

Författare:

Anton Celander

Pavel Hamawand

Melker Jarl

Oscar Hessler

Lucas Hviid

Samuel Högbom Aronson

Handledare:

Bertil Lindvall

Radiostyrd bil

Digitala System - projekt

EITA 15

Vårtermin 2023

Lunds tekniska högskola

Sammanfattning

Denna rapport fokuserar på konstruktionen av en radiostyrd bil med syftet att utforska och tillämpa principer inom mekanisk konstruktion, elektronik och fjärrstyrning. Rapporten beskriver även det arbete som genomfördes av gruppen, inklusive lärdomar, utmaningar och potentiella förbättringar. Resultaten presenteras och diskuteras även de ingående i rapporten.

Målet med projektet var att skapa en fungerande radiobil med grundläggande manövreringsförmåga som kunde styras smidigt genom en fjärrkontroll. Genom att utgå från kopplingsdäcken som en central komponent definierades specifikationer och krav för designen. Målet var att skapa en radiostyrd bil som möjliggjorde smidig körning utan att hindra rörligheten, några krav estetiska fanns alltså inte. För fjärrkontrollen fanns inte heller några estetiska krav och målet var att den skulle vara funktionell och enkel. Centrala delar i konstruktionen av bilen inkluderade programmering av ett ATmega 1284-chip och användning av radio-sändare och mottagare för att skapa ett digitalt system. Likaså byggdes fjärrkontrollen som ett komplement till bilen med samma komponenter i fokus.

Resultatet av projektet blev en enkel, men fungerande radiostyrd bil. Med elmotorer, däck som andra kopplingskomponenter, inklusive ett ATmega 1284-chip, strömkälla och en radio-sändare/mottagare konfigurerad som mottagare, kunde en fungerande radiobil skapas. Tillsammans med en fungerande fjärrkontroll, byggd på ett liknande kopplingsdäck med en joystick som en av de viktigaste delarna.

Genom denna rapport, som tydligt beskriver processen för att bygga vår radiobil, kontrollen och implementeringen av olika tekniska byggstenar, syftar rapporten även till att bidra till kunskapen inom området och skapa en grund för framtida liknande projekt och undersökningar. Den betonar också de lärdomar och den förståelse för komponenter och koncept som erhöles genom genomförandet av projektet.

Nyckelord

ATmega 1284, digitala system, radiostyrd bil, kretsschema

Abstract

This report focuses on the construction of a radio-controlled car with the purpose of exploring and applying principles in mechanical engineering, electronics, and remote control systems. The report also details the work conducted by the team, including lessons learned, challenges faced, and potential improvements. The results are presented and thoroughly discussed within the report.

The objective of this project was to develop a functional radio-controlled car with basic maneuverability that could be smoothly operated through a remote control. By considering the breadboard as a central component, specifications and design requirements were established. The goal was to create a RC car that enabled smooth driving without hindering mobility or adding unnecessary weight or bulk, no aesthetic goal was created. The same aesthetic requirements were imposed on the remote control whose only aim was to be functional and simple. Key elements in the construction of the car included programming an ATmega 1284 chip and utilizing radio transmitters and receivers to establish a digital system. Similarly, the remote control was built as a complementary device, focusing on the same components.

The result of the project was a simple yet fully functional radio-controlled car. With electric motors, and tires in addition to breadboard components, including an ATmega 1284 chip, power source, and a receiver-configured radio transmitter/receiver, a functional radio-controlled car was successfully created. Alongside it, a functional remote control was constructed, employing a similar breadboard setup and featuring a joystick as one of the primary controls.

This report provides a clear account of the process involved in constructing our radio-controlled car, designing the control system, and implementing various technical building blocks. Furthermore, it aims to contribute to the existing body of knowledge in the field and establish a foundation for future similar projects and investigations. It also

emphasizes the lessons learned and understanding gained from the utilization of components and concepts throughout the course of this project.

Innehållsförteckning

Innehållsförteckning	4
1. Inledning	5
1.1 Syfte	5
1.2 Målformulering	5
2. Teknisk bakgrund	6
3. Metod	7
4. Resultat	8
5. Slutsats	9
5.1 Utvecklingsmöjligheter	9
6. Källförteckning	10
7. Appendix	11
7.1 Kretsscheman	11
7.1.1 Kretsschema kontrollen	11
7.1.2 Kretsschema bilen	12
7.2 Bilkod	12
7.3 Kontrollkod	13

1. Inledning

Detta projekt är en del av kursen Digitala system. Projektet har som mål att bygga en prototyp med minst en mikroprocessorn ATmega 12-84. Denna prototypen skulle med hjälp av handledning, gemensamma kunskaper och en del research, vara klar efter veckors arbete. Även skulle en skriftlig rapport skrivas och en hemsida skulle skapas samt att en muntlig redovisning skulle genomföras.

1.1 Syfte

Syftet med projektet var att med hjälp av de kunskaperna vi samlat på oss under kursens konstruera en prototyp med minst en ATmega-1284 mikroprocessor som skulle uppfylla en utvald funktion som valdes i gruppen. Även har det varit viktigt med tydliga mål för att hinna klart med projektet samt att hålla en god kommunikation, speciellt under tillfällen där arbetet varit uppdelat mellan gruppen.

1.2 Målformulering

Gruppen som var på sex personer valde att skapa två prototyper: en radiostyrd bil samt en kontroll. Båda enheterna skall ha en ATmega 12-84 mikroprocessor som skall kommunicera med varandra för att kontrollenheten ska kunna ge kommandon till bilen. Därav hade vi som grupp målen:

- Bilen skall kunna köra framåt
- Bilen skall kunna svänga höger och vänster
- Kontrollen ska ge kommandon
- Trådlös kommunikation mellan prototyperna

2. Teknisk bakgrund

För att implementera den radiostyrda bilen använder vi oss av en Atmega 1284 mikroprocessor. Mikroprocessorn fungerar som hjärnan i systemet och styr de olika komponenterna i bilen såsom motorer, styrning och kommunikation med fjärrkontrollen.

För skapandet av en trådlös förbindelse mellan fjärrkontrollen och den radiostyrda bilen används en radiomodul, I vårt fall är det en Parallax 433 transceiver som används både som en transmitter samt en reciver beroende om vi skickar en etta eller en nolla till en pin på modulen. Denna modul tar emot kommandon från en joystick på styrenheten och skickar sedan dessa kommandon till samma modul på den radiostyrda bilen som sedan skickar denna information till en H-brygga och vidare ut till motorerna

H-bryggan används för att lösa problemet som framkommer när man försöker styra två komponenter med en output. H-bryggan tar emot informationen och skickar sedan ut rätt kommando till rätt motor för att uppnå önskat resultat.

För att programmera beteendet hos den radiostyrda bilen använder vi oss av C i IDE:n Atmel studio. Genom att använda C kan olika funktioner och beteenden implementeras, till exempel att köra framåt, bakåt eller svänga beroende på olika kommandon som skickas från styrenheten till bilen.

Resultatet kan vara en fungerande radiostyrd bil som styrs trådlöst med hjälp av en fjärrkontroll och i sin tur kan utföra kommandon utan större delays eller komplikationer.

3. Metod

Projektet utfördes stegvis i de stegen angivna nedan:

1. Projektidén utvecklades av författarna baserat på ämnesområdet. Efter noggrant övervägande och diskussion presenterades projektidén för kurshandledaren. Efter att ha tagit emot feedback och genomfört vissa modifieringar, godkändes projektidén.
2. Efter godkännandet av projektidén inleddes arbetet med att ta fram ett detaljerad kopplingsschema för prototypen. Kopplingsschemat skulle visa hur olika komponenter skulle kopplas samman och interagera med varandra. Detta inkluderade mikrokontrollern och andra elektroniska komponenter. Kopplingsschemat granskades sedan av handledaren för att säkerställa att det uppfyllde kraven och specifikationerna.
3. Efter att kopplingsschemat hade godkänts påbörjades monteringen av prototypen. Detta inkluderade användning av lödning och trådbindning för att ansluta komponenterna korrekt. Vissa komponenter krävde också borring för att skapa passande monteringshål. Under denna fas var det viktigt att noggrant följa instruktionerna och säkerställa att komponenterna monterades på rätt sätt för att undvika eventuella fel eller problem senare.
4. När komponenterna monterades ihop med mikrokontrollern, genomfördes tester för att verifiera deras funktionalitet och korrekthet. Detta gjordes genom att programmera mikrokontrollern med testkod och övervaka komponenterna med hjälp av spänningsmätare. Genom att utföra dessa tester kunde man säkerställa att varje komponent fungerade som avsett och att de kunde kommunicera korrekt med mikrokontrollern.
5. Efter framgångsrika tester av komponenterna påbörjades kodskrivningen. I denna fas implementerades olika funktioner och logik för att hantera data från diverse komponenter. Denna kodskrivningsfas krävde noggrannhet och noggrant testande för att säkerställa att prototypen fungerade korrekt och uppfyllde de förväntade kraven.
6. Slutligen, när koden var färdig, genomfördes en sista granskning och finslipning. Eventuella buggar eller felaktigheter identifierades och korrigerades. Parametrarna finjusterades för att säkerställa att prototypen fungerade smidigt och att önskade beteenden uppnåddes. Den slutgiltiga koden var nu klar för att implementeras och användas i prototypen.

4. Resultat

Vårt projekt består av en styrenhet och en bilenhet som tillsammans bildar en kontrollerbar bil. Bilen kontrolleras via styrenheten med hjälp av en tvådimensionell joystick som styr ett hjul med joystickens X-led och det andra hjulet i joystickens Y-led. Detta medför funktioner såsom körning framåt, bakåt och snäva svängar åt både höger och vänster. Vår bil är med andra ord optimerad för att vara agil och ha god möjlighet att ta sig fram på tekniskt utmanande platser. Bilen har en teoretisk topphastighet på 5 km/h men då kommunikationen mellan enheterna är trådbaserad körs bilen långsammare då det annars vore opraktiskt för bilen.

5. Slutsats

Projektet uppnådde delvis våra utsatta mål där bilen kunde svänga åt vänster och höger samt köra framåt och bakåt. Däremot kunde vi inte lösa den trådlösa dataöverföringen mellan bilen och kontrollen, då transmittern/receivern inte fungerade så som databladet hänvisade.

Därefter slutade vi felsöka på handledarens direktiv och istället blev det en trådbaserad kommunikation istället.

5.1 Utvecklingsmöjligheter

Det finns en del utvecklingsområden på vårt projekt. Dels att man kan vidareutveckla konstruktionen på bilen. Även kan man implementera funktioner som exempelvis att kunna köra bilen i olika hastigheter och svänga och köra framåt/bakåt samtidigt. Även hade man kunnat addera lampor som tänds när det blir tillräckligt mörkt även en tuta som aktiveras när ett objekt är framför bilen. Även vore det intressant att få använda transmittar/receiver som fungerar enligt databladet, för att kunna köra bilen längre bort ifrån kontrollen.

6. Källförteckning

ATmega-1284:

<https://www.eit.lth.se/fileadmin/eit/courses/datablad/Processors/ATmega1284.pdf>

H-Brygga:

https://www.eit.lth.se/fileadmin/eit/courses/datablad/Lawicell/drv8833_Dual_motor_drive.pdf

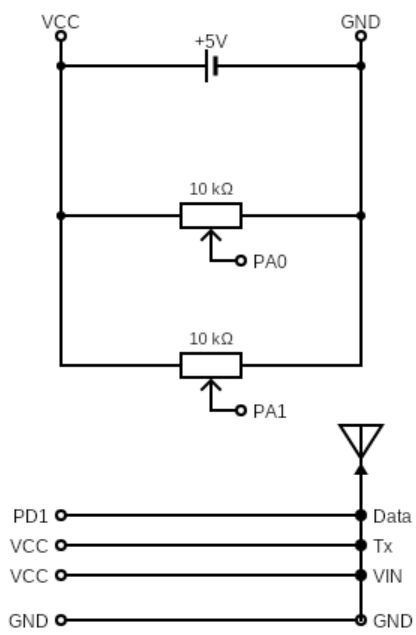
Parallax 433 transceiver:

https://www.mouser.com/datasheet/2/321/parallax_27982-433-mhz-rf-transceiver-documentation-1197467.pdf

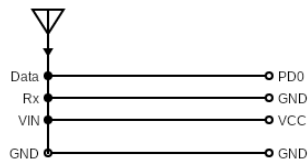
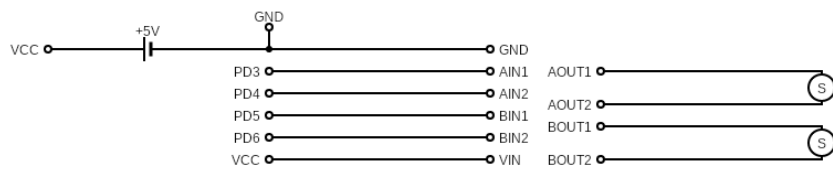
7. Appendix

7.1 Krettscheman

7.1.1 Krettschema kontrollen



7.1.2 Kretsschema bilen



7.2 Bilkod

```
/*
 * bil.c
 *
 * Created: 2023-04-30 15:17:59
 * Author : Samuel
 */

#include <avr/io.h> // IO definitions

uint8_t direction;

void init(){
    // pb3, pb4, pb6, pb7 pointing out.
    DDRB |= (1 << DDRB3) | (1 << DDRB4) | (1 << DDRB6) | (1 << DDRB7);
    // Setting baud rate to 9600 Baud.
    UBRR0 = 6;
    // turning on receiver 0 (rx0).
    UCSR0B |= (1 << RXEN0);
}
```

```

uint8_t receive(){
    // wait for data
    while(!(UCSR0A & (1<<RXC0)));
    // return data.
    return UDR0;
}

void move(uint8_t data){
    if(data == 1){ /* fram */
        PORTB |= (0 << PINB3) | (0 << PINB4) | (1 << PINB6) | (1 << PINB7);
    }
    else if(data == 2){ /* hoger */
        PORTB |= (1 << PINB3) | (0 << PINB4) | (0 << PINB6) | (0 << PINB7);
    }
    else if(data == 3){ /* bak */
        PORTB |= (1 << PINB3) | (1 << PINB4) | (0 << PINB6) | (0 << PINB7);
    }
    else if(data == 4){ /* vanster */
        PORTB |= (0 << PINB3) | (1 << PINB4) | (0 << PINB6) | (0 << PINB7);
    }

    else PORTB = 0x00;
}

int main(void)
{
    init();
    while (1)
    {
        direction = receive();
        move(direction);
    }
}

```

7.3 Kontrollkod

```

/*
 * REMOTE.c
 *
 * Created: 2023-04-29 15:17:59
 * Author : Samuel

```

```

*/
#define F_CPU 8000000UL // defining cpu speed for delay function
#include <avr/io.h> // IO definitions
#include <util/delay.h> // including delay functions

uint16_t adc_value_x;
uint16_t adc_value_y;
uint8_t direction = 0;

void init(){
    ADCSRA |= (1 << ADEN); /* Enable ADC */
    DDRA = 0x0; /* PORTDN INPUT MODE */
    ADMUX |= ((1 << REFS0) | (0 << REFS1)); // AVcc as reference voltage.

    TCCR1C |= (1 << CS02) | (1 << CS00);
    TCNT0 = 0x00;

    UBRR0 = 6; /* Set baud rate to 9600 BAUD */
    UCSR0B |= (1 << TXEN0); // enable transceiver 0.
}

uint16_t adc_read(uint8_t channel){
    ADMUX = (ADMUX & 0xF0) | (channel & 0x0F); // setting adc channel
    ADCSRA |= (1 << ADSC); // starting conversion
    while ( (ADCSRA & 0b01000000) != 0){} // wait for conversion
    return ADC; // returning conversion
}

void transmit(uint8_t data){
    /* Wait for empty transmit buffer */
    while(!(UCSR0A & (1 << UDRE0))){}
    /* send data */
    UDR0 = data;
}

int main(void)
{
    init();

    while (1)
    {
        adc_value_x = adc_read(0);
        adc_value_y = adc_read(1);
        // calculate direction
        if(adc_value_x > 1000 && adc_value_y < 1000){
            direction = 1; // Forward
        }
    }
}

```

```
else if(adc_value_y > 1000 && adc_value_x < 1000){
    direction = 2; // left
}
else if(adc_value_x < 5 && adc_value_y > 10){
    direction = 3; // Back
}
else if(adc_value_y < 5 && adc_value_x > 10){
    direction = 4; // right
}
else direction = 0; // Stand still

_delay_ms(500); // delay for timing
transmit(direction);
}
}
```