



CHALMERS



Ny funktion till smarta lastpallar

Mätning av massa och tyngdpunkt - konstruktion av tillhörande elektronik och prototyper

LUDVIG BRODÉN
JOSEFIN SVENSSON

EXAMENSARBETE 2019

Ny funktion till smarta lastpallar

Mätning av massa och tyngdpunkt -
konstruktion av tillhörande elektronik och prototyper

LUDVIG BRODÉN
JOSEFIN SVENSSON



CHALMERS
UNIVERSITY OF TECHNOLOGY

Institutionen för Elektroteknik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2019

Ny funktion till smarta lastpallar

Mätning av massa och tyngdpunkt -
konstruktion av tillhörande elektronik och prototyper

LUDVIG BRODÉN
JOSEFIN SVENSSON

Handledare: Björn Bergholm, Broccoli Engineering AB
Examinator: Göran Hult, Institutionen för Elektroteknik

Examensarbete 2019
Institutionen för Elektroteknik
CHALMERS TEKNISKA HÖGSKOLA
412 96 Göteborg
Telefon +46 31 772 1000

Framsida: Illustration av en smart lastpall som skickar data från lastgivare i pallen.

Typeset in L^AT_EX

Sammanfattning

Smarta lastpallar som hjälper företag att hålla reda på, spåra och övervaka gods blir allt vanligare. Det finns flera olika sorter och de varierar i vad de har för funktioner och format.

Den här rapporten behandlar framtagandet av en ny funktion i dessa lastpallar, att kunna veta vad det som står på en lastpall väger och hur vikten är fördelad. Mer specifikt undersöks det vad för elektronik som skulle krävas för att kunna avgöra det. Endast kommersiellt lättillgängliga komponenter övervägdes vid konstruktion och även om de med lägst pris valdes gjordes ingen kostnadskalkyl för att avgöra vad som var bäst lämpat.

Det undersöktes vilken typ av givare som skulle kunna passa för tillämpningen och hur den tillhörande elektroniken behövde fungera. Arbetet resulterade i en krets som tillsammans med en styrenhet skulle kunna göra den nya funktionen möjlig i en smart lastpall.

En stor del av arbetet gick dessutom ut på att ta fram demonstrationsprototyper för att visa ett användningsområde för funktionen. Dessvärre slutfördes inte detta inom tidsramen för projektet. Prototyperna skulle bestå av en truck som skulle ta emot data från en lastpall, för att kunna ge mer information om lasten till föraren och eventuellt kunna bidra till en säkrare transport.

Summary

Smart pallets that help companies in their material handling are becoming increasingly used in various industries. There are several different kinds of smart pallets, all of them with different functionalities and formats.

This report is about the development of a new feature in these pallets. The feature is being able to tell the mass of the goods on it and its weight distribution. More specifically, what kind of electronics would be needed to implement it. Only readily available electronics were considered during the construction. Even though the cheapest components were chosen for the project no further economic analysis was conducted.

Suitable sensors were chosen for the application and the electronics to make them work was constructed. The project resulted in a circuit that together with a control unit was able to utilize the sensors in a way that could be implemented in a smart pallet.

A big part of the time in the project was spent on developing demonstrational prototypes to show one of the areas of use. The prototypes were however not completed within the timeframe of the project. They were supposed to consist of a smart pallet and a forklift that got information about the goods to the driver to assist in safe transportation.

Förord

Detta är ett examensarbete som utförs av två högskoleingenjörsstudenter på mekatronikprogrammet. Arbetet utförs under institutionen för elektroteknik på Chalmers tekniska högskola och omfattar 15 högskolepoäng. Broccoli Engineering AB har agerat projektbeställare och deras kontor har varit huvudsakliga arbetsplatsen för projektet.

Vi vill ge ett stort tack till Björn Bergholm på Broccoli Engineering AB som har fått oss att känna oss välkomna på kontoret och gett oss förtroendet att genomföra projektet hos dem.

Ett stort tack går även till Göran Hult på institutionen för elektroteknik, examinatorn för projektet som även har handlett oss och givit oss goda råd under dess gång.

Ludvig Brodén
Josefin Svensson
Göteborg, juni 2019

Beteckningar

Smarta lastpallar - Lastpallar med inbyggda smarta funktioner så som RFID eller GPS

Raspberry Pi - Enkorts dator med ARM-arkitektur, likt den som finns i många smarta telefoner

Arduino - Mikrokontrollerkort

Pygame - Spelutvecklingsbibliotek för Python. Innehåller programbibliotek för datorgrafik.

QR-kod - En matris av punkter som läses av med kamera och som tolkas av en processor som översätter dessa till data

Lastcell - Givare som reagerar på laster. De är uppbyggda med trådtöjningsgivare inuti dem.

ISM-bandet - Beteckningen står för "Industrial, Scientific and Medical-bandet, ett licensfritt frekvensband

Xbee - Radiokommunikationsmodul. Använder frekvensen 2,4 GHz

LoRa - Beteckningen står för Long Range, en trådlös teknik för överföring av data

instrumentförstärkare - Krets som förstärker spänningsskillnader från bland annat trådtöjningsgivare

Wheatstonebrygga - Krets som används för att mäta förändringen i resistans hos trådtöjningsgivare

A/D-omvandlare - Krets som omvandlar en analog signal till motsvarande digitala signal

Excitationsspänning - Den spänning som appliceras på en brygga av givare

Qi-laddning - Trådlös induktiv laddning som följer Qi-standarden.

IoT - Internet of things

SPI - Ett kommunikationsprotokoll som ofta används ihop med mikrokontrollers

Multiplexer - Riktar om signaler och kan användas för att exempelvis läsa av flera analoga signaler med färre A/D-omvandlare

EUR-pall - Lastpall i trä som följer standarden för EUR-pallar

Innehåll

1	Inledning	1
1.1	Bakgrund	1
1.2	Syfte	2
1.3	Avgränsningar	2
1.4	Precisering av frågeställning	3
2	Teoretisk referensram	4
2.1	Befintliga smarta pallar	4
2.1.1	LogTrade - The Smart Pallet	4
2.1.2	Palletech - molnansluten pall	4
2.1.3	SMART Logistics - Smart pallets	4
2.2	Force sensing resistor - FSR	4
2.3	Lastceller	5
2.4	Kommunikation	5
2.4.1	Serial Peripheral Interface SPI	6
2.4.2	Trådlös kommunikationsmodul	6
2.4.3	Radio Frequency Identification RFID	6
3	Metod	7
3.1	Förstudie	7
3.1.1	Urval av givare	7
3.1.2	Framtagning av förstärkarkrets	7
3.1.3	Val av strömförsörjning	8
3.1.4	Val av kommunikationsmetod	8
3.1.5	Presentation av information	8
3.1.6	Energisparläge för pallen	9
3.1.7	Anslutning till pallen	9
3.1.8	Framtagning av prototyp	9
3.1.9	Prototypernas funktion	9
4	Elektronik	10
4.1	Lastceller	10
4.2	Multiplexer	13
4.3	Instrumentförstärkare	14
4.4	Hela givarkretsen	15
4.5	Programkod	16

5	Framtagning av prototyper	19
5.1	Pallen	19
5.1.1	Konstruktion av hårdvara	19
5.1.2	Styrenhet - Arduino	20
5.1.2.1	Energisparkrets	20
5.1.2.2	Batteri och laddkrets	21
5.1.3	Övrig mjukvara i pallen	22
5.1.3.1	Trådlös kommunikation	22
5.1.3.2	Avstängning och energi	22
5.2	Trucken	22
5.2.1	Hårdvara	22
5.2.1.1	Kommunikationsenhet - Arduino	23
5.2.1.2	Grafikenhet - Raspberry Pi	26
5.2.2	Mjukvara i trucken	26
5.2.2.1	Avläsning av pallens ID och kommunikation	26
5.2.2.2	Grafik och avkodning av data	28
6	Resultat	29
6.1	Elektroniken	29
6.2	Demonstrationsprototyperna	29
6.2.1	Pallen	29
6.2.2	Trucken	30
7	Slutsats	31
7.1	Utmaningar under arbetets gång	31
7.2	Fortsatt utveckling	32
	Referenser	33

1

Inledning

I följande kapitel presenteras bakgrunden och syftet till arbetet. Utöver det kommer även projektets avgränsningar och frågeställningar att specificeras.

1.1 Bakgrund

Många industrier hanterar en stor mängd gods, både i form av intern såväl som extern transport. Ett vanligt sätt att transportera gods på är med lastpallar och den klassiska EUR-pallen är bland den vanligaste typen. Att hålla reda på alla lastpallar, se till att gods kommer fram säkert och vilket gods som är på vilken pall kan för många stora företag vara en utmaning som sällan kommer som en självklarhet. Det är några av anledningarna till att det blivit allt vanligare med smarta lastpallar.

Smarta lastpallar finns i flera utföranden, vissa är helt gjorda i plast med flera olika typer av sensorer, medan andra liknar EUR-pallar med enbart inbyggd RFID-teknik (radio frequency identification). Vissa kan följa transporten av gods, logga temperaturen i omgivningen av pallen eller larma vid skakningar medan andra enbart ger information om vad som finns på dem.

Något som inte är vanligt förekommande eller inte förekommer alls i smarta lastpallar idag är möjligheten att få information om hur mycket godset på lastpallen väger eller var tyngdpunkten befinner sig. Den informationen skulle bland annat kunna användas för att spåra tillgångar hos företag, lasta lastpallar tillsammans på ett mer effektivt sätt eller avgöra vilket sätt som är säkrast att lyfta en lastpall på. I det här projektet kommer möjligheterna att implementera dessa funktioner i smarta lastpallar att undersökas. Prototyper kommer konstrueras för att visa hur det skulle kunna realiseras.

Projektet utförs i samarbete med Broccoli Engineering AB. Det är ett konsultföretag som inriktar sig mot elektronik och inbyggda system inom bilindustrin. Eftersom företaget uttrycker ett stort intresse för att undersöka annorlunda tekniska frågeställningar och till stor del sysslar med inbyggda system ansågs projektet lämpligt att utföras där.

1.2 Syfte

Arbetet fokuseras på att undersöka möjligheten att i dagens smarta lastpallar implementera ytterligare funktioner så som att ge användaren information om hur mycket lasten väger och att beräkna dess tyngdpunkt. Det finns flera användningsområden för den här nya funktionen och däribland

- Effektivare lastning
 - Om vikten och tyngdpunkten skulle vara känd på en grupp lastpallar som ska lastas tillsammans skulle det kunna bidra till en effektivare transport.
- Inventeringsverktyg
 - Om alla pallar på ett lager skulle vara anslutna i någon form av nätverk så skulle det gå att se exakt hur mycket som fanns kvar av ett visst material eller en produkt.
- Säkrare hantering med truck
 - Genom att informera truckförare om vad som är på en lastpall, vad det väger och vart dess tyngdpunkt befinner sig skulle det förmodligen gå att minska olyckor vid den typen av hantering.

Projektet riktas in på att ta fram den bakomliggande elektroniken till den nya funktionen och att demonstrera användningsområdet som skulle kunna bidra till en säkrare hantering med truck.

1.3 Avgränsningar

Följande avsnitt beskriver de avgränsningar som gjordes i arbetet.

De prototyper som konstrueras är inte färdiga produkter och ämnar bara visa ett av användningsområdena till den nya funktionen.

Vid uppbyggnad av prototyper kommer liten hänsyn tas till material- och tillverkningskostnader. Detta då smarta lastpallar finns i många olika utföranden både vad gäller dimensioner, material och metod vid tillverkning.

Vid framtagning av elektroniken till arbetet kommer endast lättillgängliga komponenter att övervägas. Eftersom det endast är prototyper som ska konstrueras och det måste gå att införskaffa delarna inom tidsramen för projektet. Det kan innebära att det inte blir de mest passande delarna som används utan snarare de som fungerar bäst av de vanligast förekommande.

Ingen undersökning i huruvida det faktiskt blir säkrare att visa information om lasten för truckförare kommer utföras.

1.4 Precisering av frågeställning

Initialt i projektet finns det ett antal frågeställningar som är avgörande för hur arbetet kommer att fortgå. De grundläggande frågeställningarna är:

- Vilken typ av givare bör användas för att kunna mäta lasten och tyngdpunkten på pallen?
- Hur ska elektroniken som styr och läser av givarna fungera?
- Vilken information ska skickas mellan pall- och truckprototyperna?
- Vilken kommunikationsmetod ska användas för att kunna skicka informationen?
- Hur ska strömförsörjningen fungera?
- Vilken information ska presenteras för truckföraren och hur?

Det finns ytterligare frågeställningar som kan komma att behandlas, men som inte kommer att tas hänsyn till under det övriga arbetet utan endast i mån av tid:

- Skulle det kunna vara kostnadsmässigt effektivt att implementera funktionerna i framtida smarta lastpallar?
- Hur skulle en smart lastpall med denna typen av elektronik på ett bra sätt skyddas mot variationer i sin omgivning som exempelvis fukt, temperatur och mekaniska stötar?

2

Teoretisk referensram

I följande kapitel förklaras den tekniska bakgrunden som krävs för att få en större förståelse för de ingående delarna i projektet och hur genomförandet har gått till.

2.1 Befintliga smarta pallar

Smarta lastpallar finns i flera utföranden, med olika funktionalitet. Här presenteras tre befintliga smarta pallar för att ge en bild av hur de kan skilja sig åt.

2.1.1 LogTrade - The Smart Pallet

Pallen är försedd med en QR-kod innehållandes ett unikt ID. Den ger alla som vill möjlighet att spåra pallen. Under pallen är en centralenhet för Bluetooth-kommunikation monterad. Det gör det möjligt att känna av gods försedda med Bluetooth-taggar och den kan avgöra hur många enheter som är på pallen[1].

2.1.2 Pallettech - molnansluten pall

Med sensorer placerade i pallens mittbräda inhämtar pallen data i realtid. Den information som kan ges till användaren är bland annat temperatur och fuktighet i omgivningen samt var godset befinner sig[2].

2.1.3 SMART Logistics - Smart pallets

Den här lastpallen kan påstås vara smart både i hur den är uppbyggd och vad den har för teknik inuti sig[3]. Den är byggd för att hålla längre än en vanlig EUR-pall och har flera IoT-funktioner. Det går bland annat att spåra pallen och med hjälp av ett databassystem övervaka företags gods.

2.2 Force sensing resistor - FSR

FSR är ett tryckkänsligt motstånd vars resistans varierar med applicerad kraft. Motståndet består av en tunn film gjord av en polymer med elektriskt ledande partiklar. Partiklarna är väldigt små för att minska motståndets temperaturberoende och för att förbättra de mekaniska egenskaperna hos materialet.

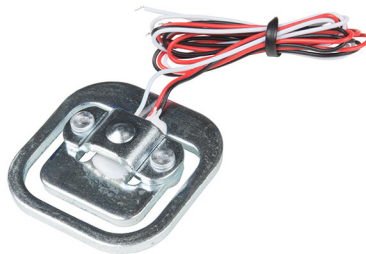


Figur 2.1: Figuren visar ett tryckkänligt motstånd, där den runda delen är den aktiva delen av givaren[4]

Några av fördelarna med FSR är att de är billiga i jämförelse med andra typer av trycksensorer, som exempelvis lastceller och är tåliga för yttre påfrestningar. Några av nackdelarna är att de har en relativt låg precision och har inte en linjär karaktäristik [5].

2.3 Lastceller

Lastceller finns i flera utföranden, dock är det vanligast att de är uppbyggda av trådtöjningsgivare kopplade som en wheatstonebrygga. En lastcells funktion är att den översätter och omvandlar applicerad kraft till en mätbar elektrisk utsignal[6, 7].



Figur 2.2: Lastcell med två aktiva givare och en kapacitet på 50 kg[8]

Lastceller kan vara relativt billiga och har ofta hög noggrannhet. Detta gör att lastceller används i många olika applikationer.

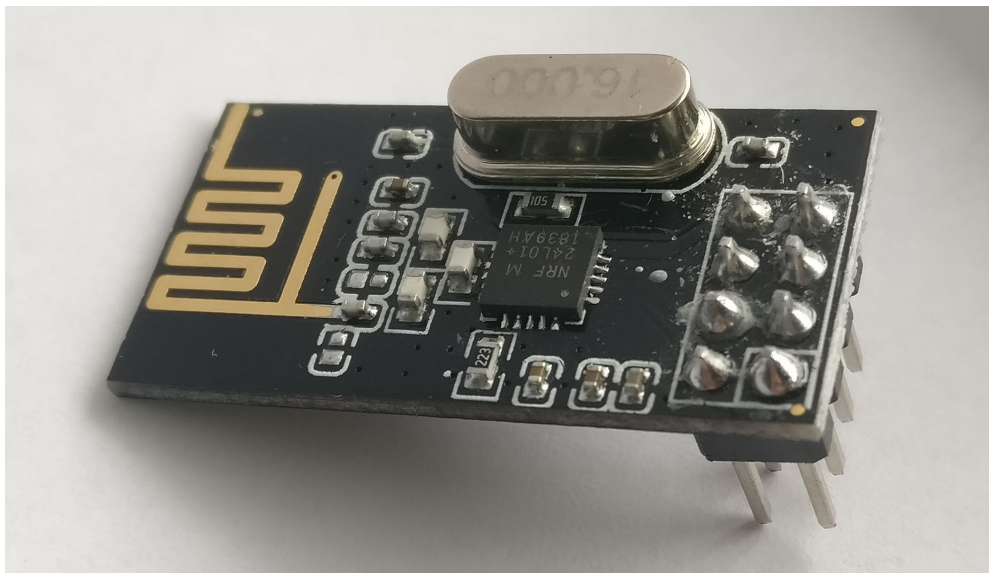
2.4 Kommunikation

Här beskrivs den kommunikationsteknik som kan vara bra att ha kännedom om för att bättre förstå denna rapport.

2.4.1 Serial Peripheral Interface SPI

Ett kommunikationsprotokoll som används för kommunikation mellan bland annat mikrokontrollers. SPI är ett synkront master-slave-baserat gränssnitt, där kommunikationen kan ske i båda riktningarna samtidigt.[9]

2.4.2 Trådlös kommunikationsmodul



Figur 2.3: Bild på den typen av NRF-modul som användes

NRF24L01 är en krets som förekommer på trådlösa kommunikationsmoduler. Den skickar radiosignaler på 2,4 GHz, vilket faller under ISM-bandet och har samma frekvensområde som Wi-Fi. Kommunikation med modulen sker med SPI och en av fördelarna med att använda NRF24L01 är att den, jämfört med många andra kommunikationsmoduler, har låg strömförbrukning både vid sändning, mottagning och då modulen är i viloläge[10].

2.4.3 Radio Frequency Identification RFID

Ett RFID-system använder sig av trådlös radiokommunikation för att identifiera objekt försedda med en RFID-tag. Systemen är uppbyggda av tre komponenter: en tagg, en läsare och ett datorsystem som styr avläsningen [11].

RFID används i bland annat busskort, vid avläsning av gods i t.ex hamnar eller i passersystem.

3

Metod

3.1 Förstudie

I projektets början var det betydelsefullt att strukturera upp och planera för hur arbetet skulle genomföras för att få en tydlig bild över de ingående delarna. Det innefattade att hitta alternativ till hur kretsen skulle konstrueras och avgöra vilka funktioner prototyperna skulle ha för att veta vilka komponenter som behövs till dessa. Det var även viktigt att avgöra vilka delar som behövde göras före andra och med det ta hänsyn till komponenternas leveranstid.

Det var även väsentligt att ställa upp en kravspecifikation för att bestämma hur systemet skulle fungera för att nå de uppsatta målen i projektet.

3.1.1 Urval av givare

Det konstaterades att det skulle behövas minst en givare i varje kontaktpunkt i en smart lastpall för att mäta massa och tyngdpunkt. Detta eftersom det skulle kunna gå nästan vilken kraft som helst i de punkter som lasten inte mäts i. Därför inleddes projektet med att undersöka vilka olika typer av lastgivare som fanns att tillgå. I urvalet togs störst hänsyn till priset men även givarnas storlek och hur det skulle gå att bygga in dem i smarta lastpallar. De kandidater som lämpade sig bäst i den här tillämpningen beställdes för att vidare kunna undersökas. Dessa givartyper blev FSR och lastceller, FSR på grund av dess minimala utformning och pris, lastceller på grund av dess precision och på en av modellerna även priset. En rigg konstruerades för att kunna testa givarnas funktion genom att anbringa olika laster. Efter några tester blev det tydligt att FSR inte var tillräckligt tillförlitlig och arbetet fortsattes med lastceller.

3.1.2 Framtagning av förstärkarkrets

För att kunna läsa av spänningsskillnaderna vid belastning av lastcellerna med högre upplösning i A/D-omvandlare var en förstärkarkrets nödvändig. Eftersom det behövs minst en givaranordning i varje kontaktpunkt på pallen skulle det krävas många förstärkarkretsar om det kopplas en till varje lastcell. Således valdes att använda multiplexers för att kunna använda en och samma förstärkarkrets till flera givaranordningar.

3.1.3 Val av strömförsörjning

Då lastpallar självklart bör vara trådlösa undersöktes två möjliga strömförsörjningsmetoder. De metoder som det stod mellan var att pallen kunde drivas av trucken direkt eller att pallen skulle vara batteridriven.

Att driva kretsen i pallen med batterier valdes eftersom det då blev möjligt att få information från pallen innan trucken får kontakt och lyfter den. Det stod mellan att konstruera batteripack med uppladdningsbara litium-jon batterier eller alkaliska batterier. Arbetet fortsattes med litium-jon men med alkaliska batterier som reservlösning.

3.1.4 Val av kommunikationsmetod

Vid val av vilken kommunikationsmetod som skulle användas undersöktes olika typer av metoder för överföring av data, även här togs i åtanke att systemet bör vara trådlöst och därav togs ett beslut om att använda trådlös kommunikation. För att avgöra vilken typ av trådlös kommunikation som bäst lämpade sig för ändamålet med de krav som ställdes på systemet gjordes ett urval mellan olika tekniker.

I urvalet undersöktes ett antal olika vanligt förekommande tekniker för överföring av information trådlöst, utifrån faktorer som var väsentliga för ändamålet och hur resterande delar av systemet byggdes upp. Då faktorerna i urvalet kan skilja sig mellan olika tillverkare och utförande gjordes en skattning mellan teknikerna, se tabell 3.1

Tabell 3.1: Urval av trådlös kommunikation.

Teknik	Räckvidd	Strömförbrukning	Pris
WI-FI	Lång	Hög	Medel
Blue-tooth	Kort	Medel	Lågt
ZigBee	Medel	Medel	Lågt
LoRa	Lång	Hög	Medel
NRF24L01	Medel	Låg	Lågt

Eftersom elektroniken i pallen skulle drivas av batteri var strömförbrukningen den faktor som avgjorde vilken teknik som användes. NRF24L01 var den modul med lägst strömförbrukning, lägst pris och som fanns lättillgänglig, därav valdes den som kommunikationsmodul för överföring av data.

3.1.5 Presentation av information

En utvärdering gjordes för att bestämma vilken information som skulle ges till användaren och hur den skulle kunna presenteras. Eftersom det inte ännu var helt klart vad som skulle gå att realisera så valdes en TFT (Thin-film-transistor) LCD-skärm för att med enkelhet kunna modifiera vad som presenteras för användaren.

3.1.6 Energisparläge för pallen

När pallen inte är i närheten av en truck är det fördelaktigt att minska strömförbrukningen eftersom den inte används. Således undersöktes olika möjligheter att realisera ett energisparläge, det vill säga att pallens enhet stänger av sig då den inte är ansluten till trucken eller inte varit det under ett visst tidsintervall.

3.1.7 Anslutning till pallen

Trucken skulle kunna identifiera och ansluta till pallen då den var tillräckligt nära. Därav krävdes en teknisk lösning för att läsa av pallen. Den teknik som valdes var RFID på grund av att det är relativt lätt att implementera och även lättillgängligt.

3.1.8 Framtagning av prototyp

Som tidigare nämnts finns smarta pallar i flera olika utföranden och därför togs liten hänsyn till pallens dimensioner eller val av material vid konstruktionen. Något som däremot togs större hänsyn till var antalet kontaktpunkter som prototypen skulle ha. Det var nödvändigt eftersom det måste sitta givare i varje kontaktpunkt för att kunna avgöra den totala lasten på pallen. Utifrån det togs beslut om att konstruera en så kallad halv EUR-pall med fyra kontaktpunkter. Som material till pallen valdes trä då det är lätt att bearbeta för hand och på så sätt bygga in elektroniken.

3.1.9 Prototypernas funktion

En del i framtagningen av prototyperna var att avgöra vilka funktioner systemet skulle ha för att demonstrera möjligheter med den nya funktionen. Funktionerna valdes så att de dels visar ett specifikt användningsområde för den typen av mätningar men även generellt vad smarta pallar kan användas till. Nedan följer en lista med de valda funktionerna

- Föraren ska kunna se informationen från pallen på en skärm i trucken. På skärmen ska följande information visas
 - Siffervärden på var pallens tyngdpunkt befinner sig i ett koordinatsystem
 - Total last på pallen
 - Pallens ID
 - Pallens batterinivå
 - Typ av gods på pallen
 - Om det är tillåtet att lyfta lasten med den beräknade tyngdpunkten
- När pallen inte varit ansluten till en truck under en viss tid ska den koppla från.
- Pallen ska kunna laddas trådlöst av trucken. Att pallen börjar laddas är även vad som startar pallen.
- Pallen ska vara helt trådlös.
- Trucken ska kunna identifiera vilken pall det är den är på väg att lyfta och ansluta till den.

4

Elektronik

I det här kapitlet beskrivs framtagandet av elektroniken som ska kunna mäta krafterna i varje kontaktpunkt i smarta lastpallar. Mätvärdena från kretsen ska sedan kunna användas för att beräkna total last och tyngdpunkt.

4.1 Lastceller

Givarna som används är lastceller som är vanligt förekommande i personvagnar och annan elektronik där tryckande krafter ska mätas och då givaren måste vara kompakt i höjdlädd, se figur 2.2. Lastcellerna är försedda med två trådtöjningsgivare i serie fastlimmade. Tre kablar är dragna från givarna, en i var ände av givarna och en mellan dem, se figur 4.1.

Tabell 4.1: Lastcellernas specifikationer

Kapacitet	40-50 kg
Maximal matningsspänning	10 V
Utsignalsspänning vid maxlast	1 mV/Vmatning
Maximal Utsignalsspänning vid maxlast	10 mV
Överbelastningskapacitet	150%
Olinearitet	0,03 %
Hysteres	0,03 %
Repeterbarhet	0,03 %
Krypning (creep på engelska) efter 3 minuter	0,03 %

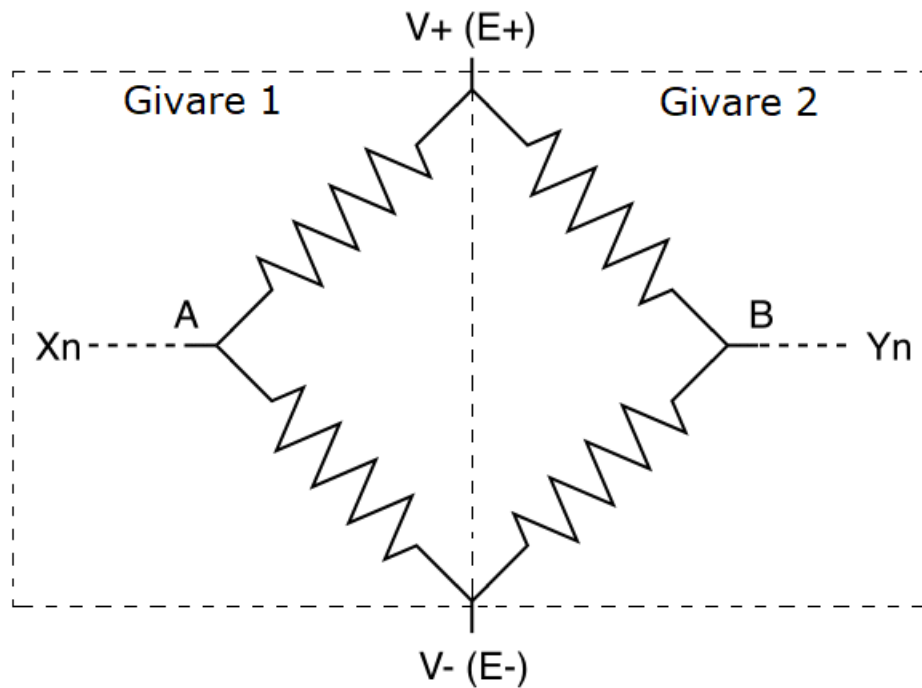
Två givare placerades i varje kontaktpunkt. Det gör att lastkapaciteten och den maximala utsignalsspänningen kan fördubblas men då krävs att givarna monteras på ett jämnt sätt och att de kopplas ihop så att de båda bidrar till en ökad spänningsskillnad, se figur 4.1 och 4.2

Givarna kan som mest få en spänning på utsignalen på 10 mV när excitationsspänningen är 10 V. När två givare kopplats ihop till en helbrygga dubblas maximala spänningen på utsignalen till 20 mV. Det står specificerat att givarna kan överbelastas med 150 %, men det undersöktes inte hur givarnas utsignal reagerade på det och därför hölls belastningen inom kapacitetsområdet.

Om Olineariteten, hysteresen och repeterbarheten läggs samman erhålls det max-

imala felet på ungefär 0,1 %. Det stod specificerat i databladet att spänningen dessutom kan krypa 0,03 % efter 3 minuter. Den noggrannheten ansågs fullgod för tillämpningen.

Alla lastcellerna skilde sig från fabrik i hur stor resistans dess givarna hade och ibland med mer än 0,2 % från nominell resistans. Därför uppmättes resistansen i givarna och de som låg närmast nominell resistans parades ihop. Det gjorde att utsignalen givarna hade utan belastning kom närmare 0 V.



Figur 4.1: lastcellernas konfiguration

För att givarna ska påverkas på det sätt som tillverkarna avsett krävs att de genomgående krafterna appliceras på mittendelen av givaren och att den kan böjas nedåt. Därför konstruerades ett hölje som ger utrymme för givarna att röra sig och att trycket hamnar på rätt ställe, se figur 4.2 nedan.



Figur 4.2: Bild på ett demonstrationsexemplar av givarna

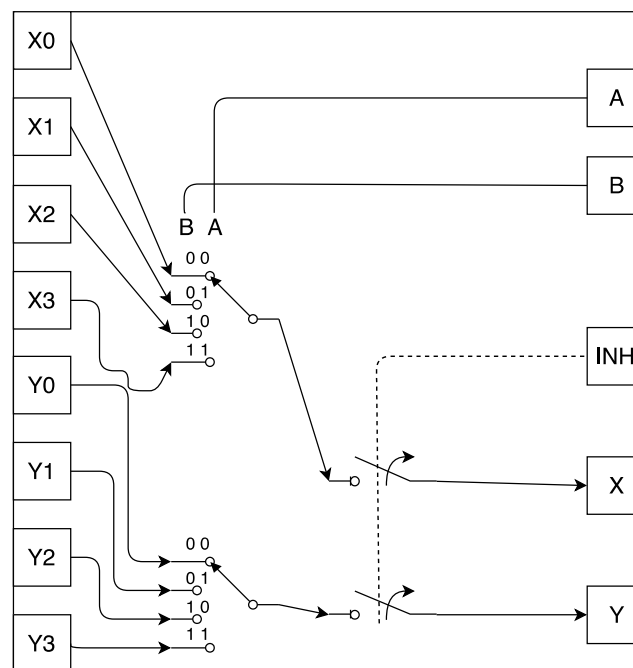
Givarna limmades fast inuti klossen på pallan. Eftersom pallar i huvudsak utsätts för kompression räcker det med att klossarna sitter fast och håller för sin egenvikt, se figur 4.3



Figur 4.3: Illustration av hur givarna placerats i klossen på en pall

4.2 Multiplexer

Multiplexern som valdes för att rikta om signalerna från givarna heter CD4052BE. Ett urval gjordes med hänsyn till pris, tillgänglighet och funktion. Just den här multiplexern har som funktion att den riktar om och släpper genom två signaler samtidigt, se figur 4.4. Det gör att det bara behövs en för att båda signaler från ett givarpar ska gå att skicka till en instrumentförstärkare. Risken för störningar på signalerna skulle kunna minskas när de behandlas samtidigt av samma krets. Det fanns fler multiplexers med denna funktionen men denna var en av de billigare och hade jämförelsevis bra dokumentation.

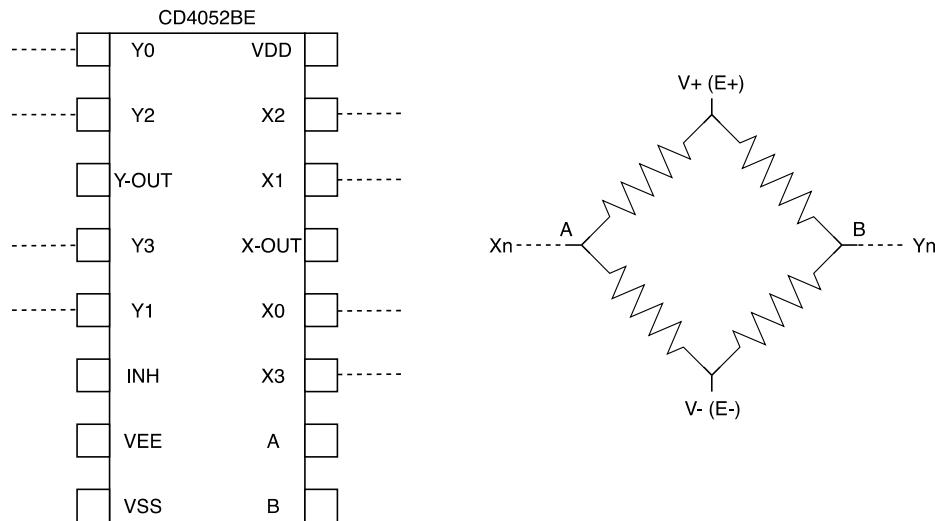


Figur 4.4: Multiplexer CD4052BE

Nedan följer en kort förklaring på vad de olika benen gör på multiplexern, se figur 4.5.

- X_n och Y_n
 - Dessa ben är kopplade till givarparens utsignaler, en kanal svarar alltså för ett givarpar.
- X-OUT och Y-OUT
 - Detta är utgångarna från multiplexern som senare kopplas till instrumentförstärkarens ingångar.
- A och B
 - Kanal på multiplexern väljes med dessa ben. Binärkombinationer av talen 0-3 ger respektive kanal med A som den minst signifikanta biten.
- VDD, VSS och VEE
 - VDD är matningsspänningen till multiplexern, ansluts lämpligen direkt till ett batteri eller annan stabil spänningskälla. VSS och VEE ansluts båda till 0 V.

- INH
 - INH står för engelskans 'inhibit' och sätts det benet till en digital etta stängs alla kanaler i multiplexern av. Den anslöts i det här fallet direkt till 0 V för att alltid vara på.



Figur 4.5: Givarna och multiplexer

4.3 Instrumentförstärkare

Eftersom spänningsskillnaden på utsignalskablar är så låg som den är behöver den förstärkas för att enklare läsas av en A/D-omvandlare. Den krets som valdes är en instrumentförstärkare och heter INA126P. Den har åtta ben med följande funktioner, se även figur 4.6.

- RG
 - Genom att variera resistansen mellan de båda RG-benen justerar man förstärkningen genom kretsen. I det här fallet anslöts en trim-potentiometer för att kunna justera den.
- V+in och V-in
 - Det är till dessa ben som utsignalerna från multiplexern ansluts och som sedan förstärks.
- Vout
 - Från det här benet kommer utsignalen från kretsen som sedan kan läsas av en A/D-omvandlare
- V+ och V-
 - Detta är positiv och negativ matningsspänning, det är vanligt att ha både positiv och negativ matning till förstärkare men i det här fallet anslöts den negativa sidan till 0 V
- REF
 - REF står för 'reference' vilket är referensspänningen. Den används för att skifta den undre gränsen för utsignalområdet uppåt.

För att ställa in förstärkningen på instrumentförstärkaren krävs det att högsta möjliga spänning beräknas så att den inte blir för hög på A/D-omvandlingång. Det givarparet med högst fel kopplades in och belastades med maximal last, samtidigt som spänningen på utgången mättes. Detta gjordes utan att ha mikrokontrollern ansluten för att undvika skador på den. Förstärkningen justerades med hjälp av trim-potentiometern och när spänningen låg strax under maximala spänningen på ingången på A/D-omvandlaren ansågs det vara en bra förstärkning.

För att på förhand veta en ungefärlig resistans på förstärkaringången användes följande formel från databladet

$$G = 5 + \frac{80 \text{ k}\Omega}{R_G}$$

För att få ut resistansen krävs att den önskade förstärkningen är känd. Den går att få fram om den maximala utsignalen från givarna uppskattas. Om den exempelvis uppskattas vara 25 mV och maximala spänningen till A/D-omvandlaren är 5 V fås

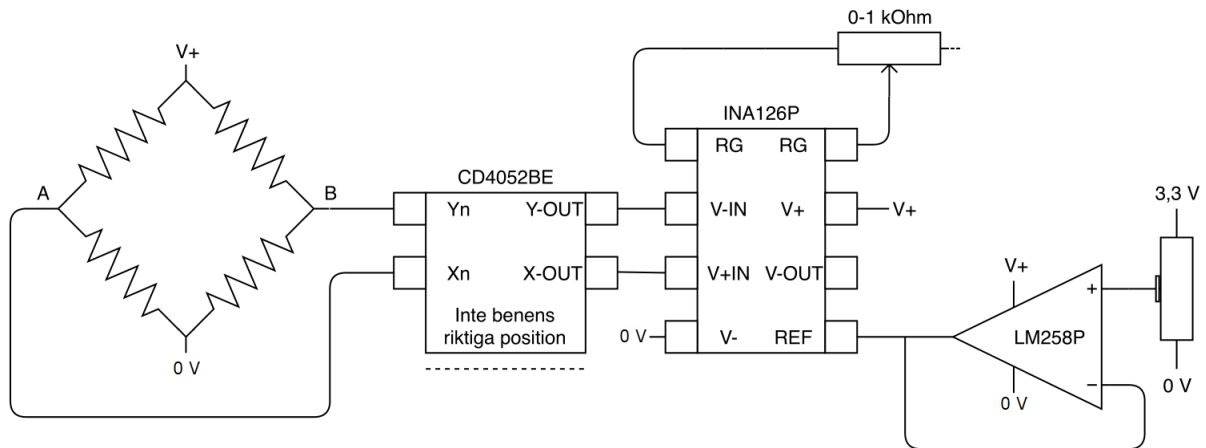
$$G = \frac{5 \text{ V}}{25 * 10^{-3} \text{ V}} = 200$$

$$R_G = \frac{80 \text{ k}\Omega}{G - 5} = \frac{80 * 10^3}{200 - 5} = 410 \text{ }\Omega$$

Det är rekommenderat att koppla referensspänningen till en förstärkare konfigurerad som en spänningsföljare. Spänningskällan till referensspänningen får då låg inre resistans, vilket den inte hade fått med exempelvis en spänningsdelare. Enligt databladet krävs en skillnad mellan negativ matningsspänning och utsignal på minst 0,95 V. Därför måste referensspänningen ställas in på minst 0,95 V eftersom matningsspänningen, i det här fallet, är 0 V. En trimpotentiometer kopplades till spänningsföljaren för att lättare kunna manipulera den under felsökning.

4.4 Hela givarkretsen

För att få en tydligare bild av hur hela givarkretsen ser ut se figur 4.6. Givarparens utsignaler kopplas till multiplexern, de riktas om till instrumentförstärkaren och där förstärks för avläsning.

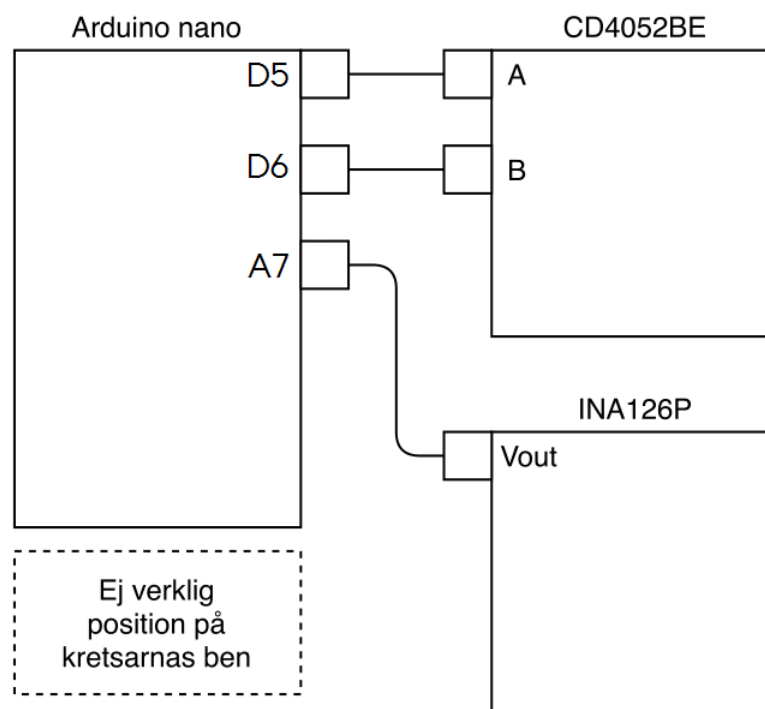


Figur 4.6: Förenklad version av förstärkarkretsen

4.5 Programkod

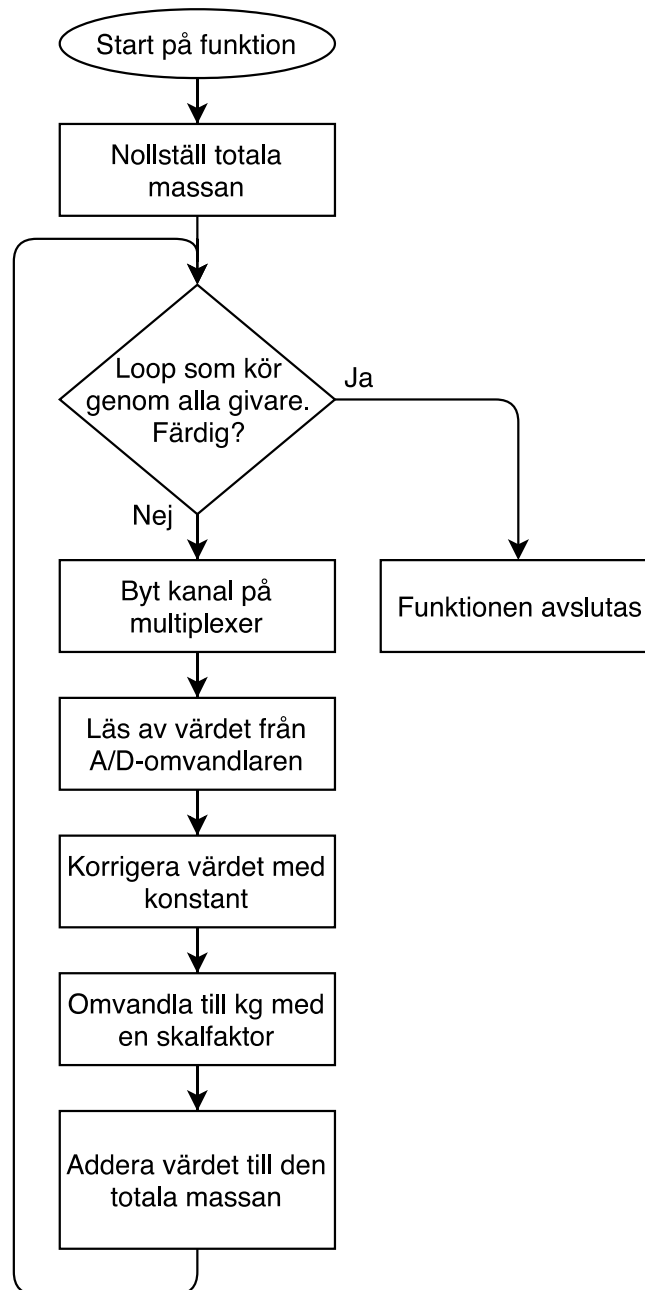
I följande avsnitt förklaras den programkod som behövs för att styra kretsen till givarna och omvandla värdena från dem till massa och tyngdpunkt.

En mikrokontroller anslöts för att välja kanal på multiplexern. En Arduino nano valdes på grund av dess kompakta format och låga pris. Två digitala ben anslöts till kanalbenen på multiplexern och ett analogt ben till utgången från instrumentförstärkaren, se figur 4.7.



Figur 4.7: Förenklad version av kopplingarna mellan givarkretsen och mikrokontrollern

För att kunna räkna ut massan och tyngdpunkten med de erhållna värdena från givarna krävs ett program. Se flödesschema, figur 4.8, med förklaring nedan.



Figur 4.8: Flödesschema till funktion som uppdaterar mätvärden från givarna

Programmet i mikrokontrollern kör en funktion som omvandlar givarnas värden till antal kg last i varje punkt på pallen. Funktionen börjar med att nollställa massan från förra körningen och loopar sedan genom alla givare och läser in nya värden. I loopen, efter läsningen från A/D-omvandlaren, omvandlar den värdena till kg genom att först ta bort en parameter för att sedan skala om värdena med en faktor som kalibreras in i förhand. Parametern som subtraheras från värdena består dels av en konstant för att parera felet från givarna i obelastat läge men även av bidraget från

referensspänningen. Parametern behöver kalibreras för varje givarpar.

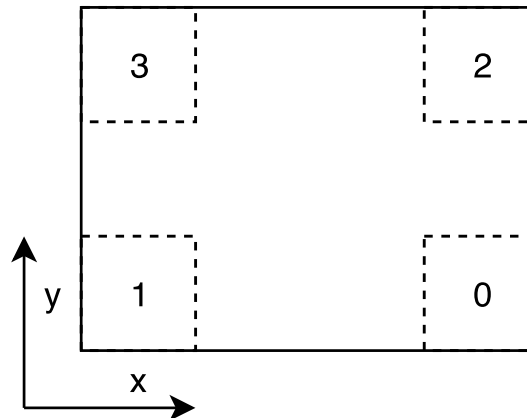
En av kontaktpunkterna är fördefinierad med koordinaterna (0,0) och kan uteslutas vid beräkning av tyngdpunkt eftersom avståndet då blir 0 till origo. Kontaktpunkt nummer 1 valdes till att vara origo, se figur 4.9 för numrering. För att beräkna tyngdpunkten används följande formler

Tabell 4.2: Variablerna som används i formel för tyngdpunksberäkning

$\bar{x}$	Tyngdpunktens x-koordinat
$\bar{y}$	Tyngdpunktens y-koordinat
$\mathbf{g_x}$	Massan från givare, numrering enligt figur 4.9
$\mathbf{L_x}$	Pallens längd i x-led
$\mathbf{L_y}$	Pallens längd i y-led
$\mathbf{m_{tot}}$	Den sammanlagda massan

$$\bar{x} = \frac{g_0 * L_x + g_2 * L_x}{m_{tot}}$$

$$\bar{y} = \frac{g_3 * L_y + g_2 * L_y}{m_{tot}}$$



Figur 4.9: Numrering av kontaktpunkterna på pallen

5

Framtagning av prototyper

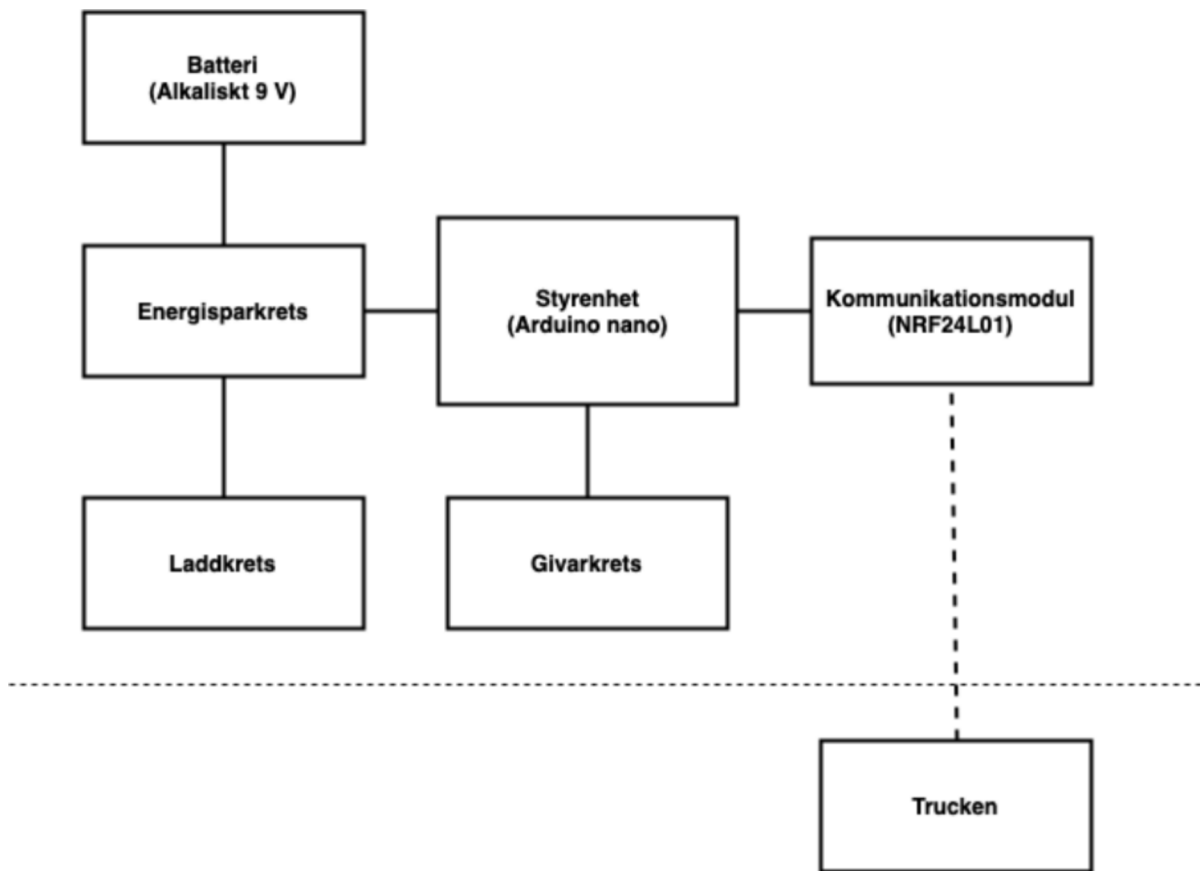
I detta avsnitt beskrivs det praktiska genomförandet i projektet. Detta innefattar framtagning av hårdvaran samt utvecklingen av mjukvaran för de olika systemen.

5.1 Pallen

Pallen är den enhet där lastceller skulle byggas in för att demonstrera hur det skulle kunna realiseras i smarta lastpallar. Här nedan beskrivs konstruktionen av systemets övriga kretsar samt utvecklingen av mjukvaran som hanterar de funktioner som specificerades under projektets förstudie.

5.1.1 Konstruktion av hårdvara

Pallens hårdvara byggdes upp av en Arduino nano som är den enhet som hanterar de mätvärden som erhålls från avläsning av givare samt beräkning av tyngdpunkten. Styrenheten hanterar även överföring av data till trucken via den trådlösa kommunikationsmodulen NRF24L01 och den krets som konstruerades för att batteriets energi skulle räcka längre.



Figur 5.1: Blockschema över prototypens hårdvara

Under arbetet med uppbyggnaden utav de kretsar som skulle användas kopplades komponenterna upp på ett kopplingsdäck för att kunna testa och säkerställa deras funktion. Då funktionerna var säkerställda monterades kretsarna på ett prototypkort.

5.1.2 Styrenhet - Arduino

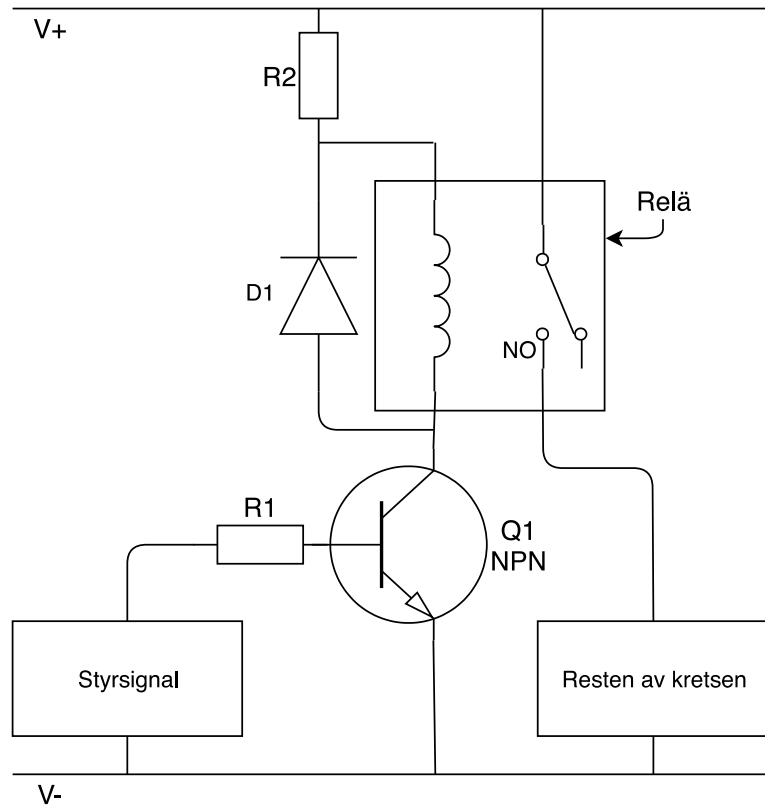
Som styrenhet användes mikrokontrollerkortet Arduino nano. Arduino nano har 14 I/O-portar och 8 analoga ingångar med en matningsspänning på 5-12 V [13]. En av anledningarna till att Arduino nano valdes var för att det är ett litet kort som passar bra för lödning på kretskort.

5.1.2.1 Energisparrets

Eftersom pallprototypens spänningskälla är batterier ansågs det lämpligt att konstruera en energisparrets som gör att pallens enhet stängs av helt när den varit inaktiv under en viss tid. Nedan beskrivs kretsen och hur den fungerar.

När pallen kommer tillräckligt nära laddningskretsen i trucken induceras en spänning som i figur 5.1 heter 'Styrsignal'. Signalen går via ett strömbegränsande motstånd (R1) till en NPN-transistor (Q1) som när pallen inte 'laddas' stoppar ström från att gå genom den. När transistorn väl släpper genom ström gör det möjligt för ett relä

att slå om och släppa genom ström till resten av kretsen i pallen. Reläet har en diod (D1) parallellkopplad över sin spole för att minska spänningsspiken som uppstår när transistor slutar släppa genom ström. Reläet som användes i kretsen hade en spole som drevs av 5 V och eftersom spänningskällan från batteri kunde vara högre än det kopplades ett motstånd (R2) i serie med den för att inte få en allt för hög spänning vid full laddning. När kretsen väl fått ström från batteriet håller den igång sig själv genom att hålla styrsignalen fortsatt hög tills pallen ska stängas av.



Figur 5.2: Förenklat diagram som visar hur energisparkretsen är uppbyggd

5.1.2.2 Batteri och laddkrets

För att hålla projektet inom tidsramarna togs ett beslut om att använda ett alkaliskt batteri men att inkludera en laddkrets för att visa hur det skulle kunna fungera med uppladdningsbara batterier. Som laddkrets för systemet användes induktiv laddning och närmare bestämt Qi-laddning. Qi valdes eftersom det finns färdiga moduler för enkel implementering.

Mottagarenheten för laddkretsen användes som en slags givare för att se när pallen skulle ha laddats och kopplades till styrsignalen för energisparkretsen. Det alkaliska batteri som användes hade en spänning på 9 V och kopplades direkt in till kretskortets positiva spänningsspol för att få en jämn spänningskälla.

5.1.3 Övrig mjukvara i pallen

Utöver den del av mjukvaran som beskrivits i tidigare avsnitt såsom styrning av multiplexer, avläsning av mätvärden från lastcellerna och beräkning av total last och tyngdpunkt finns det även en del som hanterar den trådlösa kommunikationen och styrning av energisparkrets.

5.1.3.1 Trådlös kommunikation

Vid programmering av NRF24L01-modulen användes ett redan befintligt bibliotek, RF24 [15], där de funktioner som krävdes för att kunna överföra datan från lastcellerna fanns färdiga.

I mjukvaran väljs den adress där överföring av data ska ske och den överföringshastighet som datan ska skickas med. Den trådlösa kommunikationen startas därefter och adressen för överföring ställs in.

5.1.3.2 Avstängning och energi

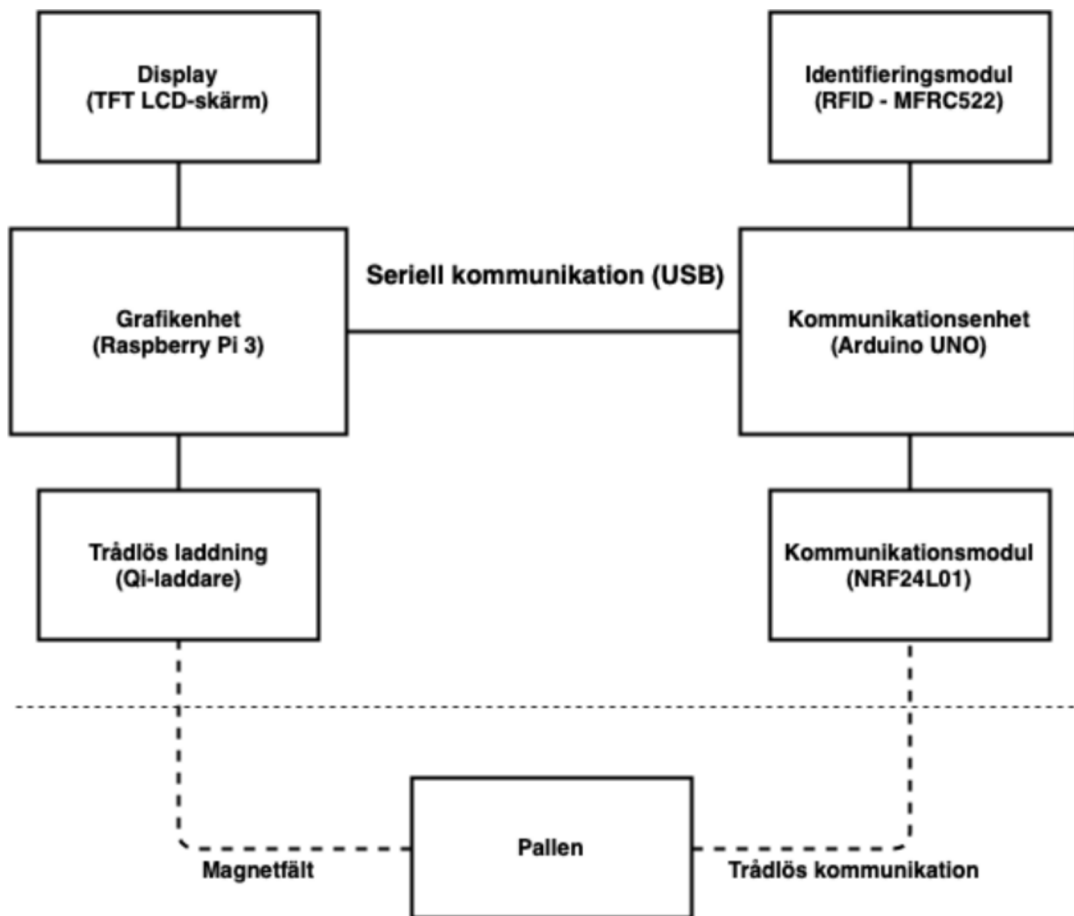
I programkoden stängs styrenheten automatiskt av om de två prototyperna inte har kontakt via den trådlösa laddningen och en viss tid har förflutit. Enheten stängs även av då batteriets spänning är för låg. Batteriets spänningsnivå mäts dels för att, genom energisparkretsen, stänga av enheten men även för skicka det avlästa värdet vidare till trucken där batterinivån presenteras på en LCD-skärm. När batterierna laddar ut sig i pallen sjunker spänningen på batterierna. Det betyder att skillnaden på utsignalerna minskar och därför behöver det kompenseras för i mjukvaran innan värdena skickas till truckprototypen.

5.2 Trucken

Trucken är den enhet där den överförda datan från pallen presenteras för användaren. Här nedan beskrivs den hårdvara som konstruerades samt den mjukvara som utvecklades för systemet.

5.2.1 Hårdvara

Truckens hårdvara byggdes upp av en Raspberry Pi som är den enhet som hanterar grafiken och en Arduino som är den enhet som hanterar identifiering utav pallens ID och den trådlösa kommunikationen mellan pallen och trucken, som sker via NRF24L01. Kommunikationsenheten överför den mottagna datan från pallen seriellt vidare till grafikenheten, där den presenteras på en LCD-skärm.



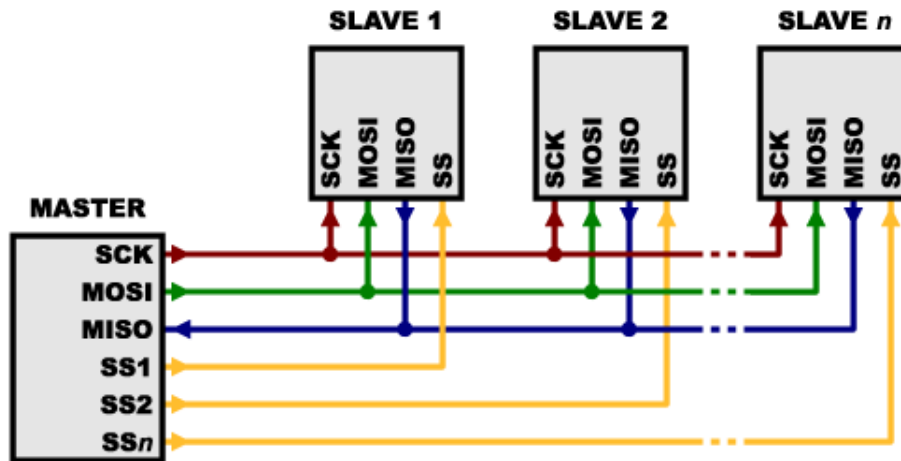
Figur 5.3: Blockschema över prototypens hårdvara

Vid uppbyggnad av hårdvaran för prototypen användes endast färdiga moduler. Under arbetet kopplades dessa samman på kopplingsdäck för att säkerställa deras funktion. Då funktionen var säkerställd monterades modulerna på ett prototypkretskort.

5.2.1.1 Kommunikationsenhet - Arduino

Som kommunikationsenhet användes en Arduino UNO, till denna anslöts en NRF24L01-modul för trådlös överföring av data från pallen och en RFID-modul för identifiering av pallens ID.

Både NRF24L01-modulen och RFID-modulen MFRC522 använder sig av SPI som kommunikationsprotokoll och då Arduino endast har stöd för en SPI-enhet i hårdvaran kopplades dessa moduler samman enligt Figur



Figur 5.4: Kopplingsschema för flera SPI-enheter [14]

Anslutningar mellan Arduino och RFID-modulen MFRC522

RFID-modulen anslöts till Arduinon enligt tabell 5.1

Tabell 5.1: Anslutningar mellan Arduino och RFID-modul

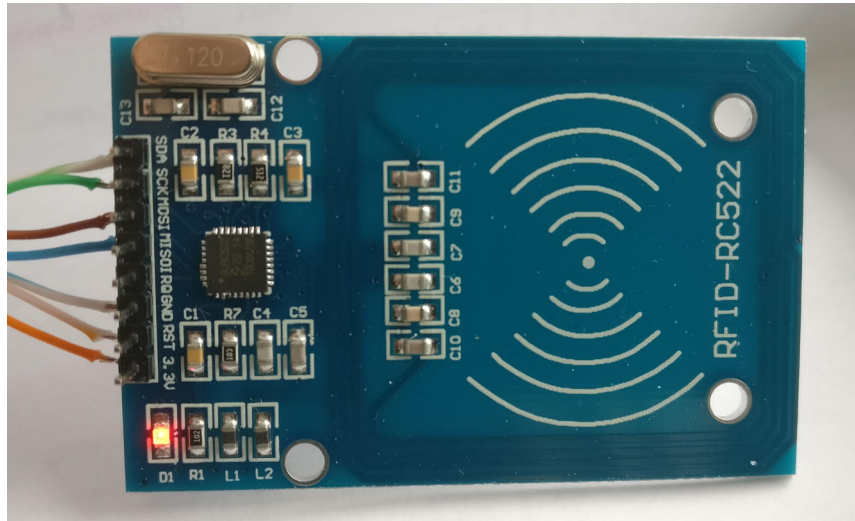
RFID-modul	Arduino
VCC	3.3 V
RST	D9
GND	GND
IRQ	-
MISO	D12
MOSI	D11
SCK	D13
SS	D10

RFID-modulens VCC-ben anslöts till Arduinons 3.3 V då det är den maximala matningsspänning som modulen kan hantera. Modulens logikben klarar dock av 5 V, därav anslöts dessa direkt in till Arduinons digitala utgångar.

Nedan beskrivs de ben som hanterar SPI-kommunikationen mellan Arduinon och RFID-modulen.

- **RST - Reset** Då benets logiknivå sätts låg stängs modulens interna strömkälla av och då logiknivå sätts hög återställs modulen.
- **IRQ- Interrupt** Benet anslöts inte då det bibliotek som användes inte stödjer funktionen.
- **MISO- Master In Slave Out** Det ben där modulen skickar data till Arduinon.
- **MOSI- Master Out Slave In** Det ben där Arduinon skickar data till modulen.

- **SCK - Serial clock** Ben där modulen tar emot klockpulser från Arduinon som möjliggör SPI-kommunikationen.
- **SS - Slave Select** Det ben som aktiverar SPI-kommunikationen.



Figur 5.5: Bild på hur kretskortet såg ut i slutändan

Anslutningar mellan Arduino och NRF24L01-modulen

NRF24L01-modulen anslöts till Arduinon enligt tabell 5.2

Tabell 5.2: Anslutningar mellan Arduino och NRF-modul

NRF-modul	Arduino
VCC	3.3 V
GND	GND
CS	D8
CE	D7
MOSI	D11
SCK	D13
IRQ	-
MISO	D12

NRF-modulens VCC-ben anslöts till Arduinons 3.3 V då det är den matningsspänning som modulen klarar av.

Nedan beskrivs de ben som hanterar SPI-kommunikationen mellan Arduino och NRF24L01-modulen.

- **IRQ - Interrupt** Anslöts inte då det bibliotek som används inte stödjer funktionen.
- **MISO - Master In Slave Out** Det ben där modulen skickar data till Arduinon.

- **MOSI - Master Out Slave In** Det ben där Arduinon skickar data till modulen.
- **SCK - Serial clock** Ben där modulen tar emot klockpulser från Arduinon som möjliggör SPI-kommunikationen.
- **CS - Chip Select** SPI-kommunikationen är aktiverad så länge benets logi-knivå hålls hög.
- **CE - Chip Enable** Det ben som aktiverar SPI-kommunikationen.

5.2.1.2 Grafikenhet - Raspberry Pi

Till grafikenheten anslöts en LCD-skärm där den data som skickas till trucken presenteras. Det anslöts även en Qi-laddare vars signal användes för att, genom pallens energisparkrets, starta och stänga av pallens styrenhet.

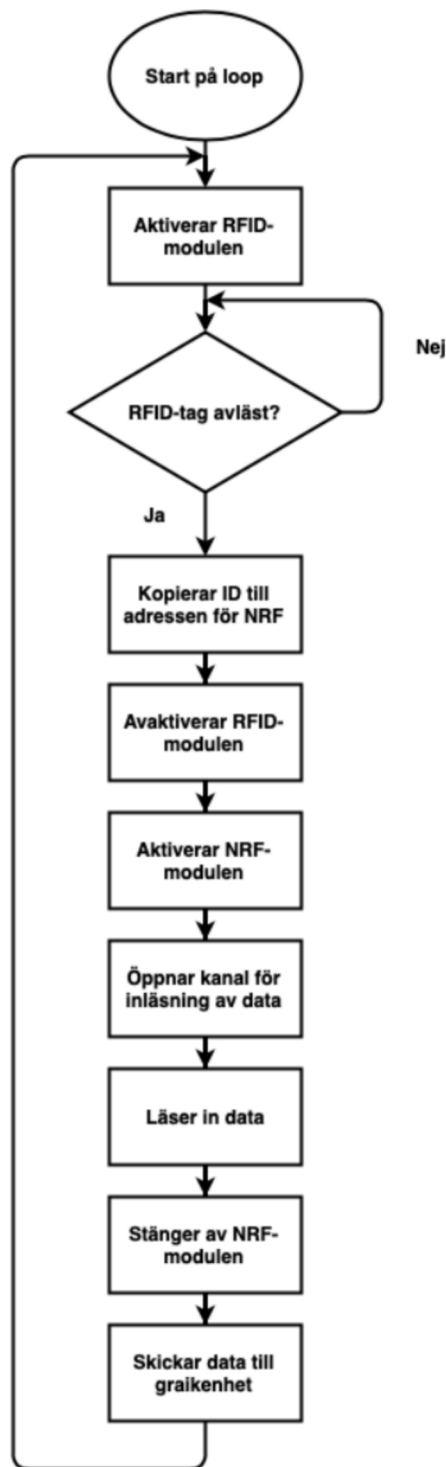
5.2.2 Mjukvara i trucken

Truckens mjukvara består av ett program för kommunikationsenheten som programmerades i Arduinos IDE och ett program för grafikenheten som programmerades i Python.

5.2.2.1 Avläsning av pallens ID och kommunikation

Programmet för kommunikationsenheten hanterar avläsning av pallens ID och den data som överförs trådlöst från pallen för att sedan skickas vidare seriellt till grafikenheten. För användning av NRF24L01-modulen och RFID-modulen MFRC522 användes redan befintliga biblioteket där de funktioner som krävdes för programmering av modulerna fanns färdiga.

Då kommunikationsenheten hanterar flera SPI-enheter ställs deras respektive SPI-ingång in på varsin digital utgång i mjukvaran. Det går bara att använda en SPI-enhet åt gången och genom att aktivera den digitala utgången för den specifika enheten kan den användas. I figur 5.4 nedan ses flödesschema för den del av programmet som hanterar de båda SPI-enheterna.



Figur 5.6: Flödesschema som beskriver anslutning till pallan och överföring av data

I programkoden aktiveras kommunikationen med RFID-modulen genom att modulens SPI-signal aktiveras, därefter läses RFID-taggen av och dess ID kopieras till adressen för NRF-modulen. Slutligen startas den trådlösa kommunikationen och data läses in för att sedan skickas till graikenheten.

Datan som skickas seriellt mellan kommunikationsenheten och grafikenheten skickas som en sträng och för att kunna avkoda och hantera datan användes skiljetecken mellan variablerna.

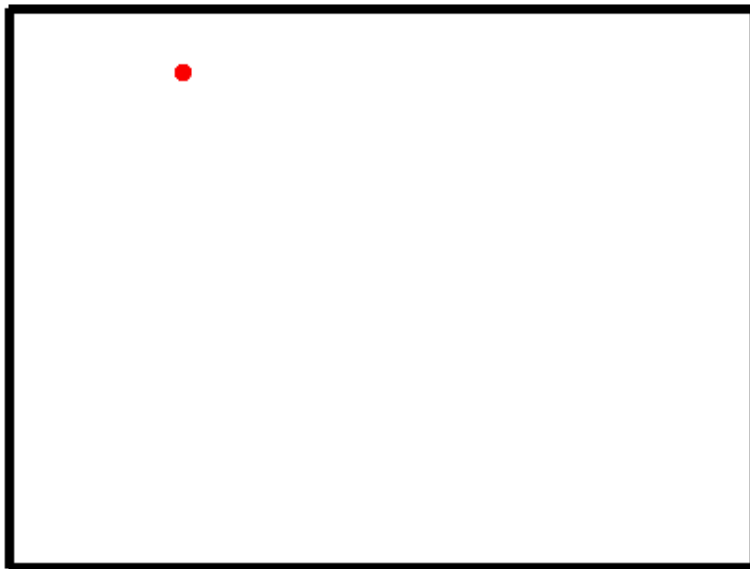
5.2.2.2 Grafik och avkodning av data

Eftersom grafikenheten programmerades i Python valdes programbiblioteket Pygame för utveckling av den del av programmet som hanterar grafiken. Enheten hanterar även mottagning och avkodning av den data som överförs seriellt från kommunikationsenheten.

Den mottagna datan skickas som en sträng och avkodas för att kunna hantera de enskilda variablerna som sedan presenteras på skärmen. Figur 5.7 nedan visar den information som presenteras, där x- och y-koordinater är angivna i millimeter och massan i kilogram. Den svarta rektangeln representerar pallen och den röda pricken dess tyngdpunkt.

Connected to pallet: 95B81A3C
Battery level: 96%
Contents: Mätinstrument

Allowed to lift



x = 177
y = 58
mass = 228

Figur 5.7: En skärmdump från trucken - visar hur grafiken kan se ut

6

Resultat

Under projektets gång var det tänkt att konstruera ett helhetssystem som skulle demonstrera den nya funktionen, att mäta massa och tyngdpunkt i en smart lastpall. I slutet av arbetet uppstod dock komplikationer med den förstärkarkrets som hade tagits fram för att läsa av givarnas mätvärden. Då kretsen hade monterats på det slutgiltiga prototypkretskortet, slutade kretsen att fungera och efter felsökning och byte av komponenter utan framgång fanns det inte tillräckligt med tid för att kunna åtgärda felet. Därför presenteras här resultatet som flera delsystem. Delsystemens funktioner säkerställdes var för sig under arbetets gång, men ett helhetssystem färdigställdes inte.

6.1 Elektroniken

Givarna och elektroniken som läser av dem är det som gör den nya funktionen till smarta lastpallar möjlig. Följande funktioner testades

- De givare som användes var lastceller, de gav en nästintill linjär utsignal vid belastning, även efter montering i pallprototypen.
- Utsignalerna från givarna lästes av en och en med hjälp av en multiplexer och på så vis behövdes endast en förstärkarkrets.
- Den förstärkta signalen kunde avläsas av en A/D-omvandlare och innehöll tillräckligt lite brus och störningar för att effektivt kunna läsa av värdena.
- Ett enkelt sätt att räkna ut tyngdpunkten togs fram men den testades inte efter att ha monterat givarna i pallprototypen.

6.2 Demonstrationsprototyperna

Prototyperna blev inte helt klara men delsystemen testades och fungerade på ett förväntat sätt innan sammanfogning.

6.2.1 Pallen

- Energisparkretsen i pallprototypen fungerade på ett tillfredsställande sätt. När pallen inte använts på en förutbestämd tid stängs den av och drar ingen ström.
- Pallen väcks ur sitt energisparläge genom att trucken, när den kommer tillräckligt nära, inducerar en spänning i pallens laddkrets.
- Mikrokontrollern i pallen kan skicka en datasträng innehållandes information om pallen trådlöst till truckprototypen

- Pallen övervakar batterinivån och stänger av sig om den är för låg.

En funktion som försökte realiseras var att pallen skulle laddas trådlöst. Den trådlösa laddningen skulle användas både för att ladda pallens batterier och för att, genom signalen från laddningen, starta pallens styrenhet. Ett alkaliskt icke-uppladdningsbart batteri används och därmed används laddkretsen endast till att starta pallen.

- Genom lastceller inbyggda i pallens klossar mäts massa och tyngdpunkten beräknas.

Pallens styrenhet startas genom att det skickas en signal från laddkretsen till energisparkretsen, därefter ansluter trucken till pallen genom att med RFID-modulen läsa av pallens ID. Då ID:t är avläst startas den trådlösa kommunikationen och data börjar skickas mellan pallen och trucken. Pallen håller sig sedan igång så länge den laddas av trucken.

Genom lastceller inbyggda i pallens fyra kontaktpunkter mäts lastens massa och tyngdpunkten i 2 dimensioner beräknas för att, genom trådlös kommunikation, skickas till truckprototypens kommunikationsenhet. Den mottagna datan överförs sedan seriellt vidare till prototypens grafikenhet, där datan avkodas för att presenteras på en LCD-skärm.

Pallen är försedd med en RFID-tag för identifiering av dess ID.

Laddkrets som används för att starta pallens enhet då den är i energisparläge.

NRF24L01-modul för överföring av data till trucken.

6.2.2 Trucken

Protoypens huvudfunktion var att presentera informationen som erhöles från de lastceller som byggdes in i pallen. Den information som efter undersökning valdes för presentation och som visas på LCD-skärmen är nedanstående:

- Tyngdpunktens x- och y-koordinater i millimeter
- Lastens massa i kilogram
- Pallens ID
- Pallens batterinivå
- Vad det är för gods på pallen
- Om det är tillåtet att lyfta pallen
- Om trucken fortfarande är ansluten till pallen

Trucken skulle även kunna identifiera och ansluta till den pall den var på väg att lyfta. Denna funktion testades och verifierades under arbetets gång. Identifieringen sker med den RFID-modul som var ansluten till prototypens kommunikationsenhet. Genom att pallens ID läses av kan trucken ansluta till den specifika pallen.

7

Slutsats

Syftet med projektet var att undersöka möjligheterna att implementera givare i befintliga smarta lastpallar för mätning av last och beräkning av tyngdpunkt. Stor vikt lades i att hitta den typ av givare som bäst lämpade sig för ändamålet och hur mätvärden från dessa skulle tas fram och hanteras. Förmodligen skulle det gå att implementera den sortens givare och den elektronik som tagits fram i en smart lastpall.

I rapportens inledande del preciseras de frågeställningar som projektet skulle undersöka.

- Vilken typ av givare bör användas för att kunna mäta lasten och tyngdpunkten på pallen?
- Hur ska elektroniken som styr och läser av givarna fungera?
- Vilken information ska skickas mellan pall- och truckprototyperna?
- Vilken kommunikationsmetod ska användas för att kunna skicka informationen?
- Hur ska strömförsörjningen fungera?
- Vilken information ska presenteras för truckföraren och hur?
- Skulle det kunna vara kostnadsmässigt effektivt att implementera funktionerna i framtida smarta lastpallar?
- Hur skulle en smart lastpall med denna typen av elektronik på ett bra sätt skyddas mot variationer i sin omgivning som exempelvis fukt, temperatur och stötar?

Nästan alla frågeställningar har besvarats i rapporten fränsett om det är kostnadsmässigt effektivt och hur man kan göra pallar med denna utrustningen tålig mot yttre faktorer.

7.1 Utmaningar under arbetets gång

Initialt i arbetet var det tänkt att beräkna pallens tyngdpunkt i tre dimensioner. På grund av projektets tidsramar togs beslut om att fokusera på andra delar i projektet för att uppnå tillfredsställande resultat. Därav utförs endast tyngdpunktberäkningar i två dimensioner.

Den trådlösa laddkrets som användes skulle till en början nyttjas både för att ladda pallens batterier och för att starta och stänga av pallen genom den energisparkrets som konstruerades. Det var först planerat att bygga ett batteripack av uppladdningsbara Litiumjonbatterier. De komponenter som beställdes för batteripacket var

dimensionerade efter en lägre spänning än vad som, efter en del modifieringar, i slutändan krävdes. Därför användes laddkretsen enbart till att hantera enhetens energisparläge.

De ändringar som ledde till att batteripacket inte kunde användas som tänkt var att boost-konvertern som skulle reglera spänningen till en högre spänning gav för mycket störningar. Efter försök att filtrera störningarna utan framgång beslöts det att använda ett alkaliskt batteri som direkt spänningskälla.

I början av arbetet med truckprototypen var det tänkt att endast använda Raspberry Pi som centralenhet i trucken. Efter många försök att få NRF24L01-modulen att kommunicera med Raspberry Pi så togs beslut om att dela upp truckprototypens elektronik i två delar. En Arduino användes som kommunikationsenhet och Raspberry Pi som grafikenhet. Valet grundade sig i kunskap om att Arduinon fick kontakt med NRF24L01-modulen, då det hade använts i pallen. Mellan Arduinon och Raspberry Pi sker då dataöverföringen via USB-kommunikation.

7.2 Fortsatt utveckling

Exempel på delar som skulle kunna förbättras, arbetas vidare på eller undersökas ytterligare i de delsystem som har konstruerats

- Kretskortet i pallprototypen hade blivit snyggare och förmodligen krånglat mindre om ett mönsterkort hade beställts istället för att löda allt på ett prototypkretskort.
- Det hade gått att ställa in givarparen så de inte fick någon startskillnad på utsignalerna. Det kan göras genom att lägga till eller minska resistansen mellan givarna med trimpotentiometrar.
- För att givarna inte ska behöva hålla uppe hela lastens tyngd skulle det gå att fästa trådtöjningsgivare direkt på klossarna i pallen.
- Pallprototypen hade en ganska lång uppstartstid ur sitt viloläge, ungefär 2 sekunder. Det hade gått att minska den genom att använda en annan mikrokontroller som startar snabbare och att förbättra initieringsdelen av programmet.
- Det hade förmodligen gått att öka pallens kapacitet genom att använda något elastiskt material mellan kloss och brädor. En gummiduk testades där och då klarade givarna en mycket högre belastning, förmodligen för att kraften fördelats bättre över klossen.

Litteraturförteckning

- [1] LogTrade - The smart pallet.
<https://www.logtrade.se/en/the-smart-pallet/> (Hämtad Apr. 18, 2019)
- [2] Pallettech - Cloud-connected pallets.
<https://www.pallettech.co> (Hämtad Apr. 18, 2019)
- [3] SMART Logistics - ICP pallets.
<https://smartlogisticsinc.com/smart-pallets/#chain> (Hämtad Jun. 10, 2019)
- [4] Sparkfun Force Sensing Resistor 0.5".
<https://www.sparkfun.com/products/9375> (Hämtad Maj. 28, 2019)
- [5] Elprocus - Force Sensing Resistor.
<https://www.elprocus.com/force-sensing-resistor-technology/> (Hämtad Mar. 23, 2019).
- [6] Omega - Load Cells & Force Sensors.
<https://www.omega.com/en-us/resources/load-cells> (Hämtad Maj. 29, 2019)
- [7] ScienceDirect - Learn more about Load Cell
<https://www.sciencedirect.com/topics/engineering/load-cell> (Hämtad Maj. 29, 2019)
- [8] Sparkfun - Load Sensor - 50kg.
<https://www.sparkfun.com/products/10245> (Hämtad Maj. 29, 2019)
- [9] Sparkfun - Serial Peripheral Interface (SPI).
<https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all> (Hämtad Mar. 20, 2019).
- [10] Lastminuteengineers - nRF24L01+.
<https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/> (Hämtad Mar. 29, 2019)
- [11] V. Daniel Hunt, Albert Puglia och Mike Puglia, "An Overview of RFID Technology", RFID: A Guide to Radio Frequency Identification, editon. Hoboken, New Jersey, Land: USA
- [12] Arduino playground - MFRC522.
<https://playground.arduino.cc/Learning/MFRC522/> (Hämtad Jun. 10, 2019)
- [13] Electronics Source - Arduino nano datasheet.
<https://www.es.co.th/Schemetic/PDF/ARMB-0022.PDF> (Hämtad Maj. 29, 2019)
- [14] Sparkfun - Serial Peripheral Interface (SPI).
<https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all> (Hämtad Apr. 23, 2019)
- [15] TMRH20 - NRF24L01+ Driver Class Documenation.
<https://tmrh20.github.io/RF24/> (Hämtad Jun. 04, 2019)

- [16] Miguelbalboa - RFID library for MFRC522.
<https://github.com/miguelbalboa/rfid>. (Hämtad Jun. 04, 2019)

Bilaga A - Programkod för styrenhet

```
1 /*Lastcellernas placering
2  * origo
3  *      2    3
4  *      1    0*/
5 #include <SPI.h>
6 #include <nRF24L01.h>
7 #include <RF24.h>
8
9 #define ch_A_pin 5 //Ben för att byta kanal på multiplexer
10 #define ch_B_pin 6 //Ben för att byta kanal på multiplexer
11 #define power_pin 4 //Ben som, när det är satt högt, håller igång kretsen
12 #define charging_pin 3 //Ben som läser av ifall kretsen får "laddning" från trucken
13 #define bat_pin A4 //Ben för mätning av batterispänning
14 #define loadcell_pin A7 //Ben för mätning av lastcellerna
15
16 #define x_len 800 //Längderna på pallen i millimeter
17 #define y_len 600
18
19 #define max_bat_volt 973 //Motsvarar ungefär 9.5 V efter spänningsdelaren
20 #define min_bat_volt 768 //Motsvarar ungefär 7.5 V
21
22 #define dc_time 5000 //Tid innan pallen ska stänga av sig när "laddning" försvinner
23
24 RF24 radio(7, 8); //Initiering av benen till NRF-modulen
25 const byte address[5] = {0x90, 0x8C, 0x7B, 0xA6, 0x00}; //Pallens ID
26
27 uint32_t since_disconnect = 0; //Variabel för hur länge pallen varit utan "laddning"
28 int sensor_values[4]={0,0,0,0}; //Värden från A/D-omvandlaren och lastcellerna
29 int battery_level=1000; //Analoga värdet från mätning av batterispänning
30 const int calib_offset[4]={200,200,200,200}; //Faktor för att ta bort offset på lastcellerna
31 //Ej kalibrerat än
32 float sensor_mass[4]={0,0,0,0}; //Konverterade värden till kg
33 const float calib_factor[4]={15,15,15,15}; //Faktor för att konvertera spänningskillnaderna till kg
34 //Ej kalibrerat än
35
36 struct{ //En struct med all data som ska skickas till trucken NRF-modulerna
37     int x=0; //Koordinater för tyngdpunkten
38     int y=0;
39     int mass=0; //Sammanlagda massan på pallen
40     byte battery=100; //Pallens batterinivå
41     bool stopConnection= false; //Ifall pallen föreslår att trucken ska sluta kommunicera
42     char contents[16]= "Dyra instrument"; //Vad som finns på pallen
43 }data;
```

```

45 void setup(){
46   pinMode(ch_A_pin, OUTPUT); //Initerar alla in- och utgångar
47   pinMode(ch_B_pin, OUTPUT);
48   pinMode(power_pin, OUTPUT);
49   pinMode(bat_pin, INPUT);
50   pinMode(charging_pin, INPUT);
51
52   digitalWrite(power_pin, HIGH); //Skickar en spänning till transistorn som tillåter att pallen hålls igång
53
54   radio.begin(); //Initering av NRF-modulen
55   radio.openWritingPipe(adress); //Ställer in modulen för att kunna skicka från den förinställda adressen
56   radio.setPALevel(RF24_PA_MIN); //Inställning av signalstyrka
57   radio.stopListening(); //Ifall modulen skulle vara i lyssningsläge stängs det av
58 }
59
60 void loop(){
61   update_bat_level(); //Kollar batterinivån och uppdaterar procentsatsen
62
63   if(data.battery == 0) digitalWrite(power_pin, LOW); //Stänger av pallen om batterinivån är för låg
64
65   if(!digitalRead(charging_pin)){ //Kontrollerar laddningen från trucken
66     if(!stay_connected()){ //Kollar om pallen fortfarande ska vara ansluten till trucken eller om den ska stänga av sig
67       data.stopConnection=true; //Ska den stängas av skickar den meddelanden som säger det till trucken
68       for(int i=0;i<10;i++){
69         radio.write( &data, sizeof(data)); //Skickar meddelande om avstängning 10 ggr med 10 ms delay emellan
70         delay(10);
71       }
72       digitalWrite(power_pin, LOW); //Stänger av sig
73     }
74   }
75
76   updateSensorValues(); //Uppdaterar värdena från varje givarpar
77   calculateCom(); //Beräknar tyngdpunkten på pallen
78
79   radio.write( &data, sizeof(data)); //Skickar structen med data
80   delay(50); //Delay för att inte skicka för ofta till trucken
81 }

```

```

83 void calculateCom(){
84 //Givarpar 1 är i origo. Multiplicerar belastning med avstånd från origo och
85 //dividerar med totala belastningen
86 data.x=int((sensor_mass[0]*x_len+sensor_mass[2]*x_len)/data.mass);
87 data.y=int((sensor_mass[2]*y_len+sensor_mass[3]*y_len)/data.mass);
88 }
89
90 void updateSensorValues(){ //Loopar genom alla givarpar och uppdaterar värdena från dem,
91 //summerar korrigerade värden till total massa
92 data.mass=0;
93 for(byte i=0;i<4;i++){
94   channel(i);
95   sensor_values[i]=analogRead(loadcell_pin);
96   sensor_values[i]-=calib_offset[i];
97   sensor_mass[i]=sensor_values[i]*calib_factor[i];
98   data.mass+=sensor_mass[i];
99 }
100 }
101
102 void channel(byte i){ //byter kanal på multiplexern
103   digitalWrite(ch_A_pin, i & 0b01);
104   digitalWrite(ch_B_pin, i & 0b10);
105 }
106
107 bool stay_connected(){ //Returnerar sant eller falsk beroende på hur länge pallet inte mottagit "laddning" från trucken
108   if(since_disconnect == 0)since_disconnect = millis();
109   if((millis() - since_disconnect) > dc_time)return false;
110   else return true;
111 }
112
113 void update_bat_level(){ //Uppdaterar batterinivå
114   battery_level=analogRead(bat_pin);
115   if(battery_level >= max_bat_volt)data.battery=100; //Om spänningen skulle vara över maxspänningen sätts nivån till 100%
116   else if( battery_level <= min_bat_volt)data.battery=0; //Skulle den vara under lägsta gränsen sätts den till 0%
117   else data.battery=byte(((battery_level-min_bat_volt)/float(max_bat_volt-min_bat_volt))*100.0);
118   //Procent av det tillåtna området batteriet får ligga inom
119 }

```

Bilaga B - Programkod för grafikenhet

```
1  #!/usr/bin/env python
2  import pygame
3  from pygame.locals import *
4  from picamera import PiCamera
5  import time
6  import serial
7  import binascii
8
9  WHITE = (255, 255, 255)
10 BLACK = (0, 0, 0)
11 RED = (255, 0, 0)
12 GREY = (128, 128, 128)
13 screen_width = 640
14 screen_height = 480
15 pallet_width = 200
16 pallet_height = 100
17
18 #Serial RPi-Arduino
19 port = "/dev/ttyUSB0"
20 ser = serial.Serial(port, 9600, timeout = 1)
21 ser.flushInput()
22
23 def print_text(string, font, x, y):
24     #Skapa textyta
25     text = font.render(string, True, BLACK)
26     #Skapa rektangel för textytan
27     textrect = text.get_rect()
28     textrect.centerx = x
29     textrect.centery = y
30     screen.blit(text, textrect)
31
32 def data_between(string, ch1, ch2):
33     newString = ""
34     passedFirst = False
35
36     for i in range(0, len(string)):
37         if passedFirst:
38             if string[i] == ch2:
39                 if len(string) > 0:
40                     return newString
```

```

41         else:
42             return "0"
43         newString += string[i]
44         if string[i] == ch1:
45             passedFirst = True
46         return "0"
47
48 # Initalizing
49 pygame.init()
50 screen = pygame.display.set_mode((screen_width, screen_height))#, pygame.FULLSCREEN)
51 pygame.display.set_caption("Forklift")
52 camera = PiCamera()
53
54 #IMAGES
55 com_image = pygame.image.load('centerOfMass.png').convert_alpha()
56 com_image = pygame.transform.scale(com_image, (10, 10))
57
58 #FONTS
59 smallfont = pygame.font.Font(None, 24)
60 smfont = pygame.font.Font(None, 30)
61 mediumfont = pygame.font.Font(None, 48)
62 largefont = pygame.font.Font(None, 58)
63
64
65
66 #START OF THE MAIN LOOP
67 while True:
68     #PREPARE GRAPHICS
69     screen.fill(WHITE)
70
71     #UPDATE VALUES FROM SERIAL COM
72     ser.flushInput()
73     data = ser.readline().decode().strip()
74     print(data)
75     #DECODE VALUES FROM SERIAL COM
76     x_com = data_between(data, '.', ',')
77     y_com = data_between(data, ',', '!')
78     mass = data_between(data, '!', '?')
79     batteryLevel = data_between(data, '?', '/')
80     stopConnection = data_between(data, '/', '*')

```

```

81     cameraStatus = data_between(data, '*', ';')
82     allowedToLift = data_between(data, ';', '#')
83     contents = data_between(data, '#', ':')
84     palletID = data_between(data, ':', '%')
85
86
87     if cameraStatus == '0':
88         if palletID != '0':
89             if stopConnection == '0':
90                 camera.stop_preview()
91                 #BLIT IMAGES
92                 pygame.draw.rect(screen, BLACK,[10, 125, 400, 300], 5)
93                 screen.blit(com_image, (int(x_com)/2, int(y_com)/2))
94
95                 #PRINT VALUES FROM SERIAL COM
96                 print_text("x = " + x_com, mediumfont, 550, 200)
97                 print_text("y = " + y_com, mediumfont, 550, 250)
98                 print_text("mass = " + mass, mediumfont, 518, 300)
99                 print_text("Contents: " + contents, smallfont, 53, 60)
100                 print_text("Battery level: " + batteryLevel + "%", smallfont, 76, 40)
101                 print_text("Connected to pallet: " + palletID, smallfont, 130,
102                             20)#palletID, smallfont, 550, 400)
103
104                 #PRINT IF ALLOWED TO LIFT OR NOT
105                 if allowedToLift:
106                     print_text("Allowed to lift", largefont, 460, 50)
107                 else:
108                     print_text("Not allowed to lift", largefont, 450, 50)
109                     else: #PRINT SCREEN WHEN NOT CONNECTED
110                         screen.fill(GREY)
111                         print_text("Can not connect", largefont, screen_width/2,
112                                 screen_height/2)
113
114             else:
115                 #PRINT STARTSCREEN
116                 screen.fill(GREY)
117                 print_text("WELCOME", largefont, screen_width/2, screen_height/2)
118                 print_text("You are not connected", mediumfont, screen_width/2,
119                             (screen_height/2)+50)
120
121     pygame.display.flip()

```

```

111         screen_height/2)
112     else:
113         #PRINT STARTSCREEN
114         screen.fill(GREY)
115         print_text("WELCOME", largefont, screen_width/2, screen_height/2)
116         print_text("You are not connected", mediumfont, screen_width/2,
117                    (screen_height/2)+50)
118
119     pygame.display.flip()
120     time.sleep(1/100)
121     pygame.display.update()
122     if cameraStatus == '1':
123         camera.start_preview(fullscreen = False, window = (100, 100, 640, 480))
124         time.sleep(10)
125         if cameraStatus == '0':
126             camera.stop_preview()
127     #END OF THE MAIN LOOP
128
129 pygame.quit()
130
```

Bilaga C - Programkod för kommunikationsenhet

```
1 #include <SPI.h>
2 #include <nRF24L01.h>
3 #include <RF24.h>
4 #include <require_cpp11.h>
5 #include <MFRC522.h>
6 #include <deprecated.h>
7 #include <MFRC522Extended.h>
8 |
9 const int SWITCH = 6; // Ben för switch
10 const int CE = 7; // Ben för Chip Enable NRF
11 const int CS1 = 8; // Ben för Chip Select NRF
12 const int RST = 9; // Ben för Reset RFID
13 const int CS2 = 10; // Ben för Chip Select RFID
14
15
16 byte adressNRF[5] = {0x99, 0x99, 0x99, 0xA6, 0x00}; //Initierar adress NRF
17
18 RF24 radio(CE, CS1); // Chip Enable, Chip Select NRF
19 MFRC522 mfrc522(CS2, RST); // Chip select, Reset RFID
20
21 struct Data{ // Struct med datan som skickas från pallen
22     int x = 0; // Tyngdpunktens x-koordinat
23     int y = 0; // Tyngdpunktens y-koordinat
24     int mass = 0; // Totala massan
25     int batteryLevel = 0; // Pallens batterivå
26     int cameraStatus = 0; // Kamera på/av
27     bool allowedToLift = false; // Om det är tillåtet att lyfta pallen
28     bool stopConnection = false; // Om trucken ska sluta kommunicera med pallen
29     char contents[16] = "-"; // Vad som finns på pallen
30 }data;
31
32 void setup(){
33
34     pinMode(CS1, OUTPUT); // Ben för Chip Select NRF sätts till output
35     pinMode(CS2, OUTPUT); // Ben för Chip Select RFID sätts till output
36     pinMode(SWITCH, INPUT); // Ben för switch sätts till input
37     Serial.begin(9600);
38     SPI.begin(); // Initiering av SPI
39     mfrc522.PCD_Init(); // Initiering av RFID
40     radio.begin(); // Initiering av NRF-modulen
```

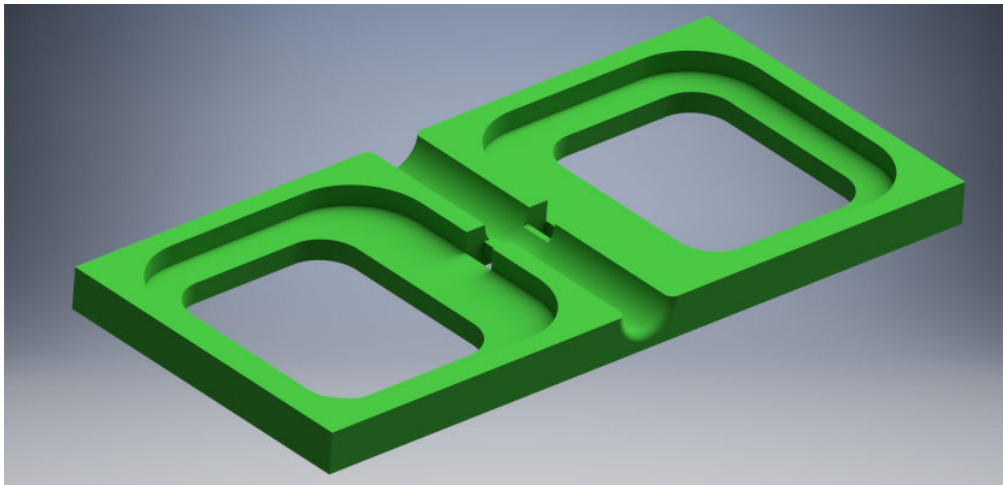
```

41   radio.setPALevel(RF24_PA_MIN); // Ställer in Power Amplifier för NRF
42 }
43 void loop() {
44
45   delay(100);
46
47   digitalWrite(CS2, LOW); // Sätter på RFID
48
49   if(mfrc522.PICC_IsNewCardPresent()){ // Kollar om RFID-kort är i närheten
50     if(mfrc522.PICC_ReadCardSerial()) { // Kollar om RFID-kort avläst
51       for (byte i = 0; i < mfrc522.uid.size; ++i){
52         adressNRF[i] = mfrc522.uid.uidByte[i];
53       } // Kopierar innehållet på RFID-kort till adressen till NRF:en
54
55       adressNRF[4]=0x00;
56
57       digitalWrite(CS2, HIGH); // Stänger av RFID
58       digitalWrite(CS1, LOW); // Sätter på NRF
59
60       radio.stopListening(); // NRF, slutar lyssna efter inkommande meddelanden
61       radio.openReadingPipe(0, adressNRF); // NRF, öppnar kanal för läsning
62       radio.startListening(); // NRF, börjar lyssna
63
64       while(!radio.available()){ }
65       radio.read(&data, sizeof(data)); // Läser in data från pallen
66       digitalWrite(CS1, HIGH); // Stänger av NRF
67
68       // SWITCH
69       data.cameraStatus = digitalRead(SWITCH); // Läser av ben för switch
70       delay(1);
71
72       send_data(); // Skickar den mottagna datan från pallen seriellt till trucken
73     }
74   }
75 }
76
77 void send_data(){ // Funktion för att seriellt överföra datan från
78   pallen vidare till trucken
79   Serial.print('.');//Start
80   Serial.print(data.x); // Tyngdpunktens koordinater

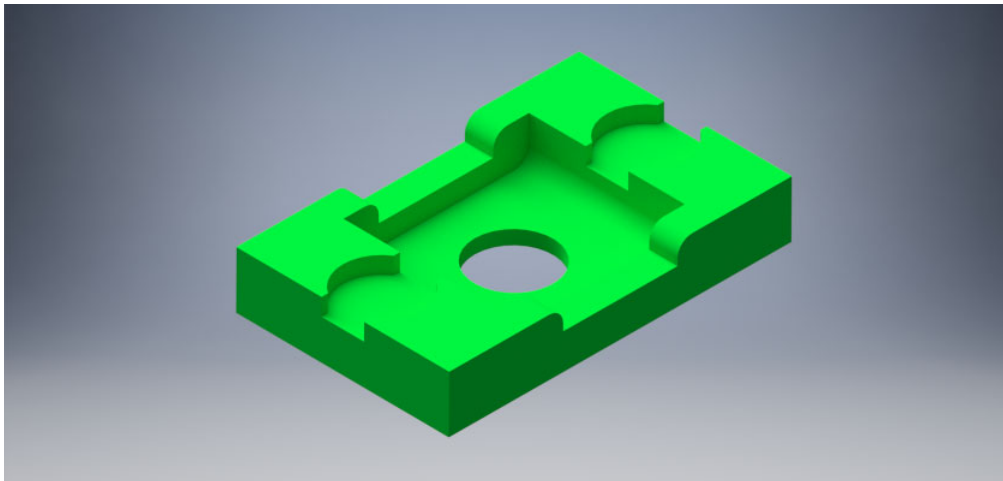
```

```
80 Serial.print(',');
81 Serial.print(data.y);
82 Serial.print('!');
83 Serial.print(data.mass);           // Totala massan
84 Serial.print('?');
85 Serial.print(data.batteryLevel);   // Pallens batterinivå
86 Serial.print('/');
87 Serial.print(data.stopConnection); // Om trucken ska sluta kommunicera med pallen
88 Serial.print('*');
89 Serial.print(data.cameraStatus);   // Kamera på/av
90 Serial.print(';');
91 Serial.print(data.allowedToLift);   // Om det är tillåtet att lyfta pallen
92 Serial.print('#');
93 for(int i = 0; i < 16; i++){
94     if(data.contents[i] >= 65 && data.contents[i] <= 122)
95         Serial.print(data.contents[i]); // Vad som är på pallen
96 }
97 Serial.print(':');
98 for(byte i = 0; i < mfrc522.uid.size; i++){
99     Serial.print(mfrc522.uid.uidByte[i], HEX); // Pallens ID
100 }
101 Serial.print('%');
102
103 Serial.println();
104 }
105
```

Bilaga D, CAD höljen för givare

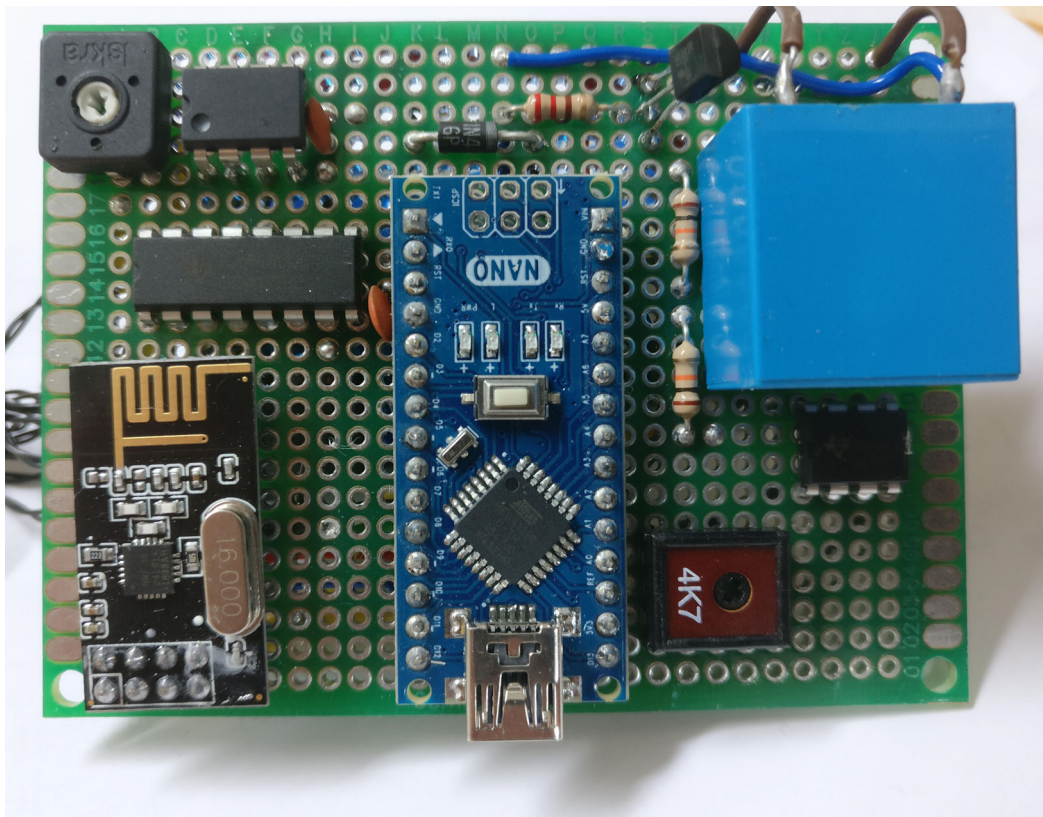


Figur .1: Hölje för givare, undre del



Figur .2: Hölje för givare, övre del

Bilaga E, Bilder på prototyperna



Figur .3: Bild på hur kretskortet i pallprototypen



Figur .4: Bild på pallprototypen



Figur .5: Bild på truckprototypen