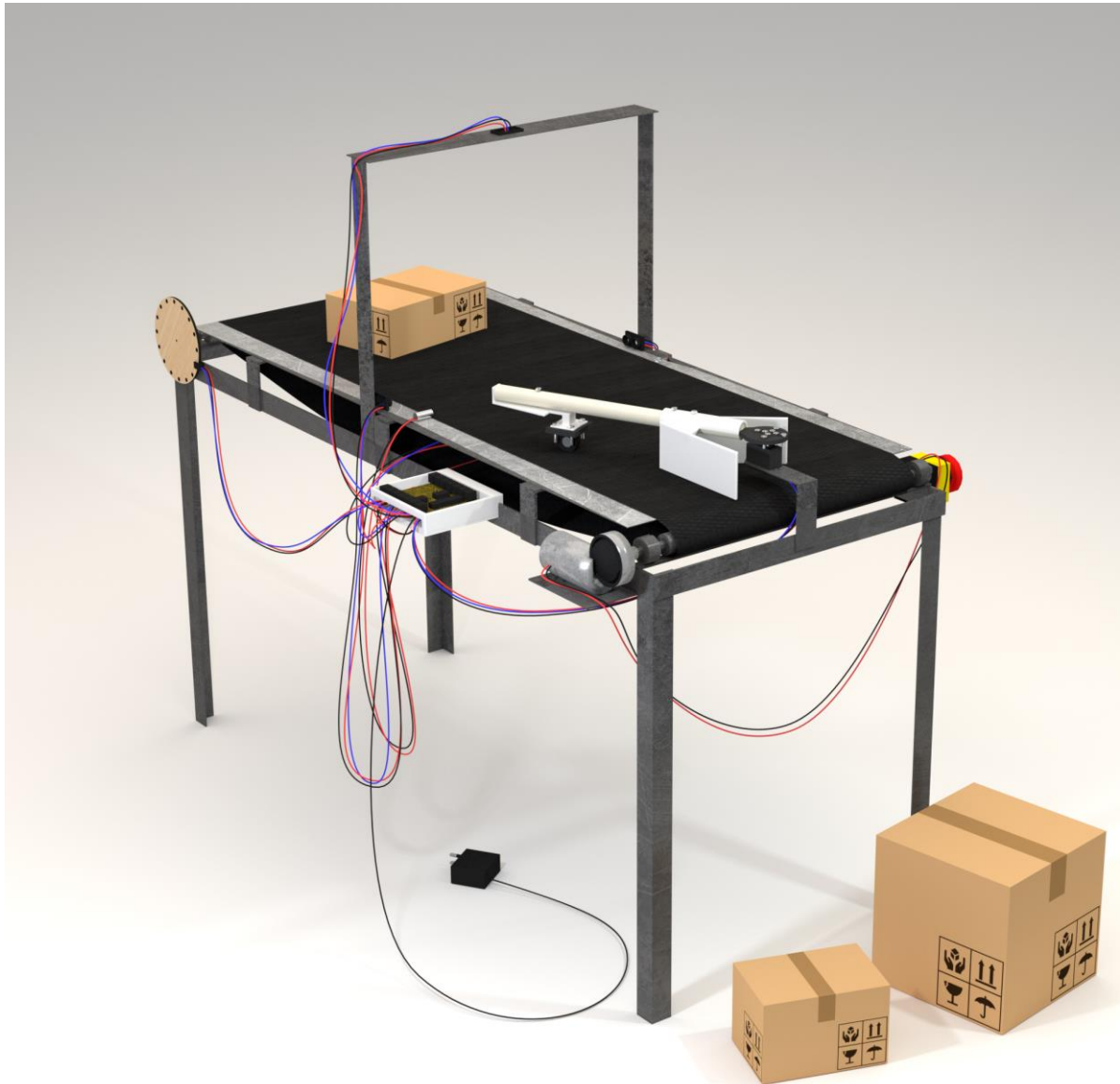

Teknik A Eksamen: Sorter med manér

Pakkesortering med transportbånd og servo-arm



Produktudvikling

Krav

Rullebåndet skal:

- Transportere pakker af op til 20 kg
- Sortere pakker i to bunker
- Være sikkert og let at bruge
- Have et nødstop, der afbryder forbindelsen til strømmen
- Gøre det muligt at se statistik for sorterede pakker

Det vigtigste er, at produktet opfylder dets formål, hvilket er at transportere pakker og sortere dem afhængig af deres størrelser. Hvis det skal kunne betale sig at bruge produktet, skal det fjerne så meget arbejde fra pakkearbejderne som muligt, og det skal derfor som minimum kunne sortere pakkerne, uden arbejderne skal gøre andet end at placere dem på båndet.

Det er også vigtigt, at rullebåndet er så sikkert at bruge som muligt, da det ikke vil kunne bruges af virksomheder, hvis deres arbejdere er i fare for at komme til skade. Vi skal derfor eliminere alle farer, vi kan ved bl.a. at fjerne skarpe hjørner og kanter. Det er dog ikke alle farer, der kan elimineres fuldstændigt, og det skal derfor være muligt at stoppe rullebåndet, hvis det bliver nødvendigt. Til det formål er det vigtigt at have en nødstop, som er nem at lægge mærke til, og som man tydeligt kan se hvad gør, så alle hurtigt vil kunne stoppe båndet, hvis noget er ved at gå galt.

Det skal også være muligt at se, hvilke og hvor mange pakker, der er sorteret. På den måde er det let for brugeren at se om en bestemt pakke er sorteret endnu uden at skulle gennemrode alle pakkerne. Desuden skal det være muligt for brugeren at se statistik for hvor mange små og store pakker, der sorteres. Så får brugeren et større overblik over sorterede pakker, og det bliver lettere at organisere arbejdskraften. Det kan nok bedst gøres med en computerapplikation, da apps er gode til at præsentere data visuelt.

Løsningsforslag

Vi skal måle og sortere pakkerne, og dette kan vi gøre på flere måder. Først er målingen af pakkerne. Her havde vi flere valg; Vi kan sortere efter størrelse, vægt eller en farvekode. Disse forskellige måder at måle pakker på kunne alle være relevante, men siden vi gerne vil hjælpe en medarbejder i at finde pakkerne, er størrelsen bedst. Derudover vil vi også gerne samle noget data omkring pakkerne, så farvekoder er ikke noget, vi vil kigge på. Derfor har vi valgt at måle pakkernes størrelse og sortere i store og små pakker. Dette er der dog også flere måder at gøre på.

Vi vil selvfølgelig gerne måle pakkerne i alle tre dimensioner, men vi ved også at pakken skal køre på et rullebånd. Derfor skal vi være lidt kreative for at måle den sidste længde. Højden og bredden af en pakke er relativt simpelt. En form for sensor, der måler afstanden til pakken i toppen og i hver side af rullebåndet, giver os det data vi skal bruge til at måle højde og bredde. I stedet for sensorer kunne vi også mekanisk sortere pakker, fx med en skrå bjælke, der skubber pakker væk fra rullebåndet, hvis de er for høje. Problemet med denne løsning er, at vi ikke samler data, og at vi kun kan sortere i en enkelt dimension. Derfor bruger vi ihvertfald en form for sensor setup.

Så hvordan finder vi længden af pakkerne? Vi tænkte først, at længden kunne måles med en sensor der kommer ned foran pakken og at vi stopper rullebåndet. Denne løsning ville dog være meget svær at implementere og involvere bevægelige dele, hvilket indføre nye måder vores rullebånd kan ødelægges. Derfor tænkte vi, at vi kunne måle, hvor hurtigt båndet kører og tiden vores højde og bredde sensorer måler, og med den data finde længden. Fx hvis sensorerne måler i 20 sekunder, og båndet kører med en hastighed på 1 m/s, så ved vi, at pakken er 20 m lang.

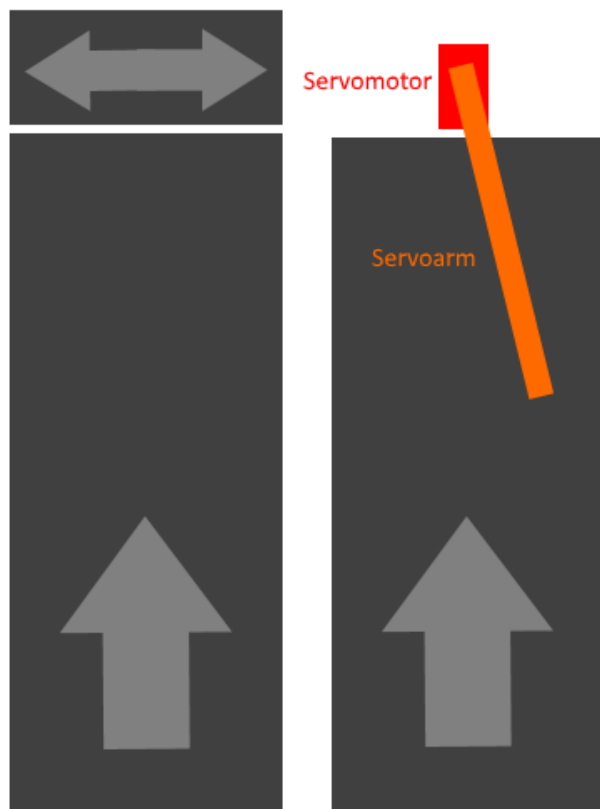
Denne løsning kræver mindst implementering, men kan være upræcis. Vi ved ikke nødvendigvis, om alle motorer kører med samme hastighed, og hvis en tung pakke skal måles, kører båndet langsommere pga. den øgede friktion mellem båndet og støttepladen. Derfor har vi valgt at måle hvor hurtigt båndet kører. På den måde ved vi altid, hvor langt båndet har bevæget sig i løbet af en målingsperiode.

Problemet er nu, at vi ikke nødvendigvis ved om sensorerne måler noget eller ej. Derfor bruger vi en laser sensor, der bliver brudt, når en pakke kommer forbi. Vi måler ikke noget med denne laser, vi ved bare, at en pakke skal måles på rullebåndet. Derfor tager vi tiden, laseren er brudt, og ganger med afstanden, vi måler, rullebåndet har bevæget sig med.

Vi har kun fundet en enkelt måde at måle, hvor langt rullebåndet har kørt. Vi bruger en skive med huller fastgjort til den ene aksel i enden af rullebåndet. Denne skive drejer, imens rullebåndet kører. Med en gaffel sensor kan vi måle hvor mange huller, der har drejet forbi. Sådan får vi alle tre dimensioner af en pakke.

Vi havde tænkt på to måder, vi kunne sortere pakkerne, efter de er blevet målt. Enten bruger vi en servomotor, der styrer en arm, der lader pakkerne skifte retning ned af rullebåndet.

Ellers kunne vi bruge endnu et rullebånd for enden af det første. Dette rullebånd kunne vi styre med en H-bro. Så kunne det køre den ene eller anden vej, angående om pakken var målt til at være stor eller lille. Denne løsning kræver bare ret meget plads og en del ekstra materiale. Derudover, vil bunkerne af sorterede pakker være på modsatte sider af båndet, hvilket ikke foretrækkes. Derfor vælger vi at ændre pakkernes retning med en servomotor.



Valget af sensorer og motorer skulle også tages. De første vi skal vælge er afstandssensorerne. Her har vi kigget på to valg. Først er der den reflektive laser-sensor. Denne type sensor bruger en infrarød laser og en fotodiode. Den kan måle afstande helt op til 5 meter, men de er lidt upræcise, og vi vil helst have den mest præcise måling vi kan. Derfor har vi valgt ikke at bruge den infrarøde laser sensor, men i stedet bruge vores anden løsning, som er ultralyds sensorer. Denne type sensor kan måle kortere afstande, men mere præcist. De er dog også lidt dyrere, men siden målingen af pakkerne er essentielt for vores produkt, vælger vi den alligevel. Motoren vi bruger til at trække rullebåndet, beregner vi senere i rapporten.



Detailprojektering - Elektronik

Introduktion

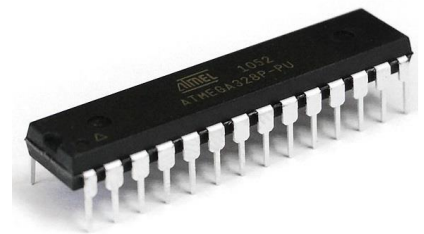
For at styre vores rullebånd skal vi bruge elektronik, og vi skal sortere pakker efter størrelse, hvilket elektronikken også styrer. Vi har tænkt os at bruge en servomotor til at lede pakkerne enten i en kasse med små pakker eller i en kasse med store pakker. For at finde pakkernes størrelse, skal vi bruge nogle afstandssensorer. Disse sensorer skal vi også kunne måle på og gemme dataene. Derfor skal vi bruge to kredsløb; et til servomotoren og et til en processor. Vores pulstog til servomotoren laves af en 555-timer og processoren, vi vælger, er en Atmega328p. Derudover bruger vi også en masse FET'er, og en Inverter, der også Schmitt Trigger signalet.

Komponenter

Atmel Atmega328p

Atmega chippen er en mikrocontroller, der har en række input og output pins, som kan programmeres ved hjælp af Arduino IDE. Programkoden behandler den data, der kommer fra sensorerne, sender dataen til computer-applikationen og sender et signal til servoen.

I starten af projektet brugte vi en Arduino UNO til at styre sorteringsprocessen. Vi indså hurtigt, at elektronikken ville fylde ret meget, og valgte derfor at lave et PCB til vores elektronik, hvor Atmega chippen ville kunne erstatte vores Arduino. Atmega chippen er den samme chip, som bliver brugt i en Arduino UNO. Det var muligt at installere Arduinos bootloader til chippen, hvilket gjorde at det var muligt for os at programmere chippen med Arduinos software.



USB to Serial Adapter (FT232RL)

En usb til seriel adapter blev brugt til at forbinde Atmega chippen med computeren. Adapteren bruges, når chippen skal programmeres.

Derudover bliver den brugt når Atmega chippen sender data til computeren.

Adapterens pins (RX, TX, GND, 5V og DTR) forbindes til Atmega chippen, og computeren forbindes ved hjælp af et mikro usb kabel.



PING - Ultralydssensor

For at finde frem til målene på pakken, bliver der blandt andet brugt tre ultralydssensorer, som kan måle afstanden til et objekt. To af sensorerne sidder overfor hinanden på hver sin side af båndet og måler afstanden til pakken. Den sidste sensor hænger over rullebåndet og måler afstanden ned til pakken. Da vi kender rullebåndets bredde samt afstanden til sensoren, der hænger over båndet, kan vi udregne målene på pakken.

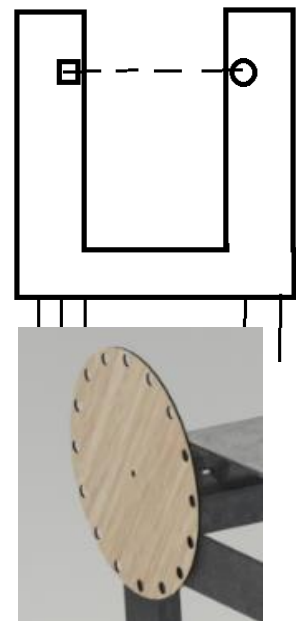
Ultralydssensoren består af en sender-del, og en modtager-del. Sensoren kan måle afstanden til et objekt ved at sende et signal og derefter måle, hvor langt tid det tager for signalet at blive reflekteret. Dette kan udregnes, da vi kender hastigheden på lyd, som er cirka 340 m/s. Ultralydssensoren kan måle op til 3 meter.



IR Gaffel sensor

Der bliver brugt en IR gaffel sensor til at beregne længden af pakken. IR gaffel sensoren er opdelt i to dele. Den ene del udsender et infrarødt lys, og den anden del måler det infrarøde lys. Når den infrarøde lysstråle bliver brudt, vil sensoren returnere et lavt signal (0). Når den infrarøde lysstråle ikke er brudt, vil sensoren returnere et højt signal (1). Vi kan på den måde se, om der er et objekt imellem IR gaffel sensoren.

For enden af den ene aksel, som roterer med båndet, er der sat en rund skive med huller i. IR gaffel sensoren sidder omkring skiven og tæller hver gang, der passerer et hul. Så længe laser-sensoren er brudt, vil IR gaffel sensoren tælle antallet af huller, som passerer. Pakkens længde kan derefter beregnes, da vi ved hvor mange huller, der passerer sensoren pr. meter.

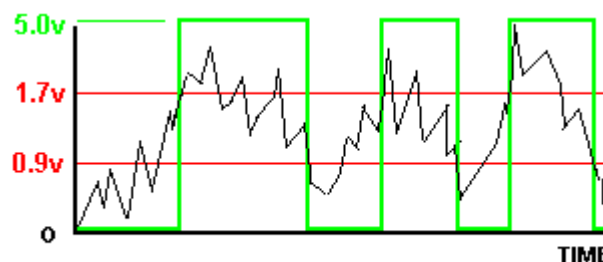
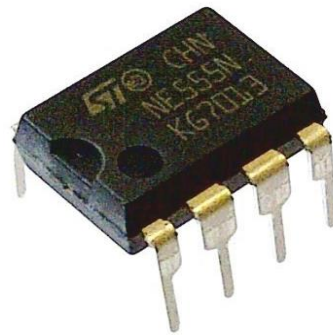


555 timer

Servo motoren skal have et signal, der bestemmer den vinkel den skal bevæge sig til. Dette signal kunne vi generere med vores atmega, men da vi startede projektet, havde vi ikke forventet at bruge den. Så derfor valgte vi at generere signalet med en 555 timer IC. Dette fastholder vi af flere årsager. Først havde vi allerede designet kredsløbet til 555 timeren. Derudover bruger vores atmega chip mindre processorkraft på en operation, vi selv kan lave. At generere en specifik puls kan nemlig være ret hårdt for vores chip. Derfor bruger vi stadig 555 timeren. Denne IC skal bruge nogle meget specifikke modstande for at generere en bestemt puls. Dette er både for at bestemme frekvensen og duty cyclen. Frekvensen er forudbestemt fra servomotor-fabrikanten. Vores skal bruge en ny puls hvert 20'ende millisekund. Pulsenes længde eller den specifikke duty cycle bestemmer vinklen. Den korteste puls er på 1 ms, og den længste er 2 ms. Ved 1 ms bevæger servoen sig til 0°, og ved 2 ms bevæger den sig til 180°. Vores servo skal rykke sig 90°, da den skal bestemme vinklen på vores sorterings arm.

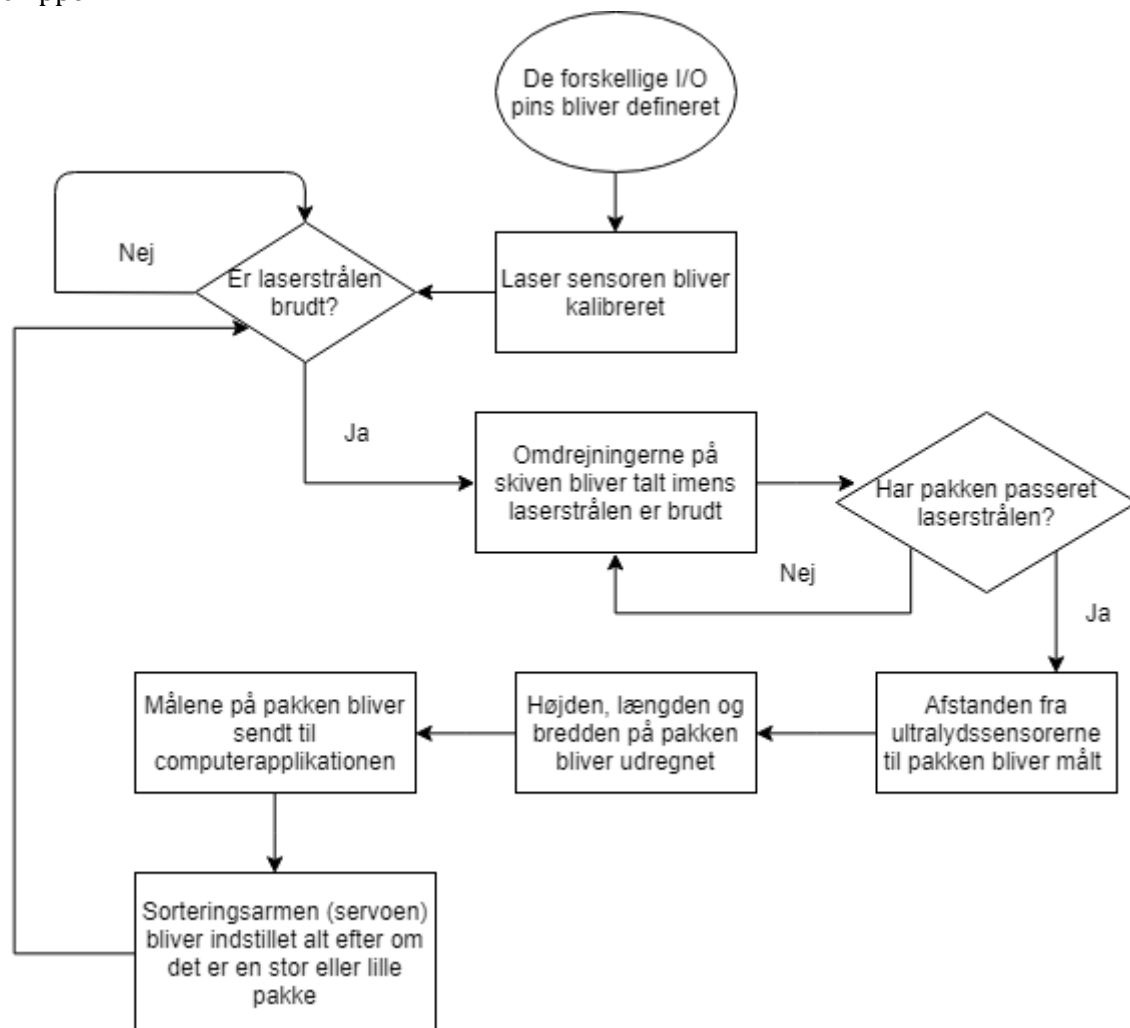
Schmitt triggered hex inverter (40106)

40106'eren indeholder seks invertere der alle har schmitt triggering. En schmitt trigger er et elektronisk hystereselag. Dette hystereselag gør, at det når der kommer fra vores sensorer bliver fjernet. Schmitt triggering er specifikt et interval, hvor der ikke må ske nogen ændring i outputtet fra IC'en. Det vil fx sige, at outputtet først ændrer sig, hvis inputtet går under 2 volt eller over 4 volt. Det betyder, at vores IR-gaffel altid giver det rigtige resultat. Der er to signaler, vi skal invertere. Det ene er IR-gaffelen, og det andet er 555 timeren. IR-gaffelens output er mere intuitivt, når det bliver invertet, og det vigtigste er hystereselaget. 555 timeren er meget præcis, så hystereselaget er ikke vigtigt. I stedet er det meget vigtigt, at vi inverterer 555 timerens output. Det skyldes, at det er nemmere at generere en duty cycle over 50%, men servo motoren skal bruge en duty cycle på under 10%. Derfor genererer vi en frekvens med en duty cycle på over 10%, og derefter inverterer vi dette signal. På figuren kan man se, hvordan outputtet fra en schmitt triggeret IC kunne se ud. X-aksen er tid, Y-aksen er volt. Den sorte linje er input volt, de røde er schmitt triggerens interval, og den grønne graf er outputtet. Vi ser her, at intervallet går fra 0,9 V til 1,7 V. Det betyder, at outputtet først skifter til 5 V, når inputtet er over 1,7 V, og først skifter til 0 V, når inputtet går under 0,9 V. Schmitt triggering gør altså, at digitale signaler bliver meget mere stabile.



Programkode

Atmega chippen bliver programmeret gennem Arduino IDE, og styrer alle processerne med måling og sortering af pakkerne. Her ses et flowchart over den kode, der bliver kørt på Atmega chippen:



Alle de forskellige værdier og input/output pins defineres.

Værdien fra lasersensoren kan variere alt efter hvilken laser, der bliver brugt, og hvordan laserstrålen rammer sensoren. Lasersensoren bliver derfor kalibreret ved opstart af programmet. Laseren slukkes, og der bliver lavet tre målinger, og derefter bliver laseren tændt, og der foretages tre målinger igen. De målte værdier, kan nu bruges til at se om laserstrålen er brudt. Hvis en pakke bryder laserstrålen, vil længden på pakken blive målt. Dette sker ved at IR gaffel sensoren måler omdrejningerne på skiven, som sidder for enden af båndet på den ene aksel. Da båndets hastighed er konstant, kræver det blot, at vi definerer, hvor mange omdrejninger, der går per meter. Vi kan derefter udregne pakkens længde. Efter længden er blevet målt, bliver ultralydssensorerne brugt til at måle de resterende sider på pakken. Endelig vil programmet sende dataen til computerapplikationen, som vil behandle dataen, og oprette pakken i vores system.

Windows Forms Applikation

Applikationen er lavet i Visual Studio, og programmeret i sproget C#. Den skal gøre det muligt for brugeren at holde styr på alle pakkerne, og se målene på hver enkelt pakke.

Når applikationen starter op, skal man forbinde til sorteringsbåndet, ved at vælge den korrekte port, og trykke "Forbind". Applikationen vil nu læse den data, som Atmega chippen sender. Når en pakke bliver målt, vil applikationen oprette et ID til pakken og tilføje den til listen. Man kan nu vælge en pakke fra listen og se alle målene på pakken. Derudover er det muligt at se, om det er en stor eller lille pakke. I toppen af applikationen kan man se, hvor mange pakker, der er sorteret, og hvor mange af de pakker, der er store/små.

Sorter med manér

R COM3 Afbryd

SMÅ PAKKER	STORE PAKKER
000113	000048

PAKKE INFORMATION

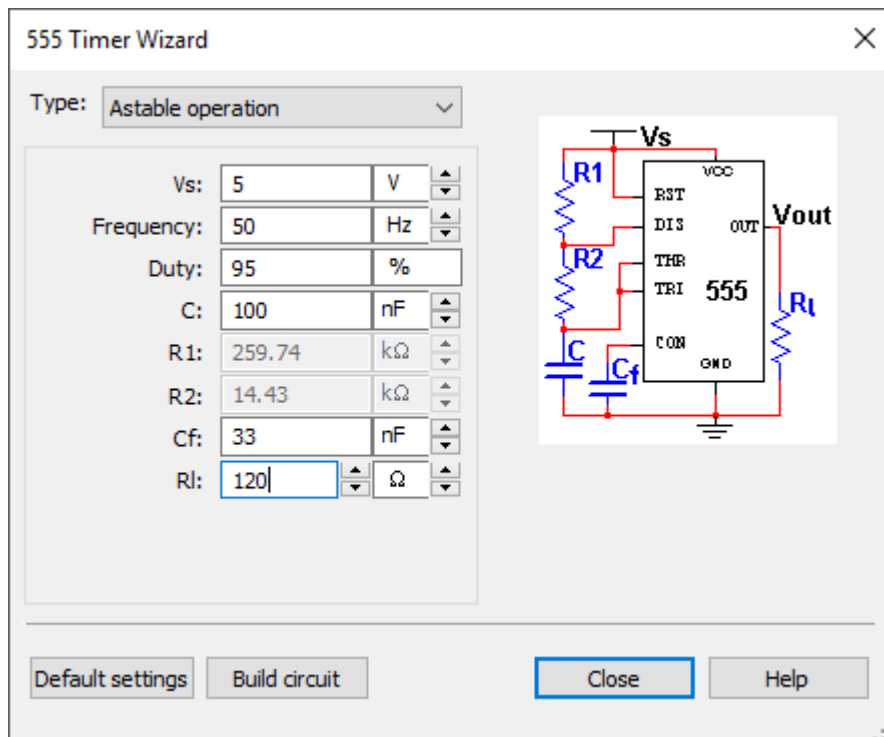
Vælg en pakke

	Information
PAKKE NR. 88	Pakke nr: 96
PAKKE NR. 89	Pakke kategori: LILLE
PAKKE NR. 90	
PAKKE NR. 91	Pakkens højde: 5 cm
PAKKE NR. 92	Pakkens bredde: 6 cm
PAKKE NR. 93	Pakkens længde: 7 cm
PAKKE NR. 94	
PAKKE NR. 95	
PAKKE NR. 96	
PAKKE NR. 97	
PAKKE NR. 98	
PAKKE NR. 99	
PAKKE NR. 100	
PAKKE NR. 101	
PAKKE NR. 102	
PAKKE NR. 103	
PAKKE NR. 104	

Ryd liste

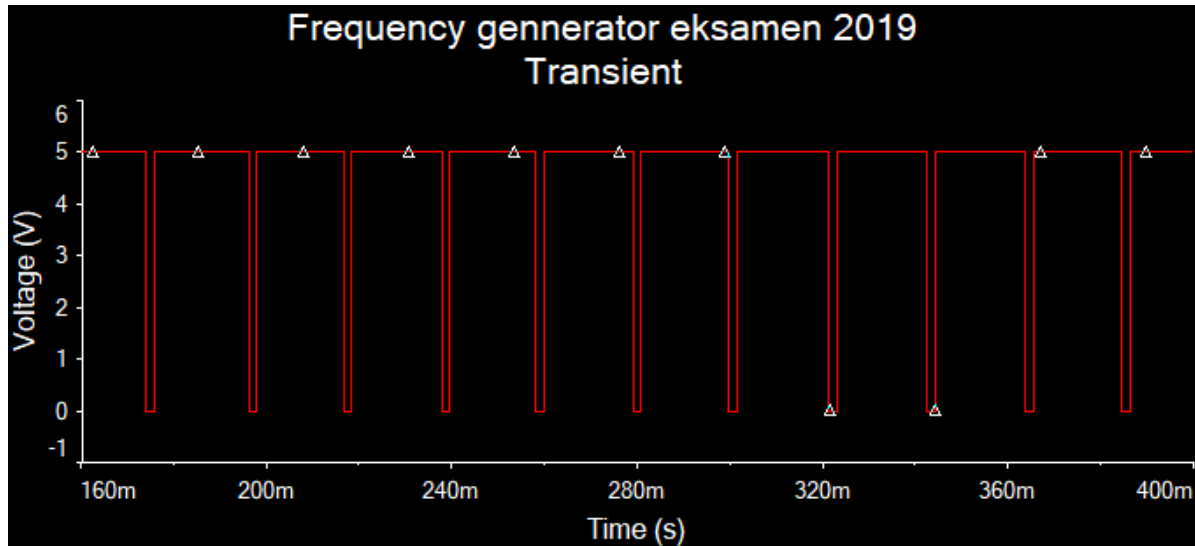
Elektronik - Kredsløb

For at starte med at bygge kredsløbet, bruger vi Multisim's '555 timer wizard'. Dette er en hurtig måde at få et kredsløb med en 555 timer.

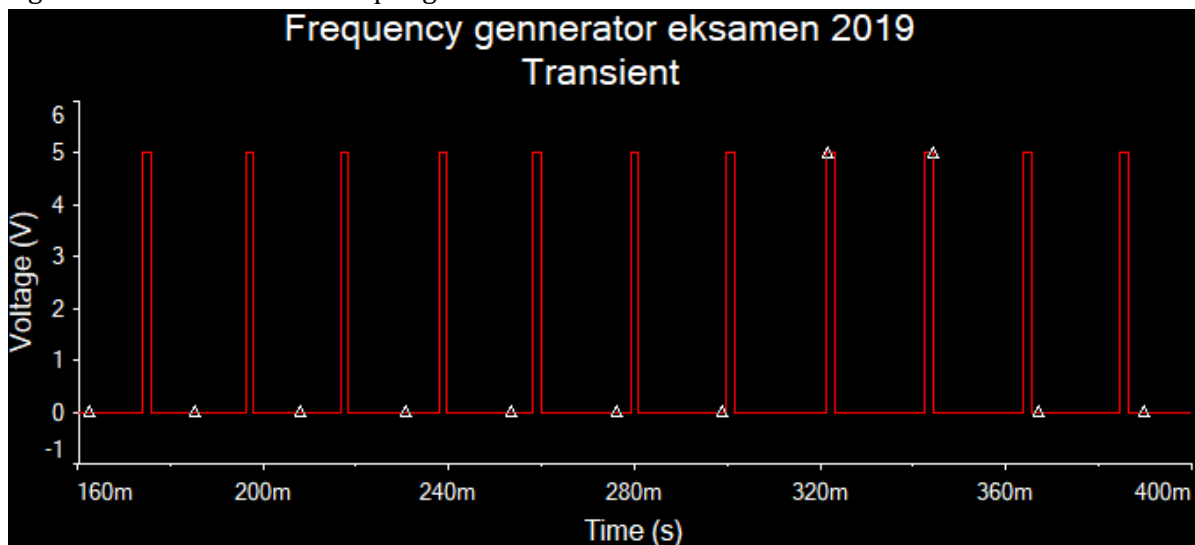


Vi ved at kredsløbet skal bruge 5 volt. Grunden til at jeg bruger 50 hz, er at servomotoren skal bruge en puls hvert 20. ms. For at omregne fra frekvens til periode tid, bruger man formelen: $F = \frac{1}{t}$. $F = \frac{1}{0.02} \rightarrow F = 50$. Dernæst bruger vi en duty cycle på 95 %, hvilket vi inverterer senere. Dette betyder, at pulsen vil være 'høj' eller 5 V i 5% af de 20 ms. Altså i 1 ms, hvilket er den korteste puls vores servomotor kan udnytte. Jeg bruger en 100 nF capacitor, da det gør, at modstanderne man skal bruge er mindre. Siden modstandene, man skal bruge er ulige tal, sætter vi to modstande i serie for at få den rigtige resulterende modstand. For at få 260 kΩ, bruger vi en 220 kΩ og en 39 kΩ bagefter. For at få 14.4 kΩ, bruger vi en 12 kΩ modstand, da det ikke er super vigtigt, at vi rammer den specifikke modstand. Alle modstande vi bruger kan i forvejen afvige med op til 5% i modstand fra det specificerede. Det vil altså sige, at vores modstand specificeret med 39 kΩ, praktisk set kan være 37.05 - 40.95 kΩ. Når vi sætter 555 kredsløbet ind, kan vi måle outputtet med multisim.

Her er outputtet direkte fra 555 timeren:

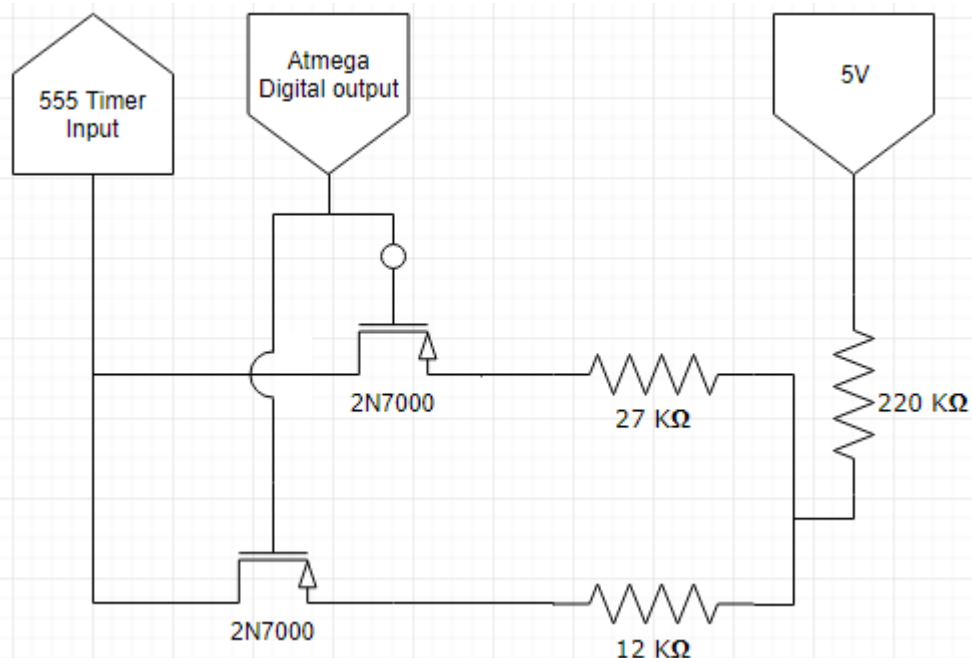


Og her er det inverterede output gennem 40106'eren:

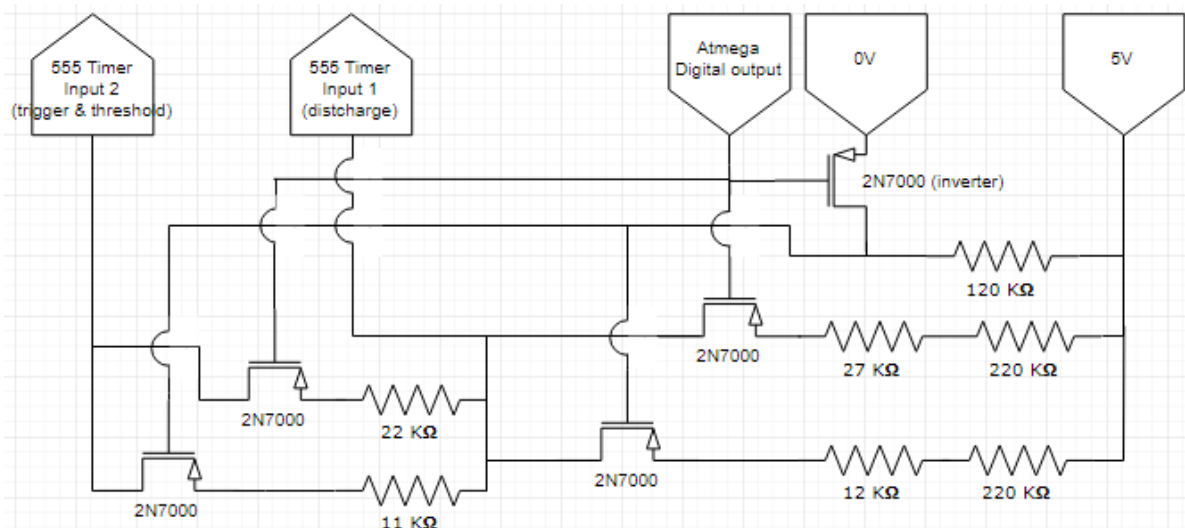


Dette pulstog sætter servomotoren ved den første position. Vi skal dog også bevæge servomotoren til den anden position, så vi kan sortere pakker. Den anden position skal være 90 grader væk fra den første position. Dette kan oversættes til en puls, der er 0,5 ms længere, da servomotoren kan bevæge sig 180 grader i alt og input-pulsene går fra 1 ms til 2 ms. Dette betyder at 0,5 ms giver 90 grader, hvilket også betyder, at den nye duty cycle vil være 92,5%. Så bliver det inverterede duty cycle 7,5%, hvilket i en 50 hz frekvens giver pulser, der er 1,5 ms lange. Nu har vi altså 2 555 timer kredsløb, der giver hver deres puls til servomotoren. Dette er selvfølgelig utroligt upraktisk og spild af PCB plads. Derfor samler vi de to kredsløb, så vi kan styre pulsen digitalt igennem en enkelt 555 timer. Til denne digitale styring, bruger vi 2n7000 FET transistorer. Vi startede med at bruge BC547 standard diode transistorer, men kredsløbets spænding igennem modstandende er meget vigtig, så en FET er et bedre valg.

Her er en meget forsimplet kredsløbstegetning over hvordan vi digitalt styre 555 timeren:

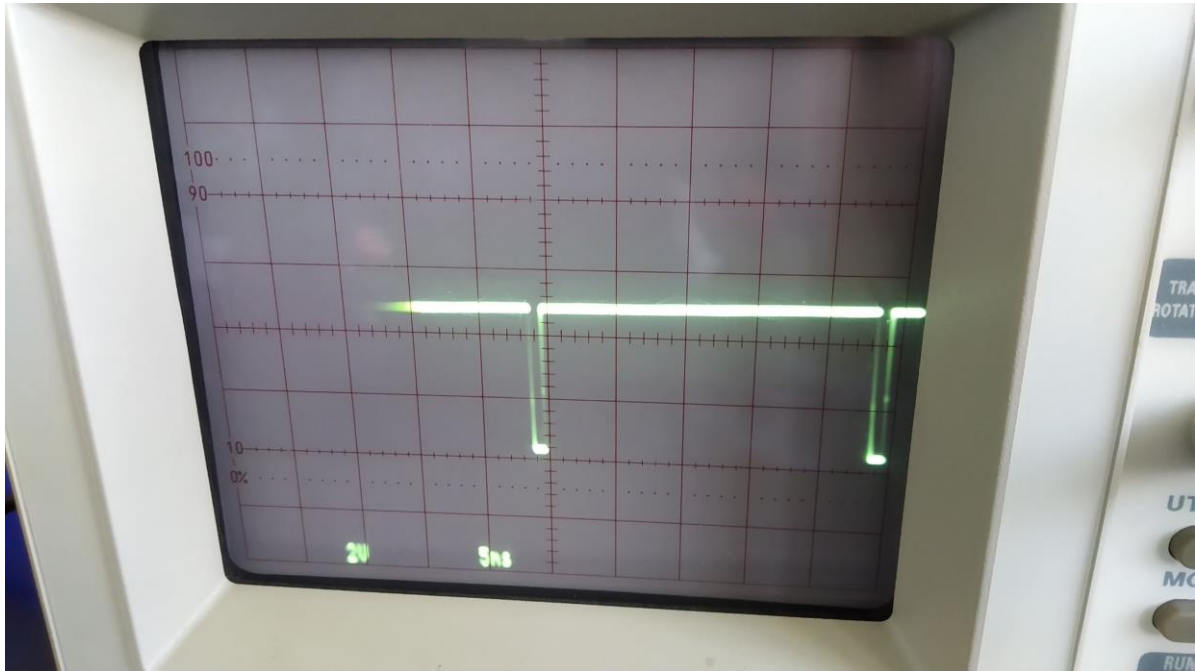


Siden at 555 timeren styres af en specifik modstand, vælger vi digitalt, hvilken der løber ind i 555 timeren. På kredsløbs tegningen, løber 555 timerens trigger igennem de forskellige modstande, men kun den ene bliver ledt videre ind i 555 timerens input. Da den ene transistor har et inverteret gate pin, fungerer kredsløbet som en toggle. I virkeligheden har 555 timeren to inputs, der hver skal have en speciel modstand. Derfor har vi også to toggle kredsløb, med 4 FET'er i alt. På tegningen bruger jeg en cirkel som inverter symbol, men i det rigtige kredsløb bruger vi også en FET som inverterer. Derfor bruger vi 1 FET, som inverterer og 4, som toggler. En mere fuldstændig kredsløbstræning er herunder:



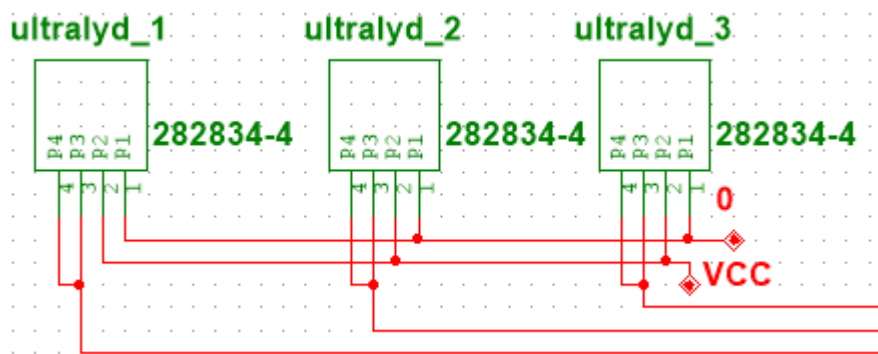
Her kan man se, hvordan vi bruger en FET som inverter, og hvordan vi sender to signaler ind i de to 555 timer input ben. Vi bruger også en 4001 diode på 5 V, så vi ikke ved en fejl sender den strøm igennem kredsløbet med den forkerte polaritet.

Vi har bygget kredsløbet på et breadboard og brugt et oscilloskop til at måle outputtet fra 555 timeren, og kan se at det er de rigtige frekvenser den sender ud:



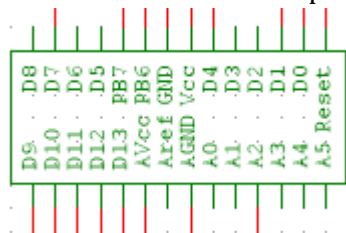
Man kan se, at hvert felt har enheden 5 ms., og vores puls fylder ca. et halvt felt. Derfor måler vi pulserne til at være 1 ms lange, hvilket er præcis det, vi skal bruge. Vi har selvfølgelig optegnet dette kredsløb i multisim. Her har vi også indsat vores atmega328 chip, hex inverteren og alle portene.

Multisim

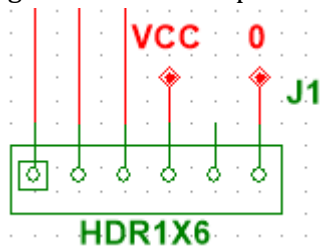


Her er et eksempel fra vores multisim-kredsløb. De tre porte er til vores ultralydssensorer. Det første ben for stel (0), det andet 5 volt (VCC) og de to sidste ben er kortsluttet og sender data. De tre datalinjer bliver sendt til arduinoen.

Arduinoen ser sådan ud på kredsløbstræningen:

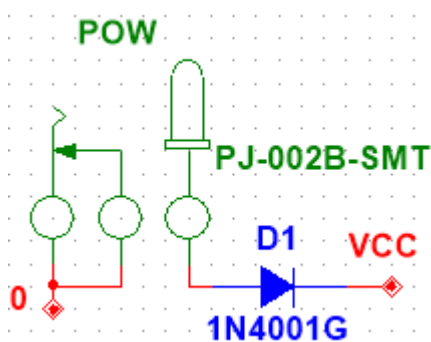


Der er flere pins vi ikke bruger, siden det er en begrænset mængde data, vi skal sende. D9 til D12 bliver brugt til sensor data. D13 bliver brugt som led pin. De to VCC og GND bliver respektivt forbundet til 5 V og grund. A2 bliver brugt som analog input. Vi bruger D7 som digital output til 555 timerens FETs. PB7 og PB6 er specielle pins, der skal bruge en crystal timer. D4 er til at kalibrere laser sensoren. D1 og D0 er vores data pins, når vi sender data gennem usb. Reset pin'en er forbundet til 5 V gennem en modstand, men kan kortsluttes igennem en knap, hvilket genstarter arduinoen. Vi sender data gennem et usb stik med en FT232RL chip. Dette komponent har vi købt færdiglavet på et PCB. Det betyder, at vi bare skal tilslutte 6 pins til vores PCB og så har vi en USB port vi kan skrive til og aflæse.



Her er porten i multisim. Den første og tredje forbindes til spænding, sådan kan vi forsyne vores PCB med 5V gennem USB stikket. De to næste bliver ledt videre til atmega-chippens TX og RX pins, altså D0 og D1. Den sidste pin er til reset. Den slutter vi til atmega chippens reset pin, igennem en capacitor for at fjerne støj, så vi er sikre på at atmegaen ikke genstarter tilfældigt. Fordi servomotoren er forbundet direkte til PCB'et, og det kan trække flere ampere end et USB stik kan levere; >0.2A, bruger vi en ekstern strømkilde. Vi har valgt at sætte et jack stik til strøm på vores PCB. Specifikt et 2 mm power jack.

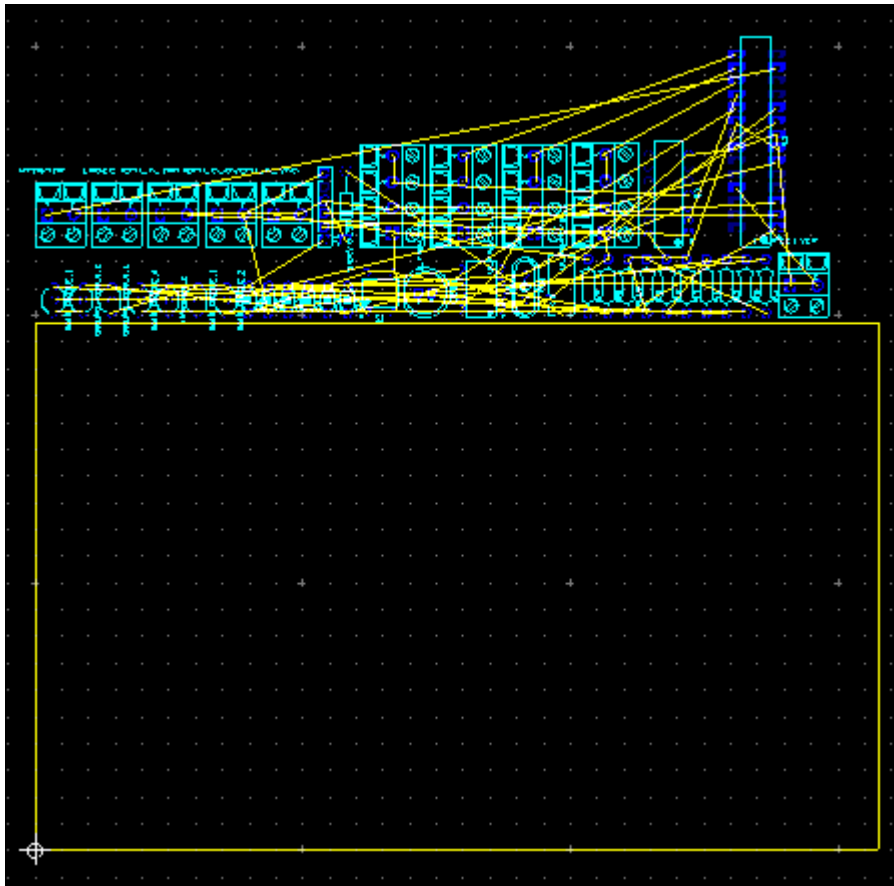
Her er power jack stikket i multisim:



Her kan man også se 4001 dioden, der sikrer polariteten er rigtig. Vi har lavet vores eget 2 mm jack stik og loddet det til et USB stik, så man kan slutte PCB'et til en mobil oplader eller lignende. Det eneste komponent vi mangler at forklare er 40106 hex inverteren, men den er ret simpel at forstå i et kredsløb. Den skal bare have 5v og stel, og så skifter inputs og output med de lige og ulige pins. Med alle vores sensorer, strøm stik og usb, har vi 11 porte i alt og det er egentligt hele vores kredsløb. Det hele er tegnet ind i multisim, men vi vil gerne printe PCBet på en kobberplade. Til denne opgave bruger vi Ultiboard.

Fremstilling med Ultiboard

Når man først overføre sin kredsløbstræning til ultiboard, for man dette billede:

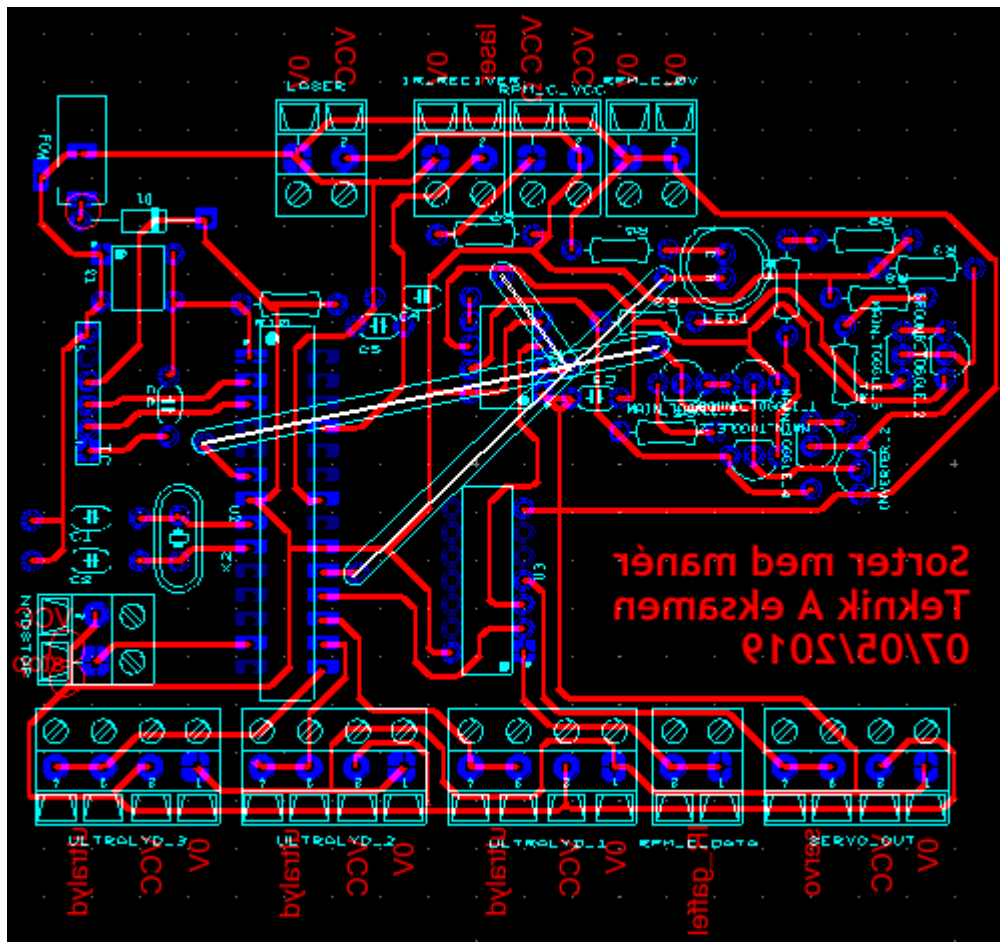


De blå objekter, er alle komponenterne, de gule linjer er de ledninger der skal routes og den gule rektangel er det område man må arbejde inden for. Vi har valgt selv at route ledningerne. Det er fordi der er nogle krav til boardet vi har opstillet.

Krav til PCB:

- 1) Vores porte skal alle være let tilgængelige.
 - 2) PCB'et skal bruge mindst muligt kobberplade.
 - 3) Det skal være nemt at troubleshoot, hvis noget går galt.
- + Hvis man manuelt router kan det være man finder en fejl undervejs.

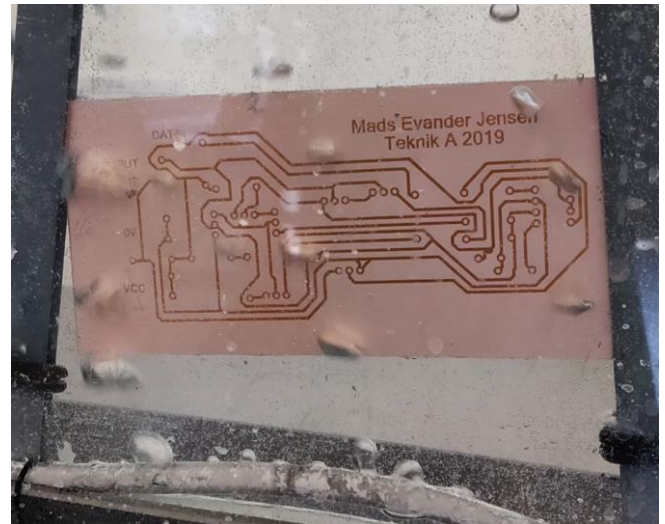
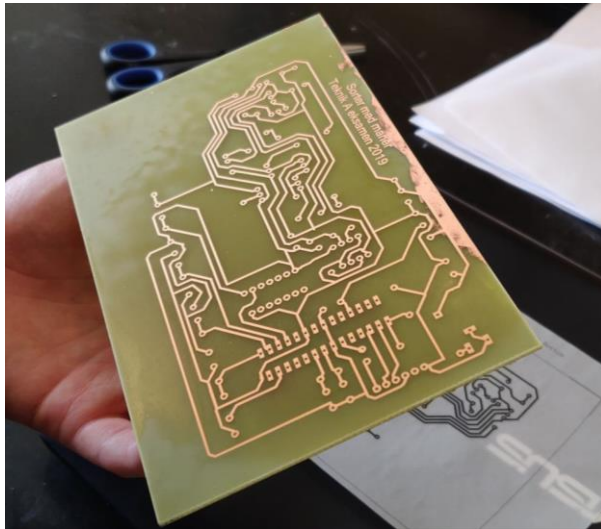
Derfor har vi manuelt gennemgået alt routing. Dette er en meget lang process, siden vi kun må bruge en enkelt side til routing, og der må selvfølgelig ikke være nogen kobber baner, der overlapper. Efter mange timers arbejde, har vi fået PCB'et ned til at bruge 3 jumper wires. Her kan man se det endelige resultat:



Vi har forsøgt at få alle komponenterne så tæt som muligt. Vi har alle portene i kanten af PCB'et og delt det op så 555 timerens kredsløb er isoleret i hjørnet, og IC'erne er i den anden ende. Denne opdeling gør at vi nemt kan finde hvis der er fejl. Vi har også gjort det nemmere for os selv og gjort alle loddehullerne større.

Før vi begynder at lave PCB'et, skærer vi en kobberplade til. Kobberpladerne er specielt lavet til PCB, og kommer med et beskyttende klistermærke. Når man først har taget det af skal man skynde sig at belyse pladen med UV-stråling siden at lyset fra omverdenen har en effekt på pladen. Arbejdstegning fra Ultiboard kan vi printe på UV-gennemsigtigt papir. Dette papir lægges vi oven på en kobberplade og belyser den med UV-stråling. UV-strålerne ødelægger et beskyttende lag overalt på kobberpladen, undtagen dér hvor vores PCB print ligger. Dernæst kommer kobberpladen i et bad med fremkaldervæske og endeligt i et syrebad. Dette fjerner det kobber der ikke længere er beskyttet og vi for ved dette step vores kobberbaner at se. Nedenunder kan man se kobberpladen i syrebadet og efter vi har sprayet det med et beskyttende lag.





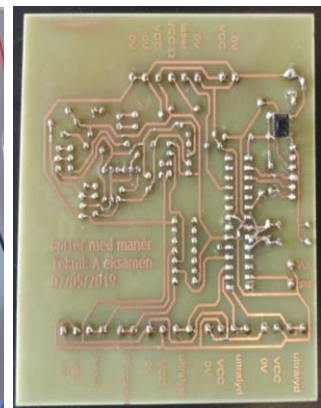
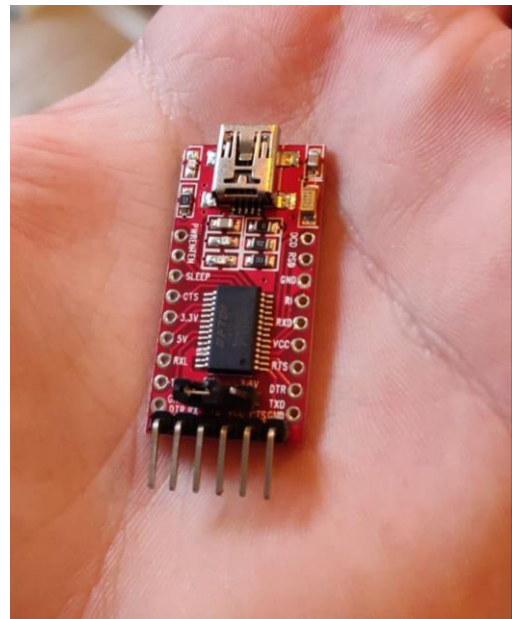
Det sidste step i processen er at bore huller i printpladen og lodde alle komponenterne på.

Vi bruger en FT232RL chip til at skrive til arduinoen igennem et USB-stik. Dette virker dog ikke med det samme, før man kan skrive til chippen, skal man bruge en speciel driver. Det er en FTDI enhed der skal bruge en D2XX driver. Dette sætter en virtuel COM port op, som man igennem Arduino's software kan skrive til. Driveren vi bruger fås her:

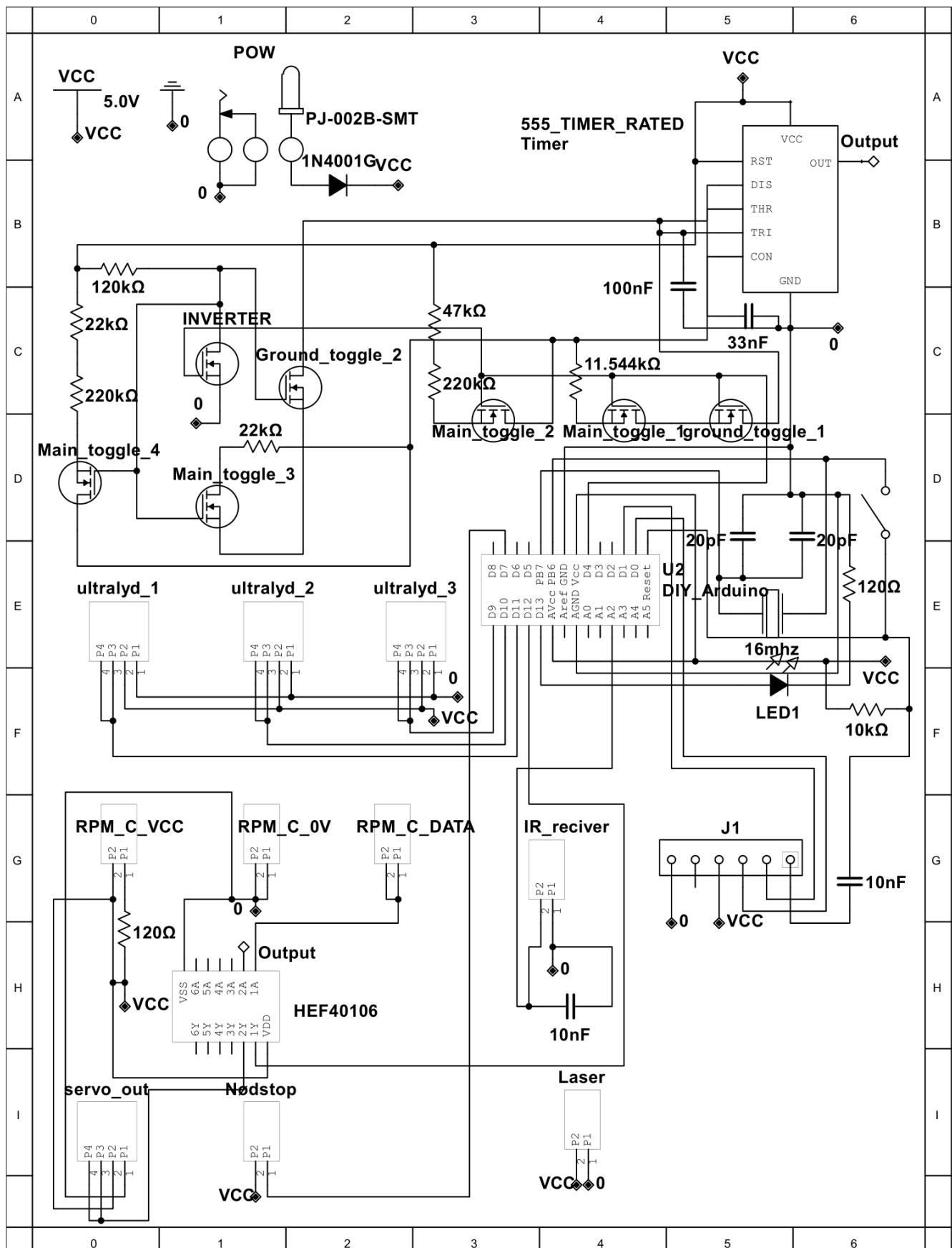
<https://www.ftdichip.com/Drivers/D2XX.htm>

Det endelige PCB ses herunder:

Man kan tydeligt se vores beskrivende tekst af hver port. Vores store atmega chip, der sidder i midten til venstre, alle portene er i kanterne og 555 timer kredsløbet med transistorerne er isoleret for sig selv. Vores strømstik er i det øverste venstre hjørne. Serial til USB-chippen sidder i headeren i venstre side. 16 mhz krydstallen sidder på venstre side af atmega chippen. Man kan også se vores genstart knap der er under strømstikket og vores test led der sidder over 555 kredsløbet. De tre jumper wires har vi prøvet at gøre så korte som muligt, så de ikke er i vejen. Alle output ledninger, vi skruer i terminal portene, er farvekoordineret. Rød og sort er selvfølgelig 5v og grund. De blå ledninger er data. Vi har testet alt på PCB'et og det hele virker som forventet.



Bilag 5



Bilag 6

Ultiboard-Frequency generator eksamen 2019 NEW NEW - 12/05/2019 - 16.26.43 Copper Bottom

