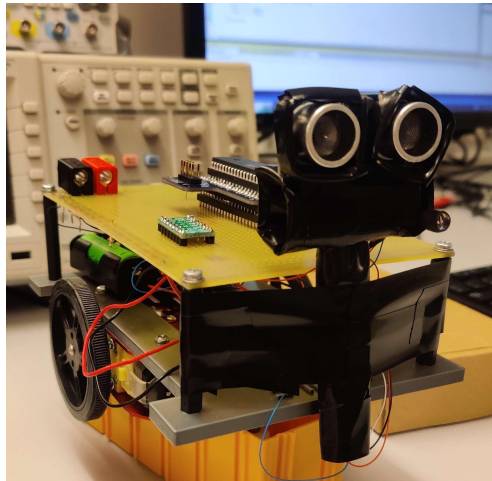




**LTH**  
LUNDS TEKNISKA  
HÖGSKOLA

# Ljussökande bil

EITA15 - Digitala projekt  
Vårtermin 2023



**Gruppmedlemmar:**

Nora Lindgren  
Sara Alvandi  
Johannes Kull  
Hampus Anderstedt  
Henrik Berthilsson

**Handledare:**

Bertil Lindvall

## **Abstract**

This report has been written as a part of the “Digital systems” course offered by LTH (Faculty of engineering). The aim of the course is to develop an application, using the knowledge acquired during the education, through a project. In this project, we have constructed a robot that detects light and then attempts to move towards it. When the robot has reached a sufficiently close distance to the light source it stops.

## Innehållsförteckning

Abstract.....	1
Inledning.....	3
1 Kravspecifikation.....	3
1.1 Den ursprungliga kravspecifikationen.....	3
1.2 Den nya kravspecifikationen.....	3
2 Arbetsmetod.....	3
2.1 Komponenter.....	3
- Hårdvara.....	3
- Mjukvara.....	5
2.2 Genomförande.....	5
3 Resultat.....	6
3.1 Problem och svårigheter.....	6
3.2 Lärdomar.....	6
4 Appendix A : Figurer.....	7
5 Appendix B: Källkod.....	8
6 Referenser.....	11

# Inledning

Målet med projektuppgiften är att skapa en prototyp som kan vidareutvecklas med dokumentation för att underlätta för framtida utveckling. Den största delen av kursen fokuserar på att designa, bygga och testa olika konstruktioner. För vår projektuppgift har vi valt att utveckla en självkörande robot som ursprungligen var avsedd att navigera runt i inomhusmiljöer och kunna hitta och släcka "bränder" genom att spruta vatten. På grund av vissa hinder i konstruktionen var det dock inte möjligt att uppnå detta mål. Istället har roboten fått en ny funktion där den signalerar genom blinkande ljus.

## 1 Kravspecifikation

### 1.1 Den ursprungliga kravspecifikationen

1. Värmesökande bilen är självgående och svänger på egen hand.
2. Värmesökande bilen känner av värme och ger signal i form av ljus (LED).
3. Värmesökande bilen sprutar vatten istället för att ge signal i form av ljus (LED).

### 1.2 Den nya kravspecifikationen

1. Ljussökande bilen är självgående och svänger på egen hand.
2. Ljussökande bilen känner av ljus och stannar framför ljuskällan.

## 2 Arbetsmetod

### 2.1 Komponenter

- Hårdvara

I detta avsnitt beskrivs de olika komponenterna som används för att bygga hårdvaran. En detaljerad kopplingsschema över hårdvaran presenteras i Appendix A, Figur 1. Kopplingsschemat visar hur de olika komponenterna är sammankopplade och hur de interagerar med varandra för att uppnå önskad funktionalitet. Det är viktigt att noggrant följa kopplingsschemat vid montering och installation av hårdvaran för att säkerställa dess korrekta funktion.

Utrustningen som vi har använt oss av är följande:

- **50:1 Micro<sup>1</sup> Metal Gearmotor HP** är en liten likströmsmotor med högt vridmoment vid låga hastigheter. Den är tillverkad av Pololu och har en utväxling på 50:1, vilket gör den lämplig för robotteknik och andra små applikationer som kräver högt vridmoment och låg hastighet.

---

<sup>1</sup> Pololu - 50:1 Micro Metal Gearmotor HP, <https://www.pololu.com/product/3051>

- **Ultrasonic<sup>2</sup> Distance Sensor - HC-SR04** fungerar genom att sända ut högfrekventa ljudvågor och mäta den tid det tar för vågorna att studsas tillbaka. Den beräknar sedan avståndet baserat på ljudets hastighet.
- **DRV8835<sup>3</sup> Dual Motor Driver** är en motor drivrutin med dubbla H-bryggor som kan styra två DC-motorer eller en bipolär stegmotor. Den fungerar från 2,7 V till 10,8 V och kan leverera en kontinuerlig ström på upp till 1,2 A per kanal, med toppströmmar upp till 2 A. Den har ett enkelt logiskt gränssnitt, som möjliggör enkel integrering i mikrokontrollerbaserade system.
- **ATMEGA<sup>4</sup> 1284p** är en kraftfull 8-bitars mikrokontroller med en klockhastighet på upp till 20 MHz och 128 KB Flash-minne. Den har många I/O-pinnar och avancerade funktioner som PWM, I2C, SPI och UART-gränssnitt, vilket gör den till en populär mikrokontroller inom industriella kontroller, medicinsk utrustning och trådlösa nätverksenheter.
- **Ljusdiod<sup>5</sup> - AMS302** är en ljusdiod som ger ut ström i ett linjärt förhållande till ljusstyrkan.

- Mjukvara

---

<sup>2</sup> ElecFreaks. HC-SR04 Ultrasonic Sensor User Manual. <https://www.electfreaks.com/11750.html>

<sup>3</sup> Texas Instruments. (2014). DRV8835 Dual Motor Driver. Datasheet. <https://www.ti.com/lit/ds/symlink/drv8835.pdf>

<sup>4</sup> Microchip Technology. (2023). ATmega1284P Microcontroller. <https://www.microchip.com/en-us/product/ATmega1284P>

<sup>5</sup> Panasonic Industry. AMS302 Light Sensor NaPiCa. <https://industrial.panasonic.com/ww/eol/pt/napica/models/AMS302>

All kod för roboten skrevs i Atmel Studio 7.0 (se Appendix B) och inga bibliotek förutom de inbyggda i Atmel Studio användes.

Koden för roboten använder sig av de tre huvudkomponenterna som används för att skapa en ljus sökande bil. För att mäta avståndet mellan bilen och ett objekt skickas en signal i 15 mikrosekunder till Ultraljudssensorn vilket gör att sensorn skickar ljudvågor. Efter en stund blir ECHO-pinnen hög och den tid som ECHO-pinnen är hög kan man sedan använda för att räkna ut avståndet. För att mäta tiden ECHO är hög använder vi oss av en intern timer i MCU:n som startar då ECHO är hög och stoppar då ECHO blir låg igen.

För att roboten ska kunna färdas används två DC-motorer som är PWM-styrda. För att kunna reglera hastigheten samt riktningen på motorerna används fyra portar på MCU:n kopplade till två olika timers. Dessa timers kan ställas in så att de istället genererar PWM signaler med hjälp av deras respektive TCCR register.

Den ljus sökande delen av roboten görs med hjälp av en fotosensor som kan detektera ljus och skapa en analog signal. För att använda denna signal utnyttjas A/D omvandling vilket sker genom den inbyggda ADC:n i MCU:n.

Till sist förs alla dessa funktionerna samman i koden för att skapa den självkörande roboten. Först startar en av motorerna och roboten börjar rotera. Sedan skickas en signal till ultraljudssensorn och därefter läses fotosensors värde. Ifall fotosensorns värde har nått en viss tröskel ska den rotera samma hålls tills detta värde går under tröskeln. För att roboten ska röra sig framåt byts det mellan vilken motor som drivs då värdet går under tröskeln. Samtidigt som detta sker räknas avståndet till närmsta objekt ut och då det går under en satt gräns stannar roboten vilket ofta är vid ljuskällan.

## 2.2 Genomförande

För att skapa roboten testades först alla komponenter separat för att förhindra störningar mellan komponenterna.

De första komponenterna som prövades var DC-motorerna. Först kopplades motorbryggan och motorerna till portarna enligt Appendix A, Figur 1 och därefter skrevs koden som genererade PWM signaler till dem. När roboten tillslut kunde drivas och styras med hjälp av PWM-signaler kunde nästa komponent testas.

Nästa komponent var Ultraljudssensorn eftersom denna testades bäst med hjälp av motorerna. Först kopplades den enligt Appendix A, Figur 1 och sedan skrevs koden för att mäta avstånd. När ett tillfredsställande resultat nåddes prövades den även med hjälp av motorerna genom att stanna när roboten var tillräckligt nära ett objekt.

Den sista komponenten som prövades var fotosensorn. Först kopplades den enligt Appendix A, figur 1 och sedan skrevs det kod för att översätta den analoga insignalen. När detta tillslut fungerade byggdes hela roboten samman och därefter kunde robotens huvudbeteende implementeras.

## 3 Resultat

### 3.1 Problem och svårigheter

Vår ursprungliga idé var att göra en bil som söker efter värme, detta gick inte då vår värmesensor vi hade fått inte fungerade. Istället bytte vi till en ljussensor som fanns i labblokalerna. Efter att vi tagit bort komponenter på kopplingsdäcket vi inte längre behövde till ljussensorn blev vi klara.

Vi fann att spänningen från våra fyra AA-batterier räckte inte för att driva alla komponenter. Detta ledde till att motorerna fick för lite spänning vid vissa tillfällen vilket fick dem att arbeta långsammare och i osynk. På grund av detta var det svårt att få en robot som kunde åka rakt och inte hacka för mycket. Vi löste detta genom att enbart köra en motor i taget och sicksacka fram.

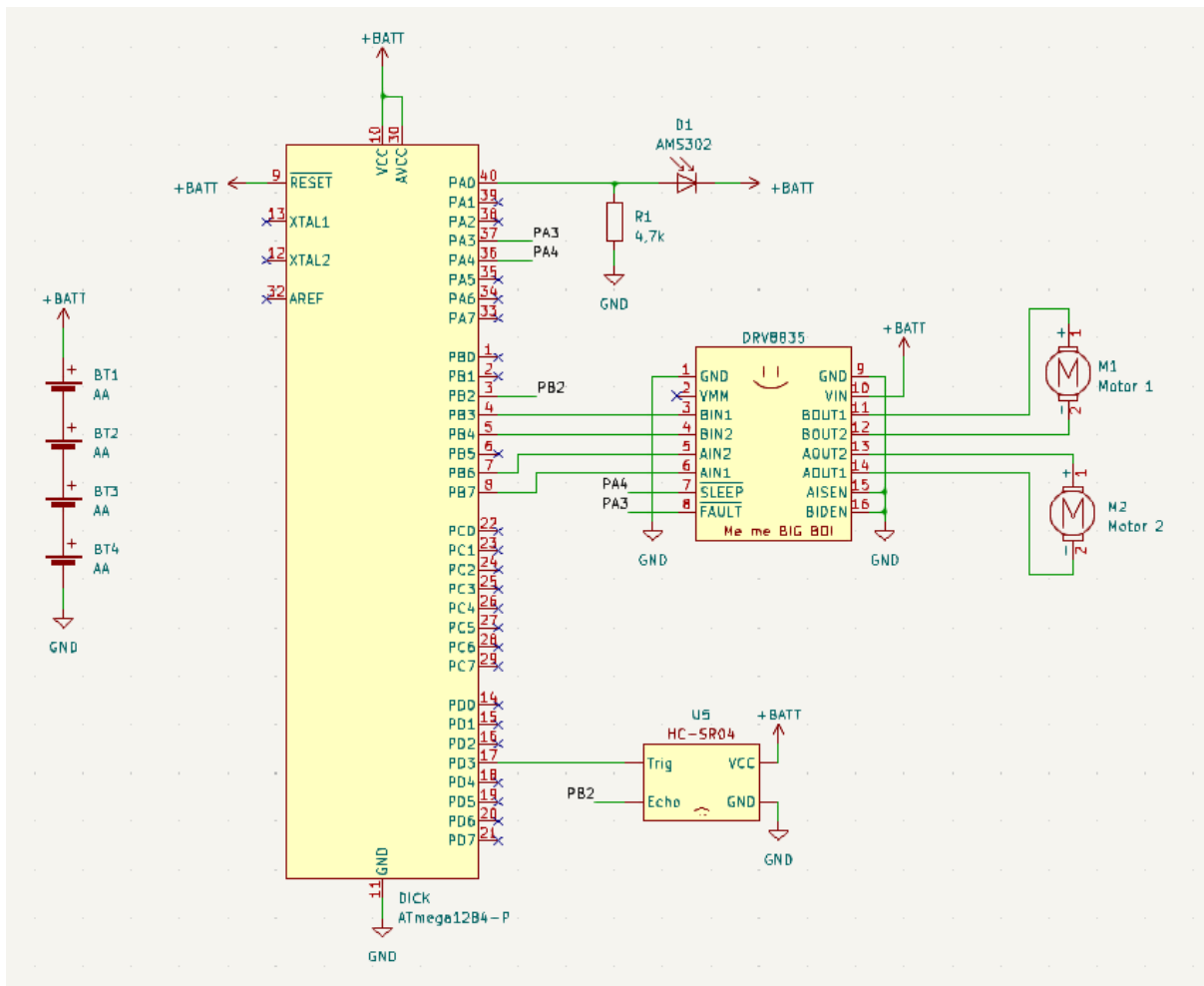
Vår distansmätare hade ofta problem att dess mätning skilde en del från den verkliga distansen, vid vissa tillfällen kunde detta vara en skillnad upp mot en meter. Det krävde då att vi fick ta ett antal mätvärden och dra medelvärde på dessa för att minska risken för mätfel. Men detta ledde till ett nytt problem då tiden för ett resultat tog längre tid vilket fick den att ibland köra in i vägen.

I vår kod så krävdes det användning för delay, både vid mätning och även vid körandet. Men för att kunna använda Atmels inbyggda delay så behövde man använda optimering för att få rätt tidsintervall. Då vi inte kunde använda oss av optimering gjorde det att väntetiden skilde ganska drastiskt från värdet som var inskrivet. Vi fick istället testa oss fram för att försöka hitta något värde som funkade.

### 3.2 Lärdomar

Kontrollera att sensorer och andra komponenter verkligen fungerar innan man fortsätter med källkoden och kopplingen. Då det senare kan vara svårt att felsöka vad som kan vara fel. Även att kontrollera att vi sagt till om alla delar vi kan behöva, eftersom vi insåg när vi skulle börja bygga bilen att vi var tvungna att 3D-printa en platta som kopplingsdäcket kunde sitta fast på.

# 4 Appendix A : Figurer





## 5 Appendix B: Källkod

```
/*
 * projektWalle.c
 *
 * Created: 2023-05-17 17:19:41
 * Author : jo6612ku-s
 */

#define F_CPU 8000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#include <stdlib.h>
#include <util/delay.h>
#define ECHO 0b00100000
#define TRIGGER 0b00001000

volatile double distance = 0;
int i = 0;
int bugfix = 0;
int start = 0;
int truedist = 0;
volatile int light = 0;

ISR(INT2_vect){ //vad som sker när ECHO ändras
    if(start == 0)
    {
        TCCR1B |= (1 << CS10); //startar timer
        start = 1;
    }
    else if(start == 1)
    {
        distance = TCNT1 * pow(10, -6) * 0.125 * 17150; //räknar ut distans i cm
        TCCR1B = 0x00;
        //TCCR1B &= ~(1 << CS10);
        TCNT1 = 0;
        start = 0;
        _delay_ms(50);
    }
}

void trigger(){ //skickar en signal på trigger pinnen som varar i 15 mikrosekunder
    PORTD = 0b00001000;
    _delay_us(15);
    PORTD = 0x00;
```

```

}

void m_init(){ //motorerna
  DDRB = 0x00;
  PORTA = 0x00;
  PORTB = 0x00;
  DDRB = 0b11011000;
  //PWM pin PINB3
  TCCR3A |= (1 << WGM32) | (1 << WGM30);
  TCCR3B |= (1 << CS32) | (0 << CS31) | (1 << CS30);
  OCR3A = 30;
  //
  TCCR0A |= (1 << WGM00) | (1 << WGM01);
  TCCR0B |= (1 << CS02) | (0 << CS01) | (1 << CS00);
  OCR0B = 30;
}

void u_init(){ //ultraljud sensorn
  DDRD = 0xff;
  DDRB &= ~(1 << ECHO);
  PORTD |= 0x00;
  EIMSK |= (1 << INT2);
  EICRA |= (1 << ISC20);
  TCNT1 = 0;
  sei();
}

void adc_init(){ //A/D omvandling
  ADMUX = 0;
  ADCSRA |= (1 << ADPS2) | (1 << ADPS1) | (0 << ADPS0);
  ADMUX |= (1 << REFS0) | (0 << MUX0);
  ADCSRB = 0x00;
  ADCSRA |= (1 << ADEN);
}

int main(void)
{
  m_init();
  u_init();
  adc_init();
  int foundlight = 0;
  int searching = 0;
  int motor = 0;
  int STOP = 0;
  /* Replace with your application code */
  while (1)
  {

```

```

ADCSRA |= (1 << ADSC);
light = ADC;
trigger();

    if(distance <= 20 && distance > 0){ //kollar ifall ett objekt är nära
        if(motor == 1){
            TCCR3A &= ~(1 << COM3A1);
        }
        else{
            TCCR0A &= ~(1 << COM0B1);
        }
    }
    STOP = 1;
}
else{
    STOP = 0;
    if(motor == 1){
        TCCR3A |= (1 << COM3A1);
    }
    else{
        TCCR0A |= (1 << COM0B1);
    }
}
if(light < 300 && foundlight == 0){ //motorerna ska vara avstängda från början
    TCCR3A &= ~(1 << COM3A1);
    TCCR0A &= ~(1 << COM0B1);
}

if(light > 300 && foundlight == 0 && STOP == 0){ //hittar den ljus ska den starta ena
motorn
    foundlight = 1;
    motor = 1;
    TCCR3A |= (1 << COM3A1);
}
if(light > 300 && STOP == 0){
    searching = 0;
}
if(foundlight == 1 && light < 300 && searching == 0 && STOP == 0){ //byt motor ifall
ljuset minskat
    searching = 1;
    motor *= -1;
    TCCR3A ^= (1 << COM3A1);
    TCCR0A ^= (1 << COM0B1);
}
    _delay_ms(250);
}
}

```

## 6 Referenser

- Pololu .50:1 Micro Metal Gearmotor HP. Åtkomstdatum 2023.05.02.  
<https://www.pololu.com/product/3051>
- ElecFreaks. HC-SR04 Ultrasonic Sensor User Manual. Åtkomstdatum 2023.05.02  
<https://www.electfreaks.com/11750.html>
- Texas Instruments.(2014).DRV8835 Dual Motor Driver.Datasheet. Åtkomstdatum 2023.05.02  
<https://www.ti.com/lit/ds/symlink/drv8835.pdf>
- Melexis. (2023). MLX90614 - Digital Plug & Play infraröd termometer TO-Can. Åtkomstdatum 2023.05.04  
<https://www.melexis.com/en/product/mlx90614/digital-plug-play-infrared-thermometer-to-can#>
- Texas Instruments. (2023). SN7407 Hex Buffer With High-Voltage Open-Collector Outputs. Åtkomstdatum 2023.05.04 .  
<https://www.ti.com/lit/ds/symlink/sn5407.pdf>
- Microchip Technology. (2023). ATmega1284P Microcontroller. Åtkomstdatum 2023.05.04  
<https://www.microchip.com/en-us/product/ATmega1284P>
- Panasonic Industry. AMS302 Light Sensor NaPiCa. Åtkomstdatum 2023.05.17  
<https://industrial.panasonic.com/ww/eol/pt/napica/models/AMS302>