

# Förstudie om kommunikationsenheten i en lagerrobot

Ebba Lundberg  
Ida Sonesson

28 april 2025

Version 1.1



## Status

Granskad	Ida Sonesson, Ebba Lundberg	2025-04-28
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**

<b>Namn</b>	<b>Ansvar</b>	<b>E-post</b>
Linus Funquist		linfu930@student.liu.se
Ebba Lundberg	Dokumentansvarig	ebblu474@student.liu.se
Andreas Nordström	Projektledare	andno7733@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	Inledning	1
1.1	Syfte . . . . .	1
1.2	Avgränsningar . . . . .	1
2	Problemformulering	1
3	Kunskapsbas	2
3.1	Raspberry Pi . . . . .	2
3.2	Trådlös kommunikation . . . . .	2
3.3	Seriell kommunikation . . . . .	3
4	Diskussion	9
4.1	Trådlös kommunikation . . . . .	9
4.2	Seriell kommunikation . . . . .	9
5	Slutsatser	10

## DOKUMENTHISTORIK

Version	Datum	Utförda ändringar	Utförda av	Granskad
0.1	2025-02-24	Första utkast	Ida Sonesson, Ebba Lundberg	Ida Sonesson, Ebba Lundberg
1.0	2025-04-14	Första version	Ida Sonesson, Ebba Lundberg	Ida Sonesson, Ebba Lundberg
1.1	2025-04-28	Första version med ändringar	Ida Sonesson, Ebba Lundberg	Ida Sonesson, Ebba Lundberg

## 1 INLEDNING

Detta dokument är en förstudie i kursen TSEA56, Elektronik kandidatprojekt, och kommer att handla om kommunikationsenheten i den lagerrobot som ska tas fram. Kommunikationsenheten syftar till den del av roboten som sköter kommunikationen mellan de olika delsystemen som ingår i roboten. De delsystem som roboten består av är kommunikationsenheten, sensorenheten, styrenheten samt en persondator, och det är endast kommunikationsenheten som kommer att behandlas. I denna förstudie kommer olika kommunikationsmetoder mellan delsystemen att studeras för att få en djupare förståelse kring hur dessa fungerar och för att få kunskap om vilka kommunikationsmetoder som är bäst lämpade för roboten.

### 1.1 Syfte

Syftet med denna förstudie är att få en djupare förståelse för olika kommunikationsmetoder mellan de olika delsystemen som roboten består av, och med denna kunskap senare resonera kring vilka kommunikationsmetoder som passar roboten bäst.

### 1.2 Avgränsningar

I förstudien diskuteras endast de kommunikationsmetoder som finns tillgängliga i kursen TSEA56, dessa är Wi-Fi och Bluetooth som är inbyggda i en Raspberry Pi samt, I2C, SPI och UART.

## 2 PROBLEMFORMULERING

Det finns olika sätt att kommunicera mellan olika processorer, och följande frågeställningar kommer att besvaras för att få en större förståelse för vardera:

- Vilka olika principer finns det för kommunikation mellan olika processorer och hur fungerar dessa?
- Vad har de olika principerna för fördelar och/ eller nackdelar och hur fungerar dessa i lagerroboten?

### 3 KUNSKAPSBAS

Information och fakta hämtas främst från vetenskapliga artiklar. Vanheden kommer att användas som en faktabas för vidare informationssökning. Nedan följer fakta kring olika ämnen relaterade till kommunikation för lagerroboten.

#### 3.1 Raspberry Pi

Raspberry Pi är en multifunktionell interaktiv dator byggd på ett enda kretskort. I denna kurs kommer en serie kallad Raspberry Pi 3B+ att vara tillgänglig. Denna dator har bland annat flera olika I/O-portar som exempelvis HDMI, ethernet och USB. Till lagerroboten kommer denna kommunikationsenhet kopplas till en trådlös enhet, i detta fall PC, med hjälp av antingen Bluetooth eller Wi-Fi. Detta kommer möjliggöra autonom körning och överföring av data [1].

Det finns också GPIO-portar som gör det möjligt för Raspberry Pi:n att kopplas samman med en elektrisk krets, vilket möjliggör flera olika funktioner [2]. Raspberry Pi:n kommer monteras på robotplattformen och ha trådad kommunikation med sensor- och styrenheterna. Raspberry Pi erbjuder kommunikation via flertalet principer, till exempel via I2C, SPI och UART, och det är dessa som kommer att diskuteras vidare.

#### 3.2 Trådlös kommunikation

Dataöverföringen mellan Raspberry Pi och PC:n kommer att ske trådlöst. Raspberry Pi, som nämnts ovan, kan kommunicera över både Wi-Fi och Bluetooth och nedan följer fakta samt fördelar och nackdelar för vardera.

##### 3.2.1 Wi-Fi

Wi-Fi är en trådlös nätverksteknik som används globalt. Den gör det möjligt för användare att koppla upp olika enheter på internet. Wi-Fi är ett varumärke som använder flera IEEE 802.11-standarder (IEEE Std 802.11) [3].

Wi-Fi har sitt ursprung från sent 90-tal då trådlös uppkoppling blev ett "revolutionärt koncept" och har sedan dess haft en drastisk utveckling. Teknologin har utvecklats under flera år och genomgått flera ändringar, exempelvis från 802.11b till 802.11ax (Wi-Fi 6). För varje modell har den nyare innefattat en förbättring inom flertalet egenskaper, såsom kapacitet, pålitlighet m.fl. Det enda som förblivit detsamma är namnet Wi-Fi som med sin enkelhet i både uttal och stavning blivit ett effektivt varumärke som idag speglar den fria uppkopplingen [4].

För att ett trådlöst nätverk ska kunna koppla upp sig mot internet används radiovågor, detta gäller även för tekniken bakom Wi-Fi. Syftet med denna trådlösa teknik är att direkt via ett mobilt bredband eller datornätverk, utan fast uppkoppling, kunna ansluta till internet. En så kallad router används för att det trådlösa datornätverket ska kunna kommunicera med internet [5].

Den trådlösa enheten, exempelvis en dator, kommunicerar som nämnt tidigare genom att sända radiovågor. Styrkan på radiovågorna är ungefär densamma som på mobiltelefoner och enligt standard på maxstyrkan så får ett datornätverk som max alstra 0,1 watt för radiovågor med frekvensen 2,45 GHz och 0,2 watt för frekvensen 5 GHz. Desto större avstånd från sändaren desto mindre styrka på radiovågen [5].

Då Wi-Fi uppfyller många fördelar för uppkoppling och delning av data så finns det risker med dess användning.

Beroende på dess säkerhet finns det risk att exempelvis angripare eller hackare kopplar upp sig och utnyttjar detta för att utföra illegala saker på internet eller även orsaka att ägarens personliga information blir offentlig. Data kan därmed bli tillgänglig för offentligheten om nätverket ej hanteras med säkerhet genom att exempelvis införa ett lösenord [6].

### 3.2.2 Bluetooth

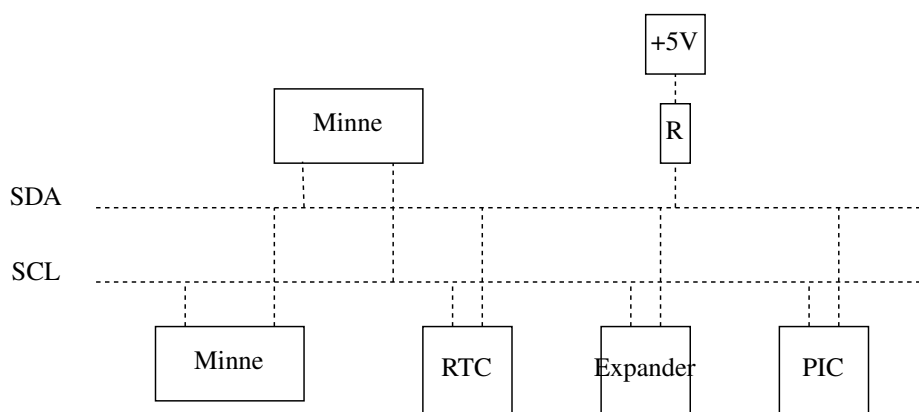
Bluetooth, eller Blåtand, utvecklades i Sverige i slutet av 90-talet och har kommit att bli standard för trådlös kommunikation på kort avstånd. Bluetooth har en relativt kort räckvidd på cirka 10 meter, och eftersom det är radiokommunikation som används så fungerar Bluetooth även om det finns hinder mellan mottagare och sändare av signalen. Här skiljer sig Bluetooth mot kommunikation som inte använder sig av radiovågor, som till exempel infraröda signaler, eftersom signalen där inte kommer fram om den inte har en fri väg mellan sändare och mottagare [7]. Bluetooth använder sig av 2.4 GHz [8].

## 3.3 Seriell kommunikation

En Raspberry Pi använder sig av 3.3V-logik, medan samtliga virkort i projektet endast har 5V TTL-logik. För att möjliggöra kommunikation mellan Raspberry Pi:n och AVR mikrokontrollerna (Alf and Vegard's RISC) krävs därför något för att konvertera nivån. Ett första alternativ, men också det vanligaste, är användning av UART via USB för att lösa spänningsnivån. Ett annat är att med en inkopplad nivåkonverterande krets använda I2C eller SPI via GPIO. Nedan följer fakta kring respektive alternativ [9].

### 3.3.1 Inter Integrated Circuit (I2C)

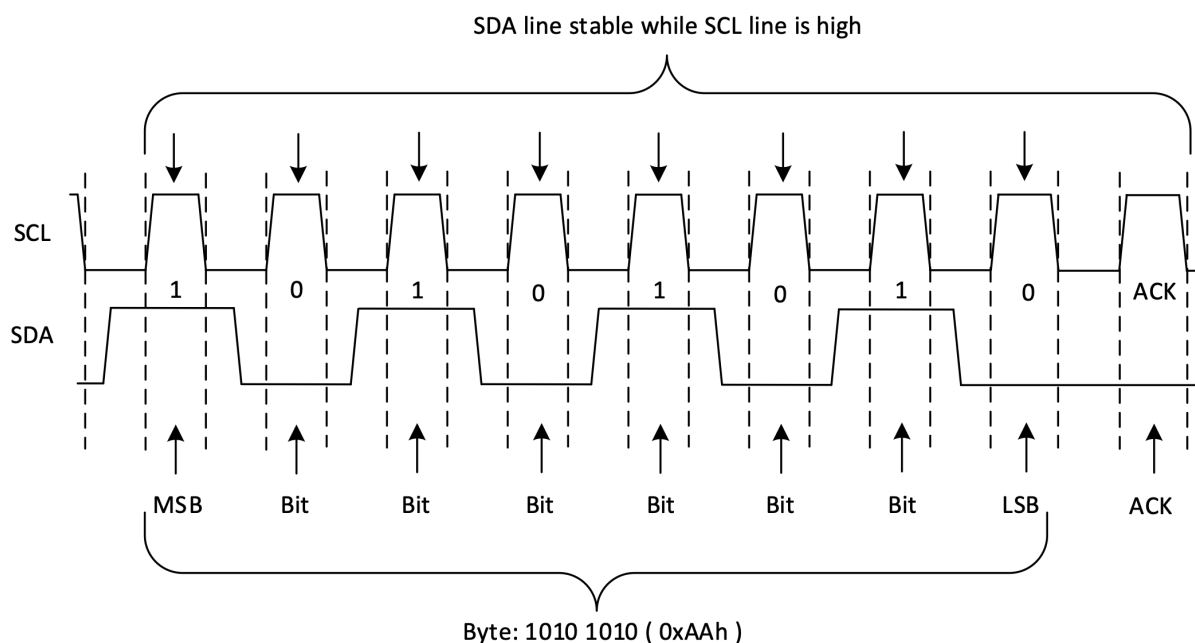
Inter Integrated Circuit, I2C, är halv-duplex och ett sätt för kommunikation via seriell buss mellan IC-kretsar. Halv-duplex innebär att kommunikation kan ske i båda riktningarna men bara en riktning i taget. En I2C-buss består enbart av två ledningar, SDA (Serial Data) och SCL (Serial CLock), till vilka man kan ansluta sina I2C-kretsar vid godtyckligt ställe på bussen, se Figur 1. Till SDA-ledningen krävs även ett så kallat pull-up-motstånd med ett standardvärde på ungefär 4-10 k $\Omega$ . Det som avgör storleken är antalet I2C-kretsar på bussen samt vilken datatakt som används. Datatakten anges av SCL-signalen och det är den så kallade masterkretsen på bussen som avgör datatakten. Till skillnad från en klocksignal så kan SCL-signalen variera i frekvens och ändrar tillstånd endast då data skickas på ledningen [10].



**Figur 1:** Översikt av I2C-buss.

På SDA-ledningen skickas endast data mellan två kretsar åt gången. Data skickas från sändaren och skickas till mottagaren. Datan från sändaren skickas alltid med 8 bitar varav riktningen skiftas därefter på SDA-ledningen och mottagaren bekräftar dataåtkomsten med att skicka en ACK-bit (ACKnowledge). Då sändaren tagit emot ACK-biten kan ytterligare 8 bitar data skickas från sändaren. För varje databit som överförs på SDA-ledningen behövs en motsvarande klockpuls på SCL-ledningen. Detta medför att en hel byte kräver totalt 9 klockpulser – åtta för databitarna och en för ACK-signalen [10].

Kommunikation mellan master och slave fungerar på så sätt att mastern först skickar ett START-villkor, och avslutas med att mastern skickar ett STOPP-villkor [11]. Ett START-villkor karakteriseras av en hög till låg förändring på SDA-ledningen samtidigt som SCL-ledningen är hög. Ett STOPP-villkor karakteriseras tvärtom av en låg till hög förändring på SDA-ledningen samtidigt som SCL-ledningen är hög. Endast en bit skickas per klockpuls av SCL. En byte, alltså 8 bitar, på SDA-ledningen kan antingen vara, till exempel, adressen till en viss enhet, en adress till ett register eller data från en slav som ska läsas. Data skickas med mest signifikant bit (MSB) först, och vilket antal bytes som helst kan skickas från master till slav mellan START- och STOP-villkor [11]. I Figur 2 visas hur dataöverföringen kan se ut.



**Figur 2:** Dataöverföring I2C [11].

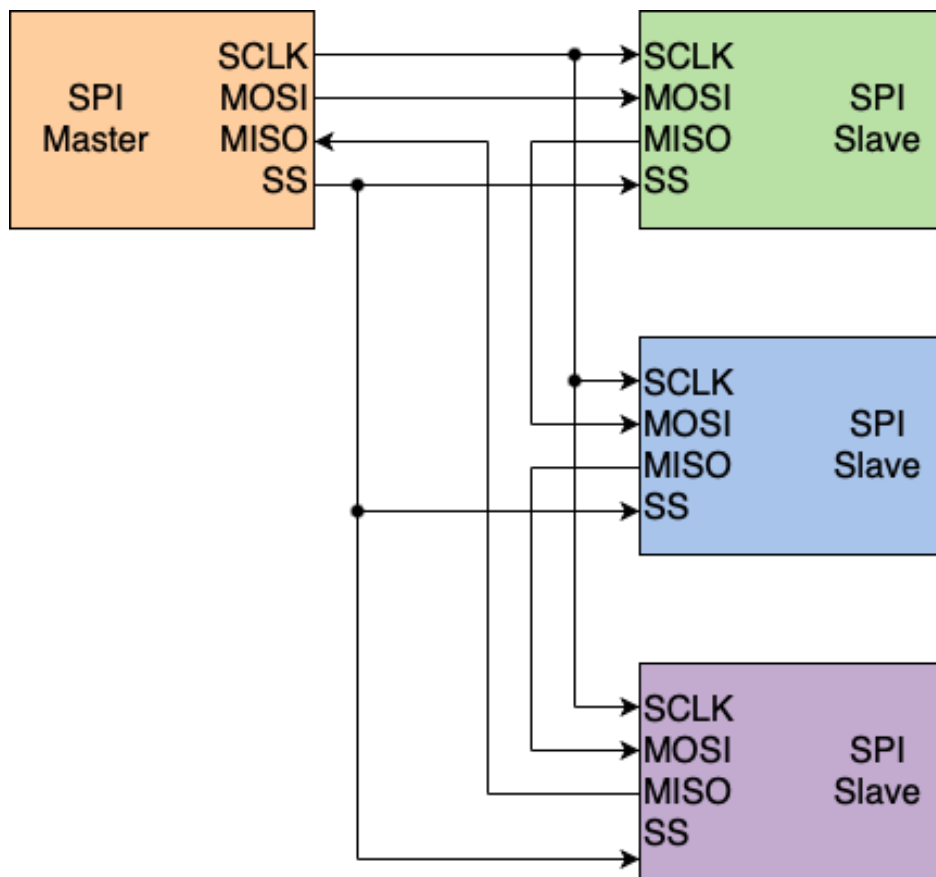
I2C-protokollet gör det möjligt att lätt ansluta flera externa enheter som exempelvis sensor, motorstyrningar och displayer. Flera enheter kan anslutas eftersom I2C är en adresserbar buss, vilket gör det möjligt att skicka data till vald enhet med hjälp av adressfältet. I2C-protokollet erbjuder stor flexibilitet då den endast kräver ett fåtal kablar [12]. Övriga fördelar med en I2C-buss är bland annat dess förmåga att bibehålla ett lågt antal signaler oavsett antalet enheter på bussen. Den inkluderar även en funktionalitet kallad ACK/NACK som förbättrar felhanteringen [13]. Med I2C-protokollet följer dock även nackdelar såsom ökad komplexitet i ”firmware” eller lågnivåhårdvara.



### 3.3.2 Serial Peripheral Interface (SPI)

Serial Peripheral Interface, SPI, är ett slags kommunikationsprotokoll där full-duplex dataöverföring är möjlig. Full-duplex innebär att kommunikation kan ske i båda riktningar samtidigt. SPI används för kommunikation mellan en enchipdator eller AVR och sensorer eller liknande enheter, och bygger på ett system av masters och slaves. En master är typiskt en enchipdator som kontrollerar hela kommunikationsprocessen och datautbytet. Slave-delen av en SPI består av en eller flera enheter som reagerar på masterns kommandon. SPI är ett effektivt kommunikationssätt med hög hastighet som även är relativt lätt att använda eftersom det inte behöver användas någon adressering [14]. Det finns fyra huvudsakliga trådar av masters och slaves, dessa syns i Figur 3 och är:

- MOSI, Master Out Slave In: överför data från master till slave.
- MISO, Master In Slave Out: överför data från slave till master.
- SCLK, Serial Clock: en konfigurerbar klocksignal som synkroniserar dataöverföringen, styrs av mastern.
- SS, Slave Select: väljer vilken slave enhet som ska vara aktiv.



**Figur 3:** Seriell koppling av SPI.

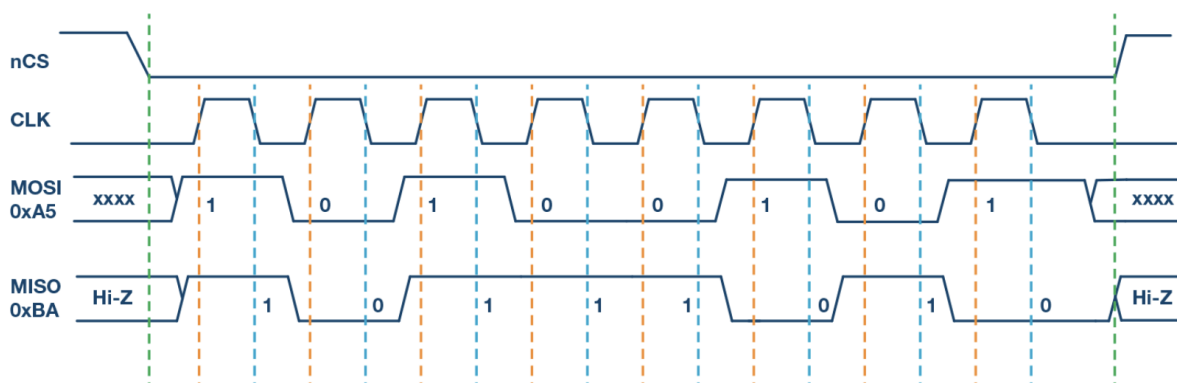
För att skicka data mellan master och slave så måste mastern först skicka en klocksignal och genom Slave select väljs vilken slave som ska vara aktiv. Slave select ska vara låg för att kunna välja slave. Eftersom SPI-är full-duplex så kan både master och slave skicka data samtidigt via MOSI och MISO [15].

SPI-datan består av en CPOL-bit och en CPHA-bit som definierar olika lägen eller "modes", vilka mastern som sagt väljer [15]. CPOL-biten bestämmer klockpolariteten under det inaktiva läget, som är definierat av att Slave select går från hög till låg i början av en dataöverföring och av att Slave select går från låg till hög i slutet av en dataöverföring. CPHA-biten bestämmer klockfasen. Beroende på värde så bestäms om det är stigande eller fallande flank som ska till exempel sampla eller skicka data. Beroende på värdena på CPOL- och CPHA-bitarna så finns det totalt fyra olika lägen tillgängliga [15], se Tabell 1.

**Tabell 1:** Tabell över modes.

Mode	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

I mode 0 är både klockpolariteten och klockfasen 0, vilket innebär att klocksignalen är låg i dess inaktiva läge samt att data samplas vid stigande flank och att data skickas vid fallande flank. I mode 1 är klockpolariteten 0 och klockfasen 1, vilket innebär att klocksignalen är låg i dess inaktiva läge samt att data samplas vid fallande flank och att data skickas vid stigande flank. I mode 2 är klockpolariteten 1 och klockfasen 0, vilket innebär att klocksignalen är hög i dess inaktiva läge samt att data samplas vid fallande flank och att data skickas vid stigande flank. I mode 3 är klockpolariteten 1 och klockfasen 1, vilket innebär att klocksignalen är hög i dess inaktiva läge samt att data samplas vid stigande flank och att data skickas vid fallande flank [15]. Figur 4 visar ett exempel på hur dataöverföring med SPI kan se ut i mode 0.

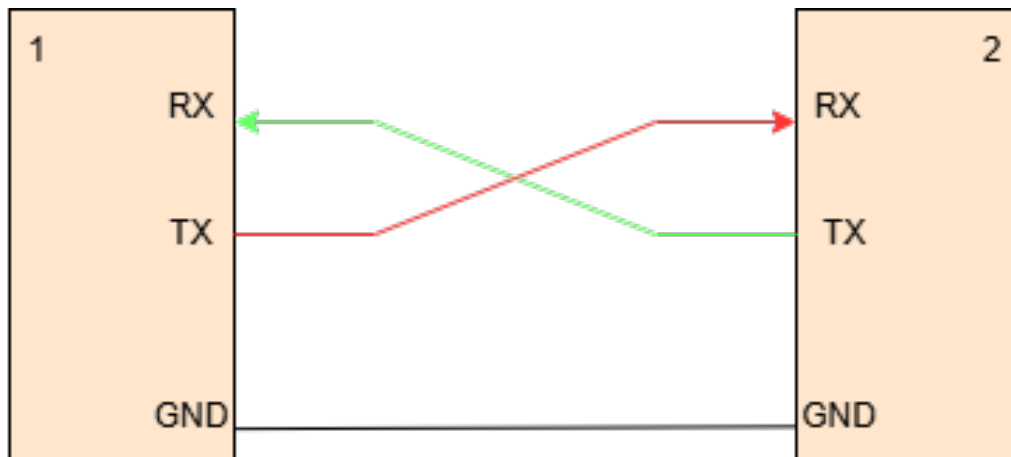


**Figur 4:** Dataöverföring SPI [15].

Jämfört med andra kommunikationsmetoder så är SPI bättre anpassat för system som kräver lågenergi. SPI kan dock vara relativt känsligt för brus och störningar [16]. SPI har endast möjlighet för ett fåtal slavar, enheter, eftersom respektive slav kräver en separat Slave Select-ledning. I praktiken används oftast 2-4 enheter innan det blir ohanterligt.

### 3.3.3 Universal Asynchronous Receiver/Transmitter (UART)

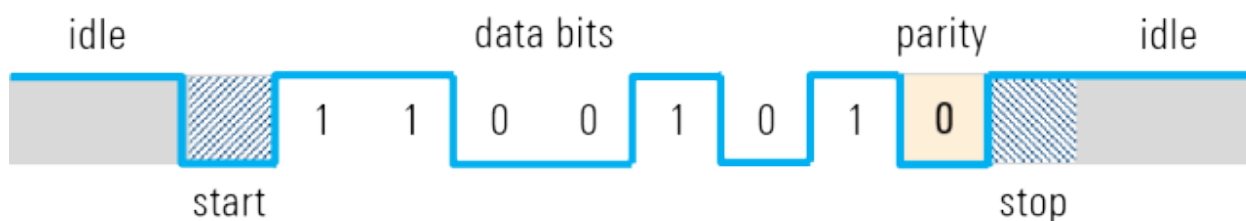
Universal Asynchronous Receiver/Transmitter, UART, är ett protokoll för att skicka data mellan olika enheter som kan agera som en slags förbindelse för asynkron kommunikation, vilket till exempel gör det möjligt för enheter med olika klockfrekvenser att kommunicera utan problem. Data sänds i form av protokoll eller ramar med ett visst antal bitar som symboliserar olika information, se Figur 5. Ramen består av en start-bit, ett specifikt antal data-bitar (vilket oftast är 8 stycken) och en slut-bit, se Figur 6. Sekvensen av bitar kan även innehålla en bit som har med felsökning att göra om detta önskas. Med UART så kommer sändare och mottagare att dela på en och samma klocka, vilket betyder att de måste enas om samma baud-rate, alltså hur många gånger per sekund signalen ändras [17].



**Figur 5:** Översikt av UART-protokoll.

Eftersom UART är asynkron så behöver inkommande data signaleras via en startbit. Startbiten är en övergång från ett inaktivt högt tillstånd till ett lågt tillstånd som sedan följs av ett visst antal användardatabitar. Slutet på datan indikeras via slutbiten som antingen är en övergång tillbaka till det höga tillståndet, eller fortsättning på det höga tillståndet. En andra slutbit kan adderas med dess funktion att ge mottagaren tid att göra sig redo för nästa ram, detta är dock ovanligt.

Databitarna är användardata och kommer som nämnt innan direkt efter startbiten. Antalet bitar är vanligtvis 7-8 men kan vara allt emellan 5-9. Databitarna är vanligtvis överförda med MSB (minst signifikanta biten) först [17].



**Figur 6:** Bitformat för UART [17].

UART har flera fördelar där den främsta är dess simplicitet. Det är enkelt att installera både dess mjuk- och programvara. UART kräver väldigt lite strömförbrukning med dess variabla hastighet och kan möjliggöra för full-duplexkommunikation med dess gränssnitt. Full-duplex innebär som nämnts att kommunikation kan ske i båda riktningar samtidigt.

Å andra sidan så besitter UART några slående nackdelar där dess gränssnitt i princip är begränsad till två enheter. I de flesta fall är UART långsammare än I2C och SPI [18].

## 4 DISKUSSION

I följande avsnitt diskuteras huruvida de olika kommunikationsmetoderna är lämpliga för användning på en lagerrobot samt vårt val av kommunikationsmetoder. Den trådlösa kommunikationen innebär kommunikation mellan PC och robot, mer specifikt Raspberry Pi:n. Den seriella kommunikationen innebär kommunikation på bussen mellan Raspberry Pi:n och styr- och sensorenheten.

### 4.1 Trådlös kommunikation

Trots att Bluetooth och Wi-Fi skiljer sig åt i funktion, så skiljer det inte mycket i montering på en robot i praktiken. Förutom räckvidden, i vilket Wi-Fi har ett övertag, var det mest frågan om säkerhet och enkelhet som bidrog som argument för det slutgiltiga valet. Som nämnts i avsnitt 4.2.1 finns det risk för att utomstående användare kopplar upp sig mot det Wi-Fi nätverket som används inom projektet. Detta kan inte bara ge störningar utan risken finns även att dessa användare lyckas ändra eller ta bort data som är av vikt för funktionaliteten hos roboten.

Gällande frågan om räckvidd så har Bluetooth endast en räckvidd på cirka 10 meter. I detta projekt kommer en mycket större räckvidd ej vara nödvändig. De olika enheterna som kommunicerar via Bluetooth, i detta fall en PC och roboten, kommer nödvändigtvis inte behöva vara längre ifrån varandra än storleken på lagret. Lagret kommer bestå av ett rutnät med känd storlek efter vilket vi kan anpassa avståndet mellan enheterna.

Utöver ovanstående argument har Bluetooth ytterligare en fördel som till stor del avgör det slutgiltiga valet. Detta rör enkelheten i hur uppkoppling via Bluetooth sker till skillnad från Wi-Fi. För Wi-Fi kommer det alltid krävas tre enheter: PC, Raspberry Pi (roboten) och en router. Bluetooth kräver endast de två förstnämnda enheterna. Detta ger oss som grupp en sak mindre att tänka på, eftersom PC:n och roboten är två enheter vi alltid bär med oss, till skillnad från en router.

### 4.2 Seriell kommunikation

De olika egenskaperna vi kollade på som låg till grund för valet av kommunikation är komplexitet, energiåtgång, hastighet samt simplicitet vid uppkoppling på virkort.

Med komplexitet hos de olika kommunikationssätten syftar vi mest på hur dataöverföringen fungerar. En fördel med I2C är dess egenskap att möjliggöra för flera masters, men det innefattar också en större komplexitet vid programmering av bussen. SPI däremot har inte denna egenskap av flertalet masters varav det krävs en mindre omfattning av kod. Färre masters ger ett mindre komplext system vilket skulle kunna underlätta implementeringen av buss. Färre masters ger också naturligt färre ledningar att hålla koll på. UART är också enkelt och kräver väldigt lite kod, men nackdelen med denna är dess begränsning i enheter. UART är i princip begränsat till endast två enheter, vilket kan bli problematiskt för en robot med flertalet servon och sensormoduler.

En skillnad mellan I2C och SPI är deras hastighet. I2C har begränsningar gällande maxhastighet vilket SPI inte har. SPI är alltså ett mycket snabbt kommunikationssätt vilket skulle kunna vara en fördel. Även UART är som nämnts relativt långsamt.

Om man tittar på energiförbrukning hos de olika kommunikationssätten så är SPI bra anpassat för lågenergisystem,

I2C däremot drar mer energi och skulle kunna vara bättre anpassat för större system. På roboten kommer vi endast ha tillgång till batterier och av denna anledning är det stor fördel med en kommunikationsenhet som är anpassat för låge-nergisystem. Utöver SPI är då även UART en bra kandidat för projektet då dess enkelhet bidrar till låg energiåtgång.

Något vi även fick mer kunskap om vid skrivandet av denna förstudie är hur data skickas vid de olika kommunikations-sätten. Vid I2C-kommunikation så skickas data med en avslutande ACK-bit efter varje byte, detta kan användas för att säkerställa att data verkligen tagits emot av respektive slav. Denna uppbyggnad av data finns inte i SPI-protokollet, och det går alltså inte att verifiera om data tagits emot korrekt eller ej. Detta kan vara användbart för oss under arbets-processen, förmodligen kommer det att underlätta i felsökningsprocesser eller liknande i fall där vi vill undersöka om data skickats korrekt eller ej.

En sista egenskap som vi ser över hos de olika systemen är hur pass komplex uppkopplingen av sladdar blir på virkortet som sitter på roboten. Då vi kommer ha flertalet enheter bidrar detta till många sladdar och därför är det en fördel om antalet kan minska för att möjliggöra kommunikationen via bussen. För UART krävs det endast ett fåtal på grund av dess simplicitet och begränsning till enheter. Som följd av komplexiteten av I2C som beskrivits ovan kan detta bidra till en ökad mängd kopplingar och sladdar/kablar. Av denna förklaring kan då SPI, precis som UART, sänka antalet kopplingar och sladdar och bidra till att projektet kan bli något smidigare att genomföra.

## 5 SLUTSATSER

I följande avsnitt kommer det att dras en slutsats gällande vilka kommunikationsmetoder som passar bäst för roboten och därmed kommer att användas inom projektet.

Baserat på ovan nämnda egenskaper kommer Bluetooth att användas som trådlös kommunikation mellan PC och Raspberry Pi:n samt SPI för kommunikation mellan Raspberry Pi:n och de olika styr- och sensorenheterna. De mest avgörande egenskaperna som är av vikt för detta projekt är låg energiförbrukning och simplicitet. Trots att även UART har en låg energiförbrukning är den inte bäst lämpad för detta projekt då UART endast är anpassat för två enheter. I2C valdes främst bort på grund av dess komplexitet och höga förbrukning av energi.

## REFERENSER

- [1] R. P. Ltd, *Getting started with your Raspberry Pi*, Hämtad: 2025-02-11, [Online]. Tillgänglig: <https://www.raspberrypi.com/documentation/computers/getting-started.html>, 2012-2025.
- [2] R. P. Foundation, *GPIO pins*, Hämtad: 2025-03-31, [Online]. Tillgänglig: <https://projects.raspberrypi.org/en/projects/physical-computing/1>, no date.
- [3] ProXPN, *Vad är WiFi och vad står det för?* Hämtad: 2025-02-11, [Online]. Tillgänglig: <https://proxpn.com/sv/wifi/vad-ar-wifi/>, last updated 2024.
- [4] C. B. Portnox, *The Origin of the Word "Wi-Fi": A Dive into Tech Etymology*, Accessed: 2025-02-23, 2024-06-25. URL: <https://www.portnox.com/blog/security-trends/origin-of-wifi/>.
- [5] Strålsäkerhetsmyndigheten, *Trådlösa datornätverk och wifi*, Accessed: 2025-02-23, 2024-12-09. URL: <https://www.stralsakerhetsmyndigheten.se/omraden/magnetfalt-och-tradlos-teknik/tradlos-teknik/tradlosa-datornatverk-och-wifi/>.
- [6] I. Internetkunskap, *Wifi*, Accessed: 2025-02-23, n.d. URL: <https://internetkunskap.se/artiklar/ordlista/wifi/>.
- [7] E. Britannica, *Bluetooth*, Hämtad: 2025-02-18, [Online]. Tillgänglig: <https://www.britannica.com/technology/wireless-communications>, last updated 2025.
- [8] S. C. Jan Storhaug Bill Christman, *Bluetooth Lingo: What is the Difference between Bluetooth and 2.4 GHz?* Hämtad: 2025-04-23, [Online]. Tillgänglig: <https://www.audiologyonline.com/ask-the-experts/bluetooth-lingo-what-difference-between-23092>, 2018-06-11.
- [9] ISY, *Kommunikation Raspberry Pi - ATmega*, Hämtad: 2025-02-23, [Online]. Tillgänglig: <https://da-proj.gitlab-pages.liu.se/vanheden/page/kommunikation/>, 2003.
- [10] ISY, *I2C*, Hämtad: 2025-02-24, [Online]. Tillgänglig: <https://da-proj.gitlab-pages.liu.se/vanheden/page/kommunikation/i2c/>, 2003.
- [11] J. B. Jonathan Valdez, *Understanding the I2C Bus*, Hämtad: 2025-04-14, [Online]. Tillgänglig: <https://www.ti.com/lit/an/slva704/slva704.pdf?ts=1744594000731>, 2015.
- [12] J. H. Nicholas Zambetti Karl Söderby, *Inter-Integrated Circuit (I2C) Protocol*, Hämtad: 2025-02-24, [Online]. Tillgänglig: <https://docs.arduino.cc/learn/communication/wire/>, 2024.
- [13] R. Keim, *Introduction to the I2C Bus*, Hämtad: 2025-02-24, [Online]. Tillgänglig: <https://www.allaboutcircuits.com/technical-articles/introduction-to-the-i2c-bus/>, 2015.
- [14] M. Sufyan, *SPI Protocol: Revolutionizing Data Communication in Embedded Systems*, Hämtad: 2025-02-24, [Online]. Tillgänglig: <https://www.wevolver.com/article/spi-protocol>, 2024.
- [15] P. Dhaker, *Introduction to SPI Interface*, Hämtad: 2025-04-14, [Online]. Tillgänglig: <https://www.analog.com/en/resources/analog-dialogue/articles/introduction-to-spi-interface.html>, No date.
- [16] S. Writer, *The Differences Between I2C and SPI (I2C vs SPI)*, Hämtad: 2025-04-14, [Online]. Tillgänglig: [https://www.totalphase.com/blog/2021/07/i2c-vs-spi-protocol-analyzers-differences-and-similarities/?srsltid=AfmBOoqlppRKT8i2vnd8i9PB8iTlLYWNZ2ocovJFf\\_FDrDGVujt0N5O9](https://www.totalphase.com/blog/2021/07/i2c-vs-spi-protocol-analyzers-differences-and-similarities/?srsltid=AfmBOoqlppRKT8i2vnd8i9PB8iTlLYWNZ2ocovJFf_FDrDGVujt0N5O9), 2021.

- [17] V. Kanade, *What Is Universal Asynchronous Receiver-Transmitter (UART)? Meaning, Working, Models, and Uses*, Hämtad: 2025-02-24, [Online]. Tillgänglig: <https://www.spiceworks.com/tech/networking/articles/what-is-uart/>, 2024.
- [18] N. Busler, *UART - Universal Asynchronous Receiver-Transmitter (universell asynkron mottagare-sändare)*, Hämtad: 2025-04-14, [Online]. Tillgänglig: <https://picockpit.com/raspberry-pi/sv/uart-the-universal-asynchronous-receiver-transmitter/#Advantages>, 2022.