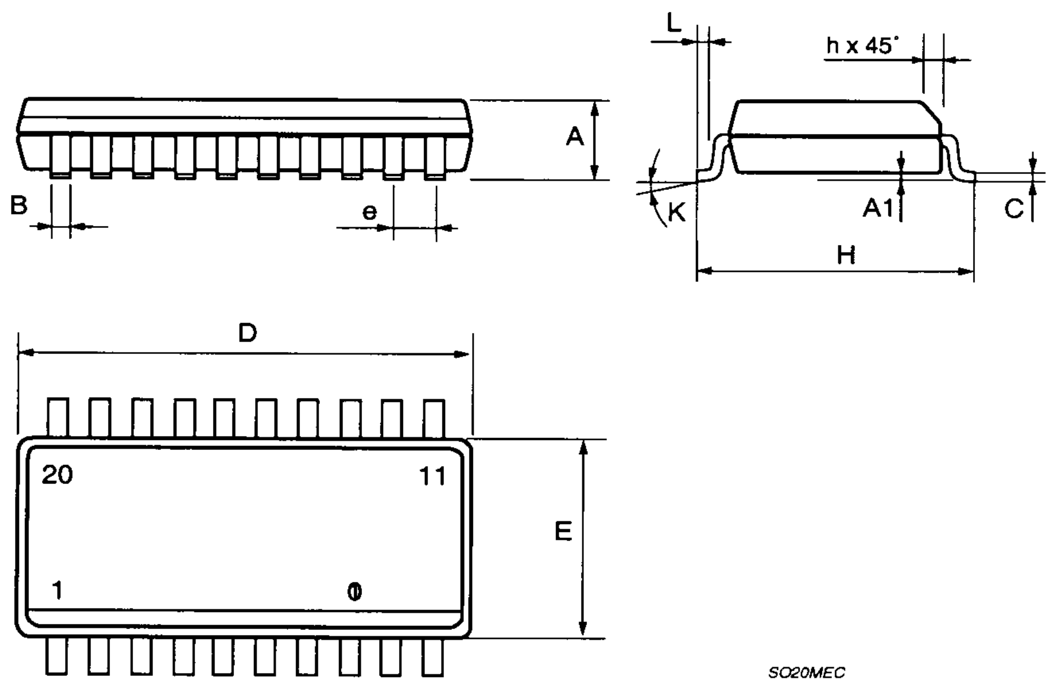


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A	2.35		2.65	0.093		0.104
A1	0.1		0.3	0.004		0.012
B	0.33		0.51	0.013		0.020
C	0.23		0.32	0.009		0.013
D	12.6		13	0.496		0.512
E	7.4		7.6	0.291		0.299
e		1.27			0.050	
H	10		10.65	0.394		0.419
h	0.25		0.75	0.010		0.030
L	0.4		1.27	0.016		0.050
K	0° (min.)8°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

OUTLINE AND MECHANICAL DATA



SO20



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

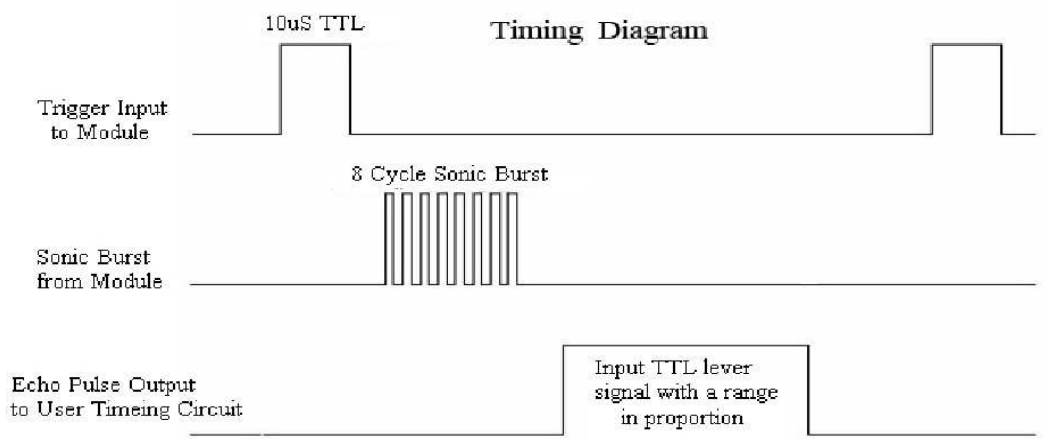
Electric Parameter

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:

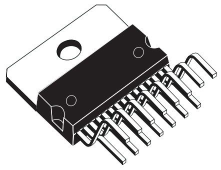
- The module is not suggested to connect directly to electric, if connected electric, the GND terminal should be connected the module first, otherwise, it will affect the normal work of the module.
- When tested objects, the range of area is not less than 0.5 square meters and the plane requests as smooth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

www.Elecfreaks.com

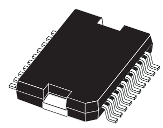
L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)



Multiwatt15



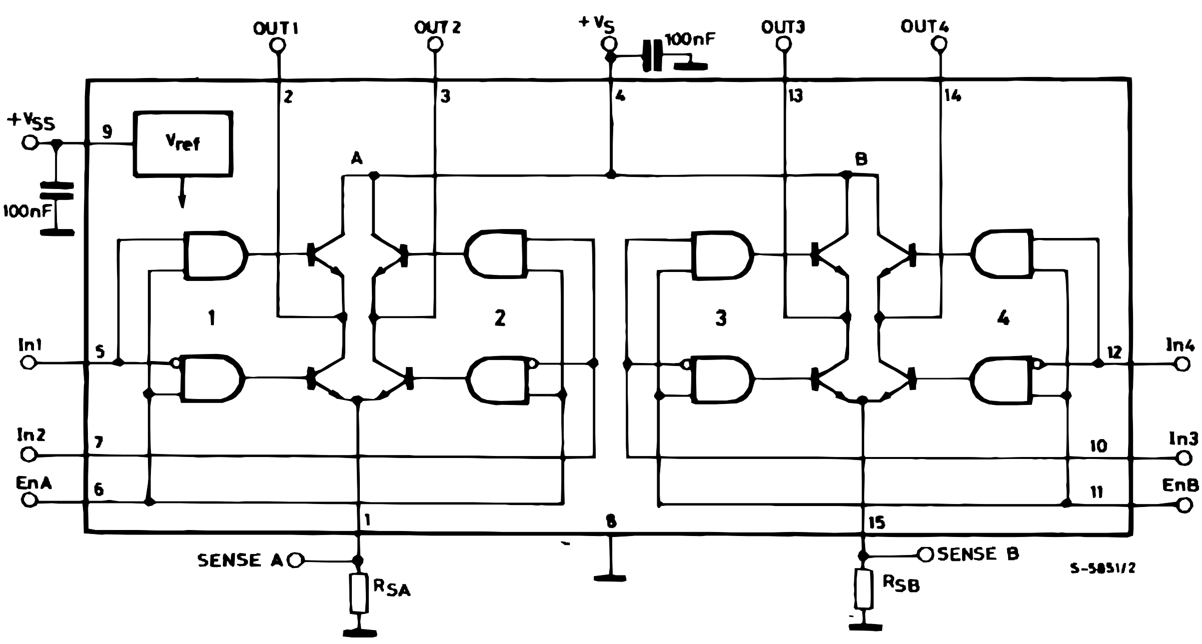
PowerSO20

ORDERING NUMBERS : L298N (Multiwatt Vert.)
L298HN (Multiwatt Horiz.)
L298P (PowerSO20)

DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

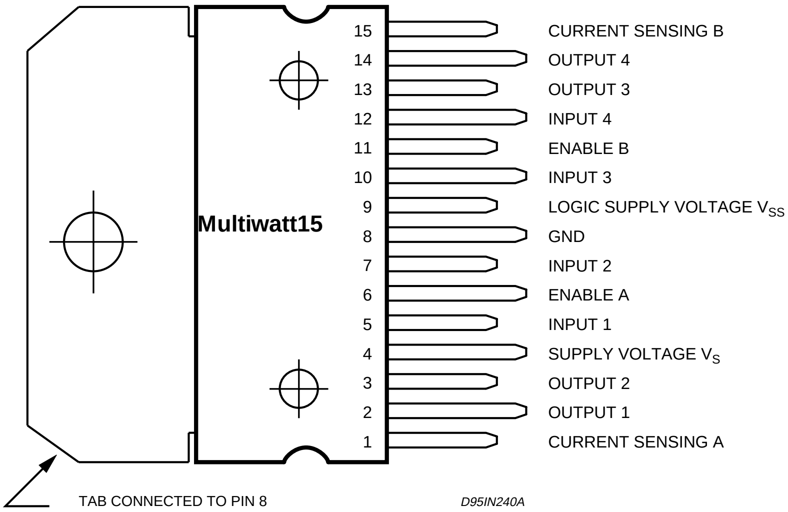
BLOCK DIAGRAM

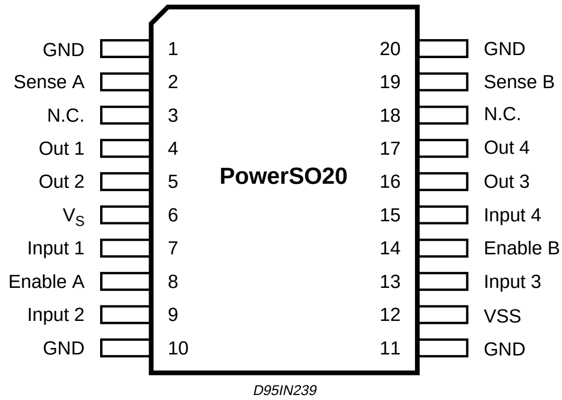


ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I,V_{en}	Input and Enable Voltage	–0.3 to 7	V
I_O	Peak Output Current (each Channel) – Non Repetitive (t = 100μs) –Repetitive (80% on –20% off; t_{on} = 10ms) –DC Operation	 3 2.5 2	 A A A
V_{sens}	Sensing Voltage	–1 to 2.3	V
P_{tot}	Total Power Dissipation (T_{case} = 75°C)	25	W
T_{op}	Junction Operating Temperature	–25 to 130	°C
T_{stg}, T_j	Storage and Junction Temperature	–40 to 150	°C

PIN CONNECTIONS (top view)





THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th\ j\text{-}case}$	Thermal Resistance Junction-case	Max.	–	3	°C/W
$R_{th\ j\text{-}amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	°C/W

(*) Mounted on aluminum substrate

PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V_S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_j = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_S	Supply Voltage (pin 4)	Operative Condition	V_{IH} +2.5		46	V
V_{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I_S	Quiescent Supply Current (pin 4)	V_{en} = H; I_L = 0 V_i = L V_i = H		13 50	22 70	mA mA
		V_{en} = L V_i = X			4	mA
I_{SS}	Quiescent Current from V_{SS} (pin 9)	V_{en} = H; I_L = 0 V_i = L V_i = H		24 7	36 12	mA mA
		V_{en} = L V_i = X			6	mA
V_{iL}	Input Low Voltage (pins 5, 7, 10, 12)		–0.3		1.5	V
V_{iH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		VSS	V
I_{iL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V_i = L			–10	µA
I_{iH}	High Voltage Input Current (pins 5, 7, 10, 12)	Vi = H ≤ V_{SS} –0.6V		30	100	µA
V_{en} = L	Enable Low Voltage (pins 6, 11)		–0.3		1.5	V
V_{en} = H	Enable High Voltage (pins 6, 11)		2.3		V_{SS}	V
I_{en} = L	Low Voltage Enable Current (pins 6, 11)	V_{en} = L			–10	µA
I_{en} = H	High Voltage Enable Current (pins 6, 11)	V_{en} = H ≤ V_{SS} –0.6V		30	100	µA
$V_{CEsat (H)}$	Source Saturation Voltage	I_L = 1A I_L = 2A	0.95	1.35 2	1.7 2.7	V V
$V_{CEsat (L)}$	Sink Saturation Voltage	I_L = 1A (5) I_L = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V_{CEsat}	Total Drop	I_L = 1A (5) I_L = 2A (5)	1.80		3.2 4.9	V V
V_{sens}	Sensing Voltage (pins 1, 15)		–1 (1)		2	V

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T_1 (V_i)	Source Current Turn-off Delay	0.5 V_i to 0.9 I_L (2); (4)		1.5		μs
T_2 (V_i)	Source Current Fall Time	0.9 I_L to 0.1 I_L (2); (4)		0.2		μs
T_3 (V_i)	Source Current Turn-on Delay	0.5 V_i to 0.1 I_L (2); (4)		2		μs
T_4 (V_i)	Source Current Rise Time	0.1 I_L to 0.9 I_L (2); (4)		0.7		μs
T_5 (V_i)	Sink Current Turn-off Delay	0.5 V_i to 0.9 I_L (3); (4)		0.7		μs
T_6 (V_i)	Sink Current Fall Time	0.9 I_L to 0.1 I_L (3); (4)		0.25		μs
T_7 (V_i)	Sink Current Turn-on Delay	0.5 V_i to 0.9 I_L (3); (4)		1.6		μs
T_8 (V_i)	Sink Current Rise Time	0.1 I_L to 0.9 I_L (3); (4)		0.2		μs
fc (V_i)	Commutation Frequency	I_L = 2A		25	40	KHz
T_1 (V_{en})	Source Current Turn-off Delay	0.5 V_{en} to 0.9 I_L (2); (4)		3		μs
T_2 (V_{en})	Source Current Fall Time	0.9 I_L to 0.1 I_L (2); (4)		1		μs
T_3 (V_{en})	Source Current Turn-on Delay	0.5 V_{en} to 0.1 I_L (2); (4)		0.3		μs
T_4 (V_{en})	Source Current Rise Time	0.1 I_L to 0.9 I_L (2); (4)		0.4		μs
T_5 (V_{en})	Sink Current Turn-off Delay	0.5 V_{en} to 0.9 I_L (3); (4)		2.2		μs
T_6 (V_{en})	Sink Current Fall Time	0.9 I_L to 0.1 I_L (3); (4)		0.35		μs
T_7 (V_{en})	Sink Current Turn-on Delay	0.5 V_{en} to 0.9 I_L (3); (4)		0.25		μs
T_8 (V_{en})	Sink Current Rise Time	0.1 I_L to 0.9 I_L (3); (4)		0.1		μs

1) 1)Sensing voltage can be –1 V for t ≤ 50 μsec; in steady state V_{sens} min ≥ – 0.5 V.
2) See fig. 2.
3) See fig. 4.
4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

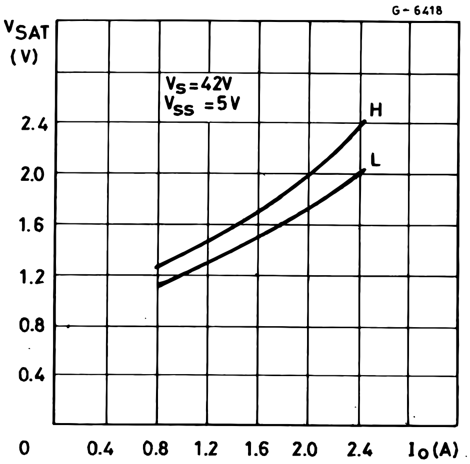
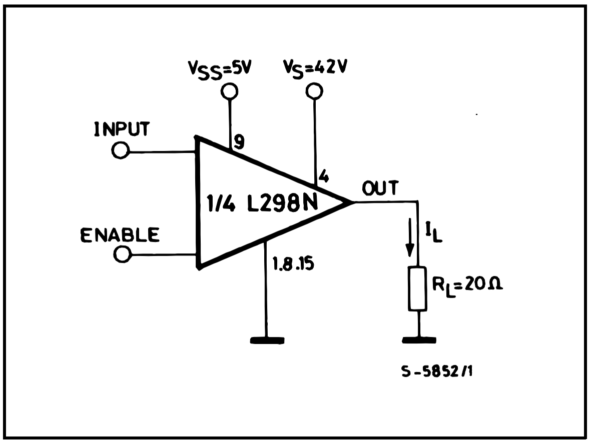


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = H

Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

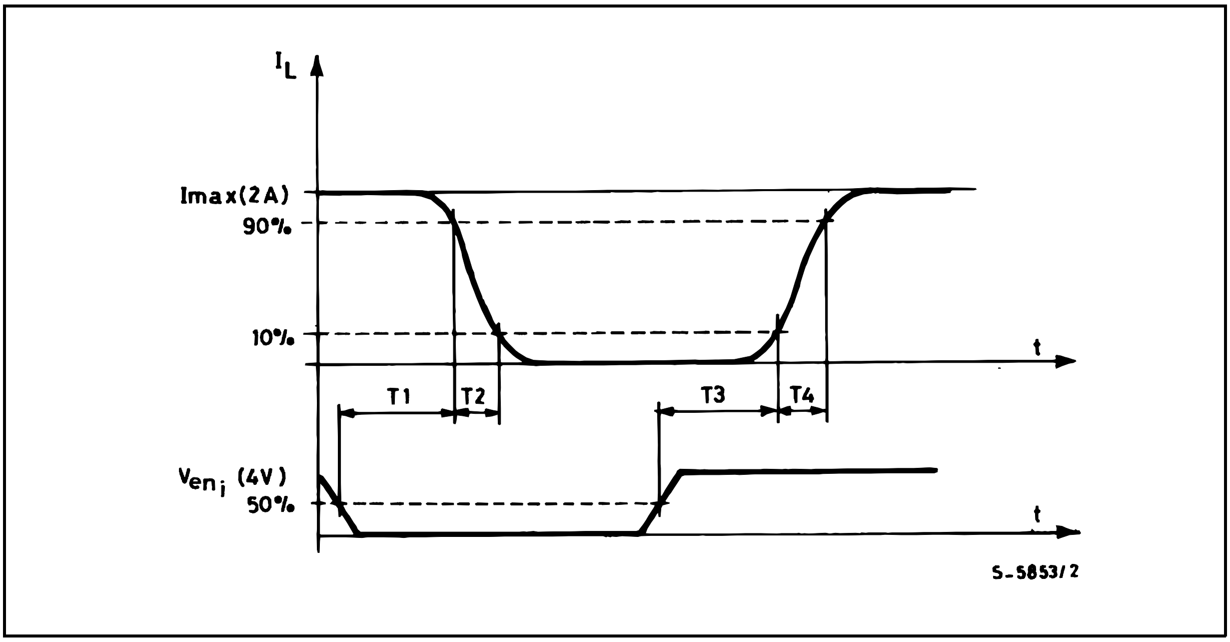
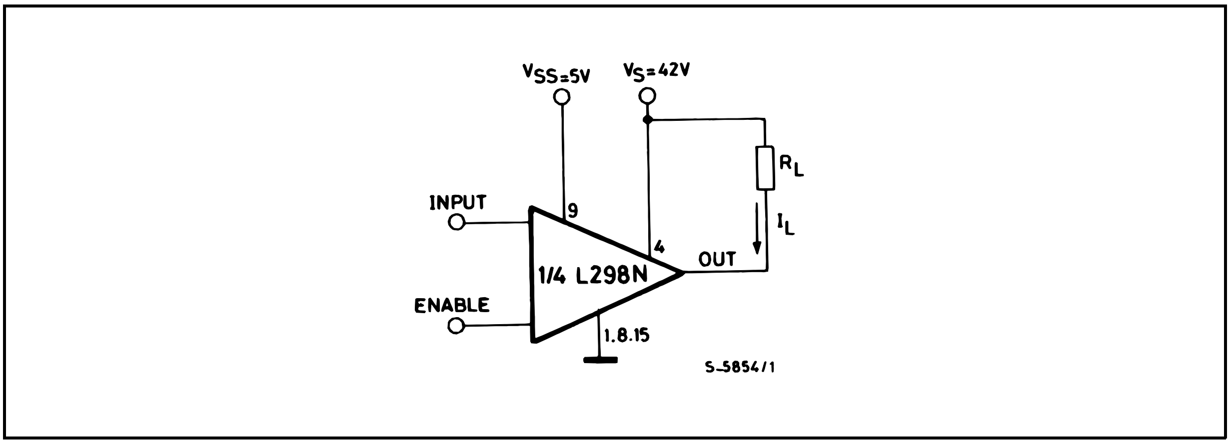


Figure 4 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = L

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

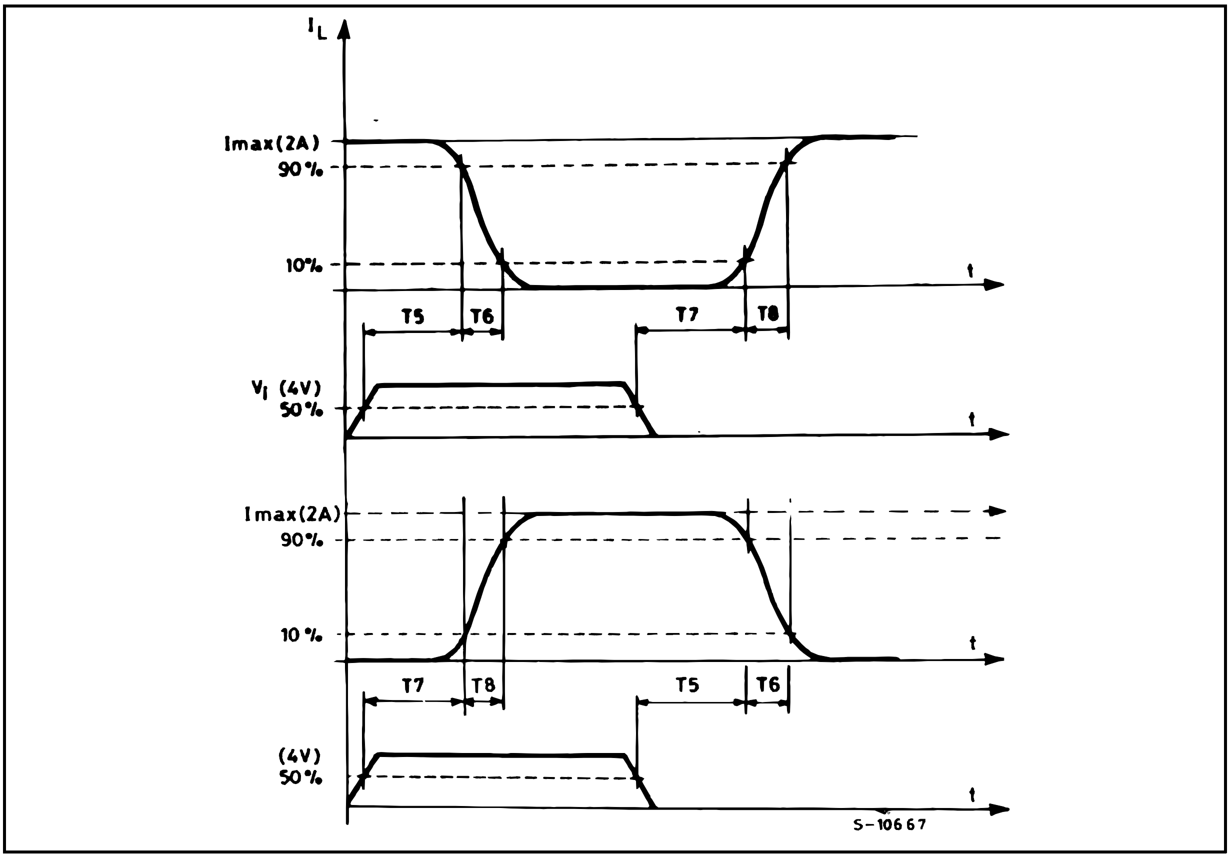
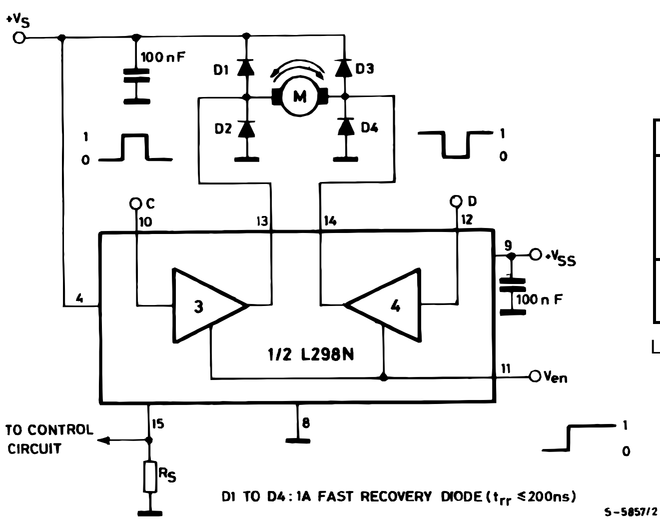


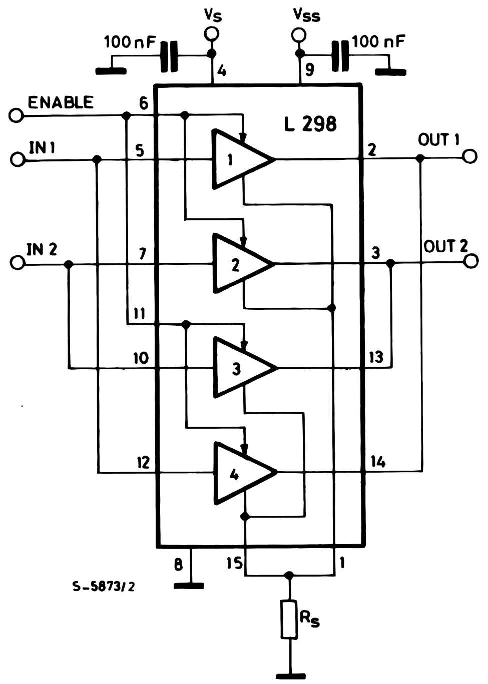
Figure 6 : Bidirectional DC Motor Control.



Inputs		Function
V_{en} = H	C = H ; D = L	Forward
	C = L ; D = H	Reverse
	C = D	Fast Motor Stop
V_{en} = L	C = X ; D = X	Free Running Motor Stop

L = Low H = High X = Don't care

Figure 7 : For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



APPLICATION INFORMATION (Refer to the block diagram)

1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A ; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differenzial mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output : an external resistor (R_{SA} ; R_{SB}.) allows to detect the intensity of this current.

1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are In1 ; In2 ; EnA and In3 ; In4 ; EnB. The In inputs set the bridge state when The En input is high ; a low state of the En input inhibits the bridge. All the inputs are TTL compatible.

2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both Vs and Vss, to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of Vs that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off : Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable input must be driven to the Low state.

3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes D1 to D4 is made by four fast recovery elements ($t_{rr} \leq 200$ nsec) that must be chosen of a V_F as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped ; Shottky diodes would be preferred.

This solution can drive until 3 Amps In DC operation and until 3.5 Amps of a repetitive peak current.

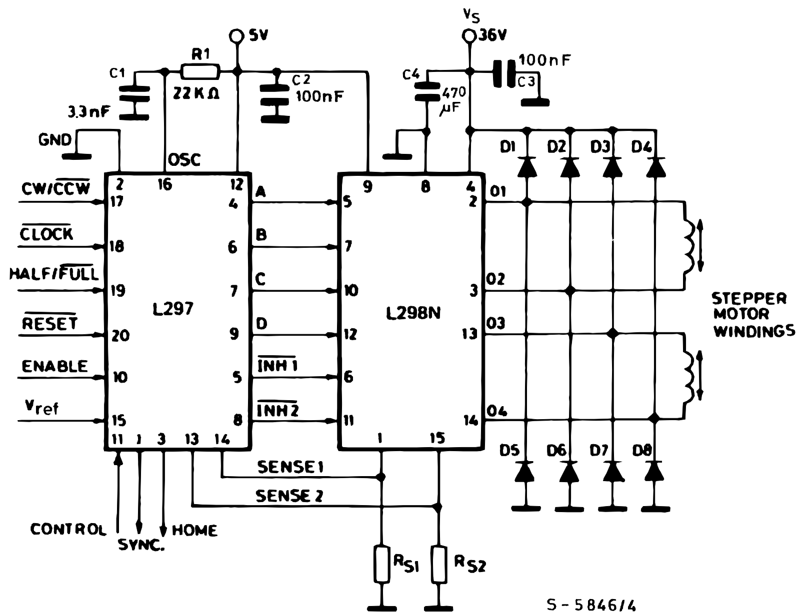
On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.

Figure 8 : Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.



$R_{S1} = R_{S2} = 0.5\ \Omega$

D1 to D8 = 2 A Fast diodes { $V_F \leq 1.2$ V @ I = 2 A; trr ≤ 200 ns

Figure 9 : Suggested Printed Circuit Board Layout for the Circuit of fig. 8 (1:1 scale).

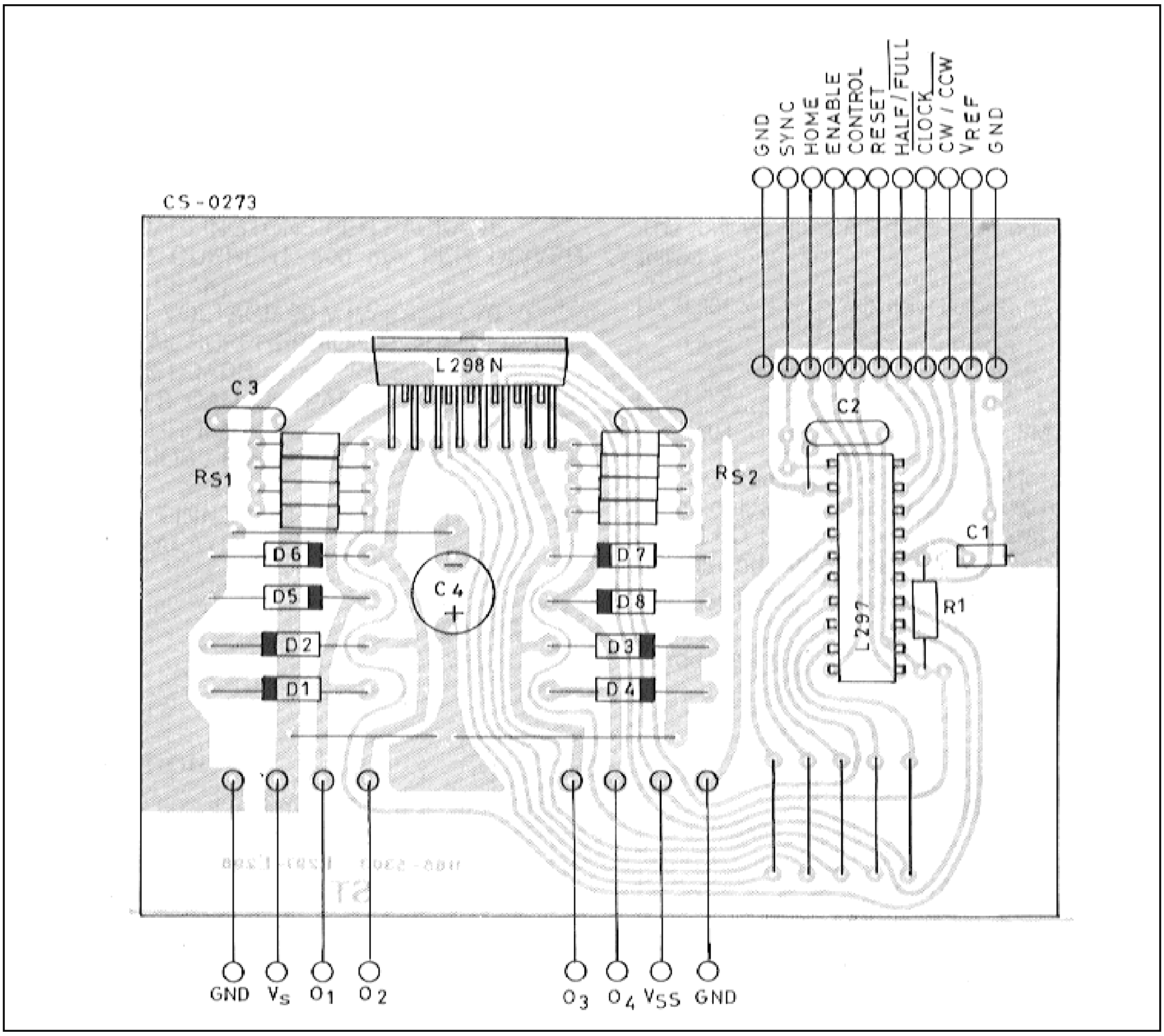
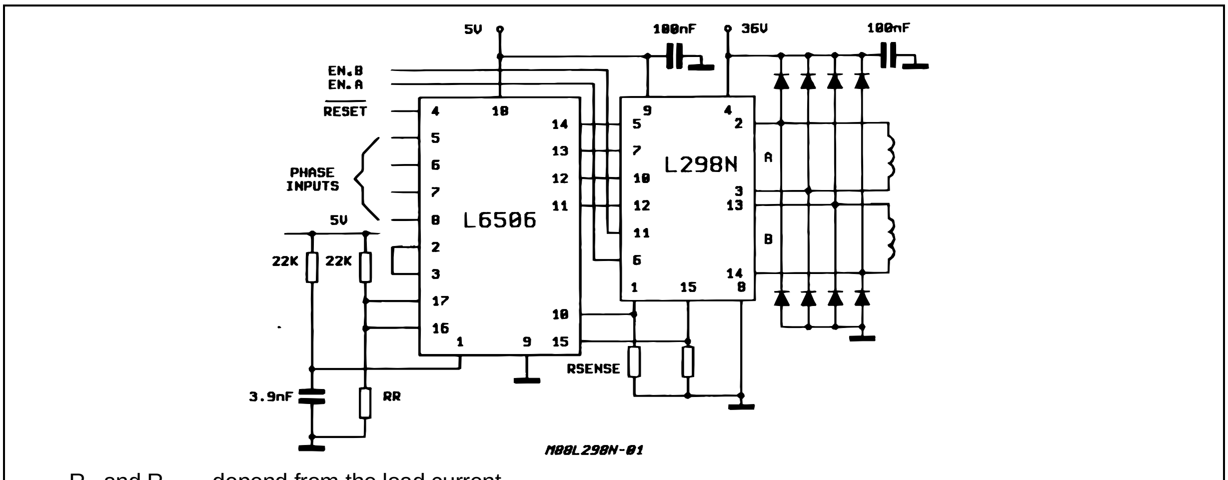


Figure 10 : Two Phase Bipolar Stepper Motor Control Circuit by Using the Current Controller L6506.



R_R and R_{sense} depend from the load current

Appendix B

Appendix B

B.1 Previous work on YouTube

1. https://www.youtube.com/watch?v=EEbdo2WdTOI
2. https://www.youtube.com/watch?v=xNrmGrjwpK8
3. https://www.youtube.com/watch?v=9hyi-0wbjdw
4. https://www.youtube.com/watch?v=0YD6MtTZSoY
5. https://www.youtube.com/watch?v=Xex4SHt6EVs
6. https://www.youtube.com/watch?v=v3P1aPGCspI
7. https://www.youtube.com/watch?v=tcSyTP44Ua4

Appendix C

Appendix C

C.1 Arduino code

```
/*

KEX 036: Robotic ball Retrieving

Ellen Severinsson
Julia Malachowska

*/

// SENSOR CONSTANTS
const float sound = 0.034;          // speed of sound in air
const float     c = 1.06;           // adjustment of sound wave speed from
experiments

// BOOLEAN
bool      started = false;          // switches value when scoop is lifted
bool isCollected = false;           // switches value after collecting
bool         done = false;          // switches value when all tasks are
fulfilled

// DISTANCES
const int   ballDistance = 7;       // when ball is within this distance,
the vehicle will stop and collect it
const int playerDistance = 120;     // when player is below this distance,
it will be detected
const int launchDistance = 25;      // when player is further away than
this, ball will be launched


// SENSOR
const int trigPin = 2;              // for Arduino pin setup
const int echoPin = 3;              // for Arduino pin setup
long duration;
int distance;

// STEPPER MOTOR
#include <Stepper.h>                // stepper library is used to control
the stepper motor
const int N1 =  12;                 // for Arduino pin setup
const int N2 =  7;                  // for Arduino pin setup
const int N3 =  8;                  // for Arduino pin setup
const int N4 =  4;                  // for Arduino pin setup
const int stepstot = 200;           // stepper motor data from data sheet
Stepper myStepper(stepstot, N1, N2, N3, N4);

// COLLECT
const int lyftmax = 80;             // number of seps, calibrated parameter
const int steg1 = 60;               // number of seps, calibrated parameter
const int steg2 = 30;               // number of seps, calibrated parameter

// DC-MOTORS
const int R_motorreverse = 10;      // for Arduino pin setup
const int R_motorforward = 11;      // for Arduino pin setup
const int L_motorforward = 6;       // for Arduino pin setup
const int L_motorreverse = 9;       // for Arduino pin setup

// DRIVING SPEED
const int    detectSpeed = 50;      // voltage applied on wheel when
collecting
const int          drive = 60;      // voltage applied on wheel when
driving to ball
```

```
// LAUNCH WHEEL
const int launchWheel_b = 5;         // for Arduino pin setup


void setup()
{
    // SENSOR SETUP
    pinMode(trigPin,         OUTPUT);
    pinMode(echoPin,         INPUT);

    // DRIVING SETUP
    pinMode(L_motorforward, OUTPUT);
    pinMode(L_motorreverse, OUTPUT);
    pinMode(R_motorforward, OUTPUT);
    pinMode(R_motorreverse, OUTPUT);

    // STEPPER MOTOR SETUP
    myStepper.setSpeed(22);
    Serial.begin(9600);             // Starts the serial communication, so
sensor data can be seen on screen

    // LAUNCH WHEEL SETUP
    pinMode(launchWheel_b,  OUTPUT);
}



void loop()
{
  if (started == false)             // This condition is only true once
  {
    myStepper.step(-lyftmax);       // Reassures that the scoop is on the
same position every time
    started = true;
  }

  if (done == false)
  {

  distance = sensor();              // Measuring distance
  if (isCollected == false)         // If the ball has not yet been
collected, this condition is true
  {
    while (distance >= ballDistance && distance !=0)   // breaks loop when
ball is within reach
    {
      analogWrite(L_motorforward, drive);
      analogWrite(R_motorforward, 0.78*drive);         // Adjusting voltage
on one of the motors to drive straight forward, 0.78 from experiments
      distance = sensor();                             // Measuring distance
    }
    delay(35);                                         // Gets the vehicle
in the right position with this delay
    analogWrite(L_motorforward, 0);                    // Stops the vehicle
    analogWrite(R_motorforward, 0);
    delay(2000);                                       // short pause

    distance = sensor();                               // Measuring distance
```

```
      do{
          collect();                                        // Collects the ball
          distance = sensor();                              // Measures distance
to make sure the ball is not on the groud after collecting is executed
        } while(distance <= ballDistance);

      isCollected = true;
      delay(1000);                                          // short pause

  }

  if (isCollected == true)                                  // Enters here when
collect is done
  {

    distance = sensor();                                    // Measures distance
    while (distance >= playerDistance)                      // breaks loop when
player is detected
    {
      analogWrite(L_motorforward, 1.2*50);                  // Vehicle rotates
because one wheel drives forward and the other revere
      analogWrite(R_motorreverse, 0.85*50);
      distance = sensor();                                  // Measures distance
    }
    delay(310);
    analogWrite(L_motorforward, 0);                         // Vehicle stops
moving
    analogWrite(R_motorreverse, 0);

   if (distance <= playerDistance && distance >= launchDistance)
   {
      launch();                                             // Ball is launched
towards player
      done = true;
   }
  }
  }
}

int sensor()
{
  // using the sensor and printing distance on screen
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = c*duration*sound/2;
  //print("Distance: ");
  Serial.println(distance);
  return distance;
}


int collect()
{
  // collecting the ball using the stepper motor connected to the lifting
arm
  myStepper.step(2*steg1/3);
  delay(2);
```

```
  analogWrite(R_motorreverse, detectSpeed);
  analogWrite(1.1*L_motorreverse, detectSpeed);
  myStepper.step(steg1/3);
  delay(400);
  myStepper.step(steg2);
  delay(100);
  analogWrite(R_motorreverse, 0);
  analogWrite(L_motorreverse, 0);
  delay(500);
  myStepper.step(-lyftmax);
  distance = sensor();

}

int launch()
{
  // launching the ball using the DC motor connected to the paddle wheel
  delay(200);
  analogWrite(launchWheel_b, 2*distance);
  myStepper.step(-28);
  delay(4000);
  analogWrite(launchWheel_b, 0);
  delay(300);
}
```

TRITA TRITA-ITM-EX 2019:52