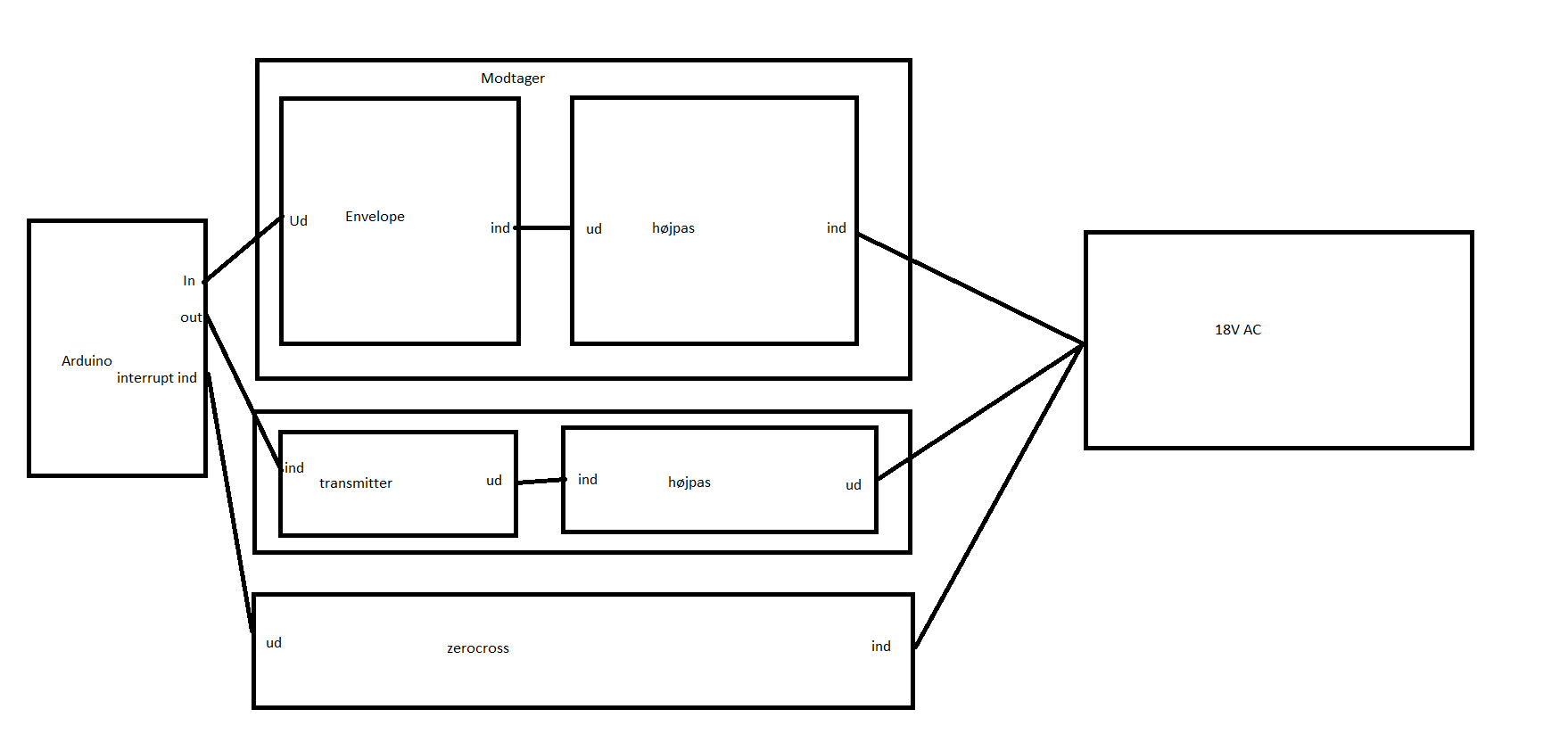
Dokumentation

Forklaring af hvorfor jeg ville gøre det sådan, i er velkommen til at træffe andre valg, men husk vi er under tidspres

# Overordnet Blokopdeling

Overordnet set ville jeg vælge følgende kombination af kredsløbsblokke for at tillade kommunikation over AC nettet ud fra X.10 krav til hvordan det fysiske signal skal formes og sendes samt modtages.



Billedet her kan være lidt utydeligt men ligger for sig selv her i mappen også, men jeg ville opdele det i 5 overordnede blokke:

## Blok 1: Arduino/MCU

Her skal der anvendes 4 pins men 3 signaler, forklaring følger:

En IO pin til at modtage data på figuren benævnt ind på arduino blokken.

En IO pin til at generere vores data signal, dette signal ville jeg umiddelbart benytte en PWM timer til at genere sender ved duty på 50% sender ikke ved duty på 0% ;)

2 Interupts, der modtager det samme signal der danner vores timing fra zero cross, det ene skal trigger på rising det andet på falling edge, forklaring på dette kommer i forbindelse med zero cross.

## Blok 2: 18 V AC

Dette er blot vores elnet i vores tilfælde er det 18 V AC. Signalet her forbindes med de 3 blokke der udgør de reelle x10.1 blokke.

## Blok 3: Zerocross

Her vil jeg anbefale at benytte design 4 af de design der er lagt op da dette giver os muligheden for at adskille hvornår vi er på en rising zero cross og et falling, hvilket er relevant i forhold til at kunne afkode transmissionen igen samt kunne sende på de rigtige tidspunkter så vi ikke får byttet rundt på hvad der er 1 og hvad der er 0. Se X.10 dokumentation for hvordan det fysiske signal udformes. Kredsløbet bliver forklaret senere.

## Blok 4: modtagerkredsløb

Modtager kredsløbet består af 2 mindre blokke, en aktivt højpasfilter for at undgå støj og sikre at filtreringen er effiktiv, men uden et unødvendigt stort overshoot ved cutoff frekvensen. Jeg vil personligt anbefale det filterdesign jeg lagde op i går da stammer fra pensum omkring aktive anden ordens højpasfiltre, mere forklaring kommer senere.

18 V AC overlagt med 120 kHz data vil passere ind i filtret og ud fra filtret vil der komme et 120 kHz data signal. Dette signal sendes videre til en envelope detector der omkoder dette signal til noget der minder om et digitalt high signal, dette signal sendes til arduino’s indgangs io pin, og aflæses i forhold til timing på interrupts.

## Blok 5: sender kredsløbet

Sender kredsløbet består lige som modtagerkredsløbet af 2 blokke, stadig et højpasfilter, men i stedet for en envelope detector benytter vi et transistor kredsløb til at generere vores 120 kHz data signal.

Da de 18 V AC samt støj vil kunne lave ravage for transistor kredsløbet og vores microcontroller skal der være et højpasfilter imellem kredsløbet, her kan vi vælge mellem 2 løsninger.

Løsning 1:

et aktivt filter magen til det fra lavpasfiltret, hvilket er den løsning jeg anbefaler, da vi mindsker støjsignaler bedre samt får et opamp output så vi sikre at filtrets evne til at filtrere ikke påvirkers af eventuelle belastninger eller andre filtre forbundet til signalen.

Løsning 2:

Ser vi bort fra dæmpningsevnen kan dette filter godt laves som et simpelt passivt højpas filter, med en op amp voltage follower som output, den voltage follower er vigtig i denne sammenhæng for netop at undgå at filtret påvirker andre filtre i kredsløbet ( se pensum om kaskade kopling af filtre, overføringsfunktionen gennem kredsløbet kan ændre sig drastisk hvis der ikke er opamp output ).

Min Anbefaling:

Benyt samme aktive filter begge steder, da der ikke er noget grund til at spilde tid på at designe 2 forskellige filtre der i realiteten udføre samme funktion, samtidig giver løsning 1 en 40db/dec’s dæmpning frem for 20db/dec som løsning 2 vil give.

# Ang. Applikationsnoten

Applikationsnoten er til inspiration, ikke en facit liste, det er mening at den skal give ideer om hvad der er behov for at lave, og så skal pensum anvendes til at løse dette på den bedst mulige måde, der er mange ting i den applikationsnote der kan give problmer hvis implementeret i eksempeltvis en moderne hustand, så der hvor vi kan lave bedre løsninger uden at det koster os mere tid, så lad os gøre det. Deres filter løsning er egentlig kun et afkoblingsfilter hvorefter de også nævner at der skal et extra filter på med en op amp tuned cutoff, hvilket også er fint nok, men det giver det mere mening at lave en bedre filter løsning ud fra de signaler vi ved der skal bruges, derfor vil i se både ligheder og afvigelser fra applikationsnoten gennem dette dokument.

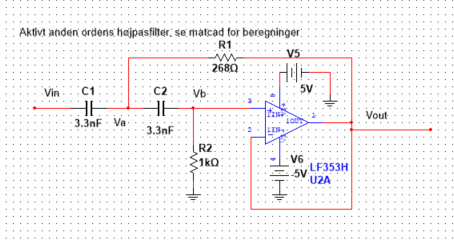
En anden ting man skal gøre sig klart er om man vil lave noget der kun virker i lab, eller stræbe efter at lave noget der reelt set kunne være brugbart uden for lab også, hvor man komme ud i et ikke lukket system med mange andre ting der spiller ind. Da vi har lært om 2. ordens filtre og deres egenskaber, ville det næsten være dumt ikke at gå den vej, når det nu ikke er meget svære at designe end eksempeltvis et 1. ordens passivt med voltage follower.

# Højpasfilter:

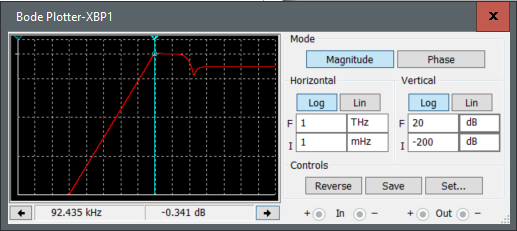
Jeg vil anbefale brugen af et aktivt højpas af 2 grunde, den ene er den øjede dæmpning pr dekade, den anden er op amp udgangen der gør at vi undgår en hel del problemer i forhold til at have flere filtre i systemet.

Højpasfiltret her er ikke meget forskelligt fra det lavpasfilter vi lavede øvelse på i ASB, alle beregningerne findes med forklaringer i det matcad dokument der ligger sammen med dette worddokument, komponentværdierne her er en smule afrundede i forhold til de beregnede hvilket giver en lille variation i cutoff og Zeta, men det er stadig inden for et acceptabelt område ud fra simuleringen.

Diagrammet for filtret alene, så der ikke er noget forvirrende elementer ud over opampens spændingsforsyning ses her:



I multisim kan der genereres et bodeplot der viser filtrets egenskaber, dette kan gøres i den vedlagte multisim fil. Men bodeblot er indsat her for overskuelighedens skyld.



Dette skal selvfølgelig testes på fumlebræt, men det giver et overordnet billede af filtrets egenskaber.

Filtret her kan anvendes både i sender og modtager kredsløbet det skal dog vendes i forhold til at senderen skal have op amp udgangen ud mod de 18 V ac, hvor modtageren skal have op amp vendt mod envelope detektoren.

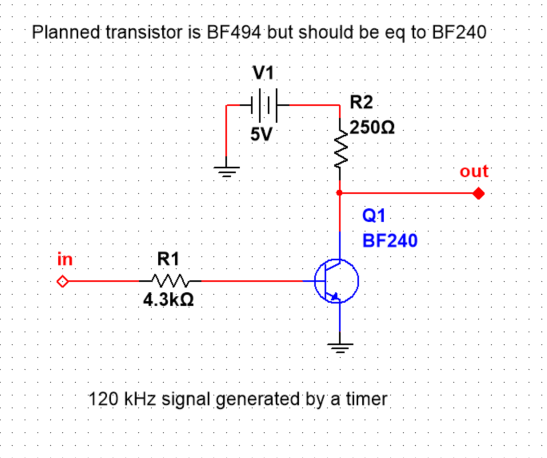
Fordelen ved at anvende dette ved senderen er den bedre dæmpning der beskytter vores microcontroller og sender kredsløb fra støj og andre udefrakommende signaler, ved godt det ikke er et problem i lab, men det vil det kunne være i den virkelig verden, og derfor syntes jeg det giver mening at lave det så det dæmper så godt som vi nu kan her og nu.

# Sender transistor kredsløb

Transistor kredsløbet er opbygget ud fra applikationsnoten, dog med valg af en transistor der ud fra datasheet har hutig nok respons til at kunne klare 120 kHz, transistoren er valgt ud fra komponentlagerets liste og hedder BF494, i multisim er anvendt BF240 da den skulle være equivalent ifølge komponenlagerets side.

Beregner findes i det tilhørende matcad dokument sammen med forklaringer på disse, og det er fra disse beregninger komponentværdierne kommer.

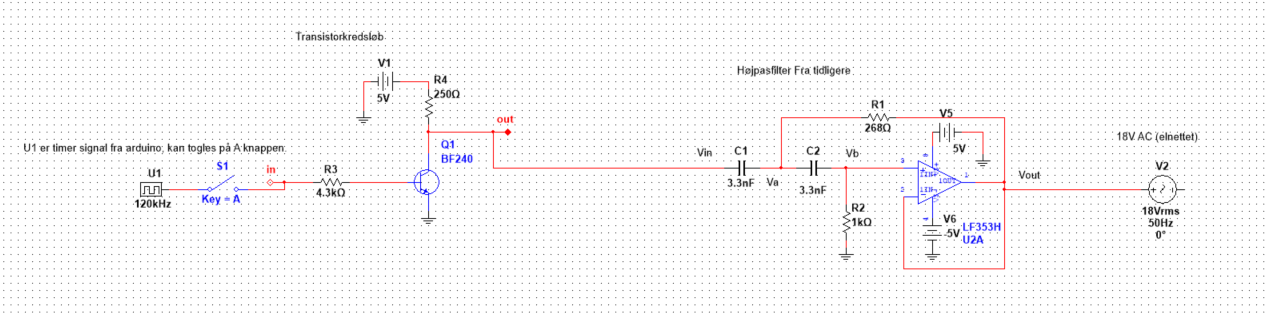
Det resulterene kredsløb ser ud som følger:



Sendes der en 120 kHz firkantpuls fra timeren ind på in porten genereres der et 120 Khz 0-5V signal på udgangen, kan simuleres i vedlagte multisim dokument.

# Samlet Sender kredsløb

De bygge sten vi har nu kan danne et senderkredsløb når de bliver sammensat korrekt:



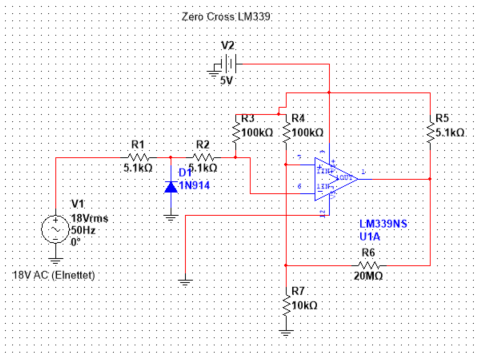
På diagrammet ses en clock der representere vores udgang på microcontroller, der går ind på indgangen af transistorkredsløbet, udgangen føres gennem højpasfiltret og videre ud på elnettet, højpasfiltret sørger samtidig for at de 18 V AC ikke passere ind til transistor kredsløbet.

Senderkredsløbet kan simuleres i den vedlagte multisim fil, zero cross er ikke med her da den del håndteres i software ud fra en clock genereret af zero cross detectoren. Ved simulering kan det ses at det muligvis kan blive nødvendigt med en pull up/down resistor på indgangen til transistor kredsen, men det kan også håndteres i timeren ved justering af duty cycle til eksempeltvis 100%.

# Zero Cross

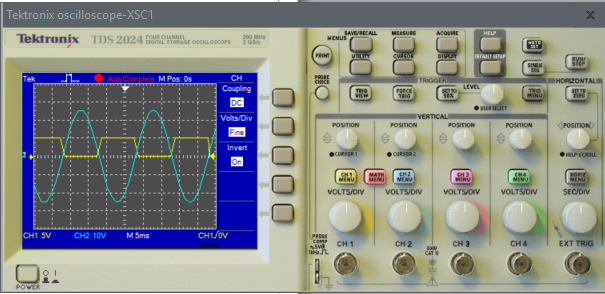
Zero cross kredsløbet er lavet ud fra en komparator da det der egentlig skal gøres er at sammenligne 2 spændinger, jeg kiggede listen igennem på komponentlageret og fandt LM339 samt nogle andre, der ligger 4 designs i en anden mappe hvoraf dette var design 4. Designet er taget direkte ud fra datasheetet og findes på side 14 i datasheet der ligger her i mappen for zero cross detektoren.

Det resulterende kredsløb i multisim ser ud som følger:



Eneste ændring i forhold til datasheet er at spændingsforsyningen til komparatoren er udskiftet til en 5V dc forsyning.

Simulering kan laves med det vedlagte multisim dokument og giver et scobe output der ser ud som følger:



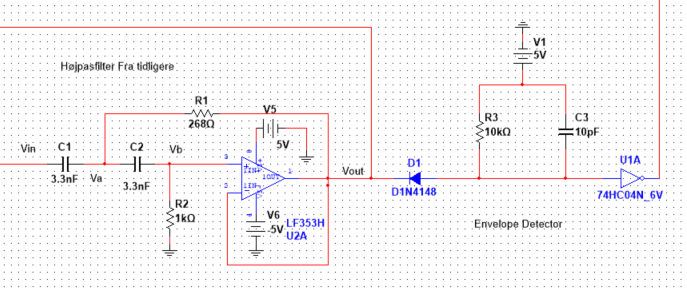
Som det ses vil der være lidt delay på timingen, men ikke noget der ikke kan korrigeres for i software hvis det i realiseringen er nogenlunde tilsvarende, dog kan dette delay nok forbedres ved brug af en evt. schitt trigger kreds inden til at gøre overgangen skarpere. samtidig giver dette mulighed for at adskille hvor vi er i perioden så ved ved om vi er på en rising eller en falling ud fra clock signalet. Dette skal køres ind på 2 interrupts på arduino. Som ud fra det kan udlæse eller sende de korrekte bits, det er ikke nødvendigt med mere end en zero cross detektor for at kunne både sende og modtage.

# Envelope detektor

Den er designet ud fra applikationsnoten af mikkel, så den må han kunne fortælle lidt om, men kort sagt omdanner den vores analoge 120 kHz til et tilnærmelsesvis digitalt signal.

# Samlet Modtager kredsløb

Det samlede modtager kredsløb består af en envelope detector samt et højpasfilter som det ses på nedenstående diagram:

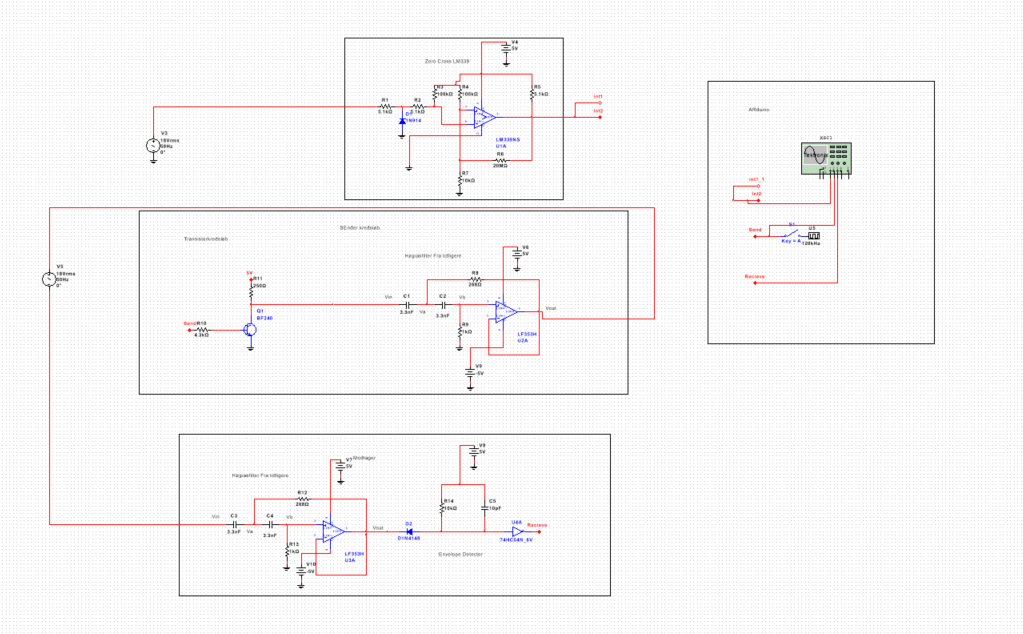


Ved at anvende samme filter begge steder undgår vi at lave 2 designs samtidig kan vi udnytte at vi har 2 opamps i 1 chip.

# Samlet x10.1 setup

Sætte vi de blokke vi har her sammen får vi et samlet kredsløb som kan ses i den vedlagte multisim fil der hedder samlet.

Jeg har tegnet lidt kasser på for at illustrere blok opdelingen.



Systemet kan simuleres i multisim filen.