

Designspecifikation

Projektgrupp 8, TSEA29

Version 1.0

Status

Status	Namn	Datum
Granskad	Jakob Palmgren	2023-10-12
Godkänd	Olov Andersson	

Kartrobot

Grupp 8, HT-23
Tekniska högskolan vid Linköpings universitet

Namn	Ansvar	Telefon	E-post
Aidin Jamshidi	Styr och gränssnittsmodul	070-789 02 55	aidja644@student.liu.se
Ali Jafaree	Kommunikation och gränssnittsmodul	076-962 44 97	alija148@student.liu.se
Aws Hassan	Projektledare och sensormodul	072-152 20 60	awsha174@student.liu.se
Jakob Palmgren	Dokumentansvarig och kommunikationsmodul	072-732 50 01	Jakpa609@student.liu.se
Kebba Jeng	Projektledare och styrmodul	072-150 65 52	kebj541@student.liu.se
Theodor Wennerbo	Sensormodul och gränssnittsmodul	073-720 76 03	thewe344@student.liu.se

E-postlista för hela gruppen: TSEA29_2023HT_XX-Grupp8@groups.liu.se
Hemsida: <https://gitlab.liu.se/da-proj/microcomputer-project-laboratory-d/2023/g08/docs>

Kund: Anders Nilsson, 581 00, Linköping,
kundtelefon +4613282635 anders.p.nilsson@liu.se

Kursansvarig: Anders Nilsson, 3B:512, +4613282635, anders.p.nilsson@liu.se
Handledare: Olov Andersson, +4613282658, olov.andersson@liu.se

Innehåll

1. Inledning	6
2. Systembeskrivning	7
2.1 Delsystem	7
2.2 Övergripande konstruktion	9
2.3 Kopplingsschema & komponenter	10
2.3.1 Beräkningsenheter	10
2.3.2 Sensorer	10
2.4 Programflöde	11
3. Delsystem	12
3.1 Kommunikationsmodul	12
3.1.1 - I2C	13
3.1.2 - Kommunikation med sensormodulen	14
3.1.3 - Kommunikation med styrmodulen	14
3.1.4 Kopplingsschema	15
3.1.5 Komponenter	15
3.1.6 - Implementation	16
3.1.7 - UART (Universal Asynchronous Receiver/Transmitter)	17
3.1.8 - Bluetooth	18
3.2 Sensormodul	20
3.2.1 Modulens funktion	20
3.2.2 Kopplingsschema	21
3.2.3 Komponenter	23
3.2.4 Data- och programstruktur	24
3.3 Styrmodul	25
3.3.1 Modulens funktion	25
3.3.2 Kopplingsschema	26
3.3.3 Komponenter	28
3.3.4 Data- och programstruktur	29
3.4 Gränssnitt	30
4. Realiseringsstrategi	31
4.1 Strategi	31
4.2 Kritiska sektorer	31

4.3 Tester	31
Referenser	32

Dokumenthistorik

Version	Datum	Utförda förändringar	Utförda av	Granskad
1.0	2023-10-12	Andra versionen	G2	Jakob Palmgren
0.2	2023-10-11	Sjätte utkastet	G2	Jakob Palmgren
0.1	2023-10-05	Första versionen	G2	Jakob Palmgren
0.05	2023-10-04	Femte utkastet	G2	Jakob Palmgren
0.04	2023-10-03	Fjärde utkastet	G2	Jakob Palmgren
0.03	2023-10-03	Tredje utkastet	G2	Jakob Palmgren
0.02	2023-10-02	Andra utkastet	G2	Jakob Palmgren
0.01	2023-10-02	Första utkastet	Kebje541	Jakob Palmgren

1. Inledning

Designspecifikationen redogör de delsystem och komponenter som tillsammans utgör kartroboten, inklusive design, kopplings- och flödesscheman. I designspecifikationen beskrivs delsystemens funktion, komponenter och programstrukturer detaljerat för att få en bra förståelse av systemet och dess delar. En realiseringsstrategi presenteras, inklusive tester för att säkerhetsställa projektets framgång.

2. Systembeskrivning

Systemet utgörs av en fyrhjulig robot, med syftet att kartlägga en bana inom en 10x10 meters yta. Systemet har förmågan att autonomt navigera genom banan samtidigt som den skapar en karta över området. Den inleder sin färd från en förutbestämd startposition och återvänder dit när kartläggningen är fullständig.

2.1 Delsystem

Systemet är moduluppbyggt och består av fyra delsystem:

- Kommunikationsmodul
- Sensormodul
- Styrmodul
- Gränssnitt

Kommunikationsmodulen är hjärnan i systemet, all information som skickas eller tas emot går igenom den. Kommunikationsmodulen ser till att informationen från sensormodulen skickas till styrmodulen för att roboten ska kunna förflytta sig samt regleras för att undvika objekt. Kommunikationsmodulen skickar samtidigt relevant data från sensormodulen till gränssnittet för att få en klar representation av roboten samt kartlägga banan.

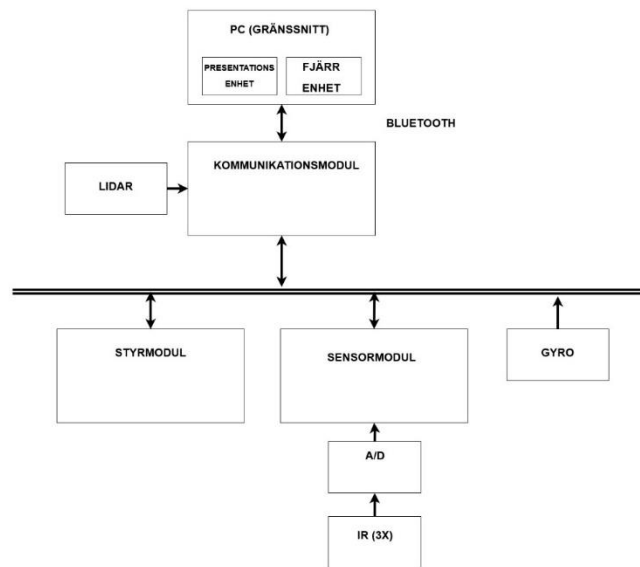
Sensormodulen är robotens sinne. Sensormodulen använder sig av olika sensorer för att klargöra var roboten är och vad den "ser". Sensorernas information behandlas och konverteras till data som de andra modulerna kan förstå. Datan skickas sedan vidare till kommunikationsmodulen.

Styrmodulen kontrollerar robotens kropp (i det här fallet är det bara hjul). Styrmodulen har logik för rörelse, den tar in information från sensormodulen genom kommunikationsmodulen för att reglera fart samt ta sig genom banan utan att kollidera med objekt.

Gränssnittet ger en visuell representation av det som händer i systemet och den finns på en bärbar dator och kopplas till roboten trådlöst. I gränssnittet ritas en karta och information om hastighet, avstånd från väggarna och liknande visas. Roboten har en knapp för att

växla mellan autonomt och fjärrstyrnings läge. Fjärrstyrningen sker genom gränssnittet med en knapp som skickar information till kommunikationsmodulen.

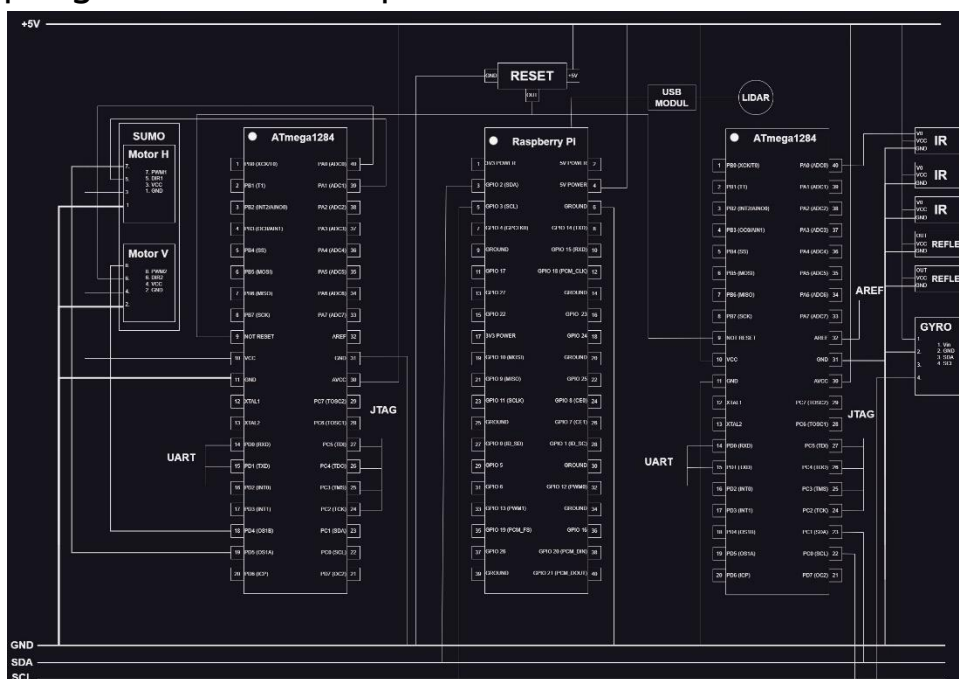
2.2 Övergripande konstruktion



Figur 1: Övergripande schema över systemet

Figur 1 visar ett övergripande blockschema. Genom I2C bussen kommunicerar styr- och sensormodulen med kommunikationsmodulen som kommunicerar med datorn via ett gränssnitt genom Bluetooth.

2.3 Kopplingsschema & komponenter



Figur 2 visar kopplingsschemat för hela systemet samt de komponenter som används i konstruktionen.

2.3.1 Beräkningsenheter

Enhet	Antal
Raspberry Pi 3	1
ATmega1284	2

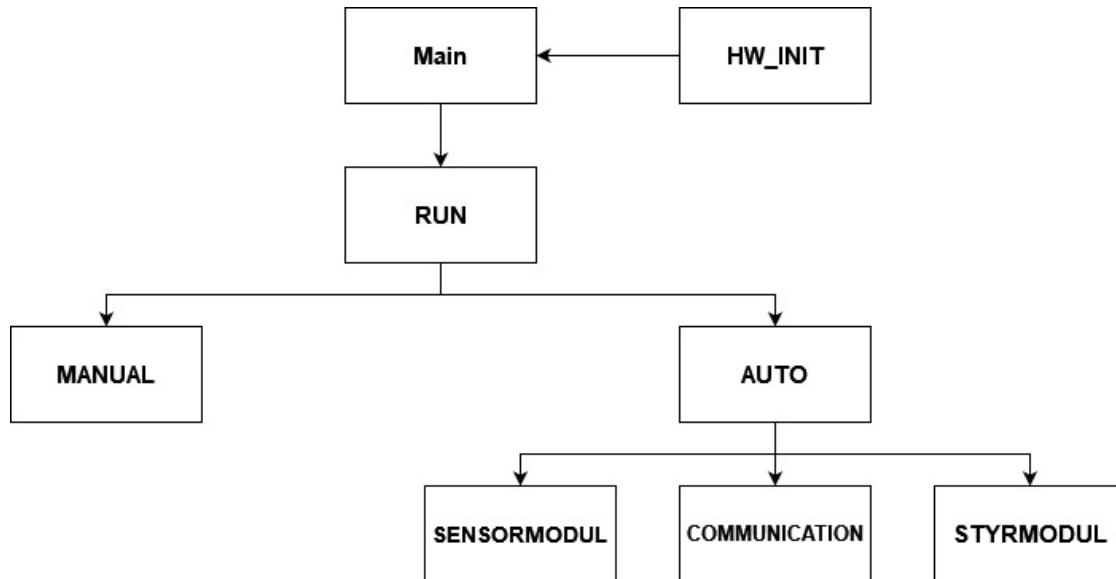
Tabell 1, tabell på beräkningsenheterna i systemet

2.3.2 Sensorer

Komponenter	Antal
GP2D120 IR (4–30 cm)	2
GP2Y0A21 IR (10–80 cm)	1
Reflexsensor	2
RPLIDAR A2	1
AdaFruit-10 DOF	1

Tabell 2, tabell på sensorerna i systemet

2.4 Programflöde



Figur 3: Systemets programflöde

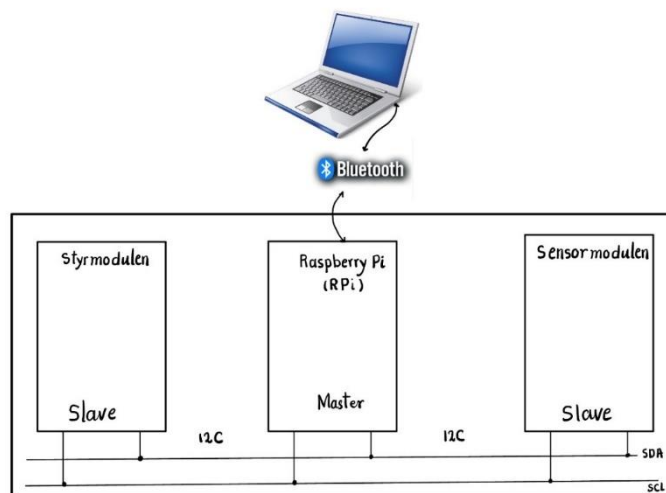
Programflödet i systemet startar med att initiera hårdvaran samt de protokoll som används i systemet. Därefter beroende på användarinput kör systemet i fjärr eller autonomt läge varefter de respektive modulerna utför dess instruktioner.

3. Delsystem

I kapitlet nedan ges en inblick i de delsystem som tillsammans utgör kartroboten.

3.1 Kommunikationsmodul

Kommunikationsmodulen gör det möjligt att överföra data mellan alla moduler som ingår i konstruktionen. Kommunikation mellan systemets enheter sker med hjälp av en intern I2C-buss, UART samt Bluetooth-uppkoppling för överföring av data från och till en dator.



figur 4: Master-slave koppling

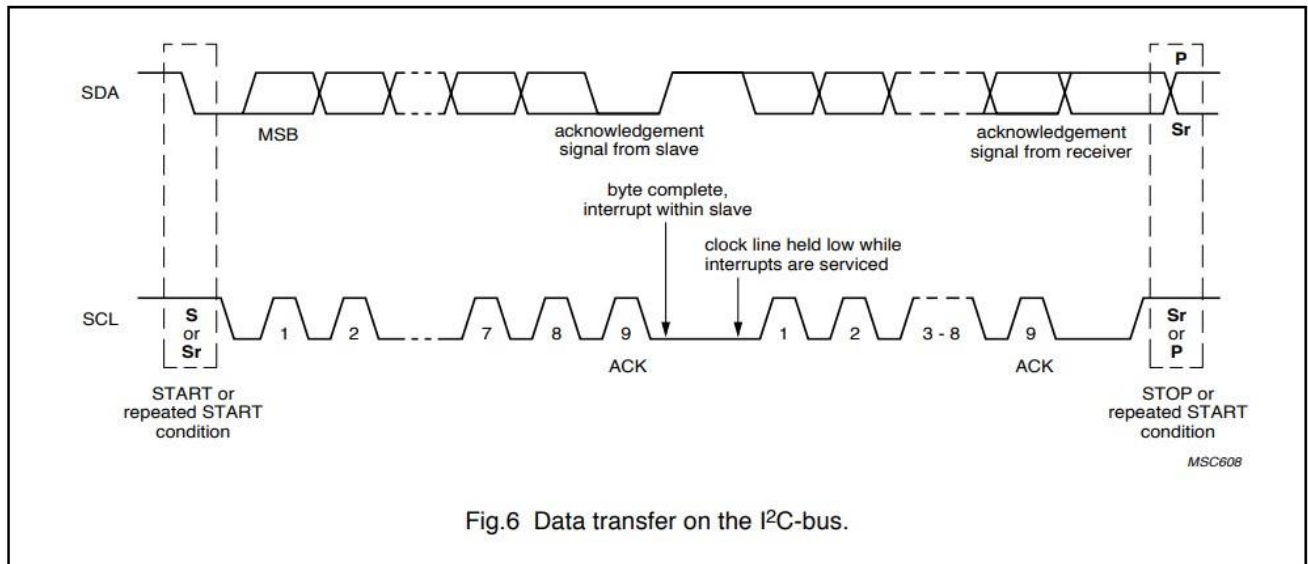
3.1.1 - I2C

Roboten har en intern I2C-buss som hanterar överföringen mellan styrmodulen och Raspberry Pi samt mellan sensormodulen och Raspberry Pi. Där Raspberry Pi är master, styr och sensormodulen är slave. I2C är uppbyggt av nedanstående delar.

Del	Beskrivning
Start-Stop	Kommunikationen i I2C inleds av en startsignal(S) från mastern och avslutas med en stop signal(P). Startsignalen indikerar att en ny kommunikationssession börjar, medan stoppsignalen indikerar att sessionen är över.
Addressering	Varje I2C enhet (slave-enhet) har en unik adress som används av mastern för att identifiera vilken som ska kommunicera med. Den vanligaste adressen är 7 bitar som kan adressera 128 slave enheter.
Dataöverföring	Data överförs i I2C som 8 bitars datapaket. Kommunikationen är synkroniserad, och båda enheterna måste vara överens om överföringshastigheten.
Läs- och skrivåtgärder	I2C-överföringar kan vara läsningar (från slav till master) eller skrivningar (från master till slave). En startsignal följs av adress och en bit som indikerar om mastern vill läsa eller skriva data. Denna bit kallas "Read/Write"-bit.
ACK och NACK	Efter varje 8-bitars överföring svarar mottagaren (slave-enheten) med en ACK eller NACK-signal. En ACK signal (logiskt låg) anger att överföringen lyckades, medan en NACK signal (logiskt hög) indikerar ett fel eller att överföringen avslutades.
Byte sändning och mottagning	I2C-överföringar sker byte för byte. Efter varje byte måste mottagaren skicka ett ACK eller NACK beroende på om den vill fortsätta överföringen eller inte.
Klocka och datapinnar	I2C använder två signalpinnar: en klocka (SCL - Serial clock) och en data (SDA - Serial data). Klockpinnen, kallad SCL, används för att synkronisera timingen för dataöverföringen mellan master och slave-enheter. Klocksignalen bestämmer när data ska avläsas eller sändas. Kommunikationen sker i pulser, och data byts normalt ut på varje stigande flank (övergång från låg till hög) eller fallande flank (övergång från hög till låg) av klocksignalen. Datapinnen kallad SDA, är den faktiska ledaren för datan i I2C-kommunikationen. Master och slave-enheter använder denna pinne för att sända och ta emot data. SDA är en

	tvåvägsbuss, vilket innebär att både master och slave-enheter kan sända och ta emot data på samma ledare.
--	---

Tabell 3: tabell på I2C delar



figur 5: Dataöverföring med I2C bussen

3.1.2 - Kommunikation med sensormodulen

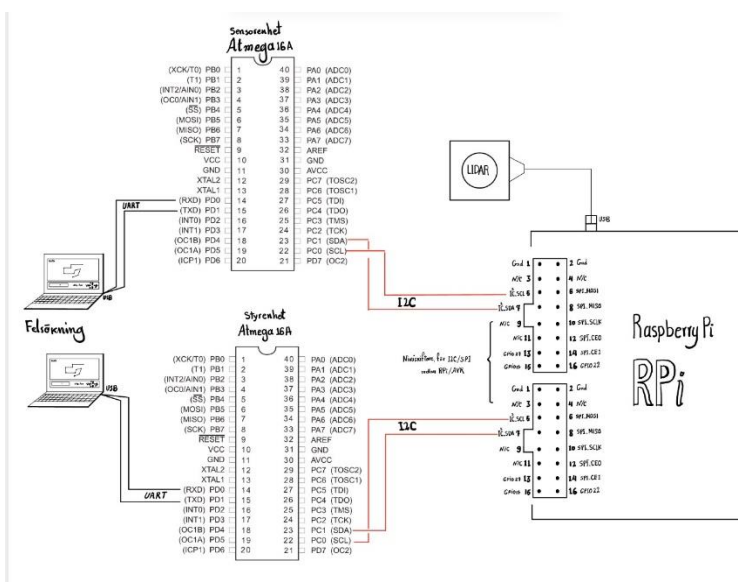
Kommunikationen med sensorenheten sker via I2C. Exempel på data som sensormodulen överför är avståndet mellan sensorerna och det detekterade objektet från IR sensorer, 3D punktmoln som representerar omgivning inkluderar koordinater för punkter som representerar objekt i området skickas av LIDAR, reflektions data som rapporterar om sensor ljuset detekterar en reflektion från ett objekt eller inte från reflexsensorn.

3.1.3 - Kommunikation med styrmodulen

kommunikationen med styrenheten sker via I2C. Exempel på data som styrmodulen överför är styrkommandon för att kontrollera robotens rörelse och beteende, instruktioner som start, stop, ändra riktning och hastighet. Skicka kartdata och positionsinformation för visualisering och navigering.

3.1.4 Kopplingsschema

Figur 6 visar kopplingsschema för kommunikationsmodulen. Raspberry Pi (RPI) använder sig av 3.3V logik på sin GPIO medan AVR:erna använder 5V därför används en nivåskiftare med två identiska 16 pinnars IDC-kontakter sön gör det möjligt att koppla styrmodulen till den ena 16 pinnars och sensormodulen till den andra.



Figur 6: I2C kommunikationen mellan styrenheten - RPi - sensorenheten.

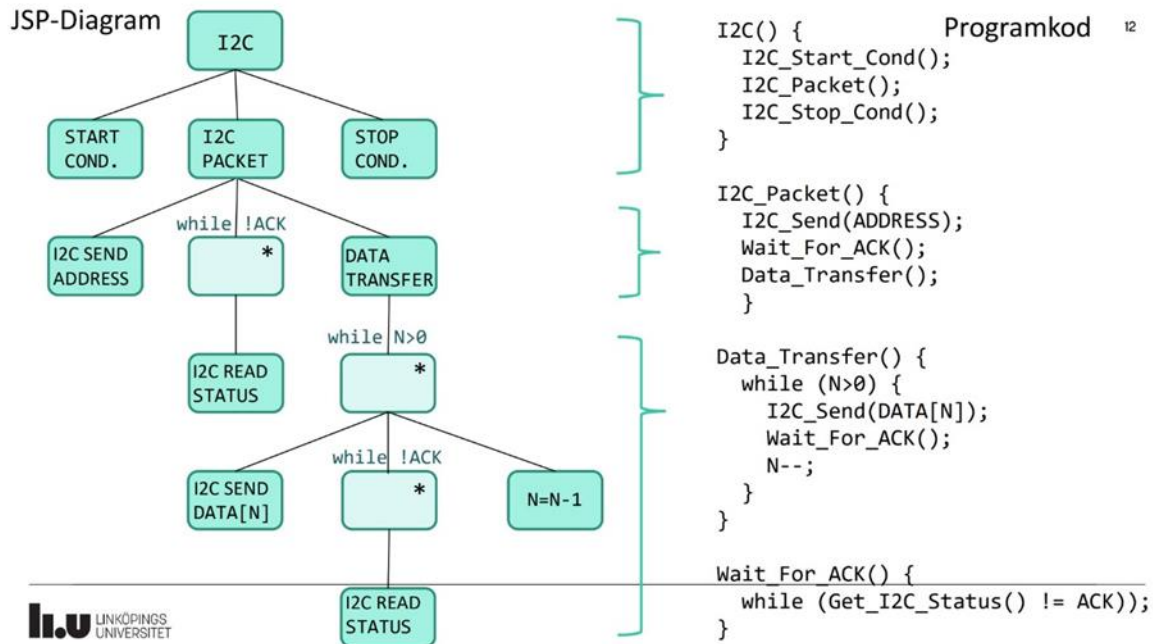
3.1.5 Komponenter

Komponenter	Antal
Raspberry Pi 3	1
Nivåskiftare RPi - AVR	1
I2C kabel	2
USB-kabel	1
Virkort	3

Tabell 4: tabell på beräkningsenheterna i systemet

3.1.6 - Implementation

Figur 7 visar JSP-diagrammet och pseudo-kod för implementation av I2C.



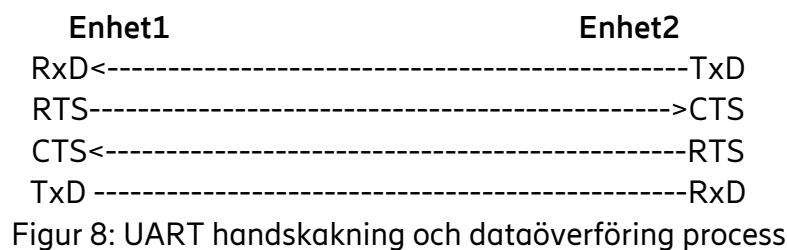
figur 7: visar implemetations detalj. [1]

3.1.7 - UART (Universal Asynchronous Receiver/Transmitter)

UART är ett seriekommunikationsprotokoll via RS232 med signalerna:

- TxD, Transmit Data (output)
- RxD, Receiver Data (input)
- CTS, Clear To Send (input), Aktiv låg
- RTS, Request To Send (output), Aktiv låg,

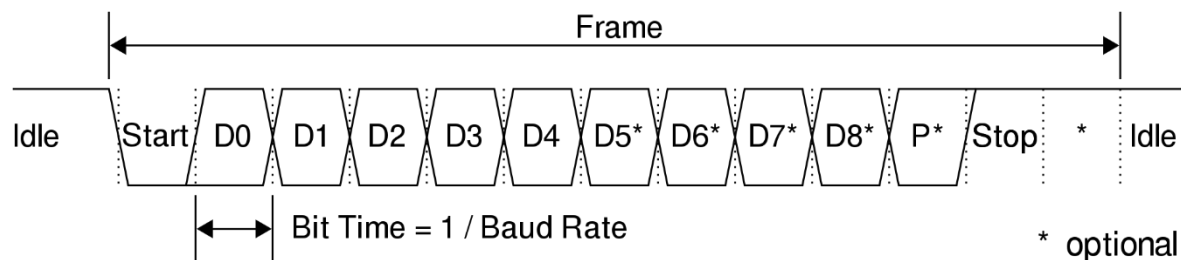
CTS och RTS är handskakning signaler som används för att kolla om kopplade enheterna i var ände är redo att initiera kommunikationen och data överförs via TxD och RxD. Figur 8 visar att med RTS låg, enhet1 signalerar att den är redo att ta emot data via RxD från TxD av enhet2 och när CTS är låg signalerar den till enhet1 att dataöverföring från enhet2 är klart.



Protokollet:

Protokollet beskriver på vilket sätt informationen överförs. Figuren 9 beskriver protokollet.

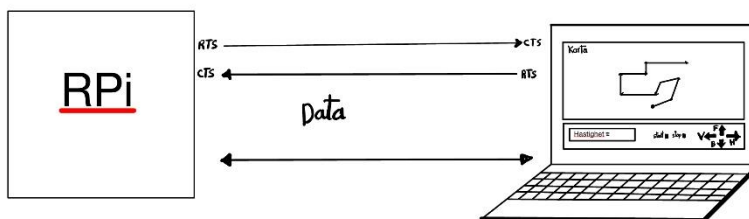
- Hastigheten: max 115200 BAUD(databitar per sekund)
- Full duplex
- Flaggor signalerar sänd/mottaget data klart
- Start: en bit
- Stop: en bit
- Data: 8 bitars



figur 9. UART protokoll

3.1.8 - Bluetooth

Bluetooth-protokollet är en trådlös kommunikationsstandard som används för att ansluta enheter över korta avstånd. Data skickas i form av bitar i paketstrukturer och överförs via radiofrekvenskommunikation. I konstruktionen används Bluetooth för kommunikationen mellan dator och roboten så att vi kan rita upp kartan samt kunna fjärrstyra roboten. Innan dataöverföring kan inledas krävs en handskakning mellan enheterna. En RTS-signal (Request To Send) används för att ange om enheten är redo att skicka eller ta emot data. Signalen avläses sedan på den andra enhetens CTS-ingång (Clear To Send). När det väl är gjort har en förbindelse mellan enheterna skapats och data kan nu skickas.



figur 10: Bluetooth mellan RPi & Datorn

I Tabell 5 beskrivs hur det Bluetooth protokollet fungerar och hur data skickas.

Del	Beskrivning
Upptäckt och Parning:	När två Bluetooth-enheter närmar sig varandra, sker en upptäcktsprocess där de letar efter varandra. Detta kan göras synligt eller osynligt beroende på inställningarna på enheterna. Om två enheter vill kommunicera, måste de paras ihop. Detta innebär att de delar säkerhetsnycklar och identifierar varandra.
RF-kommunikation (Radiofrekvens):	Bluetooth använder radiofrekvenskommunikation för att överföra data. Det innebär att den skickar elektriska signaler över korta avstånd.
Dataöverföring:	Data skickas i form av bitar. Varje bit representeras av en elektrisk signal med två möjliga värden: 1 och 0. Dataöverföringen sker genom att modulera signaler på en bärfrekvens (ofta 2,4 GHz för Bluetooth). Signalerna moduleras med olika tekniker beroende på Bluetooth-versionen.
Paketstruktur:	Data skickas i paket. Varje paket består av flera delar: Åtkomstkod (Access Code): De första 72 bitarna i ett paket används för att etablera en anslutning mellan enheterna. Huvud (Header): De andra 54 bitarna som kallas header innehåller information om paketet, inklusive längd och typ av data som skickas. Data: Den faktiska data som ska överföras. Storleken på databitarna varierar beroende på kravet och varierar mellan 136 bitar och 8160 bitar. Försumplare (Trailer): Innehåller felshanteringsinformation och slutar paketet.
Målstyrd kommunikation:	Bluetooth använder en poäng-till-punkt- eller multipunkt-anslutning. Det innebär att data skickas från en sändare till en mottagare (eller flera mottagare om det är en multipunkt-anslutning). Adressering används för att dirigera data till rätt enhet.
Frekvenshoppning (Frequency Hopping):	För att undvika störningar och säkerställa pålitlig kommunikation hoppar Bluetooth-enheter mellan olika frekvenser inom det tilldelade bandet (hoppande spridningsspektrum). Frekvenshoppning sker i en fördefinierad sekvens som båda enheter känner till.
Säkerhetsåtgärder:	Bluetooth har inbyggda säkerhetsåtgärder, inklusive kryptering och autentisering, för att skydda data under överföringen.
Protokollstack:	Bluetooth använder en protokollstack som består av flera lager, inklusive RF, LMP (Link Manager Protocol), L2CAP (Logical Link Control and Adaptation Protocol), och applikationslagerprotokoll.

Tabell 5: En översikt över hur Bluetooth fungerar och hur data skickas

3.2 Sensormodul

Sensormodulen är en slav på I2C bussen med huvudsakliga uppgiften att vara informations givaren. Den kommer bestå mest av hårdvara, olika sensorer, samt lite kod för att kunna representera inskickade data och skicka vidare från sensorerna till kommunikationsmodulen.

3.2.1 Modulens funktion

Sensormodulen kommer bestå av en AVR (ATMega1284) som med hjälp av ADC pins kommer kopplas till sensorer. Processorn kommer att läsa mätdata från sensorerna sedan med hjälp av databladen till varje sensor ska mätdata "översättas" till ett binärt värde som representerar avståndet i cm, detta sker kontinuerligt. Därefter sparas värdena i minnet. När kommunikationsmodulen begär data skickas sparade data genom I2C bussen.

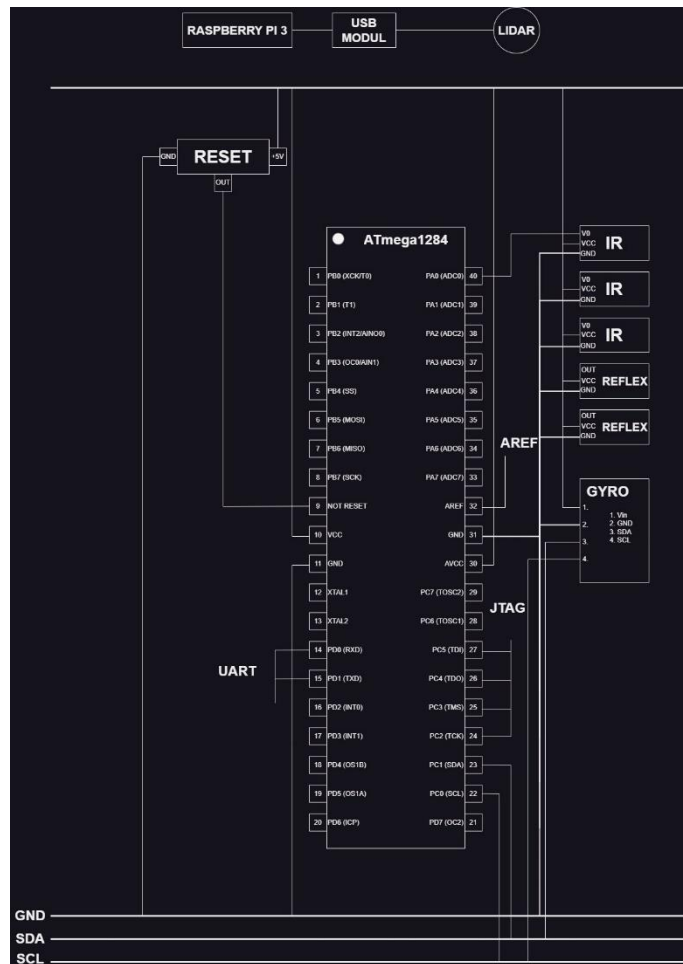
Sensormodulen kommer använda sig av tre IR-sensorer, en IR-sensor som mäter 10–80 cm på framsidan samt 2 IR-sensorer som mäter 4–30 cm på varsin sida av roboten.

Mätvärdena från sensorerna går in till Processorn genom varsin ADC pin som sen hanteras för att få läsliga data därefter sparas det på processorns minne.

Sensormodulen kommer ha två odometrar på varsin sida av roboten som kommer tala om hur långt den har åkt. Det kommer användas för reglering samt för kartläggning.

Sensormodulen kommer utnyttja en RPLIDAR. RPLIDAR:n kopplas till Raspberry pi genom USB och använder sig av UART kommunikation för att skicka avstånd i mm, riktningen i grader sedan en startflagga för att markera ny mätning. Den informationen kommer endast användas för kartläggning och kommer därför inte behöva översättas.

3.2.2 Kopplingsschema



figur 11: Kopplingsschema för sensormodulen

Nr.	Kopplingspunkt A	Kopplingspunkt B	Syfte
1	NOT RESET [Pin 9, ATmega1284]	OUT [RESET krets]	Krets för att återställa modulen.
2	GND [Pin 11, ATmega 1284]	GND buss	Jordanslutning för jordreferens.
3	VCC +5 V [Pin 10, ATmega 1284]	+5 V Buss	Förser mikrokontrollenheten med den nödvändiga spänningen för att fungera korrekt.
4	PA0 (ADC0) [Pin 40, ATmega1284]	Vo [IR modul GP2D120] #1	Utsignal från IR modulen med analog spänning.
5	GND [IR modul GP2D120] #1	GND buss	Jordanslutning för jordreferens.

6			PA1 (ADC1) [Pin 39, ATmega1284]	Vcc [IR modul GP2D120] #1	Insignal för att aktivera IR modul.
7			PA2 (ADC2) [Pin 38, ATmega1284]	Vo [IR modul GP2D120] #2	Utsignal från IR modulen med analog spänning.
8			GND [IR modul GP2D120] #2	GND buss	Jordanslutning för jordreferens.
9			PA3 (ADC3) [Pin 37, ATmega1284]	Vcc [IR modul GP2D120] #2	Insignal för att aktivera IR modul.
10			PA4 (ADC4) [Pin 36, ATmega1284]	Vo [IR modul GP2Y0A21]	Utsignal från IR modulen med analog spänning.
11			GND [IR modul GP2Y0A21]	GND buss	Jordanslutning för jordreferens.
12			PA5 (ADC5) [Pin 35, ATmega1284]	Vcc [IR modul GP2Y0A21]	Insignal för att aktivera IR modul.
13			PA6 (ADC6) [Pin 34, ATmega1284]	Out [Reflexsensor] #1	Inspänning från reflexsensor.
1 4	+5 V [Reflexsensor] #1	+5 V Buss	Inspänning till reflexsensor #1		
15		GND [Reflexsensor] #1	GND buss	Jordanslutning för jordreferens.	
16			PA7 (ADC7) [Pin 33, ATmega1284]	Out [Reflexsensor] #2	Inspänning från reflexsensor.
1 7	+5 V [Reflexsensor] #2		+5 V Buss	Inspänning till reflexsensor #2	
18		GND [Reflexsensor] #2	GN D bus s	Jordanslutning för jordreferens.	
19			USB [ATmega1284]]	Micro USB [USB modul]	Usb modul som omvandlar data från RPLIDAR XH2.54-5P/usb till RP3.
20			XH2.54-5P [USB modul]	XH2.54-5P [RPLIDAR]	Data som skickas från RPLIDAR till usb modul.
21			PC1 (SDA) [Pin 23, ATmega1284]	SDA [AdaFruit-10 DOF]	Skicka och ta emot data från Adafruit- 10 DOF via I2C bussen. Den är kopplad till SDA, vilket är den seriella dataremsan för I2C- kommunikationen.

22		PC1 (SCL) [Pin 22, ATmega1284]	SCL [AdaFruit-10 DOF]	Skicka och ta emot data via I2C bussen. Den är kopplad till SCL, vilket är den seriella klocklinjen för I2C-kommunikationen.
23		Vin [AdaFruit-10 DOF]	+5 V Buss	Inspänning till AdaFruit-10 DOF.
24		GND [AdaFruit-10 DOF]	GND buss	Jordanslutning för jordreferens.
2 5	PC2 (TCK) [Pin 24, ATmega1284]	-	JTAG: Gränssnitt för test, programmering och felsökning av integrerade kretsar.	
26		PC3 (TMS) [Pin 25, ATmega1284]	-	JTAG: Gränssnitt för test, programmering och felsökning av integrerade kretsar
27		PC4 (TDO) [Pin 26, ATmega1284]	-	JTAG: Gränssnitt för test, programmering och felsökning av integrerade kretsar
28		PC5 (TDI) [Pin 27, ATmega1284]	-	JTAG: Gränssnitt för test, programmering och felsökning av integrerade kretsar
29		PDO (RXD)	-	UART
30		PD1 (TXD)	-	UART

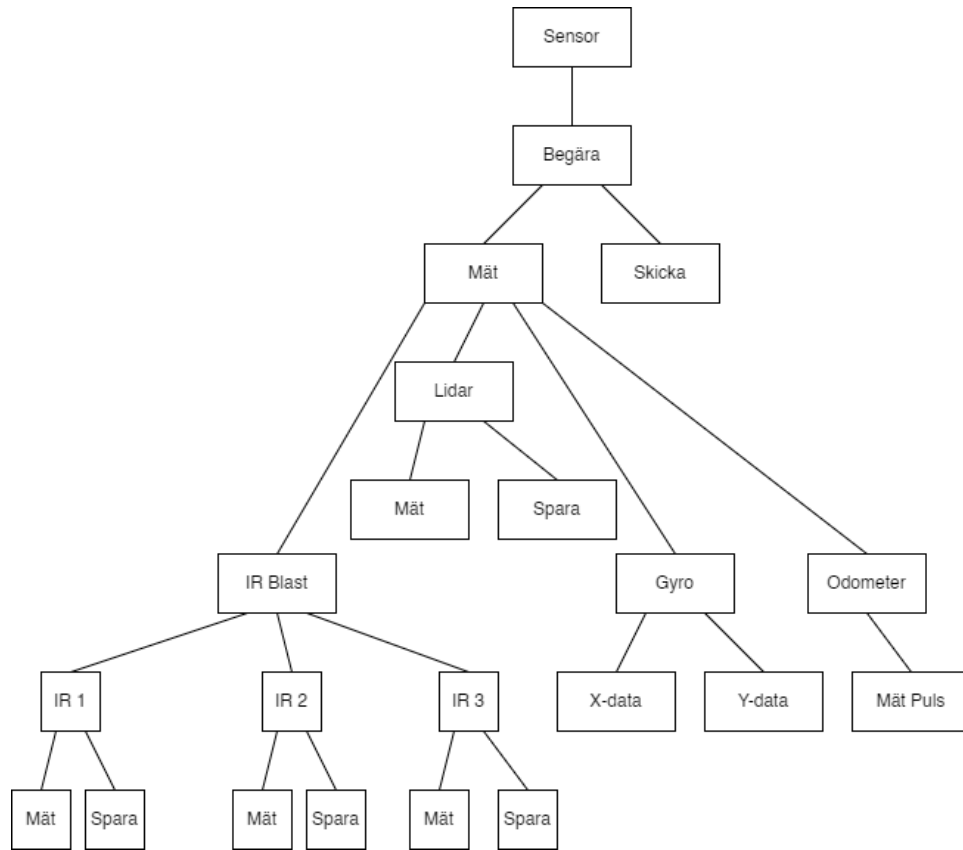
Tabell 6: Kopplingsstabell för sensormodulen

3.2.3 Komponenter

Komponenter	Antal
ATmega1284	1
GP2D120 IR (4–30 cm)	2
GP2Y0A21 IR (10–80 cm)	1
Reflexsensor	2
RPLIDAR A2	1
AdaFruit-10 DOF	1

Tabell 7: Komponenter i sensormodulen

3.2.4 Data- och programstruktur



figur 12: Data- och programstruktur för sensormodulen

3.3 Styrmodul

3.3.1 Modulens funktion

Styrmodulen är det delsystem som ser till att systemet förflyttas. Genom kommandon från huvudprocessorn regleras fyra växlade DC-motorer (7.2V, 291 RPM). Motorerna styrs parvis med två signaler per sida. DIR, som styr motorernas rotationsriktning och PWM, pulsbreddsmodulering som styr motorernas hastighet.

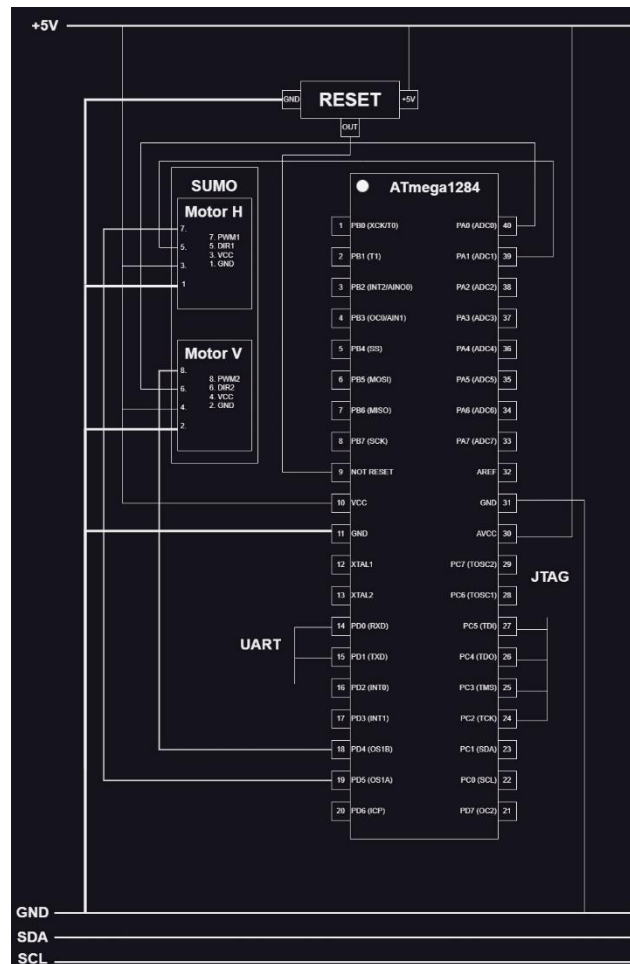
Motorparen styrs kan styras oberoende av varandra och tillsammans får de roboten att röra på sig.

Vid autonom styrning är det indata från sensormodulen som behandlas och skickas till huvudprocessorn varefter kommandon skickas till styrmodulen för styrning av enheten via en PID algoritm för att säkerhetsställa att systemet förflyttar sig enligt specifikation. Sensorer som kommer användas för att styra enheten är bland annat odometer för att räkna ut förflyttning samt Lidar för att räkna ut avstånd. Dessa sensorer hör dock till sensormodulen och kommunicerar inte direkt med styrmodulen.

Vid fjärrstyrning skickas kommandon från datorn till huvudprocessorn vidare till styrenheten för att reglera motorerna.

Styrmodulen kommer bestå av en AVR (ATmega1284) som kommer läsa och skicka data från övriga moduler.

3.3.2 Kopplingsschema



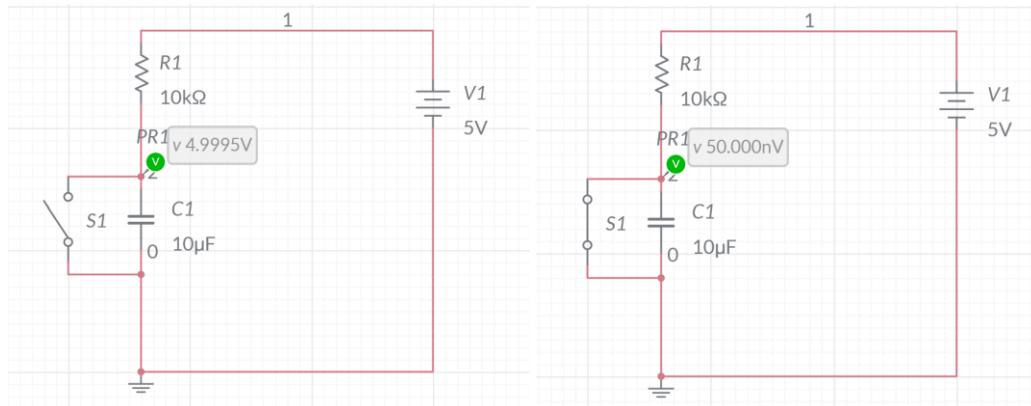
figur 13: Kopplingsschemat för styrmodulen

I tabellen nedan beskrivs kopplingarna i kopplingsschema och deras syfte i styrmodulen.

Nr.	Kopplingspunkt A	Kopplingspunkt B	Syfte
1	PB1 (T1) [Pin 2, ATmega1284]		Timer för autonomt läge med PID
2	NOT RESET [Pin 9, ATmega1284]	OUT [RESET krets]	Krets för att återställa modulen
3	VCC [Pin 10, ATmega1284]	-	Förser mikrokontrollenheten med den nödvändiga spänningen för att fungera korrekt.
4	GND [Pin 11, ATmega1284]	-	Jordanslutning för jordreferens
5	PD4 (OC1B) [Pin 18, ATmega1284]	Motor vänster	Reglera robotens hastighet med PWM
6	PD5 (OC1A) [Pin 19, ATmega1284]	Motor höger	Reglera robotens hastighet med PWM
7	PA0 (ADC0) [Pin 40, ATmega1284]	Motor vänster	Reglera robotens riktning med DIR
8	PA2 (ADC2) [Pin 38, ATmega1284]	Motor höger	Reglera robotens riktning med DIR
9	PC0 (SCL) [Pin 22, ATmega1284]	-	Skicka och ta emot data via I2C bussen. Den är kopplad till SCL, vilket är den seriella klocklinjen för I2C-kommunikationen.
10	PC1 (SDA) [Pin 23, ATmega1284]	-	Skicka och ta emot data via I2C bussen. Den är kopplad till SDA, vilket är den seriella dataremsan för I2C-kommunikationen.
11	PC2 (TCK) [Pin 24, ATmega1284]	-	JTAG: Gränssnitt för test, programmering och felsökning av integrerade kretsar

12	PC3 (TMS) [Pin 25, ATmega1284]	-	JTAG: Gränssnitt för test, programmering och felsökning av integrerade kretsar
13	PC4 (TDO) [Pin 26, ATmega1284]	-	JTAG: Gränssnitt för test, programmering och felsökning av integrerade kretsar
14	PC5 (TDI) [Pin 27, ATmega1284]	-	JTAG: Gränssnitt för test, programmering och felsökning av integrerade kretsar
15	PDO (RXD)	-	UART
16	PD1 (TXD)	-	UART

Tabell 8: Kopplingsstabell för styrmodulen



Figur 14: kopplingsschemat för RESET-kretsen

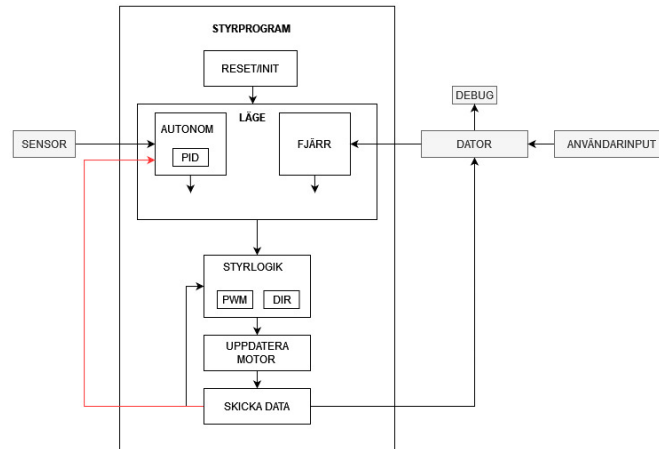
Kretsen RESET är en krets utformad för att återställa styrmodulen. Med spänningsmatning, jord, en resistor, kondensator och en brytare kan spänningen till ATmega1284 växla mellan 0V (approximativt) och 5V. Återställningsspinnen på ATmega1284 är *active low* vilket innebär att mikrokontrollern återställs när spänningen på RESET-pinnen dras ner eller förs till en logiskt låg nivå. När RESET-pinnen har en hög spänning är mikrokontrollern i sitt normala drifttillstånd. Komponentvärdena för kondensator och resistor är preliminära och kan ändras i slutprodukten.

3.3.3 Komponenter

Komponenter	Antal
ATmega1284	1
Reflexsensor	2
Sumo/Terminator	1
Kondensator	1 (delad)
Resistor	1 (delad)
Brytare	1 (delad)
Knapp	1 (delad)

Tabell 9: Komponenter i styrmodulen

3.3.4 Data- och programstruktur



Figur 15: Data- och –programstruktur för styrmodulen

- **RESET/INIT:** Programmet startar med ett RESET/INIT-block, som har till uppgift att initialisera systemet. Detta olika parametrar, variabler och komponenter ställs in för drift.
- **LÄGE:** Systemet övergår till blocket Läge, där driftläget bestäms. Det kan fungera i antingen "autonomt" eller "fjärrstyrt" läge.
 - **Fjärrstyrt-läge**
 - I läget "Manuell" bearbetas användarens indata. Användarens input kan vara instruktioner eller kommandon från en extern användare eller operatör.
 - Användarens indata bearbetas av datorn, vilket kan omfatta ytterligare bearbetning, validering och omvandling till motorstyrningskommandon.
 - Användarens input används sedan för att justera PWM- och DIR-inställningarna, vilket påverkar motorns drift.
 - **Autonomt-läge**
 - Autonomablocket ansvarar för att fatta kontrollbeslut, t.ex. att justera motor- eller ställdonsinställningarna för att svara på sensordata.
 - De data som samlas in från sensorerna matas in i Autonomablocket, som innefattar kontrollogik och bearbetning för att fatta beslut baserat på sensordata. I blocket finns en PID algoritm för att reglera systemet.
- **STYRLOGIK:** Stylogiken styr komponenterna "PWM" (Pulse Width Modulation) och "DIR" (Direction). PWM styr effektnivån eller hastigheten hos en motor, medan DIR styr dess riktning (t.ex. framåt eller bakåt).
- **UPPDATERA MOTOR:** Inställningarna "PWM" och "DIR" används för att uppdatera motorn, vilket resulterar i fysisk rörelse eller handling, baserat på de kontrollbeslut som fattas av systemet.
- **SKICKA DATA:**
 - Uppdaterade motorstyrningsdata skickas tillbaka till systemet, vilket återigen leder till systemets fysiska rörelse.
 - Systemet kan skicka data eller statusuppdateringar till den externa datorn för exempelvis övervakning, loggning eller felsökning.

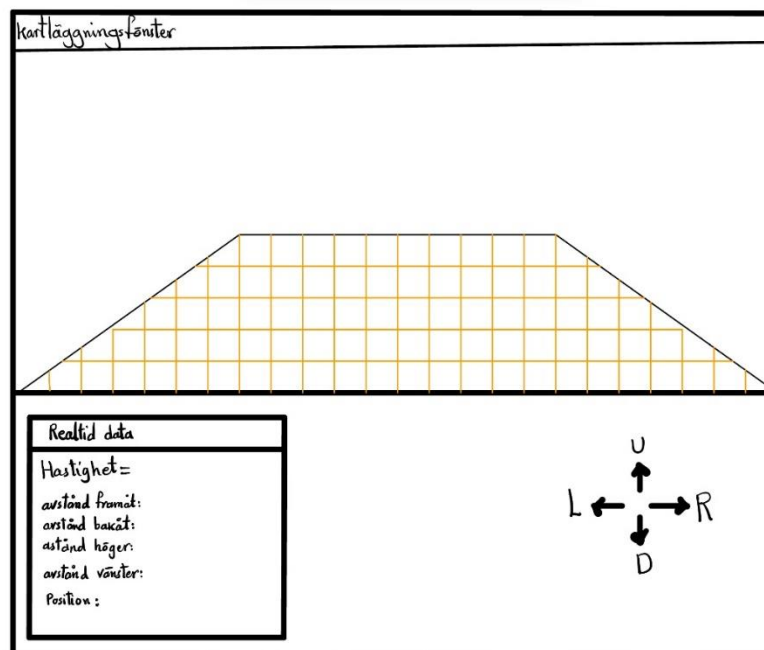
3.4 Gränssnitt

Gränssnittet är en GUI-applikation som kommer att implementeras i Python med hjälp av Tkinter-biblioteket. Gränssnittet är utformat som en uppdelad fönsterlayout enligt figur 16. Det består av två huvuddelar:

Kartläggningsfönstret (Övre delen): Det större fönstret i gränssnittet är avsett för kartläggning. Här kommer användaren att kunna visualisera den kartlagda miljön eller området. Detta omfattar visning av kartans struktur, upptäckta hinder och robotens nuvarande position. Kartvyn kommer att ta upp den mesta delen av fönstret och ge en tydlig översikt över området.

Realtidsdata och Styrning (Undre delen): Den nedre delen av gränssnittet är dedikerad till realtidsdata. På den vänstra sidan kommer användaren att kunna övervaka information som hastigheten på roboten och avståndet mellan roboten och objekt från olika riktningar. Detta ger användaren insikter om robotens prestanda och omgivningen i realtid. På den högra sidan av den nedre delen kommer det att finnas knappar och styrningsfunktioner. Dessa knappar gör det möjligt för användaren att interagera med roboten och styra dess rörelse, såsom att köra framåt, backa, svänga och stoppa.

Det uppdelade gränssnittet är utformat för att erbjuda både visuell information om kartläggningen och användarstyrning av robotens rörelse i en enhetlig och överskådlig layout.



Figur 16: Gränssnittet

4. Realiseringsstrategi

4.1 Strategi

Huvudstrategin är att ha ett test för varje krav. När vi lägger till en ny funktion ska den testas omedelbart för att se till att roboten håller kraven, därefter ska vi ha en helhetstest för att testa alla funktioner och krav.

4.2 Kritiska sektorer

De mest kritiska momenten är

- Koppla hela roboten
 - Se till att allt sitter ordentligt, att kopplingar inte glappar samt att systemet är kopplad enligt kopplingsschema.
- Gränssnittet är klart
 - Verifiera att det parametrar skickas och tas emot korrekt.
- Kommunikationsmodulen
 - Kommunikationsmodulen är hjärtat av systemet, om det inte fungerar, fallerar all kommunikation i systemet.

4.3 Tester

Roboten ska köra i en kvadratisk bana (2x2m). Roboten kommer starta i mitten av banan, köra ett varv runt banan och sedan återgå till startpunkten. Detta testar följande krav:

- Fungerande styralgorithm för att säkerställa att enheten inte slingrar sig när den kör autonomt
- Parametrar för styralgorithm från datorn och positionsdata från roboten skickas trådlöst till varandra
- Sensorer kan kalibreras
- LCD visar avståndet till väggarna
- Knapp för att starta fordonet
- Eheten kan återgå till sin startpunkt.

Olika styrlägen ska ändras under testen och diverse kommandon ska testas, exempelvis att enheten ska kunna rotera minst 90 grader och förflytta sig minst 30–40 cm. Detta testar kraven:

- Följande kommandon ska roboten reagera på: fram, fram V/H, back, stopp, rotera V/H
- Brytare för auto/fjärr

Referenser

Elektroniska källor

- [1] Nilsson, A. (2023) *TSEA29 Konstruktion med microdatorer, Fö6, Konstruktion med microdatorer, projektkurs (TSEA29 2023HT XX)*. Available at: https://liuonline.sharepoint.com/sites/Lisam_TSEA29_2023HT_XX/CourseDocuments/F%C3%B6rel%C3%A4sningar/TSEA29_F%C3%B66_23.pdf?CT=1696945710790&OR=ItemsView (Accessed: 05 October 2023).
- [2] "VanHeden," www.da.isy.liu.se. <https://www.da.isy.liu.se/vanheden/page/kommunikation/> (accessed Oct. 05, 2023).
- [3] "THE I 2 C-BUS SPECIFICATION VERSION 2.1 JANUARY 2000 document order number: 9398 393 40011 The I 2 C-bus specification CONTENTS." Accessed: Oct. 4, 2023. [Online]. Available: <https://www.da.isy.liu.se/vanheden/pdf/i2cspec2000.pdf>
- [4] "Features • High-performance, Low-power AVR ® 8-bit Microcontroller • Advanced RISC Architecture - 131 Powerful Instructions -Most Single-clock Cycle Execution -32 x 8 General Purpose Working Registers -Fully Static Operation -Up to 20 MIPS Throughput at 20 MHz -On-chip 2-cycle Multiplier • Nonvolatile Program and Data Memories -128K Bytes of In-System Self-Programmable Flash Endurance: 10,000 Write/Erase Cycles -Optional Boot Code Section with Independent Lock Bits In-System Programming by On-chip Boot Program True Read-While-Write Operation -4K Bytes EEPROM Endurance: 100,000 Write/Erase Cycles -16K Bytes Internal SRAM -Programming Lock for Software Security • JTAG (IEEE std. 1149.1 Compliant) Interface -Boundary-scan Capabilities According to the JTAG Standard -Extensive On-chip Debug Support -Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface • Peripheral Features -Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes -Two 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode -Real Time Counter with Separate Oscillator -Six PWM Channels -8-channel, 10-bit ADC Differential mode with selectable gain at 1x, 10x or 200x -Byte-oriented Two-wire Serial Interface -Two Programmable Serial USART - Master/Slave SPI Serial Interface -Programmable Watchdog Timer with Separate On-chip Oscillator -On-chip Analog Comparator -Interrupt and Wake-up on Pin Change • Special Microcontroller Features - Power-on Reset and Programmable Brown-out Detection -Internal Calibrated RC Oscillator -External and Internal Interrupt Sources -Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby • I/O and Packages -32 Programmable I/O Lines -40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF • Operating Voltages -1.8 -5.5V for ATmega1284P." Accessed: Oct. 11, 2023. [Online]. Available: <https://www.da.isy.liu.se/vanheden/pdf/atmega1284p.pdf>

