| ParkeringsPeter  Projekt i Digitala System  Lunds Tekniska Högskola  Gruppmedlemmar: Måns Hansson, Oskar Karlsson, Wiktor Gullberg, Kevin Rost & Lucas Eriksson  Handledare: Bertil Lindvall & Lars Göran Larsson  Datum: 2022-05-23  Kurskod: EITA15 |
| --- |

# **Sammanfattning**

Rapporten beskriver ett projekt som ägde rum under läsperiod fyra i kursen Digitala System, kurskod EITA 15. Projektet gick ut på att utifrån tidigare kunskaper designa och konstruera en fungerande prototyp. Projektgruppen skulle bestå av 4-5 personer.

Projektgrupp 3 valde att konstruera ett fordon som med hjälp av en ultraljudssensor kunde mäta avstånd till föremål. När fordonet rörde sig mot ett föremål skulle en ljudsignal avges och en lysdiod blinka med en frekvens som ökade i takt med att fordonet kom närmare föremålet. Vid ett förutbestämt avstånd skulle fordonets motorer stängas av och ljudsignalen avges konstant samt att lysdioden skulle lysa konstant.

Diskussioner fördes om passande komponenter till fordonet och därefter gjordes ett kopplingsschema som sedan var mall till prototypen. Koppling av hårdvara gjordes och samspelet mellan de olika komponenterna styrdes med hjälp av programmering i språket C. Kunskaper från äldre laborationer stod till grund för programmeringen. En hemsida gjordes även där prototypen presenterades.

Den konstruerade prototypen uppfyllde alla önskade funktioner förutom att frekvensen för signal med hjälp av ljud och ljus inte var helt regelbunden.

# **Abstract**

This report is written to describe a project that took place during the fourth semester of Digitala System, EITA 15. The project was to come up with an idea, design and then develop a working prototype of that idea. This was done in groups of 4-5 people.

Group number 3 chose to develop a vehicle that could measure distance to objects in front of it with the help of an ultrasonic sensor. When approaching an object, the vehicle would let out a noise and flash a light with a frequency depending on the distance to the object. At a set distance, the vehicle would stop and let out a constant noise as well as shining the light in a constant manner.

First and foremost, the right components for the job were discussed and after a conclusion on the matter was reached a wiring diagram was designed. The wiring of the vehicle was done on a protoboard in conjunction with the programming of the components, which was made in AVR C. Knowledge from prior laboratory work in the course EITA 15 was of great importance for the code. Afterwards, a basic website that displayed the prototype, wiring diagram and code was programmed.

The results of the project were that the vehicle worked as intended with some inconsistencies.

**Innehållsförteckning**

[**Sammanfattning**](#_heading=h.gjdgxs) **2**

[**Abstract**](#_heading=h.xejbp8hmny10) **3**

[**1 Inledning**](#_heading=h.30j0zll) **4**

[**Syfte**](#_heading=h.1fob9te) **4**

[**Mål**](#_heading=h.3znysh7) **5**

[**Problem**](#_heading=h.2et92p0) **5**

[**Motivering av projekt**](#_heading=h.tyjcwt) **5**

[**Avgränsningar**](#_heading=h.3dy6vkm) **5**

[**2 Teknisk bakgrund**](#_heading=h.1t3h5sf) **6**

[2.1 Hårdvara](#_heading=h.lg0tbjnfv39n) 6

[2.2 Mjukvara](#_heading=h.4hhdenuwggyf) 7

[**3 Metod**](#_heading=h.4d34og8) **7**

[**4 Analys**](#_heading=h.17dp8vu) **8**

[**5 Resultat**](#_heading=h.3rdcrjn) **8**

[**6 Diskussion**](#_heading=h.eax14uqyueh8) **9**

[**8 Källförteckning**](#_heading=h.35nkun2) **10**

[**9 Appendix**](#_heading=h.1ksv4uv) **10**

# **1 Inledning**

Projektet utförs i kursen Digitala System och direktiv fås av Bertil Lindvall samt Lars Göran Larsson. Som grupp på 4 till 5 personer ska en prototyp konstrueras från egen idé med hjälp av programmering och koppling av hårdvara. Färdig prototyp ska redovisas i form av en tillgänglig hemsida, skriftlig rapport men även muntlig redovisning inför åskådare.

## **1.1 Syfte**

## Syftet med detta projekt är att konstruera en prototyp baserad på en mikroprocessor vid namn AT Mega 1284. Erfarenhet från samarbete inom grupp fås och även kunskap inom programmering samt koppling av hårdvara.

## **1.2 Mål**

## Idén som genomfördes var att konstruera ett körande fordon. Detta fordon ska med hjälp av en ultraljudssensor kunna mäta avstånd till föremål framför och därefter spela upp ljudsignal samt blinka med ökad frekvens då fordonet kommer närmare föremålet. Utifrån egengjord ritning ska sedan tillhörande komponenter kopplas och en ATMega 1284 ska programmeras i språket C för att utföra önskade funktioner.

## **1.3 Problem**

Färdig prototyp ska kunna göra följande i mån om tid:

1. Köra framåt och sedan stanna utifrån ett förutbestämt avstånd till föremål framför nås.
2. Ljudsignal ska skickas ut med ökad frekvens ju närmare prototypen kommer ett föremål framför. Vid tillräckligt litet avstånd ska ljudet höras konstant.
3. Lysdiod ska blinka med ökad frekvens ju närmare prototypen kommer ett föremål framför. Lysdioden ska även den lysa konstant när avståndet till föremålet framför är tillräckligt litet.

## **1.4 Motivering av projekt**

Fordon med avståndsmätare uppkom från de välkända parkeringssensorerna som finns på moderna bilar. Projektgruppen enades om att detta projekt var inom kompetensnivån och att det även kunde utföras inom den tidsperiod som projektet sträckte sig över.

## **1.5 Avgränsningar**

Nedan finns en lista på de saker som projektet inte behandlade:

1. Köra åt motsatt håll efter stopp nära ett föremål.
2. Svänga åt något håll.

# **2 Teknisk bakgrund**

Konstruktion av en fungerande prototyp har gjorts med nedanstående komponenter.

## **2.1 Hårdvara**

**ATmega1284**

Kärnan till prototypen är ATmega1284 vilket är programmerbar microcontroller. ATmega1284 har en 8-bitars kärna, 128 KB flash minne, 32 input/output pins och en maximal klockfrekvens på 20 MHz. Microcontrollern har även 32 tillgängliga register med 8 bitar vardera, 2 räknare bestående av 8 bitar samt 2 räknare bestående av 16 bitar. De nämnda specifikationerna är bara ett fåtal av de som ingår hos en ATmega1284. Spänningen som en ATmega1284 arbetar inom är mellan 1,8 V till 5,5 V.

**HC-SR04 Ultrasonic sensor**

HC-SR04 är en ultraljudssensor som skickar ut ljudvågor. En givare vid namn *Trig* skickar ut ljudvågor medans den andra givaren vid namn *Echo* läser av de som kommit tillbaka. Ofta används den till att mäta avstånd till olika föremål genom att beräkna hur lång tid det tar för ljudvågen att komma tillbaka. HC-SR04 skickar ut vågor inom ett område mellan 2 cm och 4 meter och har en arbetsspänning på 5 volt samt 15mA arbetsström.

**JTAG ICE**

JTAG är ett verktyg för felsökning och överskrivning av kod till en AVR microcontroller. Genom att koppla in en JTAG kan man köra aktiv felsökning samt göra förändringar i realtid när microcontrollern är igång. En JTAG ICE är kompatibel med alla 8-bitars mikrokontrollers som har ett JTAG interface vilket gör parkoppling tillgänglig.

**H-Brygga DRV833**

En H-brygga översätter spänning från en microcontroller till likström som sedan överförs till likströmsmotorer. DRV833 har en kapacitet att driva två likströmsmotorer med 1,5 A kontinuerligt vid en matning på 5V och en temperatur runt 25 grader celcius. Högsta likström från en DRV833 är 2A.

**Summer KGX1205C**

Prototypens ljudsignal genereras från en KGX1205C. Det är en magnetisk indikator, även kallas buzzer, som avger ett ljud på ca 90 decibel. En summer har ett arbetsområde mellan 3 och 7 V samt en maximal strömtillförsel på 30mA.

## **2.2 Mjukvara**

Mjukvaran som användes för att skriva kod var ett program vid namn Atmel Studio 7. Atmel Studio 7 är en plattform för programmering samt felsökning av en AVR microcontroller. De språk som Atmel studio 7 stödjer är C++, C och Assembly. Koden för prototypen skrevs i programspråket C och tillhörande hemsida skrevs i programmeringsspråket HTML5.

# **3 Metod**

Arbetsgången under projektet varierade mellan de olika arbetstillfällena. I början av projektets gång så tillbringades tiden med att planera prototypens funktion samt vilka komponenter som behövdes. Innan kod kunde börja skrivas för samarbetet mellan komponenterna och microcontrollern så behövdes ett preliminärt kopplingsschema ritas upp för prototypen. Programmet som användes för kopplingsschemat var Autodesk Eagle. Kopplingsschemat finns på sida 11. Alla fem medlemmar i gruppen hjälptes åt med kopplingsschemat. När schemat var ritat krävdes sedan ett godkännande av kursens handledare för att komponenterna skulle köpas in.

Efter planeringsfasen påbörjades faktasökning om de olika komponenternas funktioner. Detta gjordes med hjälp av fördjupning i olika datablad som finns på sida 11. Diskussioner om hur kod för de olika komponenterna skulle skrivas fördes sedan.

Felsökning skedde i form av processorn först kopplades ihop med matningsspänning och därefter kopplades enstaka komponenter till för att sedan testas med egenskriven kod. När samtliga komponenter var testade påbörjades en korrekt ihopkoppling utifrån kopplingsschemat. Lödning gjordes på de komponenter som skulle ta emot matningsspänning.

# **4 Analys**

Efter ihopkoppling av prototyp samt testkörning ansåg projektgruppen att en strömbrytare behövdes lägga till. En strömbrytare möjliggjorde att prototypen kunde startas efter behov då den annars hade varit igång oavbrutet. En knapp samt resistor kopplades sedan till microcontrollern med funktion som reset. Detta gjordes då prototypen hade tendens att vid tillfällen inte följa den förutbestämda hastigheten. Det största problemet som behövde lösas var att frekvensen som lysdioden och summern reagerade med var oregelbunden.

Ett oscilloskop användes då för att få en tydlig bild på hur bland annat spänningen varierade utifrån ultraljudssensorn. Kondensatorer blev ett diskuterat alternativ då en kondensator har som funktion att laddas upp och tillföra saknad spänning.

# **5 Resultat**

Projektet resulterade i en bandvagn som kan köra framåt, tjuta i en frekvens som ökar i och med att bilen närmar sig något samt stanna då den kommit inom ett räckhåll bestämt av användaren.

Ström till motorerna ges från H-bryggan och strömmens RMS värde, vilket är ett medelvärde av en varierande signal, är direkt beroende av värdena från ultraljudssensorn. Genom att basera ström-pulsens bredd på hur långt en räknare hinner räkna mellan varje signal från ultraljudssensorn, så går vagnens hastighet att kontrollera genom dess avstånd till objektet framför den. Detta åstadkoms genom att uppdatera pulsens bredd vid, av programmeraren satta, gränsvärden. Vid ökning av pulsbredden ges ett högre värde på RMS volt till motorerna och detta leder till en högre hastighet. Gäller även vice versa.

Ett annat ord för signalhantering på detta vis är PWM vilket är en förkortning för *Pulse Width Modulation*.

På brädet sitter även en summer kopplad i serie med en röd lysdiod till ATmegan. Genom att skicka en etta till denna port varje gång vi uppdaterar värdet på pulsen till motorn, kommer bägge dessa komponenter ge utslag i snabbare takt ju närmare ett objekt vagnen befinner sig. I nuläget existerar dock ett antal inkonsekvenser.

# 

# 

# 

# 

# **6 Diskussion**

Överlag anser vi projektet lyckat. Parkerings Peters funktion har realiserats med viss inkonsekvens, men dess syfte skiner klart igenom. Redan vid projektets start hade vi en relativt tydlig tanke med hur vi ville att den skulle fungera. Med tiden fick vi dock förändra vår slutvision av projektet, i huvudsak på grund av att tiden inte fanns. Projektet i sig har varit mycket lärorikt för oss eftersom det har inneburit ett arbetssätt vi inte tidigare stött på. För att kunna förstå oss på och lösa de problem som uppstod under projektets gång krävdes en förståelse både för komponenter och kod, vilket ledde till att en hel del tid lades åt att förstå sig på hur dessa fungerar och hur koden för dessa komponenter skrivs. Det som skilde projektet från tidigare arbeten var att vagnen bestod av ett flertal olika komponenter vilka krävde sin egen synvinkel i den mån att vi var tvungna att förstå både kod och funktion för varje individuell komponent för att sedan kunna få den att fungera tillsammans med övriga delar.

I nuläget existerar som tidigare nämnts ett antal inkonsekvenser. Dessa innefattar :

* Vid provkörning har emellanåt summern och lysdioden slutat fungera.
* En av motorerna får vissa gånger en för hög ström och bara kör full fart framåt
* Frekvensen summern tjuter i, lysdioden blinkar i och pulsen till motorn uppdateras i takt med är inte konsekvent.

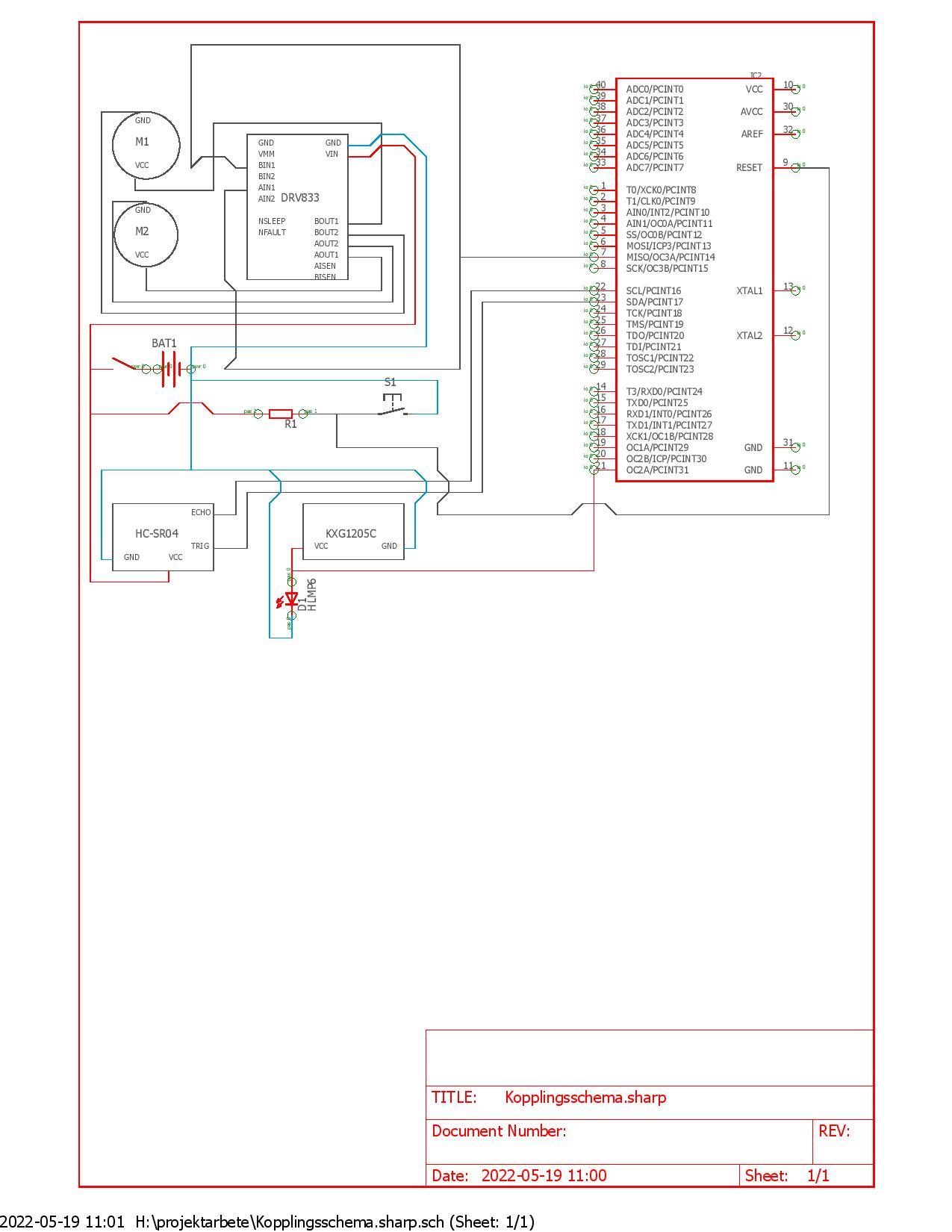
I dagsläget är det mycket svårt att säga vad dessa inkonsekvenser beror på. Koden är på inget sätt felfri, men efter felsökning konstaterar vi att problemet mest troligen ligger i hårdvaran. Vid mätning med oscilloskop går en viss störning av signalerna att se. Framtida vidareutveckling av projektet hade kunnat vara att försöka åtgärda dessa störningar exempelvis genom att koppla kondensatorer parallellt med motorerna eller processorn för att få jämnare signal.

# **8 Källförteckning**

1. <https://www.tme.eu/Document/ced9f014eee9243147c3a9cca6e82c64/ATmega1284-DTE.pdf> (atmega)
2. <https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/> (ultrasonic)
3. <https://www.eit.lth.se/fileadmin/eit/courses/datablad/Lawicell/drv8833_Dual_motor_drive.pdf> (H-bryggan)
4. <https://se.farnell.com/kingstate/kxg1205c/buzzer-magnetic-pin-85db-5v/dp/2215080?CMP=GRHB-SF-BNL> (summer)

# **9 Appendix**

Nedan visas kopplingsschemat för den konstruerade prototypen.



Kod skriven för prototypen finns nedan.

#define F\_CPU 16000000UL

#include <avr/io.h>

#define \_\_DELAY\_BACKWARD\_COMPATIBLE\_\_

#include <util/delay.h>

// för ultraljudssensorn

uint16\_t echo\_recive();

void beep();

//för pwm3

void timer3\_init();

void set\_pulse3(uint16\_t);

void set\_period3(uint16\_t);

int main(void)

{

DDRD |= 0b11111111; // lampan/zoomer

DDRC |= (1<<DDC1); // trig

timer3\_init();

set\_period3(260); // 33 ms

set\_pulse3(25); // 16 ms

while(1){

//för zommer och ultraljudssensorn

echo\_recive();

set\_pulse3(25);

if (TCNT1 < 35 && TCNT1 > 4){

set\_pulse3(25);

beep();

} else if ((TCNT1 <= 4)){

set\_pulse3(0); // stopppar motorn

PORTD = (1<< PORTD7);

}else{

PORTD = (0<<PORTD7);

}

}

}

// För ultraljudssensorn och zommer

uint16\_t echo\_recive(){

\_delay\_ms(60);

TCNT1 = 0;

PORTC |= (1<<PORTC1); // c1 trig trig

\_delay\_us(10);

PORTC &= ~(1<<PORTC1);

while(!(PINC & (1<<PINC0))){ // c0 echo

}

TCCR1B |= (1<<CS12)|(1<<CS10);

while(PINC & (1<<PINC0)){

}

TCCR1B = 0;

return (TCNT1);

}

void beep(){ //Dubbekollad, funkar

PORTD = (1 << PORTD7); // zoomer

\_delay\_ms(TCNT1\*10);

PORTD = (0 << PORTD7);

\_delay\_ms(TCNT1\*10);

}

// För PWM timer 3

void timer3\_init(){

TCCR3A |= (1<< COM3A1) | (1<< WGM31);

TCCR3B |= (1<<WGM33) | (1<<WGM32) | (1<<CS30) | (1<<CS32);

DDRB |= (1<<DDRB6);

}

void set\_pulse3(uint16\_t ocr){

OCR3A = ocr;

}

void set\_period3(uint16\_t icr){

ICR3 = icr;

}