



CHALMERS
UNIVERSITY OF TECHNOLOGY

Att lösa en labyrint - design och konstruktion av autonom robot

Kandidatarbete inom System- och Reglerteknik

Andreas Andersson, Emil Grimheden, Ida Hagström,
Emma Rådahl, Rebecca Svensson, Isak Åslund

Kandidatarbete 2019: EENX15-19-19



CHALMERS

Andreas Andersson
Emil Grimheden
Ida Hagström
Emma Rådahl
Rebecca Svensson
Isak Åslund

Handledare: Adrian Ilka, Post doc, Elektroteknik
Examinator: Nikolce Murgovski, Docent, Elektroteknik

Institutionen för Elektroteknik
Avdelningen för System- och Reglerteknik (SysCon)
Chalmers tekniska högskola
SE-412 96 Göteborg
Telefon: +46 31 772 1000

Förord

Under våren 2019 har rapporten skrivits vid avdelningen för System och Reglerteknik på institutionen för Elektroteknik vid Chalmers tekniska högskola. Kandidatgruppen består av sex studenter med ingenjörsbakgrund inom Elektroteknik, Automation och Mekatronik samt Teknisk fysik på grundnivå.

Som del av projektet har en labyrint konstruerats för att kunna testa roboten i en tävlingsriktig miljö. Labyrinten går att förändra och har därmed gett en bra testmiljö för liknande framtida projekt. Vi har uppskattat projektet och anser att området är värt att fortsätta utforska.

Gruppen vill tacka prototyplaboratoriet för hjälpen vid bygge av labyrint, speciellt tack till Jan Braee, Reine Nohlborg och Göran Stiegler. Tack till Nikolce Murgovski för att han tog sig tid att diskutera utvecklandet av projektet. Vi vill också tacka Chalmers robotförening för välvilja samt deras frikostighet. Xueqiao Xu's mjukvara för simulering av sökalgoritmer har varit uppskattat stöd för förståelse för algoritmer. Peter Harrison tackas för sitt bibliotek av digitala Micromouse labyrinter. Slutligen vill vi tacka Adrian Ilka för god handledning, alla diskussioner och råd som var mycket uppskattade.

Sammanfattning

I detta projekt designas och konstrueras en förhållandevis enkel robot vars uppgift är att kartlägga och hitta vägen genom en labyrinth med förutbestämd storlek. Liknande teknik utgör grunden för många autonoma robotar som utför hushållsuppgifter, men även robotar som används i miljöer som är svåra eller farliga för människor att vistas i. Till den konstruerade roboten utvecklas programvara innehållande en sökalgoritm som är baserad på algoritmen A*, men är anpassad för att hantera en fysisk robot.

Den autonoma roboten kan lösa labyrinthen, med begränsning av repeterbarheten, och baserat på den utförda kartläggningen kan den kortaste vägen genereras. Roboten kan sedan köra denna väg i en högre hastighet än vad den hade under kartläggningen av labyrinthen.

Abstract

A relatively simple robot is designed and built, with the purpose to map and find a path in a maze with predetermined dimensions. Similar technology is the foundation of autonomous robots used in household chores, but also of robots used in surroundings too dangerous or unreachable for humans. To run the robot, a software containing a search algorithm is created. The search algorithm is based on A*, but it is modified to better suit this problem.

The autonomous robot is able to solve the maze, with some limitations of repeatability, and based on the preformed mapping a shortest path is generated. It is thereafter possible for the robot to run this path with a higher velocity than during the mapping.

Ordlista

Huvuddator – Dator med ett operativsystem för högnivåspråkig programmering.

Mjukvara– Alla program som används till roboten, det vill säga både styrning och programvara.

Mikrokontroller – Styrenhet för hårdvarunära programmering.

Styrning – Hårdvarunära programmering på mikrokontrollern.

Programvara – Högnivåspråksprogrammet skrivet på huvuddatorn. Innehåller bland annat sökalgoritmen och kringliggande funktionalitet.

Tävlingen – Micromouse-tävlingen är en etablerad tävling där autonoma robotar tävlar om att lösa en labyrint på kortast tid.

Kartläggningsfas – Första delen i tävlingen där roboten undersöker celler och kartlägger labyrinten.

Spurfas – Andra fasen i tävlingen där roboten kör den snabbaste kartlagda vägen.

UART – Universal Asynchronous Receiver/Transmitter, ett universellt protokoll för att kommunicera mellan två datorer.

Datapakets – En binärt sekvens som skickas via mellan programmeringsenheterna, varje sekvens motsvarar en karaktär.

API – Applikationsprogrammeringsgränssnitt, ett protokoll för att underlätta kommunikation mellan olika program.

Sökalgoritm – Algoritmen som används av programvaran för att hitta vägar i labyrinten.

LiDAR – Ett optiskt mätinstrument, även kallat ljusradar.

Innehåll

1	Inledning	1
1.1	Syftesformulering	1
1.2	Beskrivning av regler för Micromouse-tävlingen	2
1.3	Problemformulering	2
1.3.1	Konstruktion av autonom robot	2
1.3.2	Programvara för kartläggning och lösning av labyrinten	3
1.4	Avgränsningar	4
2	Teori för konstruktion av robot	5
2.1	Funktionalitet hos likströmsmotorer	5
2.2	Principiell användning av sensorer	6
2.3	Grundläggande teori kring reglering	7
3	Konstruktion av robot	9
3.1	Val av motorer	9
3.2	Design av växellåda och hjul	10
3.3	Jämförelse och analys av sensorer	11
3.4	Batteri	12
3.5	Val av mikrokontroller och huvuddator	13
3.6	Design av kretskort	14
3.7	Initial prototyp av robot	14
3.8	Implementation av initialt reglersystem	15
3.9	Implementation av slutgiltigt reglersystem	16
3.10	Slutgiltig konstruktion	19
4	Datastrukturer och algoritmer	20
4.1	Grafteori	20
4.2	Algoritmer för genomsökning av datastrukturer	21
4.2.1	A*	21
4.3	Heuristik för uppskattning av avstånd	22
5	Programvara för lösning av labyrint	24
5.1	Beskrivning av slutgiltig programvara	24
5.1.1	Programmets modell av verkligheten	24
5.1.2	Generering av instruktion	25
5.1.3	Körning efter en följd av instruktioner	27
6	Kommunikation mellan huvuddator och mikrokontroller	28
7	Resultat från simuleringar av programvara	30
7.1	Jämförelse av A* och Greedy Best-First Search	30
7.2	Programvarans prestanda beroende av viktning av $h(n)$	30
7.3	Sökalgoritmens körtid	32

8	Metod för test av robot	33
8.1	Konstruktion av testmiljö	34
9	Resultat från tester av robot	35
9.1	Positionsreglering	35
9.2	Precision vid rotation	35
9.3	Sensorernas noggrannhet	36
9.4	Hastighet och acceleration	37
9.5	Batteritid	39
9.6	Redogörelse av yttre egenskaper	40
9.7	Systemets förmåga att lösa labyrinten	40
10	Diskussion	41
10.1	Resultat av hårdvarutester	41
10.2	Systemets förmåga att lösa labyrinten	42
10.3	Möjlig vidareutveckling	43
11	Slutsats	47
	Referenser	48
A	Regelverk för Micromouse tävling	II
B	Datablad: Motor	V
C	Datablad: Växellåda	VI
D	Datablad: GP2Y0A21YK0F	VII
E	Datablad: GP2Y0A51SK0	XVI
F	Datablad VL53L0X	XXIV
G	Elschema för kretskort	XXVII
H	Kretskortsritning	XXVIII
I	Labyrintkoncept	XXIX
I.1	Bottenplatta	XXX
I.2	Stolpe	XXXI
I.3	Väggar	XXXII

1 Inledning

Under de senaste åren har utvecklingen och användningen av autonoma robotar ökat markant och det läggs stort fokus på forskning inom detta område. Autonoma robotar ska kunna fungera väl utan mänsklig inverkan och de används numera i både industrin och hushåll [1]. De har blivit en tillgång som utför arbete som för människor både är slitsamt, tidskrävande och i vissa fall omöjliga [2].

Då dessa robotar ofta agerar i en miljö som inte är isolerad krävs en medvetenhet av omgivningen för att undvika skada på människor eller andra objekt. Enligt två rapporter utgivna av IEEE angående utvecklingen av mobila autonoma robotar är den mest fundamentala egenskapen i autonoma system att läsa av miljön och agera utifrån detta [1], [3]. Det är därför viktigt att det finns kraftfulla sensorer som på ett pålitligt sätt känner av omgivningen samt mjukvara som utför lämpliga handlingar baserat på datan från sensorerna.

Med en mer sofistikerad mjukvara kommer denna teknologi ha ett brett användningsområde. Roboten kan implementeras i miljöer där oförutsägbara händelser kan ske och ersätta mänsklig arbetskraft för förbättrat resultat och minskad risk för människor. En utveckling av denna princip kan nyttjas på många sätt, bland annat genom att använda en robot för att kartlägga områden, exempelvis för att hitta den mest optimala vägen till ett mål [2], [4].

Principen med att låta en robot hitta en väg mellan två platser är inte ny. Redan 1950 konstruerade ingenjören Claude Shannon en robot som kartlade och löste en labyrinth¹ [5]. Detta anses vara en av de tidigaste lyckade robotar som använder sig av maskininlärning och artificiell intelligens. Dock har både programvaran och hårdvaran förbättrats avsevärt vilket medför att tiden det tar för roboten att utföra uppgiften har minskat markant [6] men principen är densamma som används i detta projekt.

1.1 Syftesformulering

Syftet med detta projektet är att utveckla ett system där en autonom robot ska kunna hitta målet i en labyrinth på kortast möjliga tid. Det eftersträvas även att följa reglerna för Micromouse-tävlingen som finns beskrivna i avsnitt 1.2. I utformandet av systemet innefattas utvecklande av mjukvara som kartlägger och söker efter en väg i labyrinthen samt konstruktion av hårdvara som är anpassad efter den avsedda miljön.

¹Detta kan ses på: <https://www.youtube.com/watch?v=vPKkXibQXGA> [5]

1.2 Beskrivning av regler för Micromouse-tävlingen

Micromouse-tävlingen har haft en central roll i projektet och kommer genomgående i rapporten kallas för tävlingen. Det är en över 30 år gammal internationell tävling som finns i Nordamerika, Europa och Asien men ännu inte i Sverige [7], [8]. Tävlingen går ut på att en liten robot ska ta sig igenom en okänd labyrint på kortast tid. Tävlingsreglerna som följs i projektet är skrivna av Birmingham City University [7]. Stycket nedan beskriver de mest väsentliga reglerna för detta projekt, reglerna i sin helhet kan ses i bilaga A.

Labyrinten som roboten ska kartlägga och ta sig igenom är en kvadrat på 9 m^2 och består av rutor som är $0,18 \times 0,18\text{ m}^2$ stora, se bilaga A.1.1. Detta är det enda som är känt för roboten förutom position av startpunkt och målpunkten. Labyrinten ska lösas genom att roboten kör igenom labyrinten, det är därför ej tillåtet att exempelvis hoppa över väggar för att snabbare ta sig till mål, se bilaga A.2.5. Roboten har en del restriktioner och de mest centrala för projektet är storleken, den får inte vara större än $0,25\text{ m} \times 0,25\text{ m}$ samt att roboten inte får ha någon hjälp utifrån. Med detta menas att allt som roboten behöver för att lösa labyrinten behöver få plats på roboten och ingen tillförsel av information får ske efter start, se bilaga A.2.1.

Robotens körtid mäts i två steg, första är "Search time" som går ut på att roboten kartlägger labyrinten. I rapporten kommer detta tävlingsmoment benämnas som kartläggningsfas. Andra fasen kallas "Run time", då mäts tiden mellan start och mål, se bilaga A.3.1. Denna fas kommer i rapporten kallas för spurtfas. En summering av de två tider med ett handikapp på söktiden görs. Med handikapp menas att tiden för första sökningen divideras med en faktor för att påverkan av denna rundan ej ska vara för avgörande. Till den summerade tiden kan även en strafftid adderas som bot för otillåtna handlingar, som att roboten kör in i väggar eller om roboten behöver mänsklig hjälp för att färdigställa fasen, se bilaga A.3.3.

Tävlingen har en visuellt tilltalande uppbyggnad och det medför att det finns en stor potential att på ett tydligt sätt demonstrera robotteknik för människor utan teknisk erfarenhet i exempelvis rekryteringssamanghang.

1.3 Problemformulering

Projektet delades tidigt upp i två huvuddelar: konstruktion och styrning av robot och utveckling av programvara för att lösa labyrinten. I detta avsnitt presenteras problemformuleringen för dessa avsnitten var för sig.

1.3.1 Konstruktion av autonom robot

För konstruktionen av roboten bestämdes restriktioner som baserats på tävlingens regler och krav och en förstudie om vilka hastigheter och accelerationer som uppnås i

internationella tävlingar. Roboten Green Giant [9] av Green Ye som har vunnit stora tävlingar är exempel på robotar som studerades. De väsentliga krav och önskemål som bestämdes kan ses i tabell 1.

Tabell 1: Fastställda krav och önskemål på hårdvara.

Parameter	Krav	Önskemål
Hastighet	1 m/s	3 m/s
Acceleration	5 m/s ²	10 m/s ²
Vikt	500 g	200 g
Batteritid	>10 min	>20min
Antal hjul	–	4 st

För att kartlägga labyrinten och undvika kollision med väggar krävs avståndssensorer. Avståndssensorerna kräver ett tillräckligt stort mätområde för att kunna mäta avståndet till en vägg då roboten befinner sig precis intill motsatt vägg. Kravet på mätområdet bestäms av en cells bredd och robotens bredd.

Styrning av motorer och perferienheter valdes att implementeras med mikrokontroller och hårdvarunära programmering i programspråket C [10]. Programvaran för kartläggning och lösa labyrinten som beskrivs i avsnitt 1.3.2 valdes att implementeras i högnivåspråket Python [11] vilket krävde en dator [12] som kör ett operativsystem.

1.3.2 Programvara för kartläggning och lösning av labyrinten

Att utveckla mjukvara som möjliggör att en robot kan lösa en okänd labyrint är ett brett problem och för att underlätta utvecklingen måste det tydliggöras vad mjukvaran ska åstadkomma. På grund av de tävlingsregler som projektet förhåller sig till finns det konkreta förutsättningar för programvaran, att kartlägga labyrinten och med hjälp av detta hitta en väg från start till mål. Utöver detta finns ett önskemål att kartläggningen sker snabbt och att den funna vägen är den väg som tar roboten från start till mål på kortast möjliga tid.

Genom att studera tävlingens regler ytterligare kan mjukvarans utformning anpassas efter de förutsättningar som finns och därefter få en mer optimerad lösning. Först och främst är banans storlek given vilket ger en tydlig begränsning av sökområdet och begränsar tiden det tar att söka efter en väg. Utöver detta är positioner för startpunkt och mål givna, vilket ger en möjlighet att göra en så kallad informerad sökning. I en informerad sökning tas även riktningen till målet i beaktning, då ett beslut ska tas vid ett vägskäl och målet hittas därmed snabbare. Andra förutsättningar som inte påverkar val av algoritm lika mycket som tidigare nämnda premisser, men som ändå måste tas i beaktning är att området mellan startpunkten och målet är helt okänt och att banan kartläggs efter hand. Dessutom finns en fysisk robot som ska förflytta sig och dess rörelse begränsar kartläggningen av banan då flera områden inte kan utforskas samtidigt, i motsats till sökalgoritmer generellt. Labyrinten som

roboten ska lösa kan ses som ett rutnät med celler, som skiljs åt med väggar och i detta kan roboten enbart förflytta sig i fyra riktningar. Det gör implementationen enklare än om labyrinten bestått av oregelbundna celler eller om roboten hade fler möjliga rörelseriktningar.

1.4 Avgränsningar

När programvaran för kartläggning och genomsökning av labyrinten konstruerades gjordes en tydlig avgränsning i implementationen av algoritmen som ska hitta en väg från start till mål. Det beslutades att inte utveckla en algoritm från grunden utan att anpassa en känd algoritm till de behov som fanns. Kartläggningen av labyrinten begränsades till att ske då roboten rör sig från start till mål. Då målet är hittat tar sig roboten tillbaka till start utan att genomsöka labyrinten ytterligare. Det antas dessutom vara möjligt att finna en väg från start till mål.

Externa begränsningar för projektet var en budget och angivelser av tid. Budgeten var satt till 5000 SEK och därav begränsades valmöjligheterna på komponenter för att den totala kostnaden ej skulle överstiga denna budget.

2 Teori för konstruktion av robot

Hårdvaran består av robotens olika fysiska komponenter vilket innebär allt från hjul till mikrokontroller. I detta avsnitt behandlas teorin för en del av de olika komponenterna såväl som de formler som används inför val av komponenter.

2.1 Funktionalitet hos likströmsmotorer

För att räkna på hastigheter, acceleration, vridmoment med mera krävs de grundläggande förhållanden som beskriver en permanentmagnetiserad likströmsmotor, dessa kan ses nedan.

$$P = \tau \cdot \omega \quad (1)$$

Effekten P ut från en roterande axel beror på vridmomentet τ och vinkelhastigheten ω . Förhållandet styr i sin tur translationshastighet v och acceleration a genom ekvation (2) och (3), där r är radien på hjulet och m massan på hela enheten.

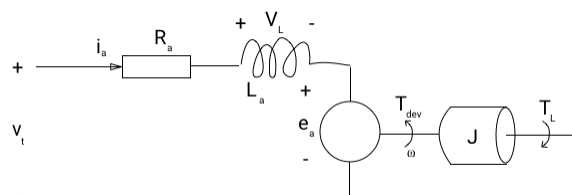
$$v = \omega \cdot r \quad (2)$$

$$\begin{aligned} \tau &= F \cdot r \\ F &= m \cdot a \iff a = \frac{\tau}{r \cdot m} \end{aligned} \quad (3)$$

Omvandling från vinkelhastighet till rotationshastighet n ges av ekvation (4).

$$n = \frac{60 \cdot \omega}{2 \cdot \pi} [rpm] \quad (4)$$

En elektrisk modell för en likströmsmotorn kan ses i figur 1. Ekvation (5) beskriver hur kretsen samverkar med hjälp av Kirchhoffs spänningslag.



Figur 1: Elektrisk krets som visar en permanent magnetiserad likströmsmotor. Eget material.

$$\begin{aligned}
e_a &= \lambda \cdot \omega \\
V_L &= L \cdot \frac{di}{dt} \\
T_{dev} &= \lambda \cdot i_a \\
T_L &= b \cdot \omega \\
V_t &= R_a \cdot i_a + v_l + e_a
\end{aligned} \tag{5}$$

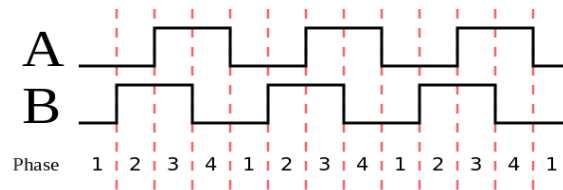
För simulering av systemet kan Laplacetransformen av differentialekvation (5) och (6) användas som kan ses i ekvation (7), där J är tröghetsmoment, T_{dev} är det utvecklade vridmomentet och T_L är det lastande vridmomentet som beror på friktionskonstanten b .

$$J \frac{d\omega}{dt} = T_{dev} - T_L \tag{6}$$

$$G(s) = \frac{\dot{\Theta}(s)}{V(s)} = \frac{\lambda}{(Js + b)(L_a s + R_a) + \lambda^2} \left[\frac{rad/s}{V} \right] \tag{7}$$

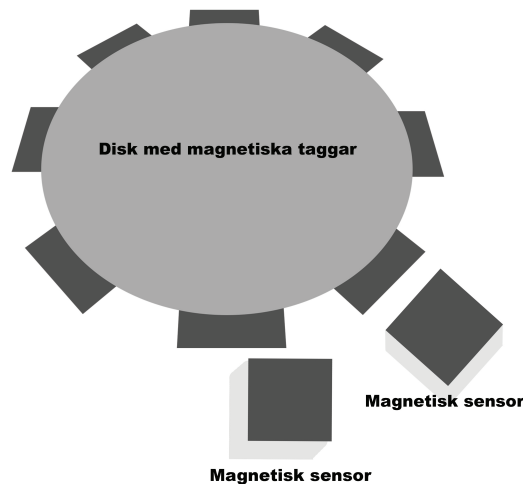
2.2 Principiell användning av sensorer

För att mäta vinkel, vinkelhastighet och vinkelacceleration kring en axel kan en pulsgivare användas. En magnetisk inkrementell pulsgivare består av en roterande disk med magneter och två givare vilket illustreras i figur 3. Givaren ger ifrån sig två fyrkantsvågor där frekvensen indikerar rotationshastigheten, se figur 2.



Figur 2: Pulsgivarens utsignaler från givare A och B, där givare B placerats så att fasförskjutningen relativt givare A är 90°. Genom att veta övergångar på en givarna och värdet på den andra givaren kan rotationsriktning bestämmas. Återgiven från [13].

Disken är fäst på en axel och vid rotation registrerar avläsaren när magneterna passerar, dessa registreringar kallas pulser. Genom att räkna pulser, tiden mellan pulser och veta antal pulser per varv kan vinkel, vinkelhastighet, vinkelacceleration och i sin tur position, translationshastighet och translationsacceleration bestämmas då hjulets diameter är känt.



Figur 3: Illustration över en magnetisk pulsgivare. Eget material.

För att kunna beräkna avståndet till en vägg eller objekt kan en så kallad time of flight sensor användas. Tekniken används i bland annat djupseende kameror och LiDAR. Sensorn får data genom att ett objekt bestrålas med ljus från en ljuskälla, vilken har en mottagare monterad bredvid. Baserat på tiden det tar för ljuset att studsas mot objektet och sedan samlas in av mottagaren kan avståndet beräknas [14].

2.3 Grundläggande teori kring reglering

En regulators syfte är att föra börvärdet, det vill säga det värde som systemet önskas anta, $r(t)$, nära ärvärdet, med andra ord det värde som systemet har för tillfället, $y(t)$. Differensen mellan börvärdet och ärvärdet benämns som reglerfelet, där $e(t) = r(t) - y(t)$, och regulatorn är en funktion som försöker minimera reglerfelet. Ett stabilt system reducerar oscillationer med tiden medan ett instabilt system ökar oscillationer med tiden och det senare är ej önskvärt. Regulatorns utsignal, $u(t)$, skapas i många fall av den mest använda regulatorn i industrin som är proportionerlig, integral och derivata (PID) regulatorn, tack vare dess enkelhet och karakteristiska egenskaper [15]. Tre vanligt förekommande varianter av regulatorn är P-regulator, PD-regulator samt PID-regulator. Den mest enkla av dessa är P-regulatorn som beräknar utsignalen genom en proportionerlig skalning, K_p , av reglerfelet. P-reglering kan i de flesta fall föra börvärdet nära ärvärdet men ej reducera reglerfelet helt. Funktionen som ger en P-regulator ses i ekvation (8).

$$u(t) = K_p e(t) \quad (8)$$

PD-regulatorn är en mer avancerad regulator som beräknar utsignalen som en P-

regulator men adderar en deriverad term, K_d för att öka systemets hastighet och stabilisera systemet. PD-reglering kan skapa problem om ärvärdet är brusigt och skapar snabba förändringar, i detta fall blir systemet mer instabilt. Dess funktion ses i ekvation (9).

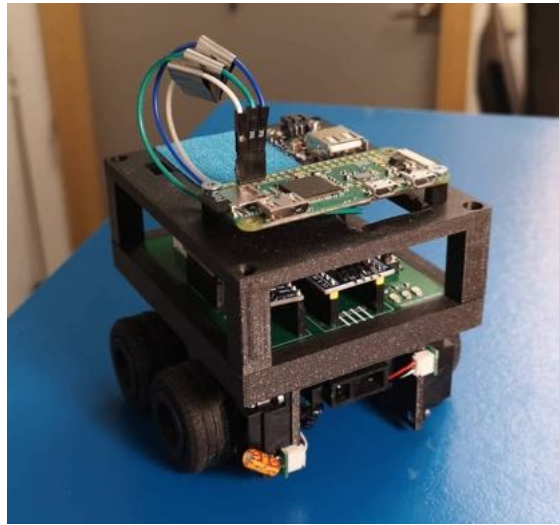
$$u(t) = K_p e(t) + K_d \frac{d}{dt} e(t) \quad (9)$$

PID-regulatorn är en kombination av proportionerlig, integrerande och deriverande reglering som tillsammans kan uppnå ett system där reglerfelet reduceras helt. Problem med PID-regulatorer är att den integrerande termen gör systemet långsammare och kan göra systemet instabilt om K_i ej dimensioneras bra. Funktionen för en PID-regulator kan ses i ekvation (10).

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{d}{dt} e(t) \quad (10)$$

3 Konstruktion av robot

För att underlätta utvecklandet av roboten delades hårdvaran upp i delsystem, där varje delsystem dimensioneras eller sätter krav på andra komponenter. En helhetsbild av den slutgiltiga roboten kan ses i figur 4. Roboten har fyra hjul som drivs av två motorer samt två nivåer, den undre innehåller komponenter för styrning av motorer och den övre besitter komponenter för att köra programvaran och batterier. Följande avsnitt kommer att redogöra för valet av olika delsystem och hur dessa sammansattes för att bilda den slutgiltiga roboten.



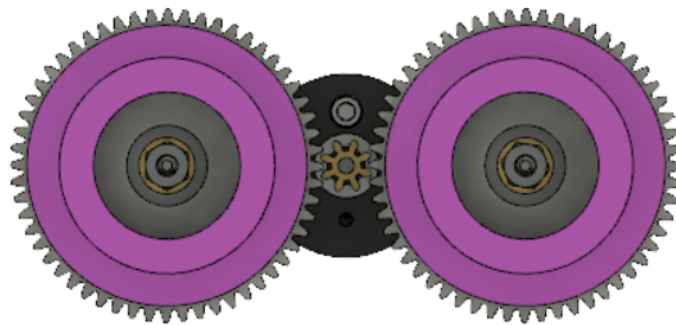
Figur 4: Roboten består av två nivåer där sensorerna sitter på den främre delen av roboten. Eget material.

3.1 Val av motorer

För att kunna garantera att kraven uppnås av motorerna gjordes simuleringar i Matlab [16] med hjälp av Simulink [17] och Simscape [18]. Databladet från tillverkarna användes för att skapa modellen med de karakteristiska parametrarna för motorn, som till exempel resistans, induktans och motorkonstanter. Simuleringen som gjordes tar hänsyn till motordynamiken, maximal matningsspänning, växellåda, robotens vikt och omvandling från rotationshastighet till translation. Modellen är ideal och tar till exempel ej hänsyn till förluster i växellådan samt antar att inget glid mellan underlag och däck sker. De simuleringar som kunde genomföras visade att det fanns få motorer som var inom den angivna budgeten och mötte projektets krav. Utifrån simuleringarna bestämdes motorerna som är av modellen 1717006SR [19] från företaget Faulhaber vars datablad finns i bilaga B. För att driva motorn valdes motordrivaren TB6612FNG [20] från Sparkfun [21].

3.2 Design av växellåda och hjul

Med hjälp av motorsimuleringarna i Simulink kunde en minsta utväxling bestämmas till 4:1 för att nå accelerationskravet. För att erhålla den önskade utväxlingen fanns alternativet att köpa en planetväxel från tillverkaren av motorn, Faulhaber [22] vars datablad finns i bilaga C. Genom att använda denna växellåda skulle motor och växellåda bli mer än dubbelt så lång som enbart motorn, vilket hade inneburit att motorerna inte hade kunnat sättas i linje som tänkt. På grund av detta beslutades det att designa och tillverka en egen växellåda. Växellådan som designades utgår från att sätta kugghjulen i rad istället för i linje längs motoraxeln. Denna design tillåter att fyra hjul kan användas vilket ger två fördelar jämfört med planetväxeln, dubbla kontaktytan mot underlaget och en smalare spårvidd². Vid design av kugghjulen fanns två avgörande begränsningar. Den största begränsningen var att kugghjulets maximala ytterdiameter var förutbestämd, då däcken valdes innan växellådan designades. Utöver detta kunde inte kugghjulen bestå av för små detaljer, då de var tänkta att tillverkas i en FDM (Fused Deposition Modeling) 3D-skrivare, som har en förhållandevis låg precision. Med hänsyn till detta valdes kuggmodulen³ till 0,5. Kugghjulen valdes till en utväxling på 56:10 vilket maximerar de större kugghjulets ytterdiameter för att öka avståndet mellan däcken. Detta kan ses tydligt i figur 5.

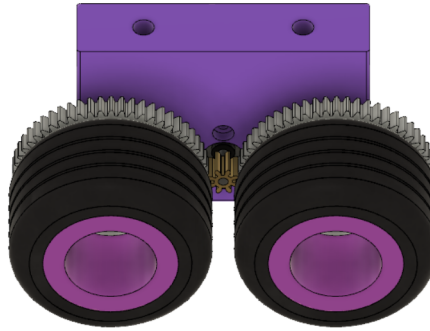


Figur 5: Det mindre kugghjulet sitter på motoraxeln som i sin tur driver de större kugghjulen. Fälgarna är lila i bilden, motorn är svart. Eget material.

Eftersom kugghjulen tillverkades med en FDM 3D-skrivare blev de fysiska kugghjulen inte exakt som de modellerade. Detta orsakade icke önskvärd friktion mellan kuggarna och för att lösa problemet köptes gjutna kugghjul i acetalplast. Kugghjulen fästes på fälgarna som designades utifrån däcken. En helhetsbild av slutresultatet kan ses i figur 6.

²Spårvidd är avståndet mellan centrum på två motstående hjul.

³Kuggmodul beskriver förhållandet mellan kuggdiamater och antal kuggar.



Figur 6: Motorfäste med fälgar, hjul och växellåda. Eget material.

3.3 Jämförelse och analys av sensorer

För att kunna mäta och kontrollera position, hastighet, acceleration och robotens rotation används en högupplöst pulsgivare, det vill säga att pulsgivaren skickar ut många pulser per roterat varv. Modellen som valdes är IE2-512 [23] från Faulhaber som med växellådan och hjuldiametern ger en teoretisk noggrannhet på under 1 mm. Att den är högupplöst ger stor noggrannhet men orsakade även ett problem, som åtgärdades genom att använda en mikrokontroller med inbyggt stöd för pulsgivare.

För att beräkna robotens rotation kring z-axeln beräknades differensen mellan pulsgivarna med hjälp av avståndet mellan hjulen och antal pulser per varv, vilket ger robotens vinkel.

För att möjliggöra robotens kartläggning av labyrinten och undvika kollisioner behövdes avståndssensorer. I detta projekt stod valet mellan IR-sensorer och laser-sensorer. Ultraljuds-sensorer valdes bort på grund av sin stora storlek i förhållande till de andra alternativen. Avståndssensorerna monterades i tre riktningar, rakt fram och åt sidorna på roboten. Eftersom alla aktuella sensorer är begränsade i hur nära de kan mäta, monterades de 20 mm in på roboten för att kompensera för det minimala mätavståndet. Detta används för att sensorerna ska kunna förse regleringen med återkoppling även då roboten befinner sig precis intill väggen. I tabell 2 kan tre sensormodellers specifikationer ses.

Tabell 2: Viktiga parametrar hos olika avståndssensorer, tagna från datablad, se bilaga D, E och F.

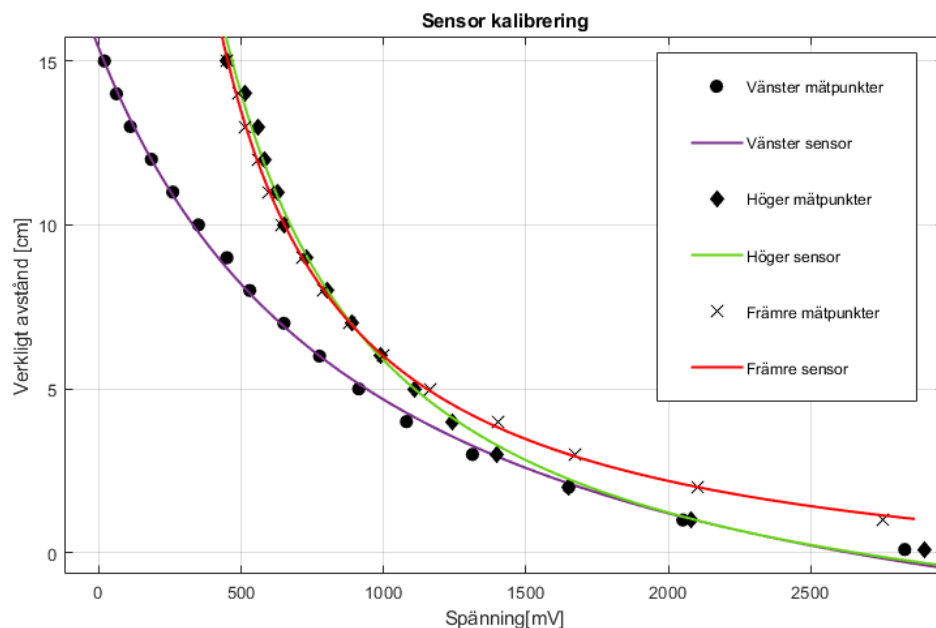
Sensor	Min mätdistans [mm]	Max mätdistans [mm]
VL53L0X(Laser)	40	2000
GP2Y0A51SK0F(IR)	20	150
GP2Y0A21YK0F(IR)	100	800

Utifrån parametrarna angivna i tabell 2, kunde IR-sensorn GP2Y0A21YK0F uteslu-

tas på grund av den långa minimala mätavståndet. Av de två återstående sensorerna, VL53L0X och GP2Y0A51SK0F valdes GP2Y0A51SK0F då den är snabbare och kan mäta på närmare avstånd. Dessutom använder VL53L0X en API som tar mycket resurser av CPU:n, vilket också gör den svårare att arbeta med.

Datan som ges från GP2Y0A51SK0F är ett analogt spänningsvärde. Databladet i bilaga E rekommenderar att en kondensator placeras mellan spänningkällan och jord, då sensorn kan dra mer ström än vad mikrokontrollern kan leverera. Detta förhindras av en kondensator eftersom den är strömtrög.

Ickelinjära värden tas emot via en analog ingång, till dessa värden har en ekvation approximerats genom att använda verktyget cftools [24] i MATLAB. Ekvationen används för att konvertera spänning till avstånd från en vägg, i detta fallet i ett intervall av 20 till 150 mm. För att minska beräkningarna som sker i mikrokontrollern valdes en rationell kurva över en exponentiell eftersom den exponentiella kräver ett bibliotek som tar mer resurser. Ekvationen är på formen $\frac{p_1 \cdot x + p_2}{x + q_1}$, där x är mätvärde, p_1 , p_2 och q_1 är konstanter framtagna av verktyget. I figur 7 redovisas kurvorna för 3 olika GP2Y0A51SK0F.



Figur 7: Kurvor som ligger som grund till ekvationerna för att konvertera spänningsvärde till avstånd. Eget material.

3.4 Batteri

Kraven på energikällan kommer ifrån den summerade energiåtgången från motorerna, sensorerna, mikrokontrollen och huvuddatorn. För att underlätta utvecklingen och öka modulariteten delades energiförsörjningen upp i två separata delar, en för huvuddatorn där programvaran körs och en för resterade komponenter.

Energiförsörjningen till huvuddatorn varierar kraftigt beroende på till exempel arbetsbelastning och om wifi eller bluetooth används. Energiförbrukningen estimerades utifrån empiriskt uppmätt data [25] till en snittförbrukning på 150 mA vid 5 V. Eftersom effektkravet var lågt valdes ett litet LiPo batteri [26] och en spänningsregulator, Power Boost 500 Basic från Adafruit [27]. Med en total energi på 0,925 Wh i batteriet beräknades batteritiden till drygt en timme, vilket är mer än nödvändigt vid tävling.

Motorerna är den del av roboten som med marginal har störst energiförbrukning. Enligt simuleringarna drar motorerna maximalt 1,2 A vardera, vilket kan jämföras med mikrokontrollen som maximalt drar 120 mA. Detta gjorde att motorerna var den del som satte krav på batteriet. För utvärdering och utveckling var en ökad batteritid önskvärd, men att ha två batterier som byts ut och laddas ansågs vara tillräckligt. Med en beräknad snittförbrukning för motorer, sensorer och mikrokontroller på ca 6,5 W och en säkerhetsmarginal med en faktor 1,5 estimerades den totala snittförbrukningen att bli ca 10 W vid höga accelerationer och hastigheter. Utifrån snittförbrukningen, de momentana strömkraven och spänningskraven från enheterna beräknades krav på batteriets egenskaper, vilka kan ses i tabell 3.

Tabell 3: Fastställda krav på batteriet för roboten.

Krav	Värde
Energi	$> 1,67 \text{ Wh}$
Momentan Ström	$> 3,75 \text{ A}$
Spänning	5 - 9 V

Utifrån dessa krav och önskemål om att minimera batteriets storlek kunde batteriet väljas till Turnigy Nano-Tech 300 mAh 2S 7,4 V [28]. Batteriet parades ihop med en linjär spänningsregulator av modell LM7805 [29] för att förse mikrokontroller och sensorer med en stabil spänning på 5 V.

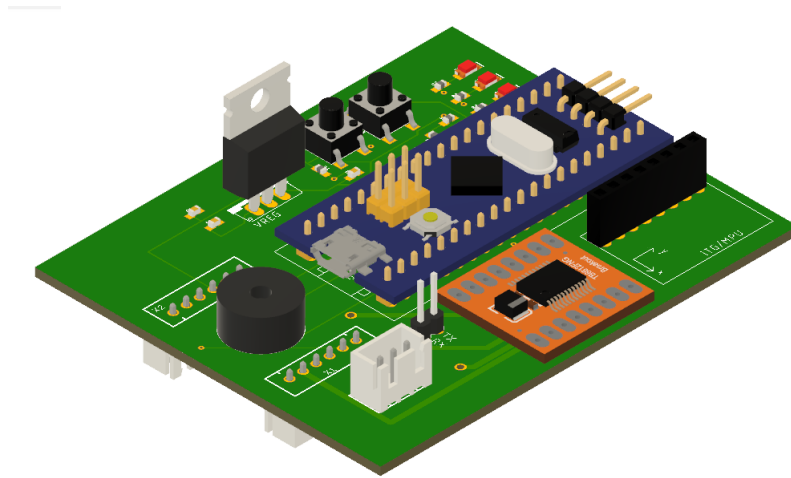
3.5 Val av mikrokontroller och huvuddator

För att kontrollera motorer, periferienheter och bearbeta data från sensorerna användes en mikrokontroller. På grund av att pulsgivarna var högupplösta krävdes en mikrokontroller som kunde hantera detta, därför valdes en så kallad Bluepill som är ett kretskort för att driva en mikrokontroller av modellen STM32F103C8 [30]. Stödet för pulsgivarna är implementerade med så kallade räknare och möjliggör läsning av specifikt register i mikrokontrollen för att få antal pulser, vilket medför att ingen mjukvara för avbrott behövs.

Valet av huvuddatorn baserades på storlek, möjligheter för kommunikation och wifi där Raspberry Pi Zero W [12] valdes. Datorn kör ett operativsystem av typen Linux [31] där programspråket Python kan köras vilket möter kraven för huvuddatorn specificerade i avsnitt 1.3.1.

3.6 Design av kretskort

Under prototypfasen användes kopplingsdäck för att göra elektriska kopplingar mellan de olika enheterna. Detta skapade problem då kontakterna var opålitliga. Efter mycket felsökning av kablar och kontakter designades ett kretskort för att skapa mer pålitliga kopplingar. För att underlätta interaktionen med roboten utökades kretskortet med knappar, summer⁴ samt ledlampor. Kretskortet designades i Eagle [32] från Autodesk och en 3D rendering av kretskortet kan ses i figur 8. Se bilaga G och bilaga H för elschema och kretskortsritning.

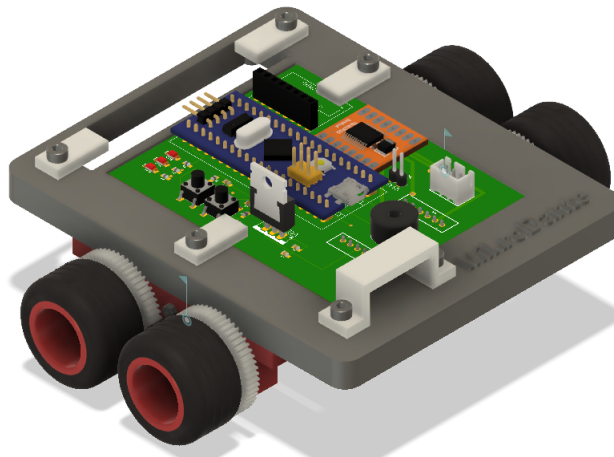


Figur 8: Kretskortet populerat med komponenter. Eget material.

3.7 Initial prototyp av robot

För att utvärdera och testa hårdvaran i väntan på att motorerna skulle levereras designades en initial plattform i CAD med programmet Fusion 360 [33] från Autodesk. Plattformen tillverkades i en FDM 3D-skrivare. Plattformen består av kretskortet, mikrokontrollen samt de komponenter som krävs för att styra roboten, dock inte huvuddatorn. Motorerna som användes på denna plattform lånades av Chalmers robotförening. Med prototypen utvecklades drivrutiner för att övervaka batteriets spänning, styra motorerna, reglera position, interagera med knappar och indikera med lysdioder.

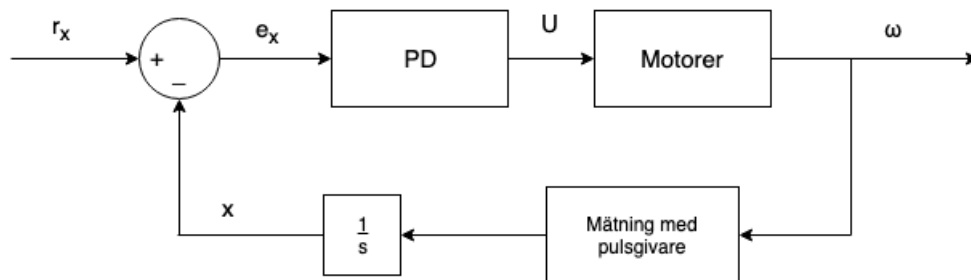
⁴En summer är en piezoelektrisk komponent som skapar entonigt ljud.



Figur 9: Initial CAD-modell på prototyp. Eget material.

3.8 Implementation av initialt reglersystem

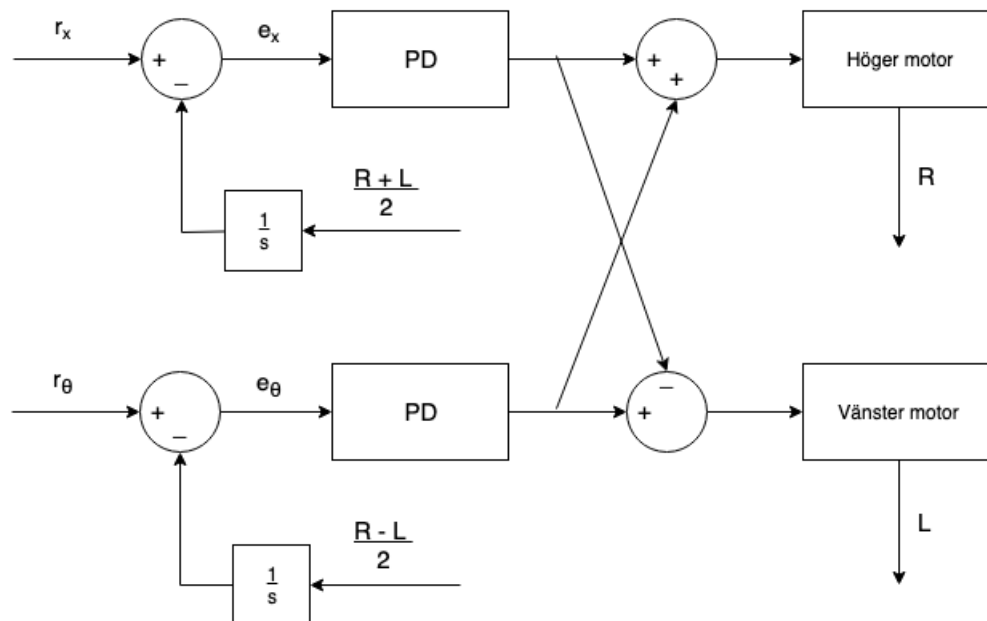
För att ta sig runt i labyrinten krävs att roboten kan ta sig till en given cell, detta görs genom positionsreglering. Positionsregleringens återkoppling kommer från motorernas pulsgivare som ger ett antal pulser per varv som sedan registreras av mikrokontrollen. Genom att räkna antalet pulser sedan tidigare avläsning och mäta tiden mellan avläsningarna bestäms hastigheten på hjulen som sedan integreras till en sträcka. Regulatorn som initialt implementerades är en PD regulator. En schematisk representation av regulatorn kan ses i figur 10.



Figur 10: Initial positionsreglering där en likadan styrsignal skickas till de båda motorerna. Eget material.

Faktorer som tillverkningen av motorerna, olika friktion i växellådorna och små variationer av storleken på hjulen leder till att translationshastigheten skiljer sig på de olika sidorna av roboten. Detta medför att roboten kommer att börja svänga åt ett håll, då ingen reglering för rotation används. På grund av detta implementerades en reglerloop för rotation vars ändamål är att utföra kontrollerade svängar i en korsning och att köra rakt. Börvädet kan antingen sättas till specificerad vinkel vid sväng eller till 0° vid raksträckor. Återkopplingen av vinkeln kommer från att differensen från pulsgivarna (roationshastighet) integreras, på liknande sätt som vid

positionsregleringen. En schematisk representation över det initiala reglersystemet kan ses i figur 11.



Figur 11: Reglersystem för position och rotation. Motorernas rotation mäts med pulsgivarna, där R och L motsvarar antal pulser sedan senaste mätning. Eget material.

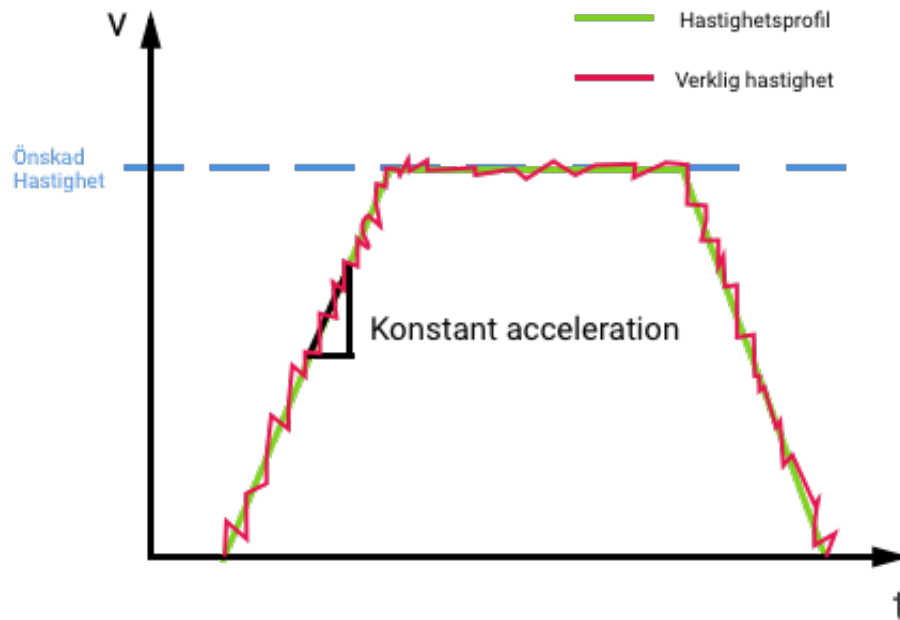
Vid rak körning valdes vinkeln r_θ till 0° och translationsavståndet r_x till önskad sträcka. Regulatorerna samverkar för att behålla robotens vinkel och därmed köra rakt. På liknande sätt bestäms vinkeln till 90° och translationsavståndet till 0 mm för att rotera åt höger utan att påverka positionen.

3.9 Implementation av slutgiltigt reglersystem

Ett stor del av robotarna i tävlingar använder sig av hastighet- och accelerationsreglering för att ta sig runt snabbare. Hastighet- och accelerationsregleringen ser till att roboten kan bibehålla maximal hastighet under längre tid än utan regleringen. En annan fördel är att vid körning av känd bana kan en optimal inbromsning göras. Genom att veta avståndet till en sväng, nuvarande hastighet och önskad hastighet vid början av sväng kan retardationen som krävs för inbromsning beräknas. Eftersom maximal retardation är känd vet roboten när inbromsning ska starta. Beräkningen för detta ges av ekvation (11).

$$\begin{aligned}
 v &= v_0 + at \\
 s &= \frac{v_0 + v}{2} \cdot t \\
 \text{Utbrytning av } t \text{ ger:} \\
 v^2 &= v_0^2 + 2as \iff a = \frac{v^2 - v_0^2}{2s}
 \end{aligned} \tag{11}$$

Hastighetsregleringen och accelerationen var från en början tänkta att implementeras med kaskadkopplade⁵ regulatorer. Denna typ av regleringen visade sig ha flera problem. Eftersom utsignalen från positionsregulatorn är proportionerlig mot avståndet gavs för höga hastigheter och accelerationer vid långa sträckor. Därför begränsades styrsignalerna till max- och minvärden. De strypta regulatorerna i samband med att de var kaskadkopplade gjorde det mycket svårt att justera PID parametrarna och få ett väl fungerande system. På grund av detta övergavs konceptet och istället implementerades en profilgenerator som kontrollerar hastighet och acceleration. Konceptet om en profilgenerator beskrivs vidare i en artikel [34] om diagonalspurter i Micromouse och i föreläsningar av Green Yee [35]. Huvudtanken är att åstadkomma konstant acceleration och retardation vid förändringar av hastighet. Istället för en stegfunktion⁶ skapar profilgeneratorn en hastighetskurva som efterliknar en parallelltrappets, vilket skapar en konstant acceleration till den önskade hastigheten, detta kan ses i figur 12.



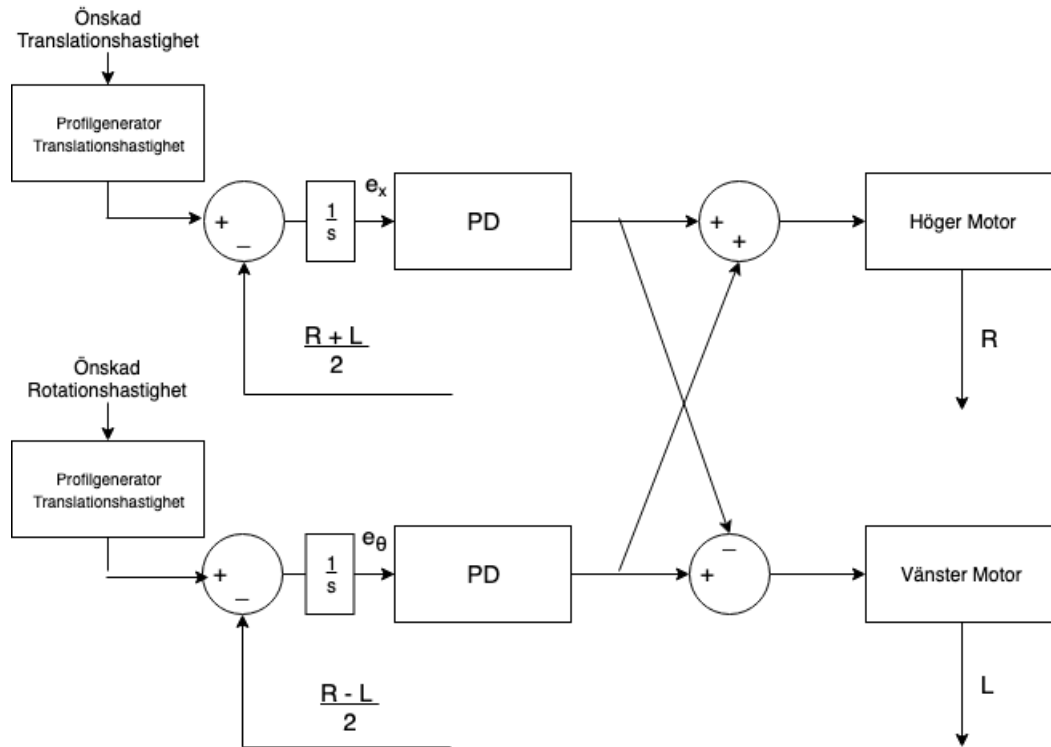
Figur 12: Schematisk illustration av en hastighetskurva skapad av profilgeneratorn med konstant acceleration för att uppnå önskad hastighet. Areorna av den ideala och den verkliga är lika stora. Eget material.

Då maximal acceleration och retardation är känd skapas profilen genom att kontinuerligt öka, respektive minska nuvarande hastighet med hjälp av accelerationen, respektive retardationen. Regulatorns börvärde kommer från profilgeneratorn och återkopplingen från den nuvarande hastigheten. Felet integreras sedan för att se till

⁵Kaskadkoppling inom reglering betyder att regulatorerna kopplas i serie för att tillsammans styra flera sammanhängande parametrar i ett system, t.ex position, hastighet och acceleration.

⁶I en stegfunktion övergår nuvarande värde till önskat värde omedelbart. Grafen kan efterliknas som ett trappsteg, därav namnet.

att roboten förflyttar sig lika långt som den teoretiska profilen anger. Detta betyder att arean, den förflyttade sträckan, på den teoretiska profilen blir lika stor som den verkliga. För att åstadkomma en positionsreglering används hastighetsregleringen i samband med ekvation (11). På samma sätt utvecklades en profilgenerator för rotationshastigheten och en schematisk representation av det slutgiltiga reglersystemet kan ses i figur 13.

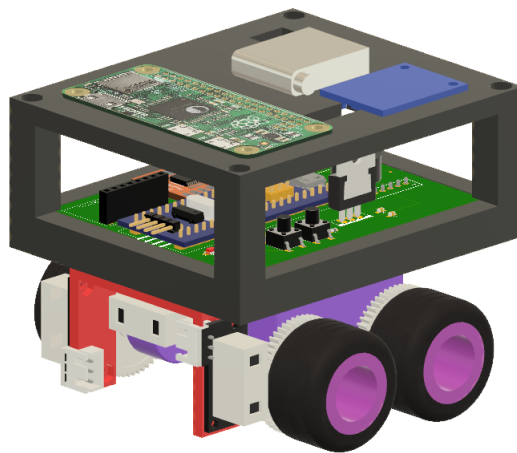


Figur 13: Schematisk representation av det slutgiltiga reglersystemet för att kontrollera position, hastighet och acceleration. Eget material.

3.10 Slutgiltig konstruktion

Den slutgiltiga designen på roboten gjordes utifrån de köpta motorerna och ett antal revisioner gjordes i CAD och 3D skrivare. En helhetsbild av roboten kan ses i figur 14.

Designen har två höjdnivåer, en för mikrokontroller med periferienheter och en för huvuddatorn, batterier och spänningsregulator. Sensorena är placerade under främre delen av plattformen vilket gör att roboten kan se alla väggar tidigare än om de hade monterats i den bakre delen. Roboten valdes att göras hög istället för bred och lång, för att minimera ytan i labyrinten.



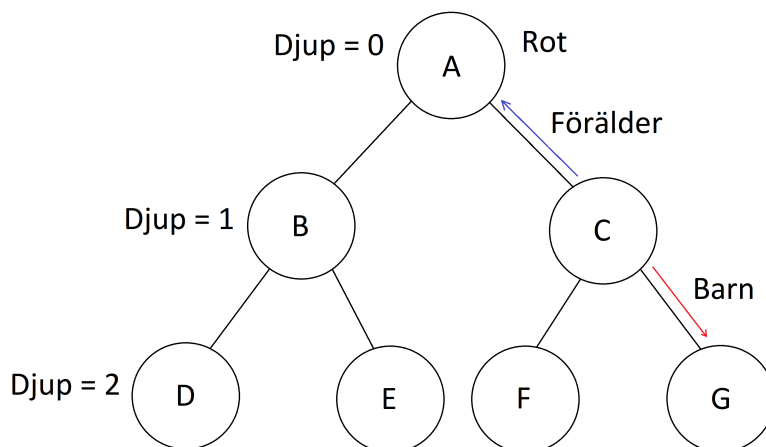
Figur 14: Slutgiltig 3D-modell av robot. Eget material.

4 Datastrukturer och algoritmer

En effektiv och välkonstruerad programvara grundar sig i genomtänkta datastrukturer och algoritmer som gör att programmet arbetar snabbt och inte använder mer minne än nödvändigt. För att roboten ska kunna lösa labyrinten på så kort tid som möjligt är det viktigt att programvarans beräkningstid inte begränsar hårdvarans kapacitet. På grund av detta är det intressant att studera befintliga datastrukturer och algoritmer som är lämpade för denna typ av problem, och undersöka hur dessa utnyttjas på ett gynnsamt sätt.

4.1 Grafteori

En graf definieras som en datamängd representerad av noder och kanter som länkar dessa till varandra [36]. Detta är en datastruktur som används inom programmering och den är effektiv för lagring av information som har ett tydligt inbördes förhållande. I figur 15 visas en underkategori av grafer, ett så kallat träd, där den inbördes relationen mellan datan är hierarkisk och består av förälder och barn. Förhållandet gäller oavsett djup och då ett barn har djupet n så har dess förälder djupet $n - 1$. Noden som är längst upp i trädet, med djupet 0, kallas för rot och den har inga föräldrar.



Figur 15: En schematisk illustration av datastrukturen träd. Eget material.

En systematisk genomsökning av en alla noder i en graf kallas för traversering. Ett sätt att traversera genom en graf är att använda algoritmen Breadth-First Search [36], som besöker alla noder på samma djup innan den går vidare till nästa nivå. En genomsökning av en graf med denna metod resulterar därmed i en symmetrisk genomlöpning av noderna med startnoden i centrum av det genomsökta området.

4.2 Algoritmer för genomsökning av datastrukturer

Det finns ett stort antal algoritmer som kan användas för att hitta en väg från en punkt till en annan och alla dessa algoritmer har styrkor och svagheter. Även om de alla kan användas för att hitta en rutt genom ett område så lämpar sig inte alla algoritmer till alla typer av problem. Valet av algoritm bör baseras på det aktuella problemets förutsättningar. Detta beror på att den givna informationen bör utnyttjas maximalt för att underlätta sökningen, men även att den implementerade algoritmen måste tillgodoses med all information den kräver för att arbeta effektivt.

Med grund i de förutsättningarna som beskrivs i avsnitt 1.3.2, finns det en kategori av algoritmer som är lämpliga att studera, nämligen Best-First Search. Det är en omfattande grupp av algoritmer med olika fördelar som är gynnsamma för problemet. Alla algoritmer i denna kategori tillämpas för genomsökning av datastrukturer och grundar sig i traversering av grafer. Karaktäristiskt för dessa är att de genomför en informerad sökning. Gemensamt för alla algoritmer som tillhör den här kategorin är att en funktion $f(n)$ används för att tilldela alla noder ett värde och den nod med lägst värde utses till den som bör besökas näst. Det som skiljer algoritmerna åt är hur funktionen $f(n)$ definieras och det resulterar i olika beteende hos de olika algoritmerna. Nedan presenteras två algoritmer som båda är av typen Best-First Search, men som har olika sätt att värdera noderna. Dessa är Greedy Best-First Search och A* [37].

Greedy Best-First Search är en algoritm som baserar beslut på vad som är bäst för tillfället. Algoritmer vilka värderar noder på detta sätt benämns som giriga algoritmer. En girig algoritm hittar lätt lösningar som är lokalt optimala, men de är mindre lämpliga om syftet är att hitta den globalt optimala lösningen. En fördel med giriga algoritmer är att de ofta hittar en väg till målet utan att genomsöka ett stort område, men de hittar sällan den bästa vägen. Funktionen $f(n)$ sätts i denna algoritm till $f(n) = h(n)$, där $h(n)$ är en uppskattad kostnad⁷ från den aktuella platsen till slutnoden och detta görs genom att använda en heuristik [38]. En heuristik är ett sätt att uppskatta en lösning på ett problem då en exakt lösning inte kan bestämmas, vilket nyttjas i många Best-First Search algoritmer. I sektion 4.3 finns två relevanta heuristiker presenterade.

4.2.1 A*

A* är en mer komplex version av Best-First Search som utöver att beräkna $f(n)$ baserat på var målet finns, även tar hänsyn till avståndet från start. Detta gör att den är mer lämplig för att hitta en kort väg från en särskild startposition till ett specifikt mål, snarare än att hitta en kort väg till det specifika målet utifrån en godtycklig position, i motsats till Greedy Best-First Search. Funktionen $f(n)$ beräknas alltså enligt $f(n) = g(n) + h(n)$, där $g(n)$ är kostnaden från start till

⁷Med kostnad menas resurs- eller tidsåtgång.

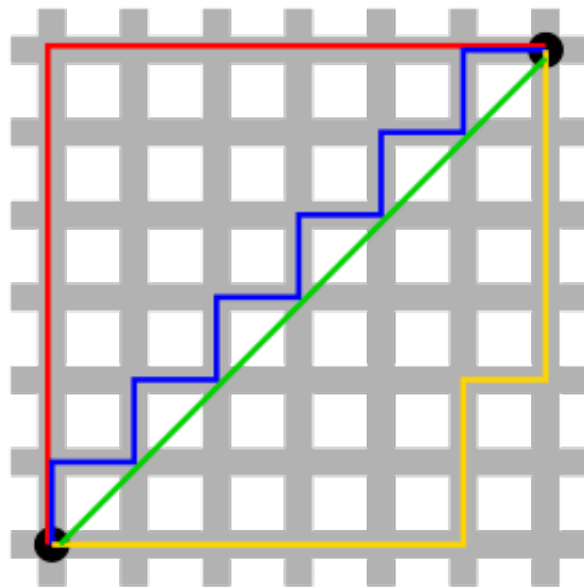
nuvarande nod. A^* tenderar att ofta hitta den kortaste vägen, eller åtminstone en väg som är nära den optimala vägen, men i utbyte mot detta genomsöker den ett större område vilket är en nackdel för detta syfte, då det är viktigt att minimera tiden det tar att genomsöka labyrinten.

Beräkningen av funktionen $f(n)$ är väldigt viktig då algoritmens prestation avgörs av dess värde. En viktig aspekt vid beräkandet av $f(n)$ är förhållandet mellan $g(n)$ och $h(n)$ [39]. I de fall då $g(n) \ll h(n)$ beter sig A^* som en Greedy Best-First Search, men om $g(n) \gg h(n)$ beter den sig istället som Dijkstras algoritm. Denna algoritm fokuserar på att hitta den kortaste möjliga vägen till en viss destination, men den genomsöker en väldigt stor del av eller hela det tillgängliga området, vilket gör att den inte lämpar sig att användas i detta projekt. Inget av dessa två scenarion är önskvärda för att lösa en labyrint där positionen av start- och slutpunkt är känd, och därför behöver $g(n)$ och $h(n)$ vara av samma storleksordning.

I en okänd miljö går inte $h(n)$ att bestämmas exakt, utan det tilldelade värdet kommer enbart vara en uppskattning av kostnaden för att röra sig från den aktuella positionen till målet. Hur denna uppskattning görs har en stor inverkan på tiden det tar att hitta en väg och hur bra den funna vägen är i förhållande till den optimala rutten. Genom att överskatta $h(n)$ kommer sökningen ske skarpt mot målet och omgivningen lämnas okänd. Detta gör att det går snabbare att hitta en väg till målet, men i utbyte kommer den funna vägen oftast inte vara optimal. Ju mer $h(n)$ överskattas, desto mer lik kommer algoritmen bli Greedy Best-First Search och till slut nås det tidigare nämnda fallet där $g(n) \ll h(n)$ och algoritmen övergår helt till en Greedy Best-First Search. Om $h(n)$ istället underskattas blir sökningen mindre riktad och ett bredare område inkluderas i sökningen vilket låter algoritmen hitta en bättre lösning och ofta den optimala lösningen, men sökningen kommer även att ta längre tid.

4.3 Heuristik för uppskattning av avstånd

Då labyrinten kan betraktas som ett rutnät av noder som skiljs åt av väggar finns det flera sätt att uppskatta $h(n)$, så kallad heuristik och dessa sätt är baserade på vektornormer. Den vanligaste vektornormen, den euklidiska vektornormen, följer ur Pythagoras sats och definieras enligt $\|\mathbf{x}\|_2 = \sqrt{\sum_{n=1}^N |x_n|^2}$. I scenarion då diagonala förflyttningar ej är möjligt är det mer lämpligt att använda manhattannormen. Manhattannormen beskrivs enligt $\|\mathbf{x}\|_1 = \sum_{n=1}^N |x_n|$. Genom att inkludera en parameter D så att $h(n) = D \cdot \|\mathbf{x}\|_1$, justeras enkelt kort söktid kontra optimal väg genom att öka eller minska D . För att fokusera på att finna den kortaste vägen bör D vara den minsta möjliga kostnaden för att förflytta sig mellan närliggande noder, vilket här kommer ge $D = 1$. I figur 16 illustreras den euklidiska normen och manhattannormen, med $D = 1$, grafiskt. För den euklidiska normen är $n = 2$, och $h(n)$ beräknas då av Pythagoras sats. Manhattannormen är enbart förflyttningen i x- och y-led och därmed fås flera rutter där $h(n)$ förändras ekvivalent.



Figur 16: Det euklidiska avståndet visas i grönt och avståndet enligt manhattan-normen visas i rött, blått och gult. Återgiven från [40].

5 Programvara för lösning av labyrint

Programmet som används för att kartlägga labyrinten och hitta en väg från startpunkten till målet skrevs i Python. Sökalgoritmen som används baseras på en algoritm av typen Best-First Search, som beskrivs i avsnitt 4.2. Där presenteras algoritmerna A* och Greedy Best-First Search, och jämförs genom att simulera lösningsmetoden för 50 slumpmässigt genererade labyrinter. I avsnitt 7.1 återfinns resultatet från dessa simuleringar. Från resultatet syns att Greedy Best-First Search oftast hittar målet på färre steg än A*, vilket innebär en kortare söktid i labyrinten. Dessutom tenderar den att hitta en längre väg än vad A* gör. Eftersom A* hittar en kortare väg beslutades att denna algoritm skulle implementeras.

För att underlätta utvecklingen av programmet delades implementationen upp i delmoment. Inledningsvis användes algoritmen right-hand rule, vilket innebär att roboten alltid svänger höger då möjligheten ges. Detta är en mycket enkel algoritm att implementera och genom att använda den kunde fokus läggas på övrig funktionalitet i programmet, till exempel att skapa modell av labyrinten och roboten, samt översättning mellan robotens perspektiv och koordinater.

5.1 Beskrivning av slutgiltig programvara

Programmet designades för att få information om den nuvarande cellens väggar från sensorerna. Därefter bearbetas informationen och en instruktion som säger vad roboten ska göra näst genereras. Inför spurtfasen genereras en serie instruktioner som ger den kortaste vägen mellan start och mål. För att verifiera att programmet fungerade som tänkt användes labyrinter som använts i tidigare tävlingar [41].

5.1.1 Programmets modell av verkligheten

En virtuell modell av roboten och dess omgivning konstrueras i programmet. Modellen består av celler, labyrinten och roboten, vilka representeras av objekt. Det objekt som motsvarar labyrinten innehåller information om labyrintens storlek, robotens startposition och målets placering. Objektet har även egenskaper som pekar på den yttersta raden och kolonnen i det utforskade området. Detta område utgör ett rektangulärt område i labyrinten som innesluter alla besökta celler. Pekarnas användningsområde beskrivs i avsnitt 5.1.2. Utöver detta finns i labyrintobjektet en matris med samma storlek som den fysiska labyrinten där varje element är ett cell-objekt. Varje sådant objekt har egenskaper som representerar status för om cellen är besökt, värde på $f(n)$, $h(n)$ och $g(n)$, samt i vilka riktningar cellen har väggar. Informationen om cellens väggar är en sträng med binär representation av väggarna med formatet norr, syd, väst och öst. Vilket värde strängen ska anta ges av sensorerna via mikrokontrollern. Denna representation av väggarna är från robotens perspektiv och ges på formatet fram, bak, vänster och höger. Det innebär att en översättning

mellan de två formaten behövdes göras; en hjälpmetod som konverterar informationen, till rätt format, beroende på robotens riktning. I figur 17 illustreras hur de två formaten relaterar till varandra. Initialt är alla celler utan väggar, med undantag för ytterväggar vilka betraktas som kända.

I objektet som representerar roboten lagras dess position i labyrinten, dess riktning samt det nuvarande värdet på $g(n)$. Riktningen anges som vädersträck. När en instruktion om förflyttning skickas till mikrokontrollern uppdateras robotobjektets position, riktning samt värde på $g(n)$. Den cell som roboten förflyttar sig till tilldelas det värde på $g(n)$ som roboten har. Om roboten besöker en tidigare besökt cell återställs robotens värde på $g(n)$ till samma som cellens.

	Höger Norr	
Fram Väst	←	Bak Öst
	Vänster Syd	

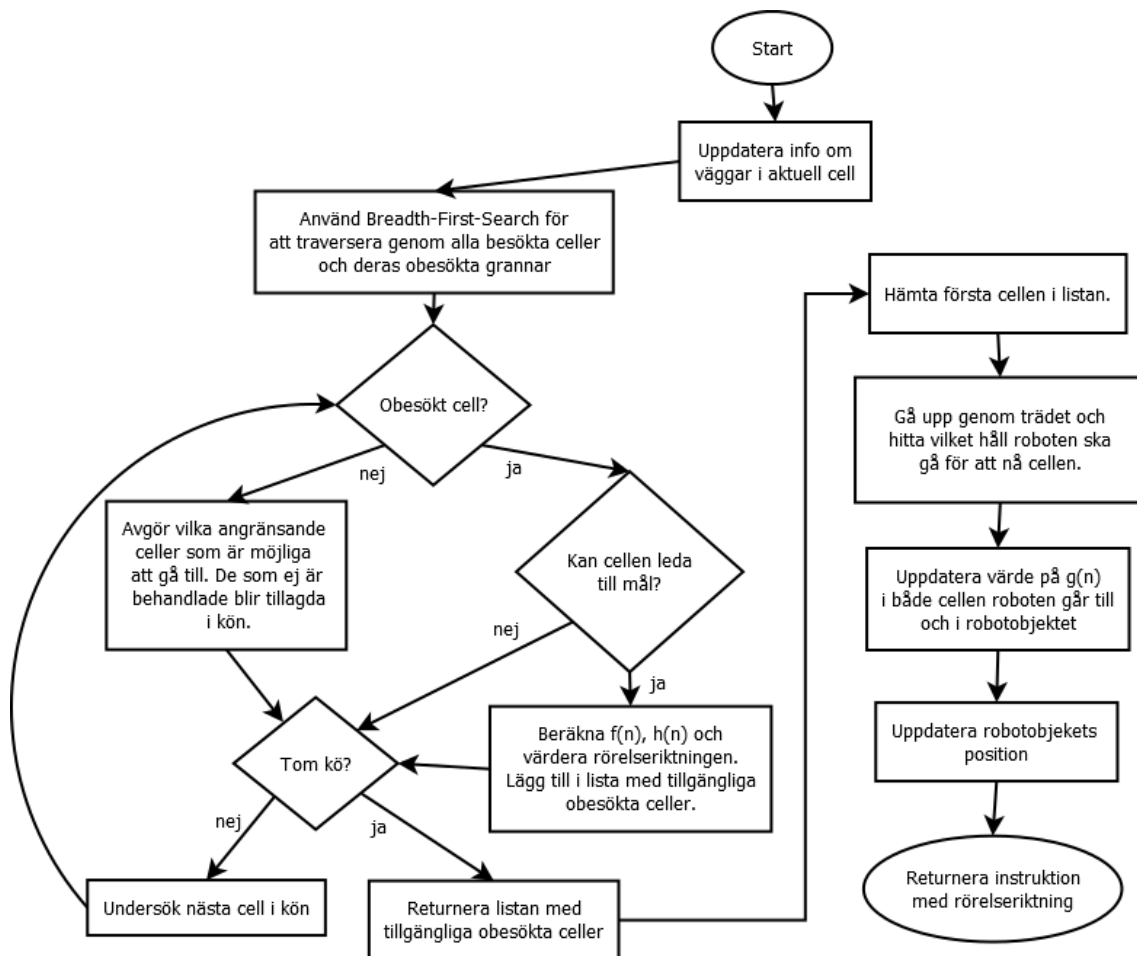
Figur 17: Figuren representerar hur de två formaten förhåller sig till varandra. Roboten representeras av pilen med riktning åt väst. Eget material.

5.1.2 Generering av instruktion

Från att kartläggningsfasen startas till att roboten nått målet anropas sökalgoritmen som returnerar en instruktion med rörelseriktning till roboten. För att avgöra vilken riktning roboten bör köra konstrueras en graf av noder med hjälp av algoritmen Breadth-First Search. Från grafens rot konstrueras följder av noder som representerar vägar till obesökta celler. Detta innebär att grafens rot är robotens aktuella position. I figur 18 illustreras programmets flöde när en instruktion genereras, och det förklaras mer i detalj nedan.

En nod motsvarar en cell, och har egenskaper som representerar $f(n)$, $h(n)$ samt grafens djup, det vill säga antalet noder till grafens rot vilket motsvarar antalet steg från nodens placering till roboten. När en ny nod genereras länkas den till sin förälder, vilket är den föregående noden i grafen. Noden har en annan egenskap som beror av i vilken riktning den är placerad i förhållande till dess förälder. Om rörelseriktningen är rakt fram ges värdet 0, annars tilldelas värdet 1, för att vid förflyttning kunna ta rörelseriktning i beaktning.

Grafen används för att hitta alla, av roboten, ej besökta celler som går att nå. När en nod motsvarar en ej tidigare besökt cell kontrolleras ifall den kan leda till målet eller ej. Detta görs genom en ny Breadth-First Search med syfte att hitta målet istället för obesökta celler. I de fall sökningen inte kan ta sig utanför det utforskade området antas cellen vara en del av en återvändsgränd. Om sökningen går utanför det redan utforskade området antas målet vara åtkomligt från noden. De noder som kan leda till målet sparas i en lista där de sorteras baserat på tre värden: i första hand $f(n)$ adderat med grafens djup, i andra hand $h(n)$ adderat med grafens djup och slutligen värdet baserat på rörelseriktningen.



Figur 18: Flödesschema för programmets instruktionsgenerering där nästlade funktioner inte är specificerade. Eget material.

För att förändra girigheten hos algoritmen har $f(n)$ definierats som $f(n) = g(n) + D \cdot h(n)$, där olika värden på D har jämförts, se resultat avsnitt 7.2, för att med statistisk analys bestämma ett värde där antalet steg kan väntas vara lägst.

Från listan hämtas den nod som anses vara bäst att besöka. Genom att stega uppåt genom trädet hämtas rotens barn som leder till den hämtade noden. Riktningen mellan dessa två noder bestäms och skickas som instruktion till roboten.

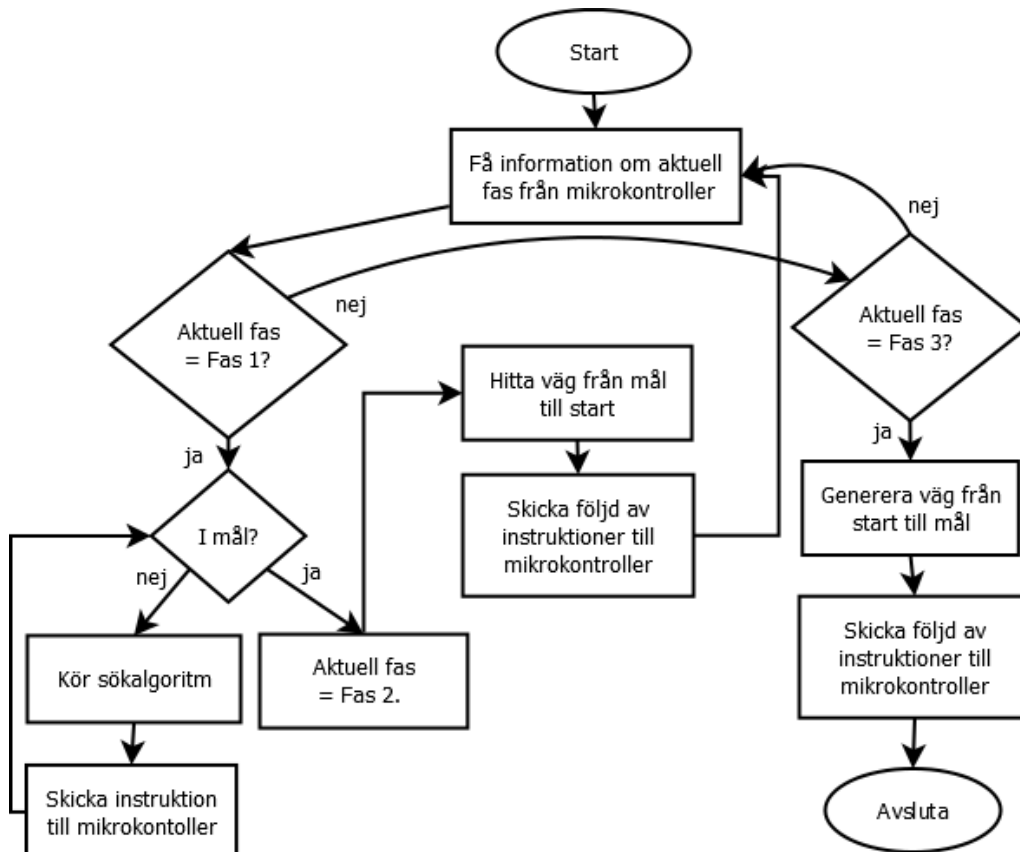
5.1.3 Körning efter en följd av instruktioner

När roboten nått målet finns alltid minst en känd väg mellan mål och start. För att bestämma vilken som är kortast skapas en ny graf som bildar vägar till start och den kortaste vägen tas ut. Genom att stega upp genom grafen och i varje steg spara riktningen mellan nod och dess förälder, skapas en lista med instruktioner som ger hela vägen från mål till start. Med dessa instruktioner tar roboten sig tillbaka till start. Då spurtfasen inleds används samma lista med instruktioner, men den inverteras så att den leder från start till mål. När roboten erhållit dessa instruktioner kör den så snabbt som möjligt till mål.

6 Kommunikation mellan huvuddator och mikrokontroller

Kommunikationsprotokollet UART [42] har använts för att möjliggöra kommunikation mellan mikrokontrollern och huvuddatorn. Protokollet skickar en sträng av karaktärer där varje karaktär blir till ett datapaket. Datapaketen är en byte långa och avslutas med en förbestämd stoppbit. Stoppbiten används för att markera att hela datapaketet har skickats och därmed blandas inte olika paket ihop. När hela strängen är skickad avslutas sändningen med att `\n` skickas, vilket är ett kommando för att påbörja en ny rad.

Då programvaran på både huvuddatorn och mikrokontrollen är beroende av varandra väntar båda sidor på information från andra parten för att fortsätta i programkoden. I figur 19 finns ett flödesschema över hur kommunikationen är uppbyggd på huvuddatorn.

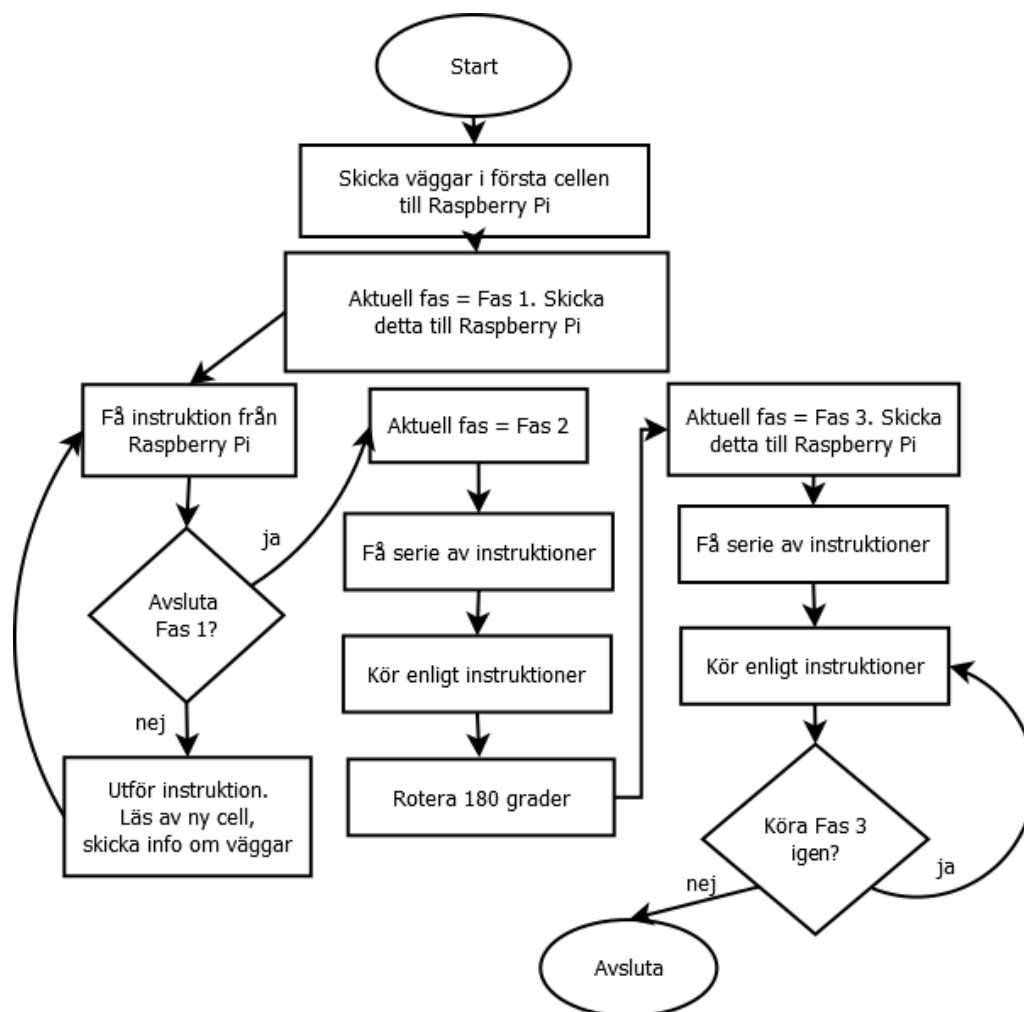


Figur 19: Flödesschema över hur sändare och mottagare är programmerade på huvuddatorn. Eget material.

I Python har den färdiga modulen pySerial [43] implementerats för att skicka och läsa information från TX och RX pinnarna på datorn Raspberry Pi Zero W. pySerial består av många kommunikationsprotokoll och det som används är simpla

grundfunktioner som att läsa och skriva. Det krävs att informationen som skickas till huvuddatorn har specificerats i förväg så att programmet vet hur många bytes som ska tas emot.

För att kunna styra den autonoma roboten används två fysiska knappar på kretskortet som är kopplade till mikrokontrollern. En av knapparna används för att starta programmet från början. När programmet avslutas kan den andra knappen användas för att starta programmet vid Fas 3. I figur 20 visas flödet i programmet som sköter kommunikationen på mikrokontrollern. Faserna i schemat används för att kommunikationsprotokollet behöver mottaga ett bestämt antal datapaket. Vid kartläggningen skickas en instruktion åt gången medan en sträng som beskriver hela vägen från start till mål skickas i spurtfasen.



Figur 20: Flödesschema över hur sändare och mottagare är programmerade på mikrokontrollern. Eget material.

7 Resultat från simuleringar av programvara

För optimering av programvaran genomfördes simuleringar av olika algoritmer och olika modifieringar av dessa. Simuleringarnas resultat låg till grund för beslut som togs vid utformningen av programvaran. Slutligen testades även den färdiga programvarans prestanda.

7.1 Jämförelse av A* och Greedy Best-First Search

För att jämföra algoritmerna A* och Greedy Best-First Search⁸ gjordes en simulering med mjukvara av Xueqiao Xu [44]. För generering av labyrint användes Prim's algoritm [45].

Tabell 4: Jämförelse mellan A* och Greedy Best-first-search, genomsnittligt resultat taget från 50 slumpade labyrinter.

Algoritm	Kortaste väg	Totalt
A*	46 steg	378 steg
GBFS	66 steg	224 steg

För att jämföra heuristik för A* användes samma mjukvara och generering av labyrint som ovan.

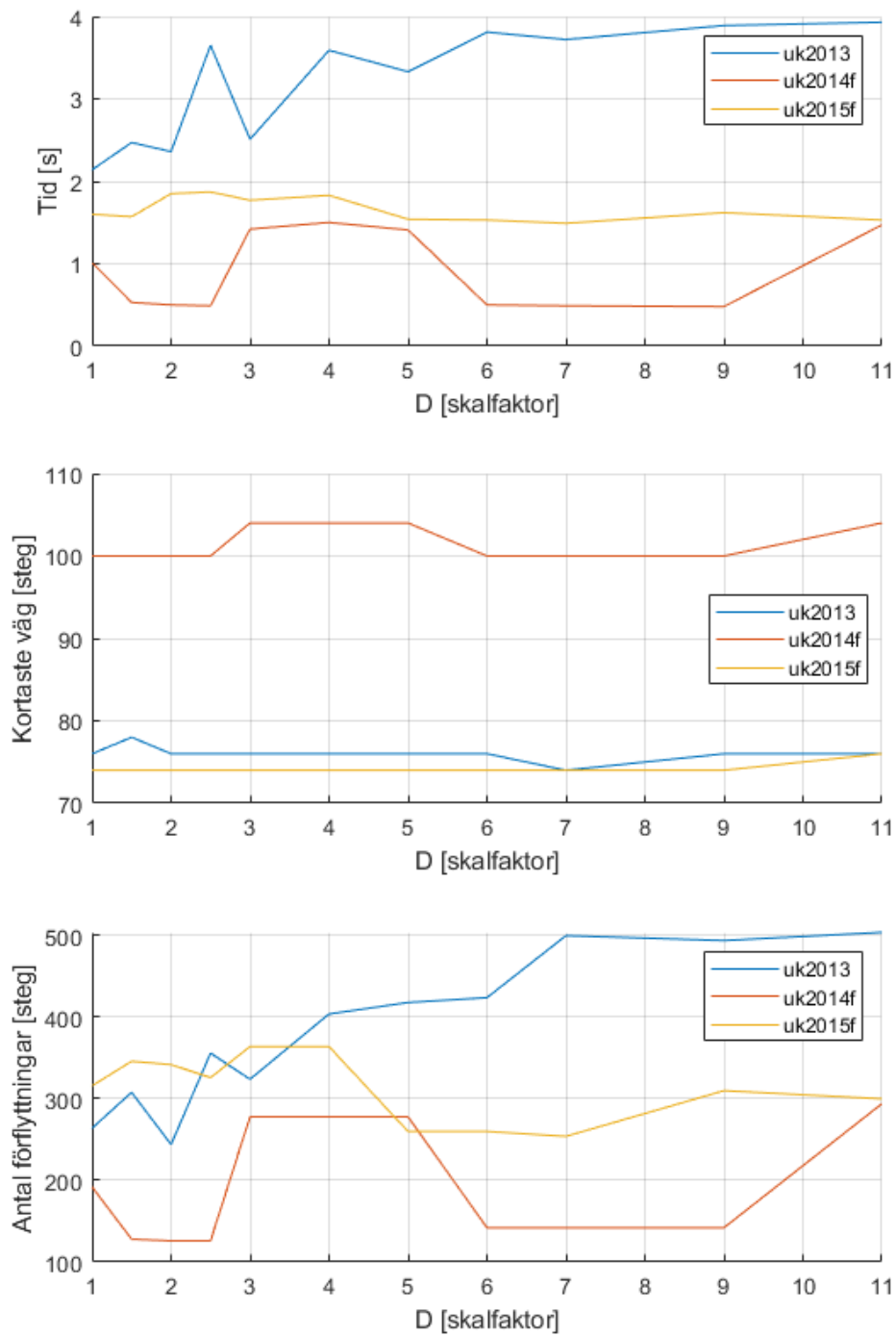
Tabell 5: Jämförelse mellan Manhattan och Euklidisk heuristik. Antalet steg är beräknade som ett genomsnitt av 40 slumpalabyrinter.

	Totalt antal steg
A* Manhattan	347 steg
A* Euklidisk	429 steg

7.2 Programvarans prestanda beroende av viktning av $h(n)$

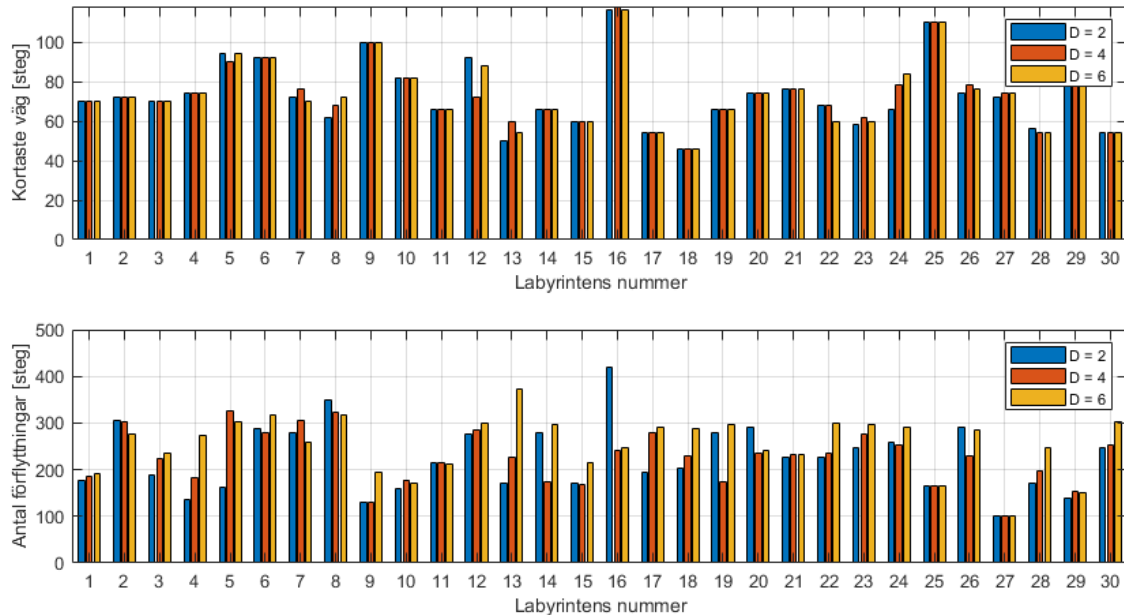
För att se skillnad mellan olika viktningar av $h(n)$ jämfördes ett antal värden på D vid tre olika labyrinter. Av figur 21 kan konstateras att $D = 2$, samt $D = 6$ gett bra resultat.

⁸Benämns i mjukvaran av Xueqiao Xu som Best First Search.



Figur 21: Jämförelse vid olika viktning av $D \cdot h(n)$ på tre olika tävlingslabyrinter. Eget material.

För att fastställa vilket av de testade viktningarna som gav bäst resultat jämfördes tre utvalda värden vid 30 labyrinter. I tabell 6 visas att $D = 2$ ger ett lägre antal förflyttningar vid 15 av labyrinterna.



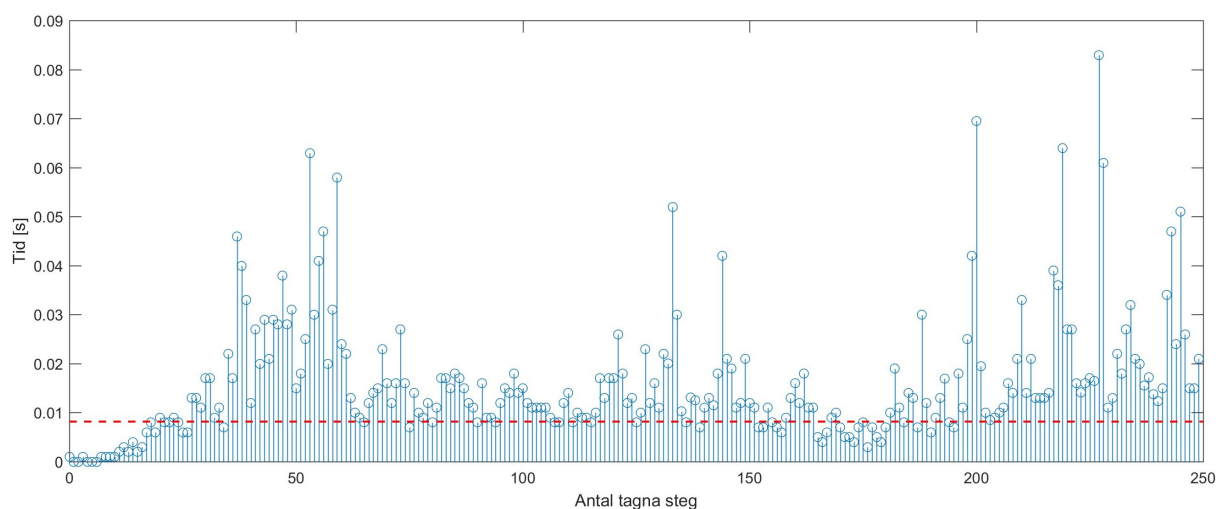
Figur 22: Jämförelse vid utvalda viktningar av $D \cdot h(n)$ på 30 olika tävlingslabyrinter. Eget material.

Tabell 6: Tabell över antalet gånger respektive viktning genererar den korrekta lösningen av labyrinten och den väg med minst antal förflyttningar vid simulering jämfört med de andra. Dessa separeras då den väg som hittas behöver inte vara den bästa lösningen av labyrinten.

	$D = 2$	$D = 4$	$D = 6$	Lika
Korrekt lösning	6	2	2	20
Minst antal förflyttningar	15	8	5	2

7.3 Sökalgorithmens körtid

Vid varje anrop av algoritmen som avgör vilken cell som ska besökas näst genomsöks hela det, i labyrinten, utforskade området och ju längre roboten kört desto större är området som genomsöks. I figur 23 visas tiden det tar att köra algoritmen beroende på hur många steg roboten har tagit i labyrinten. Algoritmen körs med $D = 2$ enligt avsnitt 7.2. Körtiden är i snitt strax under 10 ms, och därför bedöms det inte vara en begränsande faktor i robotens prestanda.



Figur 23: Variation av algoritmens körtid beroende på antalet tagna steg i labyrinten på en av de labyrinter som använts i tävlingen. Den röda streckade linjen representerar körtidens medelvärde. Eget material.

8 Metod för test av robot

För att undersöka robotens prestation jämfört med vad som är programmerat och simulerats utfördes en rad fysiska tester på den slutgiltiga produkten. I detta avsnitt beskrivs metoden för dessa tester.

För att undersöka om kraven i avsnitt 1.2 och 1.3.1 är uppfyllda har roboten mätt och vägts samt genom tidtagning har hastighet och acceleration testas.

Precisionen av roboten har testas på två olika sätt. För test av avståndsmätning och rotation erhöll roboten en instruktion och skillnaden mellan slutpunkten och det instruerade värdet mättes. Avståndet som körts jämfördes även med det avstånd som roboten uppfattat att den har transporterats. För att mäta hur väl roboten känner av avstånd var instruktionen en viss sträcka och vid rotation var det att rotera ett visst antal grader.

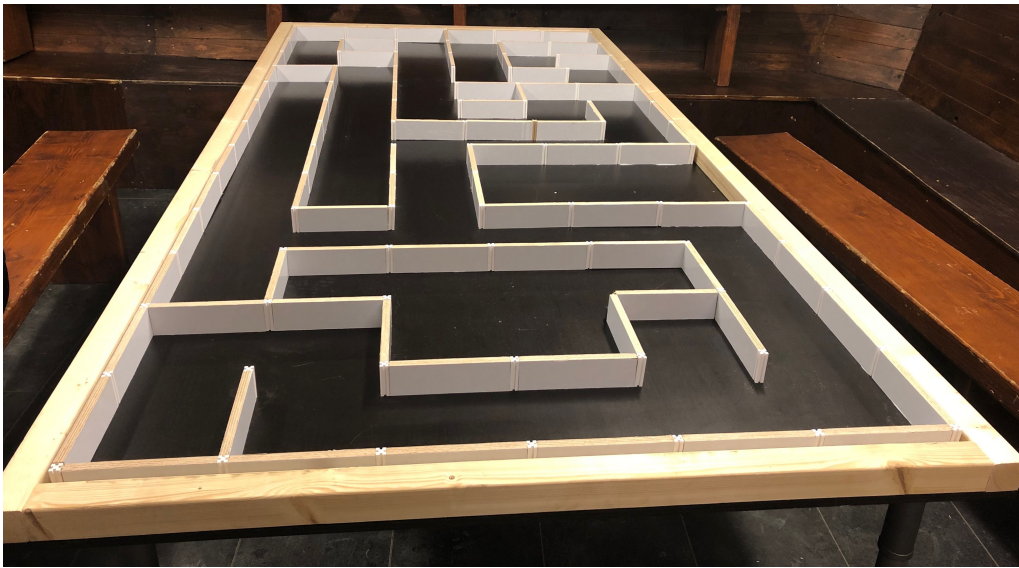
För att verifiera kvalitén på avståndsmätningen i sensorerna placerades roboten i en cell med fyra väggar på ett känt avstånd och ett antal mätpunkter från sensorerna samlades in. Genom att samla flera mätvärden utan att roboten flyttas visade på variansen i avståndsmätningen i sensorerna. Mätdata jämfördes även med kalibreringskurvan för sensorerna.

Profilgeneratoren för hastighet, acceleration och retardation mättes genom att sätta en önskad hastighet och jämföra de värden som roboten genererar vid körningen med de simulerade. Det görs även en mätning genom att mäta tiden vid körning av en känd sträcka med en viss inställd hastighet för att verifiera hastigheten som körs.

I avsnitt 1.3.1 presenteras krav och önskemål för batteritiden. Ifall dessa är uppnådda testades batteritiden på den färdiga roboten genom att låta roboten köra kontinuerligt tills batterierna var urladdade och mäta tiden under denna period.

8.1 Konstruktion av testmiljö

En testmiljö för den slutgiltiga produkten har producerats. Den utformas enligt regler och anvisningar för labryrterna som används i tävlingen, se A samt 1.2, men valdes att göras i en avskalad version. Labrynten är 1137 mm × 2340 mm stor och kan ses i figur 24. Den är modifierbar med lösttagbara väggar så att olika banor kan testas. För detaljerade ritningar och beskrivning, se bilaga I och A.



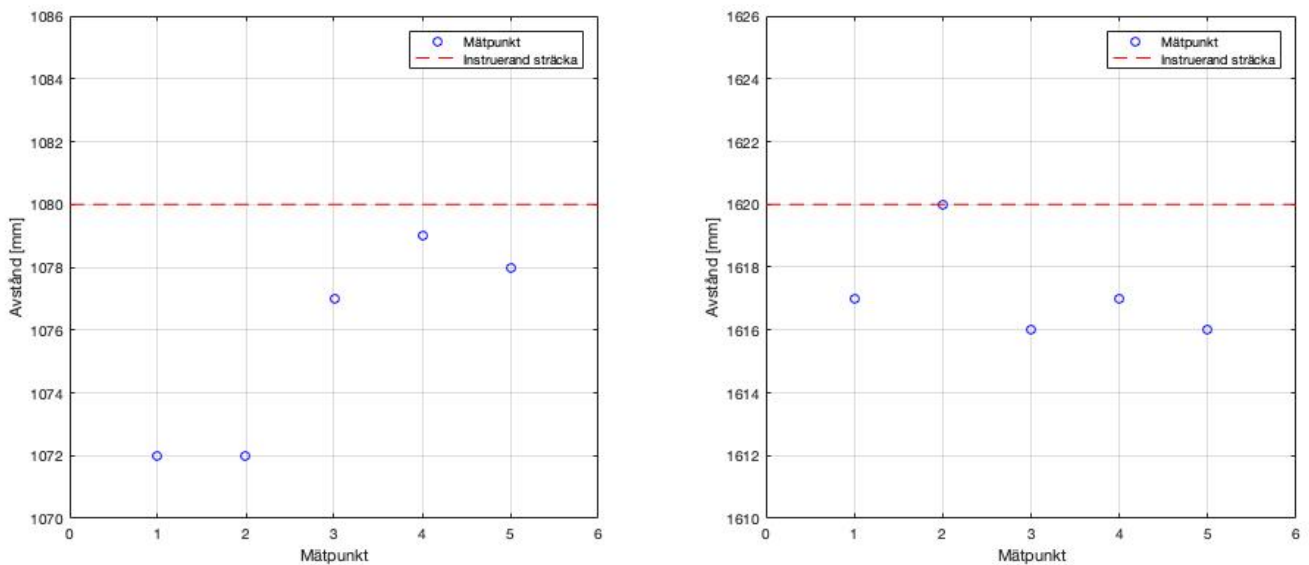
Figur 24: Bild tagen på färdigbyggd labrynt som utgör testmiljön för den färdiga roboten. Eget material.

9 Resultat från tester av robot

Detta avsnitt presenterar resultat av de tester beskrivna i avsnitt 8. Dessa testar olika egenskaper och delar av roboten för att jämföras med simulerade eller beräknade värden. Avsnittet avslutas med en redogörelse av observationer då den färdiga roboten kartlade och körde en fysisk labyrint med hjälp av den utvecklade programvaran.

9.1 Positionsreglering

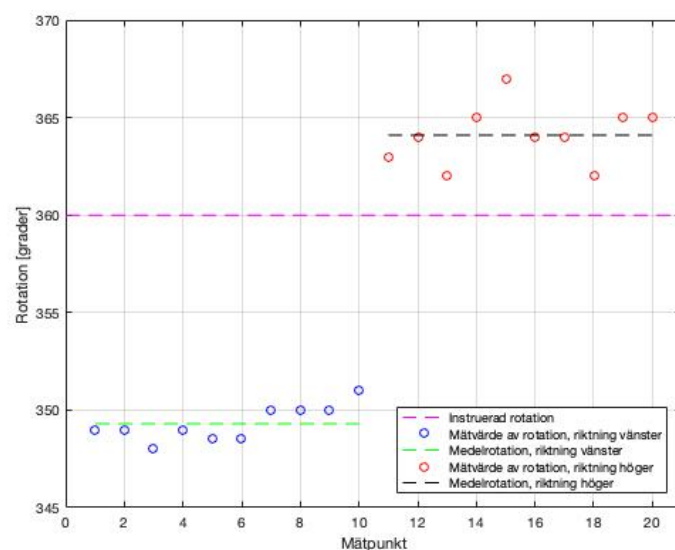
Mätningen av positionsregleringen utfördes genom att två olika avstånd mättes. Roboten instruerades att köra två olika avstånd, 1080 mm och 1620 mm. Sträckan som roboten körde vid 10 olika försök per inställt avstånd mättes och resultatet visas i figur 25.



Figur 25: Två grafer som visar uppmätta avstånd i förhållande till ett inställt värde. Graferna visar resultatet då det inställda värdet är 1080 mm, till vänster och 1620 mm, till höger. Eget material.

9.2 Precision vid rotation

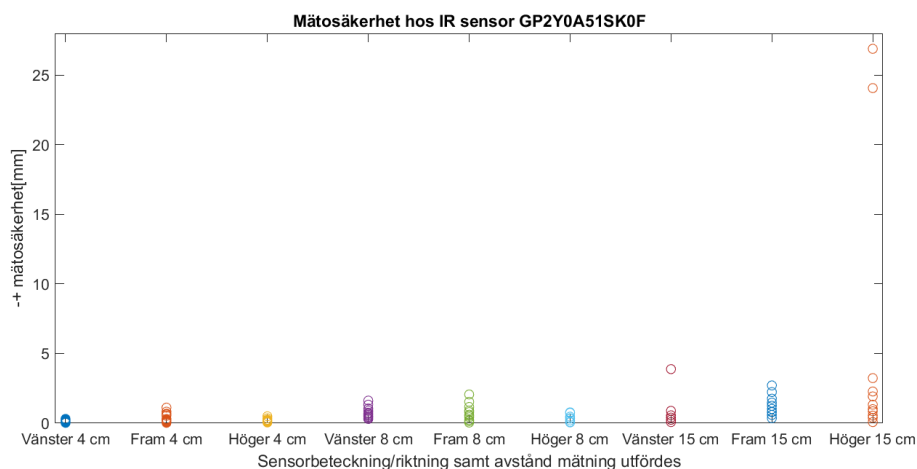
Ett test av när roboten roterar har gjorts. Vid 360° har snittfelet varit 4.1° vid högersväng och 11.3° vid vänstersväng. I figur 26 visas resultatet på alla mätvärden samt ett genomsnittsfel. Detta genomsnittsfel kan användas vid optimering av roboten.



Figur 26: En representation av rotationsfelet då roboten instrueras att rotera 360° . Grafen visar de värden som uppmättes, genomsnittsrotationen och hur dessa förhåller sig till det instruerade värdet. Eget material.

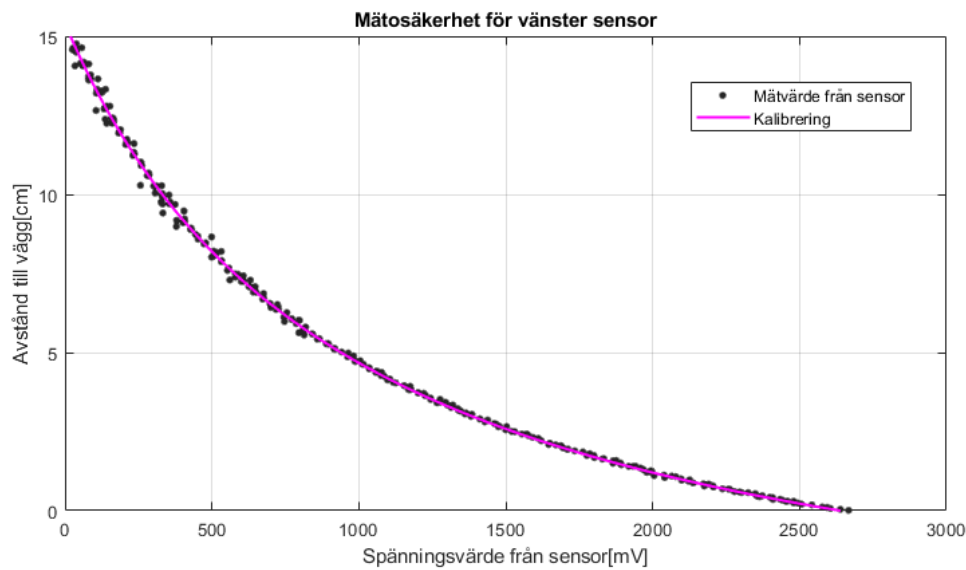
9.3 Sensorernas noggrannhet

Sensorerna visar upp vissa avvikelser i mätdata vid konstant avstånd till närmaste vägg och den största noterade avvikelser är 25 mm. Detta sker vid sensorns maximala mätavstånd, 150 mm, och förutom detta fel, är den maximala avvikelser runt 5 mm. Felen presenteras i figur 27. Vänster sensor var monterad vertikalt och hade en kondensator monterad. Den främre sensorn var horisontellt monterad och hade en kondensator monterad. Den högra sensorn var vertikalt monterad och hade ingen kondensator monterad.



Figur 27: Mätdata som visar de uppmätta avvikelserna. Eget material.

Utöver detta presenteras det uppmätta spänningsvärdet av vänster sensor beroende på avståndet tillsammans med kalibreringskurvan i figur 28.

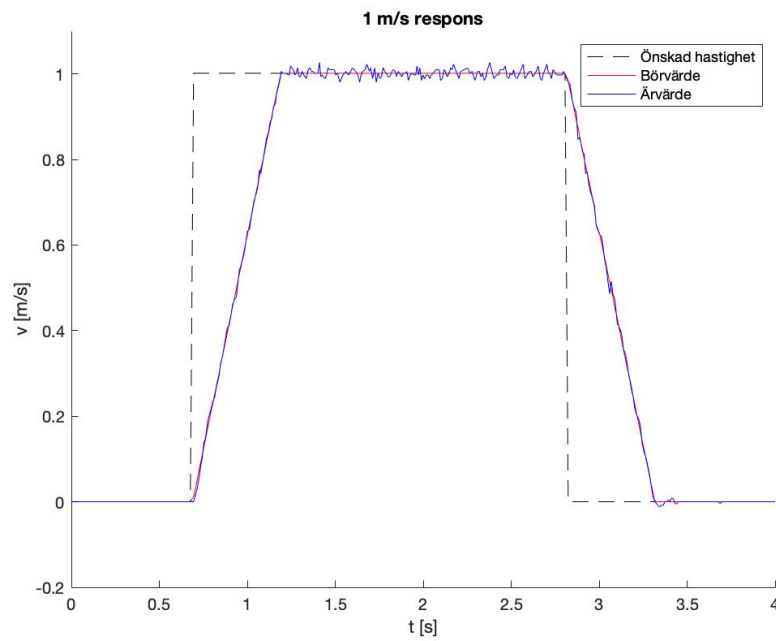


Figur 28: Mätdata ritad mot det förväntade kalibreringsvärdet för den vänstra sensorn. Eget material.

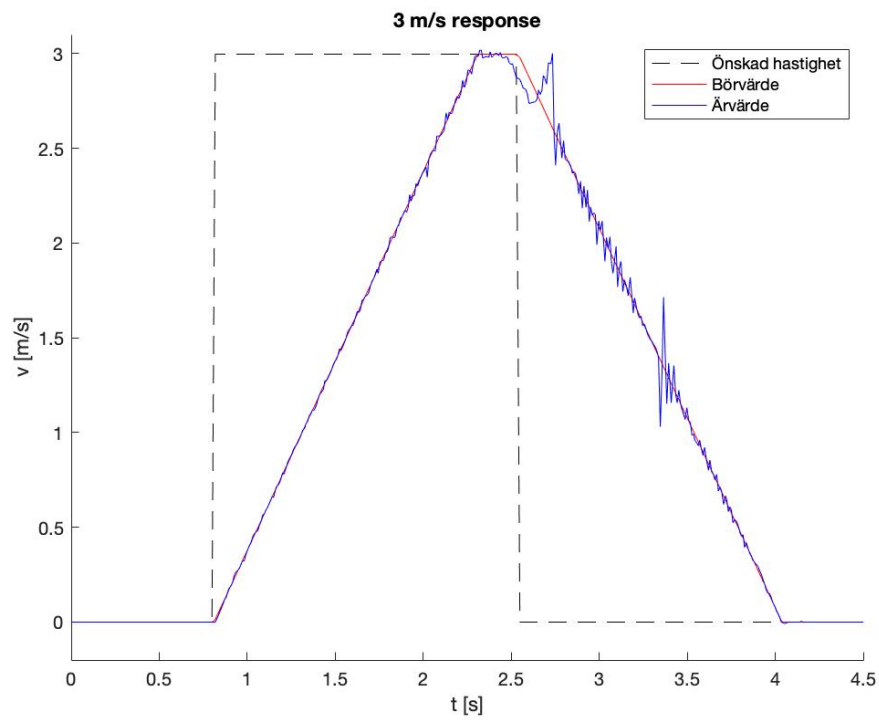
9.4 Hastighet och acceleration

Vid mätning av profilgeneratorn för hastighet sattes en önskad hastighet på 1 m/s respektive 3 m/s och variablerna i RAM-minnet⁹ på mikrokontrollen sparades kontinuerligt, resultatet kan ses i figur 29 och 30. Det ska påpekas att vid retardation från 3 m/s finns kraftiga fluktuationer som troligtvis beror på att datan mättes genom att springa bredvid roboten med datakabeln inkopplad. Resultat från simuleringen av profilgeneratorn i Simulink kan ses i figur 31.

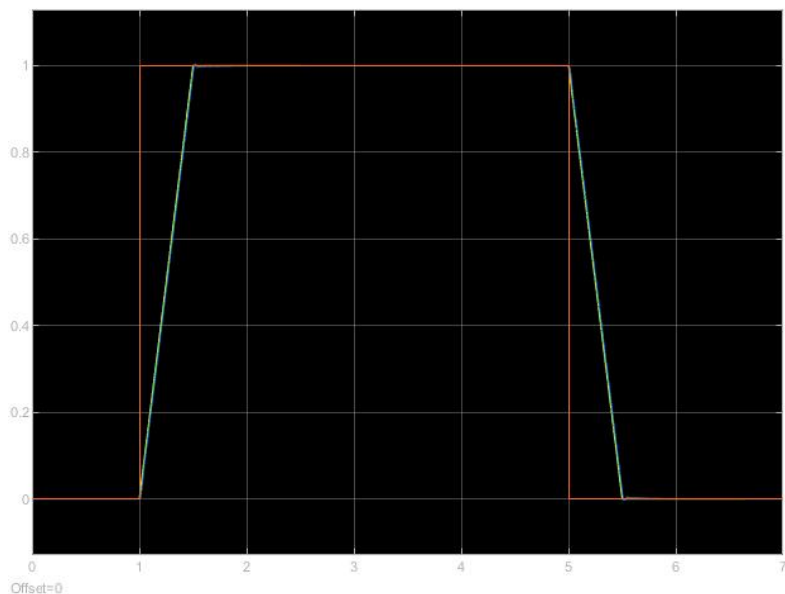
⁹RAM-minne är en typ av primärminne som försvinner vid förlorad strömförsörjning.



Figur 29: Uppmätt data vid acceleration till 1 m/s och sedan ner till 0 m/s. Eget material.



Figur 30: Uppmätt data vid acceleration till 3 m/s och sedan ner till 0 m/s. Eget material.



Figur 31: Stegsvär från simulering av profilgenerator vid 1 m/s. Eget material.

För att utöka resultatet gjordes hastighetsmätningar genom att mäta tiden för roboten att köra en sträcka på 4 m och sedan beräkna hastigheten. Dessa resultat kan ses i tabell 7.

Tabell 7: Uppmätt tid på tre, på varandra, följande mätningar för en sträcka på 4 m.

Önskad hastighet [m/s]	Uppmätt hastighet [m/s]
0.3	0.298
1	0.992
3	3.166

9.5 Batteritid

Vid mätning av batteritid för huvuddatorn användes enbart wifi och programmet för kartläggningen. För mätningen av robotens batteritid kördes en sträcka på 300 mm fram och tillbaka med en acceleration samt retardation på 2 m/s^2 och hastighet på 300 mm/s. Båda mätningarna utgick från fullt laddade batterier på 4.2 V per cell och tomma batterier på 3.2 V per cell. Resultat kan ses nedan i figur 8.

Tabell 8: Uppmätt batteritid och snittförbrukning för huvuddator och robot.

Parameter	Huvuddator	Robot
Tid	63 min	81 min
Snittförbrukning	0.88 W	1.23 W

9.6 Redogörelse av yttre egenskaper

Roboten mättes och vägdes och detta resultat visas i tabell 9. Dess vikt uppfyller kravet i avsnitt 1.3.2, och måtten följer reglerna för tävlingen presenterade i avsnitt 1.2.

Tabell 9: Presentation av värden mätta på färdig robot.

Parameter	Mätvärde
Höjd	81 mm
Längd	93 mm
Bredd	97 mm
Vikt	278 g

9.7 Systemets förmåga att lösa labyrinten

En lösbar labyrint med 13 x 6 celler byggdes upp och roboten placerades i den med uppdrag att kartlägga den. Därefter skulle roboten köra den snabbaste vägen tillbaka till start, för att sedan återigen köra till målet genom att följa den genererade vägen. Detta försök upprepades 15 gånger med samma labyrint, dock återställdes minnet i roboten mellan testerna. Av dessa försök lyckades roboten klara av uppgiften helt och hållet fyra gånger. De försök som ansågs vara misslyckade orsakades av att felaktig information om väggarna i den aktuella cellen skickades från mikrokontrollern till minidatorn. Detta gjorde att kartläggningen blev fel. Om fysiska väggar inte registrerades körde roboten in i väggen, och om väggar registrerades falskt så missades vägar som roboten borde kört. En annan orsak till misslyckade försök var att robotens rörelse var instabil och körde därmed in i väggar. Det noterades även att under kartläggningen genererade sökalgoritmen en instruktion som tog roboten till en cell som ej borde undersökts. Denna cell besöktes vid varje försök.

10 Diskussion

I detta avsnitt diskuteras projektet i sin helhet. Först diskuteras resultatet av testerna på roboten för att sedan behandla hela systemet. Avsnittet avslutas med förslag till vidareutveckling av projektet och förbättringar av systemet.

10.1 Resultat av hårdvarutester

Resultatet från tester av positionsregleringen, som presenteras i avsnitt 9.1, visar att roboten har en tendens att köra ett kortare avstånd än vad som är instruerat. Detta fel är dock mindre än en procent av den förväntade sträckan och anses därmed inte vara av större betydelse vid lösning av labyrinten, men vid körning av längre sträckor än vad som är aktuellt i detta projekt kan det märkas tydligare. Mikrokontrollern räknar ut avstånd baserat på den uppmätta diametern av hjulen och reglerar sedan avståndet med hjälp av antal hjulvarv. Därmed kan ett fel vid mätning av hjulens diameter resultera i stor avvikelse på slutresultatet. Eftersom fälgarna är 3D-printade finns det risk att de inte är exakt lika stora och inte heller exakt den storlek som används vid beräkningen av positionsregleringen. Tester på fler avstånd är önskvärt, då hade eventuellt en trend kunnat utläsas och en procentuell avvikelse kunnat räknas ut för att kalibrera roboten till att beräkna avstånd korrekt.

Då rotationen testades var variansen relativt liten, se resultat i avsnitt 9.2. Detta visar att den implementerade rotationsregleringen är tillräcklig för ändamålet. Eftersom brus i pulsgivaren samt glid mellan däcken och underlaget vid högre accelerationer och hastigheter kan orsaka kvarstående fel hade rotationssensorn som testades hjälpt. Denna hann ej implementeras då hanteringen av sensors data visade sig vara svårare att använda än trott.

Mätosäkerheten som presenterades i avsnitt 9.3 visar även hur sensorerna reagerade på att vara i horisontellt läge samt om de hade kondensator eller ej. Figur 27 visar att på 4 och 8 centimeters avstånd visar den horisontellt monterade främre sensorn en större mätosäkerhet än de andra två. Detta skulle innebära att i enlighet med databladet är det bättre att montera sensorerna i vertikalt läge [46]. Vid avståndet 15 cm syns stora avvikelser i den uppmätta datan för höger och vänster sensor. Detta beror på att ju lägre spänningen blir, desto känsligare blir sensorn för osäkerheter, vilket kan ses i figur 28. När avståndet är på 15 cm, vilket är specificerat som maxavståndet av tillverkaren, kan större fel uppräksammans. Anledningen till att bara höger och vänster visar stora variationer är för att vid gränsvärdet, 15 cm, beter sig sensorerna väldigt olika. Robotens prestanda påverkades inte av detta då det viktigaste området för att hantera alla robotens funktioner är i intervallet 2 cm till 10 cm.

Den initiala tanken var att använda lasersensorer för att registrera var i labyrinten det fanns väggar. Att implementera lasersensorer visade sig vara mer komplicerat

än väntat och därför beslutades att byta ut dem mot IR-sensorer. Bytet medförde att komponenterna inte fick plats på ett plan på roboten och därför togs beslutet att bygga roboten i två våningar. Robotens höjd ger en instabilitet och gör att accelerationsbegränsningar krävdes för att undvika stegring. På grund av detta uppnås ej den i avsnitt 1.3.1 önskade accelerationen, trots att motorerna har kapacitet för det.

Mätningarna på profilgeneratoren i avsnitt 9.4 för hastighet- och accelerationsregleringen visar att konstant acceleration erhålls och den visar även att reglersystemet har mycket litet kvarstående fel. Detta tyder på att simuleringarna i figur 31 stämde bra överens med verkligheten. Den tydligaste skillnaden är bruset i mätsignalen som fås vid mätningen på roboten då detta inte togs hänsyn till i simuleringen. När önskad hastighet är uppnådd uppstår små snabba oscillationer kring börvärdet vilket kan bero på ett för högt K_p/K_d värde eller brus på pulsgivarna. Om oscillationernas uppkomst beror på brus är det relaterat till rotationshastigheten, då inget märkbart brus finns när hastigheten är noll. Mätningarna på hastigheten som var oberoende av hårdvaran visade att den verkliga hastigheten var mycket nära den uppmätta av mikrokontrollen. Vid låga hastigheter skiljer sig den önskade och den uppmätta mycket lite men vid högre skiljer den sig desto mer. Detta beror troligen på att mätningarna gjordes med tidtagarur och genom att titta på roboten. Det leder till problem då hastigheterna blir relativt höga och det är svårt att starta och stoppa tidtagaruret vid rätt tidpunkt. Mätdata från mikrokontrollen stödjer denna teori eftersom graferna visar att börvärdet når ärvärdet vid både 1 m/s och 3 m/s. Eftersom de klockade hastigheterna stämmer väl överens med de önskade vid låga hastigheter bör de även stämma vid högre hastigheter. Då relativt låga accelerationer används i dessa simuleringar och mätningar jobbar motorer och reglersystemet långt under sin potential. För att undersöka hårdvarans beteende vid högre hastigheter krävs en mer stabil konstruktion.

Batteritidens resultat i avsnitt 9.5 skiljer mycket från det beräknade för roboten. Beräkningarna på den teoretiska batteritiden för roboten var gjorda baserade på den initiala positionsreglerings simulering där begränsningar ej fanns på accelerationen. Detta gjorde att strömmarna i motorerna kunde bli upp till och med 1.5 A vilket ledde till den höga effektförbrukningen. I verkligheten lyckades roboten aldrig nå upp till de simulerade accelerationerna eftersom roboten stegrade och accelerationen som användes var avsevärt mycket längre. I simuleringen för profilgeneratoren med accelerationer på 2 m/s² som gjordes i slutet av arbetet visar att strömmar maximalt blir 0.2 A vilket ger en mycket annorlunda energiåtgång. Batteritiden för huvuddatorn stämmer väl överens med den beräknade vilket tyder på att den uppskattade energiåtgången var nära den verkliga.

10.2 Systemets förmåga att lösa labyrinten

De varierande resultaten vid lösning av labyrinten, som presenteras i avsnitt 9.7, visar tydligt att det finns förbättringspotential hos både hårdvara och mjukvara.

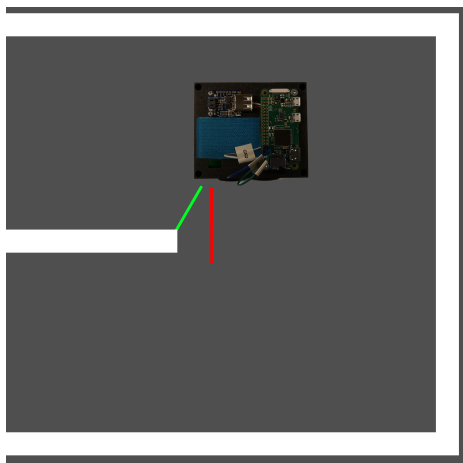
Det vanligast förekommande felet, sändning av felaktig information om väggarna till huvuddatorn, är ett allvarligt problem. Det främsta bekymret med detta är att det kan resultera i att roboten kör in i väggen, samtidigt som programvaran på huvuddatorn tror att roboten förflyttat sig. Eftersom cellen inte går att nå från robotens position kommer programmets modell av verkligheten inte längre vara korrekt. Om kartläggningen blir felaktig i det avseendet att en vägg registreras falskt blir konsekvenserna inte lika illa eftersom cellen kartläggs på nytt nästa gång roboten når den. Kartläggningen riskerar alltså att ta längre tid, men det är fortfarande möjligt för roboten att klara av uppgiften.

Det har inte kunnat entydigt bestämmas varför detta fel uppstår då avståndssensorernas mätosäkerhet är förhållandevis liten, se avsnitt 9.3. Det upplevs att felet uppstår när roboten kör in snett i en cell, men det har inte kunnat säkerställas att det är det enda scenariot då felet uppstår. När roboten inte är centrerad i cellen ökar risken att sensorerna registrerar väggar felaktigt. Detta beror på att bedömningen om huruvida en vägg existerar eller inte görs baserat på förutsättningen att roboten alltid är centrerad i en cell och endast någon centimeters avvikelse tillåts. Om större avvikelse skulle tillåtas ökar risken för felaktigt detekterade väggar i andra situationer. Genom att stabilisera robotens rörelse bör problemet uppstå mindre frekvent. En förbättrad reglering av motorerna och styrsystemet hade ökat robotens sannolikhet att klara av uppgiften avsevärt.

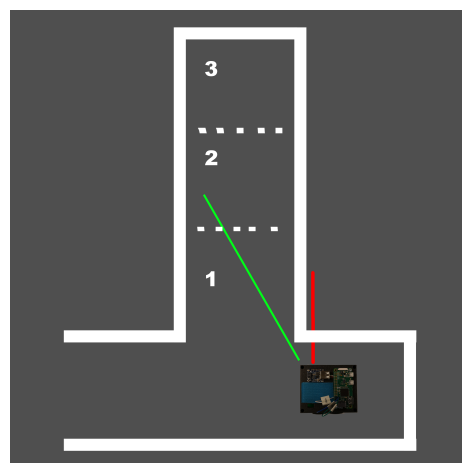
I dessa försök instruerade sökalgoritmen roboten varje gång att förflytta sig i en riktning som den inte borde rört sig i. Detta oväntade beteende var i jämförelse med övriga problem små eftersom det endast var två celler i fel riktning. Detta kan dock förändras avsevärt om en labyrinth med annan utformning används, då mycket tid kan förloras på onödiga förflyttningar. Beteendet har inte uppmärksamats i simuleringar gjorda på labyrinter som använts i tävlingen och det skulle krävas ytterligare simuleringar och tester för att förstå vad som orsakar det oväntade beslutet.

10.3 Möjlig vidareutveckling

Det finns en stor förbättringspotential gällande användningen av sensorer. De IR-sensorer som används reglerar avståndet till väggar, samt avgör om det finns väggar i cellen som roboten befinner sig i. Det vore önskvärt att sensorerna även kunde läsa av celler som roboten ej befinner sig i, till exempel cellen direkt framför roboten. Genom att vinkla IR-sensorerna placerade på sidan 30° framåt kan roboten detektera väggarna strax framför, vilket förbättrar positionsregleringen som illustreras i figur 32a. För att detektera väggar längre fram kan lasersensorer implementeras då de har ett större sökspann. Om lasersensorerna dessutom vinklas 30° framåt kan de kartlägga celler innan roboten nått dem, vilket illustreras i figur 32b. Om det hade gått att undersöka rutor längre fram hade algoritmen haft möjlighet att undersöka ett större område utan att roboten fysiskt hade behövt stå i den rutan. En utveckling åt detta håll anses vara en stor förbättring av robotens prestation men det hade också krävt en mer avancerad programvara för att kunna hantera kartläggningen.



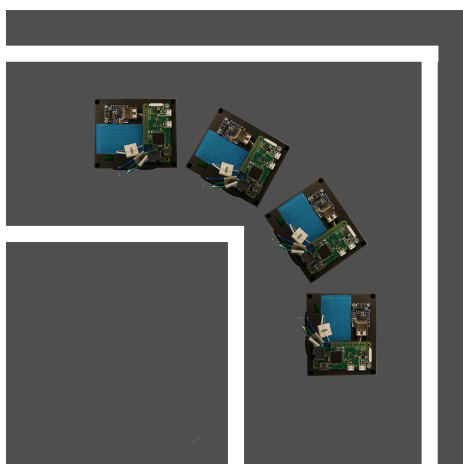
(a) När roboten är på väg in i en cell efter en sväng så kan en 30° vinkling av sensorerna framåt motverka att roboten inte detekterar väggen. Röd linje visar nuvarande konfiguration och grön linje visar en 30° vinkling av sensorerna. Eget material.



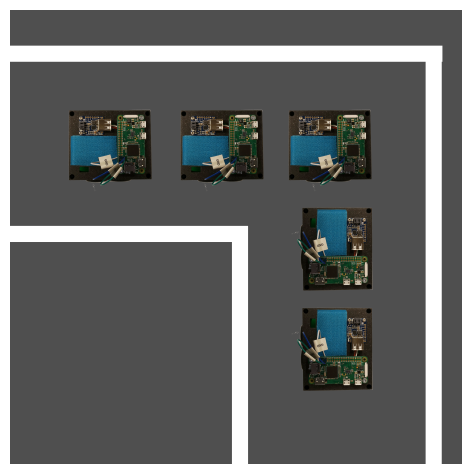
(b) Figuren illustrerar fördelar med en lasersensor som är vinklad 30° . Vid denna konfiguration skulle roboten teoretiskt sett kunna kartlägga två väggar åt sidan och därmed ge en mer komplett bild av labyrinten utan att besöka alla celler. Eget material.

Figur 32

I nuläget har roboten inte anpassats för kontinuerlig kurvtagning, utan stannar för att sedan rotera 90° . Kontinuerlig kurvtagning är att roboten saktar in utan att stanna innan den svänger samt positionerar sig för att svänga under så lång sträcka som möjligt för att kunna ha högre hastighet genom kurvan. Figur 33b visar hur roboten svänger just nu, figur 33a visar den förbättringen som förespråkas. Med kontinuerlig kurvtagning hade robotens hastighet i både kartläggningsfasen och spurtfasen kunnat vara högre.



(a) Effektiv och kontinuerlig kurvtagning. Eget material.

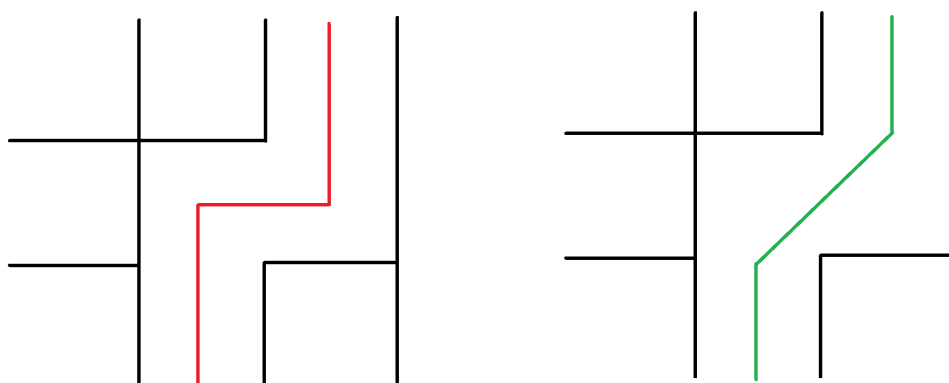


(b) Kurvtagning som det sker i nuläget. Eget material.

Figur 33

De professionella tävlingsrobotar som har studerats har alla designats så att alla delar monterats på enbart ett kretskort vilket optimerar robotens storlek. Det 3D-printade chassit som konstruerats var ett tidseffektivt sätt att utforma roboten, men det resulterade i en större konstruktion än nödvändigt. En annan anledning till att professionella robotar är mindre än den som konstruerats är att de ofta endast styrs av en mikrokontroller snarare än av både en mikrokontroller och en minidator, vilket leder till färre komponenter på roboten. En stor robot har mindre marginaler till väggarna, och därmed behöver regleringen ha högre precision. Det är därför en fördel att ha en liten robot då den kan köra i högre hastighet och med mindre precision utan att kollidera med väggarna.

Roboten accelererar mellan svängar när den kör spurtfasen. Detta gör att roboten kommer köra med högre hastighet om det är flera celler som roboten kör rakt igenom för att sedan svänga än om det är flera svängar emellan kortare raksträckor. I nuläget vägs körriktningen enbart in då $f(n)$ och $h(n)$ är lika hos två möjliga destinationer, och då prioriteras den där roboten fortsätter rakt fram. Körriktningen skulle kunna påverka mer om ytterligare en term adderas till $f(n)$, så att $f(n) = g(n) + h(n) + r(n)$ där $r(n)$ är ett värde som baseras på robotens rörelseriktning. Genom att implementera detta hade det varit enkelt att justera hur mycket roboten ska prioritera raka vägar beroende på hur mycket tid som tappas vid svängar. En annan förbättring hade varit om roboten istället för att ta flera skarpa svängar i följd lagt samman dessa rörelser till en diagonal, därmed inte förlorat lika mycket tid på svängarna. I figur 34 illustreras hur denna typ av rörelse skulle kunna se ut.



Figur 34: Till vänster visas hur roboten idag hade hanterat denna situation. Om svängarna sammanfogats till en rörelse skulle roboten kunna röra sig diagonalt genom cellerna, likt den högra delen av figuren. Eget material.

Som systemet är konstruerat i nuläget slutar algoritmen leta efter kortare vägar när roboten nått mål. Av informationen programvaran har bestäms den kortaste vägen mellan start och mål. En mer sofistikerad programvara kan väga sannolikheten att hitta en snabbare väg genom fortsatt kartläggning mot tiden det tar att göra detta. Genom att fortsätta söka efter alternativa vägar medan roboten rör sig tillbaka mot start och därmed ha möjligheten att hitta en snabbare väg kommer

kartläggningsfasen ta längre tid men spurtfasen kan bli kortare och ge ett bättre slutresultat. Det bedöms vara relativt enkelt att implementera fortsatt kartläggning av labyrinten när roboten rör sig från mål till start. Däremot anses det vara svårare att implementera ett sätt för programmet att kunna bedöma sannolikheten för att hitta en bättre väg genom fortsatt kartläggning samt om den totala tiden förbättras av detta. Ett möjligt sätt att implementera det är att programmet klockar tiden som roboten spenderat i labyrinten samt uppskattar tiden det tar för roboten att köra en specifik sträcka. Genom noggranna tester av tiden det tar för roboten att röra sig i olika riktningar är det möjligt för programmet att uppskatta tiden en viss väg skulle ta. Med denna information skulle programmet kunna avgöra om den för tillfället kortaste vägen väntas ta lång tid att köra i förhållande till den spenderade tiden på kartläggning. För att implementera denna princip på ett enklare sätt kan antalet steg användas istället för tid.

Vid varje steg som roboten tar görs en ny beräkning av sökalgoritmen även om resultatet förväntas bli detsamma nästa gång. Det medför att när roboten kört in i en återvändsgränd och sedan vänder för att ta sig ut görs nya beräkningar för varje steg även på vägen ut. Det finns ingen anledning att stanna och genomföra en ny beräkning då det inte finns någon ny information att ta i beaktning. För att undkomma detta problem bör programmet ge en följd av instruktioner som tar roboten till en cell där den kan köra en alternativ väg, snarare än att ge en instruktion i taget. Det anses inte vara problematiskt att göra ett tillägg i programmet som ger roboten en serie instruktioner när den har flera tidigare besökta celler framför sig, som fallet när den kört in i en återvändsgränd. För att implementera detta krävs små justeringar av programmen på huvuddatorn och mikrokontrollern som hanterar kommunikationen sinsemellan, men det anses vara enkelt att genomföra. I programmet som innehåller sökalgoritmen krävs större förändringar, men även detta bedöms vara genomförbart. Det tillvägagångssätt som förväntas ge bäst resultat är att låta programmet ge roboten en serie instruktioner om hur den tar sig till den obesökta cell som är troligast att leda till målet, istället för att ge instruktionerna separat.

Kommunikationen mellan huvuddatorn och mikrokontrollern gjordes med hjälp av UART som är ett simpelt kommunikationsprotokoll. Utöver detta hade protokoll som säkerställer kommunikationskvalitén samt integrerar mjukvaran på det två enheterna bättre kunnat implementerats.

11 Slutsats

Den autonoma roboten som konstruerats uppfyller sitt syfte genom att den kan hitta målet i en labyrinth och följer reglerna för Micromouse-tävlingen, men det hade vart önskvärt att den kunde förflytta sig genom labyrinthen med en högre repeterbarhet. Trots den låga repeterbarheten vid lösning av labyrinthen anses den konstruerade roboten och tillhörande mjukvara vara en bra grund att utgå från vid vidareutveckling.

Även om roboten nådde förbestämda önskemål av hastighet på raksträckor klarade roboten aldrig att uppnå hastighetskravet vid körning i labyrinth vilket tyder på dimensioneringen av komponenter var väl utförd, men inte konstruktionen och mjukvaran. Accelerationskravet uppfylldes aldrig av den slutgiltiga roboten eftersom den stegrar vid högre accelerationer, men potentialen för att uppnå både krav och önskemål finns på tack vare de väl dimensionerade motorerna. Batteritiden överträffade önskemålen tack vare det reducerade arbetsområdet för robotens motorer.

Det finns förbättringar att utforska vidare som skulle ge ytterligare djup i studien. Reglersystemet hade kunnat dra nytta av en annan konfiguration av sensorer, både för avstånd och rotation för att kunna söka snabbare och kartlägga celler framför roboten. Mjukvaran för kommunikationen har också stor förbättringspotential när det kommer till programflöde och kommunikationsprotokoll. Ett mer avancerat program för kartläggning skulle möjliggöra bättre resultat vid genomförande av både kartläggningsfas och spurtfas.

Referenser

- [1] G. Oriolo, G. Ulivi, och M. Vendittelli, “Real-time map building and navigation for autonomous robots in unknown environments,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 28, s. 316–333, 1998. [Online]. Tillgänglig: <https://pdfs.semanticscholar.org/a71f/6637cc28af86e2ef53b86d190734f62446f0.pdf> Doi:10.1109/3477.678626, Hämtad: 2019-03-28.
- [2] N. Hawes *med flera*, “The strands project: Long-term autonomy in everyday environments,” *IEEE Robotics & Automation Magazine*, vol. 24, s. 146–156, 2017. [Online]. Tillgänglig: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7948740> Hämtad: 2019-03-29.
- [3] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, och K. Konolige, “The office marathon: Robust navigation in an indoor office environment,” i *2010 IEEE international conference on robotics and automation*. Anchorage, USA: IEEE, 2010, s. 300–307, (Hämtad: 2019-03-28). [Online]. Tillgänglig: [http://library.isr.ist.utl.pt/docs/roswiki/attachments/Papers\(2f\)ICRA2010_Marder\(2d\)Eppstein/icra2010_marder-eppstein.pdf](http://library.isr.ist.utl.pt/docs/roswiki/attachments/Papers(2f)ICRA2010_Marder(2d)Eppstein/icra2010_marder-eppstein.pdf)
- [4] B. Madhevan, R. Sakkaravarthi, G. M. Singh., R. Diya, och D. Jha, “Modelling, simulation and mechatronics design of a wireless automatic fire fighting surveillance robot,” *Defence Science Journal*, vol. 67, s. 572–580, Sep. 2017. [Online]. Tillgänglig: <https://pdfs.semanticscholar.org/e94c/90af6ab51eaa076247e3c171be88a6bc9528.pdf>
- [5] C. Shannon, “Claude shannon demonstrates machine learning,” Youtube, 2016. [Online]. Tillgänglig: <https://www.youtube.com/watch?v=vPKkXibQXGA> Hämtad: 2019-05-06.
- [6] “Bell labs advances intelligent networks,” Alcatel-Lucent, Paris, Frankrike. [Online]. Tillgänglig: https://web.archive.org/web/20120722011711/http://www.alcatel-lucent.com/wps/portal/%21ut/p/kcxml/04_Sj9SPykssy0xPLMnMz0vM0Y_QjzKLd4w39w3RL8h2VAQAG0JBYA%21%21?LMSG_CABINET=Bell_Labs&LMSG_CONTENT_FILE=News_Features%2FNews_Feature_Detail_000025 Hämtad: 2019-05-07.
- [7] Robogames, “Maze solving / micromouse rules,” 2019. [Online]. Tillgänglig: <http://robogames.net/rules/maze.php> (Hämtad: 2019-02-02).
- [8] School of Engineering and the Built Environment Birmingham City University, “Techfest competitions (including uk micromouse).” [Online]. Tillgänglig: <https://www.bcu.ac.uk/engineering/news-events/micromouse/competition> (Hämtad: 2019-02-08).
- [9] G. Ye, “Micromouses,” 2018. [Online]. Tillgänglig: <http://greenye.net/Pages/Micromouse/MicromouseIndex.htm> (Hämtad: 2019-03-31).

- [10] Wikipedia, “C programming language,” 2019. [Online]. Tillgänglig: [https://en.wikipedia.org/wiki/C_\(programming_language\)](https://en.wikipedia.org/wiki/C_(programming_language)) (Hämtad: 2019-05-14).
- [11] P. S. Foundation, “Python, version 3.6.1,” mar 2017. [Online]. Tillgänglig: <https://www.python.org/>
- [12] R. P. Foundation, “Raspberry pi zero.” [Online]. Tillgänglig: <https://www.raspberrypi.org/products/raspberry-pi-zero/>
- [13] Sagsaw, “Quadrature diagram,” 2007. [Online]. Tillgänglig: https://commons.wikimedia.org/wiki/File:Quadrature_Diagram.svg (Hämtad: 2019-05-13).
- [14] T. Instruments, “Understanding optical time-of-flight (tof) technology,” *Youtube*, 2018. [Video]. [Online]. Tillgänglig: <https://www.youtube.com/watch?v=TpjnooXhOmY> (Hämtad: 2019-05-07).
- [15] B. Lennartson, *Reglerteknikens grunder*. Studentlitteratur, 2000.
- [16] Mathworks, “Matlab,” 2019. [Online]. Tillgänglig: <https://www.mathworks.com/products/matlab.html> (Hämtad: 2019-05-13).
- [17] Mathworks, “Simulink,” 2019. [Online]. Tillgänglig: <https://se.mathworks.com/products/simulink.html> (Hämtad: 2019-05-13).
- [18] Mathworks, “Simscape,” 2019. [Online]. Tillgänglig: <https://se.mathworks.com/products/simscape.html> (Hämtad: 2019-05-13).
- [19] *Series 1717 ... SR*, Faulhaber, 2019. [Online]. Tillgänglig: https://www.faulhaber.com/fileadmin/Import/Media/EN_1717_SR_DFF.pdf
- [20] *TB6612FNG*, Toshiba, 2016. [Online]. Tillgänglig: <https://www.sparkfun.com/datasheets/Robotics/TB6612FNG.pdf>
- [21] “Sparkfun motor driver - dual tb6612fng (1a),” Sparkfun, 2019. [Online]. Tillgänglig: <https://www.sparkfun.com/products/14451> (Hämtad 2019-04-15).
- [22] *Planetary Gearhead*, Faulhaber, 2019. [Online]. Tillgänglig: https://www.faulhaber.com/fileadmin/Import/Media/EN_15A_DFF.pdf
- [23] *IE2-1024*, Faulhaber, 2019. [Online]. Tillgänglig: https://www.faulhaber.com/fileadmin/Import/Media/EN_IE2-1024_DFF.pdf
- [24] Mathworks, “Cftools,” 2019. [Online]. Tillgänglig: <https://se.mathworks.com/help/curvefit/curvefitting-app.html> (Hämtad: 2019-05-14).
- [25] STMicroelectronics, “Power consumption benchmarks.” [Online]. Tillgänglig: <https://www.pidramble.com/wiki/benchmarks/power-consumption> (Hämtad 2019-04-15).
- [26] “Lipo battery 3.7v / 250mah,” Lawicel, 2019. [Online]. Tillgänglig: <https://www.lawicel-shop.se/lipo-battery-3-7v-250mah> (Hämtad 2019-04-15).

- [27] “Powerboost 500 basic - 5v usb boost @ 500ma from 1.8v+,” Adafruit, 2019. [Online]. Tillgänglig: <https://www.adafruit.com/product/1903> (Hämtad 2019-04-15).
- [28] *Turnigy Nano Tech 300mAh 2S 35-70C*, HobbyKing, 2019, (Hämtad 2019-04-12). [Online]. Tillgänglig: https://hobbyking.com/en_us/turnigy-nano-tech-300mah-2s-35-70c-lipo-pack.html?___store=en_us
- [29] *LM340, LM340A and LM7805 Family Wide VIN 1.5-A Fixed Voltage Regulators*, Texas Instruments, 2016. [Online]. Tillgänglig: <http://www.ti.com/lit/ds/symmlink/lm340.pdf>
- [30] “Stm32f103c8,” STMicroelectronics, 2019. [Online]. Tillgänglig: <https://www.st.com/en/microcontrollers-microprocessors/stm32f103c8.html> (Hämtad 2019-04-12).
- [31] R. Pi, “Raspian operating system,” 2019. [Online]. Tillgänglig: <https://www.raspberrypi.org/downloads/raspbian/> (Hämtad: 2019-05-15).
- [32] Autodesk, “Eeagle,” 2019. [Online]. Tillgänglig: <https://www.autodesk.com/products/eagle/overview> (Hämtad: 2019-05-14).
- [33] Autodesk, “Eeagle,” 2019. [Online]. Tillgänglig: <https://www.autodesk.com/products/fusion-360/overview> (Hämtad: 2019-05-14).
- [34] H. Zhang, Y. Wang, Y. Wang, och P. L. Soon, “Design and realization of two-wheel micro-mouse diagonal dashing,” *Journal of Intelligent and Fuzzy Systems*, 2016. [Online]. Tillgänglig: <https://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs169070>
- [35] G. Ye, “Project futura,” 2014. [Online]. Tillgänglig: http://micromouseusa.com/?page_id=1342 (Hämtad: 2019-04-11).
- [36] M. A. Weiss, *Data structures and algorithm analysis in Java, 3:e upplagan*. Pearson Education, 2012.
- [37] R. Sedgewick och K. Wayne, *Algorithms*. Pearson Education, 2011.
- [38] M. Welling, University of Amsterdam, “Informed search algorithms.” [Online]. Tillgänglig: <https://www.ics.uci.edu/~welling/teaching/271fall09/InfSearch271f09.pdf> (Hämtad: 2019-02-20).
- [39] A. Patel, “Heuristics,” 2019. [Online]. Tillgänglig: <http://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html> (Hämtad: 2019-03-30).
- [40] Psychonaut. (2006) Manhattan distance. [Online]. Tillgänglig: <https://commons.wikimedia.org/w/index.php?curid=731390>
- [41] P. Harrison, “Micromouse maze file collection,” jan 2018. [Online]. Tillgänglig: <http://www.micromouseonline.com/2018/01/31/micromouse-maze-file-collection/>

- [42] Circuit Basics. (2016) Basics of uart communication. [Online]. Tillgänglig: <http://www.circuitbasics.com/basics-uart-communication/> Hämtad: 2019-05-01.
- [43] C. Liechti. (2015) pyserial's documentation. [Online]. Tillgänglig: <https://pythonhosted.org/pyserial/>
- [44] X. Xu. [Online]. Tillgänglig: <http://qiao.github.io/PathFinding.js/visual/> Inställningar, standard förutom: Ingen diagonal samt inte gå genom hörn.
- [45] [Online]. Tillgänglig: https://en.wikipedia.org/wiki/Maze_generation_algorithm#Python_code_example
- [46] *IR sensor 2-15 cm*, Sharp, 2019, (hämtad 2019-05-13). [Online]. Tillgänglig: <https://www.pololu.com/file/0J845/GP2Y0A41SK0F.pdf.pdf>
- [47] *IR sensor 10-80 cm*, Sharp, 2019, (hämtad 2019-05-13). [Online]. Tillgänglig: http://www.sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf
- [48] *IR sensor 2-15 cm*, STmicroelectronics, 2019, (hämtad 2019-05-13). [Online]. Tillgänglig: <https://www.st.com/resource/en/datasheet/vl53l0x.pdf>
- [49] O. Iwasaki. [Online]. Tillgänglig: https://www.flickr.com/photos/osamu_iwasaki/3122119069/sizes/o/in/photostream/ (Hämtad: 2019-02-04).

A Regelverk för Micromouse tävling

Återgiven med tillstånd från Birmingham City University [7].

UK Micromouse Maze Solver Rules

1. The Micromouse Maze

1.1 The micromouse maze shall comprise 16 x 16 multiples of an 18cm x 18cm unit square. The walls constituting the micromouse maze shall be 5cm high and 1.2cm thick. Passageways between the walls shall be 16.8cm wide. The outside wall shall enclose the entire micromouse maze.

1.2. The side of the micromouse maze walls shall be white, and the top of the walls shall normally be red. The floor of the micromouse maze shall be made of MDF and finished with a matt variety of black paint (blackboard paint). The coating on the top and side of the wall shall be selected to reflect, and the coating on the floor shall be selected to absorb, visible and infra-red light.

1.3. The start of the micromouse maze shall be located at one of the four corners. The starting square shall have walls on three sides. The starting square orientation shall be such that when the open wall is to the 'north', outside micromouse maze walls are on the 'west', and 'south'. At the centre of the micromouse maze shall be an opening composed of 4 unit squares. This central square shall be the destination.

1.4. Square posts, each 1.2cm x 1.2cm x 5cm high, shall be placed at the four corners of each unit square (the lattice points). The micromouse maze shall be constituted such that there is at least one wall touching each lattice point, except for the destination square.

1.5. The dimensions of the micromouse maze shall be accurate to within 5% or 2cm, whichever is less. Assembly joints on the micromouse maze floor shall not involve steps or gaps of greater than 1mm. The change of slope at an assembly joint shall not be greater than 5 degrees. Gaps between the walls and posts shall not be greater than 1mm.

1.6. A start sensor may be placed at the boundary between the starting unit square and the next unit square. A destination sensor may be placed at the entrance to the destination square. The light beam from each sensor will be horizontal and positioned 1 cm above the floor.

1.7. Multiple paths to the destination square are allowed and are to be expected.

2. The Micromouse

2.1. A micromouse is subject to the following size constraints –maximum width 25cm, maximum length 25cm. There is no height limit. A micromouse must be completely self-contained and must receive no outside assistance.

2.2. The method of wall sensing is at the discretion of the builder; however, the micromouse must not exert a force on any wall likely to cause damage.

2.3. The power source will normally be batteries and electric motors, however alternative power sources will be permitted at the discretion of the judges.

2.4. If the judges consider that a micromouse has a high risk of damaging the maze it will be disqualified from the competition.

2.5 The micromouse must negotiate the micromouse maze; it must not climb or jump over the walls of the micromouse maze.

2.6 Nothing may be deposited in the micromouse maze.

2.7. Each micromouse should be fitted with a suitable hook or loop, for lifting the micromouse out from the centre of the maze, should this prove necessary. Contestants may not be allowed to climb over the micromouse maze.

3. The Competition

3.1. The time taken to travel from the start square to the destination square is called the "run" time. Travelling from the destination square back to the start square is not considered a run. The total time taken from the first activation of the micromouse until the start of each run is also measured. This is called the 'maze' or 'search' time. If the micromouse requires any manual assistance at any time during the contest, it is considered 'touched'. Scoring is based on these three parameters.

3.2. Each micromouse is allowed a maximum of 10 minutes to perform. This may be reduced if time is limited. The judges have the discretion to request a micromouse to retire in the event that it seems unlikely to reach the destination in the allocated time.

3.3. The scoring of a micromouse shall be obtained by computing a handicapped time for each run as follows:

Score Time = Run Time + Search Penalty + Touch Penalty.

Search Penalty = $1/30$ of the maze or search time, in seconds, associated with that run.

Touch Penalty = 3 seconds plus $1/10$ of the run time, in seconds, if the micromouse has been touched at any time prior to the run.

For example, if a micromouse, after being in the micromouse maze for 4 minutes without being touched, starts a run which takes 20 seconds, the run will have a handicapped time score of $20 + 1/30 (4 \times 60) = 28$ seconds. However, if the micromouse has been touched prior to the run, an additional touch penalty of $(3 + (1/10 \times 20))$ seconds is added giving a handicapped time score of 33 seconds.

3.4. When the micromouse reaches the destination square, it may stop and remain at the micromouse maze centre, or it may continue to explore other parts of the micromouse maze, or make its own way back to the start. If the micromouse stops at the centre, it may be lifted out, manually, and restarted by the handler. Manually lifting it out shall be considered touching the micromouse and will cause a touch penalty to be added on all subsequent runs. If the micromouse does not remain in the destination square at the end of a run, it may not be stopped manually and restarted.

3.5. The time for each run (run time) shall be measured from the moment the micromouse leaves the start square until it enters the destination square. The total

time on the micromouse maze (maze or search time) shall be measured from the time the micromouse is first activated.

3.6. The time taken to negotiate the micromouse maze shall be measured either manually by the contest officials, or by sensors set at the start and destination. If sensors are used, the start sensor shall be positioned at the boundary between the start square and the next unit square.

3.7. The starting procedure of the micromouse shall be simple and must not offer a choice of strategies to the handler. For example, a decision to make a fast run to the centre as time runs out must be made by the micromouse itself. The starting procedure shall be submitted to the judges when the micromouse is registered on the day of the contest.

8. The micromouse handler is given 1 minute, from the moment the micromouse is required to start, to make adjustments to the micromouse sensors (calibration). However, no selection of strategies must be made and no information on the micromouse maze configuration may be entered or captured in the memory.

9. The maze or search time clock will commence after the expiry of the 1 minute time limit even if the handler is still making adjustments to the sensors.

10. If a micromouse "gets into trouble" the handlers can ask the judge for permission to abandon the run and restart the micromouse at the start square. A micromouse may not be re-started merely because it has taken a wrong turning. The judges may add a time penalty for a restart. The judges' decision is final.

11. Modification of the control strategy of the micromouse is not permitted at any time once the mouse has entered the maze.

12. If a micromouse elects to retire because of technical problems, the judges may, at their discretion, permit it to perform again later in the contest. The micromouse will then be deemed to have taken an extra three minutes search time (i.e. if a micromouse retires after four minutes, then when re-starting it will be counted as having taken seven minutes and will have only three more minutes to run). This permission is likely to be withdrawn, if time is limited.

13. The judges will use their discretion to award the prizes, which in addition to the major prizes may include prizes for specific classes of micromouse -e.g. lowest cost, most ingenious, best presented, most entertaining etc.

14. Before the micromouse maze is unveiled, the mice must be accepted and kept in view of the contest officials. The handlers will place the mice at the start under the officials' instructions.

15. The judges reserve the right to make changes to any of the above in the interest of fair play and sportsmanship, and to ensure that all competitors have an enjoyable competition. In the event of ambiguity, the judges' interpretation of any clauses of the rules shall prevail.

B Datablad: Motor

Datablad för likströmsmotorer [19].



DC-Micromotors

Precious Metal Commutation

2,2 mNm

5 W

Series 1717 ... SR

Values at 22°C and nominal voltage		1717 T	003 SR	006 SR	012 SR	018 SR	024 SR	
1	Nominal voltage	U_N	3	6	12	18	24	V
2	Terminal resistance	R	1,07	4,3	17,1	50,1	68,8	Ω
3	Efficiency, max.	η_{max}	69	69	70	68	70	%
4	No-load speed	n_0	14 000	14 000	14 000	12 300	14 000	min ⁻¹
5	No-load current, typ. (with shaft ø 1,5 mm)	I_0	0,091	0,046	0,023	0,013	0,011	A
6	Stall torque	M_H	5,37	5,34	5,38	4,66	5,36	mNm
7	Friction torque	M_R	0,18	0,18	0,18	0,18	0,17	mNm
8	Speed constant	k_n	4 820	2 410	1 210	709	602	min ⁻¹ /V
9	Back-EMF constant	k_E	0,207	0,414	0,829	1,41	1,66	mV/min ⁻¹
10	Torque constant	k_M	1,98	3,96	7,92	13,5	15,9	mNm/A
11	Current constant	k_I	0,505	0,253	0,126	0,074	0,063	A/mNm
12	Slope of n-M curve	$\Delta n / \Delta M$	2 610	2 620	2 600	2 640	2 610	min ⁻¹ /mNm
13	Rotor inductance	L	17	65	260	760	1 040	μ H
14	Mechanical time constant	τ_m	16	16	16	16	16	ms
15	Rotor inertia	J	0,59	0,58	0,59	0,58	0,59	gcm ²
16	Angular acceleration	α_{max}	92	92	92	80	92	·10 ³ rad/s ²
17	Thermal resistance	R_{th1} / R_{th2}	4,5 / 27					K/W
18	Thermal time constant	τ_{w1} / τ_{w2}	2 / 210					s
19	Operating temperature range:							
	– motor		-30 ... +85 (optional version -55 ... +125)					°C
	– winding, max. permissible		+125					°C
20	Shaft bearings		sintered bearings		ball bearings, preloaded			
21	Shaft load max.:		(standard)		(optional version)			
	– with shaft diameter		1,5		1,5			mm
	– radial at 3 000 min ⁻¹ (3 mm from bearing)		1,2		5			N
	– axial at 3 000 min ⁻¹		0,2		0,5			N
	– axial at standstill		20		10			N
22	Shaft play:							
	– radial		$\leq$	0,03	0,015			mm
	– axial		$\leq$	0,2	0			mm
23	Housing material		steel, black coated					
24	Mass		18					g
25	Direction of rotation		clockwise, viewed from the front face					
26	Speed up to	n_{max}	16 000					min ⁻¹
27	Number of pole pairs		1					
28	Magnet material		NdFeB					
Rated values for continuous operation								
29	Rated torque	M_N	1,2	2,1	2,1	2,1	2,2	mNm
30	Rated current (thermal limit)	I_N	0,7	0,63	0,32	0,19	0,16	A
31	Rated speed	n_N	10 790	6 540	6 570	4 570	6 540	min ⁻¹

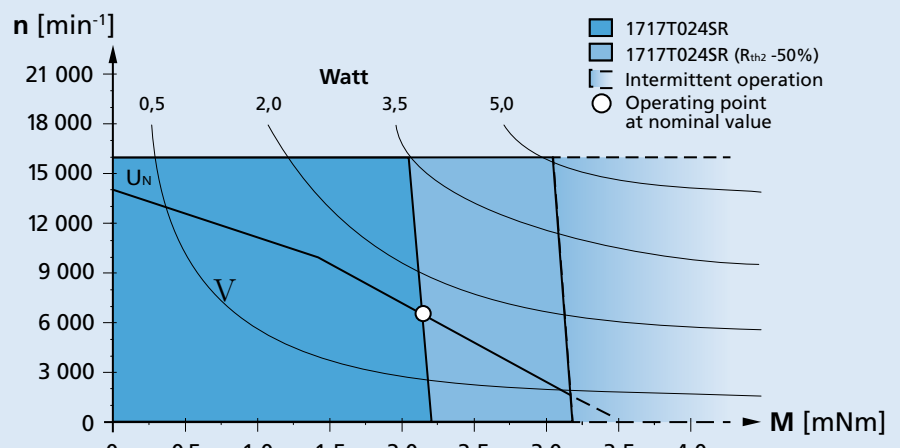
Note: Rated values are calculated with nominal voltage and at a 22°C ambient temperature. The R_{th2} value has been reduced by 0%.

Note:

The diagram indicates the recommended speed in relation to the available torque at the output shaft for a given ambient temperature of 22°C.

The diagram shows the motor in a completely insulated as well as thermally coupled condition (R_{th2} 50% reduced).

The nominal voltage (U_N) curve shows the operating point at nominal voltage in the insulated and thermally coupled condition. Any points of operation above the curve at nominal voltage will require a higher operating voltage. Any points below the nominal voltage curve will



C Datablad: Väckellåda

Datablad för väckellåda [23].



Planetary Gearheads

0,25 Nm

For combination with
DC-Micromotors
Stepper Motors

Series 15A

	15A	15AC	15AK
Housing material	plastic	plastic	plastic
Geartrain material	plastic	plastic	plastic
Recommended max. input speed for:			
– continuous operation	5 000 min ⁻¹	5 000 min ⁻¹	5 000 min ⁻¹
Backlash, at no-load	≤ 4 °	≤ 4 °	≤ 4 °
Bearings on output shaft	sintered bearings	ceramic bearings	ball bearings
Shaft load, max.:			
– radial (5 mm from mounting face)	≤ 3 N	≤ 10 N	≤ 15 N
– axial	≤ 1 N	≤ 2 N	≤ 5 N
Shaft press fit force, max.	≤ 10 N	≤ 10 N	≤ 10 N
Shaft play			
– radial (5 mm from mounting face)	≤ 0,06 mm	≤ 0,08 mm	≤ 0,09 mm
– axial	≤ 0,25 mm	≤ 0,25 mm	≤ 0,25 mm
Operating temperature range	- 30 ... + 65 °C	- 20 ... + 85 °C	- 30 ... + 85 °C

Technical data

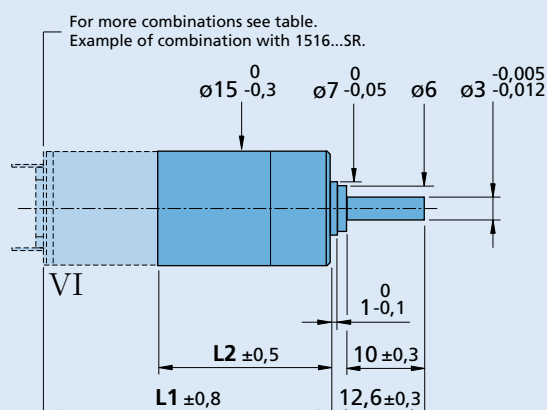
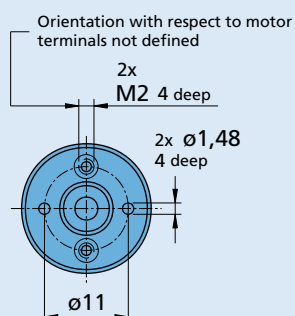
Number of gear stages		1	2	3	3	4	5	5	6
Continuous torque	mNm	50	100	100	150	200	200	250	250
Intermittent torque	mNm	100	200	200	300	400	400	400	400
Mass without motor, ca.	g	4	5	5	5	6	6	7	7
Efficiency, max.	%	87	78	68	67	62	55	52	49
Direction of rotation, drive to output		=	=	=	=	=	=	=	=
Reduction ratio ¹⁾ (rounded)	Code B ²⁾		14:1 19:1	52:1 69:1		249:1	896:1		3 225:1
	Code A ²⁾	5,33:1	28:1	102:1	152:1	369:1 546:1 809:1	1 327:1 1 966:1	2 913:1 4 315:1	4 778:1 7 078:1 10 486:1 15 534:1 23 014:1
L2 [mm] = length without motor ³⁾		14,1	17,7	21,3	21,3	24,9	28,5	28,5	32,1
L1 [mm] = length with motor	1516A/B...SR	29,9	33,5	37,1	37,1	40,7	44,3	44,3	47,9
	1524A/B...SR	37,9	41,5	45,1	45,1	48,7	52,3	52,3	55,9
	1624A/B...S	37,9	41,5	45,1	45,1	48,7	52,3	52,3	55,9
	1717A/B...SR	31,1	34,7	38,3	38,3	41,9	45,5	45,5	49,1
	1724A/B...SR	38,1	41,7	45,3	45,3	48,9	52,5	52,5	56,1
	AM1524...70	30,5	34,1	37,7	37,7	41,3	44,9	44,9	48,5

¹⁾ The reduction ratios are rounded, the exact values are available on request or at www.faulhaber.com.

²⁾ Example of ordering information: 1516 B 012 SR + 15A 19:1, not for AM1524.

³⁾ L2 + 0,7 mm, in combination with 1516A/B...SR and 1524A/B...SR.

Note: These gearheads are available only with motors mounted.



D Datablad: GP2Y0A21YK0F

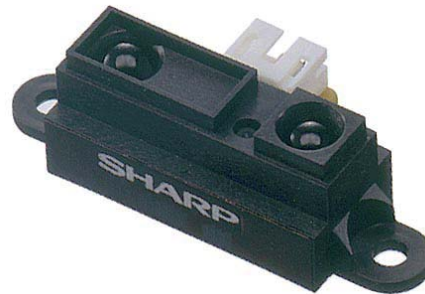
Datablad för IR-sensor 10 till 80 cm avståndsmätning [47].

SHARP

GP2Y0A21YK0F

GP2Y0A21YK0F

Distance Measuring Sensor Unit
Measuring distance: 10 to 80 cm
Analog output type



■Description

GP2Y0A21YK0F is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector), IRED (infrared emitting diode) and signal processing circuit.

The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method.

This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor.

■Features

1. Distance measuring range : 10 to 80 cm
2. Analog output type
3. Package size : 29.5×13×13.5 mm
4. Consumption current : Typ. 30 mA
5. Supply voltage : 4.5 to 5.5 V

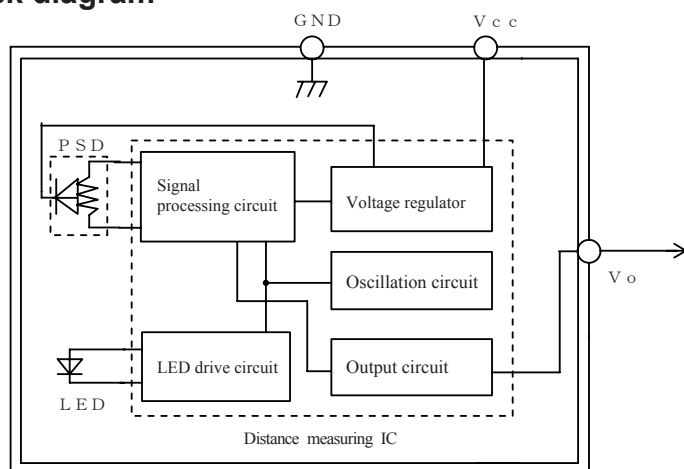
■Agency approvals/Compliance

1. Compliant with RoHS directive (2002/95/EC)

■Applications

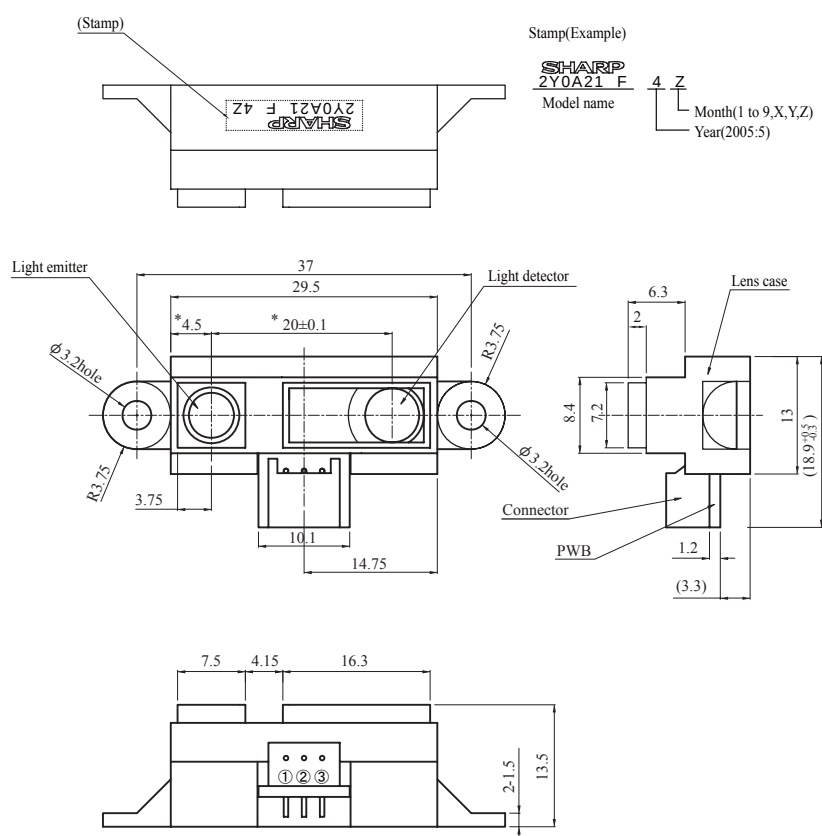
1. Touch-less switch
(Sanitary equipment, Control of illumination, etc.)
2. Robot cleaner
3. Sensor for energy saving
(ATM, Copier, Vending machine)
4. Amusement equipment
(Robot, Arcade game machine)

Block diagram



Outline Dimensions

(Unit : mm)



Connector signal

signal name
① Vo
② GND
③ Vcc

Connector :
J.S.T. TRADING COMPANY, LTD,
S3B-PH

Materials

Lens : Acrylic acid resin
(Visible light cut-off resin)
Case : Carbonic ABS
(Conductive resin)
PWB : Paper phenol

Note 1. The dimensions marked * are described the dimensions of lens center position.

Note 2. Unspecified tolerances shall be ± 0.3 mm.

Note 3. The dimensions in parenthesis are shown for reference.

Product mass : Approx. 3.6g

■ Absolute Maximum Ratings ($T_a=25^{\circ}\text{C}$, $V_{CC}=5\text{V}$)

Parameter	Symbol	Rating	Unit
Supply voltage	V_{CC}	-0.3 to +7	V
Output terminal voltage	V_O	-0.3 to $V_{CC}+0.3$	V
Operating temperature	T_{opr}	-10 to +60	$^{\circ}\text{C}$
Storage temperature	T_{stg}	-40 to +70	$^{\circ}\text{C}$

■ Electro-optical Characteristics ($T_a=25^{\circ}\text{C}$, $V_{CC}=5\text{V}$)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Average supply current	I_{CC}	L=80cm (Note 1)	—	30	40	mA
Distance measuring	ΔL	(Note 1)	10	—	80	cm
Output voltage	V_O	L=80cm (Note 1)	0.25	0.4	0.55	V
Output voltage differential	ΔV_O	Output voltage difference between L=10cm and L=80cm (Note 1)	1.65	1.9	2.15	V

* L : Distance to reflective object

Note 1 : Using reflective object : White paper (Made by Kodak Co., Ltd. gray cards R-27•white face, reflectance; 90%)

■ Recommended operating conditions

Parameter	Symbol	Rating	Unit
Supply voltage	V_{CC}	4.5 to 5.5	V

Fig. 1 Timing chart

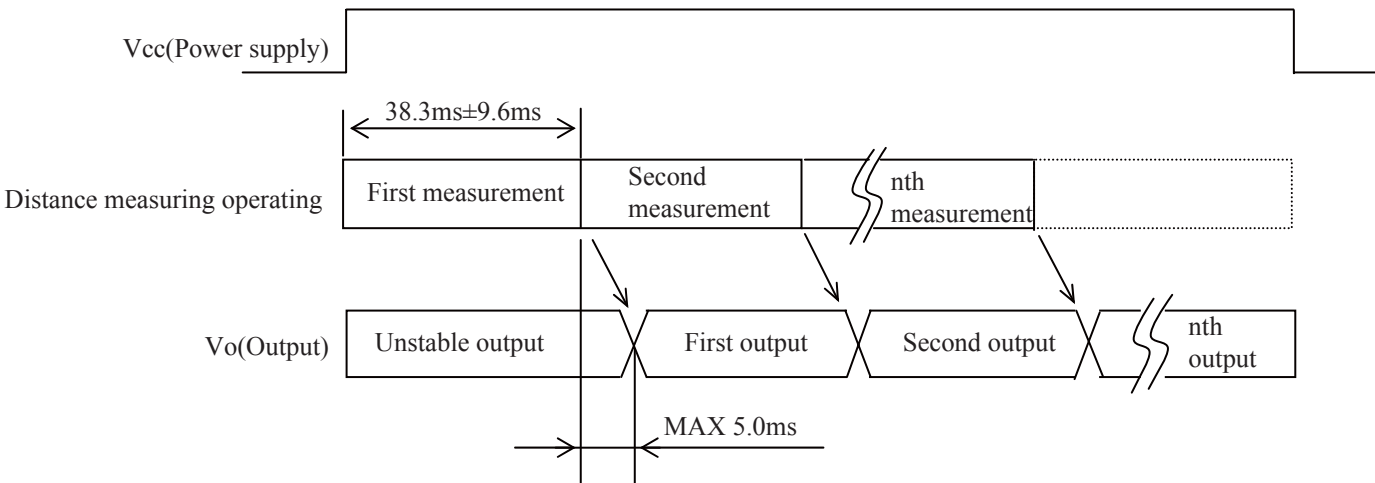
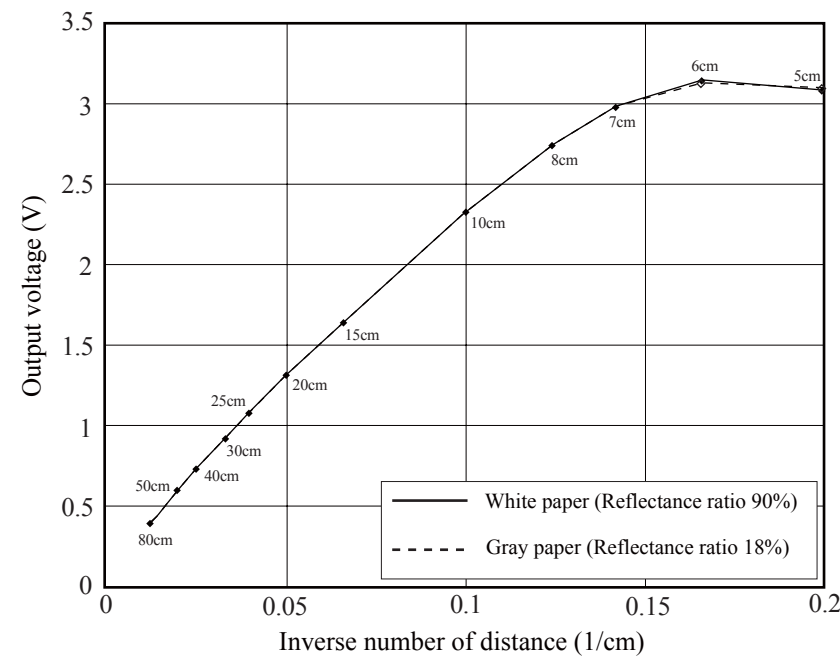
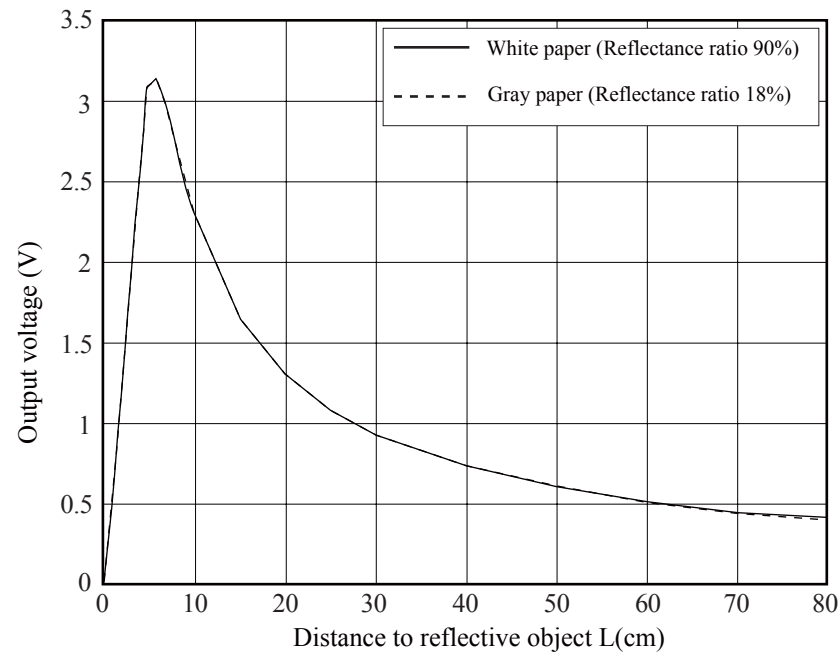


Fig. 2 Example of distance measuring characteristics(output)



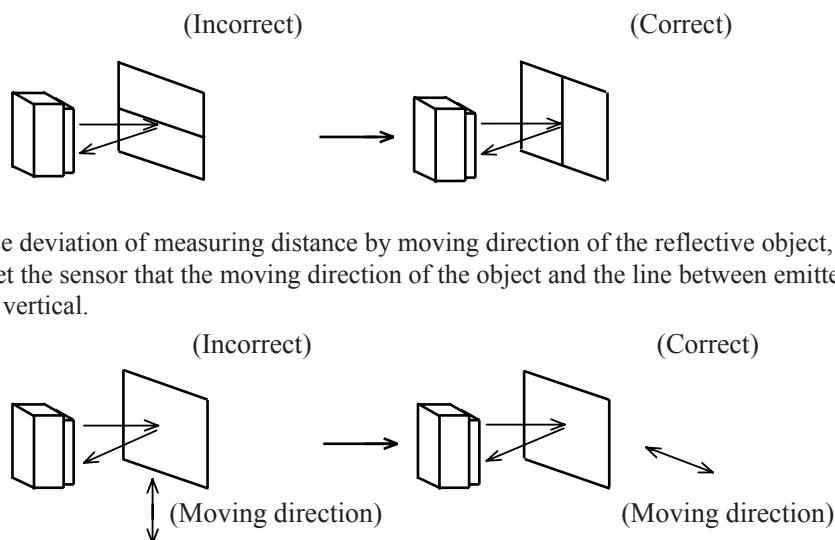
■Notes

●Advice for the optics

- The lens of this device needs to be kept clean. There are cases that dust, water or oil and so on deteriorate the characteristics of this device. Please consider in actual application.
- Please don't do washing. Washing may deteriorate the characteristics of optical system and so on. Please confirm resistance to chemicals under the actual usage since this product has not been designed against washing.

●Advice for the characteristics

- In case that an optical filter is set in front of the emitter and detector portion, the optical filter which has the most efficient transmittance at the emitting wavelength range of LED for this product ($\lambda = 870 \pm 70\text{nm}$), shall be recommended to use. Both faces of the filter should be mirror polishing. Also, as there are cases that the characteristics may not be satisfied according to the distance between the protection cover and this product or the thickness of the protection cover, please use this product after confirming the operation sufficiently in actual application.
- In case that there is an object near to emitter side of the sensor between sensor and a detecting object, please use this device after confirming sufficiently that the characteristics of this sensor do not change by the object.
- When the detector is exposed to the direct light from the sun, tungsten lamp and so on, there are cases that it can not measure the distance exactly. Please consider the design that the detector is not exposed to the direct light from such light source.
- Distance to a mirror reflector can not be sometimes measured exactly.
In case of changing the mounting angle of this product, it may measure the distance exactly.
- In case that reflective object has boundary line which material or color etc. are excessively different, in order to decrease deviation of measuring distance, it shall be recommended to set the sensor that the direction of boundary line and the line between emitter center and detector center are in parallel.



●Advice for the power supply

- In order to stabilize power supply line, we recommend to insert a by-pass capacitor of $10\mu\text{F}$ or more between Vcc and GND near this product.

●Notes on handling

- There are some possibilities that the internal components in the sensor may be exposed to the excessive mechanical stress. Please be careful not to cause any excessive pressure on the sensor package and also on the PCB while assembling this product.

● Presence of ODC etc.

This product shall not contain the following materials.

And they are not used in the production process for this product.

Regulation substances : CFCs, Halon, Carbon tetrachloride, 1.1.1-Trichloroethane (Methylchloroform)

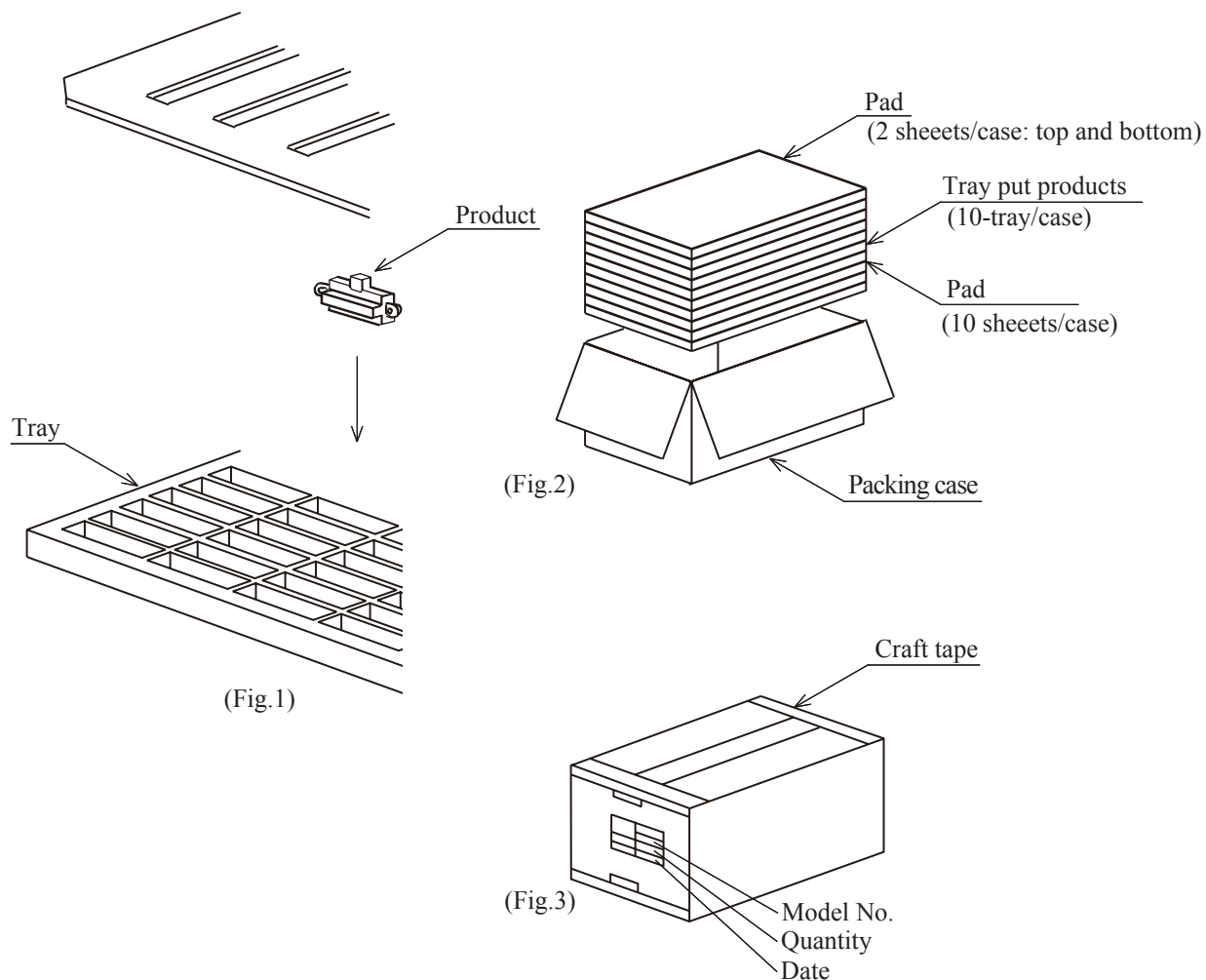
Specific brominated flame retardants such as the PBB and PBDE are not used in this product at all.

This product shall not contain the following materials banned in the RoHS Directive (2002/95/EC).

- Lead, Mercury, Cadmium, Hexavalent chromium, Polybrominated biphenyls (PBB), Polybrominated diphenyl ethers (PBDE).

■ Package specification

Package composition



Packaging method

1. Put products of 100pcs. in tray. packing method is showed in the above fig.(Fig.1)
2. Put them(10-tray) in the packing box. Put pads on their top and bottom.
And put pads on each trays(Total 10 sheets) (Fig.2).
3. Seal the packing box with craft tape.
Print the model No.,quantity,inspection date (1000 pcs./a packing box)(Fig.3).

■ Important Notices

· The circuit application examples in this publication are provided to explain representative applications of SHARP devices and are not intended to guarantee any circuit design or license any intellectual property rights. SHARP takes no responsibility for any problems related to any intellectual property right of a third party resulting from the use of SHARP's devices.

· Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device. SHARP reserves the right to make changes in the specifications, characteristics, data, materials, structure, and other contents described herein at any time without notice in order to improve design or reliability. Manufacturing locations are also subject to change without notice.

· Observe the following points when using any devices in this publication. SHARP takes no responsibility for damage caused by improper use of the devices which does not meet the conditions and absolute maximum ratings to be used specified in the relevant specification sheet nor meet the following conditions:

(i) The devices in this publication are designed for use in general electronic equipment designs such as:

- Personal computers
- Office automation equipment
- Telecommunication equipment [terminal]
- Test and measurement equipment
- Industrial control
- Audio visual equipment
- Consumer electronics

(ii) Measures such as fail-safe function and redundant design should be taken to ensure reliability and safety when SHARP devices are used for or in connection

with equipment that requires higher reliability such as:

- Transportation control and safety equipment (i.e., aircraft, trains, automobiles, etc.)
- Traffic signals
- Gas leakage sensor breakers
- Alarm equipment
- Various safety devices, etc.

(iii) SHARP devices shall not be used for or in connection with equipment that requires an extremely high level of reliability and safety such as:

- Space applications
- Telecommunication equipment [trunk lines]
- Nuclear power control equipment
- Medical and other life support equipment (e.g., scuba).

· If the SHARP devices listed in this publication fall within the scope of strategic products described in the Foreign Exchange and Foreign Trade Law of Japan, it is necessary to obtain approval to export such SHARP devices.

· This publication is the proprietary product of SHARP and is copyrighted, with all rights reserved. Under the copyright laws, no part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, in whole or in part, without the express written permission of SHARP. Express written permission is also required before any use of this publication may be made by a third party.

· Contact and consult with a SHARP representative if there are any questions about the contents of this publication.

E Datablad: GP2Y0A51SK0

Datablad för IR-sensor 2 till 15 cm avståndsmätning[46].

SHARP

GP2Y0A51SK0F

GP2Y0A51SK0F

Distance Measuring Sensor Unit
Measuring distance : 2 to 15 cm
Analog output type



■Description

GP2Y0A51SK0F is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector) , IR-LED (infrared emitting diode) and signal processing circuit. The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method. This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor.

■Agency approvals/Compliance

1. Compliant with RoHS directive (2011/65/EU)

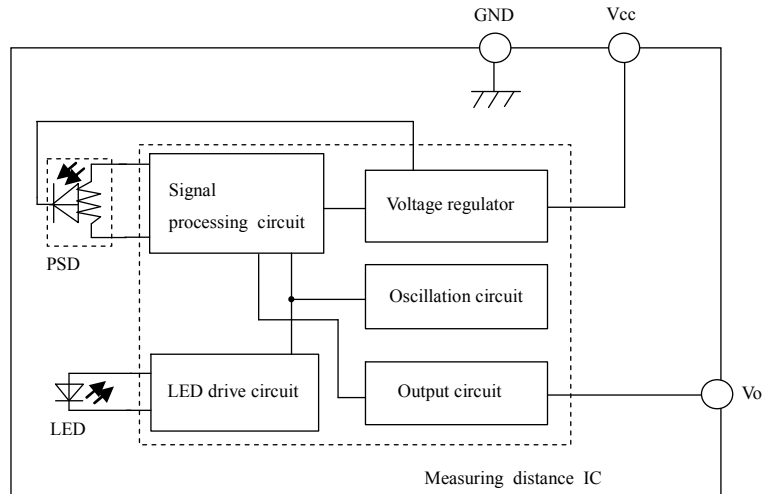
■Applications

1. Cleaning robot
2. Personal robot
3. Sanitary

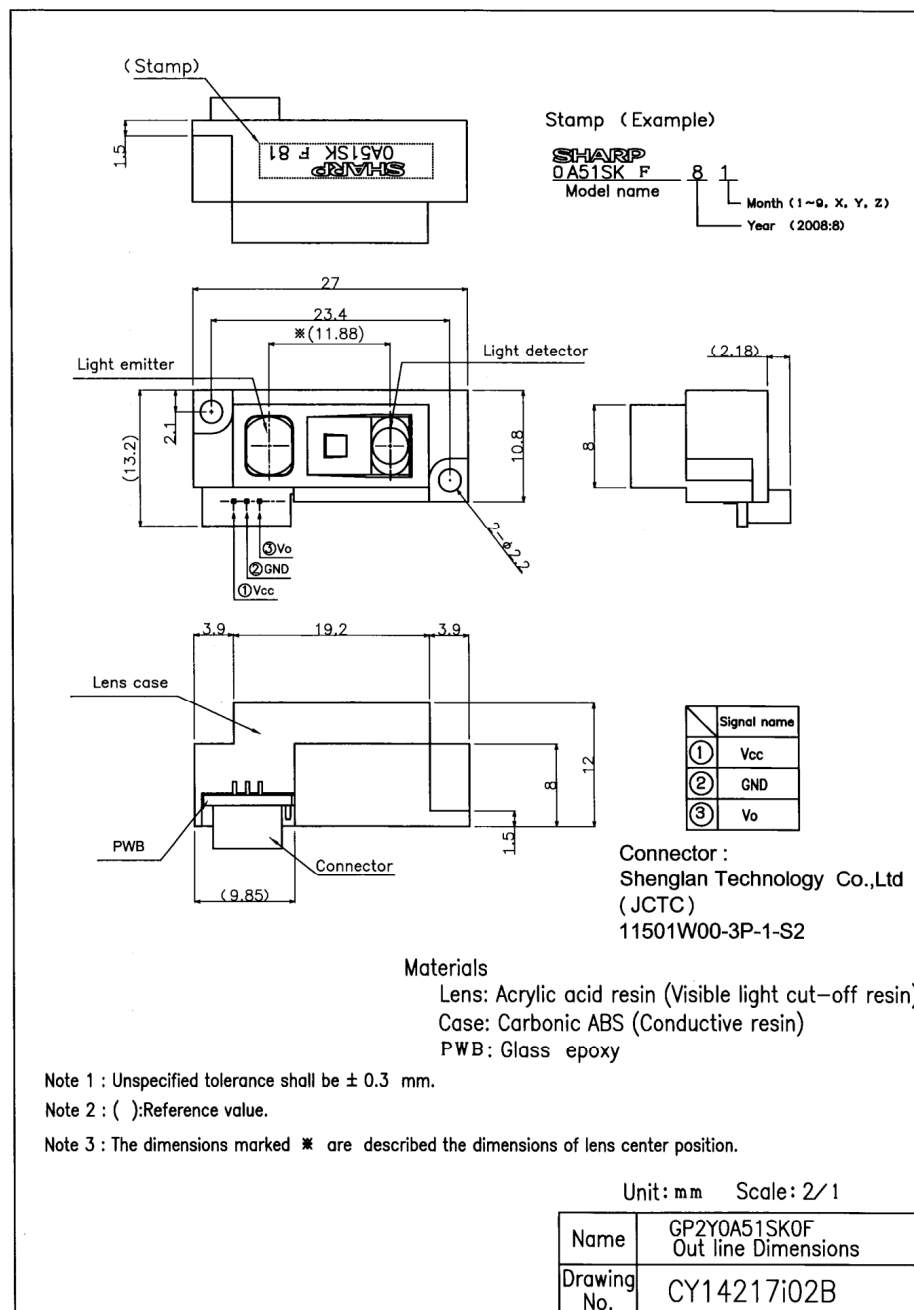
■Features

1. Distance measuring sensor is united with PSD, infrared LED and signal processing circuit
2. Short measuring cycle (16.5ms)
3. Distance measuring range : 2 to 15 cm
4. Compact size (27.0 × 10.8 × 12.0mm)
5. Analog output type

■Schematic



■Outline



■Absolute maximum ratings

(Ta=25°C)

Parameter	Symbol	Ratings	Unit	Remark
Supply voltage	Vcc	-0.3 to +7	V	-
Output terminal voltage	Vo	-0.3 to Vcc+0.3	V	-
Operating temperature	Topr	-10 to +60	°C	-
Storage temperature	Tstg	-40 to +70	°C	-

■Operating supply voltage

Symbol	Rating	Unit	Remark
Vcc	4.5 to 5.5	V	-

■Electro-optical Characteristics

(Ta=25°C, Vcc=5V)

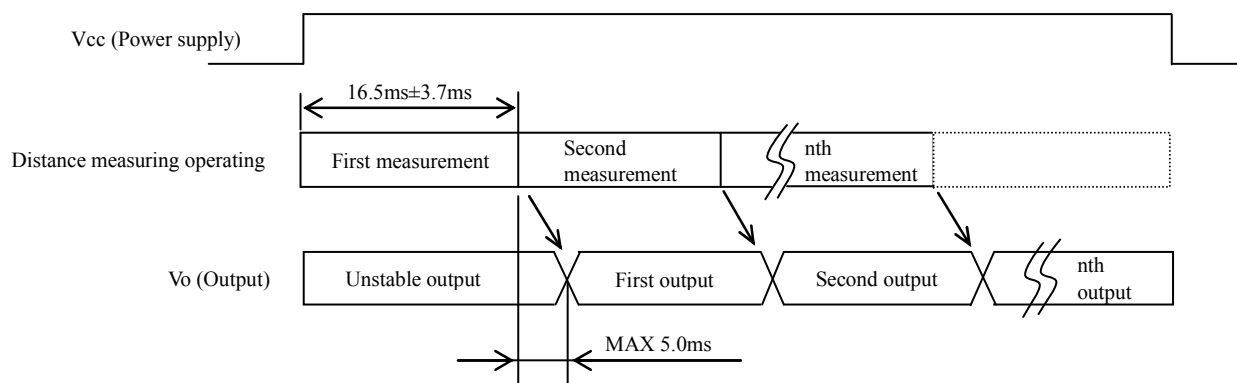
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Measuring distance range	ΔL	(Note 1)	2	-	15	cm
Output terminal voltage	Vo	L=15cm (Note 1)	0.25	0.4	0.55	V
Output voltage difference	ΔVo	Output change at L change (15cm → 2cm) (Note 1)	1.35	1.65	1.95	V
Average supply current	Icc	L=15cm (Note 1)	-	12	22	mA

※L : Distance to reflective object

(Note 1) Using reflective object : White paper

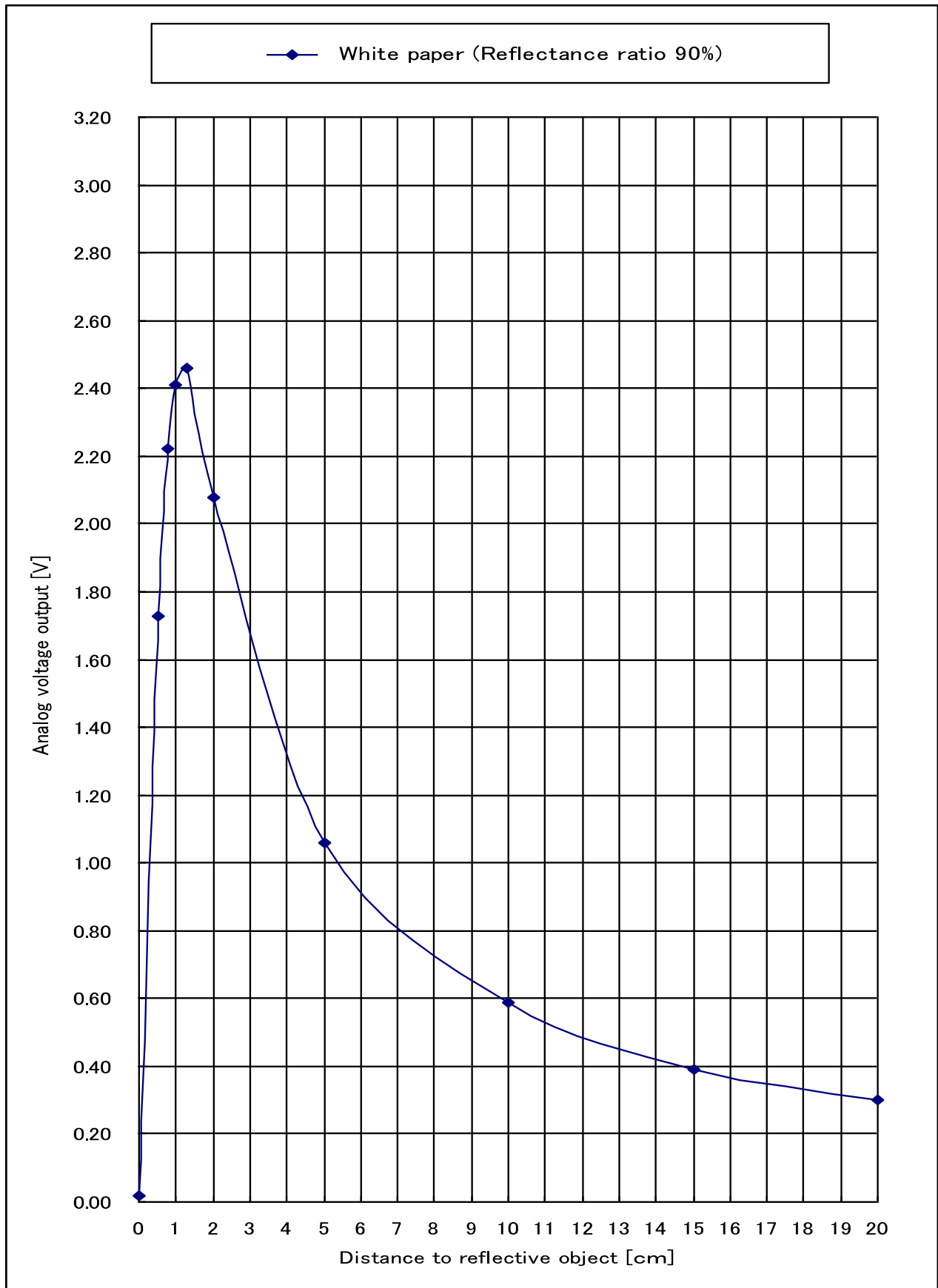
(Made by Kodak Co., Ltd. gray cards R-27 • white face, reflective ratio ; 90%)

■Timing Chart



■Supplements

- Example of output distance characteristics



- This product shall not contain the following materials.

Also, the following materials shall not be used in the production process for this product.

Materials for ODS : CFCs, Halon, Carbon tetrachloride 1.1.1-Trichloroethane (Methyl chloroform)

- Product mass : Approx. 2.7g (TYP)

- This product does not contain the chemical materials regulated by RoHS directive.

(Except for the NOT regulated by RoHS directive.)

- Specified brominated flame retardants

Specified brominated flame retardants (PBB and PBDE) are not used in this device at all.

- Compliance with each regulation

- 1) The RoHS directive(2011/65/EU)

This product complies with the RoHS directive(2011/65/EU).

Object substances: mercury, lead, cadmium, hexavalent chromium, polybrominated biphenyls (PBB)

and polybrominated diphenyl ethers (PBDE)

- 2) Content of six substances specified in Management Methods for Control of Pollution Caused by Electronic Information Products Regulation (Chinese : 电子信息产品污染控制管理办法).

Category	Hazardous substances					
	Lead (Pb)	Mercury (Hg)	Cadmium (Cd)	Hexavalent chromium (Cr ⁶⁺)	Polybrominated biphenyls (PBB)	Polybrominated diphenyl ethers (PBDE)
Distance Measuring Sensor	*	✓	✓	✓	✓	✓

This table is prepared in accordance with the provisions of SJ/T 11364.

✓ : Indicates that said hazardous substance contained in all of the homogeneous materials for this part is below the limit requirement of GB/T 26572

* : Indicates that said hazardous substance contained in at least one of the homogeneous materials used for this part is above the limit requirement of GB/T 26572

Lead in glass of electronic components (designated by "*" in the above table) are exempt from the RoHS directive (2011/65/EU), because there is no effective way to eliminate or substitute them by present scientific technology.

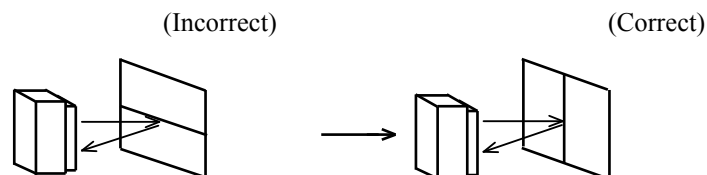
■Notes

[Advice for the optics]

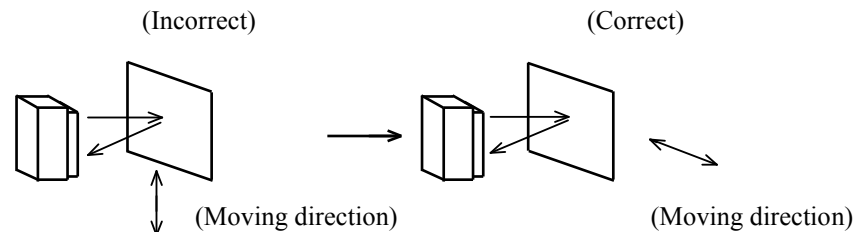
- Lens of this device shall be kept cleanly. There are cases that dust, water or oil and so on deteriorate the characteristics of this device. Please consider in actual application.
- In case that protection is set in front of the emitter and detector portion, the protection cover which has the most efficient transmittance at the emitting wavelength range of LED for this product ($\lambda=870\text{nm}\pm 70\text{nm}$), shall be recommended to use. The face and back of protection cover should be mirror polishing. Also, as there are cases that the characteristics may not be satisfied with according to the distance between the protection cover and this product or the thickness of the protection cover, please use this product after confirming the operation sufficiently in actual application.

[Advice for the characteristics]

- In case that there is an object near to light exits of the sensor between the sensor and the detected object, please use this device after confirming sufficiently what the characteristics of this sensor do not change by the object.
- When the detector surface receive direct light from the sun, tungsten lamp and so on, there are cases that it can not measure the distance exactly. Please consider the design that the detector does not receive direct light from such light source.
- Distance between sensor and mirror reflector can not measure exactly.
- In case that reflective object has boundary line clearly, there is cases that distance can not measure exactly. At that time, if direction of boundary line and the line between emitter center and detector center parallels, it is possible to decrease deviation of measuring distance.



- In order to decrease measuring error by moving direction of object, we recommend to mount the sensor like below drawing.

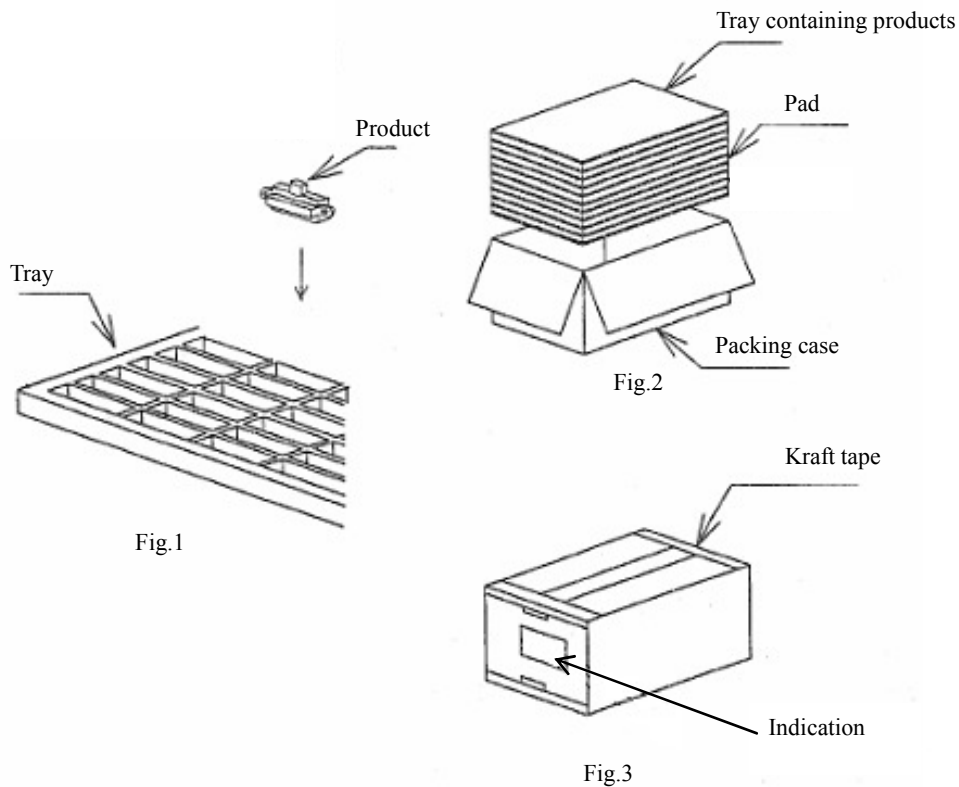


- In order to stabilize power supply line, we recommend to connect a by-pass capacitor of $10\mu\text{F}$ or more between V_{cc} and GND near this product.

[Notes on handling]

- Please don't do washing. Washing may deteriorate the characteristics of optical system and so on. Please confirm resistance to chemicals under the actual usage since this product has not been designed against for washing.
- There are some possibilities that the sensor inside the case package with lens may be exposed to the excessive mechanical force. Please be careful not to cause any excessive pressure on the case package with lens and also on the sensor's PCB at the assembly and inserting of the set.

■Packing specification



1. Packing numbers
 MAX. 50 pieces per tray
 MAX 500 pieces per case
2. Arranges in 10 stages of trays containing products into the packing case.
 Put pads on their top and bottom.
 Closes the lid of case and seals with kraft tape.
3. Indication items
 The contents of the carton indication conforms to EIAJ C-3 and the following items are indicated.
 Model No., Internal production control name, Quantity, Packing date, Corporate name, Country of origin

■Important Notices

· The circuit application examples in this publication are provided to explain representative applications of SHARP devices and are not intended to guarantee any circuit design or license any intellectual property rights. SHARP takes no responsibility for any problems related to any intellectual property right of a third party resulting from the use of SHARP's devices.

· Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device. SHARP reserves the right to make changes in the specifications, characteristics, data, materials, structure, and other contents described herein at any time without notice in order to improve design or reliability. Manufacturing locations are also subject to change without notice.

· Observe the following points when using any devices in this publication. SHARP takes no responsibility for damage caused by improper use of the devices which does not meet the conditions and absolute maximum ratings to be used specified in the relevant specification sheet nor meet the following conditions:

(i) The devices in this publication are designed for use in general electronic equipment designs such as:

- Personal computers
- Office automation equipment
- Telecommunication equipment [terminal]
- Test and measurement equipment
- Industrial control
- Audio visual equipment
- Consumer electronics

(ii) Measures such as fail-safe function and redundant design should be taken to ensure reliability and safety when SHARP devices are used for or in connection

with equipment that requires higher reliability such as:

- Transportation control and safety equipment (i.e., aircraft, trains, automobiles, etc.)
- Traffic signals
- Gas leakage sensor breakers
- Alarm equipment
- Various safety devices, etc.

(iii) SHARP devices shall not be used for or in connection with equipment that requires an extremely high level of reliability and safety such as:

- Space applications
- Telecommunication equipment [trunk lines]
- Nuclear power control equipment
- Medical and other life support equipment (e.g., scuba).

· If the SHARP devices listed in this publication fall within the scope of strategic products described in the Foreign Exchange and Foreign Trade Law of Japan, it is necessary to obtain approval to export such SHARP devices.

· This publication is the proprietary product of SHARP and is copyrighted, with all rights reserved. Under the copyright laws, no part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, in whole or in part, without the express written permission of SHARP. Express written permission is also required before any use of this publication may be made by a third party.

· Contact and consult with a SHARP representative if there are any questions about the contents of this publication.

F Datablad VL53L0X

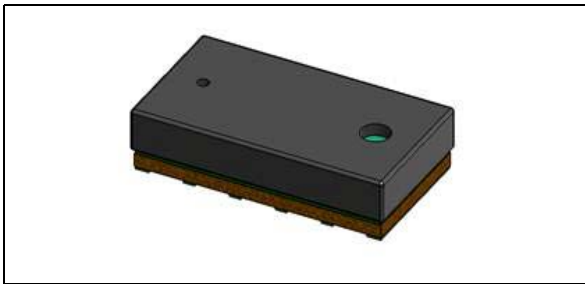
Väsentliga delar av datablad för lasorsensorn [48].



VL53L0X

World's smallest Time-of-Flight ranging and gesture detection sensor

Datasheet - production data



Features

- Fully integrated miniature module
 - 940 nm laser VCSEL
 - VCSEL driver
 - Ranging sensor with advanced embedded micro controller
 - 4.4 x 2.4 x 1.0 mm
- Fast, accurate distance ranging
 - Measures absolute range up to 2 m
 - Reported range is independent of the target reflectance
 - Advanced embedded optical cross-talk compensation to simplify cover glass selection
- Eye safe
 - Class 1 laser device compliant with latest standard IEC 60825-1:2014 - 3rd edition
- Easy integration
 - Single reflowable component
 - No additional optics
 - Single power supply
 - I2C interface for device control and data transfer
 - Xshutdown (reset) and interrupt GPIO
 - Programmable I2C address

Applications

- User detection for personal computers/ laptops/tablets and IoT (energy saving)
- Robotics (obstacle detection)
- White goods (hand detection in automatic faucets, soap dispensers etc.)
- 1D gesture recognition.
- Laser assisted autofocus. Enhances and speeds up camera autofocus system performance, especially in difficult scenes (low light levels, low contrast) or fast moving video mode.

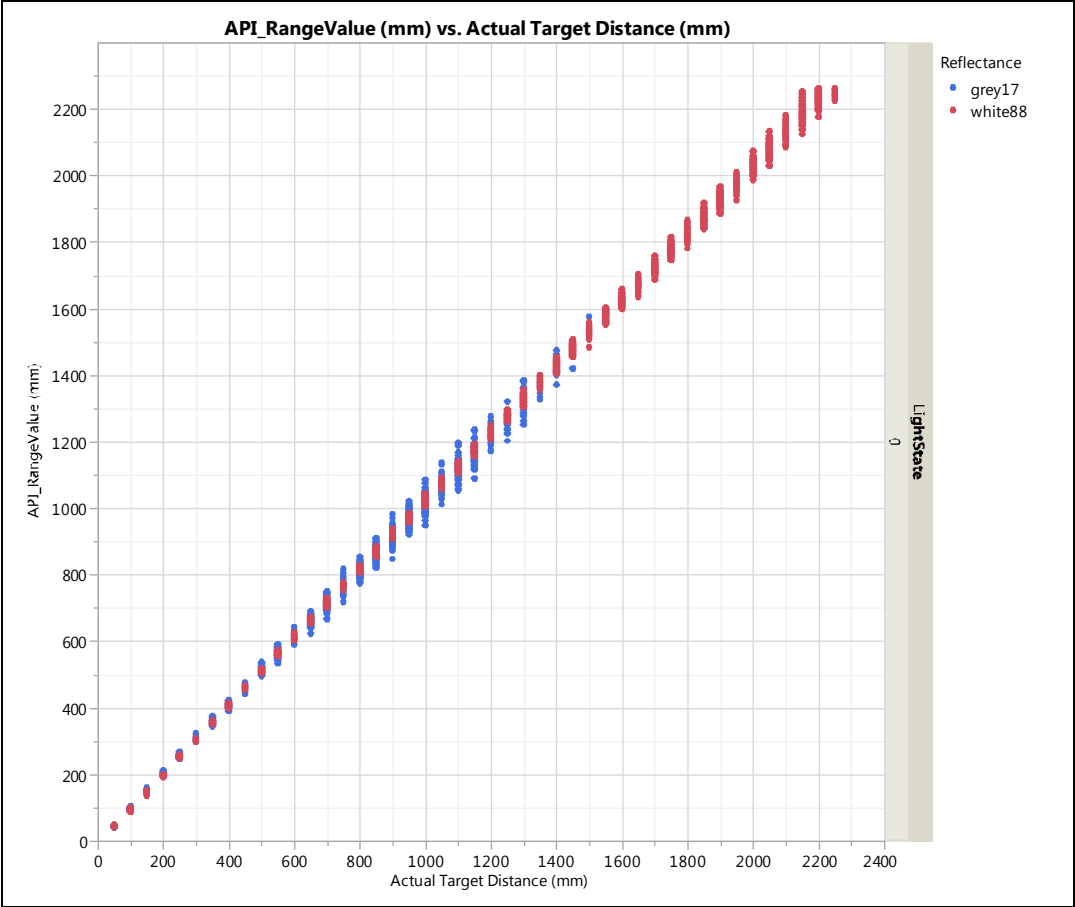
Description

The VL53L0X is a new generation Time-of-Flight (ToF) laser-ranging module housed in the smallest package on the market today, providing accurate distance measurement whatever the target reflectances unlike conventional technologies. It can measure absolute distances up to 2m, setting a new benchmark in ranging performance levels, opening the door to various new applications.

The VL53L0X integrates a leading-edge SPAD array (Single Photon Avalanche Diodes) and embeds ST's second generation FlightSense™ patented technology.

The VL53L0X's 940 nm VCSEL emitter (Vertical Cavity Surface-Emitting Laser), is totally invisible to the human eye, coupled with internal physical infrared filters, it enables longer ranging distances, higher immunity to ambient light, and better robustness to cover glass optical crosstalk.

Figure 21. Typical ranging - long range mode



5.2 Max ranging distance

Table 11 presents the ranging specification for VL53L0X bare module, without cover glass, at room temperature (23degreesC) and with nominal voltage (2.8Volts).

Table 11. Max ranging capabilities with 33ms timing budget

Target reflectance level (full FOV)	Conditions	Indoor (2)	Outdoor overcast (2)
White target (88%)	Typical	200cm+ (1)	80cm
	Minimum	120cm	60cm
Grey target (17%)	Typical	80cm	50cm
	Minimum	70cm	40cm

Note (1): using long range API profile

5.3.2 Range profile examples

[Table 13](#) details typical performance for the four example ranging profiles, as per measurement conditions in [Section 5.3: Ranging accuracy](#).

Table 13. Range profiles

Range profile	Range timing budget	Typical performance	Typical application
Default mode	30 ms	1.2 m, accuracy as per Table 12	Standard
High accuracy	200 ms	1.2 m, accuracy < +/- 3 %	Precise measurement
Long range	33 ms	2 m, accuracy as per Table 12	Long ranging, only for dark conditions (no IR)
High speed	20 ms	1.2 m, accuracy +/- 5 %	High speed where accuracy is not priority

5.3.3 Ranging offset error

The table below shows how range offset may drift over distance, voltage and temperature.

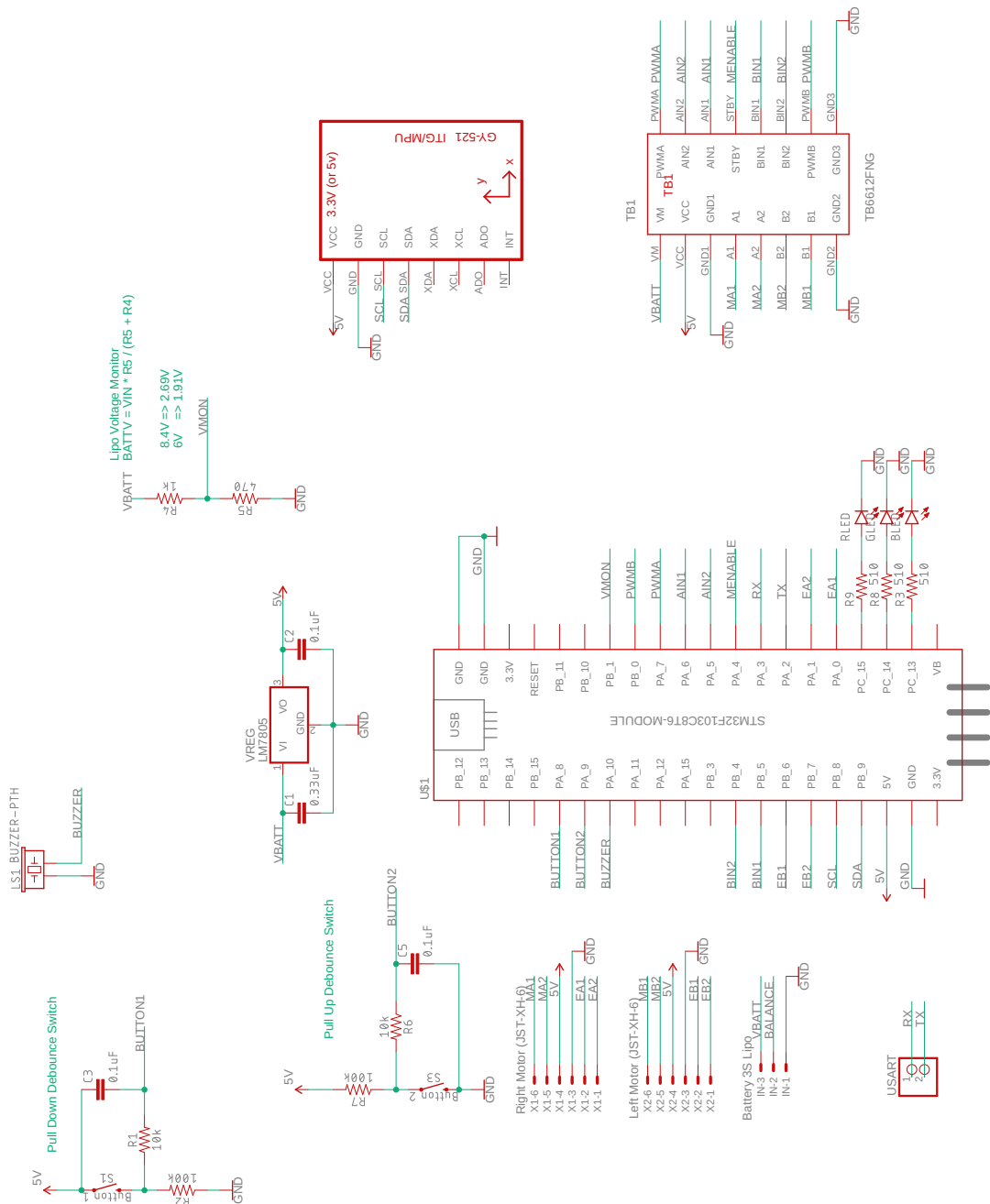
Assumes offset calibrated at 10cm. See VL53L0X API User Manual for details on offset calibration.

Table 14. Ranging offset

	Nominal conditions	Measure point	Typical offset from nominal	Maximum offset from nominal
Ranging distance	Offset calibration at 10 cm ("zero")	White 120 cm (indoor) Grey 70 cm (indoor) White 60 cm (outdoor) Grey 40 cm (outdoor)		< 3 %
Voltage drift	2.8 V	2.6 V to 3.5 V	+/- 10 mm	+/- 15 mm
Temperature drift	23 °C	-20°C to +70°C	+/- 10 mm	+/- 30 mm

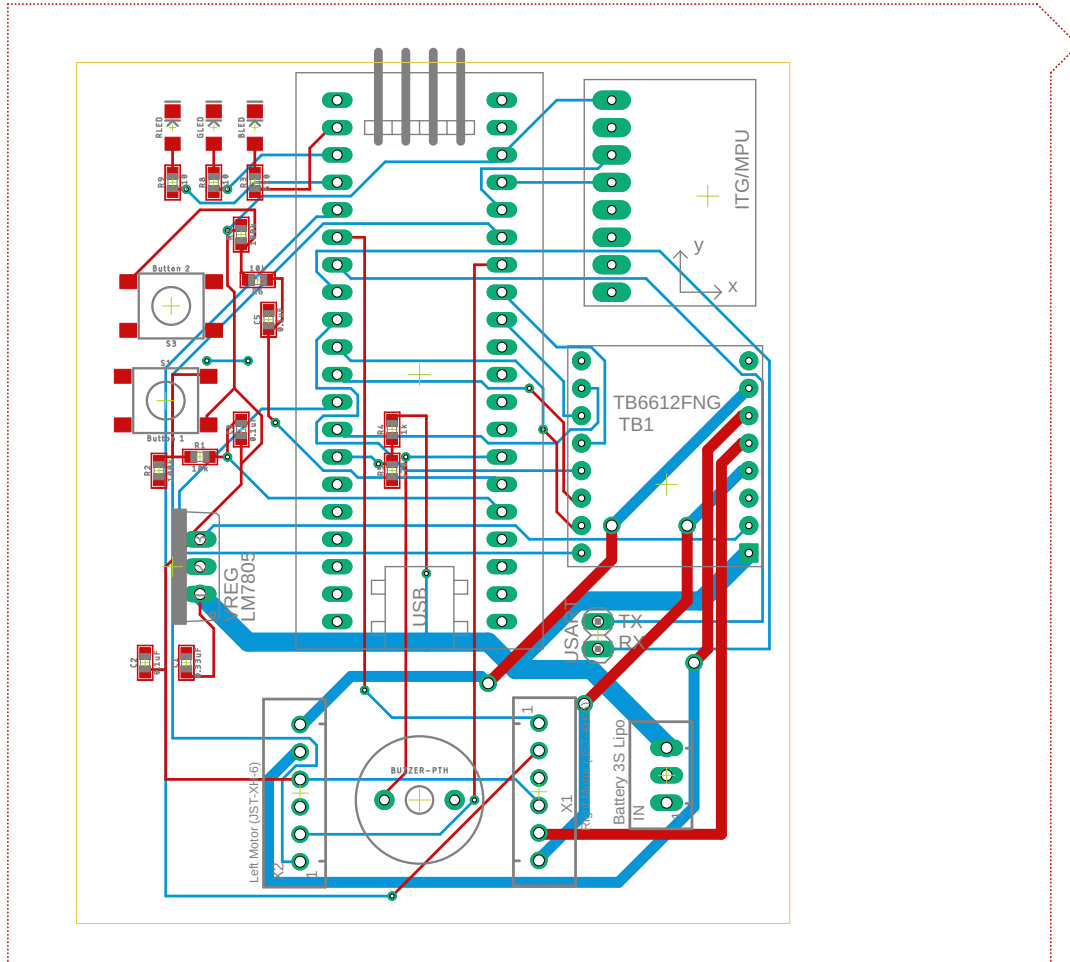
G Elschema för kretskort

Elschema skapat med Eagle från Autodesk. Eget material.



H Kretskortsritning

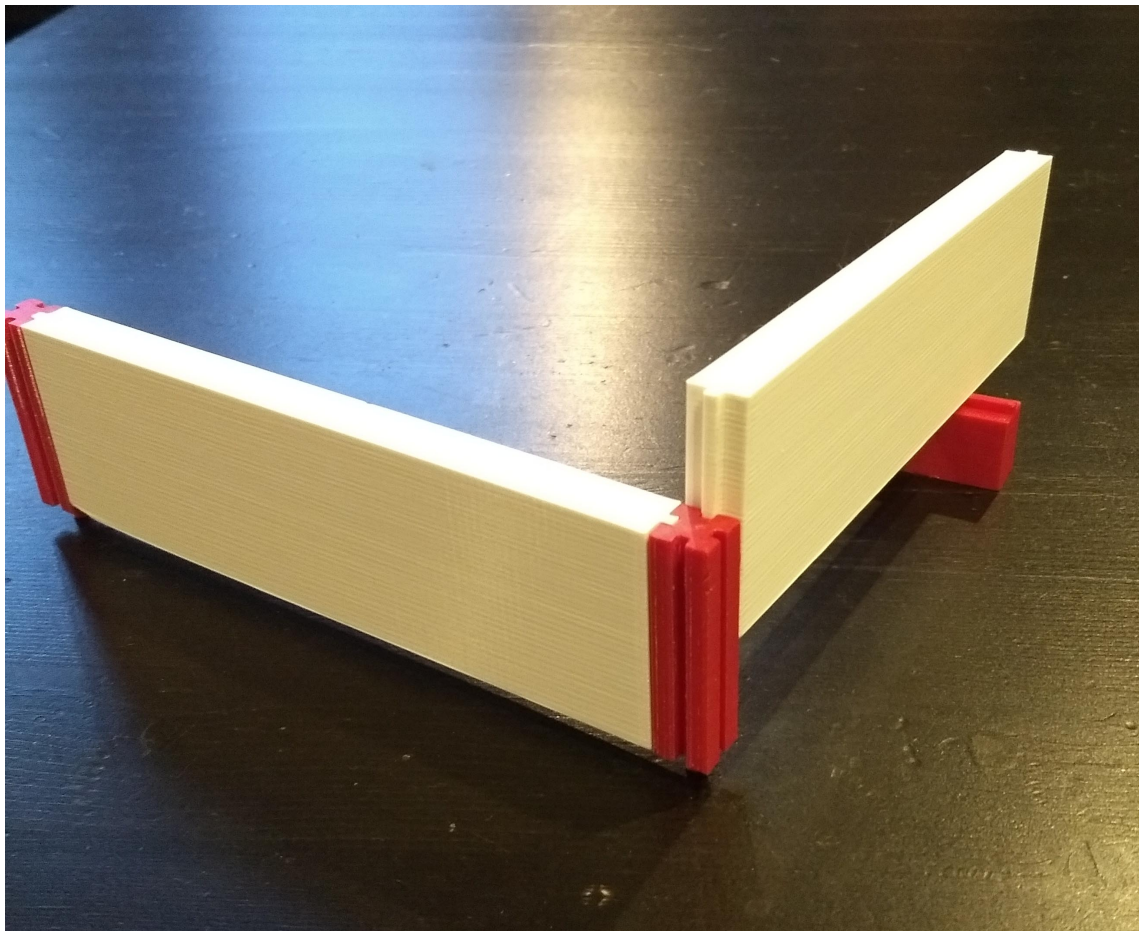
Kretskort skapat med Eagle från Autodesk. Eget material.



I Labyrintkoncept

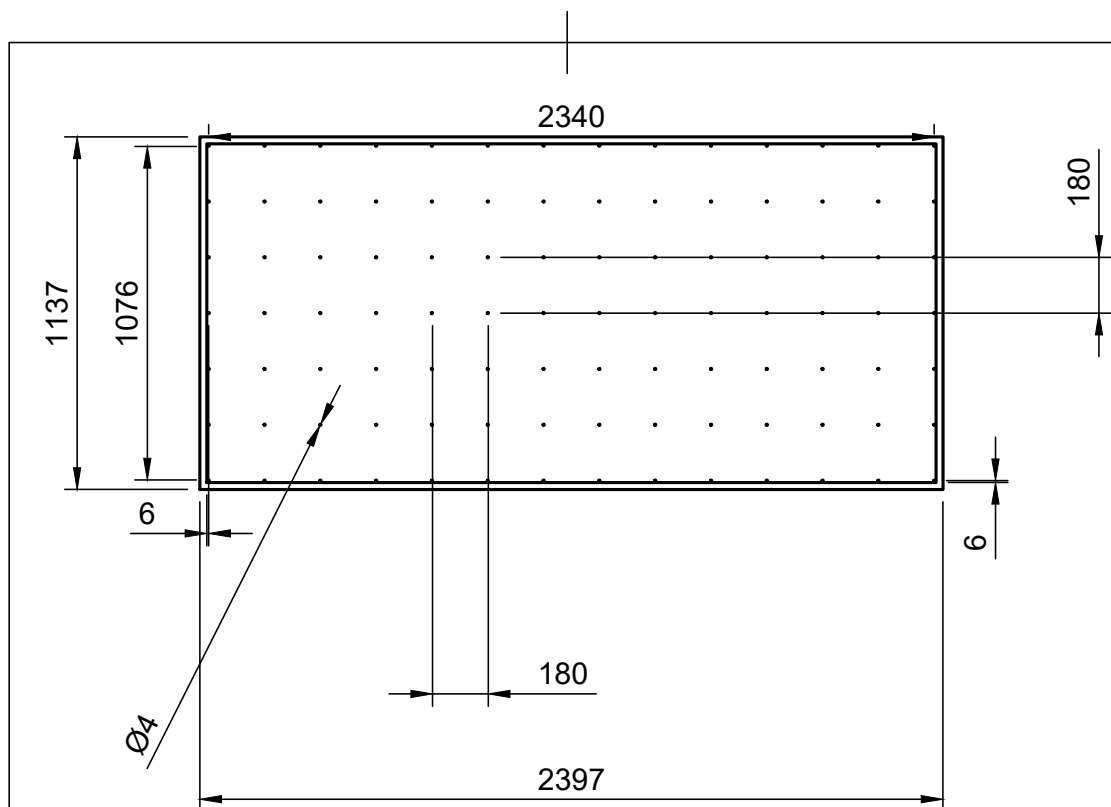
Labyrintritningarna är framtaget i programmet Fusion 360 med ett koncept framtaget med inspiration från [49].

En mindre variant av en tävlingslabyrint har byggts. Vid undersökning av material ansågs det bäst att bygga en labyrint av 6x13 stycken celler enligt reglerna för tävlingen, se bilaga A. Storleken på labyrinten ansågs tillräckligt stor för att göra relevanta tester av roboten utan att kräva för mycket resurser i form av tid och pengar. Labyrinten består av en bottenplatta, väggar och stolpar. I figur 35 kan en 3D-printat exemplar av väggar och stolpar ses. Bottenplattan har hål i hörnen av varje cell som fäste för stolparna vilka stabiliserar väggarna. Väggar, regler samt bottenplatta är byggda i trä medan stolparna har 3D-printats.



Figur 35: 3D-printad prototyp av väggar och stolpar. Eget material.

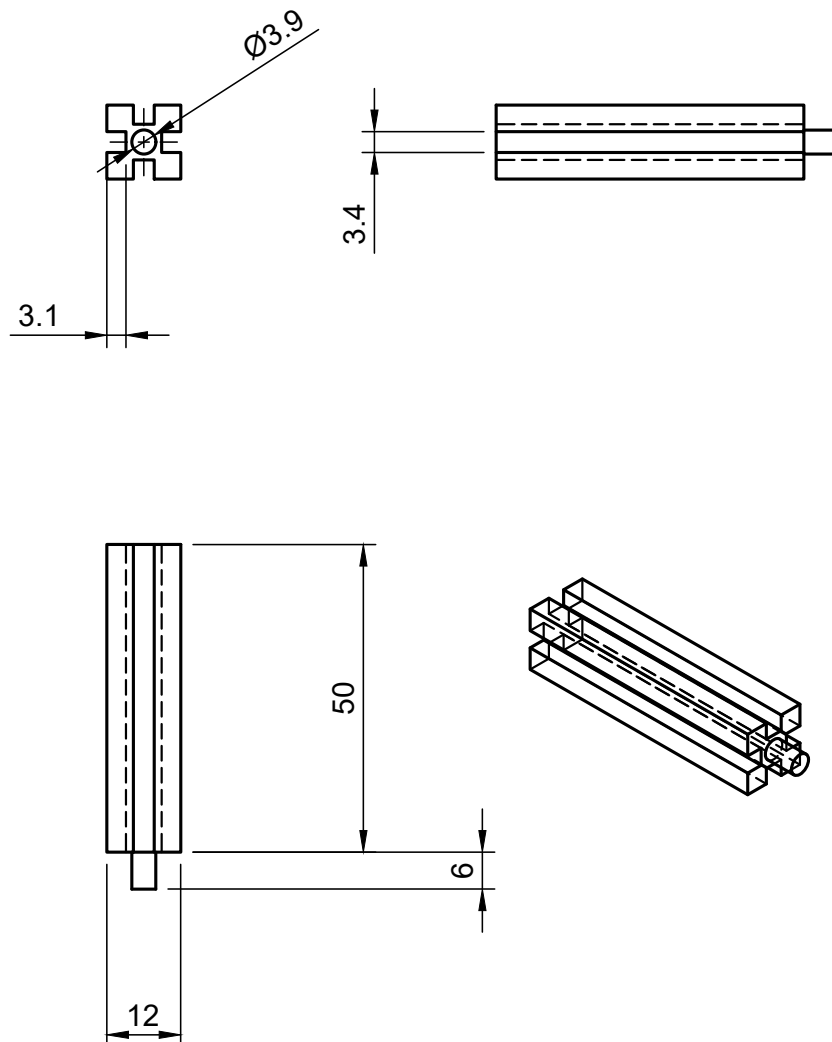
I.1 Bottenplatta



Bottenplattans tjocklek beror på vad som kan fås, hål bör var minimum 8 mm djupa

Dept.	Technical reference	Created by Ida Hagström 2019-02-19	Approved by
		Document type	Document status
		Title Bottenplatta	DWG No.
		XXX	Rev. Date of issue Sheet 1/1

I.2 Stolpe



Dept.	Technical reference	Created by Ida Hagström 2019-03-15	Approved by	
Stöd Stolpe till labyrint.		Document type	Document status	
		Title Stolpe	DWG No.	
		Rev.	Date of issue 2019-03-15	Sheet 1/1

I.3 Väggar

