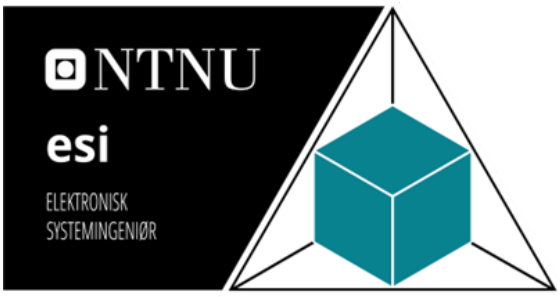


Signalnivå-generator

Tittel: Designprosjekt 1: Signalnivågenerator

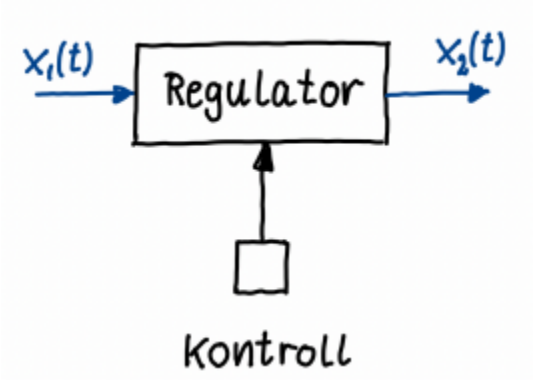
Forfatter: Andreas Lindeman

Versjon: 1.8

Dato: 15.04.2024

Inneholdsfortegelse

1. Problemstilling



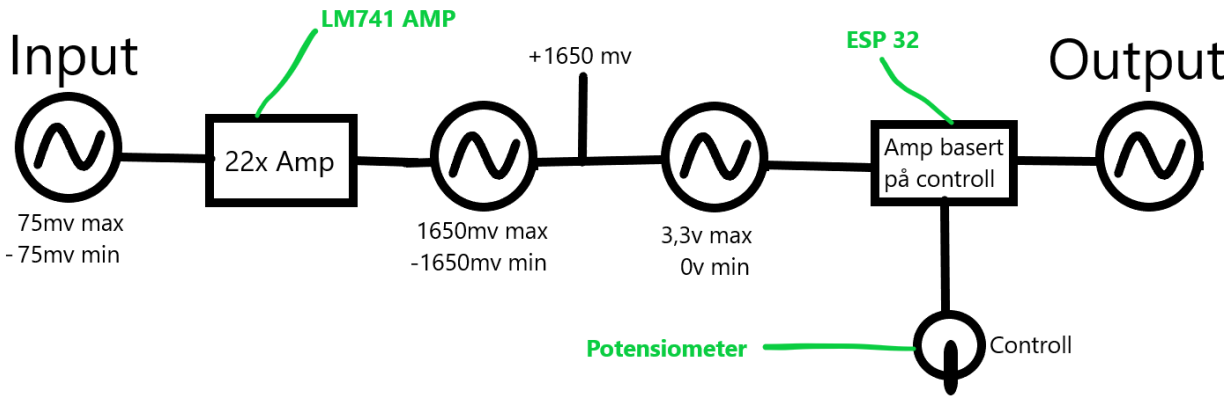
[1]Figur 1

Oppgaven innebærer design og implementering av en signalnivåregulator som opererer på et analogt signal, betegnet som X1(t), og produserer et justert signal, betegnet som X2(t). Forholdet mellom disse to signalene er gitt ved ligningen:

$$X2(t) = A*X1(t)$$

Her representerer 'A' en forsterknings- eller dempningsfaktor, avhengig av verdien. Denne faktoren bestemmes av brukeren gjennom en roterbar kontroll.

2. Prinsipiell løsning



Figur 2

Den foreslåtte løsningen er basert på prinsippet om signalforsterkning og demping. Det første trinnet innebærer forsterkning av det sinusformede signalet, X1(t), noe som forbedrer

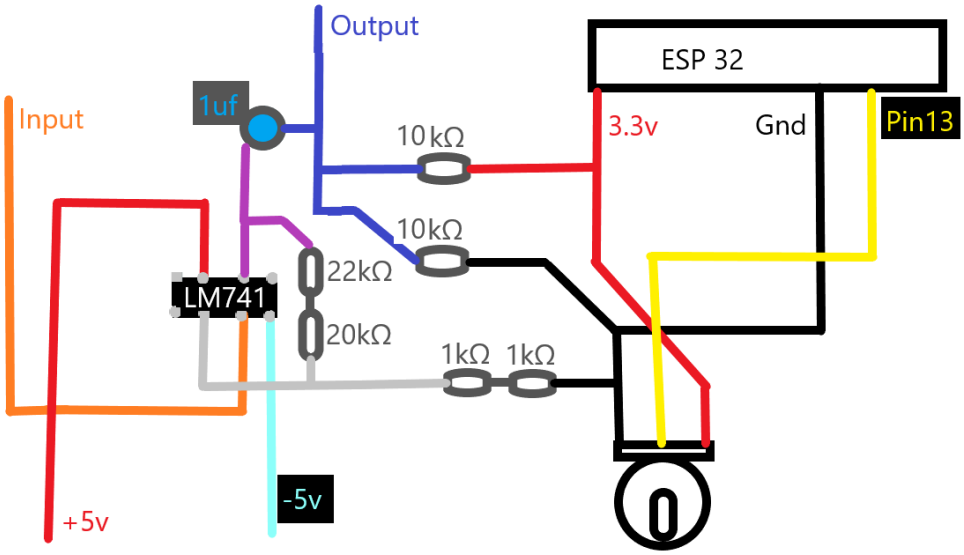
avlesnings nøyaktigheten for datamaskinen. Dette følges av fjerning av den negative spenningen, noe som sikrer at signalet forblir innenfor et bestemt område som er egnet for videre behandling.

Det bearbeidede signalet sendes deretter videre til datamaskinen, som fungerer som den sentrale kontrollenheten. Datamaskinen er ansvarlig for å justere signalet basert på brukerens inngang. Denne justeringen utføres ved enten å forsterke (øke signalstyrken) eller dempe (redusere signalstyrken) signalet.

Brukerens inngang fanges opp gjennom et potensiometer, en variabel motstand som lar brukeren modulere signalstyrken. Ved å justere potensiometeret kan brukeren kontrollere forsterknings- eller dempningsfaktoren 'A', og dermed påvirke det endelige utgangssignalet, X2(t).

Dette designet tilbyr en enkel, men effektiv løsning for regulering av nivået på et analogt signal. Det gir brukeren fleksibilitet til å justere signalstyrken etter deres krav, noe som gjør det til et allsidig verktøy for forskjellige applikasjoner.

3. Realisering og test



Figur 3: Blå output er signalet som blir ført inn i ESP32 før siste behandling.

For å oppnå presis kontroll over demping og forsterkning av et sinusformet signal, er en ESP32 mikrokontroller implementert. Inngangssignalet har en frekvens på 229Hz og en amplitude på 75mV. Siden ESP32 sin DAC bare kan generere et signal fra 0-3.3V, blir maksimal økning 3300/(75*2) = 22 (26.85dB). Siden utgangen til DAC er uint8, er den minste spenningsverdien

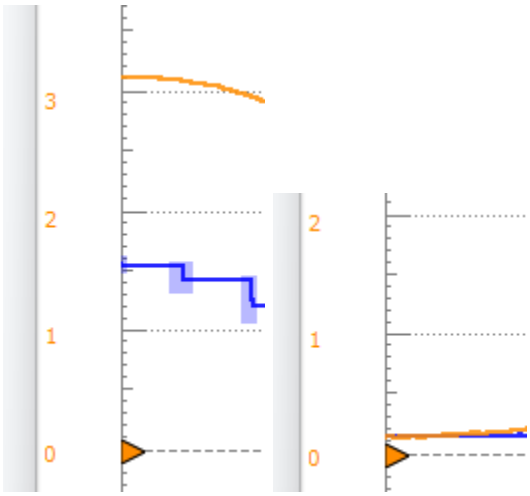
DAC kan produsere 3.3/255 Volt. Dermed blir den største dempingen (3300/255)/150 = 0.086 (-21.28dB).

Systemets oppløsning er basert på et potensiometer som måles med en ESP32. ESP32 sin analoge inngang har en oppløsning på 4096 (0 til 4095). Dette blir dermed oppløsningen for demping/forsterkning i systemet.

Oppløsning i dB: (26.85 - (-21.28))/4096 = 0.01175dB

Systemet har en teoretisk oppløsning på 0.01175dB. Denne nøyaktigheten vil naturligvis bli sterkt påvirket av støy.

For å beregne støy er det hovedsakelig to komponenter vi bør fokusere på. Den første er forsterkeren.



Figur 4: Volt på y-aksen, tid på x-aksen, orange representerer utgangen på forsterkeren

Forsterkeren (oransje linje på bildet) skal ha et forventet toppunkt på 3.3V og bunnpunkt på 0V. Den reelle verdien vi får er toppunkt på 3.1V og bunnpunkt på 0.12V. For å beregne SNR[dB] kan vi bruke denne formelen:

$$\mathrm{SNR}_{\mathrm{dB}} = 10 \log_{10} \left(\frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}} \right).$$

[2] Figur 5: Formel for SNR[dB]

Vi kan dermed bruke formelen og tallene over til å beregne SNR[dB]:

10*log10((3.3**2)/(0.2**2)) = 24.35dB
10*log10((3.3**2)/(0.12**2)) = 28.79dB

Vi får dermed en støy på SNR[dB] verdi på rundt 24dB - 28dB (rundet ned).

Dette signalet, nå med litt støy, går så inn i ESP. Ut av ESP kommer så et signal basert på verdiene fra potensiometeret. Støyen her er mye basert på hvor mye den skal skalere signalet med. Når signalet har den minste amplitude ESP kan lage ((3300)/255)/2 = 6.47mV, vil signalet se ut som et digitalt signal (bare verdier på 6.47mV og 0V). Dette vil naturligvis skape mye støy. En mulig løsning på dette er å implementere mer maskinvare som “glatter” ut signalet. Dette er en tenkelig oppgradering av systemet.

Når det kommer til koden, må den også være ganske optimalisert. Dette var for å minimere tap laget av systemet. Siden ESP bare må kjøre en funksjon for å endre utgangen på DAC er det viktig at denne koden blir kraftig optimalisert. Dermed vil vi at den konstant jobber som en forsterker eller demper og dermed gir den høyeste nøyaktigheten den kan.

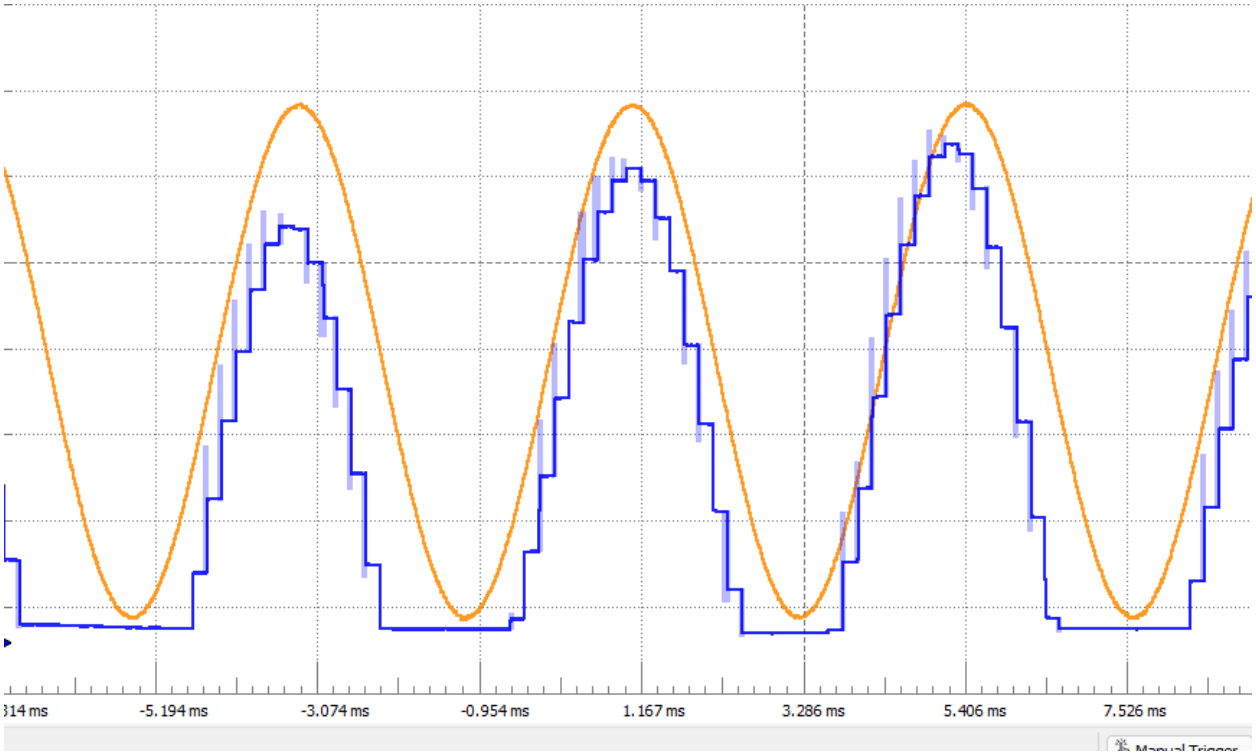
Koden som kjører konstant i en ESP32 er definert inne i en loop-funksjon. Min loop-funksjon ser slik ut:

```
void loop() {
  dacWrite(25, ((int)analogRead(35)*(int)(analogRead(potPin)))>>16);
}
```

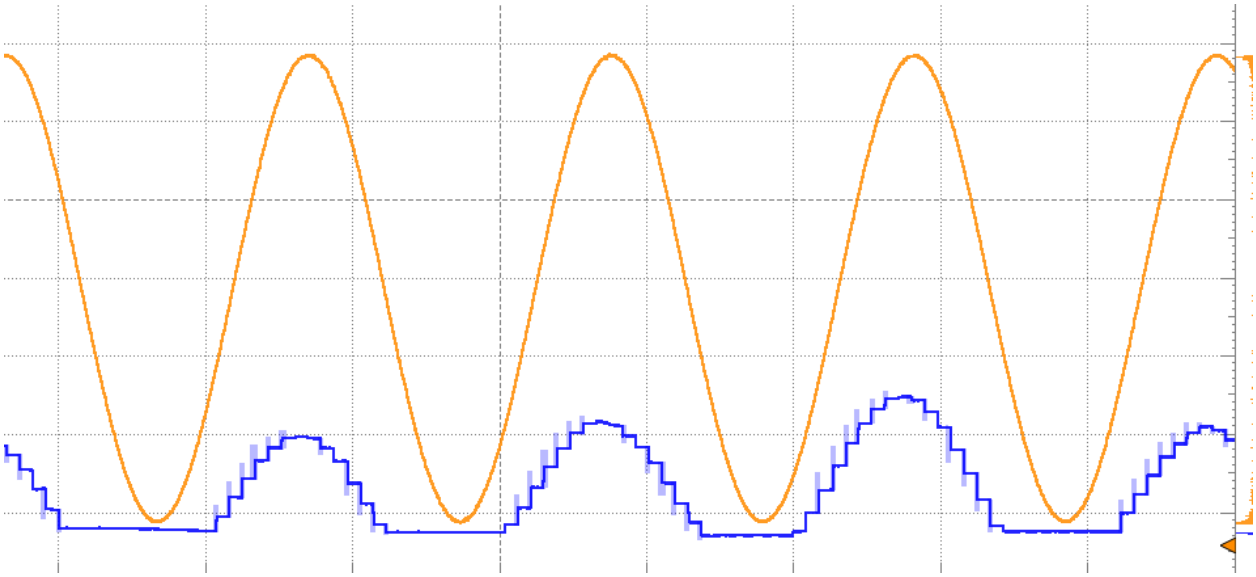
AnalogRead funksjonen leser verdier fra en ADC-port. Port 35 er valgt for systemet og port 13 for potensiometeret (referert til som potPin i koden). Verdiene er av typen uint16, som betyr at de kan variere mellom 0 og 65535. Da analogRead aldri returnerer verdier høyere enn 4095, oppstår det sjelden problemer. Imidlertid, hvis to slike verdier multipliseres og resultatet beholdes som uint16, kan det oppstå feil. For å unngå dette konverteres verdiene til int32 før multiplikasjon.

Etter multiplikasjonen får en verdi mellom 4095*4095 og 0. Disse verdiene er for store for dacWrite, som krever en uint8 (verdi mellom 0 og 255). For å konvertere disse verdiene til riktig format, brukes en bitshift-operasjon, som tilsvarer en divisjon med 2^(antall bits), men uten desimaler i resultatet. Koden bitshift-er 16 bits noe som vil gjøre verdier mellom 4095*4095-1 og 0 til verdier mellom 255 og 0, som det dacWrite bruker.

Fugerende eksempel bildet:

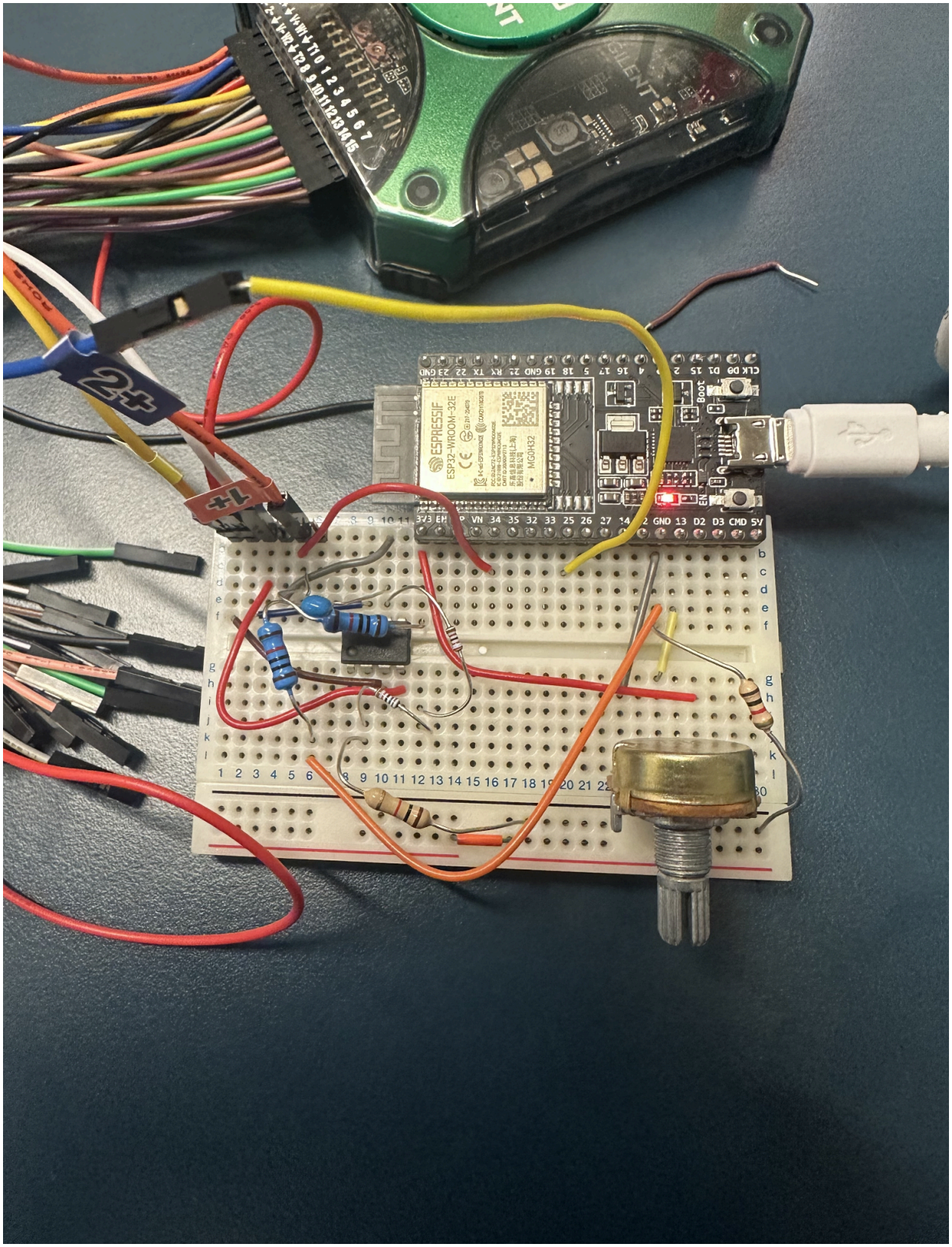


Figur 6: Systemets utgang (blå) blir sterekere og sterekere.



Figur 7: Samme mønster som på figur 6, men systemets utgang (blå) starter lavere

Under ser du bildet av systemet realisert i virkeligheten:



hh

Figur 8: Bildet av kretsen

4. Konklusjon

Gjennom implementering og testing av designet, har det blitt oppnådd en effektiv kontroll over demping og forsterkning av et sinusformet signal ved bruk av en ESP32 mikrokontroller. Systemets oppløsning, basert på et potensiometer, har vist seg å være presis med en teoretisk oppløsning på 0.01175dB. Imidlertid har støy blitt identifisert som en betydelig faktor som påvirker systemets nøyaktighet, med en estimert SNR på omtrent 24dB.

I forhold til de opprinnelige spesifikasjonene, har designet demonstrert en maksimal forsterkning på 26.85dB og en maksimal demping på -21.28dB. Dette viser at designet i stor grad oppfyller de opprinnelige kravene. Videre har potensielle forbedringer blitt identifisert, som inkluderer implementering av ytterligere maskinvare for å redusere støy og optimalisering av koden for å minimere tap.

5. Takk

Jeg vil gjerne takke Mathias Schiøtz for fruktbare diskusjoner under designarbeidet.

Referanser

[1] Elektronisk systemdesign og implementering 1 (ESDI 1), BlackBoard notat, 2024.

[2]https://wikimedia.org/api/rest_v1/media/math/render/svg/7a7b11f600ce6e7defb4f20ffa8071818b6baa14