

# Radiostyrd bil

---

## EITF11 – Digitala Projekt

**Thomas Bergsten & Marcus Halbauer**

**2014-05-19**

This report is on a project in the course EITF11 - Digital Projects, taken at LTH in the spring of 2014. The aim of the course is to give students an introduction to electronics development by building an electronic unit and illustrate the process of an industrial development project. By deciding to construct a car being controlled through wireless signals by an external remote control, the group acquired knowledge in fields that the group members had very limited experience from. With data sheets, C-programming and an instructor providing the components needed, the result was a functioning RC-car despite several setbacks. However, the wireless communication did not always work due to a noisy environment.

## Innehållsförteckning

Inledning.....	4
Målsättning .....	5
Teori.....	6
Hårdvara.....	6
Mikroprocessor – ATmega16.....	6
Transceiver – Parallax 433 MHz RF Transmitter .....	6
Motorer .....	6
H-brygga – Dual H-bridge Motor Driver DRV8833 .....	6
Transistorer – LP3855 och MCP1827 .....	6
Batteri.....	6
Mjukvara .....	6
AVR Studio 4 .....	6
Power Logic.....	6
USART-register.....	7
Metod .....	8
Teoriinläring.....	8
Kretsschema och kravspecifikation.....	8
Koppling.....	8
Programmering.....	8
Resultat .....	10
Diskussion.....	11
Svårigheter .....	11
Slutsats .....	11
Referenser .....	13
Skriftliga:.....	13
Muntliga: .....	13
Bilagor .....	14
Bilaga 1 – Kravspecifikation .....	14
Mål .....	14
Funktionella krav.....	14
Kvalitetskrav .....	14
Komponenter.....	14
Bilaga 2 – Kopplingsscheman .....	15

Bilen.....	15
Fjärrkontrollen.....	15
Bilaga 3 – Programkod .....	16
Bilen.....	16
Fjärrkontrollen .....	17
Bilaga 4 – Bilder .....	18

## Inledning

Kursen Digitala Projekt på Lund Tekniska Högskola avser att ge kunskap inom elektroteknik och illustrera ett industriellt utvecklingsarbete. För studenter på Industriell Ekonomi är det även, som så många andra kurser, ett tillfälle att bredda sin kunskap då det för många var den första gången de arbetat med något liknande. Detta ledde såklart till problem och motgångar, vilket i sin tur ledde till ökad problemlösningsförmåga och många lärdomar.

För att tillämpa de kunskaper som lärdes ut i den teoretiska delen i början av kursen valde projektgruppen att bygga en radiostyrd bil. Detta skulle innebära att, förutom ihopkoppling av komponenter och programmering av mikroprocessorer, gruppmedlemmarna fick fördjupa sig inom trådlös överföring av data.

## Målsättning

Då ingen av gruppens deltagare hade någon erfarenhet av liknande arbete sedan tidigare sattes mål utifrån vad som troddes att gruppen skulle kunna klara av. Det viktigaste var att de simplaste kraven för att få kalla konstruktionen en bil skulle uppnås. Det innebär därmed att den skulle kunna röra sig framåt, bakåt och även kunna svänga åt båda höger och vänster. När detta uppnåtts var målet att kunna styra bilen via en fjärrkontroll, utan någon som helst koppling med sladdar mellan. Skulle detta uppnås i tid fanns diverse idéer om utökade funktioner på bilen, exempelvis blinkers eller tuta. För kravspecifikation, se **bilaga 1**.

För att uppnå dessa mål skulle två separata konstruktioner behöva skapas med varsin mikroprocessor och transceiver.

# Teori

## Hårdvara

### Mikroprocessor – ATmega16

Mikroprocessor med 40 pinnar och minne på 16kB för programmering. Programmering sker via programspråket C och JTAG kan användas för debugging. I mikroprocessorn finns ett antal register som utför olika funktioner. USART-registret för synkron/asynkron trådlös överföring är ett av dessa. Se mer information om USART under "Mjukvara – USART-register".

### Transceiver – Parallax 433 MHz RF Transmitter

Sändare och mottagare för trådlös överföring av signaler. Den har två pinnar för att erhålla driftspänning på 2.7 – 3.6 V, en pinne för input eller output av data, en pinne som avgör om komponenten ska sända eller ta emot (beroende på spänning) och två pinnar som ej användes i detta projekt. Av de två sista lämnades den ena (PDN) öppen för att ej gå in i Power Down Mode och den andra (RSSI) för att den indikerar signalstyrkan, vilket inte behövdes.

### Motorer

Två motorer användes för att driva fram bilen med hjälp av larvfötter och hjul. Dessa motorer jobbar med en driftspänning på 6 Volt.

### H-brygga – Dual H-bridge Motor Driver DRV8833

Komponent som fungerar som två H-bryggor i en. Dessa styr två stycken motorer vars riktningar beror på vilken input bryggan får från mikroprocessorn. Bryggan levererar även ström till motorerna då dessa går som bäst med en högre spänning än den som kommer från mikroprocessorn. De båda interna bryggorna har var för sig två input-pinnar, två output-pinnar och pinne för jordning.

### Transistorer – LP3855 och MCP1827

Två olika typer av transistorer användes. Den ena för att erhålla en spänning på 5V åt bland annat mikroprocessorn samt den andra för att erhålla en spänning på 3V åt transceivern.

### Batteri

För att inte utgöra någon begränsning vad avser rörlighet på bilen skulle ett batteri bestående av fem seriekopplade 1,2V batteri, vilket gav en total spänning på 6V, användas som spänningskälla till bilen.

## Mjukvara

### AVR Studio 4

Mjukvara för programmering av mikroprocessorerna. Programspråk som används är C. Innehåller även möjlighet för att kompilera programmerad kod samt undersöka denna för defekter (debugging). För komplett kod till den radiostyrda bilen, se **bilaga 3**.

### Power Logic

Detta program används för att rita upp kopplingscheman. Innan bygget av bilen påbörjades gjordes ett komplett kopplingschema för att se vilka komponenter som skulle behövas. För kopplingschema, se **bilaga 2**.

## USART-register

USART står för *Universal Synchronous/Asynchronous Receiver/Transmitter* och är ett register i mikroprocessorn ATmega16. Registret kan användas för att på ett enkelt sätt sända och ta emot data trådlöst mellan två olika mikroprocessorer.

Det finns två väsentliga skillnader mellan synkron (*synchronous*) och asynkron (*asynchronous*) trådlös överföring. I en synkron överföring måste de båda mikroprocessorerna vara synkroniserade vad avser sin frekvens. Asynkron överföring kräver inte denna förutsättning.

Om USART-används sker den trådlösa överföringen genom att sändaren först skickar iväg en start bit för att initiera kommunikationen. Därefter skickas en ram på minimalt fem och maximalt nio bitar innehållande den viktiga informationen. För att markera slutet på en ram kan en eller två stopp-bitar användas. Därefter kan en ny ram skickas och tas emot.

Vidare finns det många ytterligare möjligheter med USART, varav två av dessa är att en mikroprocessor ska agera som både mottagare och sändare samt att paritetsbitar kan användas för att kontrollera om data har mottagits korrekt och inte är korrupt.

## Metod

### Teoriinläring

Den första halvan av perioden som kursen lästes bestod av föreläsningar och laborationer. Detta för att ge en grundläggande kunskap inom elektroteknik samt en introduktion till de datorprogram och komponenter som skulle användas. Detta var dock inte heltäckande och inläringen fortsatte kontinuerligt under resterande faser i konstruktionsbyggandet, bland annat med hjälp av diverse datablad som fanns tillhanda på kurshemsidan.

Redan under denna fas påbörjades idégenereringen av vad projektgruppen ville konstruera, vilket efter många olika idéer landade på en radiostyrd bil.

### Kretsschema och kravspecifikation

När det hade beslutats att en radiostyrd bil skulle konstrueras krävdes det att ett kretsschema och en kravspecifikation skulle visas upp för handledaren innan de komponenter och verktyg som behövdes tillhandahölls.

En kravspecifikation togs fram med de funktionella krav och kvalitetskrav som skulle uppnås. De viktigaste funktionella kraven var att de elementära funktionerna hos en bil skulle uppfyllas, det vill säga att kunna köra framåt, bakåt och svänga höger och vänster. Dessa kommandon skulle bilen genomföra genom att använda knapparna på fjärrkontrollen. För kravspecifikation till den radiostyrda bilen, se **bilaga 1**.

Med hjälp av mjukvaran Power Logic kunde ett kretsschema ritas upp och det var här som de teoretiska kunskaper som tagits del av i föregående fas började spela in. Med egen informationssökning samt hjälp av handledare och datablad kunde komponenter hittas som uppfyllde funktioner som skulle behövas, såsom H-brygga och transistorer. Dessa komponenter lades in och kopplades i Power Logic vilket resulterade i två stycken kopplingsscheman, ett för bilen och ett för fjärrkontrollen. För kopplingsschema till både bilen och fjärrkontrollen, se **bilaga 2**.

De komponenter som fanns i kopplingsschemat lades även till i kravspecifikationen och lämnades därefter in till handledare för godkännande.

### Koppling

Med kopplingsscheman och kravspecifikation godkända tilldelades en verktygslåda och de komponenter som nämnts i kravspecifikationen till gruppen för att kunna påbörja byggandet av bilen.

Innan komponenter kopplades in i kretskorten kontrollerades de flesta så att de fungerade, exempelvis motorerna. Det kontrollerades även att det var rätt spänning på rätt ställen så att exempelvis mikroprocessorn inte skulle få för hög spänning. När alla komponenter hade virats alternativt löts ihop skruvades bilens kretskort fast i en redan färdigställd konstruktion med hjul och larvfötter där motorerna satt fast.

### Programmering

I denna fas skulle mikroprocessorerna hos bilen och fjärrkontrollen programmeras så att bilens motorer skulle utföra de rörelser som de skulle när knapparna på fjärrkontrollen trycktes ned. Detta



skulle kodas i AVR Studio 4 med C-programmering. Här krävdes en hel del egen inläring då ingen i projektgruppen tidigare hade programmerat i detta språk.

Från tidigare projektrapporter med liknande uppgift insågs det att det var kommunikationen mellan sändaren och mottagaren som oftast var det mest problematiska. Därmed lades fokus i början på att se till att rätt signaler skickades från fjärrkontrollens mikroprocessor till sändaren och att bilen reagerade som den skulle när samma signaler skickades till bilens mikroprocessor. Detta testades genom att styra bilen direkt från datorn och goda resultat erhålls i ett tidigt stadie. För programmeringskod, se **bilaga 3**.

Efter konsultation med vår handledare beslöt gruppen för att använda sig av USART-registret som finns tillgänglig i mikroprocessorn och som möjliggör, att på ett enkelt sätt, etablera en trådlös kommunikation mellan två mikroprocessorer. Asynkron kommunikation skulle användas. Efter att ha läst på mycket om USART i databladet för mikroprocessorn kunde programmeringen för sändaren och mottagaren slutföras med inte alltför komplicerad kod.

## Resultat

Resultatet av konstruktionen blev till slut en bil som kunde röra sig framåt, bakåt och svänga. Däremot hade den begränsad rörlighet då det tänkta batteriet hade bristande kvalitéer och en fast spänningskälla fick användas. Sändningen av signaler mellan sändare och mottagare fungerade oftast men på grund av bristande kvalitet på komponenterna samt brusig miljö så fungerade det ibland mindre bra. De elementära funktionella kraven på själva bilen som angetts i kravspecifikationen uppfylldes alltså. Då det under den sista tiden innan deadline helt fokuserades på att få rätt på signalerna mellan sändare och mottagare så prioriterades de funktioner som skulle införas i mån av tid bort.

För bild av konstruktionen, se **bilaga 4**.

## Diskussion

### Svårigheter

De flesta problem som uppstod var på grund av den bristande kunskapen hos gruppmedlemmarna. Redan vid kopplingsschemat var det mycket som var oklart. Därmed krävdes det flertalet försök innan schemat blev godkänt, vilket var nödvändigt för att undvika fel vid byggandet av bilen.

Trots det färdigställda kopplingsschemat förekom det onödiga problem vid byggandet. Exempelvis användes vid flera ställen för tunna trådar för spänning på 6 Volt vilket ledde till att dessa fick bytas ut. Feltolkningar av komponenters pinnars funktioner eller hur de skulle kopplas ledde också till att tid ödslades på att vira upp tråd och göra om på rätt sätt. Även misstag som för dålig lödning gjorde att trådar lossnade och fick lödas fast på nytt.

Under programmeringsfasen av projektet uppstod även många problem. Efter att ha hämtat inspiration från tidigare projekt som också byggt radiostyrda bilar hade gruppen tänkt använda en viss typ av trådlös kommunikation. Denna innebar att fjärrkontrollen skulle skicka iväg ett antal ettor och nollor beroende på vilken knapp som tryckts in på fjärrkontrollen. Bilens mottagare skulle uppfatta detta och reagera med rätt kommando inskickad till motorerna. Detta programmeringskoncept visade sig dock vara mer komplext än gruppen tänkt sig från början.

Gruppens handledare gav oss därmed rådet att prova USART-registret i mikroprocessorn. Däremot var denna metod obeprövad av handledaren och han varnade även oss att det kanske inte skulle fungera överhuvudtaget.

I ett initialt skede av USART-programmeringen behövdes en extern kristall till varsin mikroprocessor användas för att få dessa att erhålla frekvensen 8 MHz med högre precision än den interna klockan i mikroprocessorn. Detta skulle förbättra kommunikationen mellan de två mikroprocessorerna. Det uppstod däremot problem i AVR Studio 4 vid användning av extern kristall. Efter mycket felsökande av både gruppen men främst handledaren bestämdes i samförstånd att den interna, lite mer oprecisa klockan i mikroprocessorerna skulle användas.

Vidare uppstod problem vid inställningen av USART-registret. Problemet var att rätt pinnar i registret inte hade aktiverats efter att ha skrivit in koden för valda inställningar. Efter rådgivning och felsökande konstaterades av handledaren att pinnarna aktiveras först när mikroprocessorn tar emot/sänder information och påvisar således inte inställningarna innan detta har hänt.

På grund av brus i den omgivande arbetsmiljön, föråldrad och bristande kvalitet i hårdvara (främst transeivern) samt att extern kristall inte kunde användas var den upprättade trådlösa kommunikationen mellan bilen och dess fjärrkontroll dålig. Detta innebar fördröjningar av kommandon samt att den trådlösa kommunikationen ibland helt upphörde.

Vidare var kvalitén på batteriet som skulle användas till bilen så pass bristande att den inte kunde driva bilen H-brygga, och därmed motorerna. Gruppen beslutade sig därför om att använda en icke-mobil spänningskälla med högre tillförlitlighet.

### Slutsats

Det primära målet var att kunna styra en bil framåt, bakåt, höger och vänster trådlöst med en fjärrkontroll. Detta mål uppnåddes även om den trådlösa signalöverföringen stördes av den brusiga

miljön. Således är gruppmedlemmarna mer än nöjda med projektets slutresultat. Det hade såklart varit önskvärt att ha kunnat implementera ytterligare funktioner för att uppnå de sekundära målen i kravspecifikationen. Det råder ingen tvekan om att detta hade klarats av om mer tid hade funnits tillhanda då den relevanta kunskapen som i början av projektet var nästintill obefintlig nu fanns hos gruppen.

Förutom nyvunnen kunskap inom ämnen som gruppens medlemmar aldrig tidigare hade ägnat sig åt så sattes även problemlösningsförmågan på prov. Denna kurs, inte minst på grund av dess olikhet till resterande kurser på Industriell Ekonomi, krävde att små mängder relevant information var tvunget att urskiljas ur stora mängder text som till en början kunde verka helt obegripliga. Då det oftast endast var dessa texter, i form av datablad, som fanns tillhanda vid osäkerhet utvecklades gruppmedlemmarnas förmåga att lösa problem, både som grupp och självständigt, något som är essentiellt för en civilingenjör.

## Referenser

### Skriftliga:

**ATmega16, datablad**

<http://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Processors/ATmega16.pdf>

**Parallax 433 MHz RF Transmitter, datablad**

<http://www.parallax.com/sites/default/files/downloads/27982-433-MHz-RF-Transceiver-Documentation-v1.1.pdf>

**Dual H-bridge Motor Driver DRV8833, datablad**

<http://users.ece.utexas.edu/~valvano/Datasheets/drv8833.pdf>

**LP3855, datablad**

<http://www.ti.com/lit/ds/symlink/lp3855-adj.pdf>

**MCP1827, datablad**

<http://ww1.microchip.com/downloads/en/DeviceDoc/22001b.pdf>

### Muntliga:

Bertil Lindvall (handledare), Forskningsingenjör, CI.

# Bilagor

## Bilaga 1 - Kravspecifikation

### Mål

Bygga en radiostyrd bil och vid behov utöka denna med diverse funktioner.

### Funktionella krav

- Modellen ska kunna röra sig framåt, bakåt och vridas kring sin mittpunkt.
- Modellen ska kunna reagera på styrning från fjärrkontroll.
- Modellen ska stå still då fram-/bakknappen trycks ner samtidigt eller vänster-/höger vridning initieras samtidigt.
- Vid behov ska modellen utrustas med någon av följande funktioner:
  - Rörlig laserpekare
  - Rörlig kamera och mikrofon
  - Anti-krock sensorer
  - Portabel högtalare
  - Tuta och blinkers

### Kvalitetskrav

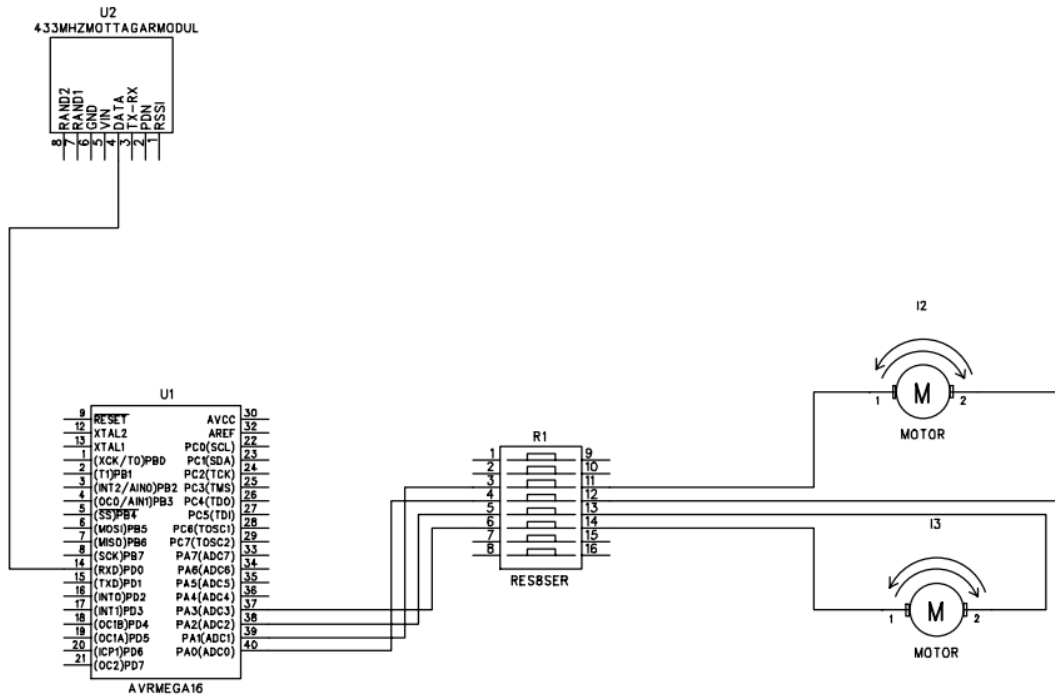
- Modellen ska vara lättstyrd. Till exempel får den inte svänga alltför snabbt.
- Modellen ska reagera på ett initierat kommando från fjärrkontrollen högst 0,5 s efter att initieringen skett. Detta är däremot ej påverkbart av oss.

### Komponenter

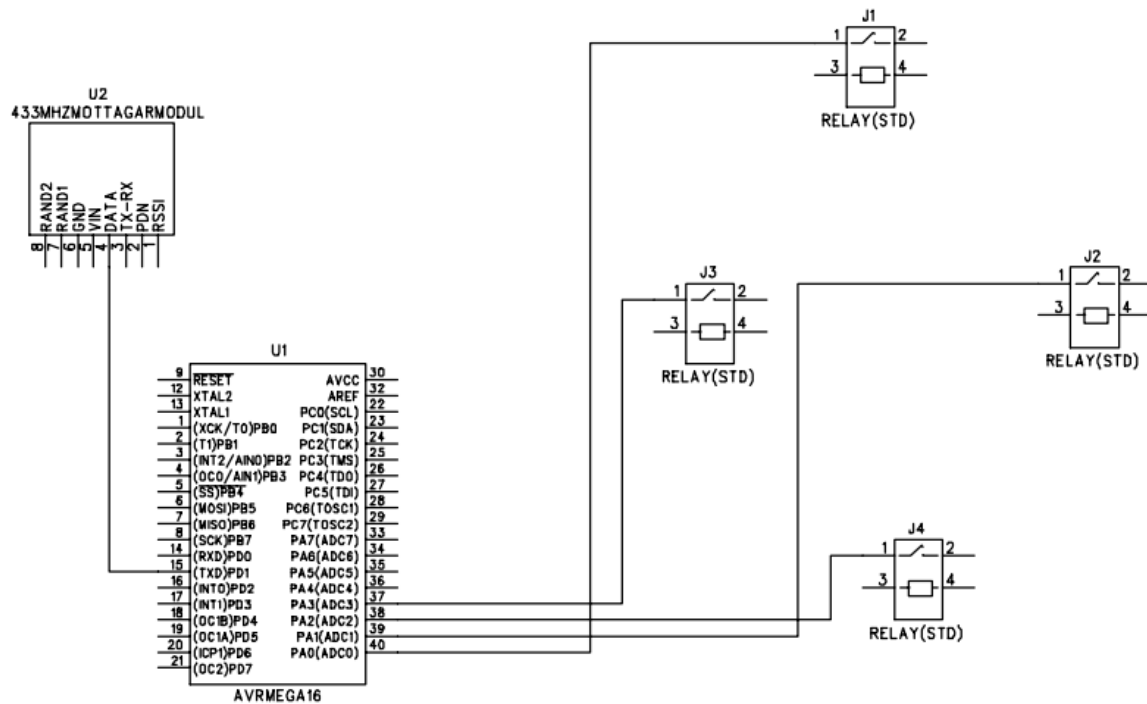
- En bas med hjul, motor och band.
- Mikroprocessor för styrning av motorn.
- Trådlös sändare (fjärrkontrollen) och mottagare (modellen).
- Knappar alternativt spak för styrning.
- Batteri till modellen och fjärrkontrollen.
- Eventuella andra komponenter till tilläggsfunktionerna

## Bilaga 2 - Kopplingscheman

### Bilen



### Fjärrkontrollen



## Bilaga 3 – Programkod

### Bilen

```
#define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>

void portSetup(void) {
    DDRA = 0xFF;
    PORTA = 0x00;
    DDRD = 0x00;
    PORTD = 0x00;
}

void USART_Init(void){
    //Set baud rate to 4800 bps
    UBRRH = 0;
    UBRRL = 103;

    //Set receiver enabled
    UCSRB = (1<<RXEN);

    //Access UCSRC, set 2 stop bits and set 8 bit-frame
    UCSRC = 0b10001110;
}

char USART_Receive(void){
    while(!(UCSRA & (1<<RXC)));
    return UDR;
}

void direction(char data){
    switch(data){
        case 'W': PORTA = 0b00001001; //Forward
        break;
        case 'D': PORTA = 0b00001010; //Turn right
        break;
        case 'S': PORTA = 0b00000110; //Backward
        break;
        case 'A': PORTA = 0b00000101; //Turn left
        break;
        default: PORTA = 0x00;
        break;
    }
}

int main(void)
{
    portSetup();
    USART_Init();
    while(1)
    {
        char data = USART_Receive();
        direction(data);
    }
}
```



## Fjärrkontrollen

```
#define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>

int main(void)
{
    setup();
    USART_Init();
    while(1)
    {
        buttonPressed();
    }
}

void setup(void)
{
    DDRA = 0x00;
    DDRD = 0xFF;
}

void USART_Init(void){
    UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) | (0<<UPM0) | (0<<USBS) | (1<<UCSZ1) | (1<<UCSZ0) | (0<
    <UCPOL);
    UBRRH=0x00;
    UBRRL=103;
    UCSRB = (1<<TXEN);
}

void USART_Transmit(char data){
    while(!(UCSRA & (1<<UDRE)));
    UDR = data;
}

void buttonPressed(){
    if(PINA == 0b11111110){//Forward-button
        USART_Transmit('W');
    }
    if(PINA == 0b11110111){//Steer right-button
        USART_Transmit('D');
    }
    if(PINA == 0b11111011){//Backward button
        USART_Transmit('S');
    }
    if(PINA == 0b11111101){//Steer left-button
        USART_Transmit('A');
    }
}

/*void transmit(int nbrOfOnes) {
    for(int count; count <= nbrOfOnes; count++){
        PORTD = 0xFF;
        //PORTD = 0x00;
    }
}

void pause(void){
    for(int i; i <= 100; i++){
        PORTD = 0x00;
    }
}*/
```

Bilaga 4 - Bilder

