

# Planering av sälpinger

Christian Aakula

Examensarbete för ingenjör (YH)-examen

Utbildningsprogrammet för elektroteknik

Vasa 2015



## EXAMENSARBETE

Författare: Christian Aakula  
Utbildningsprogram och ort: Elektroteknik, Vasa  
Inriktningsalternativ/Fördjupning: Automationsteknik  
Handledare: Roger Mäntylä, Matts Nickull

Titel: *Planering av sälpingar*

---

Datum 20.5.2015

Sidantal 49

Bilagor 4

---

### **Abstrakt**

Detta examensarbete behandlar huruvida sälpingers fungerar i praktiken, vad det finns för alternativ samt exempel på hur en sälping kan byggas. En sälping är en liten krets som används i fiskenät eller vid fiskeodlingar för att hålla sälar borta. Syftet med detta arbete var att försöka utveckla en fungerande sälping som skulle kunna återanvändas, istället för att slängas då batterierna tagit slut.

Detta arbete tar fram sätt att generera ljudbilder som antas skrämma sälar, med hjälp av en mikroprocessor samt med analog elektronik. Det tas också upp några olika sätt att försörja dessa med ström, samt ett exempel på högtalare som kan användas.

Resultatet av detta examensarbete blev en mikroprocessor med två olika koder. Detta för att jämföra strömförbrukningen av processorn då man själv ställt in all hårdvara exakt som man vill ha den kontra då man har använt Arduinos egen kod. Även en krets med en 555-klocka har tagits fram för att generera frekvenser.

---

Språk: svenska

Nyckelord: sälpingar, sälskrämma, frekvens

---

## BACHELOR'S THESIS

Author: Christian Aakula  
Degree Programme: Electrical Engineering, Vaasa  
Specialization: Automation  
Supervisors: Roger Mäntylä, Matts Nickull

Title: *Planning of a Seal Pinger*

---

Date 20.5.2015                      Number of pages 49                      Appendices 4

---

### Summary

This Bachelor's Thesis covers whether seal pingers work or not, what possible alternatives to seal pingers there are and how a seal pinger could be created practically. A seal pinger is a device fishermen use in their fishing nets or in fish farms to keep seals away from the fish. The objective of this thesis is to create a working seal pinger with the possibility to recharge the power supply instead of having to discard the pinger when it is out of power.

Several ways to generate different frequencies that supposedly scare seals are covered, both using a microprocessor and analogue technology. Ways to supply the circuits with power are also covered, and a possible buzzer that could be used has been found.

The final result of this thesis is a microprocessor with two different codes. One solution uses only the exact hardware desired, while the other uses the standard hardware settings and is created with software coding only. The reason for this is to compare the power usage between the two solutions. A circuit with a 555-timer has also been created to generate frequencies.

---

Language: Swedish              Key words: seal pinger, seal scarer, frequency

---

# Innehållsförteckning

1 Inledning.....	1
1.1 Bakgrund.....	1
1.2 Syftet med arbetet.....	3
1.3 Avgränsning av arbetet .....	4
2 Teori .....	5
2.1 Frekvenser som påverkar sällar .....	5
2.2 Uppbyggnad av kretsen .....	9
2.2.1 Digital frekvensgenerator .....	9
2.2.2 Analog frekvensgenerator.....	11
2.3 Drivsteg .....	13
2.4 Högtalare .....	16
2.5 Strömförsörjning till kretsen.....	16
2.5.1 Laddningsbart batteri.....	16
2.5.2 Utbytbart batteri.....	18
2.5.3 Förnybar energi .....	19
3 Förverkligande av sälpingern .....	21
3.1 Programmering av mikroprocessorn .....	21
3.1.1 Generering av pulser av olika frekvenser med hjälp av klockor .....	22
3.2.2 Generering av en chirpsignal med hjälp av mjukvarukodning.....	23
3.2 Analys av frekvenserna genererad av Arduinokoderna.....	25
3.3 Analys av strömförbrukning för de olika frekvensgeneratorerna.....	29

3.4	Exempel på högtalare .....	29
3.5	Val av drivsteg.....	30
4	Resultat .....	32
5	Diskussion .....	34
6	Källförteckning.....	35

## **Bilagor**

Bilaga 1: Kod som genererar pulser med fast frekvens med hjälp av hårdvara.

Bilaga 2: Kod som genererar chirp-signalpulser med hjälp av mjukvara.

Bilaga 3: Matlab-kod för analys av frekvenser, Fouriertransform, grafer osv.

Bilaga 4: Rådata strömförbrukning av frekvensgeneratorer.

# 1 Inledning

Ett problem för fiskare i Österbotten är att sälar stjäla fångst och förstör fiskenäten. Detta har lett till ett behov av att kunna hålla sälarna borta från fiskenäten. En metod som används är att skrämma bort sälarna med ljud som genereras med hjälp av en sälping.

En sälping är en liten apparat som sänder ifrån sig ljudsignaler ämnade att skrämma sälarna. Dessa ljudsignaler har olika frekvens och ljudstyrka, beroende på tillverkare. Det finns också pingers vars frekvens varierar, samt pingers som genererar så kallade chirp-signaler; det vill säga signaler som ökar i frekvens. Sälpingers appliceras i fiskenäten med ett visst avstånd för att hålla sälarna borta från alla delar av näten.

Detta examensarbete är en del av Yrkeshögskolan Novias projekt ”Energieffektivt fiske”, vars mål är att göra fisket i Österbotten mer effektivt ur energisynpunkt. Denna del av projektet behandlar utvecklingen av en sälping som inte behöver slängas då batteriet är slut, utan att den ska kunna användas under en längre tid. På så sätt minskar kostnaden för fiskare, samt uppkomsten av onödigt elektronikskrot.

## 1.1 Bakgrund

Bakgrunden till beställningen av detta arbete är att Österbottens Fiskarförbund har varit i kontakt med Forskning och Utveckling på Novia angående sälpingers. Österbottens Fiskarförbund har gjort en liten undersökning på hur dessa sälpingers påverkar fisket med nät i Österbotten. Tidigare undersökningar har gjorts i USA med positiva resultat, men i och med att djurlivet i USA:s sjöar är så olik djurlivet i Österbottens, är jämförelser av rapporter föga värdefullt. (Svanbäck, 2012)

Undersökningen som Österbottens Fiskarförbund har gjort har flera syften. Främst var syftet att se ifall sälpingers minskar på skadorna som sälarna åstadkommer i nätfisket. En annan intressant aspekt var att se ifall sälarna anpassade sig till sälpingerna. Det vill säga att sälpingerna skulle börja agera som ”vällingklocka” efter en tid, istället för att ha den

önskade skrämseffekten. Detta är något som sälvforskare har antytt kan vara en risk. (Svanbäck, 2012)

Under testet användes tre olika modeller av sälpingers: Savewave, Fumunda samt Aquamark. Dessa tre sälpingers har olika egenskaper och därför är även resultatet av användningen olika.

Tabell 1. Specifikationer på dessa tre sälpingers. (Svanbäck, 2012)

Tillverkare	Savewave	Fumunda	Aquamark
Rek. maxavstånd mellan pingers (m)	200	100	200
Vikt i luft (g)	400	230	410
Hörbar för människor	Nej	Ja	Nej
Batteri typ och antal	1 ingjuten 9 V-enhet	1 lithium	1 ingjuten D-Cell alkalisk
Batterilivstid (månader)	<3	15	16-24
Utbytbart batteri	Nej	Ja	Nej
Vatten kontakt	Ja	Ja	Ja
Tonal/Bred band	Bred band	Tonal	Bred band
Styrka (dB re 1µPa@1m)	155	132	140
Frekvens (kHz)	5-160	10	20-160
Puls varaktighet (ms)	200-900	300	200-300
Pulsintervall (s)	4-30	4	4-30



Figur 1. Bild på de tre testade sälpingarna. Från vänster sett Savewave, Futmunda, Aquamark. (Svanbäck, 2012)

Resultatet av testerna visade att den enda sälpingern som hade ett positivt resultat var Savewave-pingern. De andra två hade marginellt mer fångst i kontrollnäten (näten utan sälpingers) i jämförelse med näten som var utrustade med pingers. Trots att detta inte är ett tillräckligt stort sampel för att vetenskapligt kunna bevisa någonting, så finns det tydliga indikationer på att Savewave-pingern är effektivare i Österbottens fiskevatten än vad de andra två modellerna är. (Svanbäck, 2012)

Något som också kunde konstateras i testet var att i och med att det inte går att byta ut batteriet i Savewave-sälpingern, så blev kostnaden för att konstant ha sälpingern i nätet orimligt stor, då batteritiden enligt tabellen ovan inte överskrider tre månader. (Svanbäck, 2012)

## 1.2 Syftet med arbetet

Syftet med detta arbete är att ta fram information om hur en fungerande sälping kan tillverkas, samt att den på något sätt skulle gå att ladda upp då strömmen tagit slut, istället för att slängas.



Ett förslag på en krets som kan återskapa de frekvenser som sälars räds ska tas fram. Denna krets ska vara komplett i teorin och kan innehålla både analoga och digitala delar.

Olika förslag till lösningar på batteriproblemet ska tas upp. Även förnybar energi ska behandlas.

### **1.3 Avgränsning av arbetet**

Detta arbete ska ta fram den information som behövs för att skapa en prototyp enligt användarens önskemål. Det ska gå att ställa in vilka frekvenser som ska spelas upp, samt ställa in intervallen mellan dessa frekvenspulser. Även längden på pulserna ska gå att ställa in. Det ska också gå att välja ifall man vill använda en specifik frekvens per puls, eller om man vill använda en chirpsignal (det vill säga en puls med stigande frekvens).

Exempel på högtalare ska tas fram och testas, för att få reda på strömförbrukningen i kretsen. Strömförbrukningen ska hållas så låg som möjligt, då själva projektet går ut på att hålla nere strömförbrukningen och försöka ta fram en produkt som är hållbar även sett ur ett miljöperspektiv.

## 2 Teori

Teoridelen består av tre huvuddelar: vad som påverkar sälar, hur de påverkas samt hur en apparat som påverkar sälarna på önskvärt sätt ska byggas upp. I teoridelen kommer begreppet dB re. 1  $\mu$ Pa att förekomma, vilket är ett sätt att mäta ljudnivå under vattnet. Det som egentligen mäts är trycket; d.v.s. Pascal. Detta skrivs om till den logaritmiska skalan dB re. 1  $\mu$ Pa. Skillnaden från decibelskalan i luft är att den anges i dB re. 20  $\mu$ Pa. (Erbe, 2011)

Formeln för att räkna ut ljudnivån i dB är som följer:

$$20 \cdot \log_{10}\left(\frac{P}{P_{ref}}\right) \text{ där } P_{ref} \text{ i vår skala är } 1 \mu\text{Pa.}$$

Enligt Christine Erbe (Erbe, 2011) är trycknivån för två källor med identisk intensitet 36 dB högre under vatten än vad den är ovanför vattenytan.

### 2.1 Frekvenser som påverkar sälar

För att kunna framställa en sälping med önskvärd effekt krävs det att fiskar inte blir påverkade av det ljud som sälpingern genererar. De flesta fiskarter reagerar dock inte nämnvärt på frekvenser som är högre än 800 Hz (Popper, 1993). Detta har också påvisats av en senare undersökning gjord av Kastelein et al., som säger att 75 % av fiskarterna i Nordsjön inte blir skrämde av frekvenser över 800 Hz, ens vid ljudnivån 180 dB re. 1  $\mu$ Pa. (Kastelein, et al., 2007)

Det finns tre olika sälarter i Östersjön; gråsäl, knubbsäl och vikare. Den vanligaste av dessa är gråsäl med ca 30 000 exemplar. De andra två exemplaren består tillsammans av ca 25 000 exemplar, det vill säga under hälften av totalbeståndet. (WWF, 2015) Eftersom det i Österbotten nästan uteslutande handlar om gråsäl (Andersson & Johansson, 2013), så kommer gråsäl att vara sälarten i fokus för detta arbete.

De tre sälarter vi har i våra vatten har liknande hörselspektrum. Knubbsälen har ett hörselspektrum mellan ca 100 Hz – 100 kHz, gråsälen mellan ca 2 kHz – 150 kHz, och vikaren mellan ca 1 kHz – 90 kHz. (Andersson & Johansson, 2013)

Alla arter har dock som lägst detektionströskel (dB re. 1  $\mu$ Pa) mellan ca 1 – 40 kHz (Andersson & Johansson, 2013), vilket innebär att detta är de frekvenser som sälarna reagerar på vid lägst ljudintensitet. Gråsälens lägsta detektionströskel finns mellan 8 – 40 kHz, då är detektionströskeln lägre än 70 dB re. 1  $\mu$ Pa. (Andersson & Johansson, 2013)

Trots att sälarnas hörselspektrum inte överstiger 150 kHz, så reagerar gråsälarna på frekvenser som överstiger 200 kHz (Hastie et al., 2014), vilket tyder på att sälarna kan känna av ljudsignaler som överstiger deras hörselspektrum.

Det finns också forskning som visar att sälars detektionströskel för frekvenser ökar om ljudpulsen är kortare, det vill säga för en längre puls så krävs det lägre ljudintensitet för att sälarna ska reagera (Kastelein, et al., 2010, Hastie, et al., 2014). Vid frekvensen 40 kHz är skillnaden mellan 5-7 dB re. 1  $\mu$ Pa vid en puls på 0,5 ms respektive 1 ms (Kastelein, et al., 2010).

Redan år 1987 gjordes det tester med chirp-signaler med frekvensen 8-20 kHz med pulslängd mellan 1 och 32 ms av Mate & Harvey. (Mate & Harvey, 1987) Ljudstyrkan vid detta test var 187 dB re. 1  $\mu$ Pa, och detta producerades av en sälskrämma. En sälskrämma är en större version av en sälping, som oftast är stationär vid problemområden som till exempel fiskeodlingar. En sälskrämma kan producera kraftigare ljud än en sälping, och kräver därför en kraftigare strömkälla. Detta test pågick i fyra år, och målet var att hålla knubbsälarna borta från laxodlingar. I tre år fungerade apparaturen bra, men fjärde året var sälnivåerna tillbaka till ursprungsnivån igen.

En motsvarande undersökning gjordes samma år av två andra forskare, Geiger & Jeffries. Deras resultat var mycket annorlunda. Två fiskodlingar hade mer säl inom kontrollområdet medan sälskrämmorna var igång och en fiskodling hade en ökning av säl inom området under en treveckorsperiod, vilket stöder teorin om att sälskrämmor kan fungera som en vällingklocka. (Geiger & Jeffries, 1987)

Tabell 2. Olika sälskrämmor som användes i Götz & Janiks rapport. (Götz & Janik, 2013)

Tillverkare	Modell	Ljudnivå (re 1 $\mu$ Pa)	Frekvensstruktur
Ferranti-Thomson	MK2 Seal Scrammer	194 dB vid 27 kHz	5 frekvenser som är arrangerade i 5 olika sekvenser
Ace-Aquatec	Silent Scrammer	200 dB vid 25 kHz	28 frekvenser i 64 sekvenser
Airmar	Airmar dB Plus	192 dB vid 10 kHz	En enda frekvens i 14 ms långa pulser
Lofitech	Lofitech universal or seal scarer	182 dB vid 14,9 kHz	15 kHz i 250ms pulser
Terecos Ltc	Terecos type DSMS-4	178 dB vid 4,9 kHz	Randomiserade sekvenser med chirpfunktion.

Flera olika sälskrämmor har testats av olika tillverkare och resultatet har alltid varit varierande. Tabell 2 anger specifikationerna för alla sälskrämmor som tas upp i experimenten nedan.

Ace-Aquatec-skrämman har testats vid farmer där potentiella rovdjur är gråsäl och knobbsäl. Två odlingar visade på en reducering av skada på fiskarna, medan den tredje odlingen inte visade någon skillnad alls i sälaktivitet. (Götz & Janik, 2013)

Kastelein et. Al. gjorde också en undersökning med en Ace-Aquatec-skrämma år 2014 (Kastelein, et al., 2014). Sälarnas reaktion då skrämman var igång var bland annat att lyfta upp huvudet över vattenytan för att slippa ljudet.

En Lofitechskrämma användes under en undersökning i Östersjön år 2006 och där kunde en ökning av fångsten noteras. Mindre skada på fiskarna kunde också konstateras sett över en treårsperiod. Det bör dock noteras att effekten alltid var reducerad vid slutet av säsongen varje år, vilket tyder på att sälarna började anpassa sig till skrämman. (Götz & Janik, 2013)

Samma modell av Lofitech-skrämma användes även 2009 i två floder i Skottland under en femmånadersperiod. Där kunde en minskning av säl uppträda från skrämman på ca 50 % noteras. Sälbeståndet i hela flodsystemet bistod dock. (Götz & Janik, 2013)

Dessa tester visar på att resultatet varierar kraftigt från studie till studie. Även fler studier har gjorts med ovanstående skrämmor, och de studierna visar också att resultatet varierat från ingen effekt alls till en viss effekt. (Götz & Janik, 2013)

En undersökning av Götz & Janik från 2010 visar också att olika saker som påverkar sälars beteende är förekomsten av mat, inlärning, ljudtyp samt detektionsnivå. Denna studie visar också att sälarna anpassade sig snabbt till alla ljudtyper från de olika sälskrämmorna då ljudnivån var 146 dB re. 1  $\mu$ Pa. (Götz & Janik, 2010)

Samma undersökning visar också på att om inte mat finns närvarande på kontrollområdet så kunde en avskräckande effekt uppnås med hjälp av ljudnivåer så låga som 135 dB re. 1  $\mu$ Pa. (Götz & Janik, 2010)

Det finns också en annan aspekt att ta i beaktande då det gäller sälskrämmors effekt på sälar. Det finns undersökning som tyder på att sälars hörsel tar skada av ljudnivåer som är i nivå med de som sälskrämmorna i dagens läge ger ut. (Götz & Janik, 2013) Detta kräver dock att sälarna måste hållas inom några hundra meters radie från sälskrämmen, vilket inte är otänkbart eftersom sälskrämmorna placeras ut där det finns mycket mat åt sälarna. Det bör också tas i beaktande att ifall säl får skadad eller nedsatt hörsel påverkar det också funktionaliteten av sälskrämmorna, eftersom en säl som har skadad hörsel blir mindre påverkad av ett ljud som är avsett att skrämma densamme. (Götz & Janik, 2013)

På basis av dessa undersökningar kan det konstateras att det behövs hög amplitud på signaler för att skrämna sälar. Detta beror enligt Dr. Thomas Götz (personlig kontakt, 27.03.2015) på att en specifik reflex i hjärnstammen framkallas, vilket temporärt ger en skrämmande effekt och skrämmer bort sälen. Denna ljudnivå borde då ligga på ca 180 dB re. 1  $\mu$ Pa för att fortfarande ha en skrämmande effekt, men minimera risken för att skada sälens hörsel. För att ännu minimera risken för hörselskador så används korta pulser med långa mellanrum.

Att använda en chirpsignal istället för en konstant frekvens är också ineffektivt enligt Götz. Sälarna anpassar sig väldigt snabbt till det ljudet, och det har ingen mer avskrämmande effekt än vad någon annan ljudbild har. (Personlig kontakt, 27.03.2015)

## 2.2 Uppbyggnad av kretsen

För att skapa de frekvenser som krävs finns det olika sätt att gå till väga. Ett av sätten är att använda en mikrokontroller med en mikroprocessor och det andra sättet är att bygga upp en krets med endast analoga delar. För att förstärka signalen vidare från detta steg till högtalaren behövs ett drivsteg. Den mest lämpade högtalaren för detta ändamål, är en piezohögtalare.

### 2.2.1 Digital frekvensgenerator

För den digitala delen av kretsen bör en mikroprocessor, som är tillräckligt snabb för att klara av de frekvenser som ska genereras, användas. Mikroprocessorn måste ställas in till att göra det som önskas av den med hjälp av att antingen använda olika klockor och register som finns i processorn, eller genom att göra allting med hjälp av kod, det vill säga antingen lösa problemet med hårdvara eller med mjukvara.

Den digitala kretsen ska skapa den av användaren önskade frekvensen, den ska göra det möjligt att ställa in avståndet mellan ljudpulserna, samt ställa in hur långa ljudpulser som önskas.

En för detta ändamål lämplig mikroprocessor är Arduino Due. Det är en mikrokontroller som kan programmeras antingen med Arduinos egna kodspråk, eller med hjälp av Atmels kodspråk, ARM. I denna kontroller sitter en Atmel-processor som heter ATSAM3X8E, som har klockfrekvensen 84 MHz (Arduino, 2015).

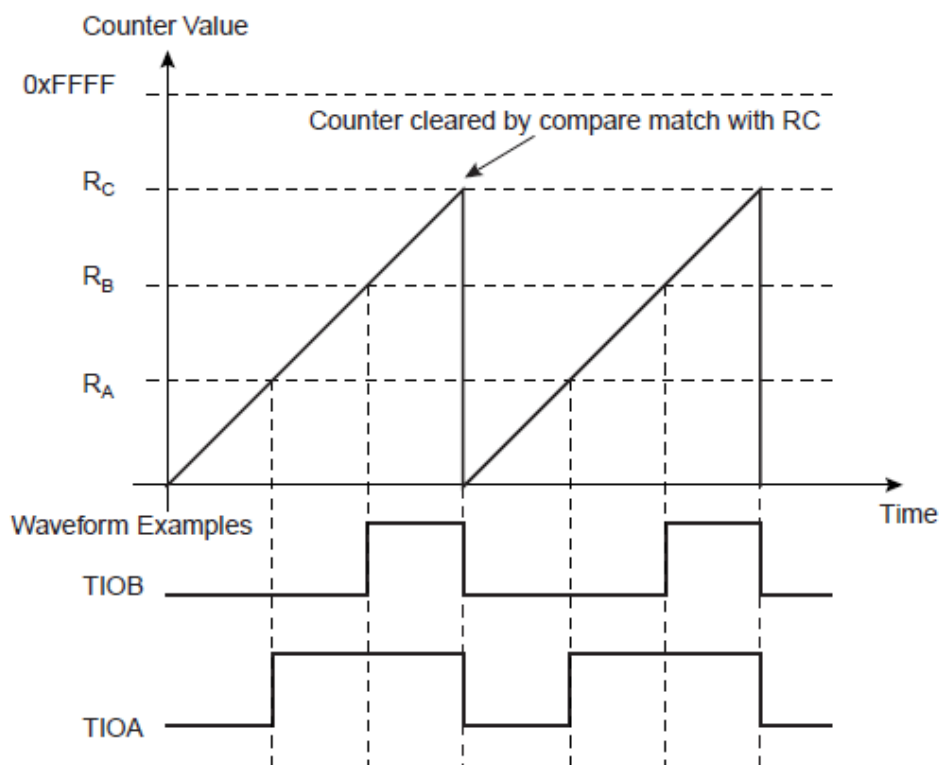
Denna processor innehåller flera olika klockor, vilka kan användas exempelvis till att generera fyrkantsvågor i olika frekvenser. För att använda dessa klockor behöver dock en hel del inställningar göras i koden (Atmel Corporation, 2012).

Först ska den ”Timer clock” som ska användas väljas. Det finns fem olika att välja mellan, `TIMER_CLOCK1` till `TIMER_CLOCK5`. Dessa klockor har alla olika frekvenser, enligt följande tabell:

Tabell 3. Tabell över de timer clocks som finns tillgängliga i ATSAM3X8E (Atmel Corporation, 2012)

Klocknamn	Maxfrekvens (MCK = 84 MHz)
TIMER_CLOCK1	MCK/2
TIMER_CLOCK2	MCK/8
TIMER_CLOCK3	MCK/32
TIMER_CLOCK4	MCK/128
TIMER_CLOCK5	SLCK (Slow Clock)

För att generera en fyrkantsvåg med dessa klockor måste ”Waveform Operating Mode” aktiveras. Detta görs genom att ställa in WAVE-parametern i ett register som heter ”Channel Mode Register”. Vilken version av WAVE-parametern som ska användas ska även beslutas här. Detta görs med hjälp av en WAVSEL-parameter. (Atmel Corporation, 2012) I vanliga fall används en inställning som heter RC\_UP, som fungerar på så vis att klockan räknar upp till det av användaren givna värdet RC, för att sedan tömmas och börja om från början igen. (Atmel Corporation, 2012)



Figur 2. En bild som visar hur RC\_UP fungerar. För att få en jämn fyrkantsvåg bör  $R_B$  vara  $R_C/2$ . (Atmel Corporation, 2012)

Dessa parametrar ställs in med hjälp av ARM-kod på följande sätt:

Vilken klocka som önskas: **TC\_CMR\_TCCLKS\_TIMER\_CLOCKX**

Aktivera WAVE-parametern: **TC\_CMR\_WAVE**

Välja vilken WAVSEL-parameter: **TC\_CMR\_WAVSEL\_UP\_RC** (i detta fall **\_UP\_RC**)

Dessa kodexempel är hämtade från Atmels Software-hemsida. (Atmel Software, 2012)

För att över huvud taget kunna ställa in WAVE-parametern i ”Channel Mode Register” så krävs det dock att nämnda register inte är skrivskyddat. (Atmel Corporation, 2012) Detta åtgärdas med hjälp av följande kodrad: (Atmel Software, 2015)

```
pmc_set_writeprotect(false);
```

För att kunna avbryta processen som körs behöver externa avbrott vara tillåtna (Atmel Corporation, 2012), vilket följande kodrad tillåter: (Atmel Software, 2015)

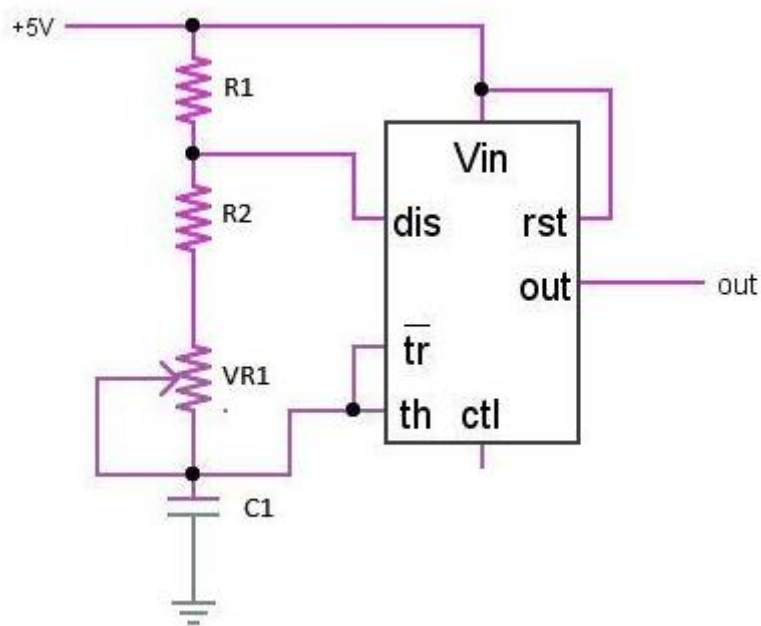
```
pmc_enable_periph_clk(uint32_t ul_id)
```

Dessa inställningar behöver bara göras om frekvensen ska genereras med hjälp av mikroprocessorns inbyggda klockor. Samma sak går också att göra med hjälp av enbart mjukvarukod, det vill säga utsignalen justeras från låg till hög med enkel Arduino-kod.

### 2.2.2 Analog frekvensgenerator

För att ytterligare få ner energiförbrukningen kan också en fyrkantsvåg med önskad frekvens genereras med hjälp av en enkel krets kring en 555-klocka. (Gustavsson, 2011) Ett exempel på hur en krets med en 555-klocka kan se ut följer nedan:





Figur 3. En enkel krets som genererar en fyrkantsvåg som frekvensen enkelt går att ändra på.

Denna krets genererar en fyrkantsvåg på utgången (out). Frekvensen för fyrkantsvågen beräknas med hjälp av följande uträkningar enligt (Gustavsson, 2011):

$$\text{Frekvensen } f = \frac{1}{t_H + t_L}$$

där  $t_H = 0,693 \cdot C1(R1 + R2 + VR1)$  och  $t_L = 0,693 \cdot C1(R2 + VR1)$ .

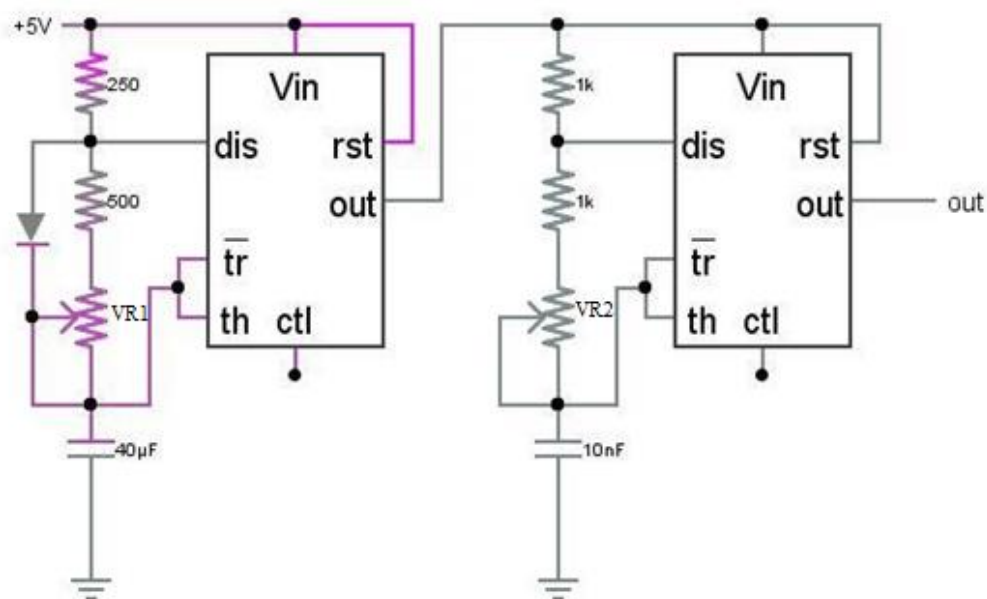
$$\text{Detta ger oss frekvensen } f = \frac{1,443}{((R1+2 \cdot (R2+VR1)) \cdot C1)}.$$

Genom att ändra på potentiometern kan frekvensen ställas in efter önskemål. Exempelvis ger följande värden frekvensen ca 7 kHz:

$$\begin{aligned} R1 &= 1 \text{ k}\Omega \\ R2 &= 1 \text{ k}\Omega \\ VR1 &= 99 \text{ k}\Omega \\ C1 &= 1 \text{ }\mu\text{F} \end{aligned}$$

Om potentiometern istället har värdet 69 k $\Omega$  blir frekvensen 10,2 kHz.

För att få denna frekvens i pulsform, det vill säga istället för att ha en konstant ton, seriekopplas en till likadan krets som inmatning till denna krets. Denna del av kretsen ställs då in på en långsammare frekvens. Den frekvensen justerar då pulslängden, medan den andra delen av kretsen blir frekvensen på ljudsignalen.



Figur 4. Exempel på krets som genererar pulser med frekvenser.

Med VR1 inställt på  $800 \Omega$  och VR2 inställt på  $99 \text{ k}\Omega$ , så blir frekvensen  $717 \text{ Hz}$ . Denna frekvens skickas ut i pulser på  $6,93 \text{ ms}$ . I och med att en diod leder spänningen förbi R2 och VR1, det vill säga  $500 + 800 \Omega$  räknas bort i uppladdningen av kondensatorn, blir pauserna längre än frekvenspulserna. Dioden gör att  $t_H = 0,693 \cdot C1 \cdot R1$ , det vill säga  $6,93 \text{ ms}$ .

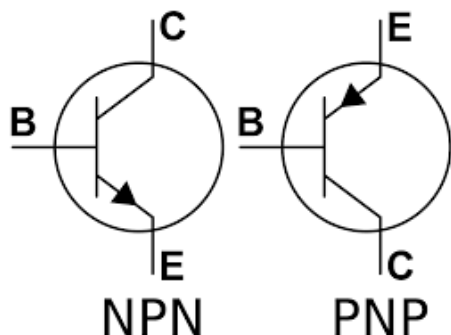
$t_L = 0,693 \cdot C1(R2 + VR1)$  förändras inte på grund av dioden. Detta gör att  $t_L$  blir  $36 \text{ ms}$ .

### 2.3 Drivsteg

Ett drivsteg förstärker strömstyrkan på signalen från frekvensgeneratorn till en lämplig nivå för högtalaren. Hur stor förstärkning som behövs beror av två saker, hur stor strömstyrka frekvensgeneratorn klarar av att generera, samt hur stor strömstyrka högtalaren behöver ha för att fungera korrekt. Strömförstärkningen som behövs är unik från fall till fall, vilket gör att det finns olika sätt att gå till väga för att förstärka strömsignalen.

Det vanligaste sättet att förstärka strömsignalen är att använda en bipolär transistor, det vill säga en NPN- eller PNP-transistor. Den funktionella skillnaden mellan en NPN-transistor

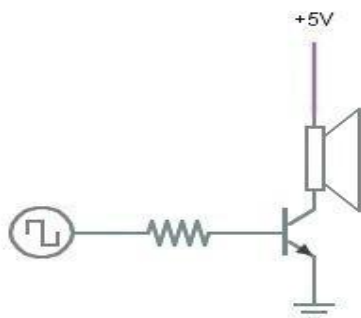
och en PNP-transistor är vilken väg strömmen går i dem. I en NPN-transistor går strömmen från kollektorn till emittorn, medan den går från emittorn till kollektorn i en PNP-transistor. (Storr, 2015) Spänningsfallet  $V_{BE}$  är ca 0,7 V hos en bipolär transistor, och spänningsfallet  $V_{CE}$  är vid maxförstärkning 0,3 V. En bipolär transistor leder inte om ingen ström finns i basen, utan är strypt så länge basen är 0. (Gustavsson, 2011)



Figur 5. NPN- och PNP-transistor. Basen, kollektorn samt emittorn utmarkerade på de båda.

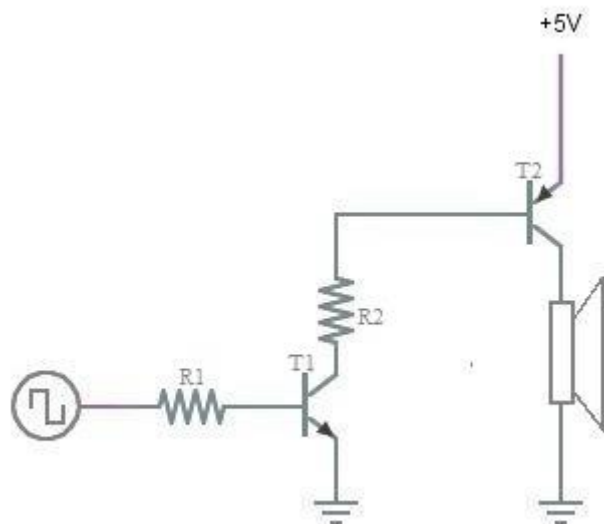
För att förstärka en strömsignal med hjälp av en NPN-transistor är transistorns strömförstärkningsfaktor,  $\beta$ , avgörande. Strömförstärkningsfaktorn finns alltid given i en given transistors datablad och kan variera kraftigt till och med mellan likadana komponenter. Strömförstärkningen är ofta angiven som ett spektrum mellan två värden, till exempel som 200...400. Då detta är fallet måste det lägsta värdet antagas vara transistorns strömförstärkning, för att undvika risken att underdimensionera drivsteget.

Strömförstärkningen fungerar på så sätt att signalen som önskas förstärkas kopplas till basen på transistorn. Denna signal förstärks sedan minst 200 gånger på kollektorn. Högtalaren kopplas därför in på kollektorn, med en extern spänningskälla kopplad till det andra benet på högtalaren. Detta leder till att högtalaren får en lämplig strömnivå på signalen som frekvensgeneratoren ger ut.



Figur 6. Förstärkning av strömsignal med hjälp av en NPN-transistor.

Om större förstärkning än vad en ensam bipolär transistor klarar av krävs går det att koppla ihop en NPN-transistor och en PNP-transistor på så sätt att strömförstärkningsfaktorn blir faktorn av båda transistorers enskilda strömförstärkningar, det vill säga  $\beta_1 \cdot \beta_2$ .



Figur 7. Krets som ger strömförstärkningsfaktorn  $\beta_{T1} \cdot \beta_{T2}$ .

Högtalaren i detta fall får spänningen 4,7 V, i och med att  $V_{CE} = 0,3$  V. Strömstyrkan på T2-kollektorn,  $I_{C2}$ , är den strömstyrka som högtalaren behöver. Denna strömstyrka divideras med strömförstärkningen  $\beta_{T2}$ , för att få strömnivån på basen,  $I_{B2}$ .  $I_{B2}$  är också  $I_{C1}$ , vilket gör att  $I_{B1}$  erhålles genom att dividera  $I_{C1}$  med  $\beta_{T1}$ .

$$I_{C2} = I_{B2} \cdot \beta_{T2}$$

$$I_{B2} = I_{C1}$$

$$I_{C1} = I_{B1} \cdot \beta_{T1}$$

Detta ger följande uträkning:  $I_{C2} = I_{B1} \cdot \beta_{T1} \cdot \beta_{T2}$

Strömförstärkningen i denna krets blir med andra ord  $\beta_{T1} \cdot \beta_{T2}$ .

Dimensionen på motstånden räknas ut genom att ta i beaktande spänningsfallen

$V_{BE} = 0,7$  V samt  $V_{CE} = 0,3$  V på båda transistorer, samt ovan uträknade strömnivåer.

## 2.4 Högtalare

En högtalartyp som är väl lämpad för detta ändamål är en piezohögtalare. Först och främst är de mindre och strömsnålare vid högre frekvenser än vad konventionella högtalare är, vilket är att föredra i detta projekt eftersom låg strömförbrukning är ett av målen i projektet. (Sonitron, 2015)

En piezoelektrisk högtalare fungerar så att en kristall expanderar och komprimerar då den får en elektronisk puls. Detta material fästs på en metallplatta och då dessa två tillsammans får en signal med en viss frekvens så genereras ett ljud. (Jain, 2012)

Piezoelektriska högtalare används ofta som undervattenhögtalare i och med att de är solida, det vill säga de är byggda endast av solitt material, de innehåller ingen luft eller gas. Detta gör att de fungerar bra under tryck. (Normandin, 2015)

## 2.5 Strömförsörjning till kretsen

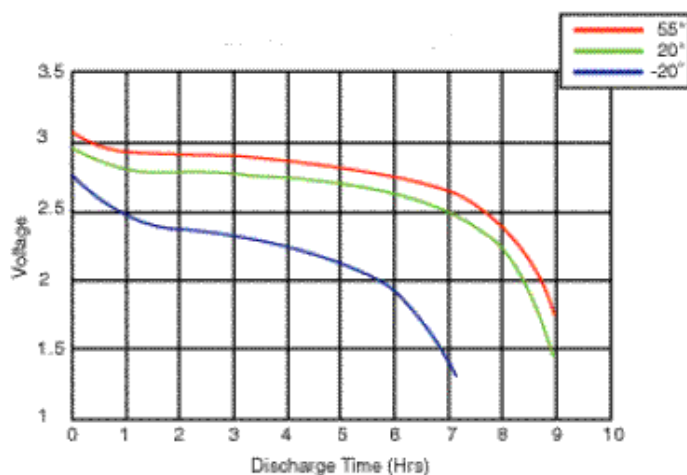
För att denna krets ska kunna fungera behövs en strömkälla. Denna strömkälla behöver vara utbyttbar, laddningsbar eller på något sätt kunna ta tillvara på energi från omgivningen för att produkten ska vara hållbar.

### 2.5.1 Laddningsbart batteri

Som exempel på laddningsbara batterier finns flera alternativ.

Litium-jonbatterier har en hög energi-densitet, det vill säga mycket energi på en liten yta, och de kräver lite underhåll. En nackdel som dock kan vara ganska avgörande för batterier i detta projekt är att då de blir nedkylda reduceras livslängden på dem. Den ideala arbetstemperaturen för denna batterityp är mellan 15 – 50 °C. Då arbetstemperaturen sjunker under 15 °C börjar antalet uppladdningar batteriet klarar av att sjunka progressivt. (Electropaedia, 2005) Dock klarar batteritypen ca 1000 uppladdningar om arbetstemperaturen är så låg som 0 °C, vilket är den absolut lägsta arbetstemperatur som

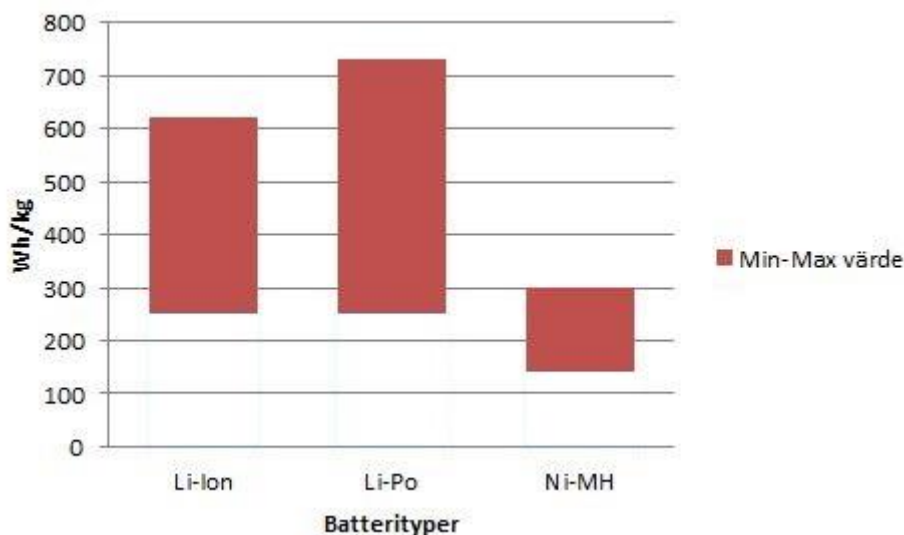
kommer att förekomma i detta projekt. Urladdningstiden ökar också då batteriet har en lägre arbetstemperatur, vilket illustreras i figur 8. En annan nackdel är att det behövs en skyddskrets för att bibehålla säkra nivåer på ström och spänning. (Battery University, 2010)



Figur 8. Urladdningstid på x-axeln, voltnivå på y-axeln, för ett Li-Io-batteri. (Electropaedia, 2005)

En annan variant som finns är litium-polymerbatterier. Dessa batterier har också en hög energidensitet, det vill säga den maximala densiteten är högre än litium-jonbatteriers i dagens läge. De är också väldigt flexibla i form, det vill säga de har inte standardiserade cell-former. Detta gör att tillverkare har mer frihet då de tillverkar batterier för specifika ändamål. Dessa batterier är dock dyra att tillverka och klarar av färre uppladdningar än litium-jonbatterier. (Battery University, 2010)

En tredje laddningsbar variant är ett nickel-metallhydridbatteri. Till skillnad från de tidigare varianterna så är detta batteri tillgängligt i de vanliga storlekarna AA och AAA. Denna variant av batteri har dock lägre specifik energi än de två föregående varianterna och så har det en högre självurladdning. Fördelarna med detta batteri är dock att det är billigare och säkrare. Detta batteri ska också förvaras kallt för att hålla prestandan, vilket är bra i detta experiment. Dessa fördelar gör att detta alternativ inte helt kan förbises. Detta batteri anses också vara mer miljövänligt, i och med att det bara innehåller svaga gifter. (Battery University, 2011)



Figur 9. Energidensitet för de tre alternativen. (Battery University, 2010)

### 2.5.2 Utbytbart batteri

Av de batterier som inte är uppladdningsbara (primärbatterier) finns två eventuella alternativ: alkaliska batterier och litiumbatterier. Dessa batterier finns bland annat i de vanliga storlekarna AA, AAA samt som 9 V batteri. (Battery University, 2015)

Alkaliska batterier har ca 50 % högre energi-densitet än de bästa litium-jonbatterierna. Problemet med alkaliska batterier är dock att de endast fungerar effektivt då de ger ut väldigt låga strömmar och effekter. Om alkaliska batterier används med högre ström och spänning så är prestandan väldigt dålig (Battery University, 2015), vilket gör detta batteri kan avskrivas för detta projekt.

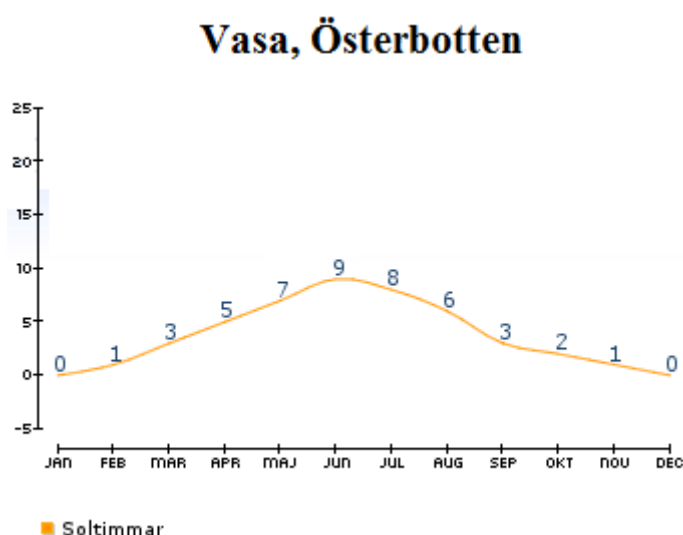
Primära litiumbatterier fungerar även då högre strömförbrukning krävs, vilket gör att det eventuellt skulle gå att använda dem i detta projekt. De håller längre än en laddning på ett litium-jon-batteri. Livslängden är dock inte ens i närheten av tillräckligt mycket längre för att det ska vara försvarbart kostnadsmässigt att använda dem istället för uppladdningsbara batterier. (Battery University, 2015)

Med tanke på att detta projekt också ska ta i beaktande miljöaspekten är det inte heller ur den synvinkeln försvarbart att använda primära batterier, även om de går att byta ut.

### 2.5.3 Förnybar energi

Det optimala vore att använda uppladdningsbara batterier och att delvis kunna ta tillvara energin som finns i omgivningen.

Den energiform som skulle kunna tas tillvara är solenergi, ifall pingern är monterad på ett nät som flyter vid ytan. I Vasa är medeltalet soltimmar 3,75 per dag sett över hela året. (Mitt Resväder, 2015) En solpanel skulle vara tämligen ineffektiv mellan november och februari i och med att medeltalet soltimmar per dygn då ligger på ca 0,5.



Figur 10. Graf som visar antalet soltimmar i medeltal per dag i Vasa. (Mitt Resväder, 2015)

En solcells effekt defineras med en enhet som heter Watt-peak,  $W_p$ . En  $W_p$  är den effekt en solpanel genererar då solstrålningen är  $1000 \text{ W/m}^2$ . (Naps Solel, 2015) Detta värde blir i Mellersta Finland  $0,9 \cdot W_p$ , i och med att solstrålningen ligger på ca  $900 \text{ W/m}^2$ . (Vasek, 2012) Detta värde räknas sedan om till  $Wh$  genom att multiplicera solpanelens effekt med antalet timmar solen lyser. Exempelvis en solcell på  $30 \text{ W}_p$  genererar en julidag i Vasa  $(30 \cdot 0,9) \text{ W} \cdot 8 \text{ h} = 216 \text{ Wh}$ . Detta förutsatt att den inte är i skuggläge någon tid under dagen samt att alla förhållanden för solcellen är optimala, dvs ljusets infallsvinkel och dylikt.



Jan	Feb	Mar	Apr	Maj	Jun	Jul	Aug	Sep	Okt	Nov	Dec	Enhet	Summa maj- aug	Summa helår
3	19	54	96	132	143	126	91	53	23	6	1	Wh	11 kWh	23kWh
0.3	1.6	4.5	8	11	11.9	10.5	7.6	4.5	2	0.5	0.1	Ah (12V)	1260Ah	1901Ah

Figur 11. Antalet Wh samt Ah som produceras av en 30 W solcell per dag på varje enskild månad, samt totala mängden under sommarmånaderna och över hela året. (Solarlab, u.d.)

Figur 11 visar hur effektiv en 30 W solcell kan vara under sommarmånaderna, samt hur stor skillnaden är mellan sommarmånaderna och de andra årstiderna.

Det finns tre olika slags solceller. Mono-kristallinkisel, poly-kristallinkisel samt tunnfilmssolceller. De två förstnämnda är de vanligaste alternativen (ca 95 % av alla solceller), på grund av att tunnfilmssolceller har lägre effektivitet i relation till ytstorleken. (Naps Solel, 2015)

En sak som bör tas i beaktande är vilken vinkel solcellen bör ha för optimal effektivitet. Under sommaren lönar det sig att ha solcellen i en vinkel på ca 15°, medan den under vintern bör ligga på ca 75°. (Importören Outlet, 2015) Om inte detta tas i beaktande, utan låter solcellen ligga i samma vinkel året om, erhålls inte den angivna effekten på solcellen (Wp). Detta värde sjunker, vilket leder till att solcellerna blir ineffektivare. (Solarlab, u.d.)

En annan sak att ta i beaktande om solceller används för att ladda upp batteriet är att i och med att det laddas hela tiden så kan batteriets laddningscyklar, det vill säga antalet uppladdningar batteriet klarar av, snabbt förbrukas, vilket kan förkorta livslängden på batteriet.

En solcell kan vara användbar i detta projekt. I och med att elektroniken som tagits upp inte förbrukar mycket ström bör en solcell räcka till som strömförsörjning. En nackdel med detta är att allting måste tätas noggrant så att inget vatten kommer åt kretsen och kortsluter den. Det krävs också en direkt sladdkontakt mellan solcellen som bör finnas på ytan och sälpingern som finns i nätet under vattenytan. En anordning som ser till att solcellen hålls ovanför vattenytan krävs också. För att få ut maximal effekt ur solcellen bör den även vara vinklad korrekt mot solen.

### 3 Förverkligande av sälpingern

Nedan följer alla steg som gjorts för att ta fram delarna som behövs till sälpingern, samt vilken utrustning som har använts.

#### 3.1 Programmering av mikroprocessorn

Som tidigare nämnts, införskaffades en Arduino Due-mikrokontroller för att generera de frekvenser som önskas till detta projekt. För att dessa frekvenser skall genereras, krävs det antingen att alla register, klockor och avbrott ställs in manuellt, eller att Arduinos ARM-kod används. Nedan följer två kodexempel, ett där alla klockor, register och avbrott ställs in manuellt, och ett som använder Arduinos ARM-kod för att generera en fyrkantsvåg med önskad frekvens. Detta görs för att se om energiförbrukningen skiljer beroende på hur dessa används.



*Figur 12. Arduino Due-mikrokontrollern. Processorn Atmel ATSAM3X8E syns i mitten. Alla ingånar och utgångar syns uppradade längs kanterna.*

### 3.1.1 Generering av pulser av olika frekvenser med hjälp av klockor

En kod som skall generera önskade frekvenser med önskade intervall och pulslängder tas fram. Denna kod bifogas i bilaga 1.

Det som önskas av koden rent praktiskt är en utsignal med önskad frekvens. Klockan i mikroprocessorn ska slå om från hög till låg enligt den inställda frekvensen och på så sätt skapa en fyrkantsvåg.

För att bygga upp denna kod krävdes först noggrann planering av vad koden rent praktiskt skulle åstadkomma. Utsignalen som önskas är en av användaren vald frekvens. Den valda klockan behöver slå om från hög till låg och vice versa vid rätt tidpunkt för att den önskade frekvensen ska genereras. För att detta ska göras korrekt måste det av användaren inmatade värdet för frekvensen konverteras så att processorn förstår vad den ska göra och frekvensen blir rätt inställd. Om frekvensen bara matas in utan att översättas åt processorn blir frekvensen inte korrekt. För att få rätt frekvens till processorn måste klockans grundfrekvens tas i beaktande. För att få korrekt frekvens måste detta värde halveras, eftersom en svängning består av två klockcykler, en hög och en låg del. Detta värde fås från tabell 3. I detta fall är grundfrekvensen  $\frac{84 \text{ MHz}}{128 \cdot 2}$ . Genom att sedan dividera detta värde med det inmatade frekvensvärdet erhålls det korrekta siffervärdet som processorn behöver för att kunna generera rätt frekvens.

Samma sak måste även göras med längden på pulsen. För att detta ska fungera, måste värdet ställas in så att det blir begripligt för processorn och resten av koden. Detta nya värde måste även ta i beaktande vad den valda frekvensen är, för att veta hur många höga och låga signaler som ska skickas ut. Detta görs genom följande formel:

$$\text{Nytt värde} = 2 \cdot \text{frekvens} \cdot \frac{\text{pulslängd}}{1000}$$

För att få rätt värde på det nya värdet, dubblas först frekvensen, för att ta i beaktande både den höga och den låga delen av en kurva i frekvensen. Därefter multipliceras detta med det valda värdet för pulslängd. Detta värde matas in som millisekunder, därav divisionen med 1000.

Exempeluträkning:

Frekvens = 100

Pulslängd = 100

$$\text{Nytt värde} = 2 \cdot 100 \cdot \frac{100}{1000} = 20$$

Detta betyder alltså att timern ska gå hög och låg 10 gånger för att pulslängden ska bli 100 millisekunder. Detta ger frekvensen 100 Hz. Då signalen har slagit om 20 gånger har 10 kurvor genererats, vilket ger pulslängden 0,1 sekunder.

För att generera utsignalen för detta används en enkel if-loop, som säger att utgången ska byta läge varje gång nya värdet räknas ner. Detta ger 20 ändringar på utgången, vilket ger korrekt antal svängningar.

```
digitalWrite(outputPin, pinState= !pinState); //Ändrar värdet mellan 0 och 1 på outputPin
  if (nytt_värde > 0) nytt_värde--; //Räknar ner nytt_värde till 0
```

### 3.2.2 Generering av en chirpsignal med hjälp av mjukvarukodning

För att generera en chirp-signal behövs en annan kod. För att jämföra strömförbrukningen mellan standardinställningar på hårdvaran kontra specifikt valda inställningar på hårdvaran användes standardinställningarna för denna kod. Start- och stoppfrekvens går att ställa in i denna kod, samt längden på pulserna. Koden som genererar detta finns bifogad i bilaga 2.

Denna kod ska byggas upp lite annorlunda än den tidigare. Här är tanken att frekvensen ska öka från en startfrekvens till en slutfrekvens, under tiden som valts för pulsens längd. Detta ger lite mer matematiska uträkningar än vad det förra exemplet hade. Frekvensen ökar hela tiden, vilket leder till att för att få pulslängden korrekt behöver både start- och stoppfrekvensen tas i beaktande då antalet svängningar per puls räknas ut.

I och med att denna kod endast använder funktioner som finns inom Arduinos egna kodsysteem så används kommandot `delayMicroseconds()`. Detta kommando lägger in en fördröjning på det valda antalet mikrosekunder mellan ändringen från hög signal till låg för att generera fyrkantssvågen. Detta, i kombination med att frekvensen ska öka linjärt, ger lite annorlunda uträkningar än då endast en frekvens används. Medelfrekvensen av start- och stoppfrekvensen måste räknas ut för att kunna bestämma pulslängden. I och med att

frekvensen ökar linjärt ger detta medelvärde ett korrekt värde för signallängden, enligt uträkningarna nedan.

Exempeluträkning:

Startfrekvens = 2000 Hz

Slutfrekvens = 6000 Hz

Medelfrekvens =  $\frac{2000+6000}{2}$  Hz

Pulslängd = 250 ms

Följande ekvation är den som används för att bestämma pulslängden:

$$duration = \frac{Pulslängd}{1000} \cdot Medelfrekvens$$

Detta värde används sedan som ett villkor i en while-loop, som räknar ner från detta värde till noll för att bestämma antalet svängningar i signalpulsen.

I detta exempel blir  $duration = \frac{250}{1000} \cdot 4000 = 1000$ . Detta betyder att för att få rätt pulslängd så behövs det 1000 svängningar. Då ska med andra ord while-loopen köras 1000 gånger.

För att sedan korrekt få ökningen av frekvensen, måste detta värde tas i beaktande i ytterligare uträkningar. Frekvensen ska öka från startfrekvensen till slutfrekvensen under ett antal svängningar. För att få en korrekt ökning av frekvensen behöver detta värde divideras med skillnaden mellan frekvenserna.

Exempeluträkning:

Frekvensskillnad = 6000-2000 Hz = 4000 Hz

duration = 1000

$$Frekvensökning = \frac{Frekvensskillnad}{duration} = \frac{4000 \text{ Hz}}{1000 \text{ svängningar}} = 4 \frac{\text{Hz}}{\text{svängning}}$$

Om frekvensen ökar med 4 Hz per svängning så blir det 1000 svängningar innan slutfrekvensen har nåtts. Startfrekvensen ökar således till slutfrekvensen under tiden pulslängd.

### 3.2 Analys av frekvenserna genererad av Arduinokoderna

Alla frekvenssampler har med hjälp av ett Picoscope-oscilloskop lästs in till en datafil, som sedan laddades upp till Matlab. Där utfördes en Fouriertransform på ett godtyckligt sampel av valfri signal och resultatet av dessa Fouriertransformer syns i graferna.

Den första signalen som analyserades var chirpsignalen, som genereras endast med Arduinos egna mjukvarukod. Ett sampel togs direkt i början av ljudkurvan, samt ett sampel precis i slutet av ljudet. Detta för att kunna analysera frekvensen i början av chirpsignalen, samt frekvensen i slutet. De värden som användes var följande:

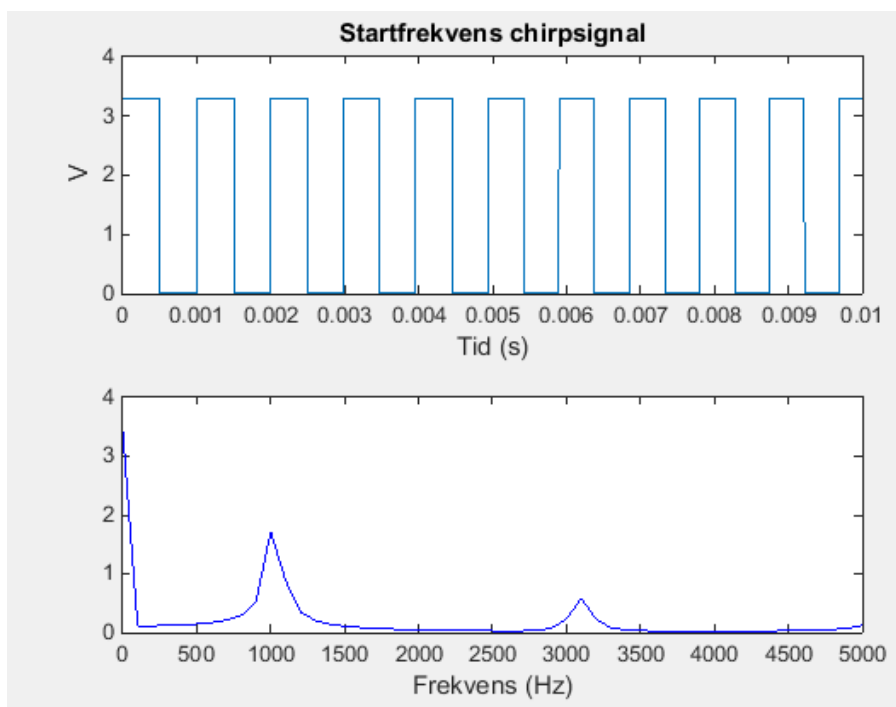
Startfrekvens: 1000 Hz

Slutfrekvens: 2000 Hz

Signallängd: 300 ms

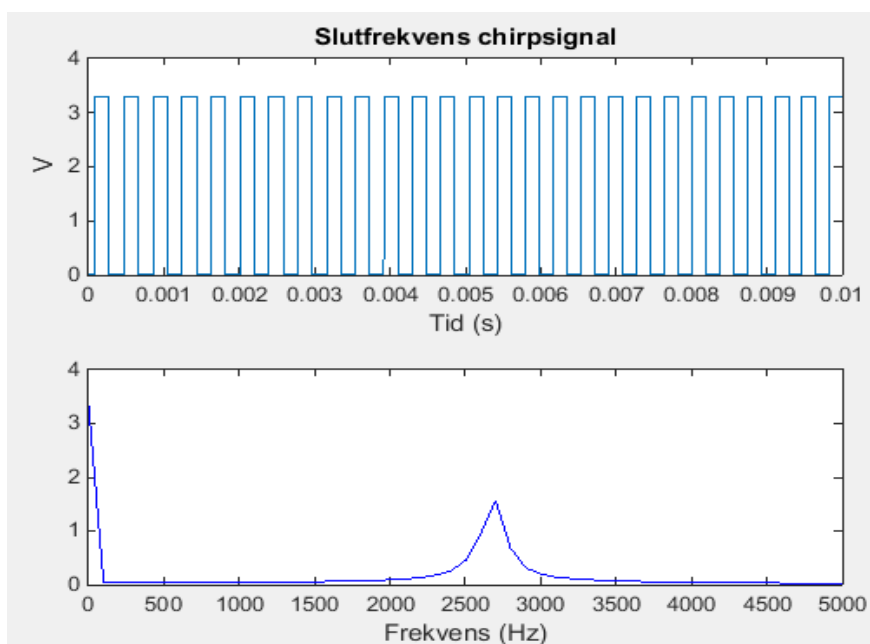
Paus mellan signalerna: 300 ms

I och med att Arduino Due inte kunde hantera flyttal, dvs decimaltal, i de kodkommandon som var nödvändiga så blev detta resultat varken exakt eller särskilt tillförlitligt. Denna kod går inte att använda på höga frekvenser i och med att minsta förändringen på frekvensen som kan göras är en mikrosekund, vilket gör att vid 20 kHz så blir den minsta förändringen ca 3500 Hz. Detta gör dock inte koden oanvändbar, eftersom den är uppbyggd så att den med några små förändringar bör fungera med en mikrokontroller som kan använda flyttal med motsvarande kommandon.



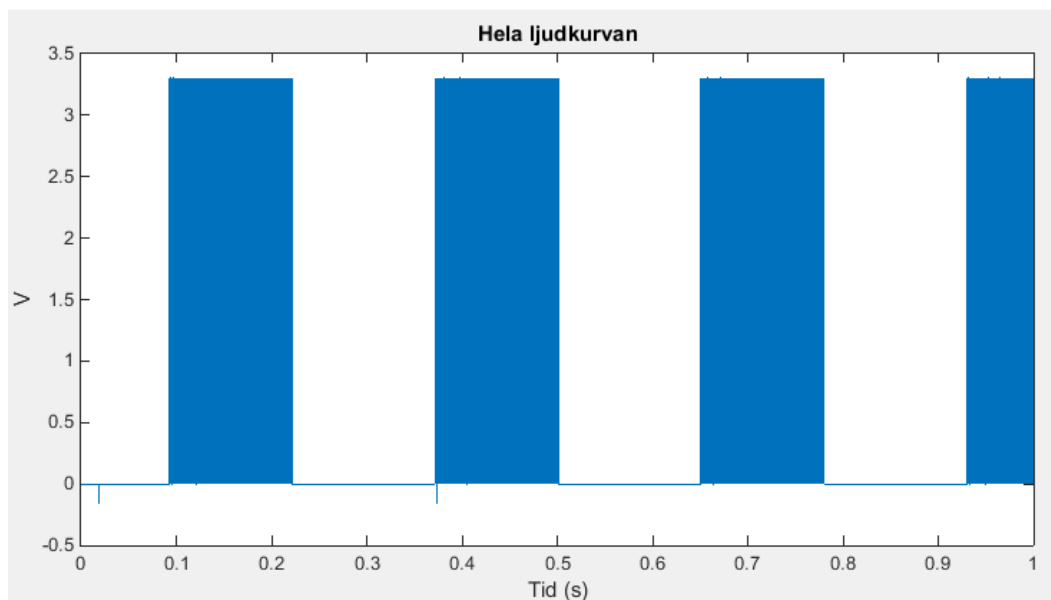
Figur 13. Startfrekvensen på chirpsignalen överst, samt resultatet av Fouriertransformen undertill.

Kurvan undertill i figur 13 visar att frekvensen på detta sampel är 1 kHz, vilket är helt enligt inställningarna. Det beror på att detta värde är hämtat rakt från ett inmatat värde, utan några uträkningar som senare leder till felaktiga värden.



Figur 14. Slutfrekvensen på chirpsignalen överst, samt resultatet av Fouriertransformen undertill.

Figur 14 visar att i och med att det inte går att göra tillräckligt exakta ökning av frekvensen så blir slutfrekvensen i detta test högre än önskat. Ökningen av stegen blir större än vad det borde bli, vilket leder till att den totala ökningen av frekvensen blir för stor.



Figur 15. Tre hela ljudkurvor genererade av chirpsignal-koden.

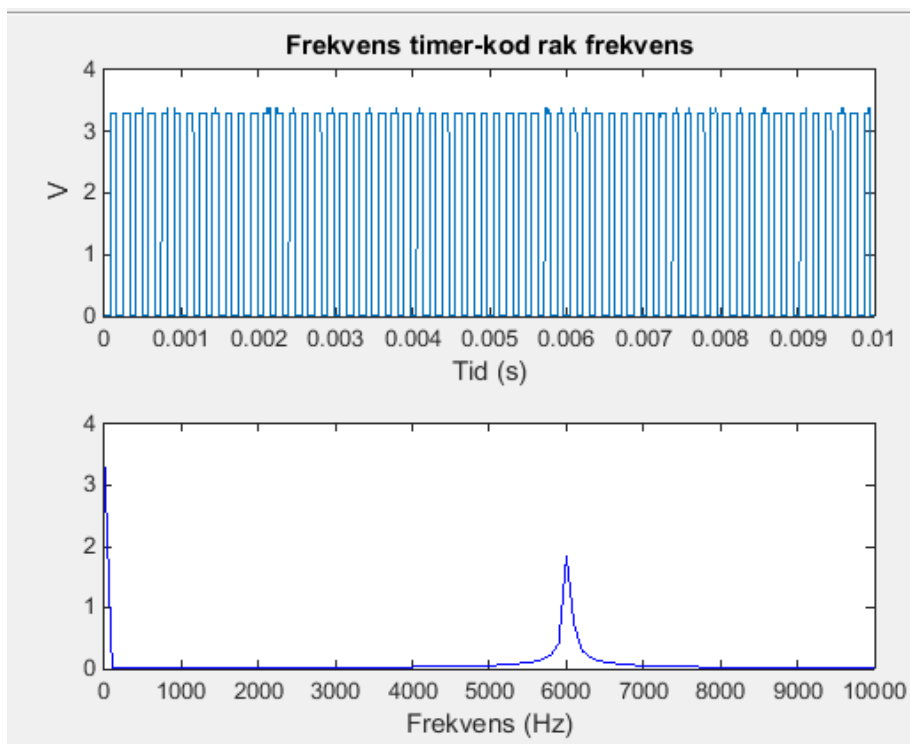
Figur 15 visar att periodlängden inte heller stämmer. I och med att frekvenserna inte blir korrekta blir periodlängderna påverkade då de är direkt beroende av frekvenserna.

Den andra koden, det vill säga frekvensen som genereras med hjälp av egna inställningar av klockor, blev desto exaktare. I och med att alla klockor ställdes in manuellt och enligt egna önskemål, och i och med att periodlängden inte är beroende av ett medeltal av många olika frekvenser, så blev detta resultat helt enligt planerna. Följande värden användes i denna kod:

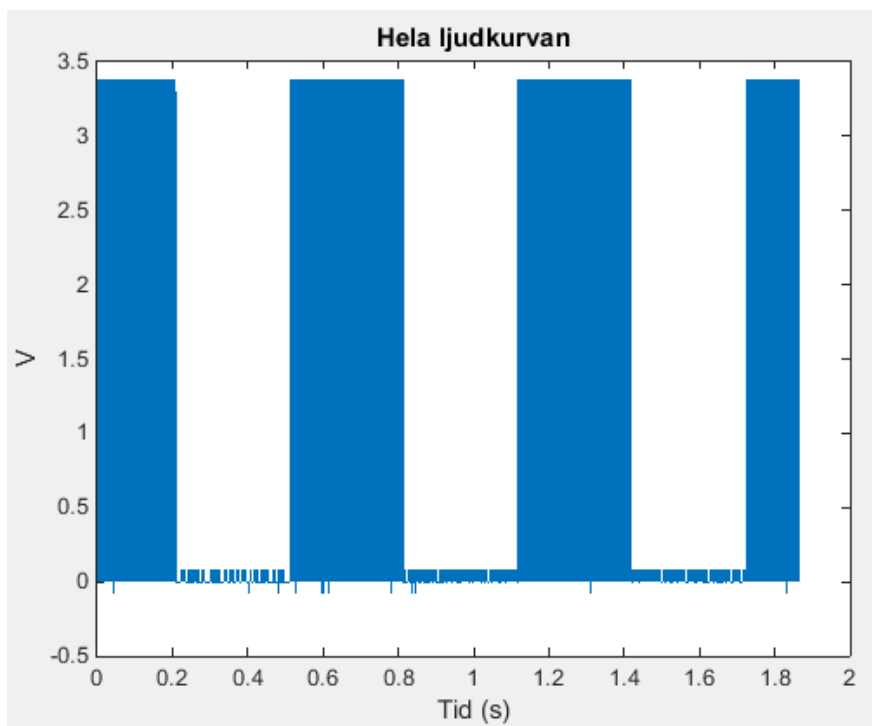
Frekvens: 6 kHz  
 Signallängd: 300 ms  
 Paus mellan signalerna: 300 ms

Resultatet från detta test blev tämligen exakt. Fouriertransformen på ett sampel av ljudsignalen gav ett resultat på 6 kHz, vilket är exakt den frekvensen som matades in som input. Koden som använts för att skapa dessa grafer samt utföra Fouriertransformerna bifogas i bilaga 3.





Figur 16. Frekvensen som genererats med hjälp av manuellt inställd hårdvara överst, samt resultatet av Fouriertransformen undertill. Kurvan undertill visar att frekvensen på detta sampel är 6 kHz.



Figur 17. Hela ljudkurvan. Ljudpulserna är 300 ms långa, och pauserna mellan är 300 ms långa, precis enligt inställningarna.

### 3.3 Analys av strömförbrukning för de olika frekvensgeneratorerna

Strömförbrukningen av alla tre frekvensgenererare som har tagits fram i detta examensarbete har testats. Spänningsmatningen på samtliga alternativ har legat på 5 V. Detta värde valdes i och med att Arduino Due får sin spänningsmatning via en USB-kabel med matningen 5 V.

Den analoga kretsen som består av två stycken 555-timers och några få komponenter drog som väntat minst ström av alternativen. Den lösningen krävde endast  $6,97 \pm 0,3$  mA.

De båda kodalternativen, koden med Arduinos egna kodsystém samt koden med egna hårdvaruinställningar, drog mer eller mindre identiskt mycket ström. Koden med timerlösning drog  $119,6 \pm 0,5$  mA, och koden baserad på Arduinos egna kodsystém drog  $120,1 \pm 0,5$  mA. Detta tyder på att det inte spelar någon roll för strömförbrukningen om alla inställningar görs manuellt eller om Arduinos egna koder och inställningar används. Detta beror sannolikt på att trots att hårdvaran ställs in enligt egna önskemål och värden, så är Arduinos kodspråk uppbyggt på det viset att en loop körs konstant. Detta gör att mikroprocessorn måste arbeta hela tiden trots att endast den önskade hårdvaran är aktiverad. Rådatan till dessa värden bifogas i Bilaga 4.

### 3.4 Exempel på högtalare

Tillverkare av piezohögtalare fokuserar i princip uteslutande på att tillverka produkter inom det hörbara frekvensspektrat. Efter att ha kontaktat Sonitron, ett företag som har specialiserat sig inom piezohögtalare, har det dock konstaterats att trots att högtalarna är listade inom exempelvis spektrumet 100 – 20 000 Hz, så betyder det inte att de inte fungerar vid en högre frekvens. Det betyder dock att tillverkaren inte kan garantera att produkten fungerar vid en högre frekvens. Med detta i åtanke, har Sonitron en högtalarmodell som kunde vara intressant till detta projekt, nämligen SPS-piezohögtalare.

SPS-högtalare är väldigt platta i sin konstruktion, samt totalt vattentäta och dammtäta. Detta gör att de lätt kan kapslas in och fortfarande fungera som de ska göra. Vad som är

mest intressant i detta projekt då högtalaren väljs är vilken spänningsmatning den behöver, samt vad den ger ut för ljudnivå. Eftersom den enda högtalaren som opererar vid lägre spänningsnivåer är SPS-2220-03, så är det den som granskas noggrannare. Detta är också den högtalaren som har högst ljudnivå och är minst till storleken, vilket är fördelaktigt.

Model	Operating voltage (Vac pp)	Frequency range (Hz)	Max. SPL dB @ 1m, average 4 points	Applications	Capacitive load	Dimensions mm (LxWxH)
SPS-2220-03	1-24	700-20000	85*	GSM, GPS, MP3, camera,...	70nF	22x20x2
SPS-3530-03	5-60	700-20000	81	PDA, GPS, notebooks,...	220nF	43.5x39x2
SPS-4640-03	5-60	400-20000	83	PDA, GPS, notebooks,...	225nF	50x43.6x2
SPS-6555-03	5-60	300-20000	83	Multimedia,...	480nF	65x55x2
SPS-8770-03	5-60	200-20000	84	Computers,...	580nF	87x70x2

Figur 18. Spänningsnivå, frekvensspektrum samt maximal ljudstyrka för de olika högtalarna. (Sonitron, 2014)

Model	SPS-2220-03
Capacitance ( $\pm 20\%$ )	70 nF
Z @ 100 Hz	22360 ohm
Z @ 1 kHz	2162 ohm
Z @ 2 kHz	1183 ohm
Z @ 5 kHz	497 ohm
Z @ 10 kHz	245 ohm
Z @ 15 kHz	168 ohm
Z @ 20 kHz	125 ohm

Figur 19. Tabell över impedansen samt kapacitansen för Sonitron SPS-2220-03. (Sonitron, 2014)

Strömförbrukningen för denna högtalare ökar ju högre frekvens som används då impedansen sjunker. Exempelvis vid frekvensen 5 kHz ligger impedansen på 497  $\Omega$ . Med spänningsmatningen 5 V så krävs en strömstyrka på 10 mA från frekvenssignalen för att högtalaren ska fungera. Vid 2 kHz blir motsvarande värde 4,2 mA.

### 3.5 Val av drivsteg

Processorn i Arduino Due ger på sina digitala utgångar en 2 V signal, med maximal strömstyrka på 2,4 mA enligt Arduinos specifikationsblad. Den lägsta impedansnivån hos Sonitron SPS-2220-03-högtalaren är 125  $\Omega$  vid 20 kHz. Om högtalaren matas med 5 V, och spänningsfallet  $V_{CE} = 0,3$  V hos en bipolär transistor erhålls strömnivån enligt ohms

lag  $U = R \cdot I$ . Strömnivån för högtalaren blir således  $\frac{4,7 V}{125 \Omega} = 0,0376 A = 37,6 mA$ . Denna strömnivå är så låg att det räcker med förstärkningen hos en ensam NPN-transistor för att förstärka signalen från frekvensgeneratoren. NPN-transistorn BC547B har strömförstärkning från  $\beta = 200$ . Detta betyder att processorn endast behöver ge ut en strömstyrka på 0,19 mA, vilket är långt under processorns högsta värde. Detta gör att processorn inte behöver pressas nära sin maxnivå utan kan arbeta på en lämplig nivå.

## 4 Resultat

Huvudsyftet med detta arbete var att ta fram information om vilka frekvenser som skrämmar sälar, samt att ta fram information till hur man tillverkar en anordning som producerar just detta. Faktum är att det fortfarande är tvetydigt vad som verkligen är effektivt i ljudväg som sälkskrämmor. Forskningar är motstridiga och sälarna är väldigt bra på att anpassa sig. Finns det mat i närheten är sälarna svårskrämda. Det är möjligt att skrämma sälar med tillräckligt höga ljud, men för att producera dessa ljud krävs det en anordning som är större än en sälping. Strömförbrukningen för de sälkskrämmor som har effekt på sälar är avsevärt högre än strömförbrukningen på de förslag som har tagits upp i detta arbete.

Det har tagits fram tre olika frekvensgeneratorer. Två stycken har tagits fram via en mikrokontroller, Arduino Due, som har kodats på två olika sätt för att generera olika frekvenser samt för att jämföra strömförbrukningen för dessa två alternativ. Den ena koden genererar en chirp-signal endast med hjälp av Arduinos egna kommandon, och den andra genererar en konstant frekvens, med hjälp av egna inställningar av klockor och register. Tanken var att det skulle vara mer strömsnålt att ställa in allting själv och inte ha saker som man inte använder aktivt, men strömförbrukningen var i det närmaste identisk för båda alternativ. Chirpsignalen fungerar inte heller som den ska, i och med att de kommandon som behövdes för att generera en frekvens endast med Arduinos egna kod inte kunde hantera flyttal. Detta gjorde att det blev omöjligt att ställa in exakta frekvenser, och alla uträkningar som gjordes blev oexakta.

Den tredje frekvensgeneratoren gjordes endast med analog elektronik. Två 555-klockor, några motstånd och två kondensatorer visade sig vara ett smidigt sätt att generera konstanta frekvenspulser med. Med hjälp av potentiometrar så kan man delvis ställa in frekvenserna och klockcyklarna. Detta alternativ är också mindre strömkrävande än en mikroprocessor.

Ett drivsteg för att förstärka signalen från frekvensgeneratoren till en högtalare har också valts. Även ett alternativ till detta drivsteg har tagits upp om större strömförstärkning krävs.

Ett alternativ på en högtalare har också tagits fram. Sonitron har en piezohögtalare i sitt sortiment som är strömsnål och som har ett frekvensomfång på 700 – 20 000 Hz, och eventuellt är det också möjligt att den klarar av högre frekvenser. Detta kunde dock inte tillverkaren garantera. Problemet med denna högtalare är dock att ljudnivån är lägre än vad forskningen visar krävs för att skrämja sälar.

Olika alternativ till batterier har också tagits upp. Det batteri som verkar vara mest lämpat för detta ändamål är ett laddningsbart Li-Ion-batteri. Det har en hög energidensitet, samt har en låg arbetstemperatur vilket är viktigt i och med att anordningen kommer att ligga i vatten som kan vara varierande temperatur beroende på årstiden.

Även möjligheten att ta tillvara på solenergi som uppladdning till batteriet har behandlats. Beroende på vad man väljer att använda för högtalare så kan det vara en bra lösning att använda en solcell som uppladdning till batteriet.

## 5 Diskussion

Detta examensarbete blev lite annorlunda än vad som var tänkt från början. Ju mer forskning som jag tog reda på och läste, desto tydligare blev det att det som vi från början försökte åstadkomma verkade vara mer eller mindre ineffektivt. Detta styrktes än mer då jag tog personlig kontakt med Dr. Thomas Götz och berättade vad jag jobbade på.

Jag hade otroligt mycket hjälp av att ha kontakt med honom personligen. Han gav mig flera artiklar och forskningar att läsa, vilket gjorde att jag fick en bra förståelse för vad som påverkade sälar och varför det som jag försökte åstadkomma med all sannolikhet inte skulle fungera. Arbetet fortskred dock som planerat, jag fick fram det material jag från början ville få fram och jag tycker att jag åstadkom det som skulle åstadkommas. All information för att bygga en första prototyp finns med i examensarbetet och med hjälp av detta examensarbete bör en prototyp kunna byggas och funktionaliteten testas i verkligheten.

Det största problemet jag stötte på under detta examensarbete var utan tvivel att koda med hjälp av Atmels ARM-språk. Det var inte lätt att få fram hur man skulle gå till väga för att ställa in alla de register, klockor och dylikt innan man kunde generera en enkel fyrkantsvåg med hjälp av Atmel-processorn. I referensdokumentet stod tydligt vilka saker man behövde aktivera, men det stod dock inte hur man skulle aktivera dem. Det fick man ta reda på på egen hand, både genom att dyka djupt ner i Arduinos källkodsfiler och genom Atmels hemsida där det fanns vaga beskrivningar på hur man skulle gå till väga och vilken kod man skulle använda för att aktivera olika saker.

Överlag är jag nöjd med mitt examensarbete. Jag tycker att jag har lärt mig mycket under tiden jag har arbetat med det. Jag har fått bra förståelse för hur en mikrokontroller fungerar, jag har lärt mig analysera signaler i Matlab, och jag har fått en god kunskap i hur man ska gå till väga för att få fram fakta och information om det man vill ha fakta om, både via elektroniska källor och via litteratur.

## 6 Källförteckning

Andersson, M. H. & Johansson, A. T., 2013. *Akustiska miljöeffekter av svenska marinens aktiva sonarsystem*, u.o.: Försvars- och Säkerhetssystem.

Arduino, 2015. *Arduino - ArduinoBoardDue*. [Online]

Available at: <http://arduino.cc/en/Main/ArduinoBoardDue>

[Använd 3 april 2015].

Atmel Corporation, 2012. *SAM3X/A Datasheet*. San Jose: Atmel Corporation.

Atmel Software, 2012. *Atmel Software Framework*. [Online]

Available at:

[http://asf.atmel.com/docs/3.4.1/sam.drivers rtc.example.sam3u\\_ek/html/group\\_s\\_a\\_m3\\_u\\_t\\_c.html](http://asf.atmel.com/docs/3.4.1/sam.drivers rtc.example.sam3u_ek/html/group_s_a_m3_u_t_c.html)

[Använd 8 april 2015].

Atmel Software, 2015. *Atmel Software Frameworks*. [Online]

Available at:

[http://asf.atmel.com/docs/latest/sam.drivers.usart.usart\\_synchronous\\_example.sam3u\\_ek/html/group\\_sam\\_drivers\\_pmc\\_group.html#gaf4a6531a439a458a39e38d17627da505](http://asf.atmel.com/docs/latest/sam.drivers.usart.usart_synchronous_example.sam3u_ek/html/group_sam_drivers_pmc_group.html#gaf4a6531a439a458a39e38d17627da505)

[Använd 8 april 2015].

Battery University, 2010. *Battery University*. [Online]

Available at: [http://batteryuniversity.com/learn/article/is\\_lithium\\_ion\\_the\\_ideal\\_battery](http://batteryuniversity.com/learn/article/is_lithium_ion_the_ideal_battery)

[Använd 8 april 2015].

Battery University, 2011. *Battery University*. [Online]

Available at: [http://batteryuniversity.com/learn/article/nickel\\_based\\_batteries](http://batteryuniversity.com/learn/article/nickel_based_batteries)

[Använd 8 april 2015].

Battery University, 2015. *Battery University*. [Online]

Available at: [http://batteryuniversity.com/learn/article/primary\\_batteries](http://batteryuniversity.com/learn/article/primary_batteries)

[Använd 8 april 2015].



Electropaedia, 2005. *Electropaedia, Battery and Energy Technologies*. [Online]

Available at: <http://www.mpoweruk.com/performance.htm>

[Använd 17 april 2015].

Erbe, C., 2011. *Underwater Acoustics: Noise and the Effects on Marine Mammals*. 3 red. u.o.:JASCO Applied Sciences.

Geiger, A. C. & Jeffries, S. J., 1987. Evaluation of seal harassment techniques to protect gill netted salmon.. *Acoustical Deterrents in Marine Mammal Conflicts with Fisheries*., Volym ORESU-W-86-001, pp. 37-55.

Gustavsson, A., 2011. *Tabell- och formelsamling inom data, ellära och elektronik*. 1:4 red. Malmö: Holmbergs.

Götz, D. T., 2015. *E-mailkonversation med Dr. Thomas Götz* [Intervju] (27 mars 2015).

Götz, T. & Janik, V. M., 2010. Aversiveness of sound in phocid seals: psycho-physiological factors, learning processes and motivation.. *Journal of Experimental Biology*, 219(9), pp. 1536-1548.

Götz, T. & Janik, V. M., 2013. Acoustic deterrent devices to prevent pinniped depredation: efficiency, conservation concerns and possible solutions. *Marine Ecology Progress Series*, Volym 492, pp. 285-302.

Hastie, G. D., Donovan, C., Götz, T. & Janik, V. M., 2014. Behavioral responses by grey seals (*Halichoerus grypus*) to high frequency sonar. *Marine Pollution Bulletin*, pp. 205-210.

Importören Outlet, 2015. *Importören Outlet*. [Online]

Available at: <http://xn--importren-57a.com/bproduktinformationb/solpaneler>

[Använd 21 april 2015].

Jain, A., 2012. *Engineers Garage*. [Online]

Available at: <http://www.engineersgarage.com/insight/how-piezo-buzzer-works>

[Använd 30 mars 2015].

Kastelein, R. A. o.a., 2014. *Hearing thresholds of harbor seals (Phoca vitulina) for playbacks of seal scarer signals, and effects of the signals on behavior*, Switzerland: Springer International Publishing.

Kastelein, R. A. o.a., 2010. The effect of signal duration on the underwater hearing thresholds of two harbor seals (*Phoca vitulina*) for single tonal signals between 0.2 and 40 kHz. *Journal of the Acoustical Society of America*, pp. 1135-1145.

Kastelein, R. A. o.a., 2007. Effects of acoustic alarms, designed to reduce small cetacean bycatch in gillnet fishnets, on the behaviour of North Sea fish species in a large tank. *Marien Environmental Research*, 64(2), pp. 160-180.

Mate, B. R. & Harvey, J. T., 1987. *Acoustical deterrents in marine mammal conflicts with fisheries*, Springfield: NTIS, SPRINGFIELD, VA (USA).

Mitt Resväder, 2015. *Mitt Resväder*. [Online]

Available at: <http://www.mittresvader.se/l/finland/klimat-vasa-temperaturer-vattentemperatur.php>

[Använd 21 april 2015].

Naps Solel, 2015. *Fakta om solceller*. [Online]

Available at: <http://www.solelkon.sulterna.se/allt-om-solel/fakta-om-solceller>

[Använd 21 april 2015].

Normandin, S., 2015. *Edison Tech Center*. [Online]

Available at: <http://www.edisontechcenter.org/speakers.html#piezo>

[Använd 30 mars 2015].

Popper, A. N. & Fay, R. R., 1993. Sound detection and processing by fish: Critical review and major research questions.. *Brain, Behavior and Evolution*, 1 August, pp. 14-38.

Solarlab, u.d. *Solarlab*. [Online]

Available at: <http://solarlab.se/solpanel/el-produktion-solceller-sverige.php>

[Använd 21 april 2015].

Sonitron, 2014. *Sonitron - Excellence in physical acoustics*. [Online]

Available at: <http://sonitron.ceros.com/sonitroncatalogue/edition2014/page/97>

[Använd 23 april 2015].

Sonitron, u.d. *Flinkenberg*. [Online]

Available at:

<http://www.flinkenberg.fi/electronics/material/sonitron/SonitronPiezoelectricSpeakerTechnology.pdf>

[Använd 30 mars 2015].

Storr, W., 2015. *Basic Electronics Tutorials Site*. [Online]

Available at: [http://www.electronics-tutorials.ws/transistor/tran\\_1.html](http://www.electronics-tutorials.ws/transistor/tran_1.html)

[Använd 13 maj 2015].

Svanbäck, G., 2012. *Slutrapport, Projekt Sälpingers*, Vasa: Österbottens Fiskeförbund.

Vasek, 2012. *På väg mot totalenergigranskning*. [Online]

Available at: <http://www.vasek.fi/vasaregionens-utveckling-av-vasek/kommunikation/artiklar/pa-vag-mot-totalenergigranskning-2>

[Använd 21 april 2015].

WWF, 2015. *Världsnaturfonden WWF*. [Online]

Available at: <http://www.wwf.se/vrt-arbete/arter/1125843-slar>

(1/2)

```

/*
Lista där man definerar timer, kanal, interrupt samt outputpin.
*/
#define TIMER TC1 //Väljer vilken timer man vill använda, TC0, TC1 eller TC2.
#define CHANNEL 0 //Väljer vilken kanal man vill använda, 0, 1 eller 2.
#define irq TC3_IRQn //Väljer vilken interrupt man vill ha.
int outputPin = 13; //Väljer vilken pin man vill ha som output.
static uint8_t TCChan = 0;
volatile static int32_t toggle;

void setup()
{
  pinMode(outputPin, OUTPUT); //Ställer in så man får outputPin som output.
}

void loop()
{
  tone(6000, 300);

  delay(300);
  stopTone(outputPin);
  delay(300);
}

void tone(uint32_t frequency, int32_t duration) //tone-funktionen skapas
{
  uint32_t rc = VARIANT_MCK / 256 / frequency; //rc ställs in till rätt värde beroende
  på vald klocka och frekvens, 256 är valt istället för 128 för att en svängning är en
  halv Hz
  toggle = 0;
  if (duration > 0 )
  {
    toggle = 2 * frequency * duration / 1000; //gör så att duration-
    värdet blir korrekt i millisekunder
  }
  else toggle = -1;

  if (!TCChan)
  {
    pmc_set_writeprotect(false); //Ställer in i Power Management Controllern, PMC, att
    registret Waveform mode, WAVE = 1, inte är skrivskyddat och går att använda.
    pmc_enable_periph_clk((uint32_t)irq); //Ställer in i PMC och enablar den valda
    perifera klockan man vill ha, i detta fall är det interrupten.
    TC_Configure(TIMER, CHANNEL, TC_CMR_WAVE | TC_CMR_WAVSEL_UP_RC |
    TC_CMR_TCCLKS_TIMER_CLOCK4);
  }

/*
TC_Configure() är kommandot där man ställer in Timer Counterns olika mode.
TIMER är vilken timer vi väljer att använda, har blivit definerad tidigare.
CHANNEL är vilken kanal i TIMER vi väljer att använda, har också blivit definerad.
TC_CMR_WAVE: Ställer in att man vill ha WaveForm mode istället för Capture Mode
TC_CMR_WAVSEL_UP_RC: Detta ställer in att man har automatisk trigger på RC Compare,
dvs. då Countern har räknat upp till RC så kommer det en trigger och countern startar
om på nytt.
TC_CMR_TCCLKS_TIMER_CLOCK4: man väljer Timer Clock 4, Detta innebär att man kör
klockan på 84MHz/128 = 656,252 kHz
*/

TIMER->TC_CHANNEL[CHANNEL].TC_IER = TC_IER_CPCS;
TIMER->TC_CHANNEL[CHANNEL].TC_IDR = ~TC_IER_CPCS;

```

(2/2)

```

/*
  TC_CHANNEL: TC offset: 0x0 ->
  TC_IER: Interrupt Enable Register (write only)
  TC_IDR: Interrupt Disable Register (write only) Dessa båda fås från SAM3X-databladet
  TC_IER_CPCS: Detta togglar RC compare.
  ~TC_IER_CPCS; Detta gör motsatsen till det förra, dvs. disablar RC Compare.
*/

NVIC_EnableIRQ(irq); // Enablar, dvs. öppnar den IRQ man har valt. Nedan finns en
tabell på vilken IRQ man bör ha jämfört med vilken kanal och timer counter man har
valt.

/*
  TC Chan NVIC "irq" IRQ handler function      PMC id
  -----
  TC0  0  TC0_IRQn  TC0_Handler  ID_TC0
  TC0  1  TC1_IRQn  TC1_Handler  ID_TC1
  TC0  2  TC2_IRQn  TC2_Handler  ID_TC2
  TC1  0  TC3_IRQn  TC3_Handler  ID_TC3
  TC1  1  TC4_IRQn  TC4_Handler  ID_TC4
  TC1  2  TC5_IRQn  TC5_Handler  ID_TC5
  TC2  0  TC6_IRQn  TC6_Handler  ID_TC6
  TC2  1  TC7_IRQn  TC7_Handler  ID_TC7
  TC2  2  TC8_IRQn  TC8_Handler  ID_TC8
*/
  TCChan = 1;
}
  TC_Stop(TIMER, CHANNEL); //Stoppar timern före den ställer in RC-registret på nytt
  TC_SetRC(TIMER, CHANNEL, rc); //Ställer in RC-registret så att det får värdet för
  den valda startfrekvensen
  TC_Start(TIMER, CHANNEL); //Startar timern TIMER på kanal CHANNEL
}

static boolean pinState = false;

void stopTone(uint32_t ulPin) //Funktion som stänger av utsignalen
{
  TC_Stop(TIMER, CHANNEL); //Stoppar timern
  digitalWrite(outputPin,LOW); //Ställer utsignalen som låg, dvs 0.
}

void TC3_Handler ( void ) //Denna function kör TC3_Handler, dvs. den hanterar det
interrupt vi använder, som man ser i tabellen ovan.
{
  TC_GetStatus(TIMER, CHANNEL); //Hämtar statusen från den timer och channel som
  används
  if (toggle != 0)
  {
    digitalWrite(outputPin,pinState= !pinState); //För över utsignalen till outputpin
    if (toggle > 0) toggle--; //Räknar ner toggle så att man får önskad längd på
    utsignalen
  }
  else
  {
    stopTone(outputPin);
  }
}

```

```
void setup() //Här görs inställningar som bara ska köras en gång.
{
pinMode(13, OUTPUT); //Väljer vilken pin man vill ha som Output på Arduinon
}

void loop() //Här lägger man in den kod man vill ska köra hela tiden.
{
  chirp(1000, 2000, 300);
  delay(300);
}

void chirp (int startFrekvens, int slutFrekvens, int signalLength) //Chirpfunktionen
{
int frekvensSkillnad = slutFrekvens-startFrekvens;
int medelFrekvens = (startFrekvens + slutFrekvens) / 2;
int duration = (signalLength/1000) * medelFrekvens; //Antalet svängningar under vald tid
int frequencyIncrease = (frekvensSkillnad / duration) * 2;
int sF = startFrekvens - frequencyIncrease; //För att få rätt startfrekvens

while ( duration > 0)
{
  int startFrekvensNew = sF + frequencyIncrease; //För att få rätt startfrekvens
  int startFrekvensReal = 500000/startFrekvensNew; //Frekvensvärdet räknas om till
  mikrosekunder
  digitalWrite(13, HIGH); //Gör utsignalen hög på pin 6.
  delayMicroseconds(startFrekvensReal);
  digitalWrite(13, LOW); //Gör utsignalen låg på pin 6.
  delayMicroseconds(startFrekvensReal);
  sF = sF + frequencyIncrease; //Ökar frekvensen med rätt värde
  duration--;
}
}
```

(1/2)

### Matlab-kod för Chirp-signal

```
clc; clear;
load slutarb.mat A
fs=1000005;
t=0:1/fs:(length(A)-1)/fs;
plot(t,A)
xlabel('Tid (s)')
ylabel('V')
title('Hela ljudkurvan')
fs2=1000005;

%% ljud 1
ljud1 = A(91970:101970);
n = length(ljud1);
tmax = n/fs2;
t2 = (0:1/fs2:tmax-1/fs2);
figure(2);
subplot(211);
plot(t2,ljud1)
xlabel('Tid (s)')
ylabel('V')
title('Startfrekvens chirpsignal')

frekvens1 = fft(ljud1);

subplot(212);
plot ((0:length(t2)/2-1)/tmax,
abs(2*frekvens1(1:length(t2)/2)/(length(frekvens1))), 'b');
axis ( [0 5000 0 4 ] )
xlabel('Frekvens (Hz)')

%% ljud 2
ljud2 = A(211400:221400);
n = length(ljud2);
tmax2 = n/fs2;
t3 = (0:1/fs2:tmax2-1/fs2);
figure(3);
subplot(211);
plot(t3,ljud2)
xlabel('Tid (s)')
ylabel('V')
title('Slutfrekvens chirpsignal')

frekvens2 = fft(ljud2);

subplot(212);
plot ((0:length(t3)/2-1)/tmax2,
abs(2*frekvens2(1:length(t3)/2)/(length(frekvens2))), 'b');
axis ( [0 5000 0 4 ] )
xlabel('Frekvens (Hz)')
```

(2/2)

**Matlab-kod för rak frekvens.**

```
clc; clear;
load rakfrekvens.mat A
fs=1000005;
t=0:1/fs:(length(A)-1)/fs;
plot(t,A)
xlabel('Tid (s)')
ylabel('V')
title('Hela ljudkurvan')
fs2=1000005;

%% ljud 1
ljud1 = A(668200:678200); n = length(ljud1);
tmax = n/fs2;
t2 = (0:1/fs2:tmax-1/fs2);
figure(2);
subplot(211);
plot(t2,ljud1)
xlabel('Tid (s)')
ylabel('V')
title('Frekvens timer-kod rak frekvens')

frekvens1 = fft(ljud1);

subplot(212);
plot ((0:length(t2)/2-1)/tmax,
abs(2*frekvens1(1:length(t2)/2)/(length(frekvens1))), 'b');
axis ( [0 10000 0 4 ] )
xlabel('Frekvens (Hz)')
```



Alla mätningar har gjorts med FLUKE 175 1,0% +3 mätnoggrannhet

10 mätningar per lösning

555: 6,95 mA  
6,98 mA  
6,99 mA  
6,99 mA  
6,96 mA  
6,97 mA  
6,95 mA  
6,98 mA  
6,95 mA  
6,98 mA

Timerlösning: 119,7 mA  
120,0 mA  
119,3 mA  
119,4 mA  
119,9 mA  
119,5 mA  
119,5 mA  
119,7 mA  
119,1 mA  
119,6 mA

Softwarekod: 119,9 mA  
119,7 mA  
120,4 mA  
120,6 mA  
120,0 mA  
119,6 mA  
119,9 mA  
120,4 mA  
120,2 mA  
120,0 mA