

# Designspecifikation

Grupp 6

2025-03-18

Version 1.0



## Status

Granskad	Andreas Nordström	2025-03-13
Godkänd	Namn	2025-xx-xx

**Beställare:**

Mattias Krysander, Linköpings universitet  
Telefon: +46 13282198  
E-post: mattias.krysander@liu.se

**Handledare:**

Theodor Lindberg, Linköpings universitet  
E-post: theodor.lindberg@liu.se

**Projektdeltagare**

Namn	Ansvar	E-post
Linus Funquist		linfu930@student.liu.se
Ebba Lundberg	Dokumentansvarig	ebblu474@student.liu.se
Andreas Nordström	Projektledare	andno773@student.liu.se
Sigge Rystedt		sigry751@student.liu.se
Ida Sonesson	Dokumentansvarig	idaso956@student.liu.se
Lisa Ståhl	Designansvarig	lisst342@student.liu.se

**INNEHÅLL**

1	Beställare	1
2	Inledning	1
3	Översikt av systemet	1
4	Sensorenhet	2
4.1	Komponenter	2
4.2	Kretsschema	2
5	Kommunikationsenhet	4
5.1	Komponenter	5
5.2	Blockschema	6
5.3	Mjukvara	6
6	Styrenhet	8
6.1	Komponenter	8
6.2	ATmega1284P	8
6.3	Körning av robotplattformen	8
6.4	Robotarmen	9
7	Persondator	11
7.1	Graphical User Interference, GUI	11
	Referenser	12

**DOKUMENTHISTORIK**

<b>Version</b>	<b>Datum</b>	<b>Utförda ändringar</b>	<b>Utförda av</b>	<b>Granskad</b>
0.1	2025-03-04	Första utkast	EL, AN, IS, LS, SR, LF	EL, AN, IS, LS, SR, LF
0.2	2025-03-12	Andra utkast	EL, AN, IS, LS, SR, LF	EL, AN, IS, LS, SR, LF
1.0	2025-03-13	Första version	IS, LS, SR	IS, LS, SR, LF

## 1 BESTÄLLARE

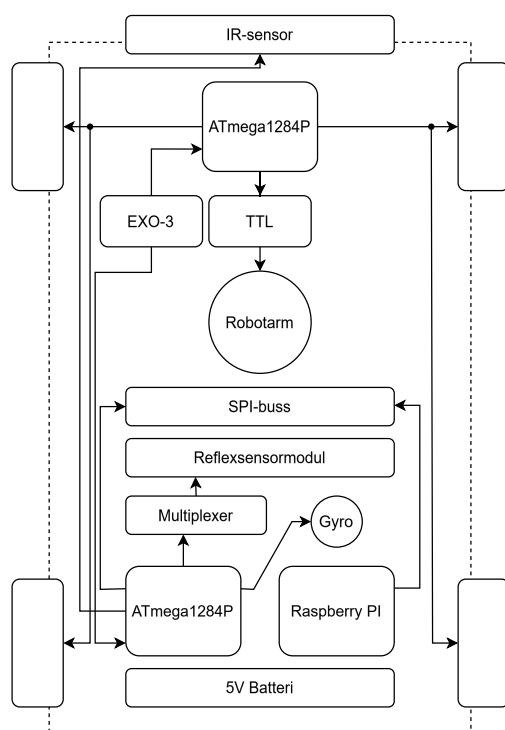
Beställare av projektet är Mattias Krysander på ISY, LiU.

## 2 INLEDNING

Detta dokument ligger till grund för projektets konstruktions- och implementationsfas. Här anges vilka komponenter som ska användas, kopplingsscheman för de olika enheterna samt grundläggande pseudokod.

## 3 ÖVERSIKT AV SYSTEMET

Systemet består av en sensor-, kommunikations- och styrenhet. Dessa kommer vara installerade på roboten som illustreras i figur 1. Följande kapitel beskriver respektive delsystem i detalj.



**Figur 1:** Blockschema över systemet.

## 4 SENSORENHET

Sensorenhetens uppgift är att samla in data som sedan skickas vidare till kommunikationsenheten. Sensordatan kommer skickas kontinuerligt från samtliga sensorer vilket utgör grunden till de styrbeslut som tas samt kartläggningen av lagermiljön.

### 4.1 Komponenter

I tabell 1 listas de komponenter som ingår i sensorenheten.

**Tabell 1:** Lista över komponenter.

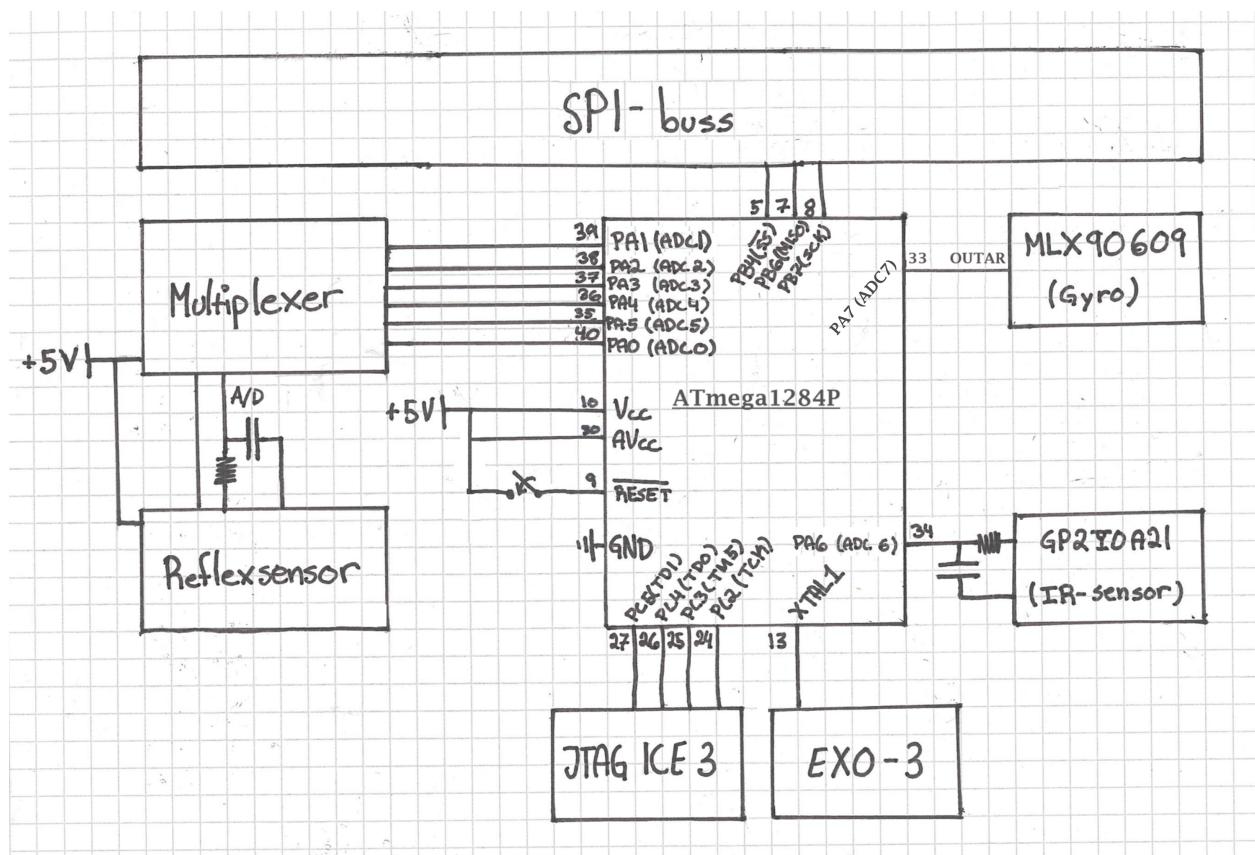
Komponent	Egenskap	Antal
Mikrodator	ATmega1284P	1
Emulator	JTAG ICE 3	1
Oscillator	EXO-3	1
Avståndssensor	GP2Y0A21, 10-80cm	1
Reflexsensormodul		1
Multiplexer	För reflexsensormodulen	1
Gyroskop	MLX90609	1
Resistor	18 kΩ, lågpåsfilter	2
Kondensator	100 nF, lågpåsfilter	2

### 4.2 Kretsschema

I detta kapitel presenteras kretsscheman för de olika delarna av sensorenheten. Notera att MLX90609 gyroskopet hade kunnat kopplas direkt till SPI-bussen men för att förenkla programmeringen har den istället kopplats till mikrotdatorn via ADC7 porten.

#### 4.2.1 ATmega1284P

Figur 2 visar en detaljerad uppkoppling för mikrotdatorn ATmega1284P där övriga delsystem är skissade. Multiplexern styrs via port 34-39 och A/D signalen från reflexsensormodulen lågpåsfiltreras innan den tas i emot i port 40 via adapttern. En JTAG ICE 3 är ikopplad via port 24-27 möjliggör programmering och simulering av mikrotdatorn. Ytterligare en EXO-3 är ikopplad i port 13 för att styra mikrotdatorns interna klocka. IR-sensorn GP2Y0A21:s A/D utsignal är likt reflexsensorn lågpåsfiltrerad och kopplad till port 39. I port 9 är en RESET inkopplad. MLX90609 gyrot ikopplad till mikrotdatorn via ADC7.



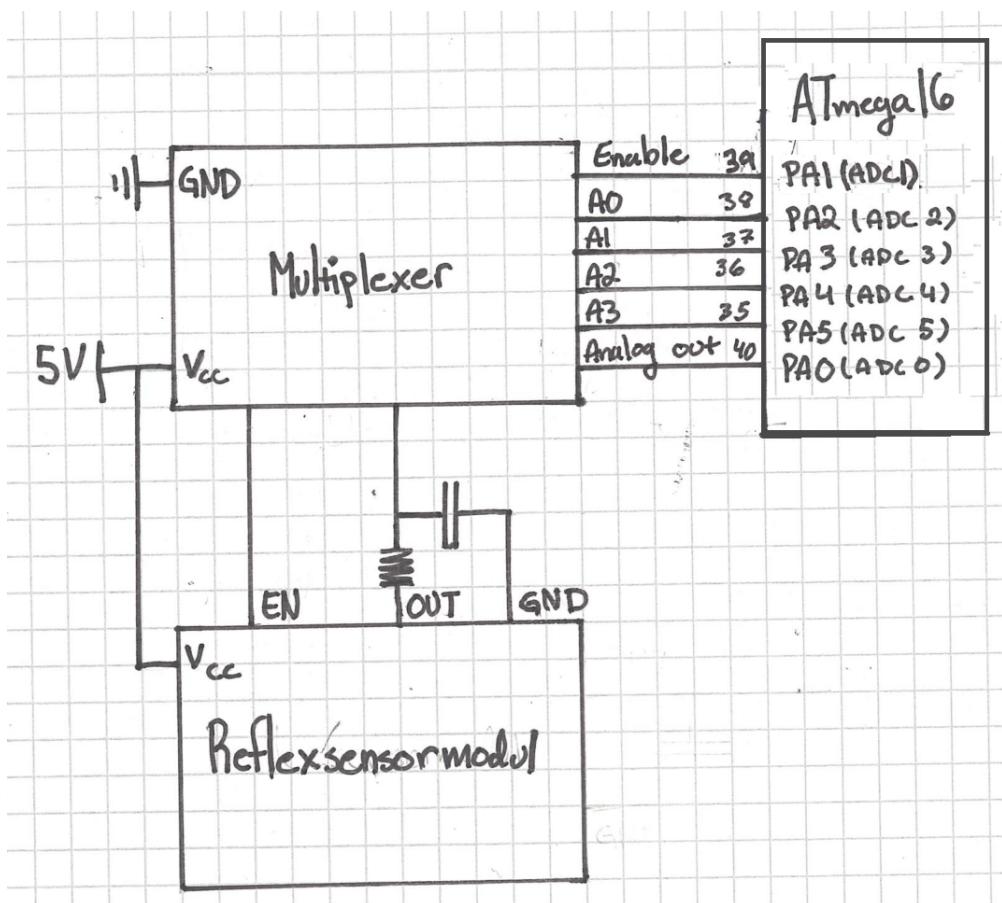
Figur 2: Kretsschema för ATmega1284P.

#### 4.2.2 Reflexsensormodul

Som illustreras i figur 3 styrs och avläses reflexsensormodulen med mikrodatorn genom en adapter som har en inbyggd multiplexer. A0-3 är kopplad till A-porten och väljer vilken reflexsensor som Enable kan aktivera. Analog out ger den lågpassfiltrerade A/D-signalen från reflexsensormodulens OUT. För att kunna avgöra om roboten ska svänga höger eller vänster genomförs här en tyngdpunktsberäkning för att ta fram data på avvikelsen från mitten:

$$k_T = \frac{\sum_k m_k k}{\sum_k m_k} \quad (1)$$

där  $k_T$  och  $m$  är tyngdpunkten respektive magnituden som en reflektor detekterar. Denna data skickas till kommunikationsenheten där själva regleringen utförs.



**Figur 3:** Kretsschema för reflexsensormodulen och multiplexern.

#### 4.2.3 Gyro

Gyrots utsignal är en analog vinkelhastighet som är kopplad till AVR-mikrodatorns ADC-omvandlare. Då styrenheten arbetar med vinklar måste vinkelhastigheten integreras över tid för att omvandla denna till en vinkel. Vid integration är tidssteget viktigt, det kan därför vara bra att sampla med en bestämd frekvens med hjälp av hårdvarutimers.

## 5 KOMMUNIKATIONSENHET

Kommunikationenhetens sköter all intern och extern kommunikation. Den ska ta emot data från styr- och sensorenheten och överför sedan information till PC:n för att kartlägga var roboten befinner sig. Kommunikationenhetens ska även kunna ta emot data via bluetooth från PC:n avsedd för styrenheten för manuell styrning. Vid autonom körning ska kommunikationenheten, med hjälp av data från resterande moduler, kunna ta beslut om kortaste väg.

## 5.1 Komponenter

Nedan följer en lista på de komponenter som ingår i kommunikationsenheten:

- Raspberry Pi
- Nivåskiftare

### 5.1.1 SPI-buss

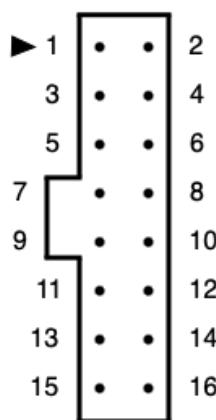
SPI är det bussprotokoll som kommer att användas för kommunikation mellan enheterna. Den kommer att kopplas via nivåskiftaren från Raspberry Pi:n och sedan vidare ut till styr- och sensorenheten och deras respektive AVR:er.

### 5.1.2 Raspberry Pi

Kommunikationsenhetens mikrodator kommer att vara en Raspberry Pi som kopplas samman med styr- och sensorenheten samt PC:n.

### 5.1.3 Nivåskiftare

Mellan Raspberry Pi:n och SPI-bussen behövs det kopplas en nivåskiftare, se figur 4, eftersom att dessa komponenter använder sig av olika spänning-logik. Här är det viktigt att sätta nivåskiftaren så att den täcker Raspberry Pi:n helt. Koppling för respektive pin visas i tabell 2.



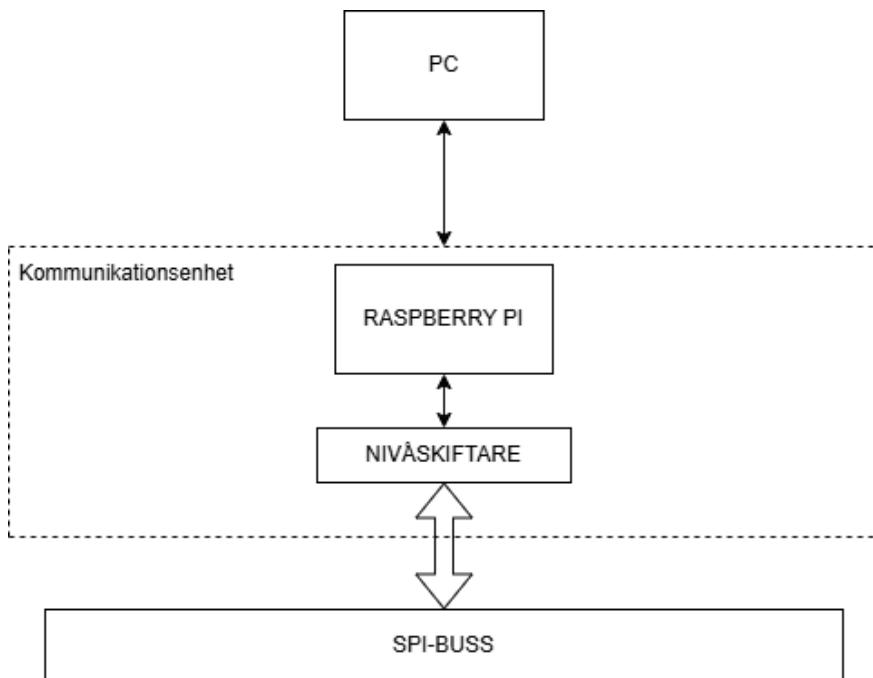
**Figur 4:** Nivåskiftare med 16 pinnar.

**Tabell 2:** Pin med respektive koppling för nivåskiftare.

<b>Pin</b>	<b>Koppling</b>	<b>Pin</b>	<b>Koppling</b>	<b>Tecken</b>	<b>Förklaring</b>
1	Gnd	2	Gnd	→	Raspberry → AVR
3	N/C	4	N/C	←	Raspberry ← AVR
5	I2C_SCL ↔	6	SPI_MOSI →	↔	Raspberry ↔ AVR
7	I2C_SDA ↔	8	SPI_MISO ←		
9	N/C	10	SPI_SCLK →		
11	N/C	12	SPI_CE0 →		
13	GPIO27 ←	14	SPI_CE1 →		
15	GPIO17 ←	16	GPIO22 →		

## 5.2 Blockschema

Nedan följer ett blockschema för kommunikationsenheten, se figur 5, med komponenterna beskrivna i avsnitt 5.1.

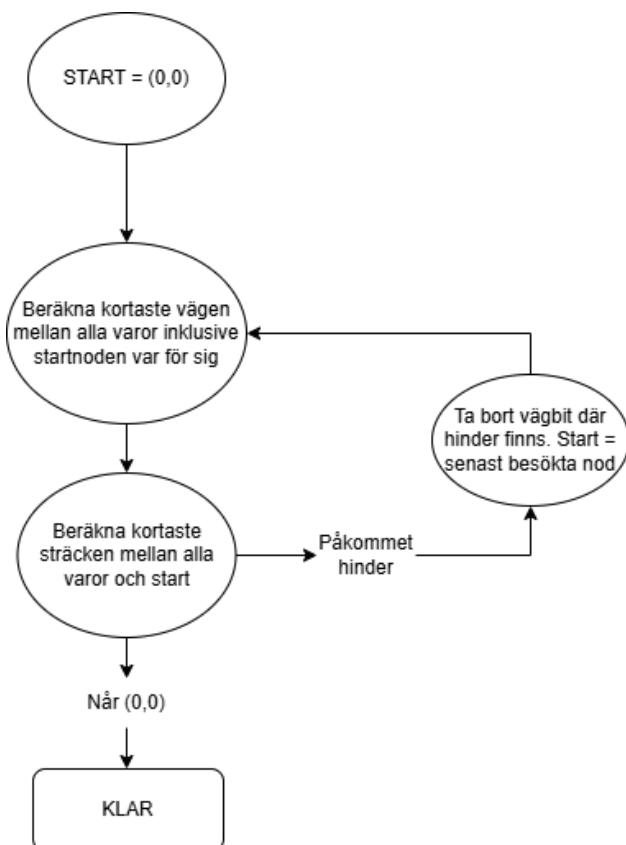
**Figur 5:** Blockschema för kommunikationseinhet.

## 5.3 Mjukvara

Nedan diskuteras mjukvaran som finns i kommunikationseinheten.

### 5.3.1 Pseudokod snabbaste vägen

Koden för att hämta alla varor i en körning kommer följa pseudokoden enligt 6. Varje korsning, hörn och vara anges som en nod och mellan varje nodpar finns en kant där alla kanter i detta fall har samma vikt (kostnad). Roboten startar alltid i ett av rutnätets hörn och anger den noden som (0,0), därifrån beräknas snabbaste vägen mellan alla två varor samt mellan alla varor och starten. När detta beräknats jämförs alla vägar där alla varor och start ingår och den kortaste vägen väljs. Om hinder upptäcks när körningen pågår förbjuds kanten där hindret ligger sedan sätts den senast besökta noden till startnod och algoritmen körs från början. När roboten sedan når nod (0,0) med alla varor hämtade är algoritmen klar.



**Figur 6:** pseudokod snabbaste vägen.

### 5.3.2 Regleralgoritm för tejpföljning

Kommunikationsenheten kommer även att hantera regleringen av roboten. Sensordata tas emot av sensorenheten och styrbeslut skickas sedan till styrenheten. I autonomt läge kommer regleralgoritmen, med hjälp av data från en tejpsensor under roboten att följa tejpbitten rakt tills det kommer ett nytt styrbeslut.

För autonom tejpföljning kommer PD-reglering att utnyttjas, se ekvation 2.  $K_P$  och  $K_D$  är konstanter som kommer

bestämmas experimentellt. Vid tejpföljning är målet att tejpen ska vara i mitten av tejpsensorn, och felet ( $e(t)$ ) blir något mått på avvikelsen från mitten.

$$u(t) = K_P e(t) + K_D \frac{d}{dt} e(t) \quad (2)$$

Kommunikationsenheten beräknar de sluttgiltiga signalerna som ska skickas till motorer eller servon, och skickar dem till styrenheten som skickar vidare dem dit de ska.

## 6 STYRENHET

Styrenhetens uppgift är att styra motorerna och robotarmen utifrån styrdata från kommunikationsenheten.

### 6.1 Komponenter

I tabell 3 nedan listas komponenterna som ingår i styrenheten.

**Tabell 3:** Lista över komponenter.

Komponent	Egenskap	Antal
Mikrodator	ATmega1284P	1
Robotarm	PhantomX Reactor	1
Hjul		4
Oscillator	EXO3	1
Resistor	10 kΩ	1

### 6.2 ATmega1284P

Kärnan av styrenheten är mikrodatorn, som är en Atmel ATmega1284P. Den kommer att vara kopplad till kommunikationsenheten via en SPI-buss, där kommunikationsenheten agerar master.

### 6.3 Körning av robotplattformen

Robotplattformens fyra hjul går att kontrollera som två hjulpar, vänster respektive höger, och styrs via en DIR och en PWM-signal från mikrodatorn (se kopplingsschema). DIR-signalen bestämmer vilket håll hjulen ska snurra åt, och PWM-signalerna bestämmer hur snabbt de ska snurra. PWM är en periodisk signal som antingen är hög eller låg, den andelen av periodtiden som signalen är hög brukar kallas signalens duty-cykel, och en hög duty-cykel ger snabbare snurr på hjulen. Då robotens olika sidor styrs separat från varandra går det att få den att svänga genom att lägga olika mycket kraft i respektive sida. Styrsignaler skickas till styrenheten från kommunikationsenheten, antingen resultatet av PD-regleringsalgoritmen som tar in data från sensorenheten, eller input från användaren.

## 6.4 Robotarmen

För att kontrollera robotarmens servon används UART som finns på pins på mikrodatorn. Däremot måste signalerna gå genom en TTL-krets, se figur 8 som ser till att det bara går signaler åt ett håll i taget, då servona kräver att UART-signalerna konverteras till halvduplex. Kommunikation med alla servon sker via samma kontakt, och i datan som skickas adresseras det relevanta servot. I servona finns det mycket data som går att läsa av och många parametrar som går att bestämma. Bland annat går målvinkel och vinkelhastighet att bestämma för respektive servo så att armen kan ställas i rätt position, och så att alla servon är färdiga med sin vridning samtidigt så att armen som helhet rör sig i en jämn rörelse.

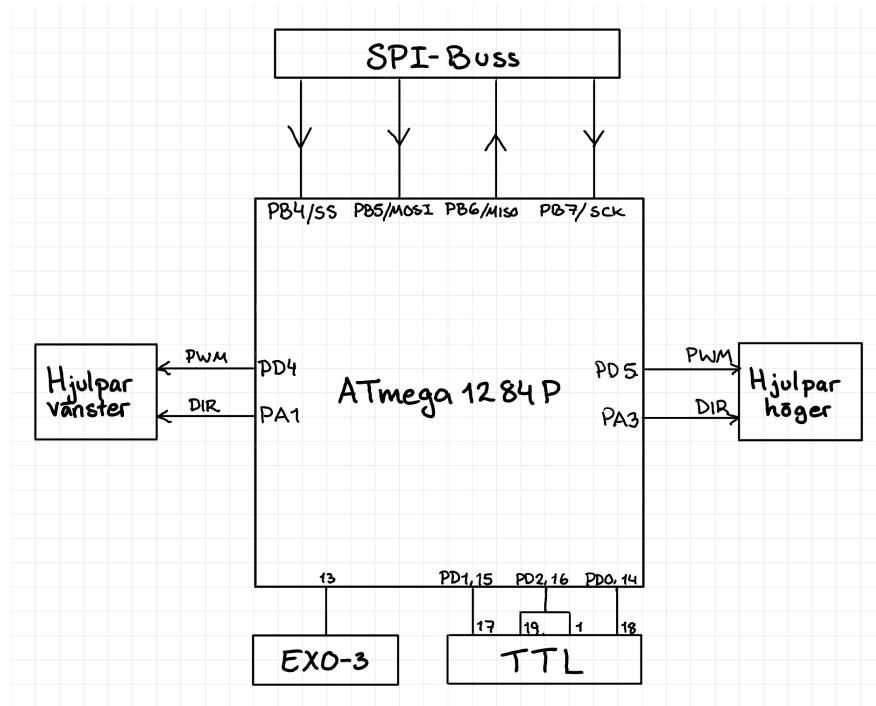
### 6.4.1 Autonom styrning av robotarmen

För att styra robotarmen autonomt kommer framförallt en sak att behöva tas reda på: var är objektet som ska plockas upp relativt robotarmens bas. Om detta är känt så kan vinklarna som robotarmens servon ska ställas in på beräknas med hjälp av invers kinematik. Detta går antingen att göra analytiskt, eller numeriskt. Dessa beräkningar kommer troligen göras i kommunikationsenheten. Dock, givet att robotten har placerat sig rätt, så kommer objektet alltid att vara placerat på samma plats relativt robotten och det skulle alltså gå att hårdkoda själva plockrörelsen. Om rörelsen hårdkodas så behövs inte invers kinematik eller ytterligare beräkningar.

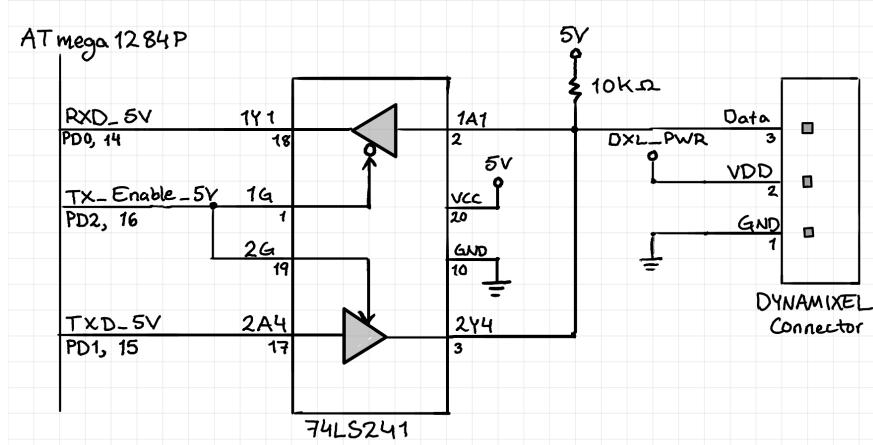
### 6.4.2 Manuell styrning av robotarmen

Manuell styrning av robotarmen skulle kunna ske på två sätt. Antingen styrs ett servo i taget, eller så matar användaren in armens generella riktning, och så får det beräknas hur servona ska vridas. Att styra ett servo i taget implementeras enkelt, men det andra är inte lika lätt. Ett sätt att lösa det på skulle kunna vara att definiera ett minsta avstånd armen kan röras, och medan användaren exempelvis styr armen uppå flyttas armens målposition upp med bestämd frekvens. Samtidigt beräknas också servonas nya vinklar med inverskinematik och skickas kontinuerligt till robotarmen, även detta med bestämd frekvens. Detta beräknas nog också i kommunikationsenheten.

Nedan följer blockschema för styrenheten, och ett mer detaljerat kopplingsschema för kretsen mellan AVR och robotarm.



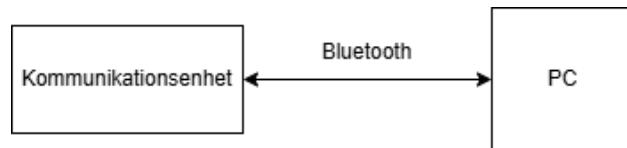
**Figur 7:** Blockschema över styrenheten



**Figur 8:** Kopplingsschema för TTL kretsen mellan AVR och robotarm

## 7 PERSONDATOR

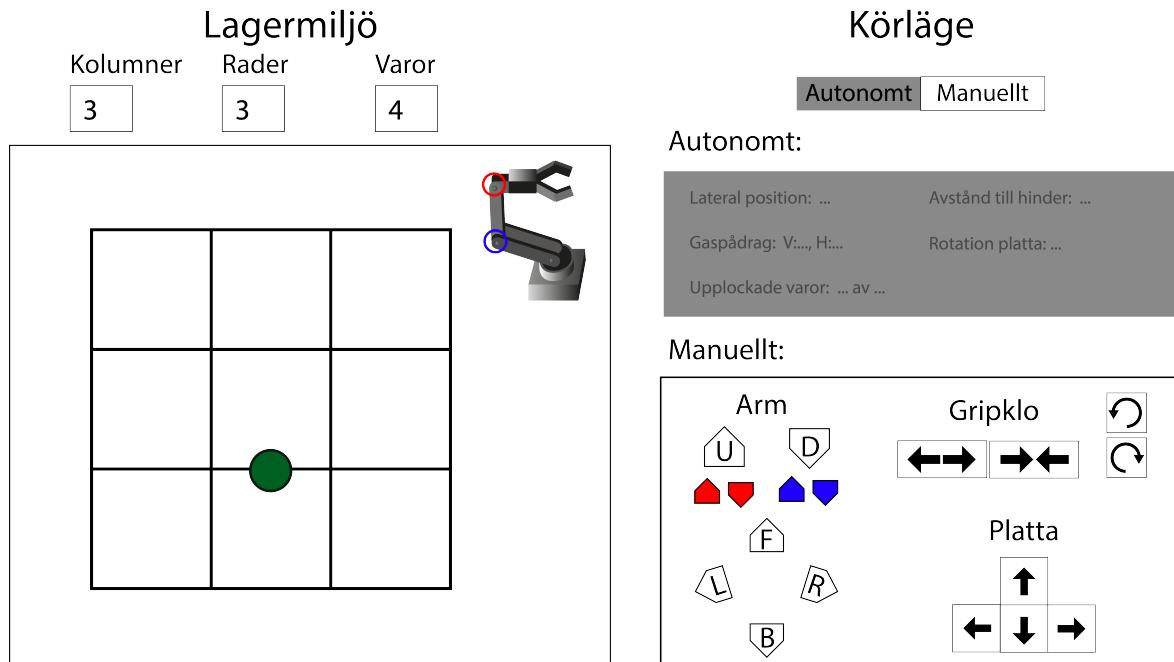
Kommunikationen mellan persondatorn och roboten kommer att ske via bluetooth-länk, se figur 9. Via persondatorn ska det, med hjälp av denna kommunikation, kunna genomföras manuell styrning via GUI:n, se figur 10. Oavsett läge ska även persondatorn kunna ta emot data från roboten för att ge en uppfattning om kartläggning och andra uppgifter såsom exempelvis styrmotorernas gaspådrag.



**Figur 9:** Blockschema över persondatorn.

### 7.1 Graphical User Interference, GUI

GUI:n kommer att programmeras i ett språk såsom antingen Python, C eller C++. För att förenkla processen kommer ramverket Qt att användas. Qt är kompatibelt med flera plattformar och kommer att göra det enklare att programmera GUI:n då den minskar antalet kodrader som behövs, och är främst framtaget för utveckling av GUI.



**Figur 10:** Översikt över GUI.

## REFERENSER

- [1] *Multiplexer till sensorskenan* (2025), Melexis. Hämtad: 2025-03-04, [Online]. Tillgänglig: <https://da-proj.gitlab-pages.liu.se/vanheden/pdf/mlx90609.pdf>.