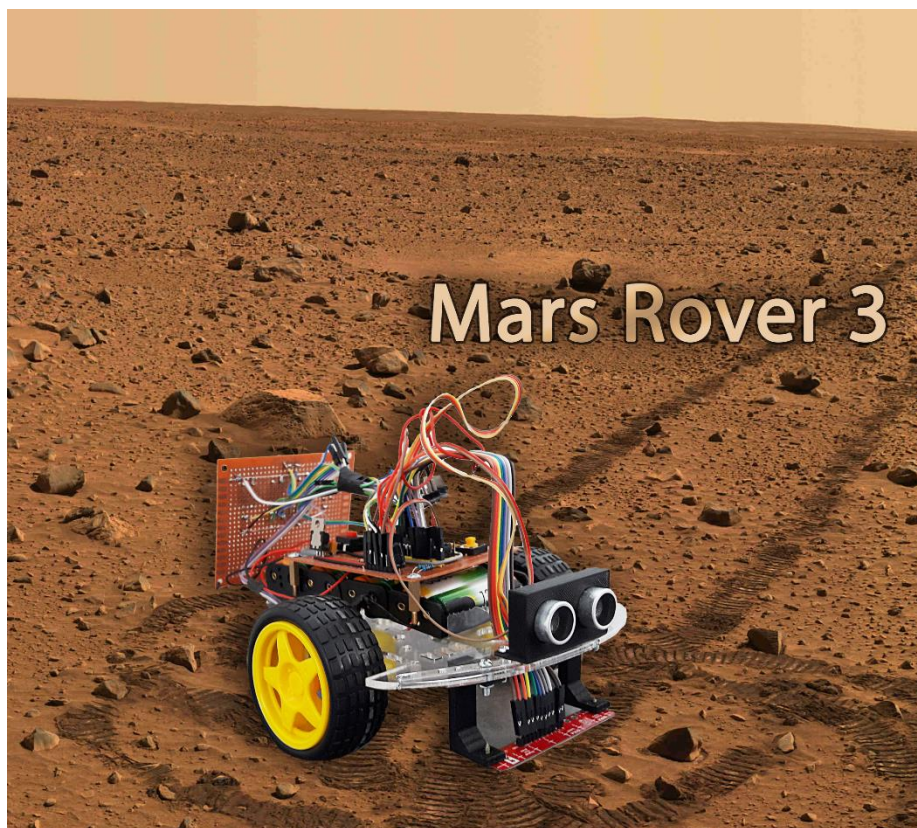


Teknisk dokumentation

Linjeföljande robot



Version	Anmärkning	Datum	Sign
1	Version 1	2016-06-09	JS

Arbetsgrupp

Namn	Ansvar
Martin Bäckström	Styrsystem
Marcus Johansson	Styrsystem
Björn Svensson	Menysystem
Johan Skog	Menysystem
Abdul Khadir Hussein	Menysystem
Hans Winzell	Styrsystem, hårdvara

Innehållsförteckning

Innehåll

1.	Inledning.....	4
1.1.	Bakgrund	4
1.2.	Syfte.....	4
1.3.	Källor.....	4
2.	Beskrivning	5
2.1.	Systemet.....	5
2.2.	Manuellt läge.....	5
2.3.	Autonomt läge.....	6
2.4.	IR-funktionen.....	6
2.5.	Piezo buzzer.....	7
3.	Reflektion	8
4.	Problematik	9
3.	Kopplingsschema.....	10
4.	Flödesschema	11
5.	Komponentlista	12

1. Inledning

Detta dokument är en teknisk beskrivning av den linjeföljande roboten Mars Rover 3, skapad av Embedded System Robotics.

1.1. Bakgrund

I kursen Programmering Inbyggda System ingick att driva ett projekt och skapa ett system som ska visas upp i slutet av kursen. Gruppen valde att göra en robot som ska kunna gå i både autonomt och manuellt läge.

1.2. Syfte

I det autonoma läget så ska roboten följa en svart linje på marken med hjälp av 5 stycken IR-sensorer. I manuellt läge så ska roboten styras utav en fjärrkontroll genom en IR-mottagare.

1.3. Källor

Källorna består utav olika sidor på internet där exempelkod samt information om komponenter hittades. Även boken Make: AVR Programming användes. Under hela projektetsgång fanns handledare till hands för råd.

2. Beskrivning

En display och knappar används för att komma in i de olika menyerna. Menyerna är:

- Manual
- Autonom
- Settings

När meny är inställd så sätts roboten i det läge som är valt.

I det autonoma läget använder roboten en sensorpanel bestående av 5st IR-dioder och IR-sensorer. Sensorpanelen gör så att roboten kan hitta och känna av var linjen är, för att sedan kunna justera hastigheten på motorerna för att kunna styra åt rätt håll och därmed följa linjen. Detta är tack vare en regleralgoritm (PID) i koden som skickar signalerna till motorerna. Det finns även en ultraljudssensor längst fram på roboten som skickar ut ljudvågor och mäter hur lång tid det tar för dem att studsas tillbaka. Med den tiden kan man då med en beräkning läsa av hur långt det är till föremålet framför. Roboten är satt att om det är något föremål som är 10 cm eller närmare så ska den stanna. För att sedan köra igång när föremålet är borta.



Bild 1: Ultra Sonic HC-SR04

I det manuella läget så används en IR-mottagare som letar efter en signal från en fjärrkontroll. I koden är det skrivet att 5 knappar används, med deras individuella hexkod, för att sedan kunna ge rätt signal till roboten för styrning.

2.1.Systemet

Roboten använder sig utav en Atmega328p samt ytterligare en Atmega 328p på en Arduino Nano. För att kommunicera mellan de två mikrokontrollerna används en I²C bus.

2.2.Manuellt läge

I det manuella läget så styrs roboten med en fjärrkontroll, detta genom en IR-sändare på fjärrkontrollen och en IR-mottagare på roboten. Varje knapp på fjärrkontrollen har en individuell hexkod. Denna hexkod används i programkoden för att ge specifika knappar specifika uppgifter.

Fjärrkontrollen styr så att roboten kan: Byta läge(Manuellt samt Autonomt), köra fram, bak, höger och vänster, samt stopp.

2.3. Autonomt läge

Det autonoma läget börjar med att kolla om det finns något hinder i vägen. Om hinder upptäcks så stannar roboten. Om inget hinder upptäcks går roboten in i en funktion. Denna funktion läser hur lång tid det tar för ljuset från IR LED-lampan till IR-mottagaren. Därefter görs en beräkning för att se vilken av de 5 IR-mottagarna som har en svart tejp under givaren. Värdet skickas till en PID-funktion som beräknar hur mycket ena eller andra motorns varvtal ska sänkas. När PID-funktionen är klar, skickas värdet till motorfunktionen. Där verifierar funktionen om värdet är positivt, negativt eller noll och ändrar den ena eller den andra eller båda motorernas varvtal utifrån resultat. Detta loopas om och om igen så länge roboten är i det autonoma läget.

2.4. IR-funktionen

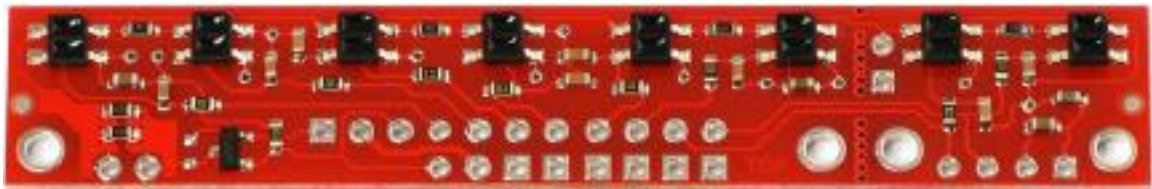


Bild 2: Linjeföljare med 8st IR-givare.

Sensor QTR-8RC

Det finns åtta par av LED och fototransistorer på kretsen. Varje fototransistor använder en urladdningskondensator som ger en mikrokontroller möjligheten att mäta reflektionen, genom att mäta tiden det tar för spänningen att bli noll. Vid mätning så sätts den aktuella porten till utgång. Därefter laddas urladdningskondensatorn upp i 10 μ s och när laddning är klar sätts porten till ingång.

Vid mörkare underlag, svart tejp (som användes i detta projekt), tar det så lång tid för ljuset att reflekteras att tidsgränsen på 2000 μ s överskrids. Detta innebär att linjen befinner sig under den aktuella givaren. När underlaget är ljusare så kan värden som exempelvis: 60 μ s komma ut från givaren.

Sensorerna är placerade på ett avstånd av 4 mm från golvet, då det rekommenderade maxavståndet från givare till golv är 9.5 mm och det mest optimala är 3 mm.

2.5. Piezo buzzer

För att göra bilkörningen mer njutningsfull för åskådaren har en musikspelare lagts till.

Tonerna alstras av en buzzer som spelar upp toner på bestämda frekvenser. För uppspelningen används en fördefinierad funktion ur Arduinos bibliotek. Det som skickas in till funktionen tone på Arduinon är tonlängd (2,4,8,16) och tonhöjd (frekvens) samt port. När roboten blir satt i autonomt läge spelas en kort nedräkningsslinga innan den kör iväg. Samtidigt som uppspelningen startas ges en order via I²C till slaven att motorerna ska föra roboten framåt.



Bild 3: Piezo buzzer

3. Reflektion

I ett stort projekt med många inblandade har vi kommit fram till att det är viktigt med tydliga roller, arbetsuppgifter och kontinuerliga avstämningsmöten. De problem som uppstod i gruppen var minskande närvaro och bristande delaktighet i drivandet av projektet. Detta måste samtliga i gruppen ta med sig och tänka på i framtida projekt. Överlag har samarbetet fungerat bra där alla har varit med och kompromissat samt diskuterat sig fram till olika lösningar.

4. Problematik

- När i2c skulle implementeras uppstod problem. Kommunikationen mellan mikrokontrollerna fungerade bra, dock bara under en begränsad tid. Utan anledning kunde hela systemet låsa sig. Efter mycket felsökning så togs kodraden `"while(!(TWCR & (1 << TWINT)))"` bort och allt fungerade felfritt. Denna kodrad ska inte vara med om man har interrupt-baserad i2c vilket detta system använder.
- När buzzer och IR mottagarens kod skulle sammanfogas uppstod konflikt mellan dem. Efter felsökning upptäcktes att konflikten var att de båda komponenterna använde samma timer-register i Arduinon. Efter att ha ändrat till ett annat timer-register i IR mottagarens bibliotek så fungerade allt som tänkt.
- Vid testkörning av roboten så fungerade allt bra till en början. Men efter en slumpartad tid började roboten att köra in i hinder den annars hade undvikit på banan. Efter efterforskning testades att montera en kondensator mellan 5volt och jord på ultraljudssensorn.
- När det var dags att testa funktionen för att följa en linje uppkom det problem. Roboten hade väldigt svårt att följa linjen och det kändes som att den gick lite hur som helst. Misstankarna gick direkt till PID beräkningarna. Det gjordes mycket ändringar av koden utan resultat. Tankarna gick då till hjulen på roboten. De var av dålig kvalitet och väldigt skeva. Hjulen byttes och det blev lite förbättring men långt ifrån bra. Återigen gick tankarna till koden och den ändrades ytterligare flera gånger, men av en slump vid testkörning låste sig det bakre "kundvagnshjulet" och roboten följde linjen enligt önskemål. Felkällan hade då upptäckts och åtgärdades genom att tejpa fast hjulet i en fast position.
- I början av utvecklingen användes alla åtta sensorer för att detektera linjens position. När de olika delarna sedan skulle slås ihop, upptäcktes att det blev konflikt med PWM och vanliga I/O portar. Programmet hann bara initiera portar innan det startade om. Vid felsökning upptäcktes att det var portar på B registret som inte ville fungera ihop med PWM. Som åtgärd kopplades de fyra portarna på B-registret loss. Efter att ha omstrukturerat de portar som blivit tilldelade till linjesensorn så används fem istället för åtta givare och vi låter givarna lysa konstant.

3. Kopplingsschema

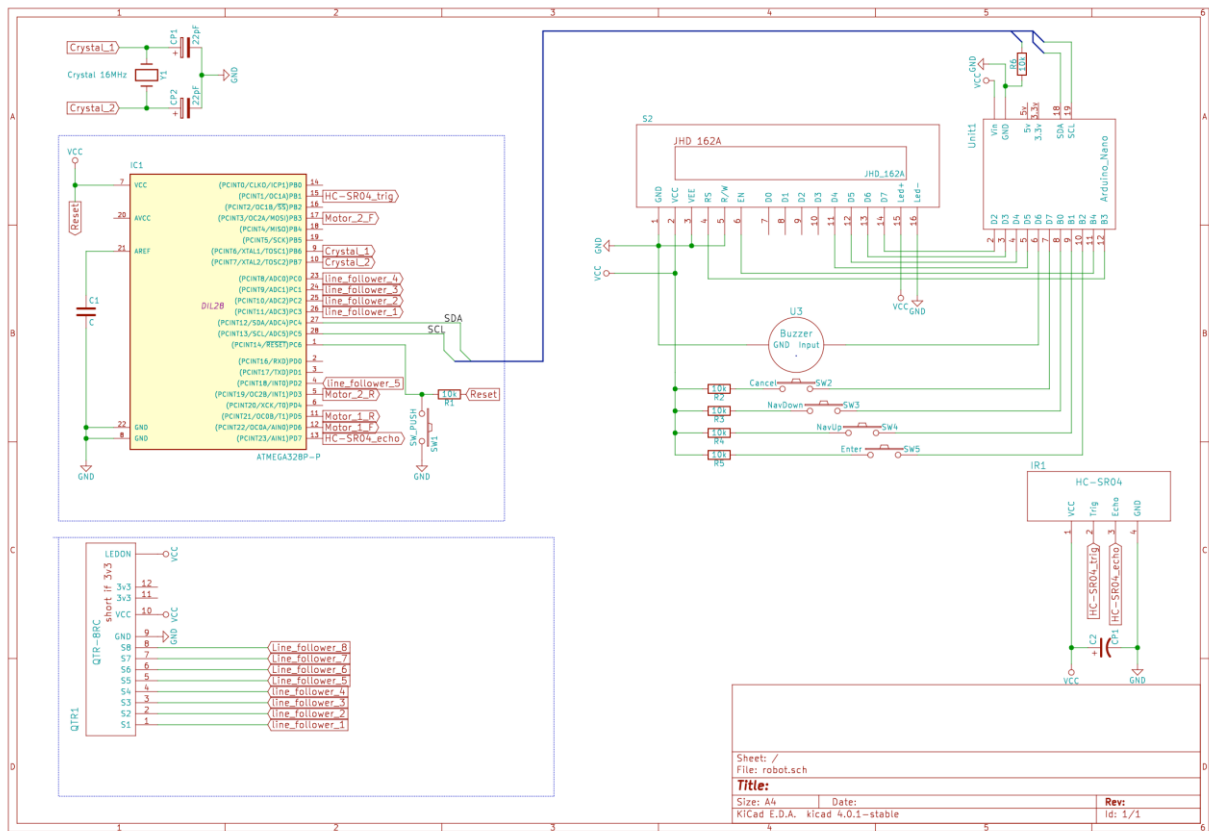


Bild 4: Kopplingsschema

4. Flödesschema

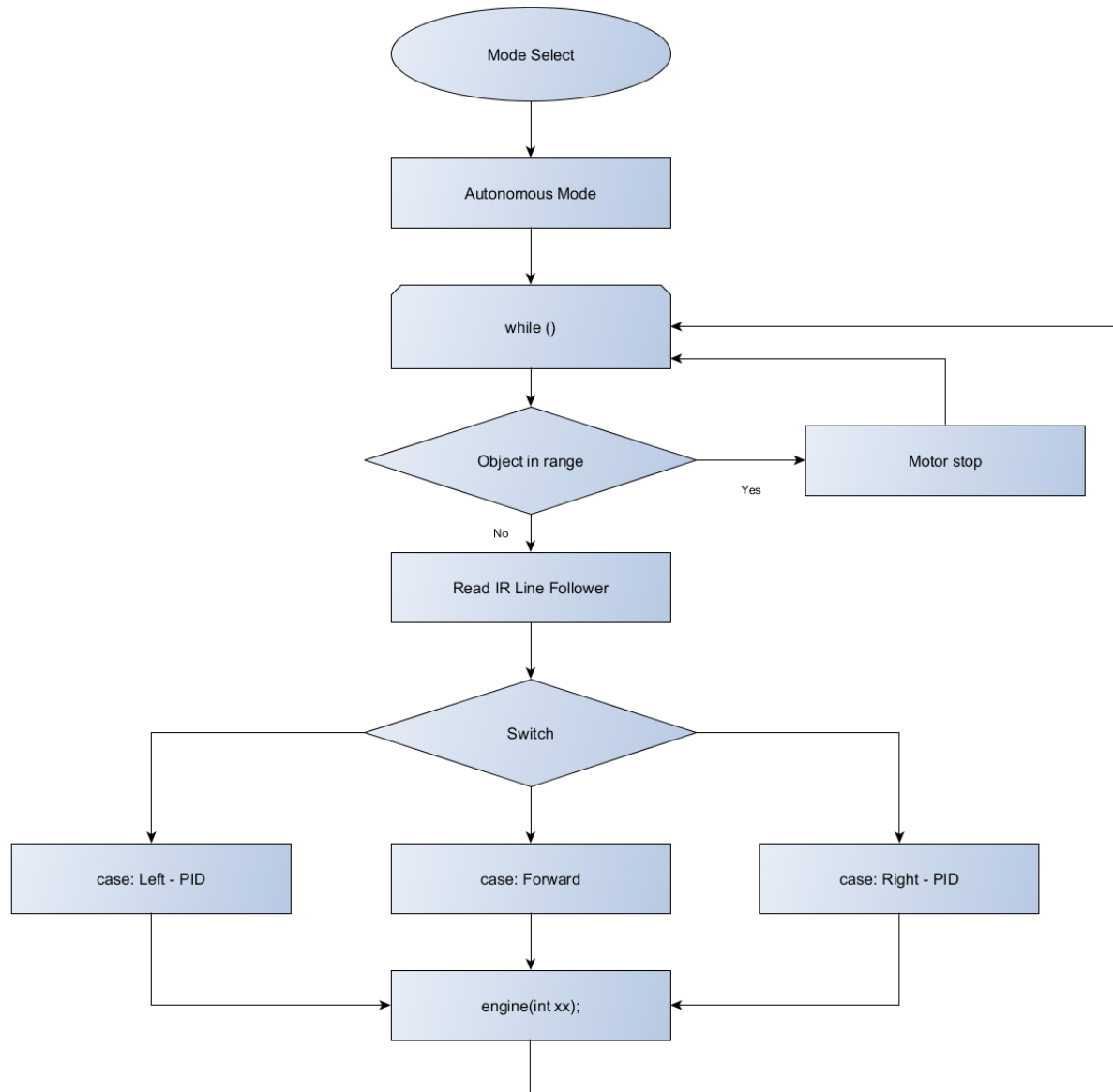


Bild 5: Autonomous flödesschema

5. Komponentlista

- QTR-8RC Reflectance Sensor Array
- IR receiver
- 74HC165
- Pololu USB AVR Programmer v2
- Capacitor 10uF/50V
- Robotplattform 2WD ink hjul och motorer
- On-Off Power Switch
- AVR Atmega328P
- IC Socket - for 28-pin 0.3" Chips
- 22pF Ceramic Capacitor
- 16MHz Crystal
- HC-SR04 Ultrasonic Sensor HC-SR04
- Piezo Buzzer
- Round Tactile Button Switch
- Dual H-Bridge Motor Driver for DC or Steppers 600mA - L293D
- Arduino Nano
- 8x AA batterier
- Batterihållare
- 5 volts regulator L7805AC-V