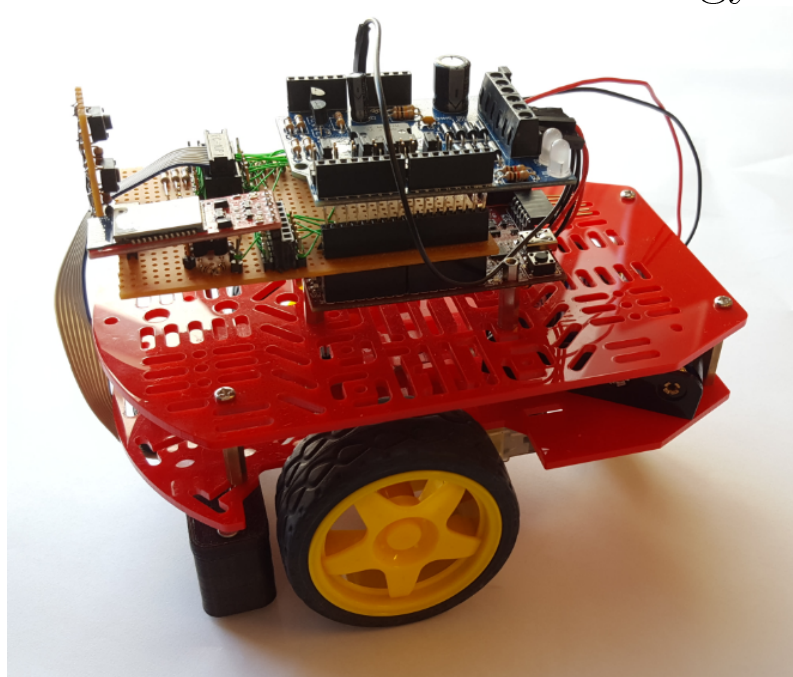


Fall Semester 2016

Autonomous Object Avoidance Robot

Group 2
3. Semester IT-Technology



Group members: Benjamin Nielsen - Henrik Jensen - Martin Nonboe - Nikolaj
Bilgrau

Supervisor: Jesper Kristensen - Steffen Vutborg

Title:

Autonomous Object Avoidance
Robot

Project Period:

3. Semester | Spring semester 2016

Projectgroup:

Group 2

Group participants:

Benjamin Nielsen
Henrik Jensen
Martin Nonboe
Nikolaj Bilgrau

Supervisors:

Jesper Kristensen
Steffen Vutborg

Pages:

Appendices:

Completed:

Preamble

This project was written by group 2, for the 3rd semester on the IT-electronics education at university college Nordjylland, Sofiendalsvej 60. The project goal is to make an autonomous robot that can navigate a course utilizing object avoidance and localization.

Benjamin Nielsen

Henrik Jensen

Martin Nonboe

Nikolaj Bilgrau

Table of Contents

3D print	3-Dimensional printing
ADC	Analog-digital conversion
GUI	Graphical User Interface
IDE	Integrated Development Environment
MCU	Microcontroller Unit
PCB	Printed Circuit Board
PID	Proportional-integral-derivative
PWM	Pulse-width modulation
THT	Through-hole-technology
UART	Universal Asynchronous Receiver/Transmitter

Introduction 1

In this section the problems found when trying to get the robot function, will be listed and analysed

2.1 Problem statement

The problem presented to the group is how to make a robot move from point A to point B, with the help of different sensors, including ultrasound and infrared, and to make use of autonomous algorithms to avoid obstacles.

Problem statement:

- Bot should be able to move from A to B
- Should be able to stop at a predetermined point
- Manoeuvre around obstacles

2.2 Problem analysis

2.2.1 Mobility from A to B

The robot receives a coordinate to reach, and will use its own starting point to determine a direction to drive towards the given coordinate. The robot will need a way to control its movement and direct current to function optimal.

The robot needs a way to effectively regulate speed and also steer itself autonomously. To dictate how quickly the robot moves, the robot will need some system that allows it to move around on a flat surface, the robot needs to be able to move around from point A to point B. .

2.2.2 Predetermined end point

After starting, the robot needs to know when to stop. The pre-determined end-point consists of a series of circles which the robot needs to detect.

2.2.3 Obstacles avoidance

As part of its functionality, the robot needs to be able to see objects that are in front of it and avoid them.

Requirements specification 3

This section specifies the requirements. The requirements have been found through the analysis.

[ucn]

- The robot needs line following capabilities
- The robot needs object avoidance
- The robot should make use of an H-bridge
- The robot should make use of Motors
- The robot needs a way to implement motor control
- The robot should make use of a micro-controller unit
- The robot should make use of the Magician chassis

Hardware section 4

4.1 Description of the hardware structure and functionality

In this section the different components of the hardware will be listed, described and explained. TBD: Mere fyld

4.2 Hardware diagram

TBD: Indsæt hardware diagram
Beskrivelse af hardware diagram

4.3 Sensors and sensor concept

TBD: Find out why this doesn't show up in build...

The robot will utilize two sets of sensors - one set of QRE1113 sensors, which will be used for line-following capabilities, they are fastened towards the end of the robot, and will give the robot a way to detect what surface it is about to enter.

The second is a hybrid set of ultrasound and infrared sensors. These will be working together to make the robot able to navigate open spaces more precisely, since infrared and ultrasound sensors work the best under different circumstances. This will end up as a product which is more optimized for usage in situations that would not be ideal for one of the other, since the hybrid design will leverage shortcomings of a given sensor method.

4.3.1 Ultrasound sensor

When a robot should be able to avoid obstacles it will need a device to inform the robot where its position is compared to the obstacle. This is where an ultrasound sensor plays an important role. For this task the " TBD " has been picked.

The way the ultrasound sensor works is by emitting acoustic waves and then waits for the waves to reflect back to the sensor. The waves are often at about 40 kHz and humans are unable to detect the sounds because of the frequencies being above the human audible range.

What is causing the device to make ultrasonic sound is a piezoelectric crystal. The crystal is receiving a rapid oscillating electrical signal, this causes the crystal to expand and contract and thereby creating a sound wave. The sound waves will then after being reflected return to a piezoelectric receiver which can then convert the

waves into voltage by using the same method as explained above.

There are several popular ways to process the information gathered from the ultrasound sensor.

- Time of flight
- Doppler shift
- Amplitude attenuation

In the scope of the project, the robot will be using "Time of flight" for sensing the distance between itself and the obstacle.

When working with the term time of flight, it means the ultrasound sensor only generates pulses of sound instead of an continuous streak of sound waves. to avoid confusion. In high speed situations this will mean there is waiting time limits.

The calculation for using the ultrasound sensor is:

t = time

r = distance travelled

c = speed of light

$r = c * t$

With this the robot can calculate the time of flight. TBD SR04 Ultrasound:

Considerations:

When using the ultrasound as a sensing tool, there are some factors that must be taken into consideration.

Temperature and humidity can affect the speed of sound, just as air currents have been known to be able to create invisible boundaries that can reflect ultrasonic waves.

Ultrasound sensors have something called a dead zone, this occurs when an object is in front of the sensors and the receiver can't keep up.

Some materials are very absorbent, which will result in less reflected ultrasound to be detected by the receiver.

4.3.2 Infrared sensor

The robot will make use of infrared sensors in symbiosis with the aforementioned ultrasound sensors. This will allow it to take readings on a wider array of surfaces, as infrared sensors are better suited for less even surfaces. They work by emitting infrared light onto a surface, and then taking a reading based on the amount of light that gets reflected. A light surface will reflect more light back than a dark one. The sensor will then output a feedback signal made of a certain amount of voltage, ranging from 1% to 100%, based on how much light was reflected back. Based on

this output voltage, it is possible to use an ADC to convert these signals into digital signals, which can be monitored more conveniently. Functionally, the robot is left with a way of knowing which surface the sensors are above - and in the case of a track with a black line to follow, this allows it to detect where the line it needs to follow is.

4.3.3 Hybrid-infrared sensors - TBD name

GP2Y0A02YK0F (long)

GP2Y0A41SK0F (short)

4.3.4 Line following sensor choice - the QRE1113

The QRE1113 sensor TBD: Er det overhovedet den vi vil bruge?

Due to past experiences, the QRE113 sensor has been chosen to be utilized on the robot to enable its line-following properties. TBD: Nyt afsnit om QRE1113

The sensor then regulates its output voltage from 1% to 100%, or 0V to 3.3V. TBD: Passer spændingen her?

TBD sensor specs.

TBD måske lille skema over specs?

4.4 Analog-to-digital converter

The usage of ADC

TBD (skal vi overhovedet forklare det igen? B: Vi skal nok forklare hvordan, og til hvad, vi udnytter det i projektet)

4.5 The chipKIT Uno32 board

The robot will utilize the chipKIT Uno 32 board to execute code. The board was chosen both due to past experiences, but also because the robot would need line-following properties, and we had already written a functional line-following robot previously, which also included some important features, including PID control and pulse-width modulation patterns.

This enabled a lot of recycled code, which was a strong point in the Uno 32's favor due to time constraints.

The board is also compatible with Arduino shields, and as such designing the H-bridge for it becomes more straightforward. It's fast enough to execute the code, and works well within the input power the robot will utilize.

4.6 The motor shield

The motor shield is containing the H-bridge and will be the board for ensuring control of the different components and motors. TBD (hvad er der helt præcist på boarded?)

4.6.1 The H bridge

The robot will make use of an H-bridge. An H-bridge is a circuit made for controlling the motor of the robot, by making sure the motor will never try to do forward and backward motion and cause errors or short circuits. The point of using an H-bridge is to ensure motor safety and functionality.

4.7 The Bluetooth transceiver

The robot will utilize the BlueSmiRF Silver bluetooth transceiver. The transceiver is made by Sparkfun, it is utilized to make use of an GUI, by sending data from the MCU to the computer (GUI) by the use of bluetooth.

The bluetooth transceiver makes it possible to monitor both the inputs and the logic behind the steering. The baudrate is between 2400-115200 bps and the transceiver can be powered from 3.3v up to 6v.

4.8 Part conclusion

TBD H-bro problemer og løsning

Software section 5

Beskriv Software section

5.0.1 Software diagram

5.1 Analog to digital conversion

5.2 PID controller

5.2.1 Proportional control(P)

5.2.2 Integral control(I)

5.2.3 Derivative control(D)

5.2.4 Loop tuning

5.2.5 Steady-state error

5.2.6 Stability

Table manual explained

5.2.7 PID Implementation

5.3 Pulse-width modulation

5.3.1 Why utilize pulse-width modulation

Pulse-width modulation, or PWM, is a way to regulate power distribution within a system. It is a software solution that manages when a device receives power, and for how long at a time it does this. This is called a duty cycle. The robot utilizes PWM for its motors, to regulate how quickly it moves. PWM can be compared to turning a switch on and off extremely quickly - much more quickly than what will affect the performance of the motors. Effectively, this means that the robot's programming will now be able to regulate speed autonomously.

5.3.2 Duty cycles

The duty cycle is used to describe how long the power is 'on' compared to 'off'. A higher duty cycle will yield more energy than a low one. The programming uses a frequency of 1000Hz, which makes it straightforward to calculate to real time, if this is needed - it also provides enough precision to make the motors responsive quickly.

5.4 The interface

Beskriv test section

6.1 Unit Testing

6.1.1 Infrared TBD: Find navn på sensor

Equipment

- Hameg HM8040-2 Triple Power Supply
- Agilent MSO-X 3024A Oscilloscope

Setup

Power to the sensor is supplied by the power supply and the output is read on the oscilloscope.

Results

30mm: 2.29V
50mm: 3.07V
100mm: 2.34V
200mm: 1.31V
500mm: 515mV

As is evident in the results above, this sensor does not work at low distances, which is also described in the datasheet.

6.1.2 Ultralyd TBD: Find navn på sensor

Equipment

- Arduino UNO
- Agilent MSO-X 3024A Oscilloscope

Setup

A small program for the Arduino has been written which allows the MCU to trigger the sensor and waits a pre-specified amount of time before triggering again, to allow for the return of the ultrasound wave.

Results

30mm: 152 us

50mm: 285 us

100mm: 600 us

200mm: 1.23 ms

500mm: 2.32 ms

6.1.3 QRE1113**Equipment**

- Hameg HM8040-2 Triple Power Supply

Setup

Power to the motors is supplied by the power supply, which also measures the current.

Results

Measured current on both engines is between 70 and 80 mA when running freely.

6.1.4 DC Motors**Equipment**

- Hameg HM8040-2 Triple Power Supply
- Fluke 45 Multimeter

Setup**Results****6.1.5 H-Bridge****Equipment**

- Elcanic Power Supply
- Fluke 45 Multimeter

Setup

Power to the H-bridge is supplied by the power supply. Current and voltage will be measured by the multimeter. All of the Mosfet transistors were removed before doing the test. It was done to avoid powering the engines.

Results

There was a suspicion concerning the H-bridge, it had been measured there was an error or a damaged component. The affected part in suspicion was the H-bridge for the left motor.

All of the transistors were tested on the part of the H-bridge in question. Since there was 0 in value on the base, it is suspected that a resistor which sat on the board just before the transistor. This and the other resistor values were measured and found to be identical, based on the measures it could be concluded that it was probably a failed transistor.

It was concluded that the damaged transistor was the: Q14 (TBD link til schematics).

The results were as following:

Normal NPN transistors: base 0.854V, collector 0.073V

Faulty NPN transistor. base 0V, collector 0V.

6.1.6 PWM

Equipment

Setup

Results

6.1.7 ADC

Equipment

Setup

Results

6.2 Integration Testing

6.2.1 PWM motor control

Equipment

Setup

Results

6.2.2 Robot to Interface communication

Equipment

Setup

Results

6.3 System Testing

Equipment

Setup

Results

6.4 Acceptance Testing

Equipment

Setup

Results

Conclusion 7

Skriv en fucking Conclusion!!

Appendices

8.1 Group collaboration agreement

8.1.1 Contact Information

Table 8.1: Contacts

Benjamin Nielsen	Tlf: 30427645	@: yipiyuk5@gmail.com
Henrik Jensen	Tlf: 28568934	@: henrik_kort@hotmail.com
Martin Nonboe	Tlf: 23827566	@: nonsens_4@hotmail.com
Nikolaj Bilgrau	Tlf: 29802715	@: nikolajbilgrau@gmail.com

8.1.2 Workflow

8.1.3 Deadline

8.1.4 Milestones and goals

Gerne en kalender der viser dage arbejdet!

List of references 9

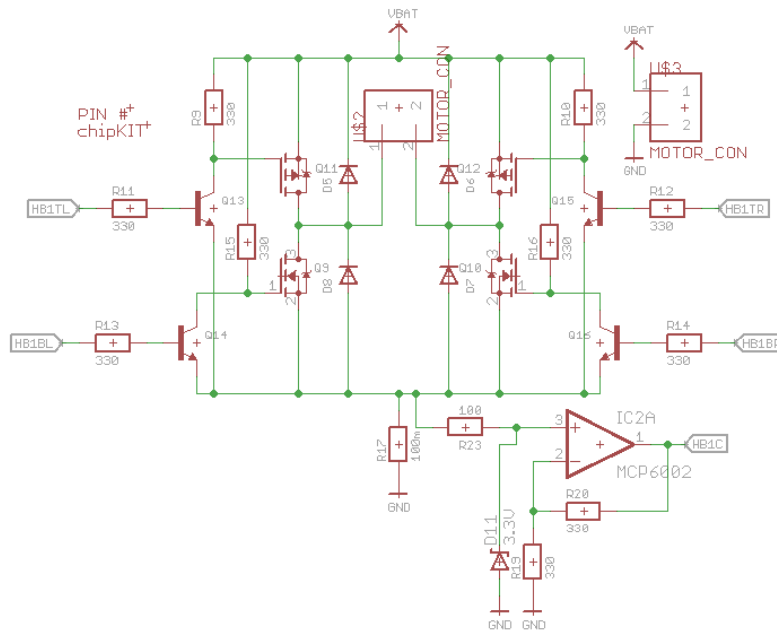
List of Figures

List of Tables

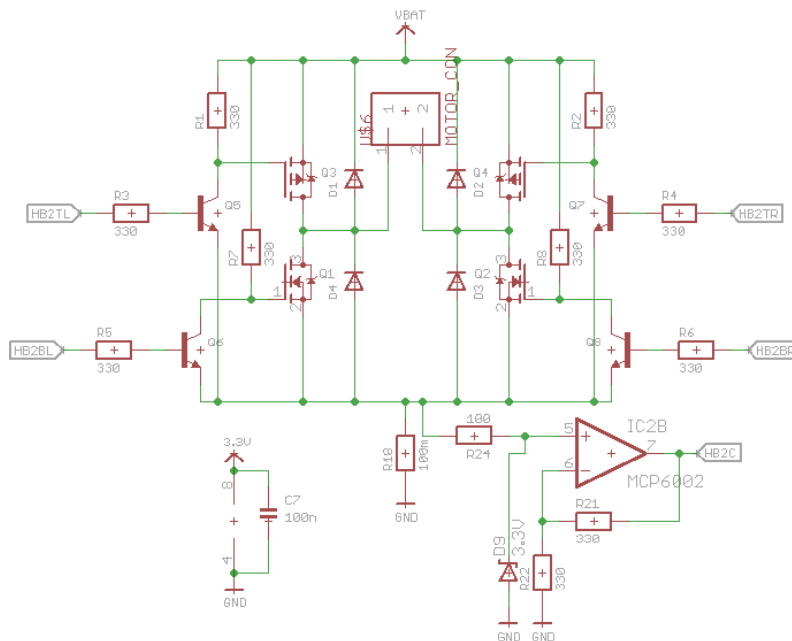
10

10.1 Hardware Schematics

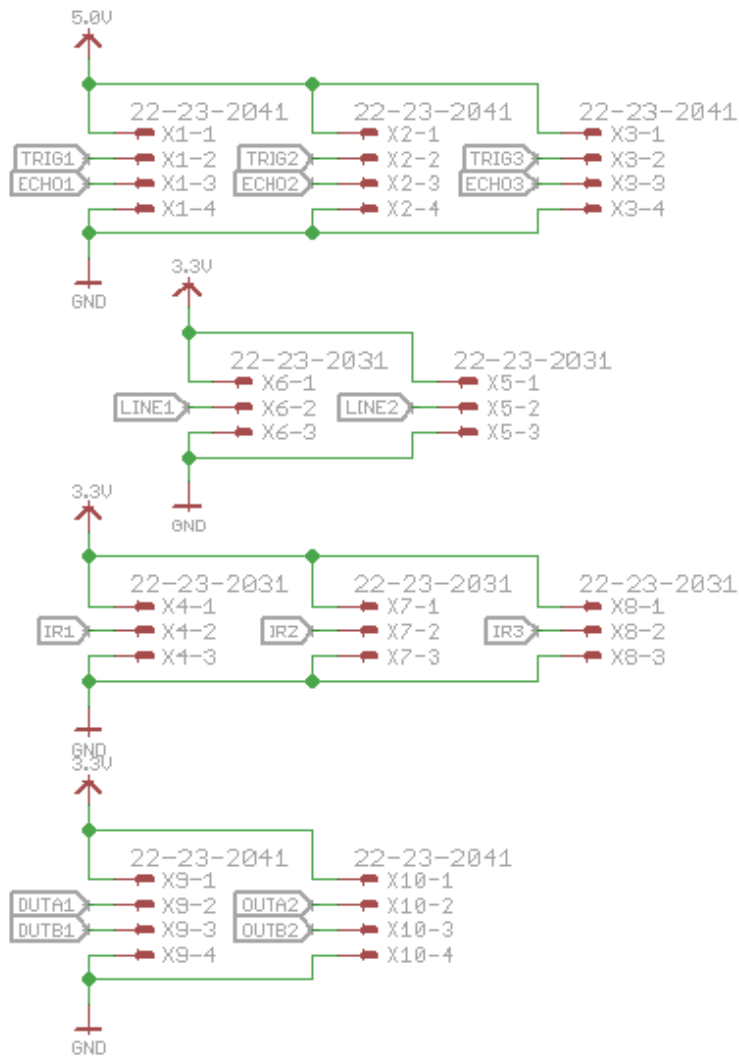
10.1.1 H-bridge 1



10.1.2 H-bridge 2



10.1.4 Power



Software appendix 11

11.1 C code

main.c:

ADC.c:

11.2 C# code - interface