



Home Protection

Semesterprojektgruppe 5

Hardware Dokumentation

Indholdsfortegnelse

Hardware design	1
HW1 Simulink	1
HW2 Zero Crossing	6
HW3 Carrier Generator	9
HW4 Carrier Detektor	19
HW5 Switch/dimmer	24
HW6 Tolerance og Produktion	27
HW7 Modultest hardware	31
Referenceliste	35

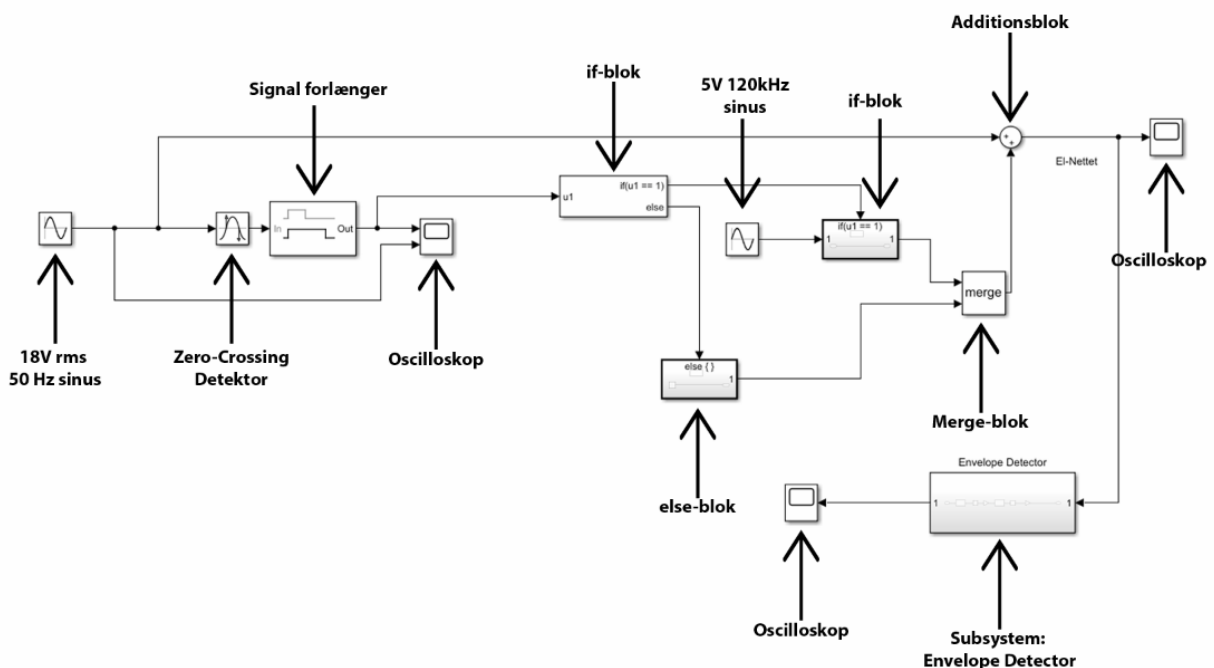
Hardware design

HW1 Simulink

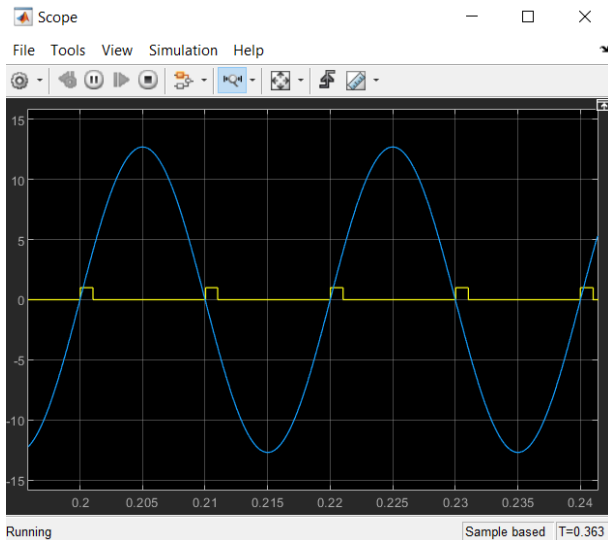
Grundet de specielle omstændigheder der har fundet sted i projektets forløb med COVID-19 har en realisering af hardwaren, samt integrationen mellem software og hardware, ikke været muligt. Det har derfor været nødvendigt at kigge på alternative metoder at designe hardwaren samt produktets funktionalitet. Der er blevet valgt at lægge en del tid i egen oplæring af simuleringsprogrammet *Simulink*, som er en del af *MATLAB*. Simulink er et grafisk værktøj som muliggør modellering, analyse og simulering af systemer konceptuelt. Programmet giver stor frihed til at opbygge sit system og teste om det vil virke. Der benyttes blokke i programmet, hvor værdier og virkemåde kan modificeres på egen hånd. De forskellige blokke kobles med en enkelt streg IN/OUT. Simulink er et værktøj utrolig mange muligheder for modellering og tests, grundet konfigurations indstillinger samt tilgængelige blokke i programmet.

Værktøjet er blevet benyttet i design-delens begyndelsesfase. Her er hele X10 systemet opbygget og der kigges på virkemåden overordnet, samt sammenspillet mellem de enkelte blokke. Det overordnede system kan ses på figur 1. Systemet er bygget op af en zero-Crossing blok, der er her tilkoblet et scope, for at se hvordan det konceptuelle Zero Crossing ville se ud. Det giver en bedre forståelse af virkemåden og bliver herved også nemmere at koble det sammen med den forlæste teori. På scopet ses et firkants-signal der symboliserer zero crossing (det gule signal). Det blå signal er det oprindelige sinus-signal på figur 2.

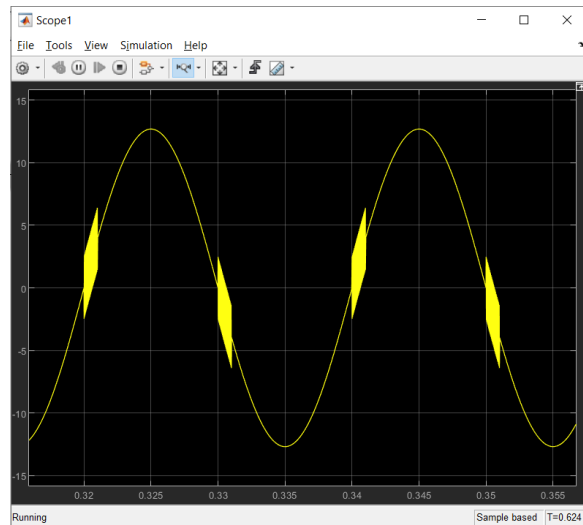
Ud fra den overordnede viden om systemet, vides det at mikrocontrolleren aflæser disse zero crosses, og derefter kan mikrocontrolleren på baggrund af dette, sende data i form af et 120kHz signal ud på lysnettet på det korrekte tidspunkt. For at lave dette koncept i Simulink, er det blevet lavet en if-blok. I alt sin enkelthed vil if-blokken sørge for at sende et 120-kHz sinus signal ud og addere det med det oprindelige sinus-signal, hvis der er zero-crossing. Hvis der ikke er zero-crossing, så sker der intet med det oprindelige signal. Herved simuleres en sinus-kurve, som den ville se ud på lysnettet på figur 3.



Figur 1: Opsætning af simulink-blokke



Figur 3: Scope billede af zero-crossing



Figur 2: Scope billede af hvordan EI-net vil se ud

Efter at der er blevet sendt data ud på lysnettet skal det være muligt at opfange disse data ved et modtager-modul. Princippet bag modtager modulet er at det udsendte data skal opfanges af en Envelope Detector, der er blevet implementeret i et subsystem. Indholdet af subsystemet kan ses på figur 4.



Figur 4: Envelope detector

I Envelope Detectoren ses først et analogt højpas-filter, hvoraf bode plottet for filteret ses på figur 5. Højpas-filteret er lavet ved at vælge en filter blok, og herefter indsætte 'Denominator coefficienterne'. Da der kun er 2 numerator coefficienter vides der at det er et første ordens-filter. Denominator coefficienten der indsættes er en α -værdi, eller knækfrekvensen, for den overføringsfunktion der ønskes for filteret. Højpas-filteret bliver designet så der er en knæk-frekvens på 10 kHz vha. formelen:

$$f_c = \frac{1}{2\pi RC}$$

Frekvensen sættes ind, samt en kondensator med værdien 10 nF.

$$R = \frac{1}{2 \cdot \pi \cdot 10000 \cdot 10 \cdot 10^{-9}} = 1.592 \cdot 10^3$$

Modstanden der skal bruges i høj-pas filteret vil altså være 1.6 k Ω og kondensatoren 10 nF.

Ved at opstille høj-pas filteret kan man finde overføringsfunktionen ved:

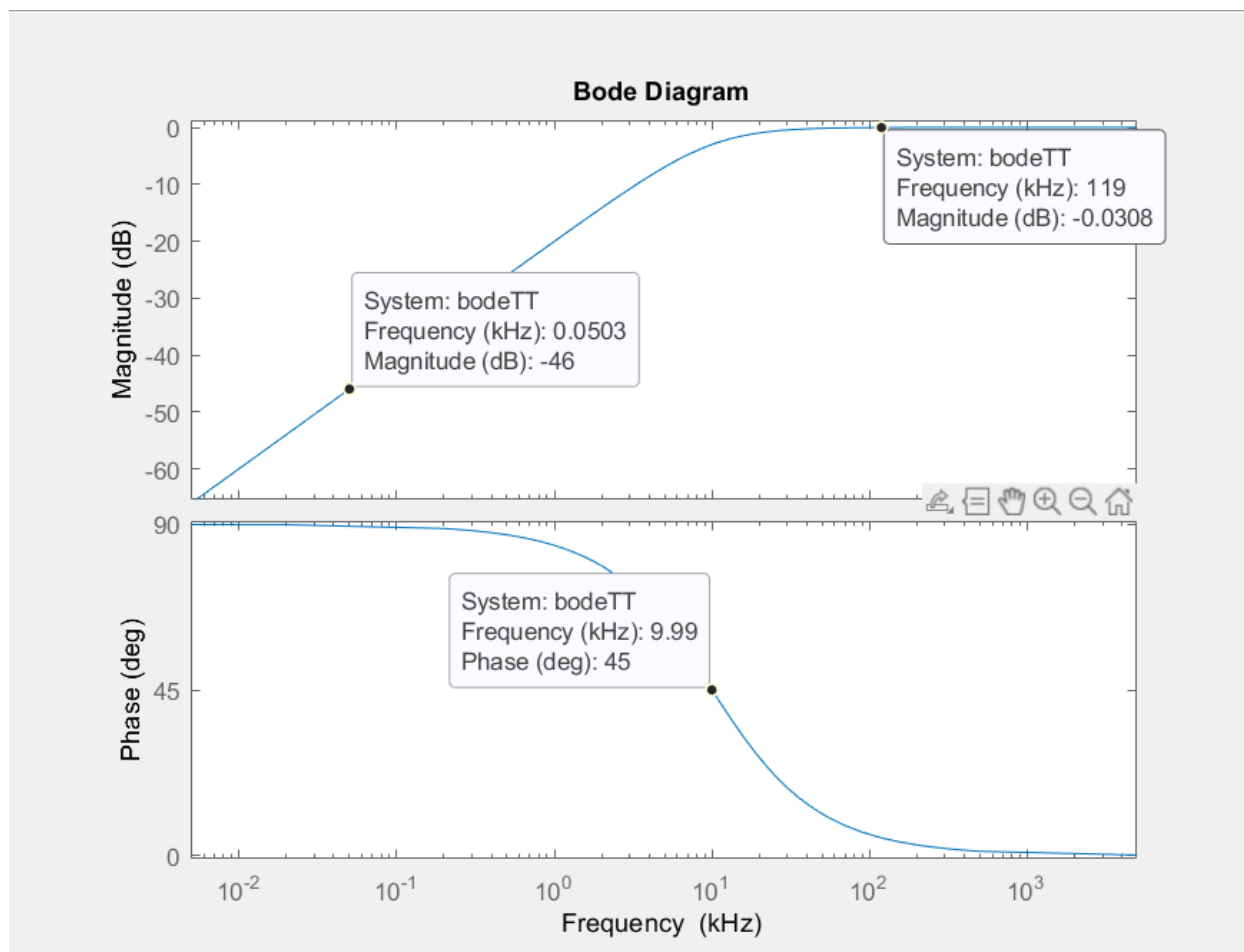
$$G(s) = \frac{R}{\frac{1}{s \cdot C} + R} \Rightarrow \frac{R \cdot C \cdot s}{R \cdot C \cdot s + 1} \Rightarrow \frac{s}{s + \frac{1}{RC}} \Rightarrow \frac{s}{s + \alpha}$$

Hvis R og C indsættes fås overføringsfunktionen:

$$G(s) = \frac{1.592 \cdot 10^3}{\frac{1}{s \cdot 10 \cdot 10^{-9}} + 1.592 \cdot 10^3} = 1 - \frac{6281}{s + 6281}$$

Ud fra standard bodeplot¹ vides at α er lig med knækfrekvensen, standard bodeplottet stemmer overens med det bode plot der er blevet genereret af Simulink.

Se figur 6 for at se implementering af høj-pas filteret. Bodeplottet viser, at der ved 50 HZ ca. dæmpes med 56 DB, ved 120 kHz dæmpes der ikke. Samt at knækfrekvens er ved 10 kHz.



Figur 5: Bode plot af det først filter, som er et højpas filter

¹ Ref[1]

Parameters

Numerator coefficients:
[1 0]

Denominator coefficients:
[1 6.281e4]

Absolute tolerance:
auto

State Name: (e.g., 'position')
"

Figur 6: Høj-pas filter

Det næste filter der ses i envelope detectoren er et digitalt midlingsfilter, som tager gennemsnit over de sidste samples. Midlingen foregår over 1000 gange sample time, hvor sample time er $2e^{-7}$. Se figur 7 for implementering af filteret.

Main Data Types

Coefficient source: Dialog parameters

Filter structure: Direct form

Coefficients: 1/1000.*ones(1, 1000)

Input processing: Elements as channels (sample based)

Initial states: 0

Control

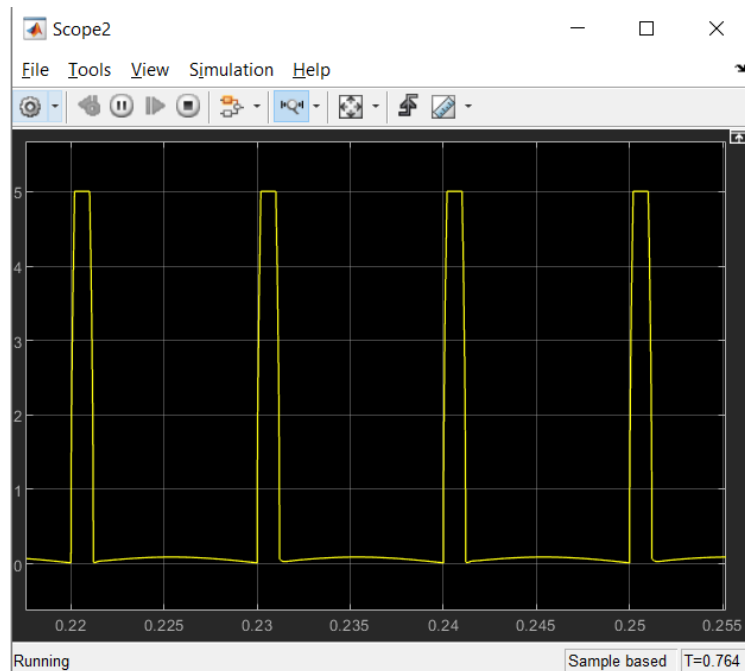
☐ Show enable port

External reset: None

Sample time (-1 for inherited): 2e-7

Figur 7: Midlingsfilter

Det sidste scope, på figur 1, viser outputtet efter envelope detektoren se figur 8 for billedet af scopet. Her ses et logisk højt signal hver gang der er 120 kHz bursts. Dette simulerer hvordan modtageren skal virke. Når der bursts er der et logisk højt signal og når der ikke er bursts så er der et logisk lavt signal. Ved at modtager arduinoerne kan se disse logiske høje og lave signaler, og sammenligne med med zero crossing kan der skabes en sekvens af binære tal vha. manchester kode.



Figur 8: Envelope Detector

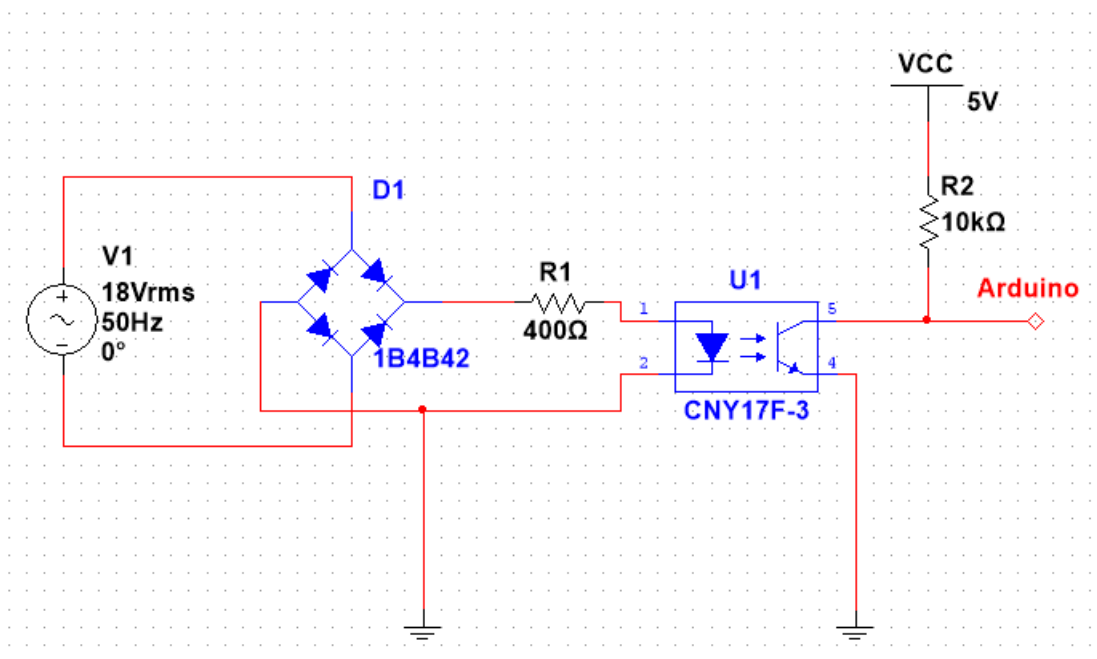
Ved at se på det samlede kredsløb og de forskellige scopes kan der konkluderes at det konceptuelle kredsløb burde virke. Der er altså blevet påvist at de 3 moduler, Zero-Crossing, Sender, og Modtager kan kommunikere med hinanden. Dette vil dog kun være muligt hvis de forskellige moduler bliver designet således at de forskellige signaler der bliver sendt ud på kredsløbet, overholder de samme principper, som der er blevet påvist i Simulink. Samt at signaler bliver bearbejdet på samme måde som de forskellige blokke i simulink bearbejder signalerne.

Simulink redskabet er blevet brugt som en tilføjelse til projektet, da det ikke har været muligt at bygge et produkt. For at lære Simulink redskabet at kende, har en del af projektgruppen brugt en stor mængde tid på at eksperimentere med programmet samt fulgt et online kursus², for at få en basis forståelse af programmet. Herefter har der med en smule vejledning været muligt at danne basale kredsløb samt basal forståelse for programmet til fremtidigt brug.

² Ref[2]

HW2 Zero Crossing

For at kunne kommunikere over lysnettet er det nødvendigt at vide hvornår der er zero crossing. Dvs. når 18 V AC-signalet på 50 Hz krydser 0 V. For at signalere til arduinoen hvornår der er zero crossing opbygges kredsløbet således at der er et output på 5 V, altså et logisk 1, når der er zero-crossing, og 0 V når der ikke er. Se figur 9 for design af zero-crossing.



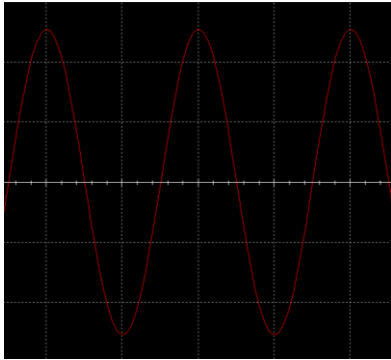
Figur 9: Zero-crossing opsætning

Zero Crossing modulet er lavet ud fra en online undersøgelse³, her blev der hurtigt fundet ud af at der skulle bruges en optocoupler og en diodebro. Ved behandling af det harmoniske signal på 50Hz ønskes det at de negative halvperioder vendes til positive, det er her diodebrokobleren benyttes. Dette illustreres på figur 10 og 11. Diodebrokobler er i sin simpelhed 4 dioder koblet sammen. I designet er der benyttet diodebro komponenten 1B4B42⁴, samt optocoupler CNY17F-3⁵, specifikationerne for hver komponent kan findes i deres respektive datablade.

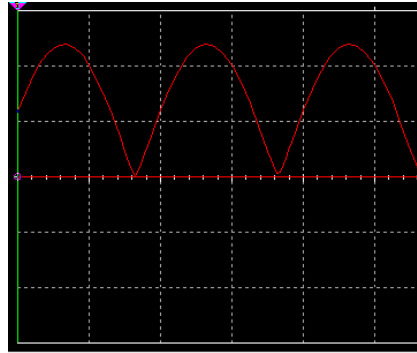
³ Ref[3]

⁴ Ref[4]

⁵ Ref[5]



Figur 10: Før diodebro



Figur 11: Efter diode bro

Optocoupleren benyttes så Arduinoen beskyttes fra kredsløbet, da den ikke kan klare 18 V 50 Hz kredsløbet. Optocoupleren benytter sig af en lysdiode, når lysdioden lyser, løber der en strøm igennem fototransistoren. Herved løber de 5 V fra forsyningen direkte til ground. Når sinuskurven nærmer sig 0 vil lysdioden stoppe med at lyse og spændingsforsyningen kan løbe ind til Arduinoen, der sættes en 10k Ω modstand foran for at sikre at der ikke løber en for stor strøm ind i Arduinoen.

$$\frac{5V}{10k\Omega} = 0.5 \text{ mA}$$

Arduinoens i/o pins kan ikke håndtere mere end 40 mA⁶, herved sikres at mikrokontrolleren ikke ødelægges.

Ud fra optocouplerens datablad er der blevet bestemt at der er en max-current på 60 mA og et spændingsfald på 1.65 V. Herved bestemmes for-modstanden således at max-current ikke bliver overskredet.

Ved et 18 V RMS-kredsløb bestemmes peak amplituden.

$$18V \cdot \sqrt{2} = 25,456 \text{ V}$$

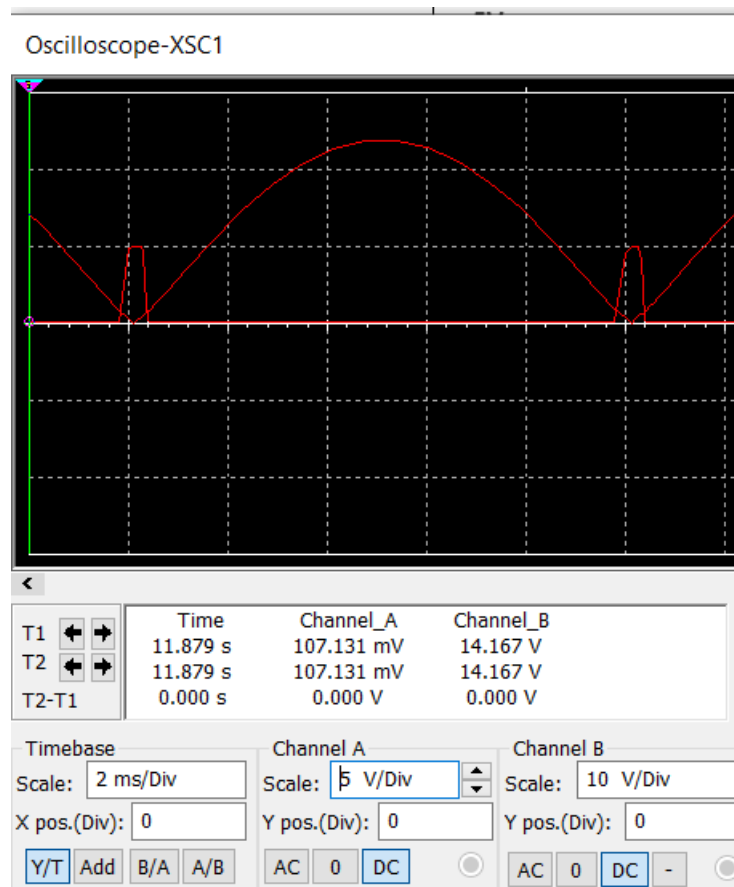
Herefter bestemmes for modstanden.

$$\frac{25,456V - 1,65V}{60mA} = 396,77\Omega$$

For modstanden bliver altså 400 Ω .

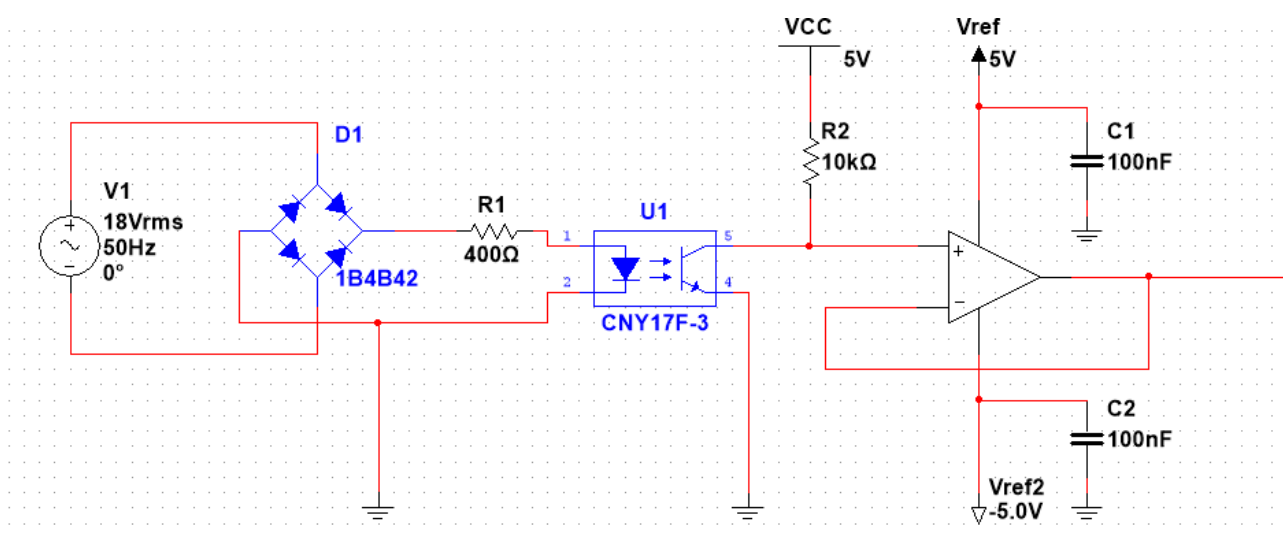
⁶ Ref[6]

På figur 12 ses der hvordan zero-crossing signalet ville komme til at se ud, her peak'er det på 5 V, hvilket er perfekt til at mikrokontrolleren kan aflæse et højt signal når der er zero-crossing.



Figur 12: Zero-crossing og sinus-kurve

Til sidst sættes en operations forstærker på for at sikre at mikrokontrollerens impedans ikke har nogen indflydelse på kredsløbet samt at der sikres at kaskadereglen herved overholdes, se figur 13.

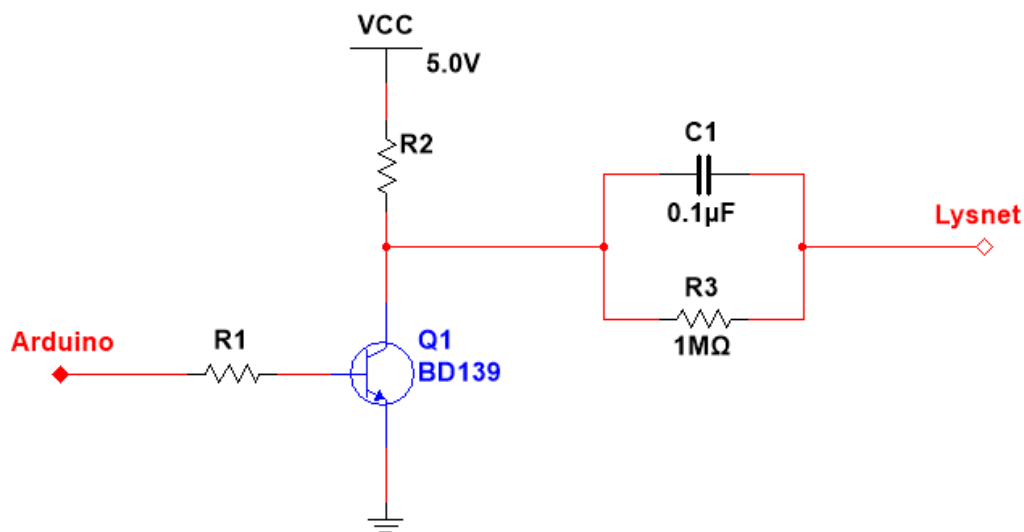


Figur 13: Færdige zero-crossing modul

HW3 Carrier Generator

Carrier Generator, har i systemet til formål at assistere med udsendelse af 120 kHz på lysnettet. Disse pulser eller 'bursts' kan blive detekteret af Carrier Detector, hvor dens output bliver et 0-5V signal, dermed kan dette senere læses som et logisk højt eller lavt. På denne måde er det muligt at sende konkret data over lysnettet, da man gennem softwaren selv kan bestemme på hvilke zero-crosses disse bursts skal udsendes. Selve burstet bliver produceret af Arduinoen, men det er nødvendigt med et elektrisk kredsløb for at sende det ud på lysnettet. Til dette kredsløb benyttes en transistor samt et afkoblingsled bestående af en resistor og en kondensator.

Kredsløbet er opbygget med inspiration fra applikationsnoten⁷ og kan ses på figur 14.



Figur 14: Carrier generator

120kHz firkantsignalet som bliver produceret af Arduinoen sættes til base-benet på transistoren. Antager man at transistoren er i mætning, vil transistoren 'åbne sig' når der bliver sendt 120kHz ind. Signalet vil passere afkoblingsleddet og derefter ud på elnettet. Tilkoblet collector-benet er det fornævnte afkoblingsled, dette har til formål at muliggøre en kobling mellem svagstrømskredsløbet og lysnettet, mere om dette følger senere i afsnittet. Overordnet kan man kort kigge på kondensatorens egen impedans. Som bekendt er en kondensators impedans blandt andet bestemt af frekvensen:

$$Z_C = \frac{1}{2 \cdot \pi \cdot f \cdot C}$$

Dette betyder altså ved en lav frekvens vil afkoblingsledet opfattes som en høj impedans, hvorimod ved høj frekvens vil det ses som lav impedans. Modstanden i afkoblingsleddet vil i kredsløbet give kondensatoren en mulighed for at aflade af sikkerhedsmæssige grunde, der kigges nærmere på kondensatoren senere i afsnittet.

$$C := 0.1 \mu F$$

$$f_1 := 50 \text{ Hz}$$

$$f_2 := 120 \cdot 10^3 \text{ Hz}$$

$$Z_{C_{50}} := \frac{1}{2 \cdot f_1 \cdot \pi \cdot C} = 31830.989 \Omega$$

$$Z_{C_{120}} := \frac{1}{2 \cdot f_2 \cdot \pi \cdot C} = 13.263 \Omega$$

⁷ Ref[7]

På transistorens collector-ben sidder en pull-up resistor, R2, som har til formål at forhindre en kortslutning til emitter-benets stel når transistoren åbner sig, her er valgt 50Ω med inspiration fra applikationsnoten.

Formodstanden, R1, til base-benet skal udregnes således at transistoren er i mætning, lykkedes dette ikke, vil transistoren ikke 'åbne sig' når der bliver udsendt 120kHz fra Arduinoen.

Arduinoens pulser er på 5V og spændingsfaldet på BD139 i mætning er $V_{CE_{SAT}} = 0.5V$ som kan ses ud fra databladet⁸. Det er nu muligt at udregne collectorstrømmen.

$$V_{in} := 5 \text{ V} \quad V_{ce_{sat}} := 0.5 \text{ V} \quad R2 := 50 \Omega$$

$$I_c := \frac{V_{in} - V_{ce_{sat}}}{R2} \rightarrow \frac{5 \cdot V - 0.5 \cdot V}{50 \cdot \Omega} = 90 \text{ mA}$$

Efterfølgende er det muligt at udregne basestrømmen således den ønskede funktionalitet af transistoren opnås. I databladet oplyses det at en collector-strøm på 150mA er en forstærkningsfaktor, H_{FE} , på 40. Da der er udregnet en collector-strøm til 90mA sættes forstærkningsfaktoren til 20, da transistoren helst skal være i 'Deep Saturation' så transistoren kan fungere som switch, vælges en mindre forstærkningsfaktor i udregning risikeres at transistoren ikke er i mætning. Basestrømmen findes på følgende måde:

$$H_{FE} := 20 \quad I_B := \frac{I_c}{H_{FE}} \rightarrow \frac{9 \cdot \text{mA}}{2} = 4.5 \text{ mA}$$

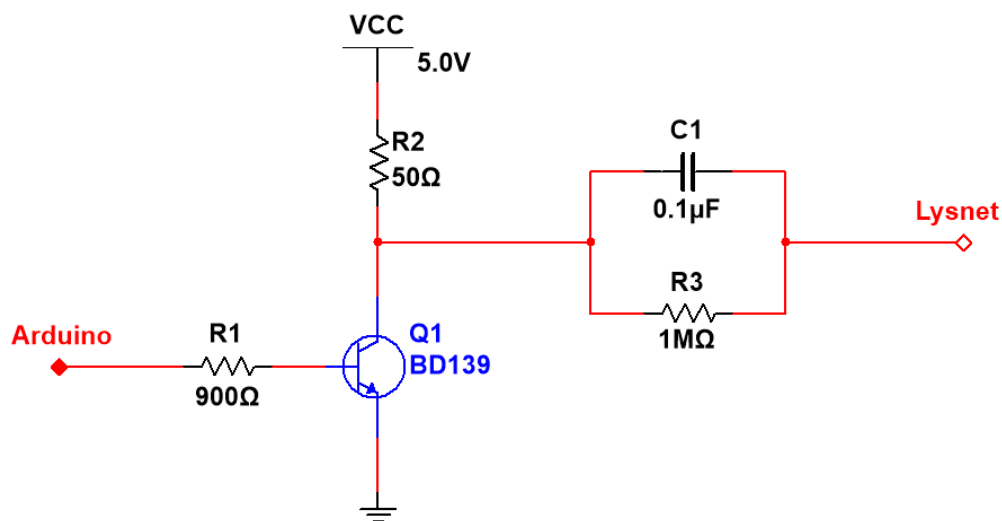
Det aflæses i databladet at spændingen mellem base- og emitter-benet, V_{BE} , er 1V. Det er nu muligt at udregne formodstanden til base-benet på transistoren.

$$V_{BE} := 1 \text{ V} \quad R_B := \frac{V_{in} - V_{BE}}{I_B} = 888.889 \Omega$$

Der udregnes en modstandsværdi på ca. 888Ω. Ud fra Standard Decade Value Table⁹ vælges en resistor på størrelsen 900Ω med 5% tolerance, hvor dette i praktisk muligt ville være resistorer i serie. Kredsløbet er nu blevet opdateret og kan ses på figur 15.

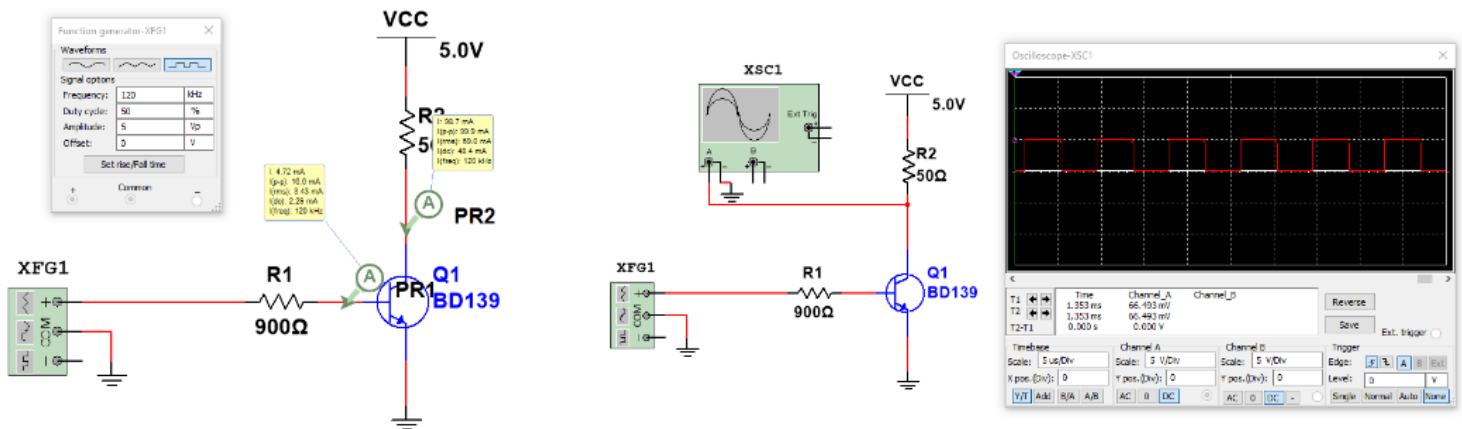
⁸ Ref[8] BD139

⁹ Ref[9] Standard decade table



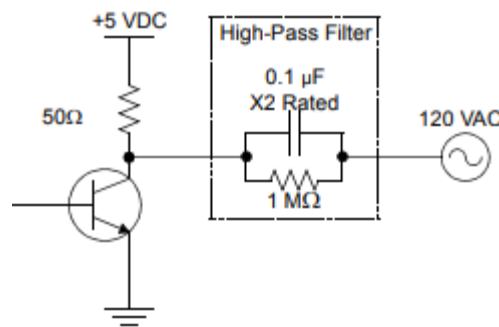
Figur 15: Carrier generator med komponent værdier

Kredsløbet testes nu i Multisim. Der fås en $I_C = 98.7\text{mA}$ og $I_B = 4.72\text{mA}$. Dette passer nogenlunde med udregningerne, den lille variation kan skyldes at $V_{CE(SAT)}$ muligvis ikke er helt den samme i Multisim. Det vises også på figur 16 at firkantsignalet passerer og dermed opnås den ønskede funktionalitet som transistor som en switch.



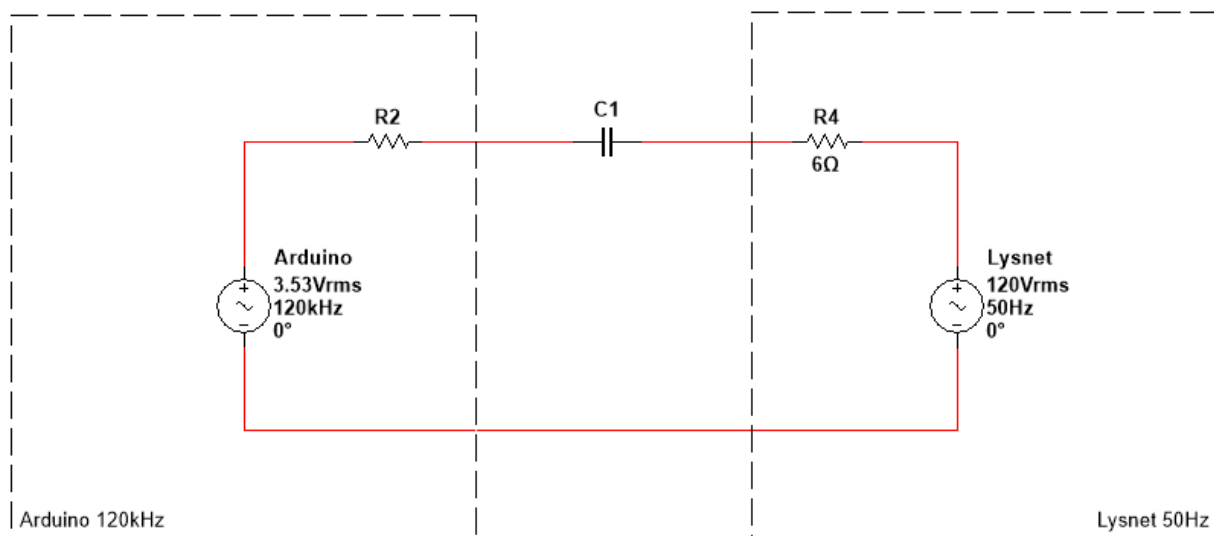
Figur 16: Påvisning af transistor som switch

Der kigges nu nærmere på afkoblingsleddet, nærmere bestemt kondensatoren. I applikationsnoten er dette tilkendegivet som et højpasfilter, se figur 17. Der er altså to kilder der påvirker kredsløbet, henholdsvis lysnettet på 18V AC 50Hz og Arduinoens 120kHz 0-5V firkant signal.



Figur 17: Applikationsnotens eksempel på Carrier Generator

Det er muligt at opstille Thevenin-ækvivalenter for kredsløbet, hvor kredsløbet kan analyseres vha. superposition. Der kan så opstilles overføringsfunktioner som standardfunktioner hvorefter der kan skabes bodeplots, som kan fortælle påvirkning mellem de to kredsløb, og om det er forsvarligt at koble svagstrømskredsløbet direkte på lysnettet. På figur 18 ses kredsløbet opdelt i en 'lysnet'-side og 'Arduino 120kHz'-side. Bemærk særligt modstand R4, som er transformereens impedans. Denne transformer skulle have været brugt i en eventuel realisering af kredsløb, for at transformere 230V AC til 18V AC, da dette ikke er muligt grundet omstændighederne, regnes det alligevel med i følgende. Impedansen sættes til 6Ω. Følgende afsnit med udregninger tager udgangspunkt i applikationsnoten¹⁰ samt udgivet teknisk materiale¹¹ til projektarbejde.

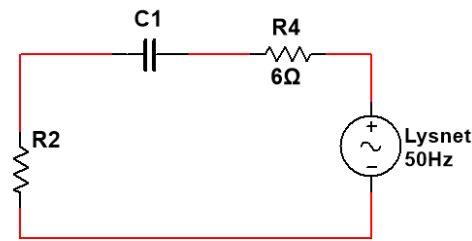


Figur 18: Thevenin illustration

Der undersøges hvordan lysnettet påvirker elektronikken. Ved superposition slukkes Arduino-kilden og der fås nedenstående, figur 19 som kredsløb.

¹⁰ Ref[7]

¹¹ Ref[10] Tekniskmateriale



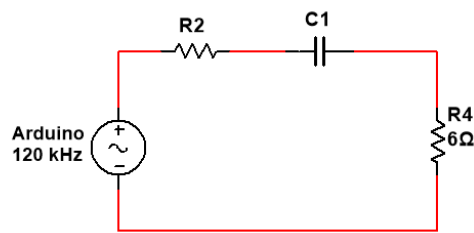
Lysnet påvirkning på elektronikken

Figur 19: Superposition når Arduino er slukket

Ud fra dette fås overføringsfunktionen:

$$T_{V1}(s) = \frac{V_{out}}{V_{in}} = \frac{1}{(R2 + R4) \cdot C} \cdot \frac{s}{s + \frac{1}{(R2 + R4) \cdot C}} \cdot \frac{1}{R2 \cdot C}$$

Dernæst undersøges hvordan Arduino 120kHz påvirker lysnettet. Ved superposition slukkes lysnet-kilden og der fås nedenstående figur 20 som kredsløb.



Arduino påvirkning på net

Figur 20: Superposition når lysnet er slukket

Ud fra dette fås overføringsfunktionen:

$$T_{V2}(s) = \frac{V_{out}}{V_{in}} = \frac{1}{(R2 + R4) \cdot C} \cdot \frac{s}{s + \frac{1}{(R2 + R4) \cdot C}} \cdot \frac{1}{R4 \cdot C}$$

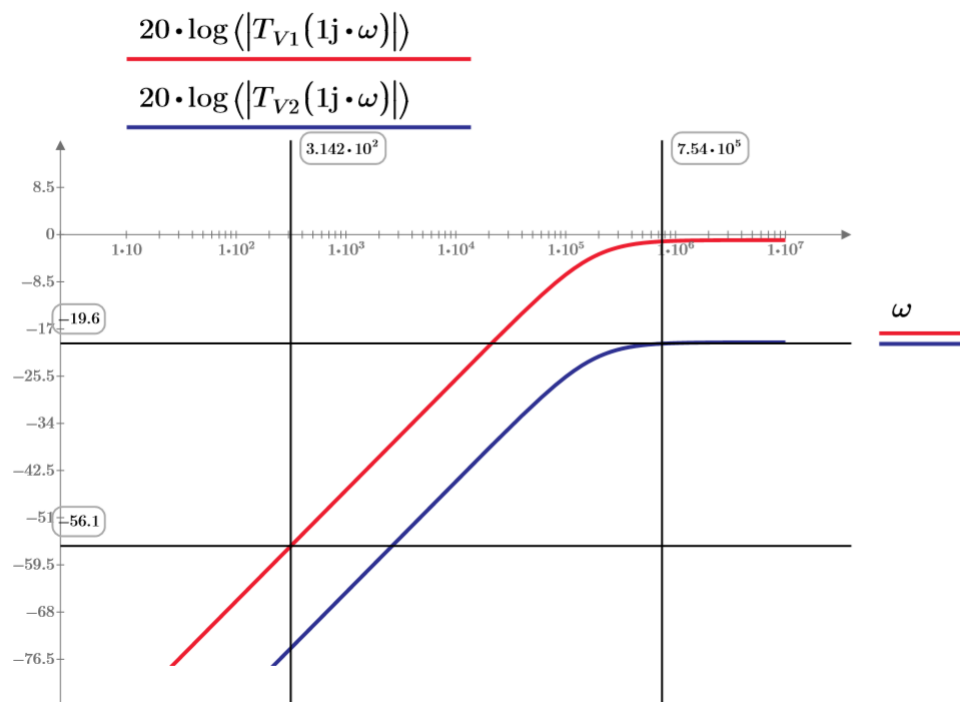
Det ses at de to overføringsfunktioner er bestående af to standardled og der kan herved laves en frekvensanalyse.

$$\frac{\alpha_1}{s + \alpha_1} \cdot \frac{s}{K_1} \quad \text{og} \quad \frac{\alpha_1}{s + \alpha_1} \cdot \frac{s}{K_2}$$

Ud fra applikationsnoten¹² vælges der at R2 sættes til 50Ω, også grundet at denne værdi er brugt tidligere til udregning af modstande til transistor. R4 er valgt til 6Ω som er et estimat på transformerens

¹² Ref[7]

impedans. Kondensatoren, C1, bliver valgt til 0.1μF også efter applikationsnoten¹³. Ud fra disse komponentværdier har det været muligt at udforme et bodeplot, se figur 21.



Figur 21: Bodeplot af filter

Hvor de vertikale cursors er frekvensen i rad/s og horisontale cursors er dæmpning i dB, begge udregninger vises på nedenstående.

$$120000 \text{ Hz} \cdot 2 \cdot \pi = 753982.237 \frac{\text{rad}}{\text{s}}$$

$$50 \text{ Hz} \cdot 2 \cdot \pi = 314.159 \frac{\text{rad}}{\text{s}}$$

$$20 \cdot \log(|T_{V1}(1j \cdot 50 \cdot 2 \cdot \pi)|) = -56.078$$

$$20 \cdot \log(|T_{V2}(1j \cdot 120000 \cdot 2 \cdot \pi)|) = -19.638$$

Der ses altså en dæmpning på -19.5 dB på 120kHz signalet med de valgte komponentværdier. Hvorimod der er en stor dæmpning på -56.1dB på 50Hz signalet. Kigger man dertil på en formodet udgangs amplitude, findes det at en dæmpning på -19.6 dB giver en gg på omtrent 0.101 og en dæmpning på -56.1 dB giver en gg på omtrent 0.001.

¹³ Ref[7]

$$gg_{50} := 0.001571$$

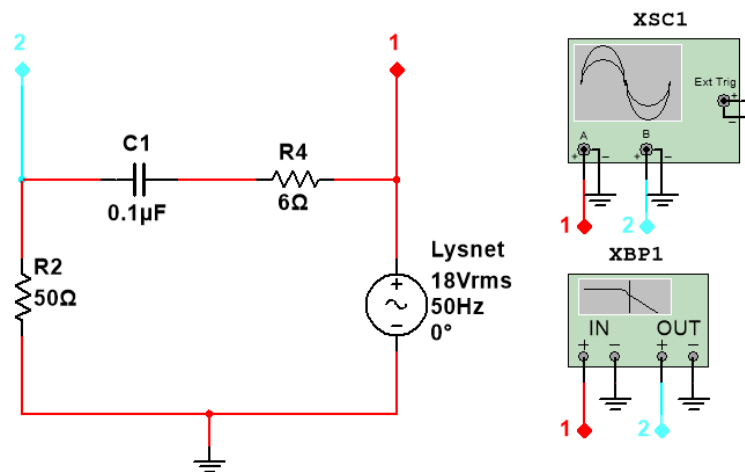
$$V_{18Vrms} := 18 \text{ V} \cdot \sqrt{2} = 25.456 \text{ V}$$

$$V_{out_A_120kHz} := (V_{18Vrms} - 0.5 \text{ V}) \cdot gg_{50} = 39.206 \text{ mV}$$

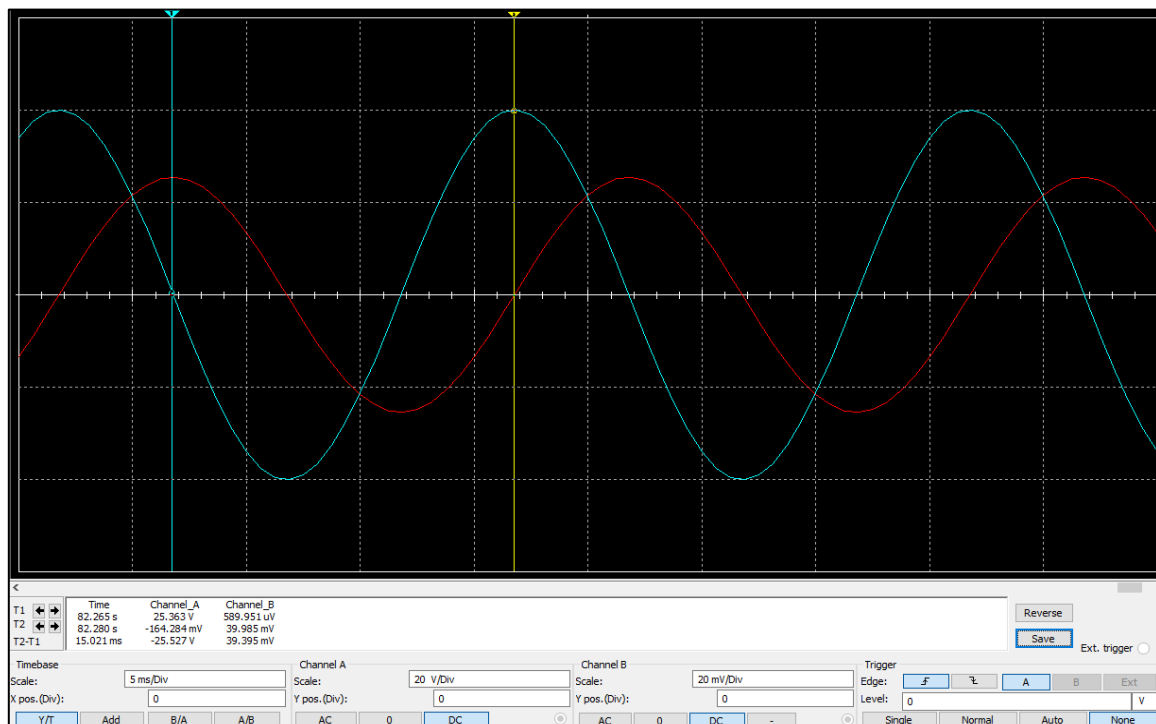
$$gg_{120} := 0.104256$$

$$V_{out_A_120kHz} := (5 \text{ V} - 0.5 \text{ V}) \cdot gg_{120} = 469.152 \text{ mV}$$

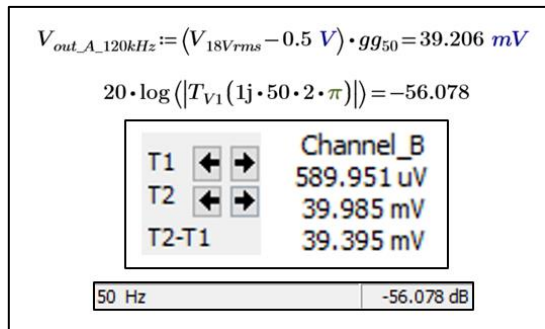
Kredsløbene testes nu i Multisim. Figur 22, 23, 24 og 25 viser testen for lysnet-delen. Bodeplotteren viser fornuftigt den udregnede værdi på -56 dB dæmpning ved 50 Hz. Der ses også en udgangsamplitude på ca. 39.4mV som også passer ganske fint med det udregnede.



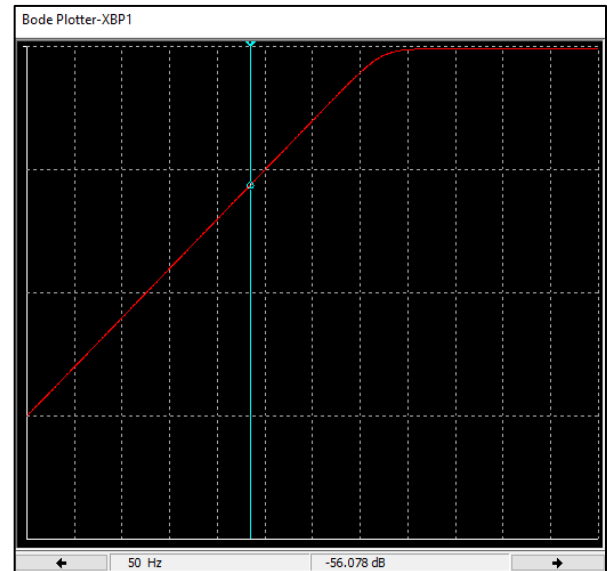
Figur 23: Opstilling af test for lysnet-del



Figur 22: Scope af test



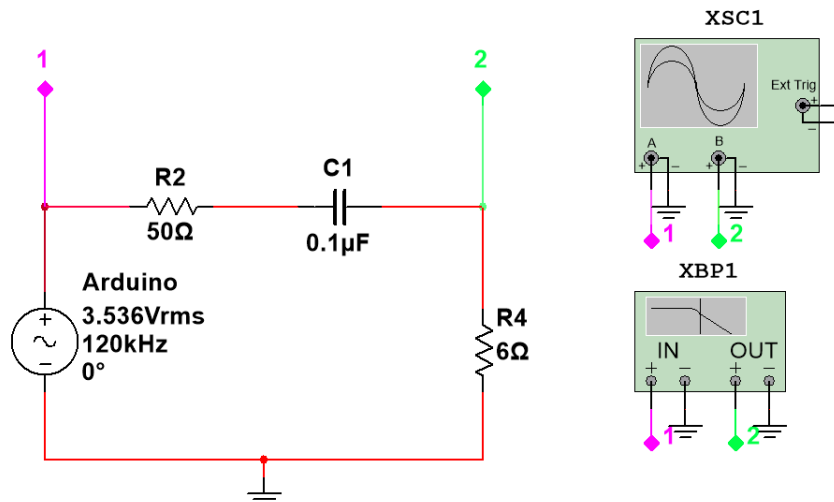
Figur 24: Udregninger samt resultater



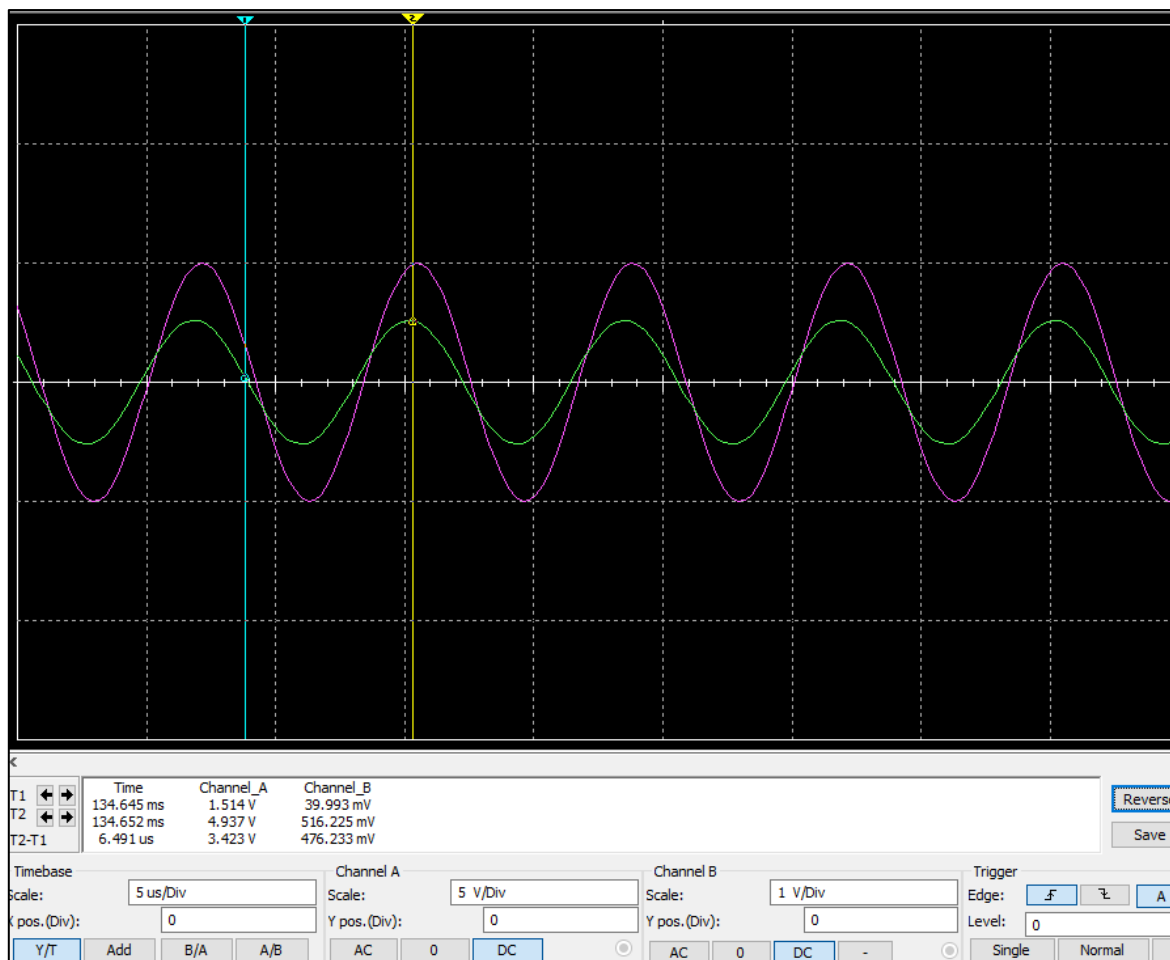
Figur 25: Bode plotter ved 50 Hz

Konkluderende kan det siges at lysnet-delen bliver dæmpet en del og er ikke i nærheden af pasbåndet, men må derimod siges at være i stopbåndet. Dæmpningen resulterer i en udgangsamplitude på ca. 40mV på svagstrømskredsløbet og der regnes ikke med at det ville resultere i nogen komplikationer.

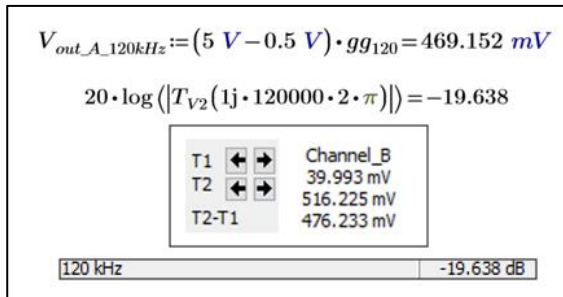
Nedenstående viser testen for Arduino-delen, se figur 26, 27, 28 og 29. Bodeplotteren viser fornuftigt den udregnede på -19 dB dæmpning ved 120 kHz. Der ses også en udgangsamplitude på ca. 476 mV som også passer udmærket med det udregnede.



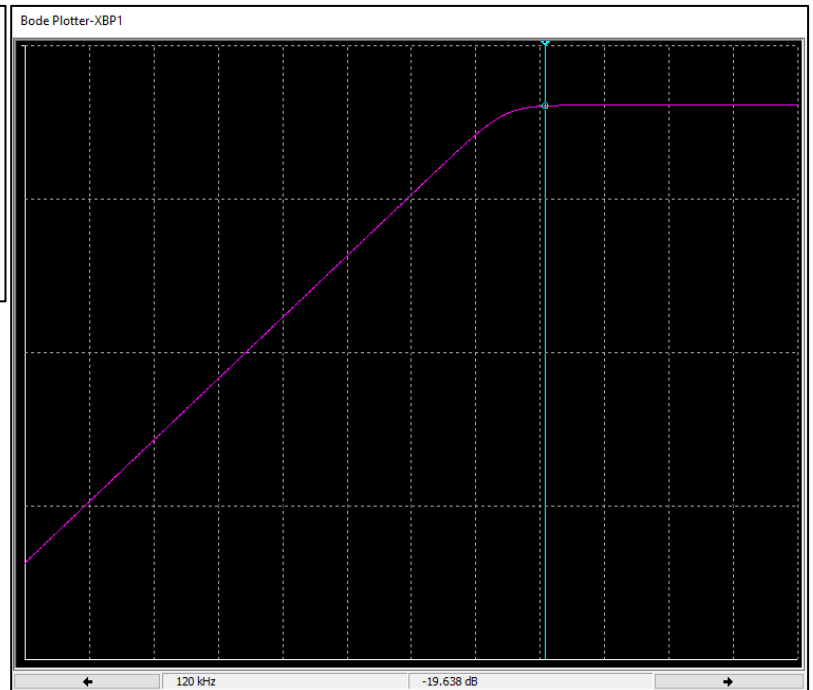
Figur 26: Opstilling af test for Arduino del



Figur 27: Scope for test Arduino del



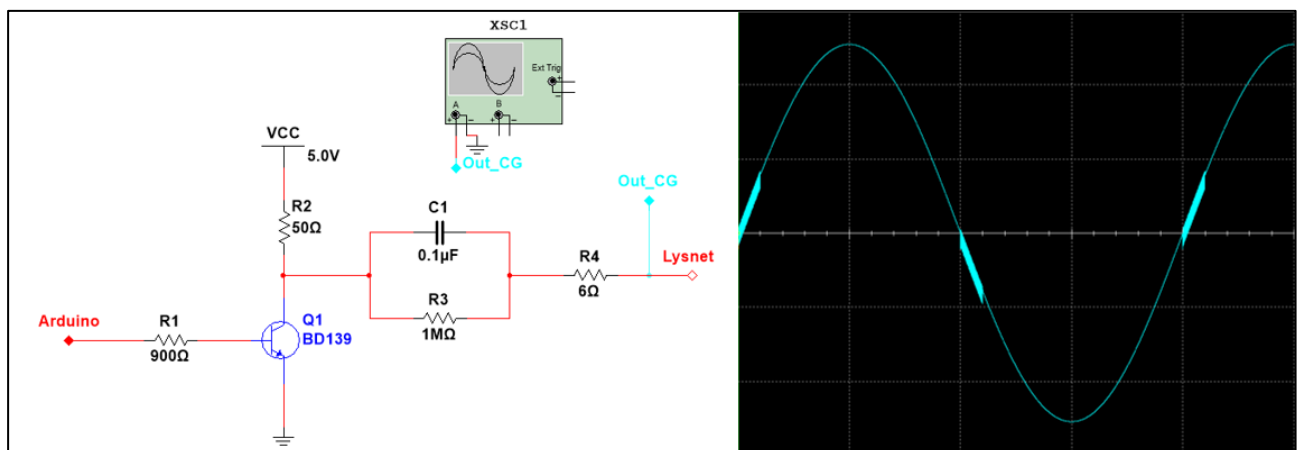
Figur 29: Udregninger samt resultater



Figur 28: Bodeplotter ved 120 kHz

Konkluderende kan det siges at der er en lille dæmpning på Arduino-120kHz, dog vil det siges at kunne accepteres da det stadig passerer og kan forstærkes hvis nødvendigt. Som også set på bodeplotteren fra Multisim, må det siges at Arduino-120kHz er ved, eller næsten i, pasbåndet og undgår stopbåndet. Samlet set kan det siges at koblingen mellem lysnet og et svagstrømskredsløb er muligt, selvom det ikke er normalt procedure.

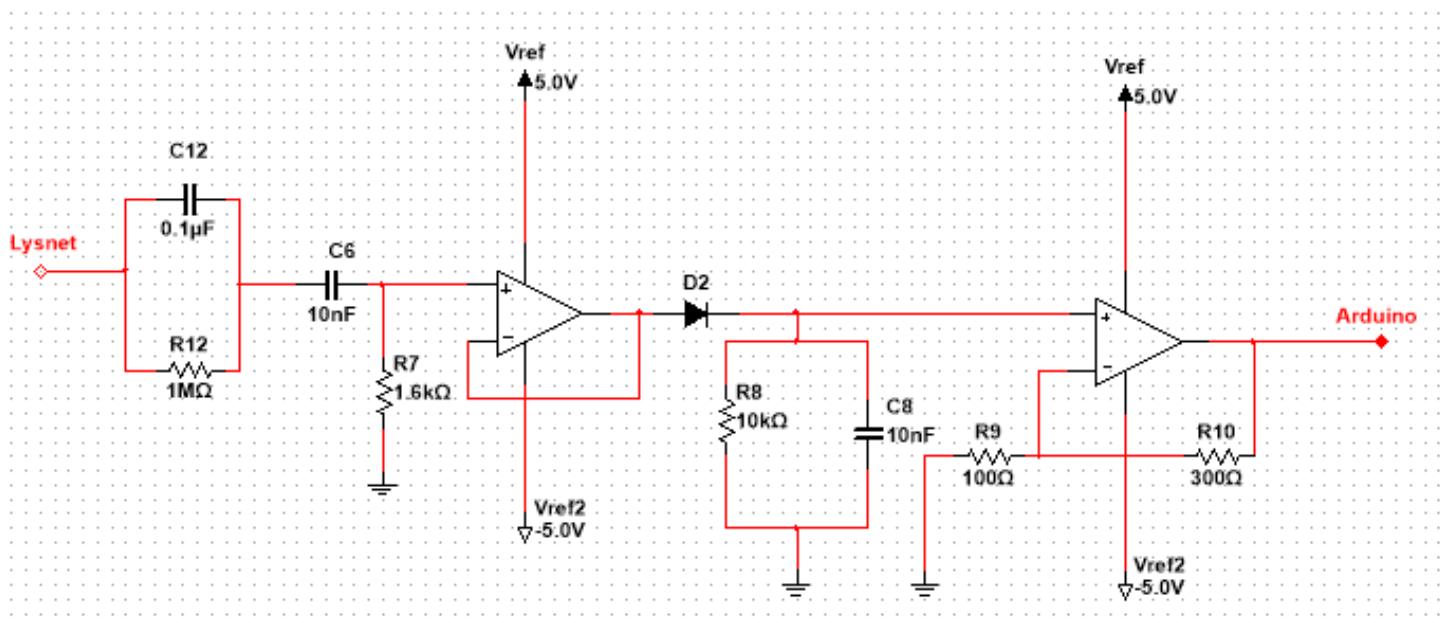
Den endelige funktionalitet af Carrier Generator ses på figur 30.



Figur 30: Endelige funktionalitet af Carrier Generator

HW4 Carrier Detektor

Carrier Detektoren er blevet bygget på baggrund af aplikationsnoten¹⁴. Carrier Detektor modulet (se figur 31) bliver lavet vha. en række forskellige filtre og operations-forstærkere. Det første filter der ses i kredsløbet, er magen til det der er blevet lavet i Carrier generator kredsløbet. Som der også er forklaret i Carrier Generator afsnittet, er dette sat på for at beskytte svagstrømskredsløbet når det kobles direkte på lysnettet. Dernæst kommer der et aktivt-højpasfilter, hvori der er blevet lavet nogen undersøgelser¹⁵, magen til det der er blevet skrevet i Simulink afsnittet. Den sidste del af modtageren laves vha. en Envelope Detektor, som er blevet undersøgt online¹⁶, herefter sættes en operations forstærker på til at forstærke signalet til den rigtige spænding, 5 V, denne operations forstærker har den egenskab at den 'klipper' envelope signalet. Herved fås et pænere 5 V signal, der langsomt aflades, i stedet for et ujævnt signal, der potentielt ikke nåede en høj nok udgangs spænding. Operationsforstærkeren efter høj pas-filteret sørger for at filtret og envelope-detektoren ikke har indflydelse på hinandens impedanser, og derved ikke forstyrrer input signalet på de 120 kHz.



Figur 31: Carrier detektor

Høj-pas filteret bliver designet således at der er en knæk-frekvens på 10 kHz.

$$f_c = \frac{1}{2\pi RC}$$

Frekvensen sættes ind, samt en kondensator med værdien 10 nF.

¹⁴ Ref[7]

¹⁵ Ref[12]

¹⁶ Ref[11]

$$R = \frac{1}{2 \cdot \pi \cdot 10000 \cdot 10 \cdot 10^{-9}} = 1.592 \cdot 10^3$$

Modstanden der skal bruges i høj-pas filteret vil altså være 1.6 kΩ og kondensatoren 10 nF.

Ved at opstille høj-pas filteret kan man finde overføringsfunktionen ved:

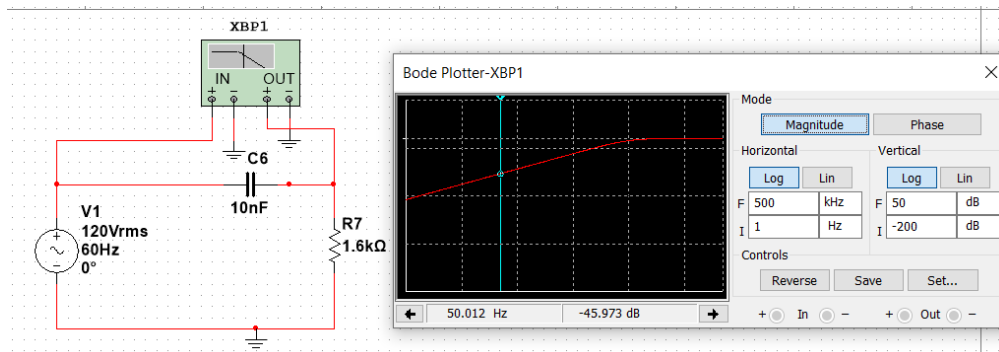
$$G(s) = \frac{R}{\frac{1}{s \cdot C} + R} \Rightarrow \frac{R \cdot C \cdot s}{R \cdot C \cdot s + 1} \Rightarrow \frac{s}{s + \frac{1}{RC}} \Rightarrow \frac{s}{s + \alpha}$$

Hvis R og C indsættes fås overføringsfunktionen:

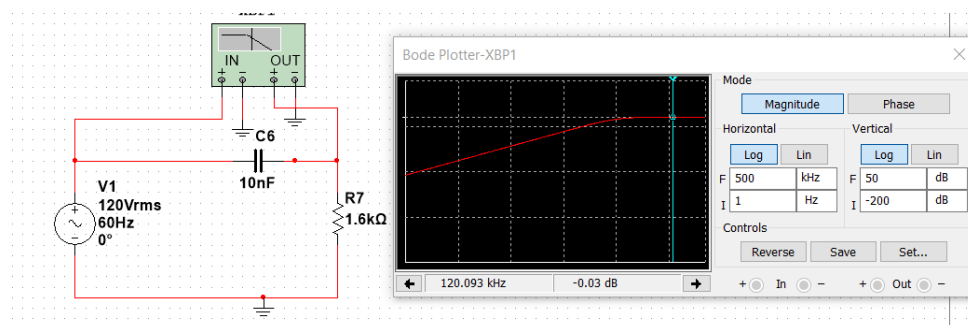
$$G(s) = \frac{1.592 \cdot 10^3}{\frac{1}{s \cdot 10 \cdot 10^{-9}} + 1.592 \cdot 10^3} = 1 - \frac{6281}{s + 6281}$$

Herved vides at bode plottet vil se ud som bodeplottet der blev lavet i Simulink. Her dæmpes der med 46 dB ved 50 Hz. På denne måde kommer 50 Hz signalet ikke igennem, hvorimod 120kHz signalet vil løbe uhindret igennem.

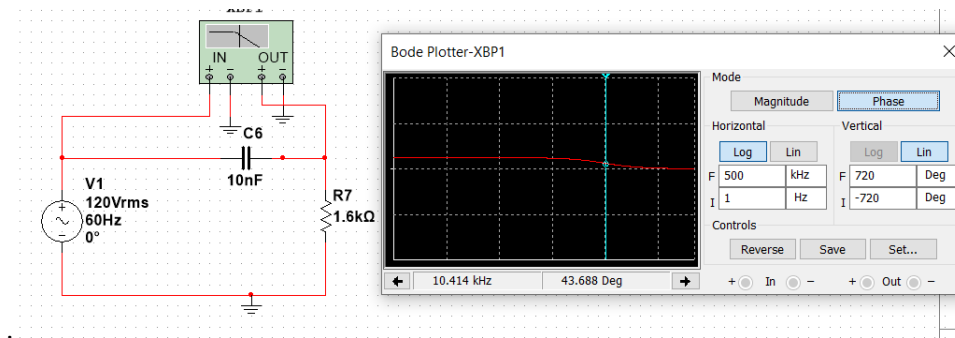
Der er blevet lavet et bodeplot over Mutlisim filteret, hvor der kan ses at det stemmer overens med det forventede bodeplot. Her dæmpes der med 46 dB ved 50 Hz, på figur 32. På figur 33 ses der også at der ikke dæmpes ved 120 kHz og på figur 34, kan der ses at knæk frekvensen ligger i de 10 kHz, som α var blevet beregnet til.



Figur 32: 46 dB dæmpning ved 50 Hz



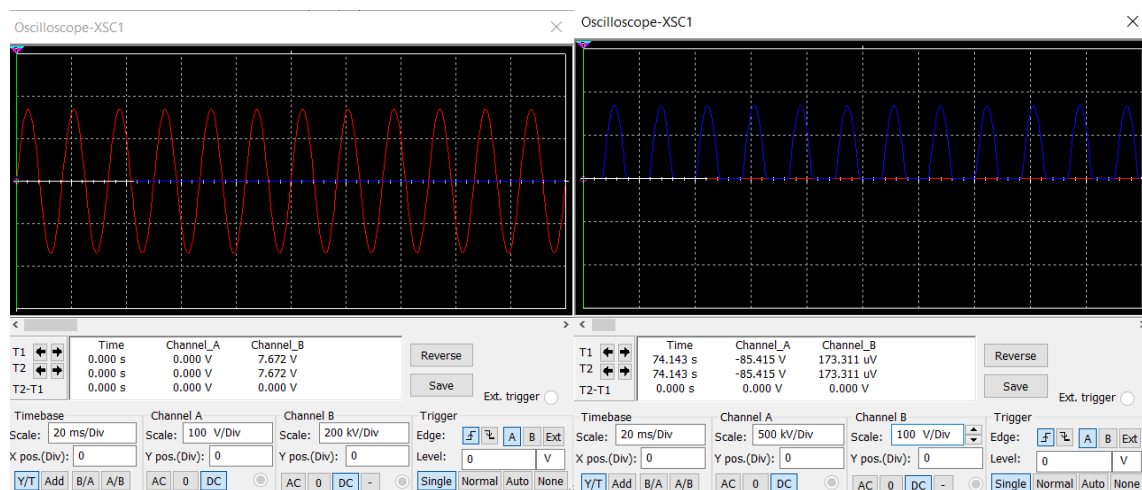
Figur 33: 0 dB dæmpning ved 120 kHz



Figur 34: Knækfrekvens på 10 kHz ved bodeplotter

Efter dioden på figur 31 ses Envelope Detektoren, der består af et parallelt RC-kredsløb. Her er der blevet valgt værdier magen til dem fra applikationsnoten. Envelope-detektoren, lader 10 nF kondensatoren (C8) op, når der kommer et sinus-signal. Når sinus-signalet nærmer sig 0, vil kondensatoren blive afladt, da kondensatoren sørger for at der ikke kommer bratte ændringer i spænding. Pga. dioden, vil sinus-signalet dog ensrette sig.

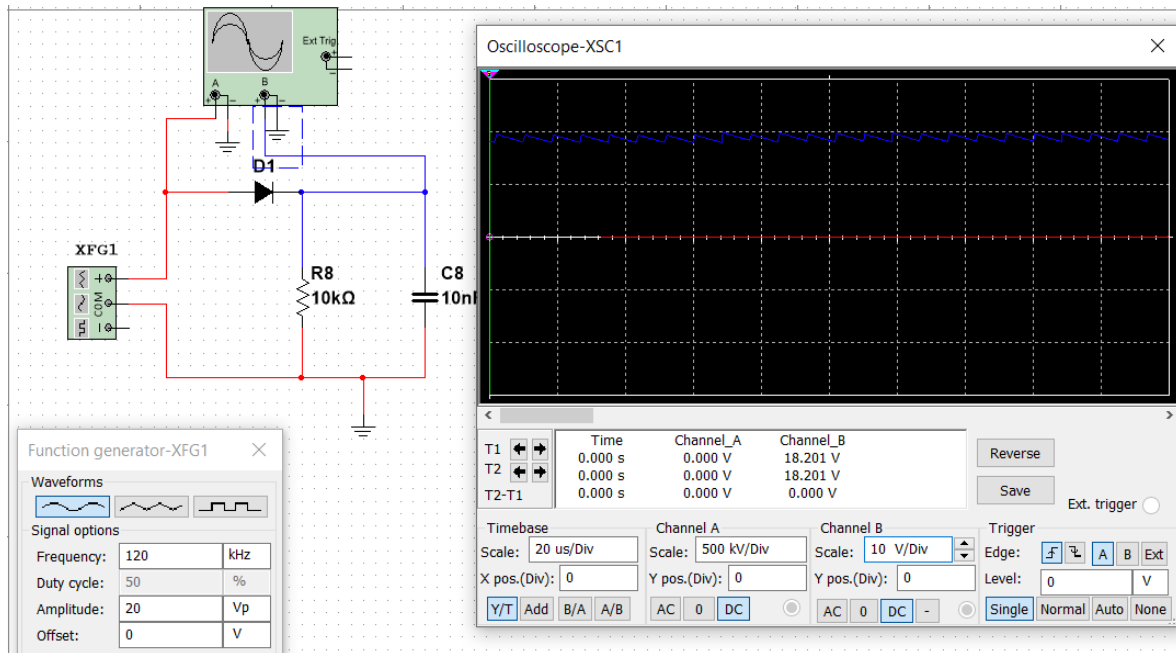
Et normalt sinus-signal ville se ud som på figur 35, men efter at dioden har ensrettet signalet ser det ud som på figur 36. Herved opstår der et dc-offset på signalet. Dette dc-offset vil passere direkte igennem filteret og derved vil der opstå et højt signal. Da dioden ikke virker lineært, er det ikke muligt at udregne et stationært sinus-respons på signalet.



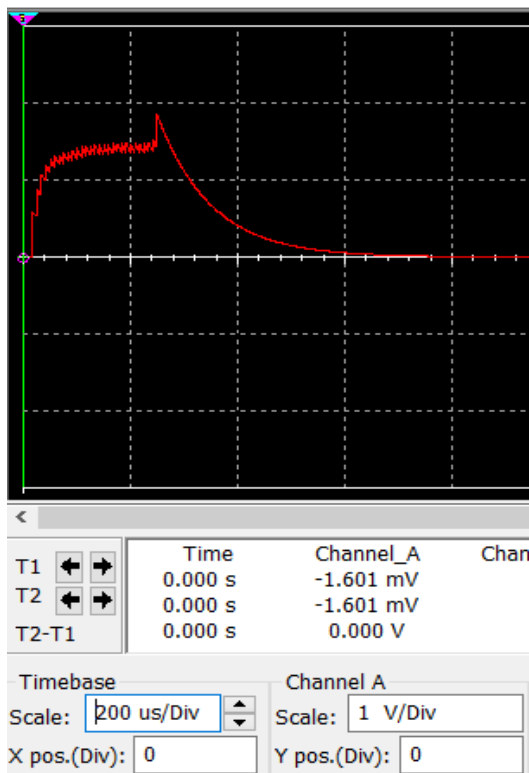
Figur 35: Normal sinusurve

Figur 36: Ensrettet sinusurve

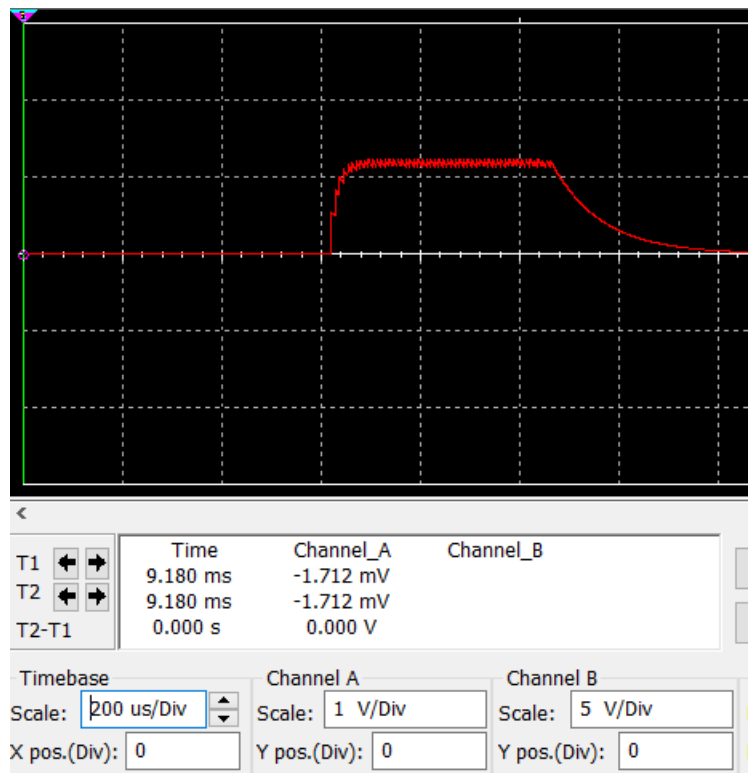
Efter at have passeret igennem både diode og Envelope Detector vil signalet se ud det blå signal på figur 37. Her ses et ujævnt men højt signal.



Figur 37: Signal efter passering af diode-bro og Envelope Detektor



Figur 38: Udgangs signal ved opadgående sinus

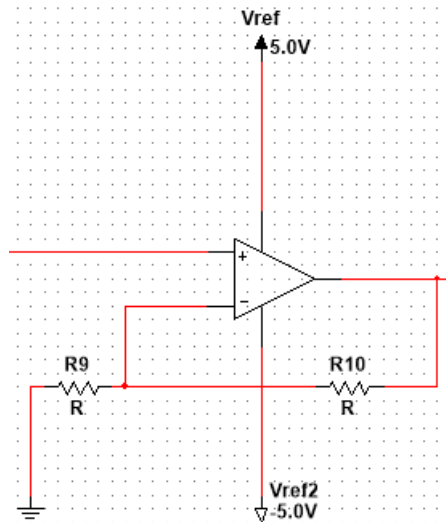


Figur 39: Udgangs signal ved nedadgående sinus

Ved en ændring af spænding vil udgangssignalet se ud som signalet på figur 38 ved en opadgående sinuskurve med 120 kHz bursts, og se ud som på figur 39 ved en nedadgående sinuskurve med 120 kHz

bursts. Disse 2 signaler ønskes forstærket således at der kan aflæses et logisk højt signal, når der er 120 kHz bursts.

For at få et brugbart signal sættes en operations forstærker på kredsløbet, operations forstærkerens opbygning ses på figur 40



Figur 40: Op-amp

Forstærkningen beregnes ved formelen:

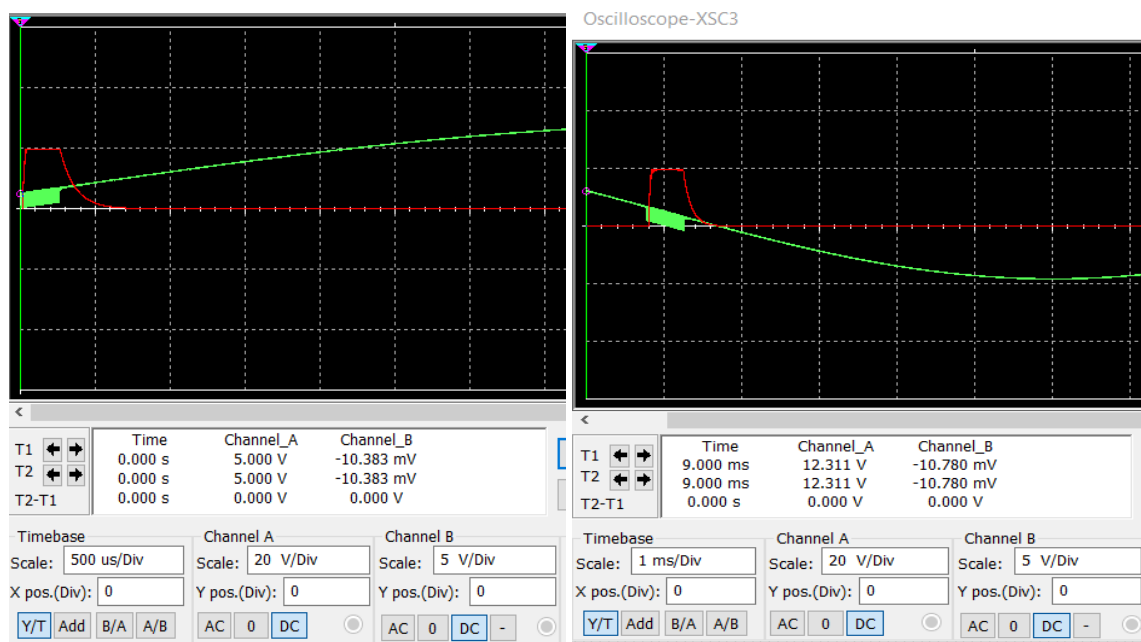
$$K = \frac{z2(s) + z1(s)}{z1(s)}$$

Her vurderes at signalet skal forstærkes med 4, derved fås værdierne

$$K := \frac{300 + 100}{100} = 4$$

R10 bliver altså 300 ohm og R9 bliver 100 ohm.

På figur 41 og 42, kan de tilsvarende forstærkede signaler ses. Der ses en lille afladnings tid på envelope-detektoren, men det er blevet vurderet til ikke at være signifikant for modtager modulet. På figur x, ses at de oprindelige spike er blevet klippet af, dette skyldes at operationsforstærkeren ikke kan sende et signal ud, der er større end de 5 V, på dets sources.



Figur 41: Udgangs signal efter op-amp(opadgående).

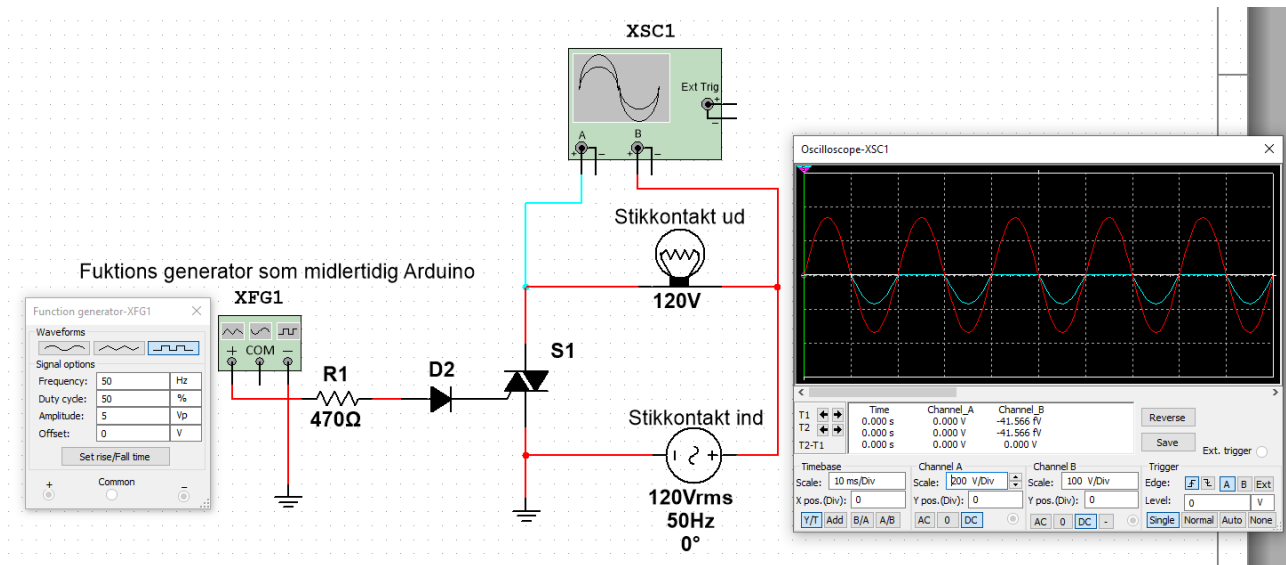
Figur 42: Udgangs signal efter op-amp(nedadgående)

HW5 Switch/dimmer

Switch/dimmer blev bygget på baggrund af applikationsnoten¹⁷. I teorien skulle den fungere på den måde at man sender et signal ind på gaten af en triac, som bestemmer om lampen skal være tændt eller slukket. Når der bliver sendt et signal, skal den være tændt og når der ikke gør, er den slukket. Der kan derfor opnås en dimmer ved at tænde og slukke for signalet ind på triac'ens gate, flere gange i sekundet. Dette skaber en illusion af at lampen lyser svagere, eftersom menneskets øjne ikke er i stand til at opfange, at lampen faktisk blinker.

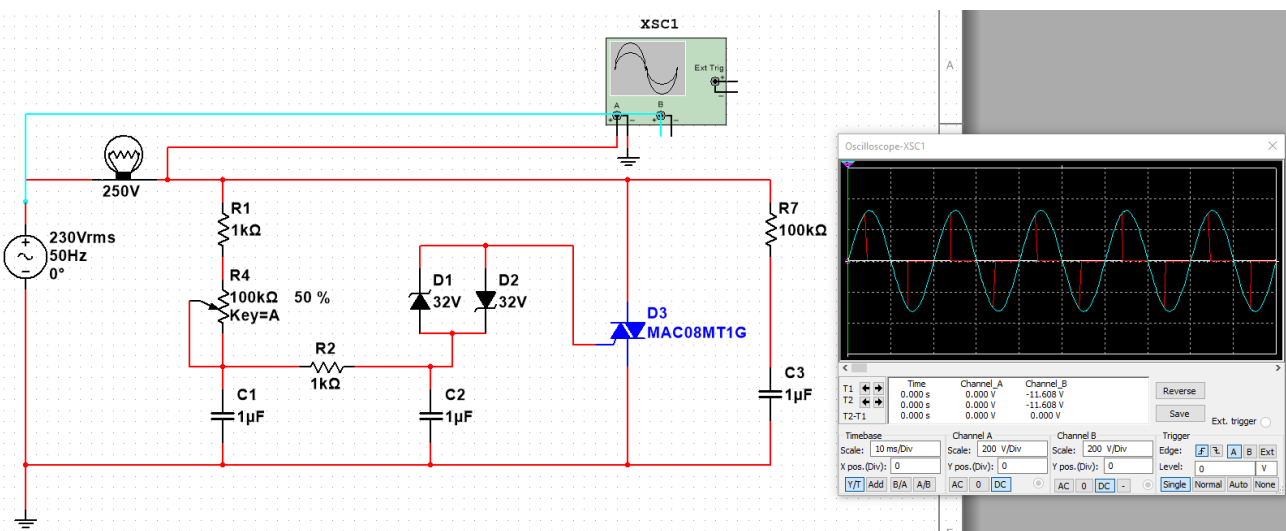
Der blev først lavet en version hvor en Triac blev brugt, men dette skabte nogle problemer eftersom Multisim simulationerne viste, at det ikke ville virke. Udgangssignalet der blev skabt af triac'en var ikke et AC-signal som blev cuttet af ved zero-crossing af PWM-signalet. Figur 43 viser hvordan triac'en opførte sig, den skabte et negativt ac signal, som gik lavt når AC-signalet fra stikkontakten gik lavt. Den var også 0 når AC-signalet fra stikkontakten. Den blå graf er lampen og den røde er AC-signalet fra elnettet.

¹⁷ Ref[7]



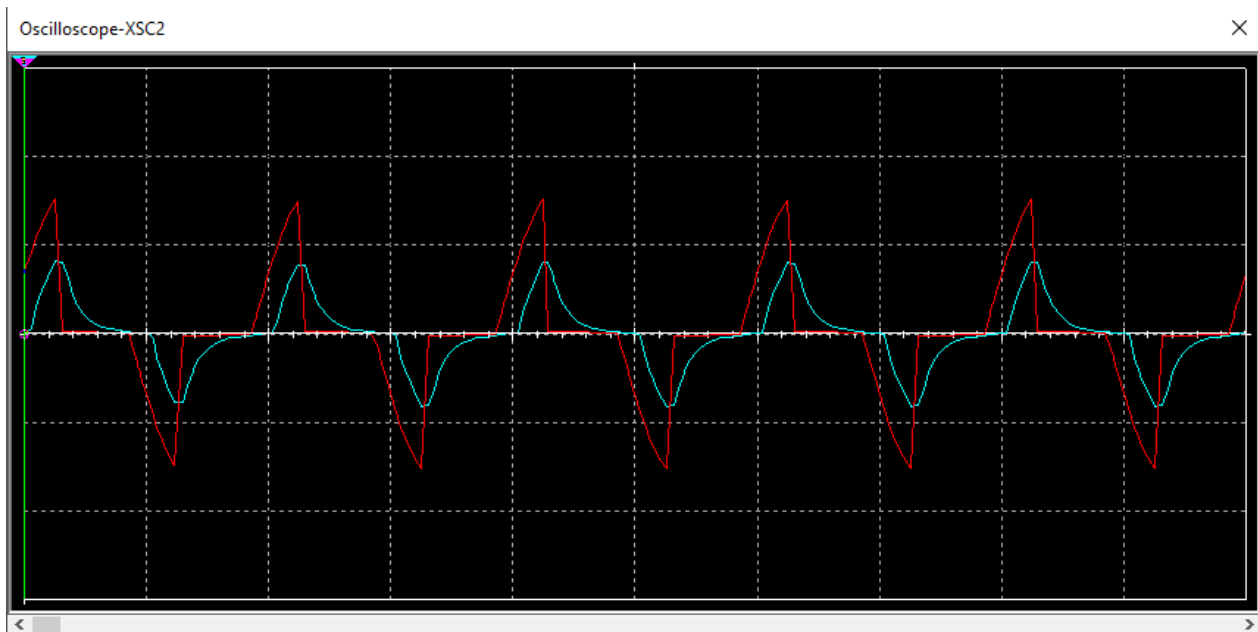
Figur 43: Dimmer med triac

For at se hvordan et signal til en AC-lampe skulle se ud, blev der designet et kredsløb baseret på en moderne standard væg dimmer. En væg dimmer fungerer ligesom en stikkontakt, hvor man kan dreje på en knap for at ændre lysstyrken. Dette opnår man ved at koble et potentiometer på kredsløbet, som ændre den mængde af strøm som triac'en får. Der bliver samtidigt brugt kondensatorer til at bestemme hvornår strømmen til triac'en skal aflades. Dette skaber den cutoff effekt som blev nævnt tidligere, som vist på figur 44.



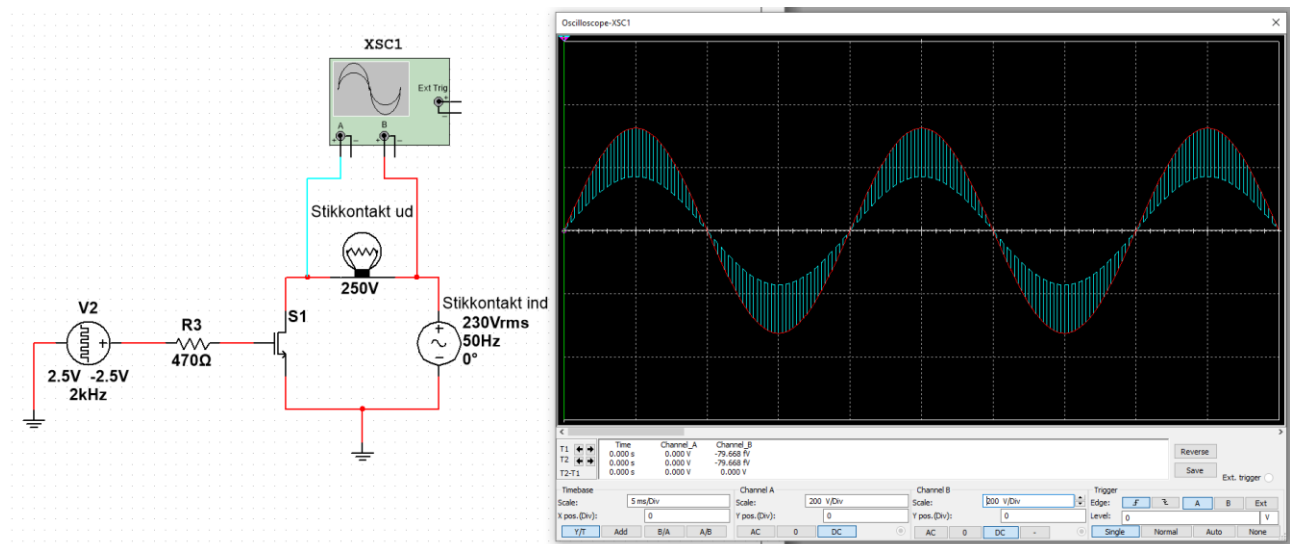
Figur 44: Standard væg dimmer

Problemet ved at gøre det med en Arduino er at der bliver sendt et DC-signal i stedet for et AC-signal. Dette gør det betydeligt anderledes at arbejde med, eftersom indgangssignalet på Triac'en er anderledes. Figur 45 viser hvordan indgangssignalet (blå) og udgangssignalet (rød) på Triac'en ser ud. Den viser hvordan indgangssignalet bliver afladt og skaber et cutoff i udgangssignalet.

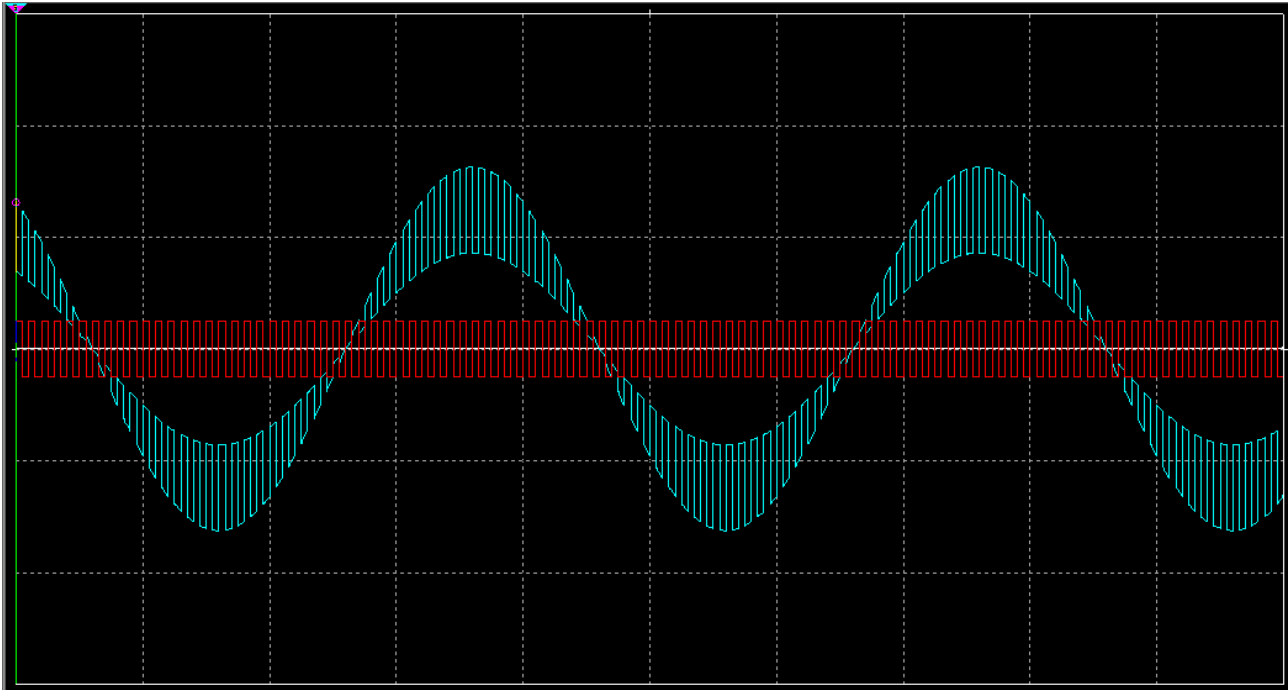


Figur 45 : Standard væg dimmer Triac signaler

Efter en samtale med vejlederen blev der lavet en ny version hvor triac'en er blevet erstattet af en transistor. Grunden til at den bliver skiftet er at en transistor ændre signalet fra Arduinoen, som så ændre signalet der kommer ud af lampen. Dette kan ses tydeligere på figur 46 hvor det blå signal er udgang signalet fra lampen. Den viser hvordan transistoren tænder og slukker alt efter om PWM-signalet er højt eller lavt, det er inverteret hvilket betyder at lampen er tændt, når PWM-signalet er lavt. Denne sammenhæng er tydeligere vist på figur 47, hvor man kan se PWM-signalet i sammenhæng med udgangssignalet. Der er blevet valgt at sende 2 kHz PWM-signal ud af Arduinoen, eftersom det gav en mere præcis kurve. Denne mængde af Hz kan ændre efter behov.



Figur 46 : Switch/Dimmer design 3



Figur 47 : Switch/Dimmer design 3 PWM- og lampe signal

Resistoren på 470 Ohm er overflødig i simuleringen af kredsløbet, eftersom transistoren er virtuel og kan tåle alle mængder af ampere. Men hvis kredsløbet skulle bygges i virkeligheden, ville der skulle være en resistor på f.eks. 570 Ohm til at skalere ampere ned til hvad transistoren kan tåle. Der kan eventuelt også tilføjes en diode for at ensrette strømmen og røge sikkerheden mod kortslutninger, eftersom der arbejdes med en høj- og lav- spændings kredsløb. Ved at tilføje en resistor på 470 Ohm bliver ampere lavet om til en mængde som transistoren kan håndtere:

$$\frac{5 \text{ V}}{470 \text{ } \Omega} = 10.638 \text{ mA}$$

HW6 Tolerance og Produktion

I projektarbejdet med hardwaren er tolerancen på komponenter anset som værende en vigtig faktor til en eventuel fremtidig produktion. Komponenterne som havde været tilgængelige i ASE-lageret er relativt billige komponenter og har en relativt stor tolerance. Modstandene har en tolerance på 5% og kondensatorerne en modstand på 10%. For at sikre at produktet virker på trods af tolerancer er der blevet lavet nogle enkelte tests på nogle kredsløb for at undersøge funktionaliteten ved deres ydre grænser. Ved at undersøge om kredsløbene stadigvæk virker efter hensigt kan der derved spares penge ved komponenternes kvalitet. Lykkedes det at opbygge et kredsløb med komponenter hvori fornævnte komponenter har en høj tolerance kan produktet fremstilles billigt og der sikres derfor cost-efficiency. Det er aktuelt da komponenter med større tolerancer typisk vil være billigere, så jo højere tolerance på de benyttede komponenter, jo billigere vil en produktion blive.

Nedenstående dokumenterer enkelte test med komponentværdi i deres ydre grænser, alt efter tolerance. Disse enkelte kredsløb simuleres i Multisim med tolerance værdier, for at undersøge om den ønskede funktionalitet stadig er aktuel. På baggrund af at Carrier Detektoren benytter sig af både

kondensatorer og modstande, er komponenterne først blevet skiftet ud for dette modul, se figur 48 til 51.

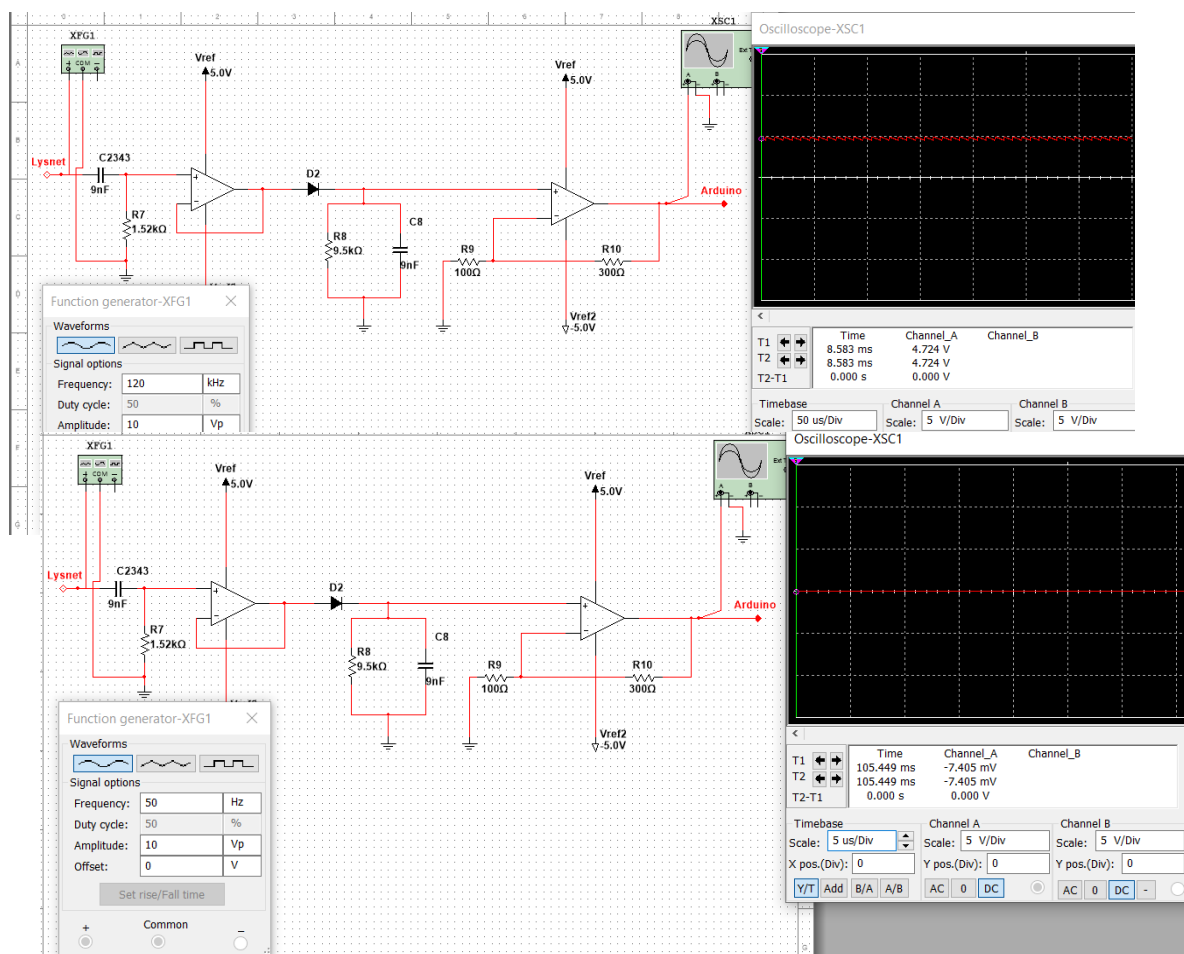
De ydre grænser er blevet regnet ud ved

$$C2343 \quad 10nF \cdot 10\% \pm 10nF = 9nF \text{ og } 11nF$$

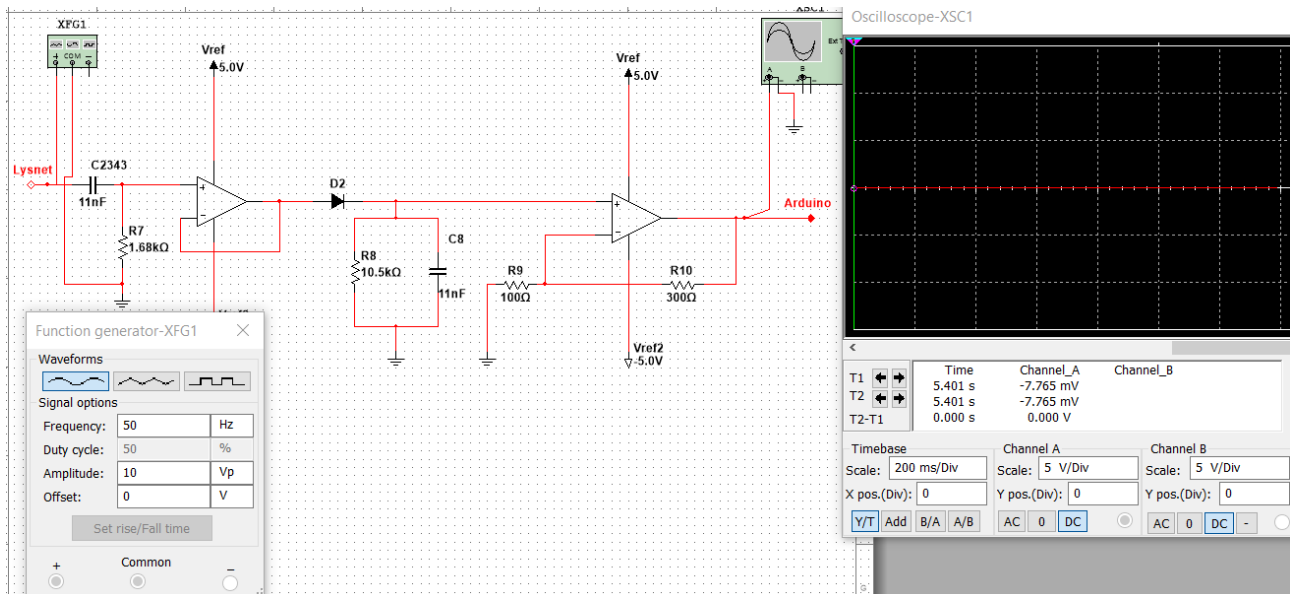
$$R7 \quad 1.6k\Omega \cdot 5\% \pm 1.6k\Omega = 1.52k\Omega \text{ og } 1.68k\Omega$$

$$R8 \quad 10k\Omega \cdot 5\% \pm 10k\Omega = 9.5k\Omega \text{ og } 10.5k\Omega$$

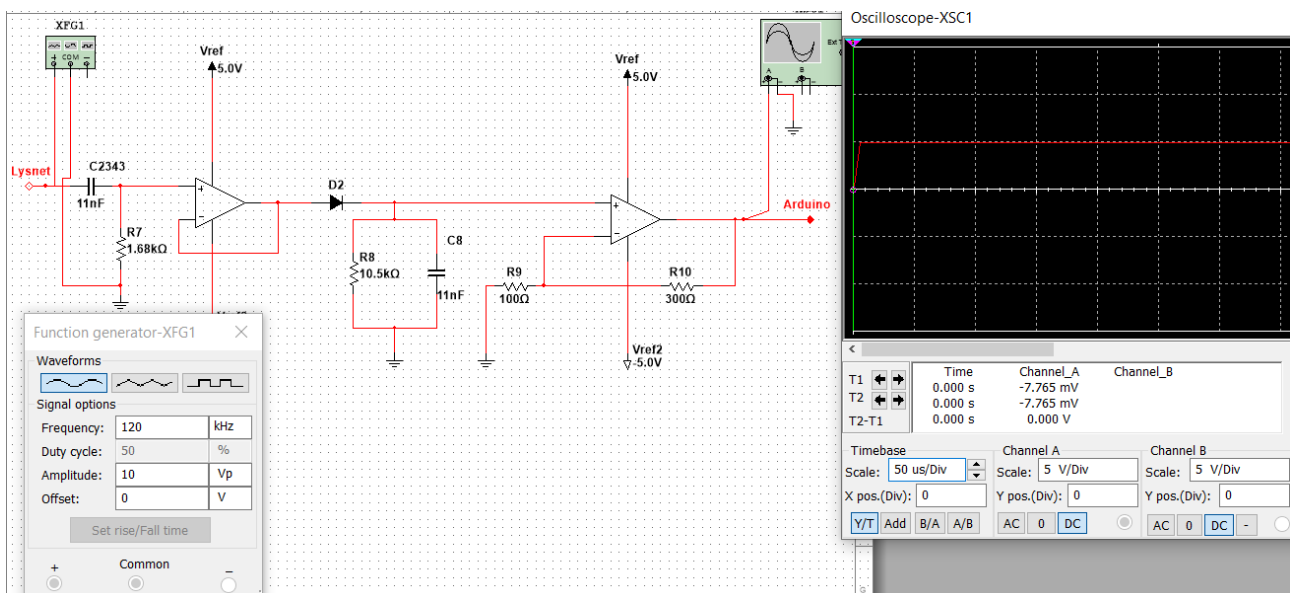
$$C8 \quad 10nF \cdot 10\% \pm 10nF = 9nF \text{ og } 11nF$$



Figur 49: 50 Hz nedregørse



Figur 50: 50 Hz AC- øvre grænse



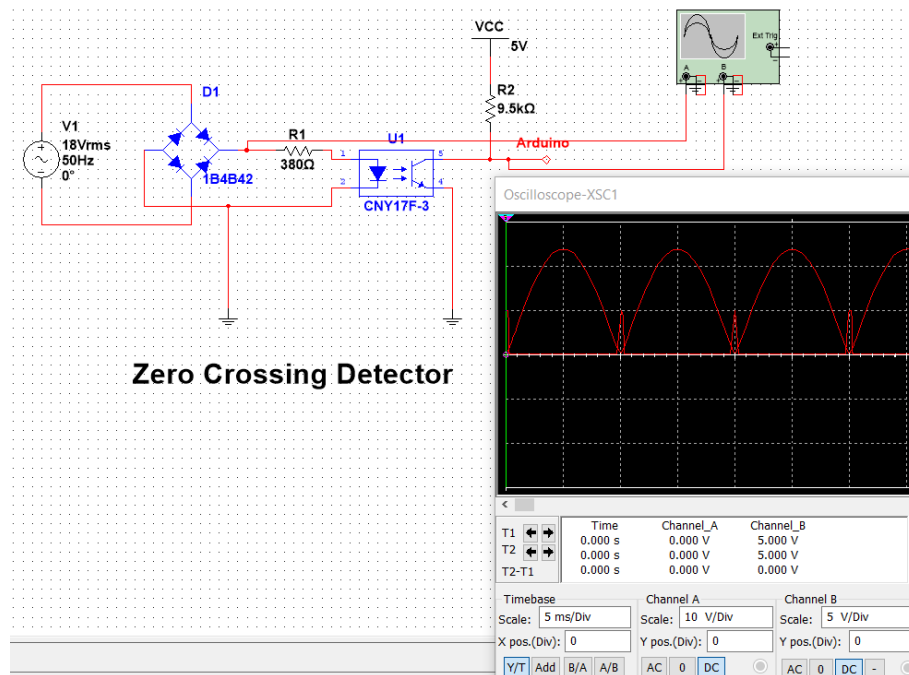
Figur 51: 120 kHz burst- øvre grænse

På baggrund af disse simuleringer kan der konkluderes at Carrier Detektor modulet ville virke i de ydre grænser af komponent-tolerancerne.

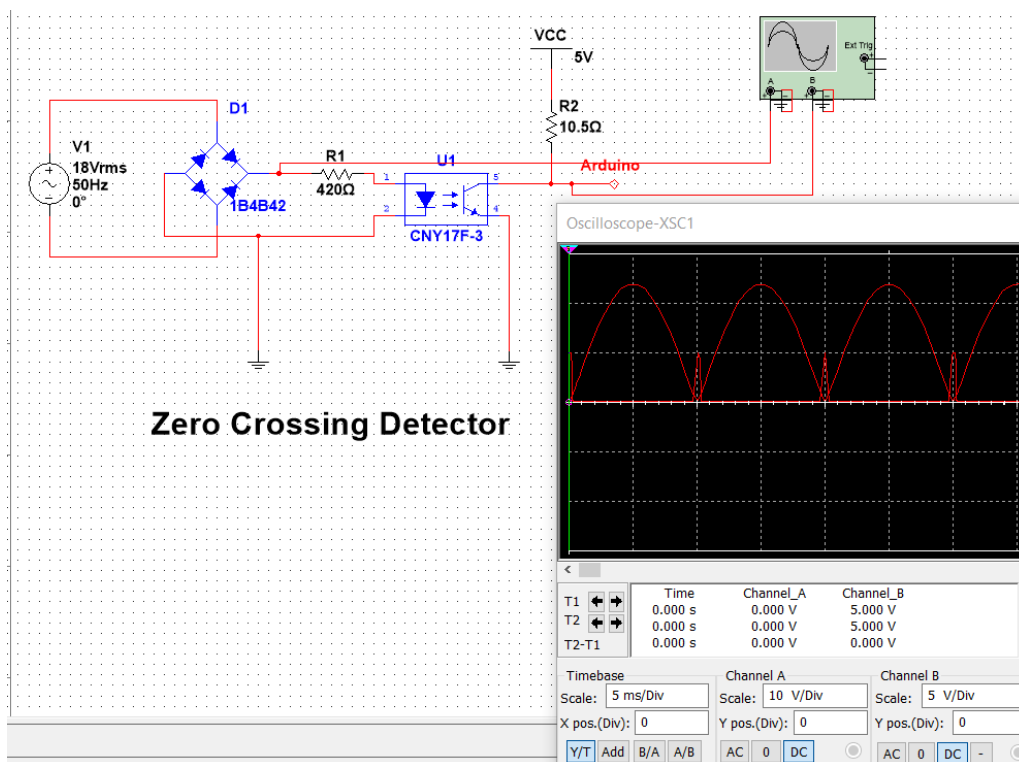
For ZeroCrossing modulet er der blot 2 modstande der skal testes, R1 og R2. Her er grænserne blevet beregnet til

$$R1 = 400\Omega \pm 5\% = 380\Omega \text{ og } 420\Omega$$

$$R2 = 10\text{ k}\Omega \pm 5\% = 9.5\text{ k}\Omega \text{ og } 10.5\text{ k}\Omega$$



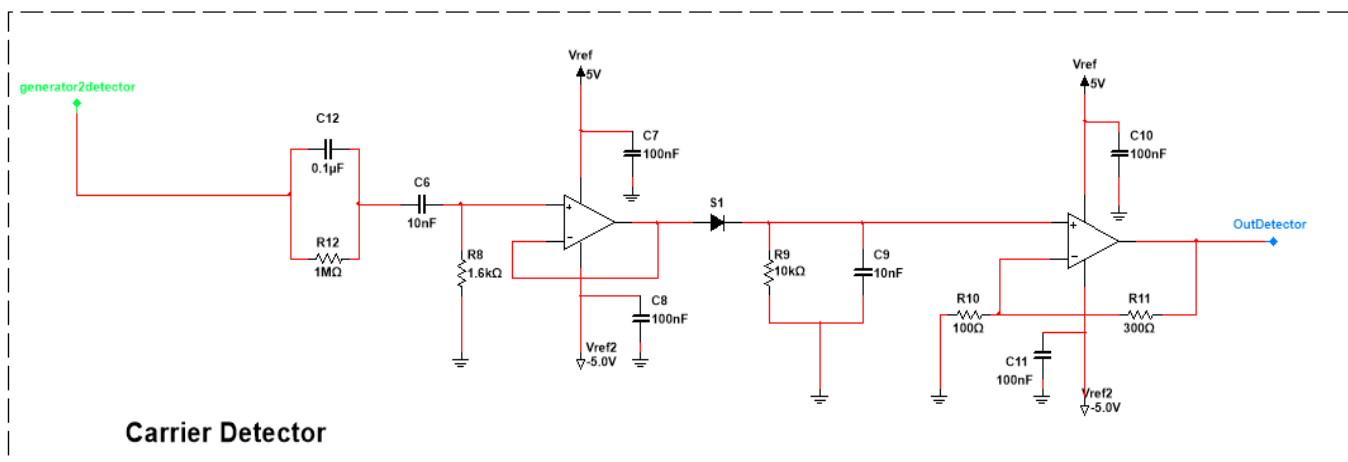
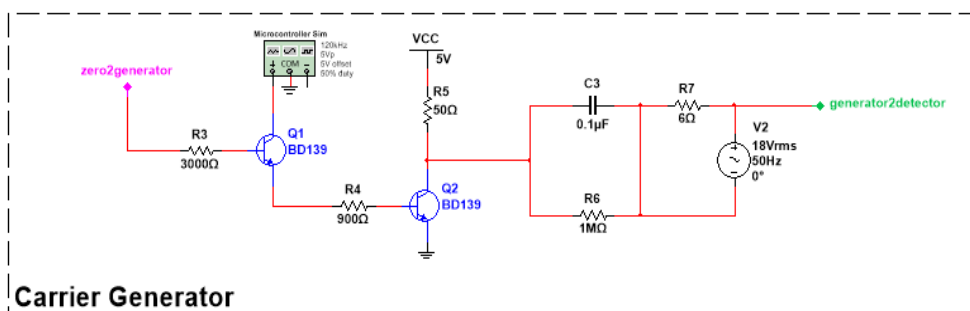
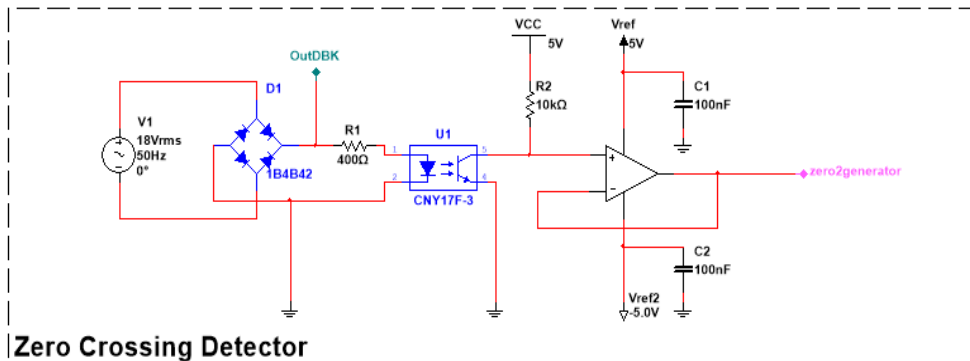
Figur 52: Nedre grænse



Figur 53: Øvre grænse

Der ses på de 2 figurer 52 og 53, at zero-crossing modulet er upåvirket af tolerancen på 5% af de to modstande, den vil derfor kunne fungere uden problemer.

HW7 Modultest hardware



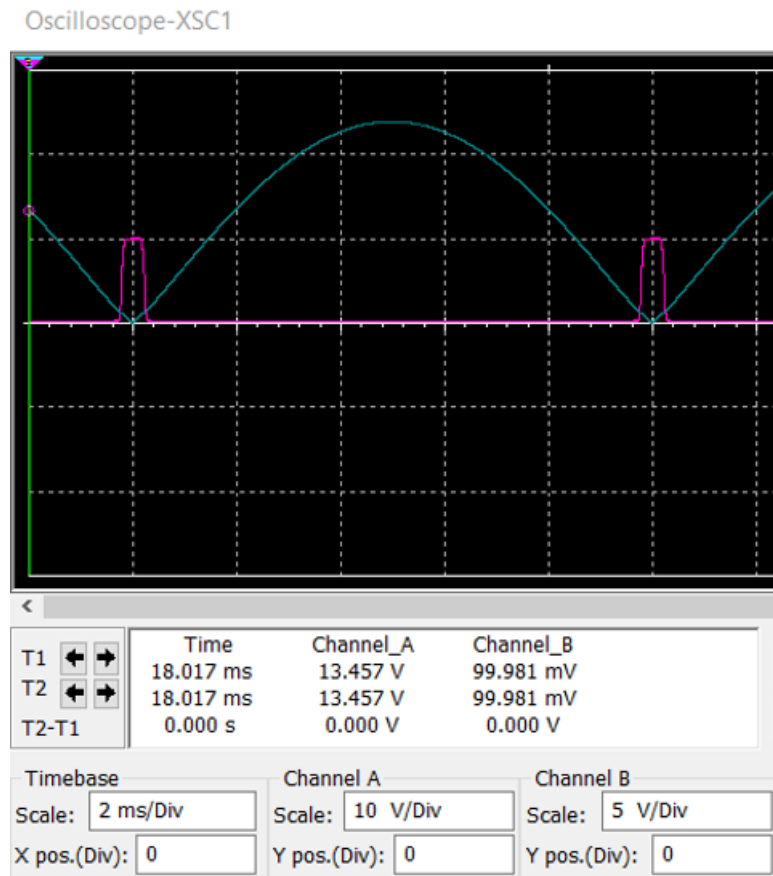
Figur 54: Samlet kredsløb for test af modulerne

For at lave modul test af de forskellige hardwaremoduler er der blevet valgt at forbinde de forskellige moduler, som der ses på figur 54. Her kan der ses at de forskellige dele der skal kommunikere, er blevet slået sammen, da det ikke har været muligt at bygge kredsløbet. Ved Carrier Generatoren er der blevet valgt at påsætte en transistor og en funktionsgenerator før kredsløbet. Dette skal fungere som om Arduinoen sender 120kHz bursts ud. Normalt ville der vha. software bestemmes hvornår der skulle sendes bursts ud, men dette er ikke muligt i en simulering. I stedet er transistoren blevet sat ind, der her virker som en switch. Når der er zero-crossing, bliver transistoren mættet og funktionsgeneratorens output på 120 kHz løber ind i sender kredsløbet.

Fordi der kaskadekobles forskellige kredsløb samt for at holde styr på impedanserne, er der blevet sat operations forstærkere imellem kredsløbene. På hver operations forstærker er der blevet indsat afkoblings kondensatorer på 100 nF for at minimere det støj der potentielt havde været der når

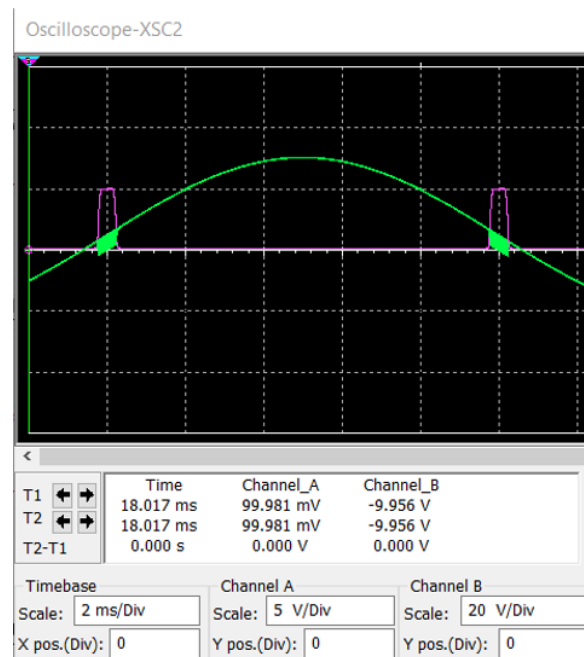
kredsløbet blev bygget. Herved tages de nødvendige forholdsregler der normalt var blevet taget under opbygning af det reelle kredsløb.

Ved at se på figur 55, ses oscilloskopet for hvordan Zero Crossing signalet ser ud sammenlignet med 18 V AC-kredsløbet, efter det er blevet ensrettet af diodebroen.



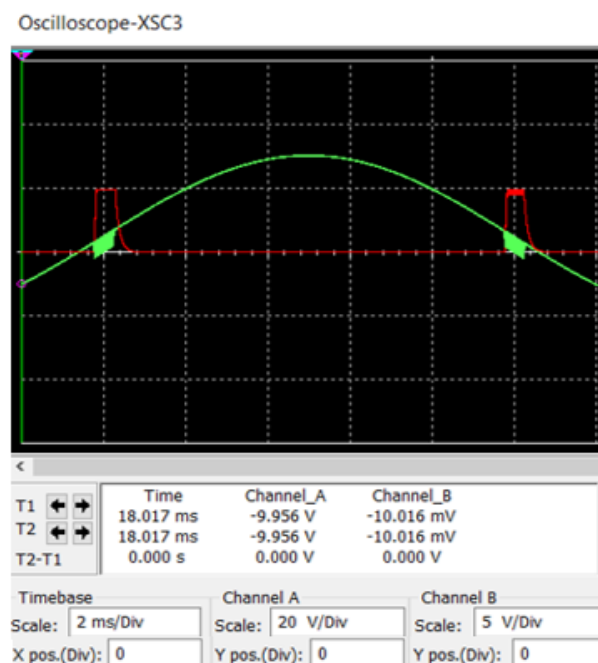
Figur 55: Zero Crossing scope

På figur 56, vises hvordan det lukkede kredsløb vil se ud efter at sender modulet har sendt bursts af 120 kHz ud på 18 V AC-kredsløbet på den grønne kurve og på den lilla kurve ses zero-crossing i forhold til. Ved at sammenligne de to grafer ses der at Zero-Crossing og bursts af 120 kHz ligger lige oven i hinanden. Det skal dog også særligt bemærkes at de 120 kHz bursts ikke kommer nøjagtigt på Zero-Crossing, begge signaler er forskudt. Dette skyldes at Carrier Generator kredsløbet bliver pullet op med 5 V. Den måde der er blevet valgt at slå 120 kHz burst og 18 V 50 Hz signalerne sammen, resulterer i et offset på 5 V. Der kan også ses at channel A på figur x, er scalet med 20 V/Div, herved kan der ses at 50 Hz, signalet peaker på 30 V, i stedet for de ~25 V der blev beregnet i design afsnittet. Herved er Zero-Crossing blevet forskudt med 5 V i simuleringen, i praksis ville dette ikke ske.



Figur 56: El-net efter Carrier generator, samlet kredsløb

På figur 57, vises hvordan Envelope Detektor modulet ville håndtere input med 120 kHz bursts fra senderen. Her kommer der bursts ved hvert Zero-Crossing på den grønne kurve. Den røde kurve på figuren viser at der ved hvert burst a 120 kHz, vil være et logisk højt signal på 5 V, hvilket kan oversættes til 1'ere når der er et logisk højt signal på modtageren, og 0'er når der ikke er nogen.



Figur 57: Carrier detektor ved samlet kredsløb

Ud fra de 3 figurer med scopes på hvert hardware modul kan der ses at de 3 moduler kan kommunikere med hinanden. Senderen sender bursts ud når der kommer Zero-Crossing og modtageren laver et logisk højt signal, når den modtager bursts a 120 kHz. I realiteten vil de 3 blokke ikke side på en enkelt

hardware enhed. Sender enheden vil have en Zero-Crossing detektor og en Carrier-Generator. Modtageren vil også have et Zero-Crossing modul og en Carrier-Detektor. Zero-Crossing modulet er det eneste modul der er på begge hardware enheder, da det er instrumentalt at de forskellige enheder ved, hvornår der er Zero-Crossing. Herved sikres der at der kan aflæses 1'ere og 0'ere, da de forskellige moduler ellers havde læst i 'blinde'. En strøm af 1'ere og 0'ere kan altså ikke aflæses uden Zero-Crossing detektoren, da der ellers ikke vides, hvornår der ikke bursts ved Zero-Crossing.

Referenceliste

- Ref[1]** \Dokumentation\Bilag\PDF-filer\Standard Bodeplot.PDF
- Ref[2]** <https://www.youtube.com/watch?v=vxzR3W2BcRk> (Simulink med Dr. Ryan Ahmed)
- Ref[3]** <http://www.bristolwatch.com/ele2/zcnew.htm> ZeroCross inspiration
- Ref[4]** \Dokumentation\Bilag\PDF-filer\1B4B2.PDF
- Ref[5]** \Dokumentation\Bilag\PDF-filer\CNY17F.PDF
- Ref[6]** \Dokumentation\Bilag\PDF-filer\Arduinomega2560.PDF
- Ref[7]** \Dokumentation\Bilag\PDF-filer\AN236_ApplicationNote.Pdf
- Ref[8]** \Dokumentation\Bilag\PDF-filer\BD139.PDF
- Ref[9]** \Dokumentation\Bilag\PDF-filer\Standar_ decade_value_table.PDF
- Ref[10]** \Dokumentation\Bilag\PDF-filer\120_kHz_elnet.PDF
- Ref[11]** https://www.st-andrews.ac.uk/~www_pa/Scots_Guide/RadCom/part9/page2.html
- Ref[12]** https://www.electronics-tutorials.ws/filter/filter_3.html