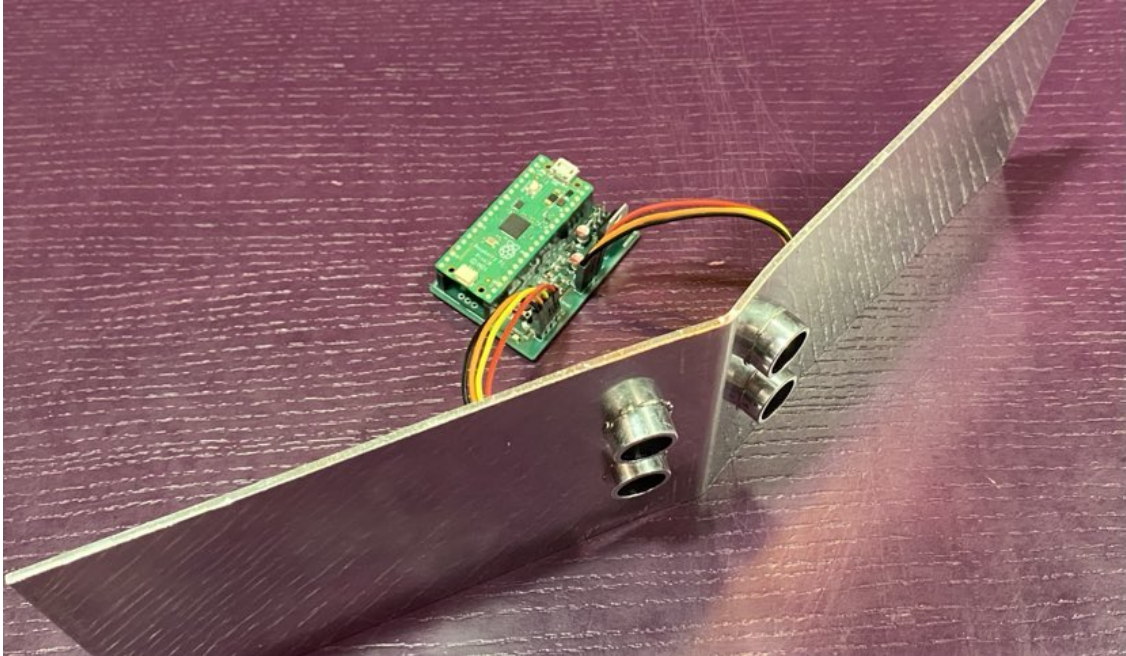




CHALMERS



# Sensorlösning för att detektera objekt inom ett begränsat arbetsområde

Sensorlösningen detekterar och skyddar människor från kollision med en rörlig enhet

Examensarbete inom högskoleprogrammet mekatronik

Martin Bengts  
Anton Olsson

**INSTITUTIONEN FÖR ELEKTRONIK**

CHALMERS TEKNISKA HÖGSKOLA  
Göteborg 2024  
[www.chalmers.se](http://www.chalmers.se)



EXAMENSARBETE 2024

**Rapporten redogör hur en sensorlösning kan  
konstrueras för att detektera objekt inom ett  
begränsat arbetsområde**

Sensorlösningen detekterar och skyddar människor från kollision med  
en rörlig enhet

Martin Bengts  
Anton Olsson



**CHALMERS**

Institutionen för Elektronik  
CHALMERS TEKNISKA HÖGSKOLA  
Göteborg 2024

Sensordlösning för att detektera objekt inom ett begränsat arbetsområde  
Sensordlösningen detekterar och skyddar människor från kollision med en rörlig enhet  
Martin Bengts och Anton Olsson

© Martin Bengts och Anton Olsson, 2024.

Handledare Chalmers: Björn Langborn, Institutionen för Elektronik  
Examinator Chalmers: Thomas Eriksson, Institutionen för Elektronik  
Handledare företag: Luca Petric, Combitech AB  
Handledare företag: Alexander Fransson, Combitech AB

Examensarbete 2024  
Institutionen för Elektronik  
Chalmers Tekniska Högskola  
SE-412 96 Göteborg  
Telefon +46 31 772 1000

Omslagsbild: Sensordlösning uppkopplad i testrigg.

Skriven i L<sup>A</sup>T<sub>E</sub>X  
Göteborg 2024

Sensorlösning för att detektera objekt inom ett begränsat arbetsområde  
Sensorlösningen detekterar och skyddar människor från kollision med en rörlig enhet  
Martin Bengts  
Anton Olsson  
Institutionen för Elektronik  
Chalmers Tekniska Högskola

## Sammanfattning

Följande rapport presenterar ett projektarbete, för Combitech, med uppgift att utveckla en sensorlösning som ska detektera objekt som befinner sig innanför ett specifikt arbetsområde. Syftet med detta är framförallt att skydda människor från skada genom att förhindra kollision med en rörlig enhet. Detta projekt ska dock fungera mer som ett förarbete där slutprodukten inte behöver vara redo för serieproduktion och det finns redan en del befintliga ingångsvärden som sensorlösningen måste anpassas efter.

Utifrån syftet med projektet ställdes en del frågeställningar som mer konkretiserade vad det egentligen var som skulle redas ut. Dessa frågeställningar handlade främst om vilken typ av sensor och hur många sensorer som skulle behöva användas. De handlade också om hur själva sensorlösningen skulle kunna konstrueras för att styra sensorerna och ha en god användarvänlighet. Inför projektet fanns även en del målsättningar, där största fokus låg i att mäta sensorlösningens förmåga att detektera objekt inom arbetsområdet samt inte detektera objekt utanför arbetsområdet eller helt enkelt objekt som inte anses vara värda att detektera. Andra målsättningar sattes även upp som handlade om till exempel lösningens justerbarhet, förmåga att klara av olika miljöfaktorer samt ett mål om kostnadsfrågan.

Projektets arbetsgång började med kortare studier och tester av olika sensortyper. Dessa sensorer styrdes med hjälp av ett utvecklingskort med mikrokontroller (programmerbar styrenhet) och enklare programkoder. Denna fas följdes sedan upp med ett slutval av sensorer följt av uppbyggnaden av det elektriska systemet för de valda sensorerna. Detta system skulle innehålla själva styrlogiken av sensorerna och andra viktiga funktioner på sensorlösningen. Utifrån det elektriska systemet kunde sedan ett kretskort designas och beställas tillsammans med de valda elektriska komponenterna. I projektets slutfas utfördes ett arbete med lödning av komponenterna på kretskortet samtidigt som en slutgiltig programkod för systemet byggdes upp. Till sist funktionstestades kortet för att evaluera dess prestanda utifrån givna mål.

Projektets sluttester visade goda resultat när det kom till detektion inom det specifika arbetsområdet med hela 99 procent säker detektering av ett objekt med en area på 200 x 200 mm. Samtidigt uppmättes inga falska detektioner när det försökte framkallas trots den snabba detektionstiden på 22 millisekunder. Sensorlösningens arbetsområde kan justeras med en tryckknapp och med hjälp av en blå LED på kretskortet indikeras vilket avstånd som är inställt. På kretskortet sitter också en

---

röd LED som indikerar när sensorlösningen detekterar objekt innanför arbetsområdet.

Sammanfattningsvis representerar rapporten ett arbete fram till en slutlösning för en produkt som till stor del uppfyller huvudmålen och en god detektionssäkerhet. Trots detta finns stora utvecklingsmöjligheter inom bland annat kostnadseffektivitet, testning samt optimering för prestandan i olika miljöer, vilket lämnar en del arbete för nästa version av produkten.

Nyckelord: sensorlösning, ultraljudsensorer, kretskort, detektion, mikrokontroller.

---

## Abstract

The following report presents a project assignment for Combitech, tasked with developing a sensor solution to detect objects within a specific work area. The primary purpose is to protect individuals from harm by preventing collisions with a moving unit. This project is more of a preparatory work, where the final product does not need to be ready for mass production, and there are already some existing input values that the sensor solution must be adapted to.

Based on the project's purpose several questions were asked to clarify what challenges needed to be addressed. These questions included everything from, which type to quantity of sensors, as well as how the sensor solution could be constructed to be able to effectively control the sensor while maintaining user-friendliness. The project also contained several objectives. The main focus of those was to be able to measure the sensor's ability to detect objects inside the working area while not show detection for objects outside the area or other disturbances. Other objectives for the project were also set regarding the sensor solution's adjust-ability and cost-effectiveness.

The project's workflow began with brief studies and tests of different kinds of sensor types. These sensors were controlled using a development board with a microcontroller (programmable control unit) and simple programming codes. This phase was then followed by a final selection of sensors along with construction of the electrical system for the chosen sensors. This system would encompass the control logic of the sensors and other important functions of the sensor solution. Based on the electrical system, a circuit board could then be designed and ordered along with the selected electrical components. In the final phase of the project, the work was to solder the components onto the circuit board while simultaneously developing a final program code for the system. Finally, the card was functionally tested to evaluate its performance based on predefined goals.

The project's final tests showed good results regarding detection within the specific work area, with a 99 percent accurate detection of an object measuring 200 x 200 mm. At the same time, no false detections were recorded despite the fast detection time of 22 milliseconds. The sensor solution's working range can be adjusted using a push button, and the distance set is indicated by a blue LED on the circuit board. Additionally, there is a red LED on the circuit board indicating when the sensor solution detects objects within the work area.

In summary, the report represents a work of a final solution for a product that largely meets its main objectives and provides good detection reliability. However, there are significant development opportunities in areas such as cost-effectiveness, testing, and performance optimization in various environments, leaving some work for the next version of the product.

Keywords: sensor solution, ultrasonic sensor, circuit board, detection, microcontroller.



## Förord

Detta är ett examensarbete som har skrivits under våren 2024 och har gjorts under institutionen för elektronik på Chalmers Tekniska Högskola. Examensarbetet omfattar 15 högskolepoäng och är den avslutande delen på programmet Mekatronik, Högskoleingenjör, 180 högskolepoäng. Med vår gemensamma framåtlutade och nyfikna inställning har arbetet flutit på bra, och vi har njutit av att utforska och göra framsteg tillsammans.

Vi vill tacka Combitech Jönköping som gav oss möjligheten att göra detta examensarbete. Vi har lärt oss mycket inom elektronikutveckling och inbyggda system och hoppas vi även har bidragit med något till er.

Ett STORT tack till våra handledare på Combitech, Luka Petric och Alexander Fransson, vars ovärderliga stöd och uppmuntran har varit en avgörande faktor för vår framgång under hela projektets gång.

Vi vill slutligen även tacka vår handledare på Chalmers, Björn Langborn. Björn har stöttat, svarat på frågor och kommit med feedback i tekniska såväl som planering och rapportmässiga frågor.

Martin Bengts och Anton Olsson, Göteborg, Maj 2024



# Innehåll

<b>Figurer</b>	<b>xiii</b>
<b>Tabeller</b>	<b>xv</b>
<b>1 Inledning</b>	<b>1</b>
1.1 Bakgrund . . . . .	1
1.2 Syfte . . . . .	2
1.3 Mål . . . . .	2
1.4 Avgränsningar . . . . .	3
1.5 Frågeställningar . . . . .	3
<b>2 Teoretisk och teknisk bakgrund</b>	<b>5</b>
2.1 Teoretisk bakgrund . . . . .	5
2.1.1 Uppkomst och funktion av ultraljudssensorn . . . . .	5
2.1.2 Sensor HC-SR04 . . . . .	7
2.1.3 Raspberry PI Pico . . . . .	8
2.1.4 Spännings Växling (7805) . . . . .	9
2.1.5 RC-filter . . . . .	10
2.1.6 PCB och PCB-design (KiCad) . . . . .	11
2.2 Tekniska specifikationer för projektet . . . . .	11
2.2.1 Arbetsområdet . . . . .	12
2.2.2 Elektriskt interface . . . . .	13
2.2.3 Signal logik . . . . .	13
<b>3 Metod</b>	<b>15</b>
3.1 Arbetsgång . . . . .	15
3.2 Tillvägagångssätt . . . . .	16
3.2.1 Förstudie . . . . .	16
3.2.2 Testning av sensorer . . . . .	16
3.2.3 Elschema och PCB-design . . . . .	18
3.2.4 Programmering . . . . .	18
3.2.5 Assembly och Sluttest . . . . .	19
3.3 Informationsinsamling . . . . .	22
<b>4 Resultat</b>	<b>23</b>
4.1 Elektrisk krets och komponentval . . . . .	23
4.1.1 Styrenehet . . . . .	24

4.1.2	Spänningsregulator . . . . .	24
4.1.3	Val av sensor . . . . .	24
4.1.4	Sensorer och montage . . . . .	25
4.1.5	Detektionssignal . . . . .	26
4.1.6	Justering . . . . .	27
4.1.7	Visualisering . . . . .	28
4.2	Programkod . . . . .	28
4.2.1	Setup . . . . .	28
4.2.2	Delfunktionen “measure_distance” . . . . .	29
4.2.3	Del Funktionen “button_interrupt” . . . . .	31
4.2.4	Huvudfunktionen . . . . .	32
4.3	Uppfylld kvantitativa mål . . . . .	33
<b>5</b>	<b>Diskussion</b>	<b>37</b>
5.1	Anpassning av mål och frågeställningar . . . . .	37
5.2	Storlek och typ av objekt . . . . .	37
5.3	Logik och mikrokontroller . . . . .	38
5.4	Anpassning och utvecklingspotential av mjukvaran . . . . .	38
5.5	Nedskalning och kostnadsbesparingar . . . . .	39
<b>6</b>	<b>Slutsats</b>	<b>41</b>
	<b>Litteraturförteckning</b>	<b>43</b>
<b>A</b>	<b>Kretsschema</b>	<b>I</b>
<b>B</b>	<b>Programkod</b>	<b>III</b>

# Figurer

2.1	<i>Illustration för hur ljudvågor färdas från sändare till mottagare när de studsar mot ett föremål.</i>	6
2.2	<i>Bild på en HC-SR04</i>	7
2.3	<i>Illustrerar hur Echo-pins utsignal varierar beroende på vilket avstånd som mäts.</i>	8
2.4	<i>Bild på en Raspberry Pi Pico.</i>	8
2.5	<i>Bild urklipp för hur en 7805-regulator ska kopplas upp.</i>	9
2.6	<i>Bild på en spänningsregulator 7805.</i>	10
2.7	<i>Visualiserar en elektrisk krets för ett låpassfilter</i>	10
2.8	<i>Visualisering av arbetsområdet sett från ovan</i>	12
3.1	<i>Testprotokoll från slutttest</i>	20
3.2	<i>Bild från slutttest då objektet befinner sig innanför arbetsområdet.</i>	21
4.1	<i>Bild på det slutgiltiga kretskortet</i>	24
4.2	<i>Bild över sensorernas detekteringsbredd i arbetsområdet.</i>	26
4.3	<i>Krets för detektionssignal till huvudkort.</i>	26
4.4	<i>Knapp med tillhörande RC-filter</i>	27
4.5	<i>LED kretsar.</i>	28
4.6	<i>Programkod för funktionen <code>measure distance</code>.</i>	30
4.7	<i>Programkod för interruptfunktion <code>button_interrupt</code>.</i>	31
4.8	<i>Bild på ett oscilloskop som visar periodtiden mellan varje ny mätning för en sensor som mäter 150 cm.</i>	33
4.9	<i>Bild på ett oscilloskop som visar periodtiden mellan varje ny mätning för en sensor som mäter 70 cm.</i>	34
4.10	<i>Protokoll från slutttest</i>	35



# Tabeller

2.1	Ljudvågors hastighet “c” i luft vid olika temperaturer. . . . .	6
3.1	Testprotokol sensorer. . . . .	17
4.1	Ifyllt testprotokol sensorer. . . . .	25



# 1

## Inledning

I samhället idag finns det flertalet tekniska lösningar som hjälper oss människor att leva säkrare. Ett exempel på en sådan lösning är till exempel hissdörrar som inte bara automatiskt stängs om någon står emellan, utan också öppnas igen för att undvika att någon kläms.

I denna rapport redovisas hur en sensorlösning kan konstrueras för att kunna detektera och skydda människor i ett begränsat arbetsområde, från att inte bli påkörda eller skadade av en rörlig enhet.

### 1.1 Bakgrund

Företaget har fått i uppdrag av en kund att se över en redan befintlig sensorlösning som har i uppgift att detektera objekt inom ett begränsat arbetsområde. Arbetsområdet som sensorlösningen vakar över är placerat bakom en automatöppnad grind. Grindens uppgift är att kontrollera in- och utpassage i en vanlig butik. Grinden fungerar på så sätt att den enbart släpper in personer från ett håll och detta för att butikspersonalen enkelt ska kunna kontrollera in- och utpassage i butiken.

Utmaningen med denna grind är att grinden öppnas inåt i butiken och om det befinner sig människor inom grindens arbetsområde riskerar de att bli träffade av grinden. Hela idén med sensorlösningen är att se till att grinden inte öppnas om människor befinner sig inom grindens arbetsområde. Utan denna sensorlösning kommer grinden alltså att kunna öppnas vid fel tillfälle och personer kommer då att kunna bli träffade och i värsta fall skadade.

Barn kan vara extra sårbara i denna typ av miljö. Detta för att barn i många fall inte har samma konsekvenstänk och inte förstår att grinden kommer att öppnas när personer är på väg in i butiken. Just denna grindtyp sitter också ofta i huvudhöjd för barn och slår den upp och träffar ett barns huvud vid fel tillfälle kan det leda till allvarliga skador. Denna aspekt påverkar både dimensioneringen av sensorns arbetsområde, men exempelvis även hur snabbt sensorn måste kunna utföra en detektion.

Andra applikationer som denna lösning skulle kunna appliceras på är till exempel andra typer av automatiska dörrar som inte ska kunna slå upp när människor befinner sig inom dess räckvidd. Säkerhetssystem för maskiner och robotar är också ett intressant område för denna typ av sensorlösning. Tidningen Automation [1], skriver

bland annat att användandet av robotar har ökat kraftigt i Sverige på senaste tiden och därför har arbetet med arbetsplatsolyckor varit utmanande. Enligt Arbetsmiljöverket [2], har industrin toppat listan för antalet dödsolyckor på arbetsplatsen under flera tidigare år, detta orsakat av bland annat olika maskiner. För att bidra till en säkrare industri som troligtvis kommer växa med fler robotar och maskiner framöver så kommer säkerhetssystem som är baserade på liknande sensorer som i detta arbete att behöva utvecklas. Denna aspekt skapar både ett ekonomiskt mervärde, för sådana typer av lösningar och en vilja från samhället att hålla nere arbetsplatsolyckor.

Just nu finns det redan en befintlig sensorlösning som har i uppgift att skydda människor från att inte bli träffade av grinden. Dock har det visat sig att sensorlösningen inte har varit helt pålitlig och att den inte har fungerat så som kunden hade önskat. Det har hänt att sensorlösningen både har varnat för saker som inte har funnits i arbetsområdet och att den har missat att detektera människor och slagit upp på dem. Därför har företaget nu startat upp ett projekt som går ut på att ta fram en helt ny lösning som är oberoende av den gamla. Detta projekt ska alltså skapa en alternativ lösning för att ge ett förslag på hur denna lösning skulle kunna se ut i framtiden.

## 1.2 Syfte

Huvudsyftet är att utveckla och presentera en sensorlösning som har i uppgift att skydda människor från att utsättas för fara genom att förhindra kollisioner med rörliga enheter. Fokus ligger i att göra sensorn så detektionssäker som möjligt genom att den ska täcka arbetsområdet i så stor utsträckning som möjligt. Detta medan aspekter som varierande temperaturer, olika ljud och ljusförhållanden kan förekomma, men bör ej påverka systemets tillförlitlighet.

Syftet är även att skapa en flexibel lösning som kan justeras utefter olika arbetsområden, men samtidigt lägga fokus på enkelhet och att hålla nere kostnader för de ingående komponenterna. Detta eftersom att det är eftersträvansvärt med låga produktionskostnader samtidigt som att den ska vara enkel att installera och att lösningen enklare ska kunna appliceras på olika enheter.

## 1.3 Mål

I detta avsnitt presenteras och beskrivs arbetets uppsatta mål. Det stora kvalitativa målet med detta arbete är att ta fram en prototyp på en sensorlösning som ska skydda människor från en rörlig enhet. Detta genom att sensorn detekterar människor eller objekt i människostorlek inom arbetsområdet. För att kunna uppnå detta kvalitativa huvudmål och företagets egna krav på sensorlösningen har även kvantitativa mål listats i en punktlista nedan:

- Minst 90% säkerhet att sensorlösningen detekterar när ett objekt av dimensioner 200 x 200 mm (X och Y-led) är inom arbetsområdet.

- Högst 10% falska detektioner, vilka kan uppstå exempelvis när objekt befinner sig strax utanför arbetsområdet eller pga störningar i omgivande miljö.
- Närvarodetektion av människa i arbetsområdet ska ske på högst 20 millisekunder.
- Arbetsområdet ska kunna justeras i intervallet 50 - 150 cm (X och Y-led) med minst fem steg.
- Sensorlösningens detektionsområde ska ha en undre gräns på högst 40cm i höjddled.
- Sensorlösningen ska kunna fungera med omgivningen inom temperaturintervallet 0–45 °C.
- Totalkostnad för funktionskritiska komponenter (funktionen kräver en specifik typ av komponent) och kretskort ska inte överstiga 150 kronor.

## 1.4 Avgränsningar

I detta avsnitt presenteras de avgränsningar som kommer att göras i detta arbete.

- Sensorlösningen används endast när den rörliga enheten är i viloposition i ena änden av arbetsområdet för att säkerställa att det är fritt fram att påbörja en rörelse. Detta eftersom det är en envägskommunikation från sensorlösningen till huvudkortet.
- Då sensorlösningen som tas fram ska fungera mer som ett förarbete behöver slutprodukten inte vara en helt färdigutvecklad lösning redo för serieproduktion.
- Sensorlösningen behöver inte skilja på människor och andra typer av objekt. Det är människor som är det viktiga och hur andra objekt upptäcks ska enligt företaget inte ligga i fokus för arbetet.
- Då detta är ett förbättringsprojekt för en redan befintlig sensorlösning behöver sensorlösningen därför anpassas för att uppfylla tidigare förutbestämda design. Med detta menas att båda placering av sensorer och hur de ska anslutas redan är bestämt.

## 1.5 Frågeställningar

I detta kapitel kommer arbetets fem frågeställningar att presenteras.

- Hur kan en sensorlösning konstrueras för att detektera objekt inom ett begränsat arbetsområde med fokus på att säkerställa människors skydd från en rörlig enhet?
- Hur många sensorer behövs för att kunna detektera hela arbetsområdet och kommer det fungera att kombinera flera sensorer eller kommer de att störa ut varandra?
- Vilka eller vilken sensortyp skulle kunna användas för att uppnå de kvantitativa målen listade i avsnitt 1.3?
- Hur kan en lösning byggas så att sensorlösningen enkelt ska kunna justeras och anpassas efter olika storlekar på arbetsområden? Kan logiken hanteras utan en programmerbar enhet?

## 1. Inledning

---

- Hur mycket kan ändringar i miljöförhållanden, i termer av ljud, ljus och temperatur, påverka sensorlösningens pålitlighet? Kan sådana förändringar tas hänsyn till i sensorlösningen på något sätt, utan att kompromissa med prestanda under normala miljöförhållanden?

# 2

## Teoretisk och teknisk bakgrund

I detta kapitel redovisas både den teoretiska bakgrunden för de komponenter som används i projektet men också den tekniskt specificerade bakgrunden som sensorlösningen behöver förhålla sig till.

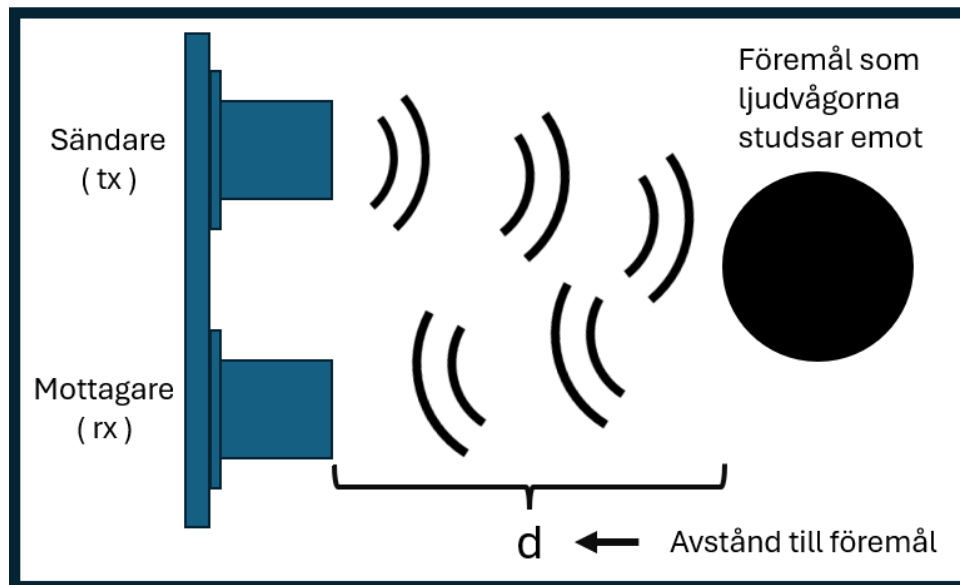
### 2.1 Teoretisk bakgrund

I detta avsnitt presenteras en teoretisk bakgrund av de fördjupade teknikerna och tekniska komponenterna som används i projektet.

#### 2.1.1 Uppkomst och funktion av ultraljudssensorn

År 1880 upptäckte de franska bröderna och forskarna Pierre Curie och Jacques Curie det elektriska fenomenet piezoelektriciteten [3]. De kom på att när ett piezoelektriskt material utsätts för mekanisk påverkan genererar det också elektricitet. Genom denna upptäckt har ingenjörer och forskare senare använt tekniken för att konstruera ultraljudssensorer som år 1917 uppfanns av Paul Langevin [4]. Detta var dock bara början av utvecklingen av ultraljudssensorn och vidare under första och andra världskriget utvecklades sensorerna mycket för att kunna detektera ubåtar.

Ultraljudsdetektering är en teknik som använts för att bestämma avståndet till ett objekt utan att fysiskt inte behöva röra vid objektet [5]. Tekniken bygger på att en sändare skickar ut högfrekventa ljudvågor som färdas mot objektet med ljudhastigheten “c”. När ljudvågorna sedan träffar objektet kommer de att studsas och samtidigt börja färdas tillbaka mot en mottagare som i sin tur kommer att ta emot ljudvågorna. I figur 2.1 nedan ses en teknisk modell på hur sändare, mottagare och föremål samspelar.



**Figur 2.1:** Illustration för hur ljudvågor färdas från sändare till mottagare när de studsar mot ett föremål.

Avståndet  $d$  till ett objekt kan avgöras med hjälp av ekvation 2.1 nedan. Till den behövs tiden när sändaren skickar ut ljudvågor  $t_{tx}$ , samt tiden när den tas emot igen,  $t_{rx}$ . Dessa två variabler används tillsammans med ljudets hastighet  $c$  för att sedan beräkna avståndet  $d$ .

$$d = \frac{(t_{rx} - t_{tx}) \cdot c}{2} \quad (2.1)$$

Ett sätt att ta reda på tiden är att använda en klocka och mäta tiden det tar för ljudvågorna att skickas från sändaren till mottagaren. Det görs genom att starta en klocka och mäta starttiden som  $t_{tx}$  när sändaren skickar ut ljudvågorna. Genom att sedan även mäta klockslaget då mottagaren tar emot ljudvågorna som  $t_{rx}$  kan tiden beräknas som  $t_{rx} - t_{tx}$  vilket också kan ses i ekvation 2.1. För att nu också kunna bestämma ljudhastigheten  $c$  på ljudvågorna behöver även temperaturen i luften bestämmas. Eftersom ju varmare luften är desto snabbare färdas ljudvågorna [6]. I tabell 2.1 nedan redovisas olika hastigheter för ljudvågor i luft, beroende på temperatur.

**Tabell 2.1:** Ljudvågors hastighet “ $c$ ” i luft vid olika temperaturer.

Temperatur [C°]	-30	0	5	10	15	20	25	30	35
$c$ [m/s]	312	331	334	337	340	343	346	349	352

Genom att nu använda formel 2.1 ovan samt rätt tid och rätt hastighet i denna formeln kan avståndet till ett föremål beräknas. För att formeln också ska fungera är det viktigt att sändare och mottagare sitter på samma avstånd till objektet. Anledningen till att formeln innehåller en division med två är för att ljudvågorna

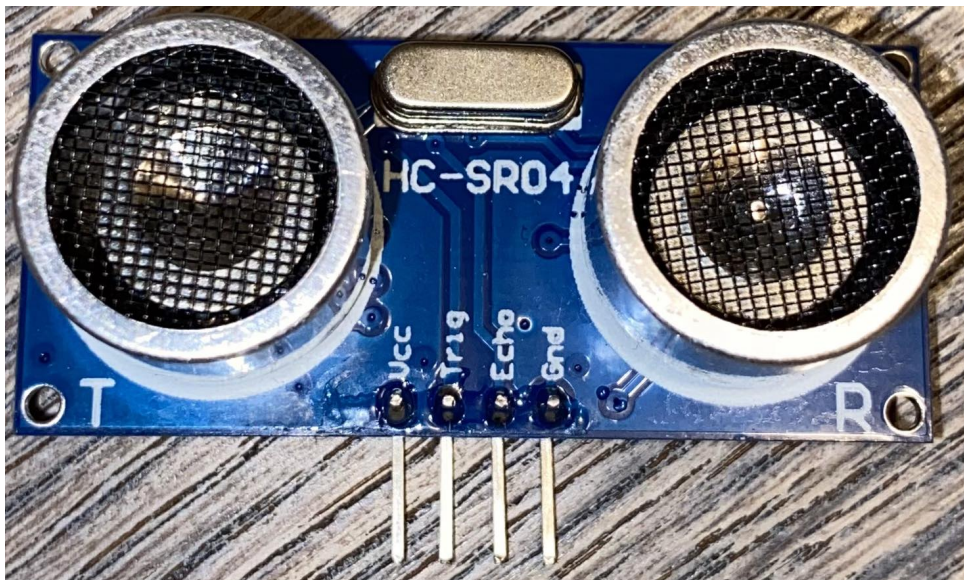
färdas både fram och tillbaka vid mätning och genom att dela på två halveras tiden och avståndet blir rätt.

### 2.1.2 Sensor HC-SR04

HC-SR04 är en sensor som med hjälp av ultraljud kan mäta avstånd från två till 400 cm, med en upplösning på tre mm. Sensorn drivs på fem volt och vid mätning drar den 15 mA [7]. Varje HC-SR04 har fyra pinnanslutningar: VCC, TRIG, ECHO och GND. För att kunna använda sensorn ska alla fyra pinnar anslutas enligt följande:

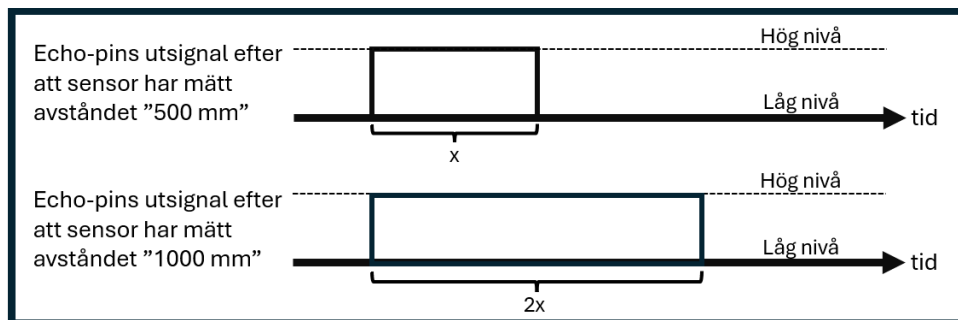
- VCC - 5V
- TRIG - Output (I/O-port)
- ECHO - Input (I/O-port)
- GND - Jord/Gnd

Här nedan i Figur 2.2 ses en bild på en HC-SR04



**Figur 2.2:** Bild på en HC-SR04

HC-SR04 ultraljudssensorn är uppbyggd av tre huvuddelar, en sändare, en mottagare och en kontrollkrets. När det är dags att mäta ett avstånd aktiveras sensorn genom att Trig-pinnen sätts till en hög nivå. Denna signal aktiverar sändaren att börja skicka ut ultraljudsvågor. De utskickade ultraljudsvågorna färdas genom luften tills att de träffar ett föremål och då börjar vågorna att studsas tillbaka mot sensorn. Mottagaren fångar då upp vågorna och med hjälp av den inbyggda kontrollkretsen på sensorn mäts den tid det har tagit för ljudvågorna att ha skickats från sändaren tills att mottagaren har tagit emot ljudvågorna. Den smarta kontrollkretsen sammanställer sedan tiden och sätter echo-pinnen hög, motsvarande den tid det har tagit för ljudvågorna att ha skickas från sändaren till mottagaren. I Figur 2.3 nedan visas hur echo-pinnens utsignal varierar beroende på vilket avstånd som har mätts.

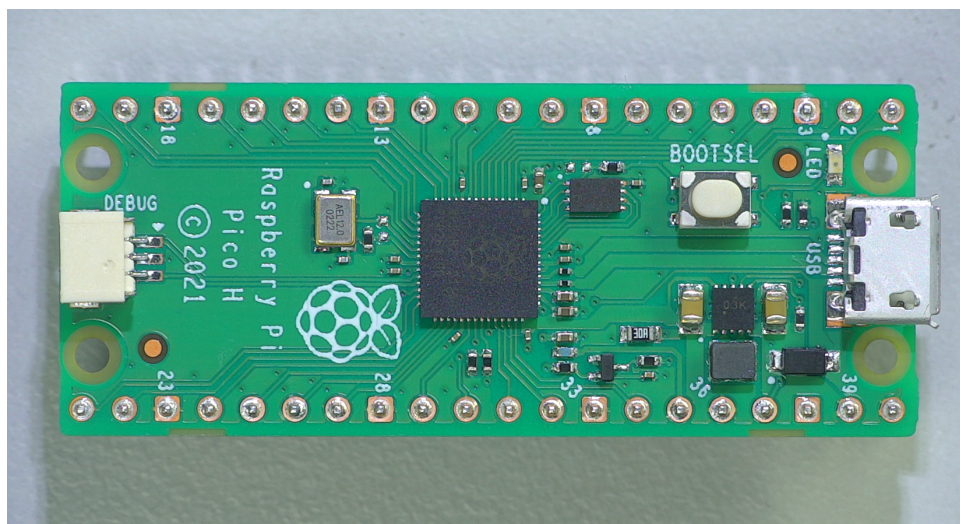


**Figur 2.3:** Illustrerar hur Echo-pins utsignal varierar beroende på vilket avstånd som mäts.

Alltså med hjälp av en mikrokontroller eller liknande kan nu tiden mätas från att Echo går hög tills att den går låg. För att sedan räkna fram rätt avstånd till föremålet, multipliceras tiden med ljudvågornas rätta hastighet som beror på temperaturen och sedan delas dess produkt på två och då har avståndet till föremålet beräknats med hjälp av sensorn.

### 2.1.3 Raspberry Pi Pico

Raspberry Pi enkortsdatorer utvecklas av Raspberry Pi Foundation [8]. Korten är främst utvecklade för att främja undervisning och hobbyister i området programmering och elektronik. Det har dock visat sig att dessa kort med fördel också fungerar bra i utvecklingsstadiet för produktutveckling hos teknikföretagen. I figur 2.4 nedan ses en bild på en Raspberry Pi Pico.



**Figur 2.4:** Bild på en Raspberry Pi Pico.

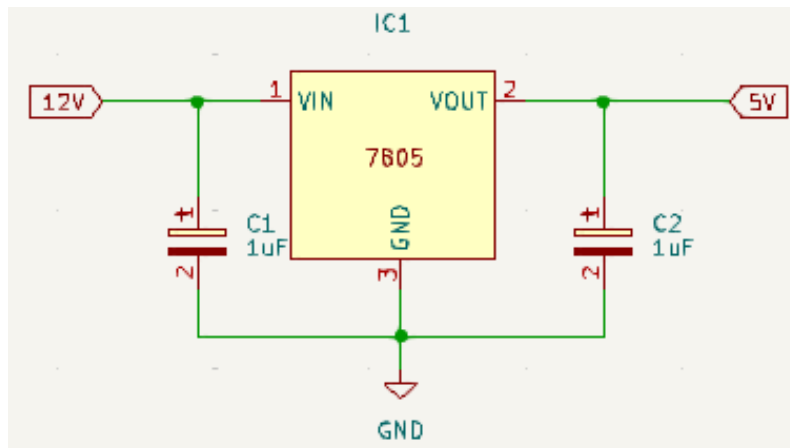
Raspberry Pi Pico är en enklare mer nedskalad modell som är mikrokontrollerbaserad [9]. Kortets kompakta form, låga strömförbrukning och låga pris ger utmärkta förutsättningar för mindre projekt inom elektronik och inbyggda system. Kortet innehåller en mikrokontroller (RP2040) som är baserad på ARM-Cortex-M0+ arkitektur med inbyggt minne, GPIO-pinnar, In och utgångar för 5V och 3,3V samt ett

flertal kommunikationsprotokoll. På kortet finns även till exempel utrustning som kristalloscillator för att generera en stabil klocksignal men även analog till digital-konverterare för att möjliggöra hantering av analoga signaler.

Raspberry Pi Pico har stöd för flertalet programmeringsspråk, där det officiella språket är Raspberry Pi Pico C/C++ SDK men enheten går även bra att programmera i programspråk som microPython och CircuitPython.

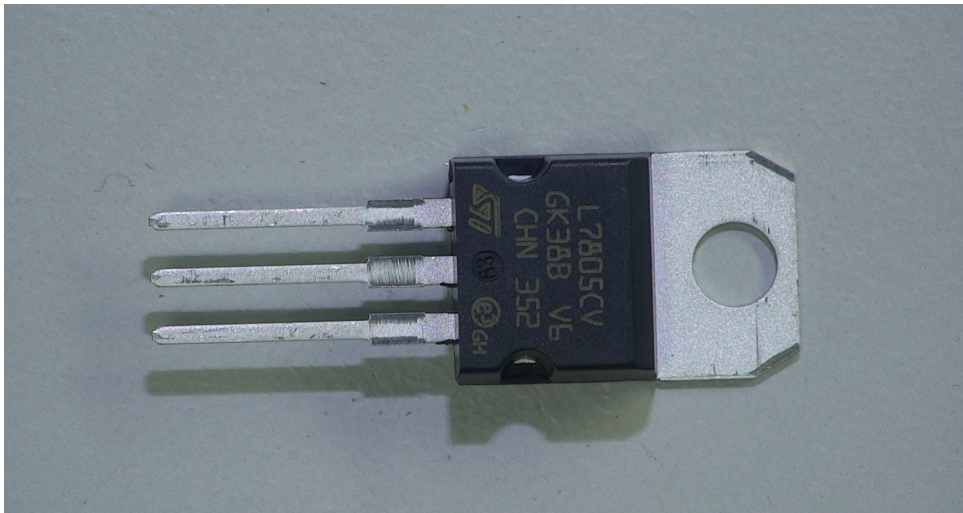
### 2.1.4 Spännings Växling (7805)

En 7805-spänningsregulator är en linjär regulator som säkrar 5 V ut-spänning och behöver minst 7 V, men tål upp till 35 V inspänning [10]. Regulatorn kan arbeta med en strömstyrka upp till 1 A men har en tendens att bli varm vid högre effekter och därför är den lämpligast att användas i applikationer med låg strömförbrukning. I Figur 2.5 nedan visas leverantörens rekommendation på hur den ska kopplas i en elektrisk krets.



**Figur 2.5:** Bild urklipp för hur en 7805-regulator ska kopplas upp.

7805-regulatorn är uppbyggd av transistorer, förstärkare och spänningsdelare för att styra flödet av ström genom enheten som bidrar till önskad spänning. Den ska enligt tillverkaren kopplas upp enligt Figur 2.5 ovan, parallellt med två elektrolytkondensatorer som har en kapacitans mellan 0,1 och 10,0  $\mu\text{F}$  för att hantera ostabila insignaler. I Figur 2.6 nedan ses en spänningsregulatorn 7805 i fysisk form.

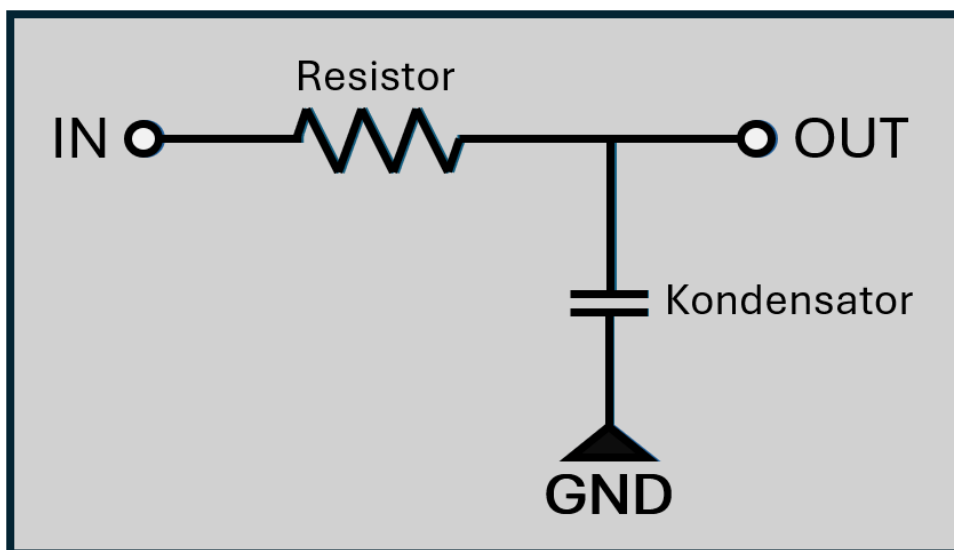


**Figur 2.6:** Bild på en spänningsregulator 7805.

### 2.1.5 RC-filter

Ett RC-filter är en elektrisk krets som består av en resistor  $R$  och en kondensator  $C$  som filtrerar elektriska signaler [11]. Det finns flera olika typer av RC-filter som: lågpasfilter, högpasfilter, bandpassfilter och bandspärrfilter. Vilket filter som är lämpligast att använda beror dock helt på vad som är filtrets uppgift. Ska filtret endast släppa igenom höga frekvenser är det till exempel ett högpasfilter som ska användas och tvärtom ett lågpasfilter för låga frekvenser.

Genom att använda ett RC-filter går det alltså att bestämma och styra vilka signaler som ska gå igenom filtret och det kan vara mycket användbart i många applikationer. I Figur 2.7 nedan visualiseras hur ett RC-filter kan se ut och med denna uppsättning av resistorn och kondensatorn skapas ett lågpasfilter.



**Figur 2.7:** Visualiserar en elektrisk krets för ett lågpasfilter

För att sedan kunna bestämma vilka frekvenser som ska kunna gå igenom ett lågpasfilter behöver både värden för resistor och kondensatorn bestämmas. Denna gräns för frekvenser kallas Cut-off frekvens  $f_{-c}$  och kan bestämmas med hjälp av ekvation 2.2 nedan.

$$f_{-c} = \frac{1}{2\pi RC} \quad (2.2)$$

Till exempel om en resistor på  $100 \Omega$  och en kondensator på  $1\text{F}$  bildar ett lågpasfilter med en gränshfrekvens på cirka  $1592 \text{ Hz}$ . Vilket innebär att filtret då dämpar alla frekvenser över  $1592 \text{ Hz}$  och släpper igenom frekvenser under  $1592 \text{ Hz}$ .

### 2.1.6 PCB och PCB-design (KiCad)

En PCB (printed circuit board) är ett kretskort konstruerat utifrån en cad (Computer-Aided Design) [12]. En PCB-cad är uppbyggd av fotavtryck representerande elektriska komponenters fästytter, kopparnät, fyllda zoner samt andra elektriska förbindelser som kopplar dessa komponenter samman.

PCB-design processen börjar med att välja ut och designa symboler samt fotavtryck för de komponenter som ska ingå i designen. Symbolerna krävs för att sedan kunna bygga upp ett kretsschema. I detta schema plottas komponentsymbolerna ut där rätt fotavtryck kopplas till rätt symbol som sedan binds samman till en elektrisk krets. Utifrån detta kan ett grunddesign för mönsterkortet genereras.

I mönsterkortsredigeraren kan sedan komponenterna flyttas runt till de fysiska positioner som de elektriska komponenterna ska sitta och sedan bindas ihop med kopparnät och andra elektriska förbindelser av önskad design. I detta verktyg kan även till exempel tjocklek på kopparledningarna ställas in. I verktyget finns även ett inbyggt koordinatsystem som gör det enkelt att ställa in korrekta avstånd mellan komponenter.

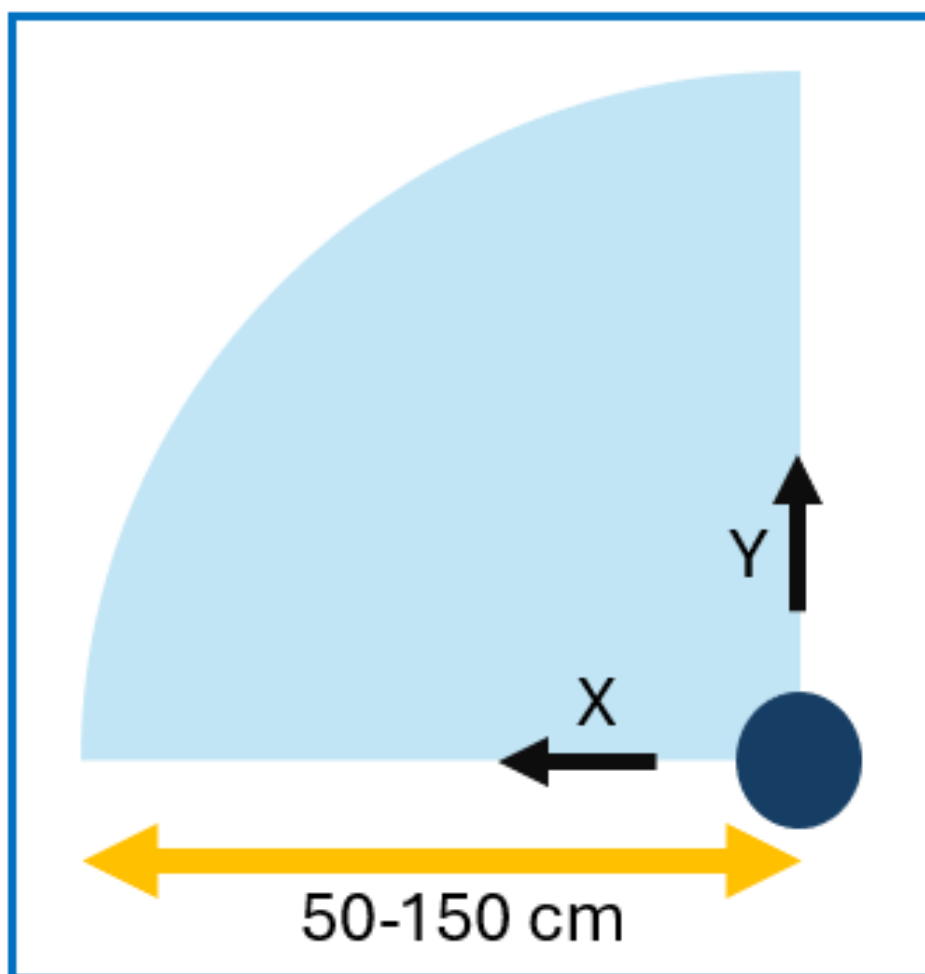
Priset på ett kretskort kan variera mellan några upp till tusentals kronor, där prisskillnaden för enskilda kort främst ligger i storlek, antal lager, samt vilken prestanda och teknologi kortet kräver [13]. Den stora prisskillnaden kommer dock när produktionen av ett kort kan skalas upp. Detta eftersom att PCB:er har stora fasta kostnader i jämförelse med själva produktionen av kortet som till exempel kostnaden för arbetet med maskininställning. När fler PCB:er beställs kan dessa fasta kostnader fördelas över fler kort och minska styckepriset markant.

## 2.2 Tekniska specifikationer för projektet

I detta kapitel förklaras de tekniska ingångsvärden som tilldelades inför projekt.

### 2.2.1 Arbetsområdet

Arbetsområdet definierar det område som sensorlösningen har i uppgift att läsa av, vilket är baserat på den rörliga enhetens längd och räckvidd. Den rörliga enheten kommer endast att kunna göra en svepande rörelse om  $90^\circ$ , där det begränsade arbetsområdet är visualiserat som skuggat område i figur 2.8 nedan. Storleken på arbetsområdet ska kunna bestämmas av storleken på den rörliga enheten och detta gör att detektionsområdet måste kunna justeras för att passa till olika applikationer. Sensorlösningen ska även i höjdlid kunna klara av att detektera objekt minst 40 cm från golv.



**Figur 2.8:** Visualisering av arbetsområdet sett från ovan

Sensorlösningen och den rörliga enheten kommer båda att vara fixerade och monterade på samma position i x och y-led vilket är illustrerat med mörkblå cirkel i figur 2.8 ovan. Sensorlösningen kommer inte vara fastmonterad på den rörliga enheten och därför kommer den rörliga enhetens rörelse inte förflytta sensorlösningen.

## 2.2.2 Elektriskt interface

Den elektriska anslutningen till sensorlösningen består av tre poler. Den första polen är strömförsörjning på 12 V, andra polen är jordanslutning och tredje polen är en detektionssignal som går till huvudkortet. Detta kort är redan en färdig lösning och kan ses som ett moderkort för samtliga funktioner för den rörliga enheten och styr bland annat start- och stoppfunktion. På huvudkortets ingångsben för detektionssignalen finns en pull-up-resistor på 4.7 k $\Omega$ . Denna resistor ska säkerställa en logisk hög signal när ingen aktiv sensorsignal är närvarande, vilket är en viktig parameter i signal-logiken som förklaras mer ingående i kommande avsnitt.

## 2.2.3 Signal logik

Mottagarpinnen på huvudkortet kommer i normalt läge vara konstant hög, vilket innebär en spänning på upp till 12 V. Att mottagarpinnen är hög ska för huvudkortet betyda att det inte finns något hinder i arbetsområdet. Sensorlösningens uppgift är att sätta mottagarpinnen till låg när ett hinder upptäcks inom arbetsområdet. Att sätta pinnen till låg betyder att dra denna utspänning på pinnen ner till jord. Vid en fri situation utan hinder ska detektionssignalen återgå till sitt normala läge (hög signal). Alltså är sensorlösningens enda uppgift att dra ner pinnens spänning till jord (när hinder upptäcks) oavsett vilken spänning som kommer ut. Den information som finns är att det kan vara en spänning på max 12 V och därför måste sensorlösningen kunna hantera spänningar upp till denna nivå.

När sensorer är baserade på ljud eller ljus, kan det förekomma viss typ av störning. Speciellt när de används i miljöer med mycket reflekterande ljus eller högljudda och bullriga miljöer. Sensorlösningen bör vid behov ha någon form av filtrering för att filtrera bort dessa signaler. Om denna filtrering behövs ska den anpassas efter vilket media som sensorn arbetar inom och målet är att optimera sensorlösningens och motverka falska detektioner.



# 3

## Metod

### 3.1 Arbetsgång

För att säkerställa ett strukturerat och välplanerat projekt, delades arbetet upp i fem kronologiska steg. Det första steget var att planera och på ett hypotetiskt sätt upprätta en tidsplan som enkelt redovisade aktiviteter vecka för vecka. Med tidsplanen gjord fanns det en tydlig röd tråd genom arbetet och samtidigt ökade tidsplanen chansen att inget moment glömdes bort eller tog för lång tid, för att hinna klart med arbetet innan deadline.

Steg två bestod av att försöka hitta, testa och jämföra olika sensortyper och försöka få en bild av vilka sensorer som skulle kunna vara tänkbara och intressanta för just detta arbete. Efter noggrann informationsinhämtning och diskussioner med experter från Combitech kunde fem olika sensorer beställas. När sensorerna sedan kom hem inleddes en intensiv fas av testning där sensorernas prestanda utförligt dokumenterades och sammanställdes. Med hjälp av allt testande och resonerande kring testningen fanns det egentligen en klar bild av vilken sensor som skulle väljas att gå vidare med.

Det tredje steget handlade om att forma och bygga hela sensorlösningen kring den valda sensorn och samtidigt integrera det tillsammans med moderkortet. Detta steg initierade en rad av arbetsmoment som bestod av att designa elschema, kretskort och styrenhet men även besluta om hur många sensorer som behövdes samt dess position. Men efter mycket jobb med detta kunde kretskort slutligen beställas tillsammans med en tillhörande lista med elkomponenter. Nu fanns också en detaljerad plan för hur slutprodukten skulle se ut och därmed kunde steget avslutas.

Steg fyra började egentligen först när kretskortet och elkomponenterna kom hem för då påbörjades montering och de komponenter som skulle lödas löddes fast på kretskortet. Med ett färdigt kretskort i handen påbörjades testing av kretskort och parallellt fin justerades programkoden för att fungera så effektivt som möjligt. Här började det även planeras för sluttestet. Det sista steget, steg fem handlade främst om att strukturera ihop all dokumentation i en projektrapport och försöka knyta ihop hela arbetet. Parallellt med detta hölls även det sista sluttestet där hela sensorlösningens funktion granskades och bedömdes. Sluttestet visade även konkret vilka mål som blev uppfyllda och i vilken grad. Efter sluttestet hölls en slutlig presentation av arbetet och med presentationen gjord avslutades arbetet.

### 3.2 Tillvägagångssätt

I detta avsnitt kommer arbetsstegen i kapitel 3.1 mer konkret att presenteras.

#### 3.2.1 Förstudie

För att inte behöva testa allt för många sensorer men fortfarande se till att relevanta komponenter testas genomfördes en enklare förstudie. Efter förstudien (informationsinhämtning av sensorer) var målet att köpa hem och testa de bäst lämpade sensorerna utifrån den insamlade informationen. Tanken med den korta förstudien var att effektivisera testprocessen och möjliggöra att det senare sensortestet endast innehöll 3-5 olika sensorer. Detta så att inte allt för många sensorer behövde köpas in och testas. Förstudien inleddes med att undersöka de vanligaste sensortyperna för detektering och avståndsbedömning. Parametrar som spridningsvinkel och avstånd var viktiga för att få en uppfattning för hur de olika sensorerna skulle kunna täcka ett arbetsområde. Att även undersöka vilka fördelar och nackdelar de olika sensortyperna har, var en viktig del i denna fas. När de mest intressanta typerna av sensorer hade sällats fram användes Combitechs leverantörers hemsidor för att lokalisera datablad för de mest passande komponenterna som sedan kunde beställas.

#### 3.2.2 Testning av sensorer

För att ha möjlighet att testa och jämföra de olika sensorerna användes ett enklare utvecklingskort med mikrokontroller (Raspberry Pi Pico) där alla sensorer kunde anslutas. De sensorer som testades ses i punktlistan nedan:

- VL53L0X Time-of-flight sensor
- VL53L1X Time-of-flight sensor
- AM312 PIR (Passive Infrared Sensor)
- HC-SR04 Ultraljudsensor
- US-016 Ultraljudsensor

Combitechs labb och utrustning i Jönköping användes för att bygga upp och skapa ett 90-gradigt arbetsområde. De enklare tester som gjordes är listade i tabell 3.1 nedan.

**Tabell 3.1:** Testprotokol sensorer.

Sensor	A	B	C	D	E
Typ	Ljus	Ljus	Ljud	Ljud	Ljus
Modell	VL53L0X	VL53L1X	HC-SR04	US-016	AM312
Analog/Digital	Digital	Digital	Digital	Analog	Digital
Maxlängd sensor mäter [mm]					
Stabil och korrekt mätning:					
500mm					
1000mm					
1500mm					
Spridningsvinkel [grader]					
Kostnad/st [sek]					

Arbetsområdet delades in i tre olika zoner: 0-500 mm, 0-1000 mm och 0-1500 mm och här studerades sensorernas förmåga att mäta stabilt och korrekt i de olika avstånden. En stabil detektering betydde att sensorn kunde upptäcka och ge ett stabilt värde (avstånd) till ett objekt på 200x200 mm inom sensorns synfält. Spridningsvinkeln mättes på liknande sätt men här fördes objektet in från sidorna och när sensorerna fått ett stabilt mätvärde till objektet markerades vart objektet befann sig. När detta skett från båda håll och med olika avstånd från sensorn kunde markeringarna mätas ut och en spridningsvinkel beräknas. I tabellen ingår även kostnad för de olika sensorerna eftersom detta var en viktig faktor vid jämförelsen av de olika sensorerna.

Enklare tester gjordes även för att se hur de olika sensorerna påverkades om de utsattes för variationer i det medie som den specifika sensorn använde sig av. De mer specifika testerna gjordes eftersom en sensor kan bete sig optimalt i ett testrum som har till exempel stabilt ljud och ljus men när den kommer ut i en mer varierad miljö börjar den ge ut ojämna värden. Det främsta testet av denna typ var att sätta två likadana sensorer bredvid varandra och se om dessa kunde störa ut varandra på något sätt. Ett exempel på detta var att sätta två stycken ultraljudsensorer mot varandra och se om det kunde skapa konstiga värden utifrån detta, då båda sensorerna använder ljud med samma frekvens. Om så inte var fallet kunde det skapa en trygghet om att inte heller ljud från andra frekvenser skulle bidra till störningar av något slag.

Det var även viktigt att kontrollera hur sensorn och sensortypen reagerade och detekterade olika objekt med olika former, färger och temperatur. Detta både ur ett etiskt perspektiv då sensorlösningen ska upptäcka alla olika typer av människor men också ur ett rent funktionellt perspektiv, hur de detektera olika typer av objekt som en människa kan tänkas ha med sig in i arbetsområdet.

Genom att sedan testa sensorerna enskilt och kombinerat med varandra i olika uppsättningar diskuterades för- och nackdelar. Resultatet av detta presenterades sedan för Combitech som godkände och gav klartecken för att arbetet kunde fortsätta med en vald sensor.

### 3.2.3 Elschema och PCB-design

Efter valet av sensortyp påbörjades arbetet med den elektriska krets som skulle kunna styra och läsa av de valda sensorerna, bearbeta den inkommande informationen samt kommunicera till huvudkortet om något befinner sig i arbetsområdet. För att uppfylla de uppsatta målen fanns det krav på att kretsen utrustades med någon form av spänningsväxling, detta för att växla ner inkommande 12V till den spänning som de andra ingående komponenterna behöver. Det behövdes även ett användarvänligt sätt att justera arbetsområdet samt enkla lösningar för att se hur de olika funktionerna fungerar utan att behöva koppla in kortet till en dator. Tanken var att lösa detta med tryckknappar för justering samt lampor för indikation och visualisering av de olika funktionerna.

Under projektets gång användes enklare simuleringsprogram som till exempel LTSpice. Detta för att kunna visualisera och testa det elektriska systemet innan den riktiga kretsen fanns på plats. Med detta verktyg gavs det även möjlighet att testa många specifika komponenter och om de komponenter vi valt ut gav den utsignalen som beräknats.

Som tidigare nämnt behövdes någon form av mikrokontroller för att få sensorerna att fungera korrekt. Det var därför också något som behövdes inkluderas i elschema och i förlängningen placeras ut på kortet. Eftersom projektets mål var att skapa ett kort med en sensorlösning och fokuset låg på att få denna att fungera så bra så möjligt, så togs beslutet ganska tidigt om att använda ett mindre så kallat utvecklingskort istället för att bygga upp ett helt nytt eget microcontroller-kort. Här stod det främst mellan modeller som Arduino, Raspberry Pi eller andra kort med liknande egenskaper. Dessa kan programmeras i till exempel python och C/C++.

När beslut tagits om sensoralternativ, komponenter till elschema och styrenhet var det dags för själva designen av kretskortet (PCB:n). Här användes programmet KiCad, där först samtliga komponenter skapades i symbolbibliotek samt fotavtrycksbibliotek. När samtliga komponenter var skapade kunde elschema ritas upp i programmet. Cad projektet avslutades med att en PCB-design genererades utifrån elschema och sedan korrigerades efter önskad design på det fysiska kortet.

Från KiCad genererades sedan beställningsfiler för PCB som tillsammans med en komponentlista för önskade komponenter skickades till Combitech för vidare beställningar av fem stycken kort från företagets leverantörer.

### 3.2.4 Programmering

När valet blev att använda en programmerbar mikrokontroller integrerat i sensorlösningen har uppbyggnaden av sensorlösningen mycket handlat om att använda dess smarta funktioner så mycket som möjligt. Detta har i sin tur gjort att programkoden har fått en mycket central del i hela lösningen.

De funktioner som behövdes från mikrokontroller och därmed även från program-

met var lösningar för att skicka och ta emot information från sensorn, men även något som hanterar den information som kommer in. Det behövdes även kod för hantering av olika sorters visualisering samt justering av arbetsområde kopplat till Raspberryns I/O-portar. För att sedan få programmet komplett behövdes en sammanställning av indata samt ett kommando för att skicka en detektionsignal på en utport kopplad till huvudkortet. Det krävdes även små extra tillägg i koden för att till exempel minska påverkan från enstaka felaktiga mätvärden från sensorerna och därmed öka säkerheten för detektionssignalen.

### 3.2.5 Assembly och Sluttest

Efter att den beställda PCB:n levererats, användes Combitech lab för att för montering (lödning) av samtliga komponenter. Detta följdes av enklare funktionstester samt mindre modifieringar av delar som inte blivit riktigt som planerat. De fundamentala tester som utfördes under denna fas var följande.

- Se att sensorlösningen kunde anslutas till 12 volt och att den kunde fungera utan att använda en tillhörande dator.
- Att samtlig elektronik på sensorlösningen fungerade enligt plan som till exempel den del som utvecklats för justering av arbetsområde eller de delar som utvecklats för enklare visualisering av sensorlösningens olika funktioner.
- Att sensorlösningen ledde ner mottagarpinnens ström i jord vid detektion.

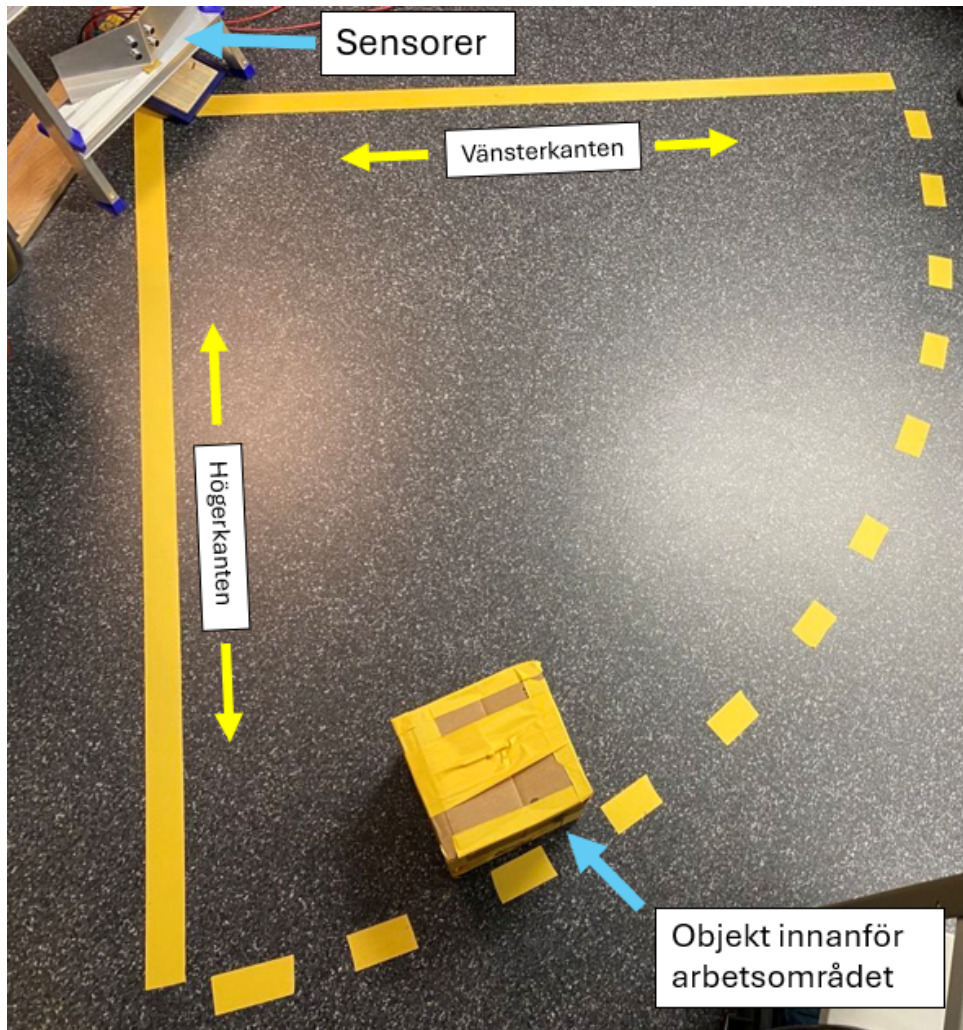
Ett sluttest utformades sedan för att undersöka hur sensorlösningen stod sig i förhållande till målen för en detektering med minst 90% säkerhet och högst 10% felde-  
tektering vid olika sorters störningar. Denna testdata har även hjälpt till att svara på några frågeställningar som ställts inför projektet. Här i figur 3.1 nedan ses det testprotokoll som har använts vid sluttestet.

### 3. Metod

Minst 90% säkerhet att sensorlösningen detekterar när ett objekt av dimensioner 200 x 200 mm (X och Y-led) är inom arbetsområdet.		
Föremål kommer in i arbetsområdet från vänster		
Detekteras:	X/30	X= <span style="background-color: yellow;"> </span>
Föremål kommer in i arbetsområdet från höger		
Detekteras:	Y/30	Y= <span style="background-color: yellow;"> </span>
Föremål kommer in i arbetsområdet rakt mot sensorn (från cirkelbågen)		
Detekteras:	Z/30	Z= <span style="background-color: yellow;"> </span>
Detektionssäkerhet i procent:	<b>? %</b>	<b>&lt;-- Över 90% är målet uppfyllt</b>
Högst 10% falska detektioner, vilka kan uppstå exempelvis när objekt befinner sig strax utanför arbetsområdet eller pga störningar i omgivande miljö som till exempel ljud eller ljus.		
Placerar föremål utanför arbetsområdet på väster sida		
Detekteras:	X/30	X= <span style="background-color: yellow;"> </span>
Placerar föremål utanför arbetsområdet på höger sida		
Detekteras:	Y/30	Y= <span style="background-color: yellow;"> </span>
Placerar föremål utanför arbetsområdet mitt emot sensorerna		
Detekteras:	Z/30	Z= <span style="background-color: yellow;"> </span>
Falskdetektion i procent	<b>? %</b>	<b>&lt;-- Under 10% är målet uppfyllt</b>

Figur 3.1: Testprotokoll från sluttest

Hur dessa mål uppfylldes, kontrollerades med testprotokollet i figur 3.1 ovan. För att testa sensorlösningen i säker detektering placerades ett objekt på 200 x 200 mm längs kanterna samt i mitten på arbetsområdet. I figur 3.2 nedan ses en bild från sluttestet där objektet placerades i arbetsområdet.



**Figur 3.2:** Bild från slutttest då objektet befinner sig innanför arbetsområdet.

I figur 3.2 ovan ses hur objektet är placerat på insidan av arbetsområdet vid en av mätningarna. Objektet placerades alltså på 30 olika distanser mitt i arbetsområdet följt av 30 mätningar på högerkanten och 30 mätningar på vänsterkanten. Här fördelades de 30 mätningarna jämnt från startpunkten bara några centimeter ifrån sensorerna till max distansen på arbetsområdet (150cm). En mätning ansågs som säker när objektet stått på mätpunkten tio sekunder utan att ge någon felsignal (att kortet inte indikerar för detektion). För att sedan testa lösningen för falska detektioner gjordes liknande tester men här placerades objektet precis utanför arbetsområdet. Två tester där objektet placerades ut på höger och vänster sida och ett test när objektet placerades utefter den bortsidan av arbetsområdet. Här betydde istället en detektion inom tio sekunder på en mätpunkt utanför arbetsområdet att mätningen blivit falsk.

Med hjälp av detta slutttest blev det sedan lätt att validera funktioner och delar av de uppsatta målen som sensorlösningen lyckats uppfylla.

### 3.3 Informationsinsamling

Flera metoder för att samla information har använts för att bidra med teknisk information och vägledning i olika beslut under projektets gång.

En detaljerad genomgång av de datablad som finns tillhanda för de olika komponenterna har genomförts. Detta gav en bild över vilka specifikationer, begränsningar samt möjligheter som finns med de olika komponenterna. Även akademiska texter har använts för att få en djupare förståelse över vilka tekniker de valda komponenterna använder sig av.

För att understödja dessa processer har experter på området både från Combi-tech och från Chalmers rådfrågats. Detta har bidragit med både nya perspektiv och nyttig information men också viktig erfarenhet av människor som arbetat med liknande projekt innan.

Den fysiska testningen som genomförts löpande under hela projektet kompletterar och verifierar datainsamlingen. Där information som inte kan hittas i nämnda metoder ovan helt enkelt kan testas fram i labbet. Detta säkerställer att projektet byggs upp både på såväl teoretiska som praktiska grunder.

# 4

## Resultat

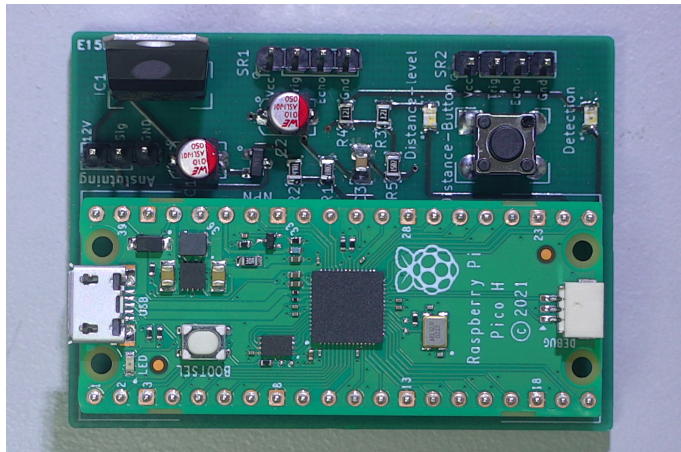
Resultatet av detta projektet är en fungerande sensorlösning. Sensorlösningen detekterar människovistelse innanför arbetsområdet, vilket leder till att sensorlösningen blockerar grinden från rörelse. Därmed skyddas människor från att inte bli påkörda eller skadade vilket gör att det stora kvalitativa målet med hela projektet, att skydda människor från en motoriserad rörlig enhet, är uppfyllt. Vidare i detta kapitel kommer det att presenteras hur sensorlösningen är uppbyggd och hur den fungerar. Samtidigt kommer även uppfyllnadsgraden av de kvantitativa målen att redovisas.

### 4.1 Elektrisk krets och komponentval

Det mest ingående komponenterna i det slutgiltiga kretsschemat består av en Raspberry Pi Pico, en linjär spänningsregulator av modellen 7805, två HC-SR04 ultraljudssensorer samt en transistor krets för att hantera detektionssignalen. Kretsschemat innehåller även en knapp samt två dioder för enkel justering av arbetsområdet samt visualisering av de olika funktionerna i systemet. För att enkelt kunna koppla in de yttre signalerna för detektion och strömförsörjning används vanliga stiftlistor. Nedan följer en komponentlista innehållande samtliga komponenter på sensorlösningen:

- 1st Raspberry Pi Pico SC0917
- 2st Digital ultraljudssensor HC-SR04
- 1st Tryckknapp ADTS81KV
- 1st LED-Detection RED 150080RS75000
- 1st LED-Blink BLUE 150080BS75000
- 1st Transistor MMBT3904-TP
- 2st 120Ohm Res CR0805-JW-121ELF
- 1st 4,7kOhm Res CR0805-FX-4701ELF
- 1st 10kOhm Res CR0805-FX-1002ELF
- 1st 15kOhm Res CR0805-FX-1502ELF
- 1st Spänningsregulator L7805CV
- 2st Kondensator 865080640001
- 1st Konsator C0805C153K5RAC7800
- 2st Stiftlist Hona 20-pin 2,54mm PPTC201LFBN-RC
- 1st Stiftlist Hane 16-pin 2,54mm PH1-16-UA

I figur 4.1 nedan ses det slutgiltiga kretskortet med alla komponenter där HC-SR04-sensorerna ska kopplas på SR1 och SR2 anslutningarna. Det fullständiga kretsschemat hittas i bilaga A.



**Figur 4.1:** Bild på det slutgiltiga kretskortet

### 4.1.1 Styrenehet

Stora delar av sensorlösningen baseras på en Raspberry Pi Pico och dess microcontroller RP2040. Dessa med hjälp av programkoden styr sensorernas funktioner tillsammans med justering av arbetsområde, visualisering vid justering och detektion samt den signal som uppmärksammar huvudkortet när sensorerna upptäcker något i arbetsområdet. Allt detta sker med hjälp Raspberryns I/O-portar.

### 4.1.2 Spänningsregulator

För att lösa funktionen att växla ner inkommande 12V till den spänning som komponenterna på kortet behöver, används en L7805CV. Denna krets har kopplats parallellt med två elektrolytkondensatorer på 1uF säkerställer en stabil spänning ut på kortet som är 5V.

### 4.1.3 Val av sensor

Det som helt avgjorde vilken sensortyp och sensor som valdes att bygga den slutgiltiga sensorlösningen på var den sensor som bäst tillfredställer stabila mätvärden med en stor spridningsvinkel och som även kan köpas till ett lågt pris. Fem sensorer testades och snabbt visade det sig att den passiva infraröda sensorn AM312 inte kommer kunna användas då den enbart signalerar hög eller låg signal och att avståndet den mäter på inte går att bestämma. Vidare i tabell 4.1 nedan visas en sammanställning över de övriga sensorernas testresultat och deras tillhörande pris.

**Tabell 4.1:** Ifyllt testprotokol sensorer.

Sensor	A	B	C	D
Typ	Ljus	Ljus	Ljud	Ljud
Modell	VL53L0X	VL53L1X	HC-SR04	US-016
Analog/Digital	Digital	Digital	Digital	Analog
Maxlängd sensor mäter [mm]:	2000	1500	1800	1900
Stabil och korrekt mätning:				
500mm	Ja	Ja	Ja	Ja
1000mm	Nej	Nej	Ja	Ja
1500mm	Nej	Nej	Ja	Ja
Spridningsvinkel [grader]	16	15,5	45	55
Kostnad/st [sek]	60-80	76-100	22	70

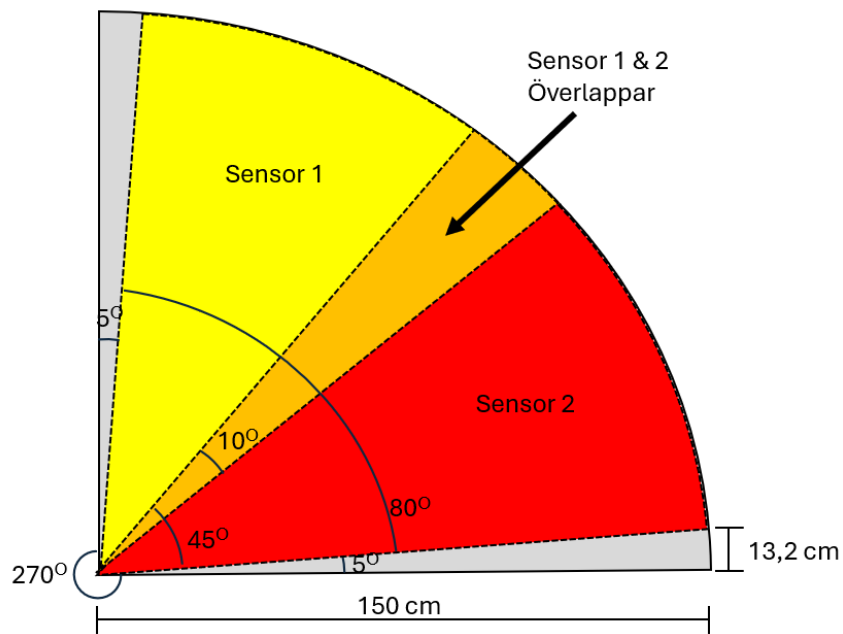
Genom att studera testresultaten i tabell 4.1 ovan bevisas det att ultraljud-sensorerna C och D helt klart uppvisar bäst resultat för ändamålet. Båda sensorerna uppfyller stabil och korrekt mätning i hela arbetsområdet upp till 150 cm, samtidigt som en stor spridningsvinkel bibehålls. Redan med detta resonemang kunde beslutet tas om att sensorlösningen skulle baseras på sensortypen ultraljud. I och med detta skulle sensorlösningen behöva bestå av minst två stycken sensorer. Eftersom en spridningsvinkel på 45 eller 55 grader inte skulle räcka för att täcka hela arbetsområdet på 90 grader. Sensorerna klarade sig även bra i de djupgående tester som gjordes med att variera den miljö som sensorn befann sig i. Sensorerna påverkades inte när försök gjordes på att störa ut den med hjälp av andra ultraljudsensorer. Inte heller olika typer av golv/underlag fick något speciellt utslag på sensorerna.

Båda ultraljud-sensorerna visade sig vara likvärdiga ur de flesta synpunkter, där sensorerna troligtvis använder liknande sensorelement (sändare och mottagare). Beslutet baserades till sist främst på priset för sensorerna, där sensor C var 48 kr billigare än sensor D och med två sensorer (oavsett modell) skulle denna skillnaden i pris bli 96 kr vilket är en stor del utav den kostnad i det kostnads mål som sattes upp inför projektet. Detta gjorde att valet föll på att gå vidare med sensor C (HC-SR04).

#### 4.1.4 Sensorer och montage

De sensorer som används i den slutgiltiga lösningen är alltså två stycken HC-SR04 ultraljudsensorer. Dessa styrs och hanteras genom att Raspberryn skickar ut signaler till TRIG och tar emot signaler från ECHO via två av dennes I/O-portar.

Enligt testerna kan en sensors spridningsvinkel för ljudvågorna säkerställas på 45 grader upp till minst 150 cm. I och med detta är en rimlig lösning att montera två sensorer med 10 graders överlapp och fortfarande säkerställa en sammanställd spridningsvinkel på 80 grader vilket lämnar 5 grader fritt på varje sida. Sensorernas täckning beskrivs i bild 4.2 nedan.

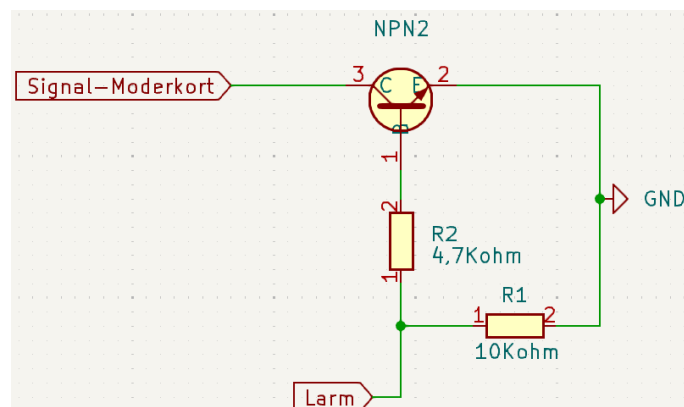


**Figur 4.2:** Bild över sensorernas detekteringsbredd i arbetsområdet.

Denna uppsättning säkerställer att objekt på 200x200mm fortfarande kommer att detekteras när det befinner sig inom det 90-gradiga arbetsområdet på den längsta distansen (150cm). Detta trots att 5 grader lämnats i ytterkant eftersom det i så fall skulle resultera i ett avstånd 132 mm på distansen 150 cm vilket är mindre än kravet på 200 mm-objekt.

#### 4.1.5 Detektionssignal

För att sätta mottagarepinne (Signal-Moderkort) på huvudkortet till låg när något befinner sig i arbetsområdet används en NPN transistor (MMBT3904-TP) tillsammans med kretsen i figur 4.3 nedan.



**Figur 4.3:** Krets för detektionssignal till huvudkort.

Etiketten Larm i figur 4.3 ovan illustrerar I/O porten som styrs av mikrokontrollern och vidare vid detektion sätts den till hög, dvs porten sätts till 3.3 V. När detta

sker skickas det då en basström till transistorn och kollektorströmmen börjar flyta. För att nu begränsa basströmmen sitter det en resistor på  $4,7\text{ k}\Omega$  som gör att basströmmen endast blir ca:  $0,53\text{ mA}$ . Den valda transistorn har en hfe-faktor på minst 70 vilket innebär att kollektorströmmen inte kan överstiga ca:  $37\text{ mA}$ . Detta är dock inget problem eftersom kollektorströmmen som ska ledas ner i jord inte kommer att överstiga  $2,6\text{ mA}$ . Att kollektorströmmen inte kan överstiga  $2,6\text{ mA}$  beror på den inbyggda pullup resistorn på  $4,7\text{ k}\Omega$  tillsammans med spänningen från detektionsbenet som är högst  $12\text{V}$ . För att sedan säkerställa att basströmmen snabbt leds ner i jord när detektion inte längre signaleras, har en pulldown resistor på  $10\text{ k}\Omega$  använts.

#### 4.1.6 Justering

I lösningen för att kunna uppfylla målet, justering av arbetsområdet med minst fem steg användes en vanlig tryckknapp (Tactile-Switch).

Komponenten är sammankopplad med ett RC-filter bestående av en resistor på  $15\text{ k}\Omega$  och en kondensator på  $15\text{ nF}$  vilket skapar ett lågpasfilter vilket illustreras i Figure 4.4 nedan. Detta medför en cutoff-frekvens på cirka  $700\text{ Hz}$  vilket är en sammanvägning av tillverkarens och handledarens rekommendation på  $100\text{ Hz}$  respektive  $1500\text{ Hz}$ , med motivationen att det är fördelaktigt att hålla frekvensen hög för att knappen inte ska bli alltför trög. Den filtrerade signalen är sedan kopplad in på ett interrupt-ben på raspberryn för att räkna upp en justeringsvariabeln.

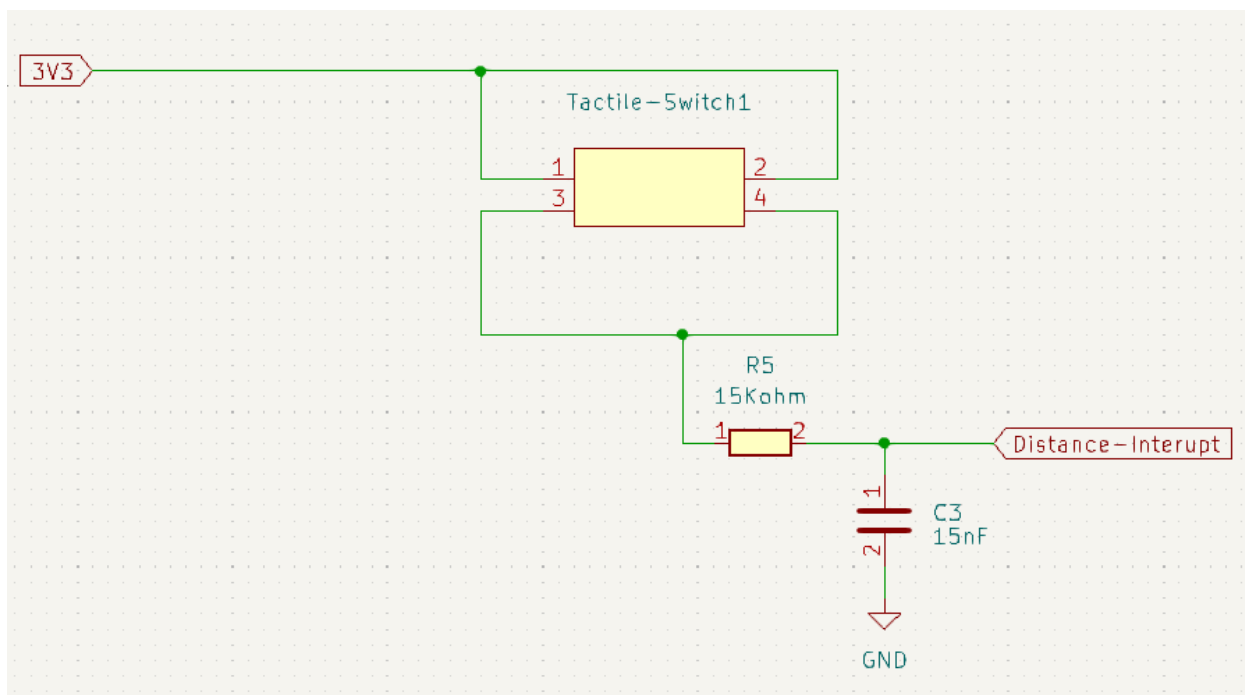
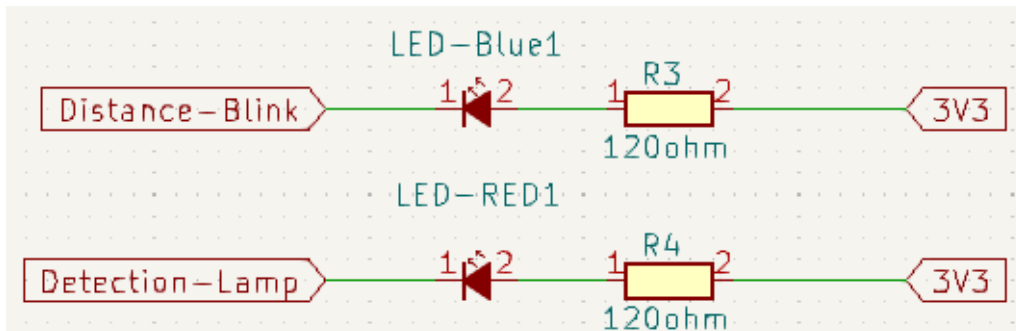


Figure 4.4: Knapp med tillhörande RC-filter

### 4.1.7 Visualisering

För att sedan visualisera både justering och detektion används två olika ljusemitterande dioder som är kopplade på hög signal 3,3 V tillsammans med 120  $\Omega$  resistor för att ge en ström på cirka 11 mA. Detta ger enligt databladet ett lagomt ljus för både ett rött ljus vid detektion och en blå blinkning när knappen trycks in, där antal blinkningar illustrerar vilken distansnivå som mikrokontrollern är inställd på. Det vill säga hur stort arbetsområde som programmet är inställt på, vilket baseras på vilken applikation sensorlösningen sitter monterad.



**Figur 4.5:** LED kretsar.

Distance-Blink och Detection i Figur 4.5 ovan är kopplade till raspberryn och är normalt höga (3,3 V) men vid användning dras de låga vilket skapar en spänningsskillnad på 3,3 V som gör att dioderna lyser.

## 4.2 Programkod

Programkoden som styr mikrokontrollern och hela sensorlösningen är uppbyggd av setup-inställningar, en huvudkod/huvudfunktion “try” och två delfunktioner “measure\_distance” och “button\_interrupt” varav den sistnämnda är en interruptfunktion. Hela programkoden finns som bilaga B men nedanför förklaras koden mer ingående del för del.

### 4.2.1 Setup

Setup-inställningarna innefattar importering av biblioteken, “time” och “machine” som är två standardbibliotek. Biblioteken används sedan i koden, eftersom programkoden använder sig av en klocka behövs time biblioteket och för att initiera och använda I/O-portar används machine biblioteket. Setup innefattar även initiering av vilka portar på mikrokontrollern som används och om porten ska vara en in- eller utgång. I punktlista nedan redovisas vilka I/O-portar som har valts:

- Pin 25 (LED\_PIN) sattes till output för att aktivera inbyggda lysdioden på mikrokontrollern.
- Pin 22 sattes till output för att kunna skicka ut signal (basström) till transistoren. Pin 21 (TRIGGER\_PIN1) sattes till output för att kunna skicka ut signaler till sensor et.

- Pin 20 (ECHO\_PIN1) sattes till input för att kunna ta emot signaler från sensor ett.
- Pin 19 (TRIGGER\_PIN2) sattes till output för att kunna skicka ut signaler till sensor två.
- Pin 18 (ECHO\_PIN2) sattes till input för att kunna ta emot signaler från sensor två. Pin 17 (BUTTON\_PIN) sattes till input för att ta emot interrupts.
- Pin 15 och 16 sattes till output för att kunna tända varning\_lamp och button\_lamp.

I setup-inställningarna initieras även tre globala variabler: larm, distance\_1\_2\_3\_4\_5 och distance\_check. Alla tre variabler tilldelas värde noll och används senare i koden.

De två sista delarna som görs i setup-inställningarna är att funktionerna button\_interrupt och measure\_distance definieras och samtidigt initieras även interrupt hanteraren med inställning att verka på fallande trigg. Fullständig kod över setup-inställningarna kan hittas i bilaga B.

### 4.2.2 Delfunktionen “measure\_distance”

Delfunktionen measure\_distance uppgift är att mäta avståndet till ett objekt och sedan returnera avståndet i centimeter. Funktionen har två invärden “trigger” och “echo” och beroende på vilken sensor som ska mäta skickas antingen variabler för sensor 1 eller 2 in i funktionen. Programkoden för funktionen styr alltså hur sändaren på sensorn skickar ut ljud på triggerpin och hur den sedan tar emot en tidspuls från echopin.

```
26 def measure_distance(trigger, echo):
27     # Skicka en kort puls till triggre
28     trigger.value(1)
29     time.sleep_us(10)
30     trigger.value(0)
31     # Vänta på att echo ska gå hög
32     start_check = time.ticks_us()
33     while echo.value() == 0:
34         pulse_start = time.ticks_us()
35         #Felhantering när while-loop fastnar
36         if pulse_start - start_check > 8750:
37             pulse_end = pulse_start + 8750
38             break
39     # Vänta på att echo ska gå låg
40     while echo.value() == 1:
41         pulse_end = time.ticks_us()
42         #Ser till att mätning avbryts vid 150cm eller längre
43         if pulse_end - pulse_start > 8750:
44             break
45     # Beräkna tiden för ljudpulsens att resa fram och tillbaka
46     pulse_duration = pulse_end - pulse_start
47     # Omvandla tiden till avstånd (avstånd = tid * ljudets hastighet / 2)
48     distance_cm = pulse_duration * 0.0343 / 2
49     # Ljudets hastighet är ca 343 meter per sekund
50     return distance_cm
```

Figur 4.6: Programkod för funktionen *measure distance*.

I figur 4.6 ovan visas på rad 28 till rad 30 hur sändaren sätts aktiv (skickar ut ljudvågor) och att den är aktiv i tio mikrosekunder innan den inaktiveras. Efter att sändaren har inaktiverats startas sedan en while-loop (rad 33) som har i uppgift att reagera och föra över den aktuella tiden när echo-pinnen går från 0 till 1. Andra while-loopen på rad 40 har villkoret att bli aktiv när echo-pinnen är 1 (Hög). I denna loop loggas sedan en sluttid tills det att echo-pinnen blir 0 (Låg). Båda while-looparna innehåller även en if-sats som är till för att programmet inte skulle kunna fastna kvar i någon av looparna. Dessa är inställda på 8750 mikrosekunder, vilket motsvarar en mätning på cirka 150 cm som är den längsta distansen som behöver mätas.

Genom att nu på rad 46 subtrahera sluttid minus starttid kan echo-pulsens tidsspann räknas ut och sparas i `pulse_duration`. Detta är med hjälp av sensorernas interna krets och logik den tid det tar för ljud att färdas från sensor till ett objekt och sedan tillbaka. För att nu till sist på rad 48 beräkna avståndet, används som tidigare nämnt ekvation 2.1 (avståndsformeln). Genom att multiplicera tiden i mikrosekunder med ljudets hastighet i enheten cm per mikrosekund. När detta sedan delas med två fås avståndet fram till "objektet". Det sista som sedan görs i koden är att avståndet returneras.

### 4.2.3 Del Funktionen “button\_interrupt”

Delfunktionen `button_interrupt` är en interruptfunktion och kan bli kallad närsomhelst i hela programmet. Det funktionen gör är att den räknar upp den globala variabeln `distance_1_2_3_4_5` genom att räkna antalet knapptryckningar. Befinner sig den globala variabeln mellan ett och fem kommer funktionskoden styra en LED-lampa att initialt blinka och indikera vilket mätavstånd som är inställt. Blinkar LED-lampan en gång är avståndet inställt på läge ett och blinkar den två är avstånd två inställt och så vidare. Dock när programmet startas är variabeln `distance_1_2_3_4_5` inställd med värde noll och då blinkar inte leden.

När variabeln `distance_1_2_3_4_5` är inställd med värde noll innebär det ett arbetsområde på 50 cm. Vidare när tryckknappen trycks in räknas variabeln `distance_1_2_3_4_5` upp och då ökar också arbetsområdets längd med 20 cm vid varje knapptryck upp till 150 cm. Detta innebär att arbetsområdet kan ställas in med sex olika avstånd för att lätt kunna anpassas till olika applikationer.

Skulle dock antalet knapptryckningar istället överstiga fem kommer hela programmet att återstartas. Denna återstartsfunktion finns där av två anledningar. Den första är för att kunna gå från avstånd fem till att komma ner till avstånd noll igen och den andra anledningen är för att kunna återstarta hela programkoden om någonting låser sig i programmet. I figur 4.7 nedan visas hela programkoden för interruptfunktionen.

```

52 def button_interrupt(pin):
53     interrupt_start = time.ticks_us() # Mät tiden när avbrottet inträffar
54     interruptport_low = interrupt_start + 1 #sätt initialt värde
55     while BUTTON.value() == 1: # Loopa så länge knappen är intryckt (hög)
56         interruptport_low = time.ticks_us() # Uppdatera tiden när knappen är i läget "låg"
57         #Endast släppa igenom signal unde 700Hz
58     if (interruptport_low - interrupt_start) > 1400:
59         # Avbryt avbrottsfunktionen för att undvika fler avbrott medan vi utför åtgärder
60         BUTTON.irq(handler=None)
61         # Vänta en kort stund för att undvika flera åtgärder på kort tid
62         time.sleep(0.7)
63         # Öka avståndsvariabeln
64         global distance_1_2_3_4_5
65         distance_1_2_3_4_5 += 1
66         if distance_1_2_3_4_5 > 5: # Om antalet avstånd är större än 5, återställ enheten
67             machine.reset()
68         for i in range(0, distance_1_2_3_4_5): # Blinka lampan för varje avstånd som har räknats
69             button_lamp.value(0)
70             time.sleep(0.3)
71             button_lamp.value(1)
72             time.sleep(0.5)
73     # Återaktivera avbrott för knappen när åtgärderna är klara
74     BUTTON.irq(trigger=machine.Pin.IRQ_RISING, handler=button_interrupt)

```

Figur 4.7: Programkod för interruptfunktion `button_interrupt`.

Det första som händer kan liknas med en filter-funktion där tiden som knappen tryck in mäts. Om den inte varit intryckt mer än 1400 mikrosekunder hanteras den som en kontaktstudds. Detta säkerställer att frekvenser på över ungefär 700Hz filtreras bort. Själva funktionen börjar sedan med att interrupt hanteraren inaktiveras (rad 60) och det är för att interrupts inte ska kunna triggas under tiden som funktionen

körs. Efter en fördröjning på 0,7 sekunder räknas variabeln `distance_1_2_3_4_5` upp med ett steg. Anledningen till att denna fördröjning används är för att LED-lampan som senare i koden används inte ska blinka för fort efter ett knapptryck. Är nu variabeln större än fem, så kommer villkoret för if-satsen på rad 66 att uppfyllas, vilket kommer köra funktionen “`machine.reset()`”. Detta återställer hela programmet och leder till omstart av enheten. Om nu variabeln `distance_1_2_3_4_5` istället har ett värde mellan ett och fem efter knapptrycket kommer istället en for-loop att köra lika många varv som de värde variabeln har. Till exempel har variabeln värde fyra, kommer leden att blinka fyra gånger och då också visa att avståndsnivå fyra är inställd. Det sista som sedan görs i funktionen är att interrupt hanteraren återaktiveras och är då redo för att hantera nya knapptryckningar.

Något som kan noteras och kan se lite konstigt ut i koden är att när `button_lamp.value` sätts till noll som det gör på rad 69 i figur 4.7 ovan så lyser LED-lampan. Det blir i princip omvänd logik och det är för att dennes anod är kopplad på mikrokontrollerns 3,3 volts utgång. För att nu kunna tända leden måste det alltså bildas ett spänningsfall över den och det görs enbart när I/O-porten sätts till noll volt.

### 4.2.4 Huvudfunktionen

Huvudfunktionen “try” är som hela programmets skelett och det är även den längsta funktionen i hela programkoden. Denna funktion kan hittas i sin helhet i bilaga B. Det första som händer i try-funktionen är att detektionsleden (`varning_lamp`) och avståndsleden (`button_lamp`) sätts till hög nivå och det för att säkerställa att de inte lyser inledningsvis när programmet startas. Likaså sätts även utgången “sig” till låg för att säkerställa att det inte går någon basström till transistorn.

För att sedan signalera att systemet är igång blinkar den inbyggda leden till en gång med en halv sekund. Detta görs för att användaren ska få en visuell bekräftelse på att systemet har satts igång korrekt.

Efter att LED:n på mikrokontroller kortet har varit släkt i en halv sekund startar en while-loop med villkoret sant. Detta villkor gör att while-loopen kommer köras hela tiden och därför kan denna loop ses som själva programcykeln. Det första som då händer i while-loopen är att `measure_distance` funktionerna körs för både `sensor1` och `sensor2`, om och om igen tills det att en detektion har upptäckts. Detta sker genom en if-sats som får programmet att hoppa till låsläge om villkor för detektion uppfylls.

Låsläget inleds med att varningslampan tänds och signal skickas till lås funktionen på grinden. Något som sker genom att mikrokontrollern skickar en basström till transistorn som drar moderkortets höga signal till låg. Detta följs sedan av en while-lopp som kontrollerar om sensorerna fortfarande detekterar objekt inom arbetsområdet. Denna loop måste köras 15 gånger utan detektion för att programmet ska gå ur låsläget och tillbaka till den vanliga programcykeln.

Sammanfattningsvis handlar huvudfunktionen try om att while-satsen hela tiden

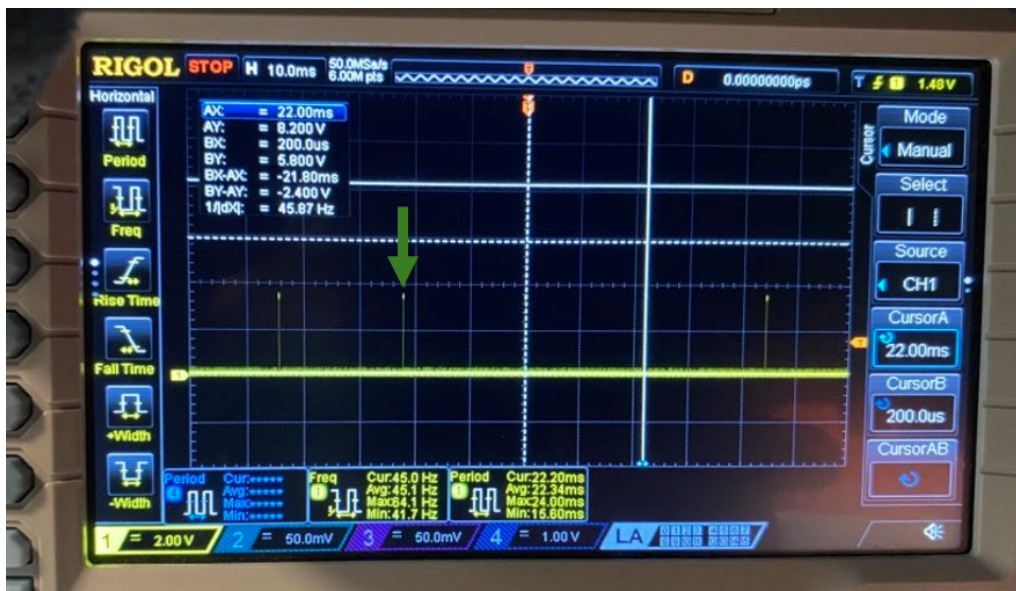
körs och i while-satsen används funktionen `measure_distance`. Beroende på vad `measure_distance` funktionen mäter `distance1` och `distance2` till, kommer signalen på moderkortet antingen hållas hög eller dras låg. Dras signalen låg har sensorerna uppmätt avstånd innanför arbetsområdet och grinden blockeras från rörelse, för att skydda en potentiell människa.

### 4.3 Uppfyllnad kvantitativa mål

I detta avsnitt presenteras hur den slutgiltiga sensorlösningen uppfyller de uppsatta kvantitativa målen.

För att sensorlösningen skulle fungera på ett effektivt sätt, sattes det upp ett mål som innebar att närvarodetektion av människa i arbetsområdet ska ske på högst 20 millisekunder. Detta mål har varit en viktig del i hela lösningen då lösningens syfte hela tiden har varit att skydda människor i arbetsområdet. Skulle detektion inte ske under en kort tidsperiod, finns risken att grinden hade kunnat öppnats ändå och olyckligtvis träffat en människa som då kanske befann sig i arbetsområdet.

Genom att nu titta på den slutgiltiga lösningen och mäta periodtiden mellan två mätningar på en av sensorerna fås den maximala detektionstiden. Detta eftersom en mätning sker en gång per programcykel och via koden kan det säkerställas att detektions-funktionen alltid kommer att kunna nås under denna periodtid. Mätningen gjordes genom att ett oscilloskop kopplades upp och anslöts till TRIG-pin på en av sensorerna. I figur 4.8 nedan ses en bild från mätningen med oscilloskopet.



**Figur 4.8:** Bild på ett oscilloskop som visar periodtiden mellan varje ny mätning för en sensor som mäter 150 cm.

## 4. Resultat

Figur 4.8 ovan visa oscilloskopets skärm att sensorn gör en ny mätning varje gång det kommer en puls. En puls visualiseras på skärmen som ett gult lodrätt streck. Med ett gult streck menas ett sådant som den gröna pilen pekar på i figuren. För att nu mäta periodtiden mäts nu avståndet mellan varje puls och då fås en periodtid på cirka 22 millisekunder, som också kan ses högst upp till vänster (AX = 22.00ms). Detta innebär att målet på att kunna detektera en människa på högst 20 millisekunder inte helt är uppfyllt. Dock ska det tilläggas att dessa 22 millisekunder är den högsta tid det tar för programmet att gå ett varv, vilket bevisar att en detektion aldrig kommer ta längre tid än detta. Detta exempel är också när arbetsområdet är inställt på 150 cm. Sätts arbetsområdet istället på till exempel 70 cm fås en periodtid mellan varje ny mätning på 12,4 millisekunder se figur 4.9 nedan.



**Figur 4.9:** Bild på ett oscilloskop som visar periodtiden mellan varje ny mätning för en sensor som mäter 70 cm.

I figur 4.9 visas det nu tydligt att pulserna inträffar tätare än vad de gjorde i figur 4.8. Dock är inte detta så konstigt eftersom ett större arbetsområde kräver att ljudet färdas längre och då tar det också längre tid för sensorn att mäta.

För att sedan undersöka hur sensorlösningen uppfyller de uppsatta målen kring minst 90 procent detektion i arbetsområdet och att falska detektioner som högst får vara 10 procent, används protokollet från sluttestet. Se ifyllt protokoll i figur 4.10 nedan.

<b>Minst 90% säkerhet att sensorlösningen detekterar när ett objekt av dimensioner 200 x 200 mm (X och Y-led) är inom arbetsområdet.</b>			
Föremål kommer in i arbetsområdet från vänster			
Detekteras:	X/30	X=	30
Föremål kommer in i arbetsområdet från höger			
Detekteras:	Y/30	Y=	29
Föremål kommer in i arbetsområdet rakt mot sensorn (från cirkelbågen)			
Detekteras:	Z/30	Z=	30
Detektionssäkerhet i procent:		<b>99%</b>	<b>&lt;-- Över 90% är målet uppfyllt</b>
<b>Högst 10% falska detektioner, vilka kan uppstå exempelvis när objekt befinner sig strax utanför arbetsområdet eller pga störningar i omgivande miljö som till exempel ljud eller ljus.</b>			
Placerar föremål utanför arbetsområdet på väster sida			
Detekteras:	X/30	X=	0
Placerar föremål utanför arbetsområdet på höger sida			
Detekteras:	Y/30	Y=	0
Placerar föremål utanför arbetsområdet mitt emot sensorerna			
Detekteras:	Z/30	Z=	0
Falskdetektion i procent		<b>0%</b>	<b>&lt;-- Under 10% är målet uppfyllt</b>

**Figur 4.10:** *Protokoll från slutttest*

I figur 4.10 ovan ses resultatet från sluttestet som visar att sensorlösningen har en detektions säkerhet på 99 procent, för ett föremål som är 200 x 200 mm innanför arbetsområdet. Sluttestet visade även att inga falska detektioner alls inträffade när objektet placerades fem centimeter utanför arbetsområdet. Detta innebär att målen med minst 90 procent detektions säkerhet och högst 10 procent falska detektioner är uppfyllda. Det som dock ska noteras från sluttesterna är att objektet vid varje mätning alltid noga placerades med sin plana sida rakt mot sensorerna.

Ett annat mål var att sensorlösningen skulle ha ett justerbart arbetsområde i minst fem steg i intervallet mellan 50 - 150 cm. Detta har också uppfyllts då sensorlösningen försätts med en tryckknapp, vars funktion är att justera längden på arbetsområdet. Den slutgiltiga lösningen har sex inprogrammerade avstånd vilket uppfyller målet att minst ha fem. Detta kan även ses mer utförligt i avsnitt 4.1.2.3 Del Funktionen `button_interrupt` där funktionen mer beskrivs.

För att sensorlösningen också ska uppfylla målet med detekteringens undre gräns på 40 cm från golv, är egentligen det enda kravet att sensorerna monteras någonstans från golvet upp till 40 cm och stående rakt upp 90 grader. Detta medför alltså att höjden kan anpassas (mellan 0 och 40 cm) beronde på applikation och innebär att målet kan uppfyllas om sensorerna monteras på detta vis.

Det näst sista målet som sattes upp var att sensorlösningen skulle klara av att fungera i ett temperaturintervall på 0-45 °C. Detta mål testades inte rent fysiskt då det bedömdes svårt att med enkelhet få till ett test där en temperatur mellan dessa intervall kunde säkerställas. Här fick istället databladet till alla komponenter användas. Genom att ha gått igenom alla datablad för samtliga komponenter och inte hittat någon komponent som inte ska klara en temperatur mellan 0-45 °C är bedömningen att även detta mål också är uppfyllt.

Målet var att ha en totalkostnad för funktionskritiska komponenter (där funktionen kräver en specifik typ av komponent) samt kretskortet inte skulle överstiga 150 kronor. Den slutgiltiga lösningens styckpris för ett kort samt en uppsättning av dessa komponenter ser ut som följande:

- I detta "proof of concept" projekt beställdes 5 stycken kretskort till en kostnad på 1370kr vilket ger ett styckpris för ett kretskort på 274kr.
- 1st Raspberry Pi Pico SC0917 52,9kr st
- 1st Spänningsregulator L7805CV 6,29kr st
- 2st Digitala ultraljudsensor HC-SR04 22kr st

Total kostnad: 377,19kr. Denna totalkostnad uppfyller inte målet om 150 kronor, mycket på grund utav den höga kostnaden på kretskortet. Med hjälp av information på leverantörers hemsidor visade det sig att den bästa metoden för att få ner styckespriset på kretskorten vara att öka volymen för beställningen, vilket även redovisas i teorikapitlet. Om en beställning istället skulle ske på 100 st kretskort från samma leverantör skulle detta kosta ungefär 4140kr. Isåfall skulle detta ge ett styckpris på 41,4kr och totalkostnad för sensorlösningen på 144,59kr och därmed skulle målet vara uppfyllt.

# 5

## Diskussion

I detta kapitel diskuteras projektets mål gentemot den slutgiltiga erhållna produkten. De delar som diskuteras är bland annat de mål och frågeställningar som var svåra att besvara, hurvida en mikrokontroller skulle användas eller inte följt av fördelen att faktiskt använda mjukvara i lösningen. Slutligen diskuteras även sensorernas komplexitet samt nedskalning på styckespriset med hjälp av en mer utvecklad sensorlösning och ett högre antal producerade enheter.

### 5.1 Anpassning av mål och frågeställningar

En aspekt som hela tiden försökte tas hänsyn till under projektet var att skapa en sensorlösning som klarar av olika typer av miljöer. Detta var både något som togs upp i frågeställningarna och i målsättningarna, men det visade sig vara en betydande utmaning i praktiken. Att säkerställa funktionen för sensorlösningen ur miljösynpunkt är avgörande för dess pålitlighet, men var en ytterst komplex uppgift att lösa till fullo. Detta då de kräver väldigt mycket data och tester för att se om lösningen fungerar i "alla" potentiella miljöer. I och med detta kan diskussioner föras om att dessa frågeställningar och mål kan ses som optimistiska att få svar på då det skulle kräva ett helt eget projekt att ge sådana testdata rättvisa. En frågeställning som kanske hade varit bättre formulerad som fortfarande får in miljöfaktorn hade kunnat varit något i stil med att säkerställa sensorlösningen i butiksmiljö eller likvärdigt. För denna frågeställning hade tester kunnat göras för alla potentiella störningar som en vanlig butiksmiljö skulle kunna uppbringa men bortse från miljöer som till exempel har ljud och ljus med ovanliga frekvenser med mera.

Vidare kan även målet att sensorlösningen skulle fungera i temperaturintervallet 0–45 °C också diskuteras vara ambitiöst satt, då några slutester för detta var svåra att utföra i och med avsaknaden av rätt testutrustning. En rimligare frågeställning här hade istället kunnat vara att formulera sig med mindre temperaturskillnader samt också här vrida det mer mot en butiksmiljö. Detta hade kunnat varit enklare att testa än att just säkerställa sådana höga och låga temperaturer.

### 5.2 Storlek och typ av objekt

Något som noterades från slutesterna var att när mycket plana objekt skulle detekteras var det viktigt att de placerades med sin plana sida hyffsat rakt mot sensorlösningen. Troligtvis eftersom en för bra reflektor med plan yta i felvinkel kan

se till att ljudvågorna studsar ifrån istället för tillbaka till sensorerna. Detta kunde erfaras med den kub som användes i testerna, men är något som inte upplevts vara något problem när funktionen istället testats med till exempel en kroppsdel. Frågan är hur en miss-detektion som skapas utav att ett väldigt plant objekt med fel vinkel ska bedömmas. Enligt de avgränsingar som gjordes för projektet så är det människor som är det viktiga. Eftersom denna typ av detektion inte visat några problem så kan en diskussion föras om att de kanske är okej för just denna applikation. Detta är dock något som kan utredas mer innan sensörlösningen sätts i produktion och används i andra applikationer.

En till erfarenhet som kan tas med från dessa tester är att en riktigt bra reflektor faktiskt kan ha en liten påverkan på sensorernas spridningsvinkel. Det upptäcktes genom att kuben som användes faktiskt detekterades lite bättre på kanterna (i arbetsområdet) än förväntat, medans till exempel ett ben detekterades ungefär enligt beräkningarna. Detta var något som till viss del hade noterats i mycket tidigare tester men här blev det tydligare. Därför kan det vara viktigt att justera vinkeln på sensorerna utefter vilka objekt som ska detekteras och vart sensorerna ska sitta.

### 5.3 Logik och mikrokontroller

I arbetet har alla frågeställningar utom frågeställningen “Kan logiken hanteras utan en programmerbar enhet?” mer eller mindre lyckats att besvaras. Anledningen till varför den inte har blivit besvarad är för att den valda sensorn (HC-SR04) i den slutgiltiga sensörlösningen kräver en programmerbar enhet för att kunna användas. Med denna sensor har alltså logiken inte varit möjlig att implementera utan en programmerbar enhet. Vidare medför detta isåfall att svaret på denna frågeställning hade blivit att logiken inte går att lösa med denna typen av sensor. Om frågan skulle vilja utredas hade det alltså ha varit till priset av att möjligtvis använda en sämre sensor och att undvika annan teknik på kretskortet som också använder sig av mikrokontrollern. Att bygga en sensörlösning helt utan en mikrokontroller känns som en svårare uppgift, men den är säkert möjlig.

### 5.4 Anpassning och utvecklingspotential av mjukvaran

Genom att den slutgiltiga sensörlösningen till stor del styrs av mjukvara, innebär det att sensörlösningen också kan ses som relativt flexibel och anpassningsbar. Detta kommer vidare kunna göra det lättare för en potentiell användare av sensörlösningen att kunna ändra och anpassa sensörlösningens funktioner till att mer precis kunna likna användarens önskemål. Ett exempel som skulle kunna inträffa är att en grindlängd inte helt överensstämmer med de redan inprogrammerade standard längderna för arbetsområdet (50, 70, 90, 110, 130 och 150) som ställs in via tryckknappen. Här kan alltså arbetsområdet helt fritt ställas in beroende på hur lång grinddörren

är i spannet 0-150 cm och det görs då i programkoden.

Ytterligare ett exempel skulle kunna vara att användaren kanske vill justera tiden som grinden hålls låst efter att ett föremål senast detekterades. Så som koden är uppbyggd just nu är dörren låst från rörelse i ca en halv sekund efter detektion. Med denna korta fördröjningstid ökar det chansen att en person förhoppningsvis hinner iväg en bit från arbetsområdet och då minskar också risken att personen blir träffad av grinden. Men samtidigt är det även viktigt att tänka på inpassage in i butiken och inte sätta en allt för lång fördröjning heller. En för lång fördröjning kan innebära att folk som vill in i butiken kanske blir otåliga och istället för att vänta på att grinden automatiskt öppnar sig kanske de trycker upp grinden själva, vilket i sin tur kan leda till att grindsystemet tar skada och det kan leda till dyra framtida reparationer.

I detta projekt har förväntningarna på sensorlösningen hela tiden varit att den ska kunna detektera människor i en typisk butiksmiljö, vilket tester har visat att den också behärskar på ett bra sätt. Detta genom att den testmiljö som har använts vid testerna i detta projektet kan likna en typisk butiksmiljö. Men genom att sensorlösningen återigen till stor del är uppbyggd av mjukvara, kommer troligtvis lösningen också kunna anpassas till att fungera i andra miljöer också. Sensorlösning kanske kan användas i en smutsig industrimiljö där falska detektioner blir ett stort problem. För mycket damm och smuts i luften kanske uppfattas som ett föremål och då detekterar sensorlösningen det som ett föremål. Här kan sensorlösningen ställas in så att den måste detektera föremål med en viss algoritm för att detektion ska signaleras. Detta gör vidare sensorlösningen till en applikation som förhoppningsvis kommer kunna användas i flertalet områden i syfte att skydda människor.

## 5.5 Nedskalning och kostnadsbesparingar

Under hela projektet har det varit tydligt att ett utvecklingskort med mikrokontroller (Raspberry Pi Pico) troligtvis skulle bli överflödigt då den innehåller flera funktionaliteter som inte skulle komma att användas. Utvecklingskortet valdes ändå att användas eftersom den gav möjlighet till samtliga funktioner i utvecklingsfasen och satte färre begränsningar för att testa fram en så optimal funktion för ändamålet så möjligt. Efter att den slutgiltiga sensorlösningen har utvecklats kan detta också konstateras, då lösningen endast i princip använder funktioner som är direkt kopplade till mikroprocessorn (RP2040).

Vidare kan även själva sensorkorten diskuteras innehålla för mycket teknik i sig självt och vara onödigt avancerade. Det mesta av tekniken skulle troligtvis kunna hanteras av mikrokontrollern, vilket skulle undvika användning av de komponenter som sköter logiken som sitter på sensorkorten just nu. Detta skulle kunna göra så att bara sändare och mottagare används från dessa.

I och med dessa aspekter skulle lösningen kunna skalas ner genom att istället sätta en sådan processor direkt på sensorlösningens kretskort samt endast använda sändare och mottagare som tillsammans med mikrokontrollern skulle kunna ska-

på en ultraljudssensor. Dessa utvecklingsmöjligheter skulle kunna göra att onödiga komponenter på utvecklingskortet och sensor korten skulle kunna undvikas. Något som potentiellt kan skala ner kostnaden men dock även satt högre krav på den egenutvecklade elektriska designen. Denna lösning kanske kan ses som ett för stort arbete för detta projekt men fortfarande en intressant förbättring för vidare utveckling av sensorlösningen i framtiden.

Utifrån kostnadsanalysen som gjordes bevisades att kostnadsmålen inte kunde uppfyllas inom ramen för detta projekt. Resultat kan ändå ses som ganska väntat i och med den bakgrunds information som samlades i början av projektet. Detta kan också vara ett argument för en omformulering av den målsättning som sattes inför projektet. I resultatet redovisades även en snabb beräkning för en större produktion och det bevisade att kostnadsmålet då ganska snabbt skulle kunna uppfyllas. Målsättningen hade därför kanske kunnat formulerats om med att uppfylla 150 kronor vid en större produktion på runt 50 eller 100 enheter. Genom en omformulering av denna målsättning på ett sådant vis hade kanske kostnaderna för de olika delarna inte begränsat utvecklingen av produkten på samma sätt. Där komponenterna för utvecklingen hade kunnat kosta mer så länge man såg en rimlig kostnadsminskning vid större produktion och en nedskalad slutprodukt i framtiden.

# 6

## Slutsats

Projektet har resulterat i en justerbar sensorlösning som detekterar och skyddar människor från en rörlig enhet i ett begränsat arbetsområde. Sensorlösningen har även visat sig både vara robust och pålitlig genom sin höga detektionssäkerhet på 99 procent, med även minimal risk för falska detektationer. Även genom den snabba detekteringen på maximalt 22 millisekunder skyddas också människor på ett bra sätt från att inte bli träffade av den rörliga enheten.

Att längden på arbetsområdet kan justeras med hjälp av en enkel tryckknapp har möjliggjort sensorlösningen till att vara enkel att anpassa för olika situationer där storleken på arbetsområdet varierar. Tryckknappen tillsammans med tillhörande blå LED gör det också lätt att installera då en installatör enkelt kan ställa in rätt längd på arbetsområdet men också kan se på LEDen vilket avstånd som faktiskt är inställt.

Trots goda resultat i detektionssäkerhet presenteras några delar med mer varierande resultat. En del var att totalkostnaden för alla komponenter till sist översteg det uppsatta kostnads målet. Här har arbetet dock visat att kostnaderna minskas rejält vid storskaliga beställningar. Redan vid en beställning av hundra sensorlösningar är det uppsatta kostnads målet uppfyllt. En annan stor del var att sensorlösningen inte kunde testas i alla möjliga miljöer och därmed kunde de inte säkerställas att fungera i alla dessa heller. Med det sagt så testades sensorerna i en normal miljö som kan liknas med en butiksmiljö och här fungerade dessa bra. Därför är kanske inte denna del speciellt kritiskt för detta projektet men lösningen rekommenderas att testas och optimeras utefter den miljö och omgivning en sådan sensorlösning ska appliceras i.

Sammanfattningsvis har arbetet resulterat i en pålitlig och effektiv sensorlösning som har uppfyllt kraven och målen på ett bra sätt. Genom detta arbete har det skapats en lösning som bidrar till ökad säkerhet kring rörliga enheter.



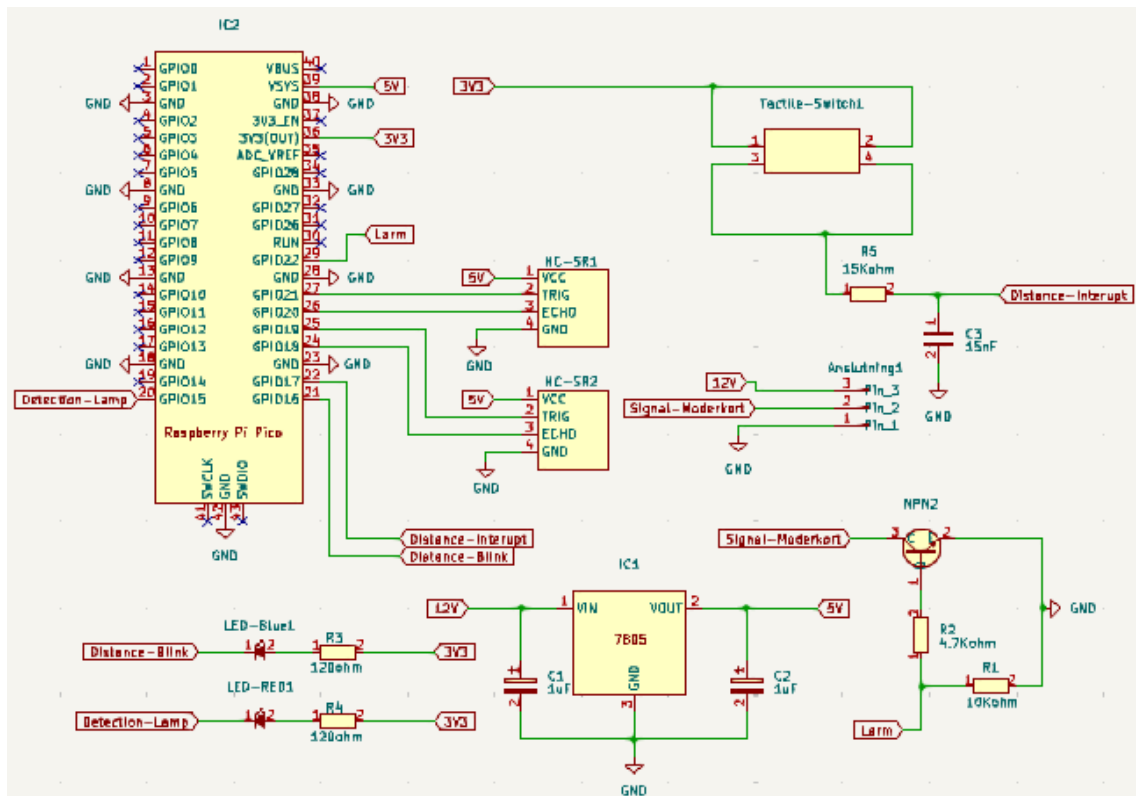
# Litteraturförteckning

- [1] M.Örnborg, "Fjolårets statistik över dödsfall sticker ut", Automation, Sep, 08, 2023. [Online]. Tillgänglig: <https://automation.se/fjolarets-statistik-over-dodsfall-sticker-ut/>, Hämtad: 2024-01-31
- [2] Arbetsmiljöverket. "Arbetsolyckor med dödlig utgång 2011-2020.", u å, [Online]. Tillgänglig: <https://www.av.se/globalassets/filer/publikationer/rapporter/arbetsolyckor-med-dodlig-utgang-2011-2020.pdf>, Hämtad: 2024-02-01
- [3] D. Voss, "March 1880: The Curie Brothers Discover Piezoelectricity" APSNews, vol. 23, nr. 3. Mar. 2014 [Online]. Tillgänglig: <https://www.aps.org/publications/apsnews/201403/physicshistory.cfm>, Hämtad: 2024-03-12.
- [4] F. A. Duck, "Langevin's Ultrasonic Metrology," in IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, vol. 70, no. 2, pp. 173-180, Feb. 2023, doi: 10.1109/TUFFC.2022.3222085. Hämtad: 2024-03-12
- [5] S. Sudtana, N. Sisuk, K. Prompak, A. Keawpoolsuk, N. Maneegirapragarn and P. Wardkein, "Ultrasonic Distance Detection Based on Forced Oscillation of Quadrature Oscillator," 2020 8th International Electrical Engineering Congress (iEECON), Chiang Mai, Thailand, 2020, pp. 1-4, doi: 10.1109/iEECON48109.2020.229561. Hämtad 2024-04-05
- [6] Ljudteknikern, "Ljudhastighet i luft vid olika temperaturer" 2021. [Online] Tillgänglig: <https://ljudteknikern.se/ljudhastighet/> (hämtad: 2024-04-06).
- [7] Ultrasonic Ranging Module HC - SR04, ElecFreaks, [Online]. Tillgänglig: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>, Hämtad: 2024-04-20.
- [8] Raspberry Pi [Online]. Tillgänglig: <https://www.raspberrypi.com/> (hämtad 2024-05-05).
- [9] Raspberry PI Pico Datasheet, Raspberry Pi Ltd, 2024. [Online]. Tillgänglig: <https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf>, Hämtad: 2024-05-05.
- [10] Positive voltage regulator ICs, STMicroelectronics, 2018. [Online]. Tillgänglig: <https://www.st.com/content/ccc/resource/technical/document/datasheet/41/4f/b3/b0/12/d4/47/88/CD00000444.pdf/files/CD00000444.pdf/jcr:content/translations/en.CD00000444.pdf>, Hämtad: 2024-04-10.
- [11] S. Bano and T. Altaf, Lowpass, bandpass and highpass filters using current inversion type negative impedance Converter, "2011 IEEE 14th International

- Multitopic Conference, Karachi, Pakistan, 2011, pp. 62-66, doi: 10.1109/INMIC.2011.6151511. Hämtad: 2024-04-03.
- [12] KiCad EDA, “A Cross Platform and Open Source Electronics Design Automation Suite”. 2024 [Online]. Tillgänglig: <https://www.kicad.org/> (hämtad: 2024-05-11).
- [13] Wellpcb, “Reducing PCB Cost –Top 7 Tips to Help Save Cost During PCB Manufacturing and Assembly”. [Online]. Tillgänglig: <https://www.wellpcb.com/reducing-pcb-cost.html> (hämtad: 2024-04-20).

# A

## Krettschema





# B

## Programkod

```
1 import machine
2 import time
3
4 # Definiera GPIO-pinnar
5 TRIGGER_PIN1 = 21
6 ECHO_PIN1 = 20
7 TRIGGER_PIN2 = 19
8 ECHO_PIN2 = 18
9 BUTTON_PIN = 17
10 varning_lamp = machine.Pin(15, machine.Pin.OUT)
11 button_lamp = machine.Pin(16, machine.Pin.OUT)
12 sig = machine.Pin(22, machine.Pin.OUT)
13 larm = 0
14 distance_1_2_3_4_5 = 0
15 distance_check = 0
16 LED_PIN = 25
17
18 # Initiera GPIO-pinnar
19 BUTTON = machine.Pin(BUTTON_PIN, machine.Pin.IN, machine.Pin.PULL_DOWN)
20 trigger1 = machine.Pin(TRIGGER_PIN1, machine.Pin.OUT)
21 echo1 = machine.Pin(ECHO_PIN1, machine.Pin.IN)
22 led = machine.Pin(LED_PIN, machine.Pin.OUT)
23 trigger2 = machine.Pin(TRIGGER_PIN2, machine.Pin.OUT)
24 echo2 = machine.Pin(ECHO_PIN2, machine.Pin.IN)
25
```

## B. Programkod

---

```
26 def measure_distance(trigger, echo):
27     # Skicka en kort puls till trigger
28     trigger.value(1)
29     time.sleep_us(10)
30     trigger.value(0)
31     # Vänta på att echo ska gå hög
32     start_check = time.ticks_us()
33     while echo.value() == 0:
34         pulse_start = time.ticks_us()
35         #Felhantering när while-loop fastnar
36         if pulse_start - start_check > 8750:
37             pulse_end = pulse_start + 8750
38             break
39     # Vänta på att echo ska gå låg
40     while echo.value() == 1:
41         pulse_end = time.ticks_us()
42         #Ser till att mätning avbryts vid 150cm eller längre
43         if pulse_end - pulse_start > 8750:
44             break
45     # Beräkna tiden för ljudpulsens att resa fram och tillbaka
46     pulse_duration = pulse_end - pulse_start
47     # Omvandla tiden till avstånd (avstånd = tid * ljudets hastighet / 2)
48     distance_cm = pulse_duration * 0.0343 / 2
49     # Ljudets hastighet är ca 343 meter per sekund
50     return distance_cm
51
52 def button_interupt(pin):
53     interupt_start = time.ticks_us() # Mät tiden när avbrottet inträffar
54     interuptport_low = interupt_start + 1 #sätt initialt värde
55     while BUTTON.value() == 1: # Loopa så länge knappen är intryckt (hög)
56         interuptport_low = time.ticks_us() # Uppdatera tiden när knappen är i läget "låg"
57         #Endast släppa igenom signal unde 700Hz
58         if (interuptport_low - interupt_start) > 1400:
59             # Avbryt avbrottsfunktionen för att undvika fler avbrott medan vi utför åtgärder
60             BUTTON.irq(handler=None)
61             # Vänta en kort stund för att undvika flera åtgärder på kort tid
62             time.sleep(0.7)
63             # Öka avståndsvariabeln
64             global distance_1_2_3_4_5
65             distance_1_2_3_4_5 += 1
66             if distance_1_2_3_4_5 > 5: # Om antalet avstånd är större än 5, återställ enheten
67                 machine.reset()
68             for i in range(0, distance_1_2_3_4_5): # Blinka lampan för varje avstånd som har räknats
69                 button_lamp.value(0)
70                 time.sleep(0.3)
71                 button_lamp.value(1)
72                 time.sleep(0.5)
73     # Återaktivera avbrott för knappen när åtgärderna är klara
74     BUTTON.irq(trigger=machine.Pin.IRQ_RISING, handler=button_interupt)
75
76     BUTTON.irq(trigger=machine.Pin.IRQ_RISING, handler=button_interupt)
```

```
78 try:
79     # Säkerställ att lampor är avstängd och ingen signal detektionssignal skickas
80     button_lamp.value(1)
81     varning_lamp.value(1)
82     sig.value(0)
83     # Slå av och på inbyggd LED:
84     led.value(1)
85     time.sleep(0.5)
86     led.value(0)
87     time.sleep(0.5)
88     while True:
89
90         while True:
91             print("-----", distance_1_2_3_4_5, "-----") # Skriver ut aktuellt värde för avståndsvariabeln
92             print("Free Area")
93             # Mät och skriv ut avståndet på sensorerna
94             distance1 = measure_distance(trigger1, echo1)
95             #print("Avstånd1:", distance1, "cm")
96             if(2 < distance1 < (45 + (distance_1_2_3_4_5 * 20))):
97                 break
98             distance2 = measure_distance(trigger2, echo2)
99             #print("Avstånd2:", distance2, "cm")
100            # Om avståndet är inom det otillåtna intervallet för larm, utför nödgärder
101            if(2 < distance2 < (45 + (distance_1_2_3_4_5 * 20))):
102                break
103
104            # Slå på varningslampan och skicka ut detektionssignalen
105            varning_lamp.value(0)
106            sig.value(1)
107            time.sleep(0.001)
108            # Säkra upp att arbetsområdet är fritt innan larmet slås av
109            while larm < 15:
110                print("Detection")
111                distance1 = measure_distance(trigger1, echo1)
112                distance2 = measure_distance(trigger2, echo2)
113                time.sleep(0.001)
114                # Om avstånden är utanför larmintervallet, öka larmräknaren
115                if distance1 > (45 + (distance_1_2_3_4_5 * 20)) and distance2 > (45 + (distance_1_2_3_4_5 * 20)):
116                    larm = larm + 1
117                    time.sleep(0.001)
118                # Om avstånden är inom larmintervallet, återställ larmräknaren
119                else:
120                    larm = 0
121                    time.sleep(0.001)
122                # Återställ larmräknaren när larmet är av
123                larm = 0
124                varning_lamp.value(1)
125                sig.value(0)
126
127 except KeyboardInterrupt:
128     print("\nAvslutar programmet.")
129
```

**INSTITUTIONEN FÖR ELEKTRONIK  
CHALMERS TEKNISKA HÖGSKOLA**

Göteborg, Sverige

[www.chalmers.se](http://www.chalmers.se)



**CHALMERS**