

Visslingsenhet för delfiner

Niclas Fristedt

2015



LUNDS
UNIVERSITET

Examensarbete i elektrisk mätteknik

Lunds tekniska högskola

Institutionen för biomedicinsk teknik

Handledare: Prof. Hans W Persson

Abstract

The Kolmården wildlife park located in Sweden, asked for a device to be used in their dolphinarium to enable the trainers to call individual dolphins. This is to be done using their signature whistle, which is unique for every animal. Since dolphins themselves are using their signature whistle to make contact between each other, the hypothesis is that teaching them to understand that their own unique signature whistle is a recall signal, will be easier than with an arbitrary sound. In the current situation knocking the metal fish buckets on the pool side or a sharp hand splash in the water surface are used as recall, but it will call the whole group. The idea is that the trainer selects which animal he/she wants to work with and plays the corresponding signature whistle in the water. The whole device or only the transducer will be submerged into the pool, a spherical hydrophone will act as a transmitter to spread out the whistle sound evenly. The aim with this master thesis was to build and test a device that is capable to transmit the relative low frequency signature whistle, 5kHz-20kHz. The prototype used an Arduino Nano, a microcontroller as a platform. At the end of the project, the prototype was tested in Kolmården. The prototype worked fine to the extent that the signature whistles were properly transmitted into the waters, and the dolphins could clearly hear them. However, the whistles had no recall effect this has to be taught to the dolphins, as anticipated. All in all the project was successful and there is great potential to further develop the device, e.g, to incorporate not only signature whistles, but also other sounds that can be used to communicate with the dolphins

Sammanfattning

I Kolmårdens djurpark efterfrågades en enhet som ska kunna användas för att kalla in delfinerna individuellt m.h.a deras signaturvisslingar som är unika för varje delfin. Idag knackar delfintränarna med de rostrfria fisk-skålarna i bassängkanten eller plaskar snärtigt i vattenytan för att kalla på delfinerna. Detta innebär dock att alla delfiner kommer; det går inte att kalla på en enskild individ. Tanken är att en delfintränare ska kunna välja vilken delfin som ska kallas in och spela upp motsvarande signaturvissling. Djuren får dock tränas att förstå detta. I projektets första del skall en prototyp utvecklas med arduino som plattform, och med en sfärisk piezoelektrisk, rundstrålande givare som kommer sprider ljudet jämnt i hela bassängen. I slutet av projektet testades prototypen i Kolmården med signaturvisslingen från två av delfinerna. Prototypen fungerade bra så tillvida att visslingarna sändes ut i bra kvalitet och tillräckligt högt för att delfinerna kunde höra dem utan problem. Dock reagerade de negativt vilket var förväntat; de måste lära sig att förstå att signaturvisslingarna skall fungera som inkallningssignal. Detta förväntas gå snabbare och enklare med signaturvisslingar än med godtyckligt valda andra ljud. Förmodligen kommer signaturvisslingarnas nya funktion också intuitivt att förstås av de andra i flocken. Projektet anses i sin helhet som ett lyckat arbete och det finns stor potential att vidareutveckla enheten att t.ex, inte bara kunna sända signaturvisslingar utan andra ljud som kan användas för att kommunicera med delfinerna.

Förord

I samband med ultraljudskursen EEMN15 2013 gjordes en resa till Kolmården djurpark där första kontakten med Mats Amundin (forskningschef i Kolmården och gäst-professor vid Linköping Universitet) togs. Detta projekt är Amundins idé och jag är tacksam att fått förtroende att utveckla denna prototyp. Jag vill tacka min handledare professor Hans Persson och universitetslektor Monica Almqvist som har varit min biträdande handledare under arbetet. Jag vill även tacka postdok Christian Antfolk som har kommit med förslag gällande programmeringen och hjälpt mig med de praktiska moment som har utförts. Arbetet startade hösten 2014 och blev färdigt sommaren 2015.

Innehåll

1	Inledning	8
1.1	Rapportens struktur	8
1.2	Klargörande av författaren	8
2	Metodik	10
2.1	Hur skapar delfiner ljud?	10
2.1.1	Vad är en signaturvissling?	13
2.2	Vikning	14
2.3	PWM (Pulse Width Modulation)	15
2.4	SPL (Sound pressure level)	16
2.5	SeaWave	16
2.6	Kort om filformatet WAV	17
2.7	Analys av delfinvisslingarna	18
2.8	Blockdiagram	21
2.9	Programmering av arduino	21
2.10	Externt minne - SPI kommunikation	23
2.11	Givare	24
2.12	Lågpasfilter	24
2.12.1	Medelvärdes-avbildning av Arduinos PWM signal	27
2.13	D/A-omvandlare	27
2.14	AC-koppling	28
2.15	Förstärkning	29
3	Resultat	32
3.1	Lågpasfiltreringen	32
3.2	Förstärkning	34
3.3	Frekvensanalys av visslingarna	35
3.4	Prototyp	39
3.5	Sluttest i Kolmården	40
4	Diskussion	42
4.0.1	Framtida arbete	43

5	Appendix	44
5.1	Arduino-koden	44
5.2	Kretsschema	48

1 Inledning

I samband med kursen ultraljudsfysik och teknik som ges på Lunds tekniska högskola gjordes en studieresa till Kolmårdens djurpark. Där togs den första kontakten med Mats Amundin. Mats efterfrågade en enhet som ska kunna användas för att kalla in en specifik delfin. Idag knackar delfintränarna med de rostrfria fiskskålarna i bassängkanten eller plaskar snärtigt i vattenytan för att kalla på delfinerna. Detta innebär dock att alla delfiner kommer; det går inte att kalla på en enskild individ. Tanken var att kommunicera med delfinerna på deras språk.h.a. deras s.k. signaturvisslingar. Det system som har utvecklats är baserat med en mikro-kontroller som plattform, specifikt Arduino Nano. Om detta system implementeras i det daliga arbetet skulle det bl.a. kunna leda till effektivare träning av delfinerna då mindre tid skulle läggas på att få fatt i rätt delfin. Den här rapporten presenterar alltså en fungerande prototyp, baserat på Mats Amundins idé.

1.1 Rapportens struktur

Rapporten börjar med ett mindre teoriavsnitt där de intressanta och mest användbara uttrycken för projektet går igenom. Detta följs av ett metodavsnitt där uppställning för arbetet presenteras, författaren gör detta i kronologisk ordning. Senare presenteras resultat, som innehåller diagram och figurer utifrån både simuleringar och mätningar. Slutligen presenteras ett diskussionsavsnitt där tänkbara ändringar, funderingar och framtida arbete utreds. I appendix återfinns arduino-koden och kretsschemat för den slutgiltiga prototypen.

1.2 Klargörande av författaren

I projektet användes två olika men ändå snarlika tillvägagångssätt för att lösa uppgiften med att bygga denna enhet. Detta skulle författaren vilja förklara närmare redan nu för att undvika missförstånd senare i läsningen. I projektets första fas var utsignalen i form av en PWM (pulsbreddmodulerad signal) ifrån Arduinon, p.g.a. begränsningar och problem som uppkom med detta övergavs denna idé till förmån för en 8-bitars digital utsignal.

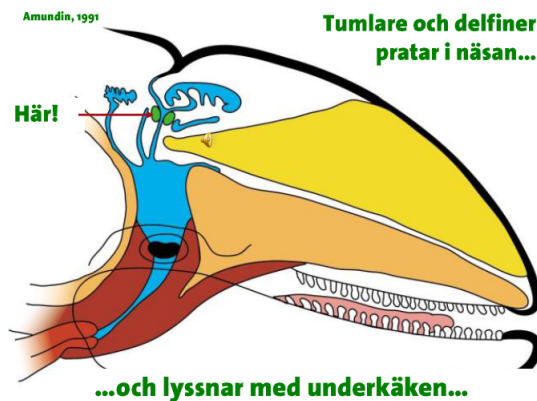
Den enda praktiska skillnaden mellan dessa två lösningar är användningen av en D/A-omvandlare och justeringar i programmeringen, i övrigt är de två lösningarna identiska.

2 Metodik

Arduino valdes som plattform i projektet, delvis p.g.a. tidigare erfarenhet i andra projekt. Utsignalen studerades med ett oscilloskop, frekvens och amplitudnivåerna var av störst intresse. Kretsen byggdes upp på en kopplingsplatta och testades i ultraljudlabbet på institutionen innan experiment hos Kolmården kunde genomföras. Simuleringar av den analoga delen i projektet gjordes i LTspice, ett gratisprogram som ges ut av Linear Technology [5]. Fördelen med detta program är att de faktiska komponenterna som har använts kan testas innan de kopplas upp. Detta sparade tid och användes flitigt. Även Matlab användes för att skapa en del diagram. Utsignalen ifrån arduinon behövde filtreras och förstärkas så att en utsignal så lik som möjligt delfineras ursprungliga vissling kunde erhållas. Frekvensanalyser av utsignalen gjordes i programmet SeaWave [6] och även i Matlab. I början av projektet lades utsignalen ut i form av en PWM-signal, detta ändras sedan, som beskrivits ovan till förmån för en D/A omvandlare. Både dessa tillvägagångssätten presenteras och går igenom i följande avsnitt.

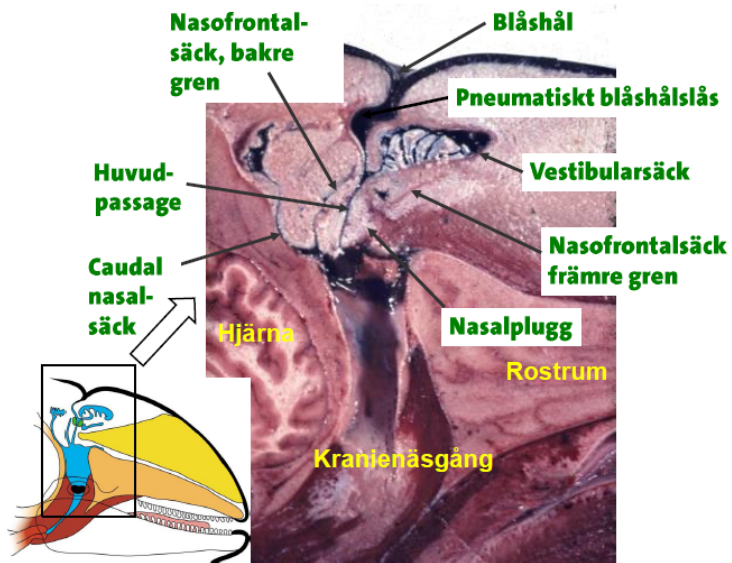
2.1 Hur skapar delfiner ljud?

Den allmänt accepterade hypotesen hurvida delfiner skapar sina ljud är följande. Människor skapar ljud genom att stämbandets i struphuvudet vibrerar när luft strömmar igenom. Vår mun, tunga och läppar formar dessa ljud till tal. Delfiner saknar stämband, istället skapas deras ljud, visslingar och klickljud, genom att luft under högt tryck pressas förbi de s.k ljudläpparna [1] i blåshålspassagen. Man kan alltså säga att delfiner pratar i näsan, se figur 1.



Figur 1: Grafisk presentation av delfiners ljudorgan [1]

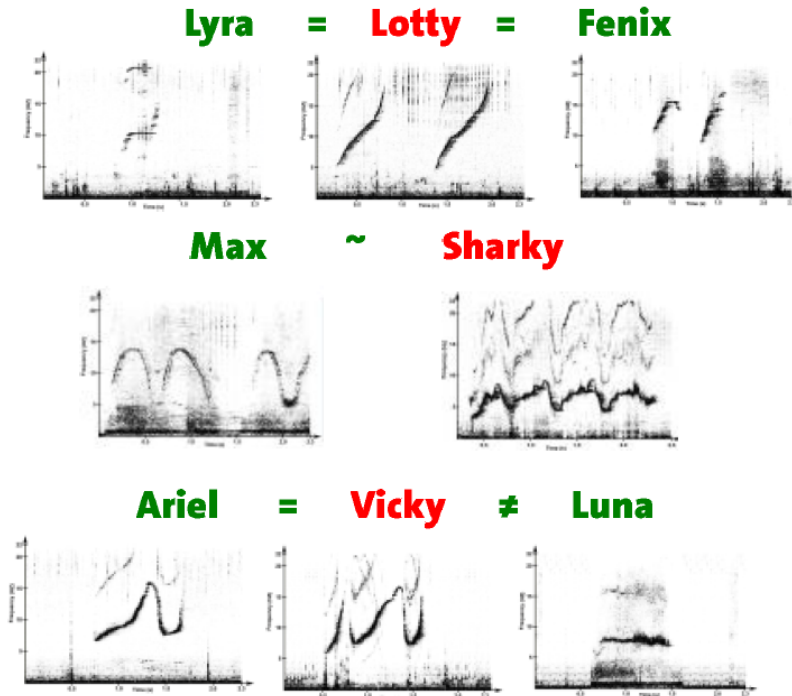
När delfin befinner sig under vatten stängs blåshålet till. Ett kraftigt övertryck skapas i kranienäsgången genom att muskler som omsluter den s.k. epiglottis, kontraherar. Denna tryckluft släpps sedan förbi ljudläpparna, som sätts i vibration; dessa vibrationer fortplantar sig genom melonen, som fungerar som en akustisk lins (se fig. 1). Den akustiska impedansen i melonens yta är densamma som i vattnet, vilket gör att det inte blir några förluster. Luften som använts samlas i de s.k vestibularsäckarna, som ligger alldeles under blåshålet. Detta kan liknas vid en ballong som är fylld med luft och klämma till änden och låter luft passera, se figur 2. Intressant nog finns det två uppsättningar av ljudläpparna, ett par till vänster och ett till höger i nasalpassagen, vilket gör det möjligt för delfiner att producera klick och visslingar oberoende av varandra och samtidigt. [1]



Figur 2: Närbild på en delfins nasala system [1]

2.1.1 Vad är en signaturvissling?

Varje delfin har sin egen signaturvissling. Till skillnad från övriga ljud som delfiner skapar t.ex. klick för ekolokalisering som innehåller höga frekvenser ända upp till delfinernas övre hörgräns, som ligger vid 150kHz, så ligger signaturvisslingarna vanligtvis i 5kHz-20kHz-området. Signaturvisslingen är ett socialt ljud som djuren använder sig av för att hålla reda på varandra. Delfinernas hörtröskel ligger vid dessa frekvenser mellan 50-60 dB re. 1 μ Pa. Lägre frekvenser färdas också längre vilket kan vara fördelaktigt då delfinerna ute på öppet vatten kommer ifrån flocken och ska hitta tillbaka. Delfinerna kan även härma varandras visslingar för att skapa kontakt. Så om delfin Vicky letar efter sin dotter Luna kan hon härma hennes vissling, och om då Luna hör detta svarar hon med sin egen signaturvissling för att säga "Här är jag!". Det kan liknas vid ett namn såsom vi själva använder det (eftersom alla visslingar är unika). Visslingen utvecklas under det första levnadsåret och ändras inte efter det. Delfiner ifrån samma familj och flock kan ha liknade visslingar men det behöver inte vara så. Detta är inte konstigt då ungarna influeras av speciellt mamman men också flocken då den växer upp. De tar då till sig en liknande vissling, detta kan ses i figur 3. Till exempel så påminner Fenix vissling om hans mor Lottys medan Lunas vissling inte alls är lik sin mor Vickys.



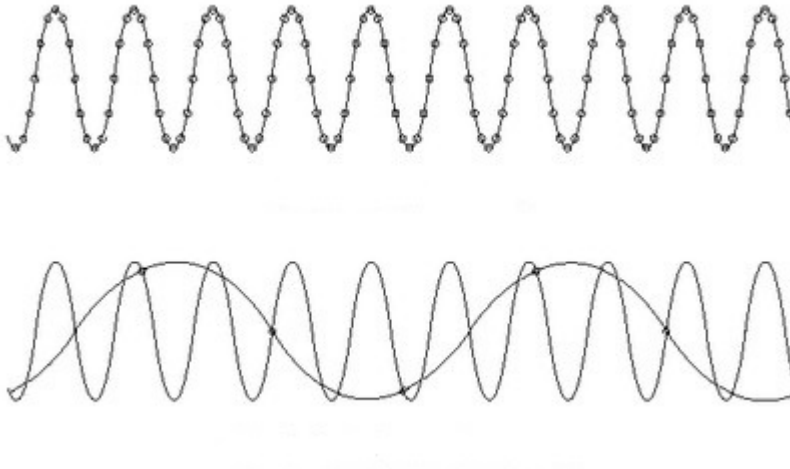
Figur 3: Exempel på signaturvisslingar [1]

I projektet användes två signaturvisslingar, från Vicky och Luna. [1]

2.2 Vikning

Under första delen av arbetet uppkom problem med vikning. Enligt Nyquist samplingsteorem måste en signal samplas med minst den dubbla frekvensen som signalen innehåller för att kunna återskapa signalen korrekt. I princip så brukar samplingsfrekvensen ligga något högre än det dubbla, t.ex använder CD-spelare en samplingsfrekvens på 44,1kHz vilket är mer än dubbelt så mycket som det mänskliga örat kan uppfatta på 20kHz. Om detta inte uppfylls uppstår vikningseffekten. Detta uppstår alltså när samplingsfrekvensen är för låg sett till frekvensinnehållet på den samplade signalen. Effekten av detta är att de höga frekvenserna ”viks ner” till lägre

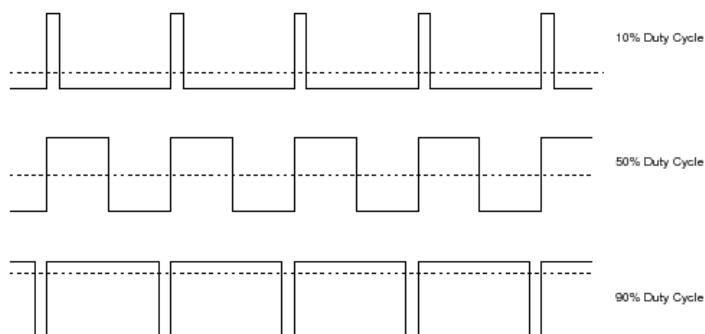
frekvens och kan uppfattas som brus eller annan störning. Figur 4 visar ett exempel på vikning när en tillräcklig samplingsfrekvens för att kunna återskapa signalen respektive en signal som har undersamplats och en lägre frekvens erhålles. Vikningfenomenet uppkommer i samband med A/D och D/A konvertering.



Figur 4: Ett exempel på vikningseffekten

2.3 PWM (Pulse Width Modulation)

PWM (på svenska pulsbreddsmodulering) är en teknik som används för att koda information till en pulserande utsignal. PWM signalen skapas i vanliga fall genom att slå av och på spänningen med en viss pulskvot (eng. duty-cycle), detta uttryckes i procent. Se figur 5. Denna teknik användes under första delen av projektet innan det övergavs för att lägga ut utsignalen genom en D/A-omvandlare. [15]



Figur 5: Exempel på PWM signaler med tre olika pulskvoter. 10, 50 respektive 90 procent. [15]

2.4 SPL (Sound pressure level)

För att mäta ljudnivåerna i arbetet användes ofta SPL, vilket ger ett värde i decibel. Decibel är ett logaritmiskt mått och har traditionellt använts för att beskriva intensitet samt tryck av akustiska vågor och används flitigt inom ultraljudsteknik. Decibel är bekvämt sätt att jämföra ratios och absoluta värden under vatten då känsligheten och uteffekt hos de hydrofoner och givare som används specificeras i decibel. SPL definieras som 2.1; [2]

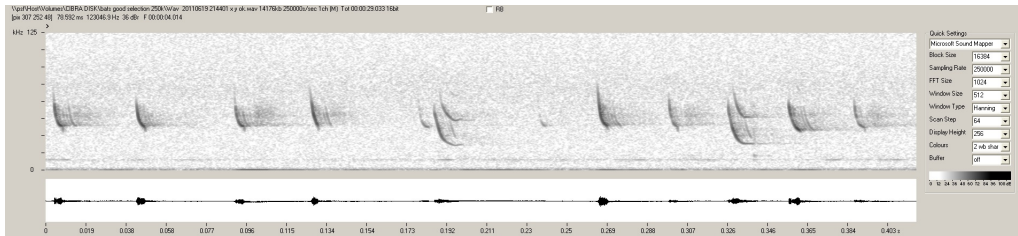
$$SPL = 20 \log\left(\frac{p}{p_0}\right) \quad (2.1)$$

Där p_0 är referensvärdet, vanligtvis $1 \mu Pa$ under vatten. SPL är alltså ett mått för att bestämma ljudnivån på det utsända ljudet. Enligt specifikationer ifrån Mats A. ska givaren ha en källstyrka på 120db re $1 \mu Pa @ 1m$. [1]

2.5 SeaWave

SeaWave är ett program som i realtid framställer ett spektrogram, vilket bl.a. kan användas för en akustik-analys av delfiners signaturvisslingar. Ett spektrogram är ett diagram med tre dimensioner, frekvens i y-led, tid i x-led och intensitet i z-led (via svärtningsgraden eller färgskala). Se ett

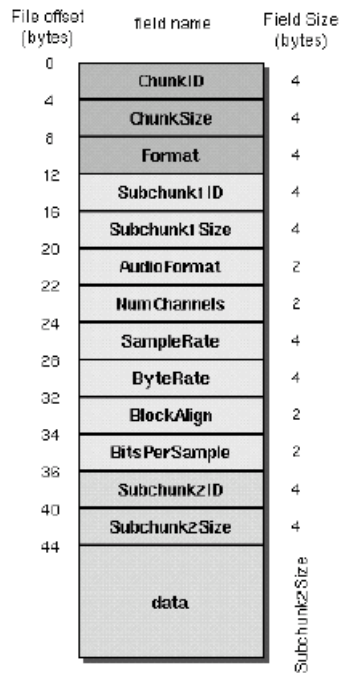
exempel av inspelning i SeaWave i figur 6. Detta program användes till viss del när analyser gjordes av signaturvisslingar, men för det mesta användes Matlab för att analysera visslingarna i ett spektrogram. [6]



Figur 6: En inspelning av fladdermöss ultraljud i SeaWave [6]

2.6 Kort om filformatet WAV

Inspe­lingar av signaturvisslingarna gjordes i Kolmården. Ljudfilerna var i WAV (Waveform audio file) format. WAV filer är ofta okomprimerade, detta medföljer att filerna kan vara stora. WAV-headern preciseras enligt följande, se figur 7. Detta är av betydelse sedan programmeringen grundar sig i själva headern.

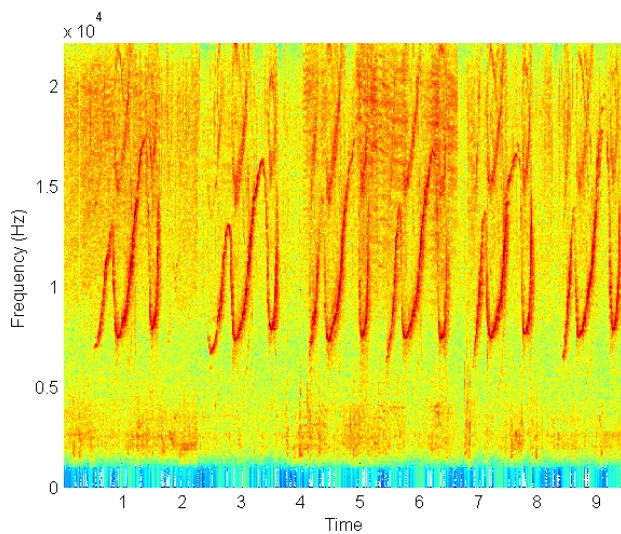


Figur 7: WAV-header [3]

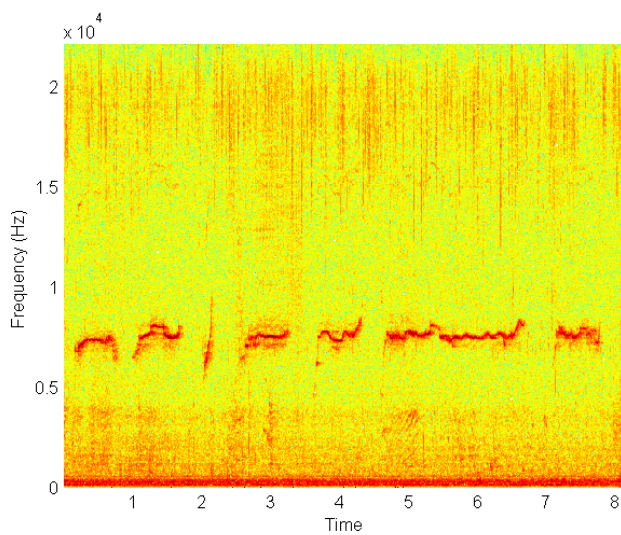
Information om ljudfilen specificeras i det första segmentet i form av 44 bytes, t.ex information som samplingshastighet och antal kanaler. Sedan följer ett större datasegment med själva ljudet. [4]

2.7 Analys av delfinvisningar

Kolmården skickade som sagt två signaturvisningar. Från Vicky och Luna. Deras signaturvisningar analyserades i matlab vilket kan ses i figur 8 och 9.

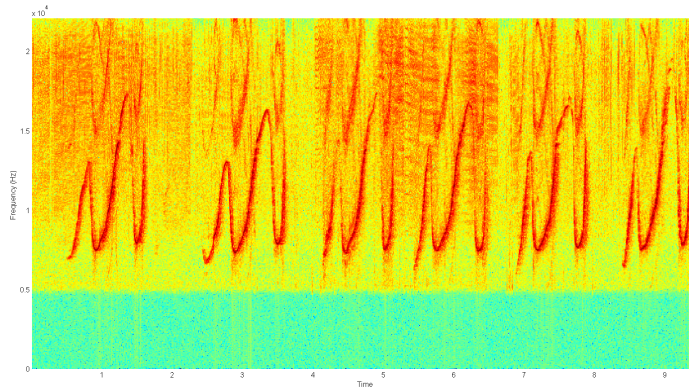


Figur 8: Originalinspelning av Vicky i Matlab

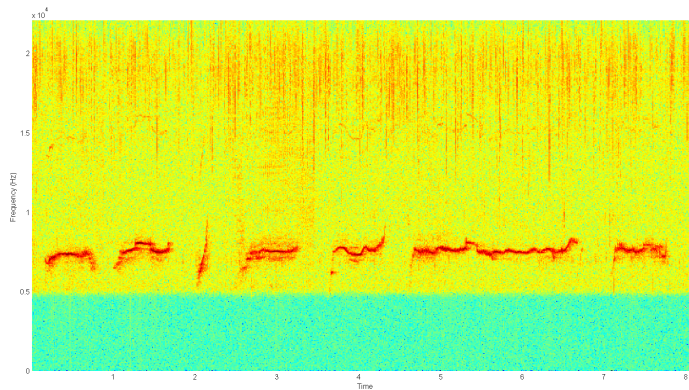


Figur 9: Originalinspelning av Luna i Matlab

Det som kan ses, speciellt i Lunas spektrogram är att det finns störningar i de lägre frekvenserna. Förmodligen orsakat av vågor under inspelningstillfället. P.g.a. detta gjordes en digital filtrering av filerna i matlab. Detta i form av ett 50-ordningens-högpasfilter med brytfrekvens 5kHz. Resultatet av denna filtrering kan ses i figur 10 och 11. Utöver detta så normaliserades ljudnivån i filerna, också det i Matlab.



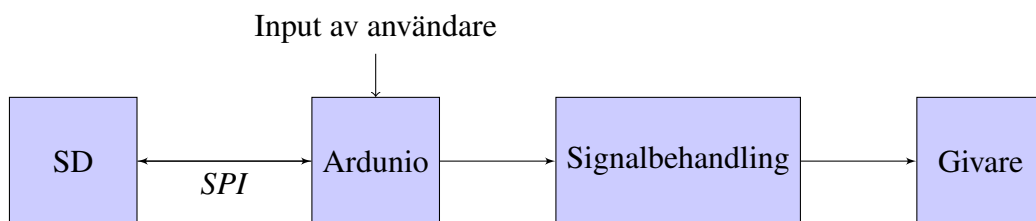
Figur 10: Inspelning av Vicky med pålagt digitalt filter i matlab



Figur 11: Inspelning av Luna med pålagt digitalt filter i matlab

2.8 Blockdiagram

En grafisk presentation av hur kretsen är uppbyggd i figur 12. Användaren väljer vem eller vilka delfinvisling som ska kallas in genom att mata in detta till Arduinon. Först gjordes detta genom att använda Arduinos serial monitor. Arduinon hämtar sedan respektive delfinvisling, som ligger lagrad på SD-kortet över SPI-kanalerna (mer om detta i 2.10). I projektets första fas skickade Arduinon ut denna ljudfil som en PWM-signal, p.g.a. detta signalbehandlas signalen för att återskapa den ursprungliga visslingen. Störningar filtrerades även ut innan ljudet slutligen förstärktes till nödvändig nivå för givaren. Se figur 12.



Figur 12: Blockdiagram av kretsen

2.9 Programmering av arduino

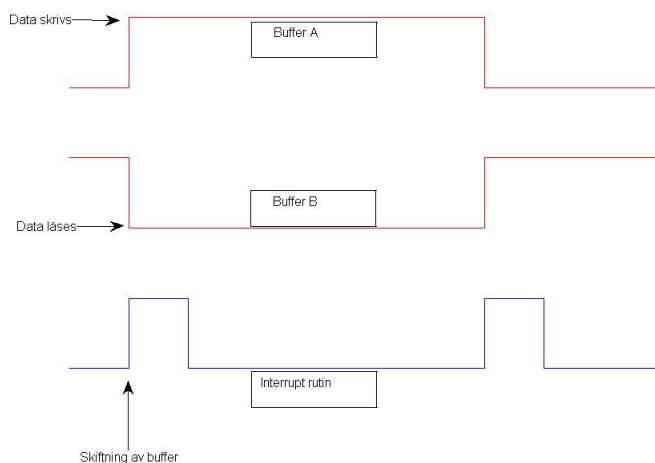
En stor del av projektet ägnades åt programmering. Arduinos mjukvara består av en IDE (Integrated Development Environment) och kärnbibliotek. IDE skrivs i java medan kärnbiblioteken kan skrivas i både C och C++. För detta projekt har endast kodning i IDE skett.

Processen vad som sker efter input av användaren beskrivs här.

1. Arduinon läser kontinuerligt av om en vissling ska spelas upp. När en input kommer ifrån användaren öppnas korresponderande ljudfil över SPI-kanalen ifrån SD-kortet.

2. Två buffrar om 256 bytes skapas. Detta för att kunna läsa data ifrån den ena buffern medan den andra skrivs in med de nästkommande 256 bytes ifrån ljudfilen. När en buffer har skickat ut sin data skiftar dessa plats.
3. Ett interrupt skapas utifrån samplingsfrekvensen av vald ljudfil. Detta för att kunna skicka ut data i samma hastighet som den är inspelad i. Interrupt rutinen kollar även om buffrarna ska skifta plats.

Detta fungerar relativt väl då inläsningen av ny data ifrån SD-kortet är mycket snabbare än hastigheten som signalerna skickas ut. Över SPI-kanalen kan data läsas in med en hastighet av 8M bytes/s. T.ex om utsignalen har en hastighet på 32k bytes/s så spelas alltså 256 bytes upp under 8ms medan det endast tar 0,03ms att lägga in 256 bytes i en buffer.

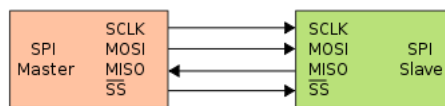


Figur 13: Grafisk presentation av buffer-skiftet

Anledningen till att bufferstorleken är 256 bytes är p.g.a. att arduinons minne är begränsat, se 2.10. Hela ljudfilen kan alltså inte laddas in i arduinons minne utan måste läses från det externa minnet där filerna lagras. I början av projektet lades utsignalen ut på en kanal i form av pulsbreddsmodulering, men i projektets senare del byttes detta ut till en 8bitars D/A omvandlare. Alltså behövdes utsignalen läggas ut på 8 kanaler, med en bit för varje kanal. Programmering för dessa två metoder påminde om varandra och använde samma princip med buffrarna och interrupt-rutin.

2.10 Externt minne - SPI kommunikation

Ljudfilerna lagras externt på ett SD-kort då Arduino Nano har ett begränsat minne på 32kb. Filerna är i wav-format och har konverterats ner ifrån 44.1kHz till 32kHz för att uppfylla Nyquistkriteriet eftersom Arduinon har en samplingshastighet på 64kHz. Konverteringen av ljudfilerna gjordes i itunes. En kommunikation mellan dessa två enheter sattes upp genom att använda serial peripheral interface (SPI) kanalerna på arduinon, dessa befinner sig på kanalerna 10-13. SPI är ett synkront data-protokoll som används av en mikrokontroll för att kommunicera med externa enheter, bl.a. med SD-kort men kan också användas för att två mikrokontroller ska kommunicera med varandra. I en SPI-anslutning agerar en enhet master, i det här fallet arduinons mikrokontroll ATmega328 och SD-kortet agerar som slav. Följande fyra anslutningar används; MISO (master in slave out), MOSI (master out slave in), SCLK (serial clock), och CS (chip select, ibland kallat SS för slave select) Fördelen med SPI är att full duplex kommunikation kan ske, se figur 14. Utöver dessa anslutningar behövs kortet matas med 3,3V samt en jordkontakt. [7], [8]



Figur 14: SPI kommunikation

2.11 Givare

Ifrån Kolmården skickades först en okänd sfärisk piezoelektriskt givare ner. Avsaknaden av datablad gjorde det nödvändigt att undersökta den i impedansanalysatorn. Här studerades främst bandbredden, vilket visade sig vara mellan 5kHz-95kHz. Givarens bandbredd begränsas uppåt av dess resonansfrekvens på 100kHz. Se figur 15. Givarens yttre hölje är gjord av polyvinylidfluorid, vilket är en typ av plast som har en akustisk impedans som ligger nära vattnets. Detta gör att reflektioner minimeras när givaren skickar ut signaler. Själva givaren inuti plasthöljet är sannolikt av keramisk typ. När spridningen av ultraljudet studerades upptäcktes det att ljudet ej spreder sfäriskt (vilket man lätt kan tro vid en första blick på givaren) utan i ett plan. Detta är anledningen till att givaren anses vara i form av en platt skiva.[1]

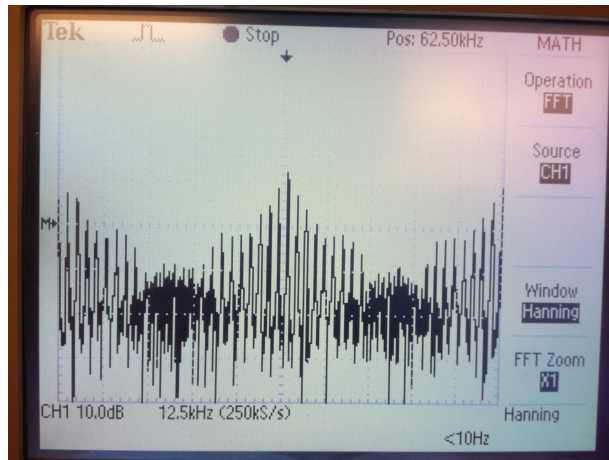


Figur 15: Givaren som användes i större delen av projektet

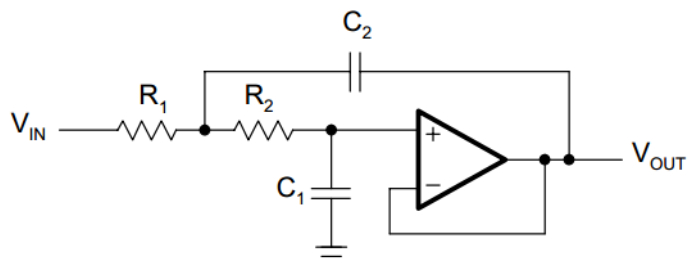
2.12 Lågpasfilter

I fallet med PWM utsignalen behövdes en filtrering för att medelvärdesanpassa signalen samt minimera störningar. Ett aktivt fjärde-ordningens butterworth lågpasfilter valdes, dels p.g.a dess minimala förstärkning i passbandet som passar bra tillsammans audio (bild på bodeplotten). Brytfrekvensen sattes till 16kHz. Simulering av lågpasfiltreringen gjordes i LTspice [5] och diagram kunde erhållas enkelt bl.a. bodediagram. Eftersom delfiner har ett hörbart område upp till ca. 150kHz, är det alltså av stor relevans

att filtrera bort störningar över visslingsfrekvenserna. Samplingsfrekvensen på 64kHz ifrån arduinon var t.ex en störning som vi ville dämpa, se figur 16. Lågpasfiltreringen konfigurerades i Sallen-Key topologi, figur 17. Genom att kaskadkoppla två av dessa filter uppnås ett fjärde-ordningens filter.



Figur 16: Frekvensanalys av utsignalen ifrån arduinon av en 5kHz ton



Figur 17: Ett andra ordningens Unity-Gain Sallen-Key lågpasfilter [9]

Nedanför går beräkningarna igenom som användes för att bestämma parametervärdena för komponenterna.

Generell överföringsfunktionen för ett lågpassfilter. Där A_0 är DC-förstärkningen av signalen, a_i och b_i är filterkoefficienter.

$$A_i(s) = \frac{A_0}{(1 + a_i s + b_i s^2)} \quad (2.2)$$

Överföringsfunktionen för ett andra ordningens filter. Detta kan förenklas ytterligare genom att jämföra med ekvation 2.2.

$$A(s) = \frac{1}{1 + \omega_c C_1 (R_1 + R_2) s + \omega_c^2 R_1 R_2 C_1 C_2 s^2} \quad (2.3)$$

$$A_0 = 1 a_1 = \omega_c C_1 (R_1 + R_2) b_1 = \omega_c^2 R_1 R_2 C_1 C_2 \quad (2.4)$$

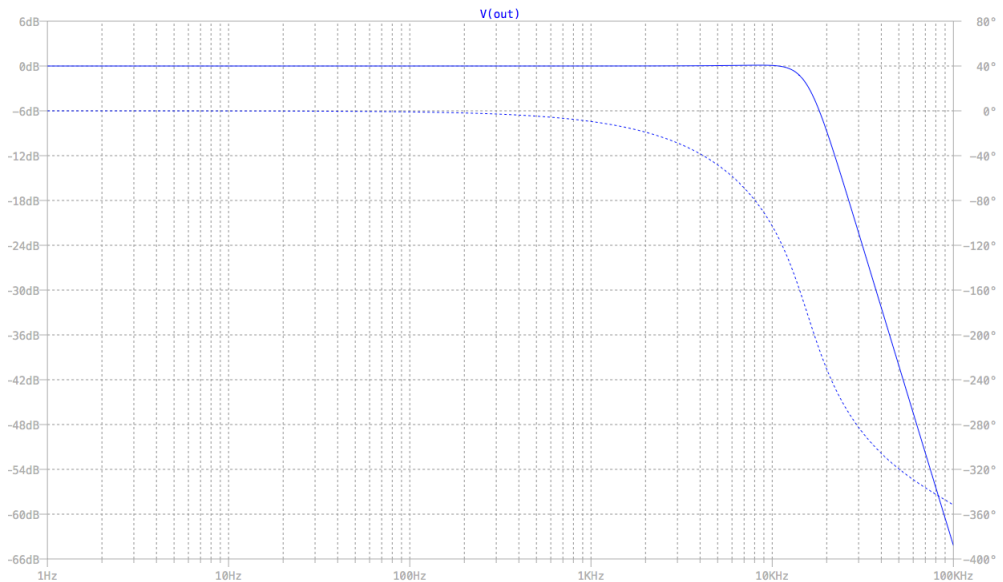
De två resistorerna kan då beräknas enligt 2.5.

$$R_{1,2} = \frac{a_1 C_2 \pm \sqrt{a_1^2 C_2^2 - 4 b_1 C_1 C_2}}{4 \pi f_c C_1 C_2} \quad (2.5)$$

För att undvika roten ur ett negativt tal måste också detta uppfyllas, ekvation 2.6.

$$C_2 \geq \frac{4 C_1 b_1}{a_1^2} \quad (2.6)$$

Utifrån detta kan lågpassfiltret konstrueras om C väljs till lämpligt värde. För att använda en högre ordning på filtren, i detta fall som sagt ett fjärdeordningens filter användes tabeller från Texas Instrument där a samt b var givna. Den slutgiltiga lågpassfiltrering som erhöles visas i bodediagrammet, figur 18. [9]



Figur 18: Simulerad bodediagram av fjärde-ordningens lågpasst butterworth filter

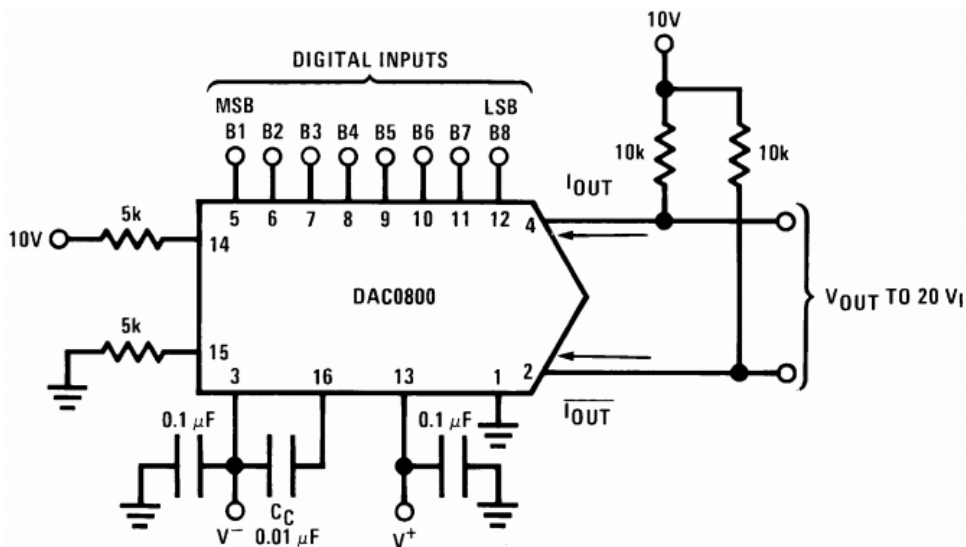
2.12.1 Medelvärdes-avbildning av Arduinos PWM signal

Utsignalen ifrån arduinon är en PWM signal (pulse width modulation, på svenska pulsbreddsmodulering). Enligt Texas Instrument [13] kan en lågpastfiltrering återskapa den ursprungliga sinusformade delfinvisningen. Detta involverar en analog lågpastfiltrering som uppnås med samma krets som ovan (17). Artikeln går igenom både passiv och aktiv filtrering, om aktiv filtrering används måste hänsyn tas till op-förstärkarens bandbredd som måste vara 5-10 gånger så stor som den största förväntade utsignalen, vilket i detta fall innebär 80kHz-160kHz vilket inte är något större problem för de flesta op-förstärkarna. [10]

2.13 D/A-omvandlare

I den senare delen av projektet valdes utsignalen att läggas ut genom en D/A-omvandlare. För detta användes DAC0800 ifrån Texas instruments

[11].



Figur 19: Kopplingsschema över DAC0800 [11]

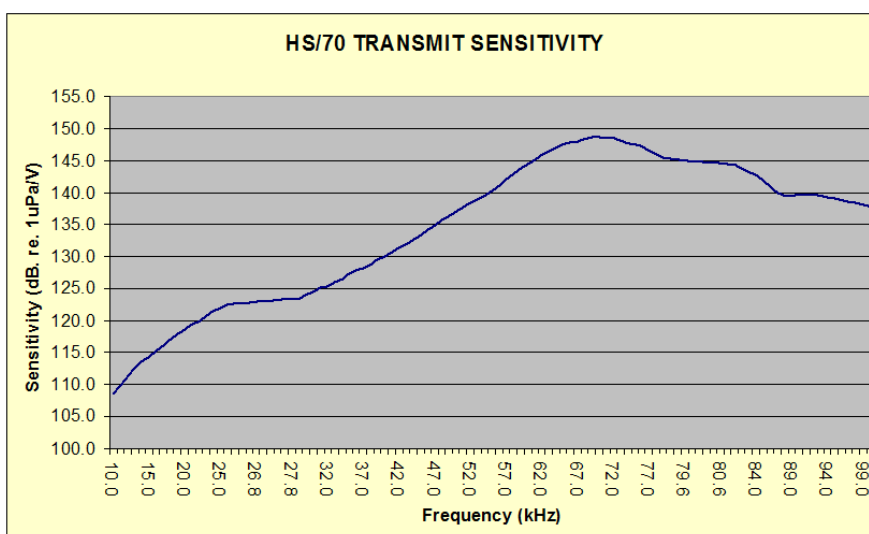
DAC0800 är en 8-bitars D/A-omvandlare med en typisk stabiliserings-tid (eng. settling time) på 100 ns.

2.14 AC-koppling

Utsignalen ifrån arduinon var som sagt en PWM-signal mellan 0-5V. Detta medförde att det fanns en oönskad DC-offset på 2,5V efter lågpassfiltreringen. Detta var ett problem när signalen skulle förstärkas eftersom signalen kunde klippas vid de högre amplituderna. Därför behövdes en AC-koppling som tog ner denna DC-offset till 0V. En DC-offset på utsignalen har annars ingen påverkan på frekvensinnehållet och ljudkvalitén, förutom just att den kan klippas. Detta implementeras med ett högpasfilter med brytfrekvens på 100Hz mellan filtreringen och förstärkningen.

2.15 Förstärkning

För att bestämma förstärkningen av signalen gjordes tester utifrån Mats A's rekommendation att givaren skulle ha en källstyrka av 120dB. En referenshydrofon, TC4013 [12] användes för detta. I en vattentank studerades impulssvaret ifrån givaren med TC4013 direkt kopplad till ett oscilloskop för att avläsa amplitudnivåerna. Utifrån givarens transmissions-egenskaper (fig 20) kan den ungefärliga drivspänningen V_{in} beräknas genom ekvation 2.7. P.g.a. givarens transmissions-egenskaper i fig 20, där högre frekvenser förstärks betydligt mer än i området där signaturvisslingarna befinner sig valdes det i slutändan att behålla lågpassfiltreringskretsen ifrån föregående avsnitt, dock nu en med brytfrekvens på 20kHz.



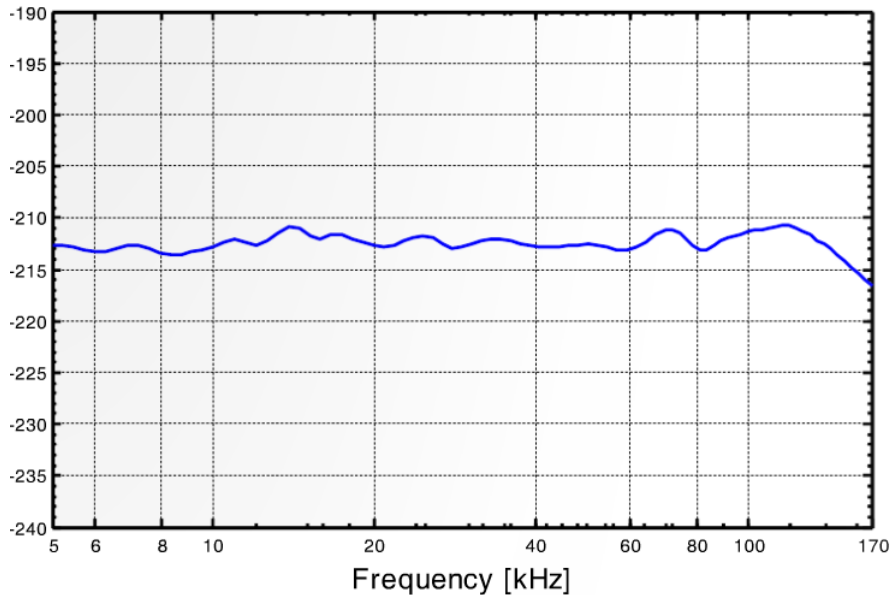
Figur 20: Givaren HS/70 transmissions känslighet [1]

Den ungefärliga drivspänningen beräknas genom: [14]

$$SL = S_v + 20 \log_{10}(V_{in}) \quad (2.7)$$

Där S_v är givarens transmissions-känslighet vid en viss frekvens i dB re. $1 \mu\text{Pa/V}$. SL är ljudintensiteten i dB. Om 120dB önskas som källstyrka för

en 10kHz signal krävs då en spänning på ungefär $4V_{p-p}$. För att kalibrera detta användes som sagt en referenshydrofon. Denna hade en känslighet på -211 dB re. $1V/\mu\text{Pa}$ på 1m avstånd. Se figur 21.



Figur 21: Hydrofon TC4013 mottagare känslighet i dB re. $1V/\mu\text{Pa}$ [12]

Ett avstånd på en meter skulle bli problematiskt i den vattentank som användes, p.g.a reflektioner. Hydrofonden placerades därför nära givaren. I samband med detta måste transmissions-förlusterna bortses som annars finns på 1 meters avstånd enligt följande, förenklade formel.

$$TL = 20\log(R) \quad (2.8)$$

Där TL står för transmission loss i dB och R är avståndet ifrån källan. SPL kan sedan beräknas på vilket avstånd som helst genom att subtrahera (2.9) ifrån (2.8). På ett avstånd av 1mm mellan givare och hydrofon, med frekvensen satt till 10kHz och en amplitud på $4V_{p-p}$ beräknades ljudnivån i dB. Detta blev 180dB vilket också stämde relativt bra överens med den uppmätta ljudnivån på 183dB. Inspänning kalibrerades utifrån detta.

$$SPL = |M_x| - G + 20\log(V) \quad (2.9)$$

Där M_x är hydrofonens mottagar-känslighet i db re. $1V/\mu Pa$, G är förstärkning i dB och V är den i oscilloskopet uppmätta spänningen i volt från hydrofonen.

Eftersom givaren har en varierande transmissions-känslighet sett till frekvensen måste alltså låga frekvenser förstärkas mer än de högre (se figur 2). Detta åstadkommes med förstärkningen i kretsen. Utifrån figur 2 och ekvation 2.8 beräknades det ut via Matlab att förstärkningen behövde avta med -12db/oktav vilket är ekvivalent med -40db/dekad . Detta motsvarar ett andra ordningens aktivt filter med en låg brytfrekvens för att få -40db/dekad kurvan. De båda stegen förstärkte signalen med en ratio på 100:1 samt 2,5:1, det andra steget monterades med en potentiometer för att ytterligare kunna reglera förstärkningen om nödvändigt. Tester gjordes senare i vattentanken för att bestämma att den faktiska källstyrka verkligen låg på 120dB. Några exempel på drivspänningar, relativt frekvensen var för en 10kHz signal $4V_{p-p}$ och för en 12kHz signal $3,5V_{p-p}$.

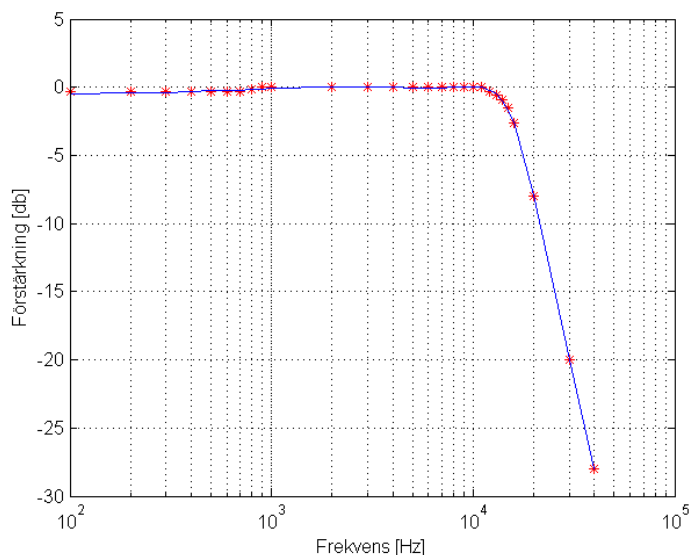
OP-förstärkaren TL072 ifrån Texas instrument används både i lågpassfiltreringen samt till förstärkning. fördelen med denna är att den har låg strömförbrukning och lågt brus vilket gör att den passar bra som audio-förstärkare i detta sammanhang. Bandbredden är också tillräcklig för detta projekt. Ytterligare en fördel är att den är av dual typ, alltså har två förstärkare per komponent vilket sparar plats på kopplingsplattan. [13]

3 Resultat

I följande avsnitt presenteras resultatet ifrån arbetet. Fokus ligger på frekvensanalysen av visslingarna.

3.1 Lågpasfiltreringen

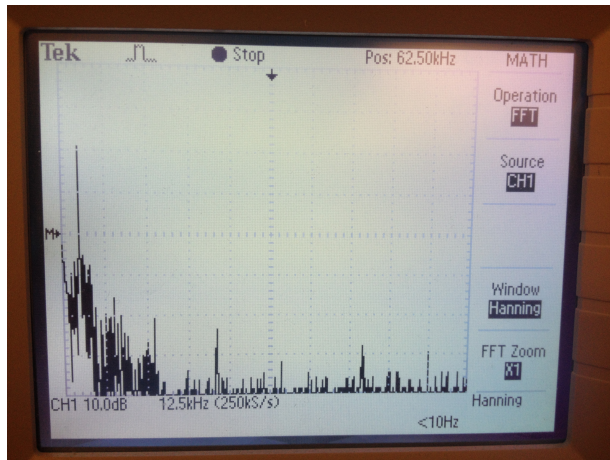
Mätningar på kretsens lågpasfilter och förstärkningssteg genomfördes. I figur 22 ses lågpasfiltreringen. Det som kan ses är att dämpningen är -20db/oktav, idealt sett borde ett fjärde-ordningens filter ha en dämpning på -22db/oktav. Brytfrekvensen ligger vid 16,2kHz. Bodediagrammet i figur 22 togs fram genom att använda en funktionsgenerator med en konstant amplitud och stega frekvensen från 100Hz-40kHz, samt då samtidigt notera utsignalens amplitud på motsvarande frekvens.



Figur 22: Lågpasfiltreringen på kretsen, förstärkning som funktion av frekvens

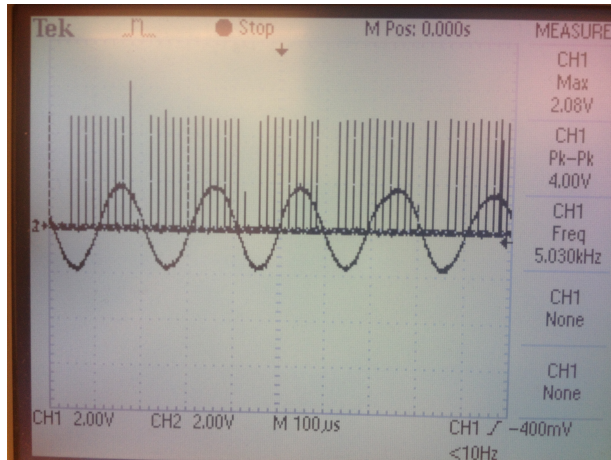
De störningar som existerade innan lågpasfiltreringen har nu minime-

ras till stort sett obefintliga. Ett exempel på detta syns i figur 23.



Figur 23: Frekvensanalys av utsignalen ifrån arduinon efter lågpasfiltreringen av en 5kHz ton, jämför med figur 16

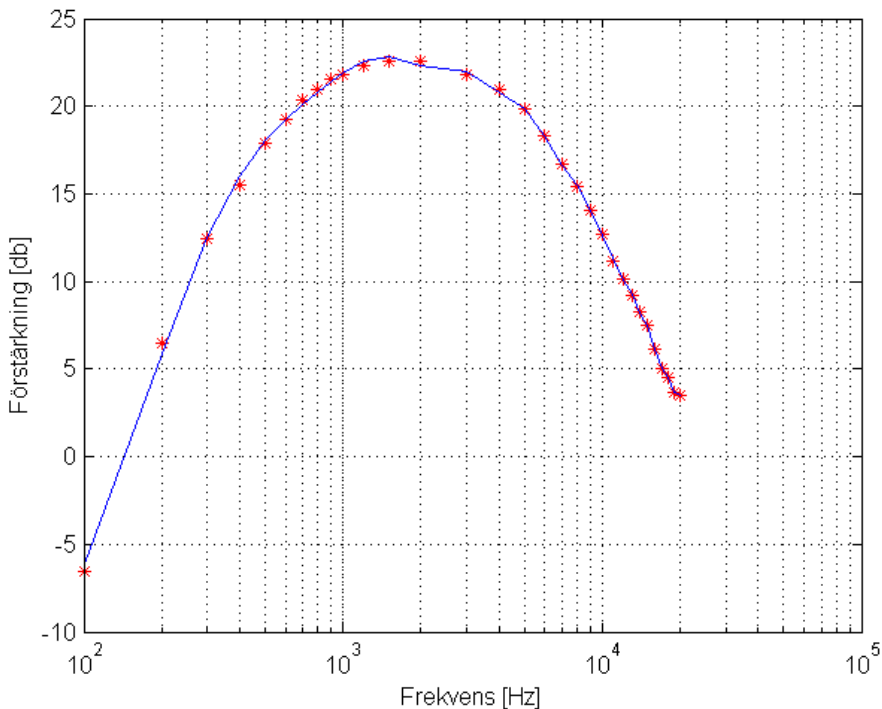
Figur 24 visar en PWM signal med frekvens 5kHz och hur denna signals medelvärde avbildas till en sinus.



Figur 24: Visar utsignalen i oscilloskopet före och efter lågpassfiltrering. I figuren ses den ursprungliga PWM signalen ifrån Arduinon.

3.2 Förstärkning

Förstärkningskurvan togs fram genom att använda en funktionsgenerator med fast amplitud på 2V och sedan stega frekvensen från 100Hz till 20kHz. Se figur 25.

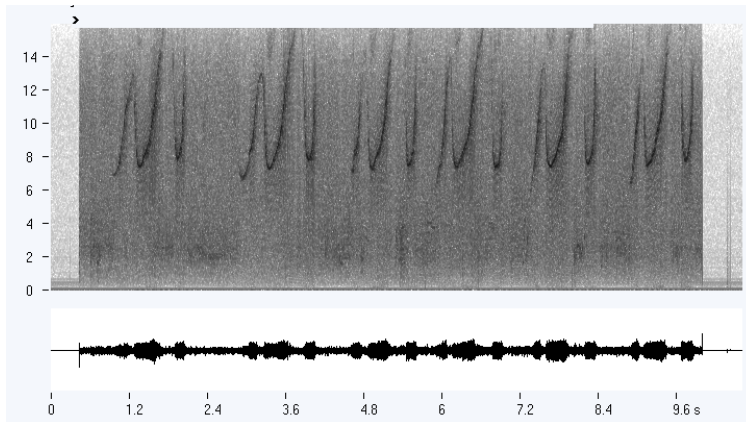


Figur 25: Förstärkning mot frekvens

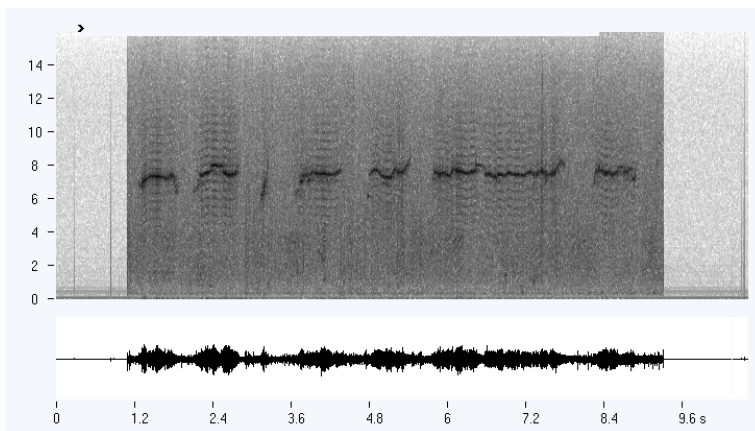
Denna signal skickades in vid AC-kopplingen, detta kan även ses i figur 25 där låga frekvenser dämpas. Detta är väntat då AC-kopplingen är ett högpasfilter. Vidare har vi från 5kHz en dämpning på -9.1 dB/oktav. Idealt ska detta vara -12 dB/oktav.

3.3 Frekvensanalys av visslingarna

I SeaWave togs dessa bilder fram, figur 26 samt 27. Dessa visar frekvensinnehållet i kHz som en funktion av tid. Bilderna är inspelningar av utsignalen från kretsen. Figur 26 visar Vickys signaturvissling och figur 27 Lunas signaturvissling. Dessa figurer är ifrån en tidigare del i projektet då utsignalen lades i form av en PWM signal.

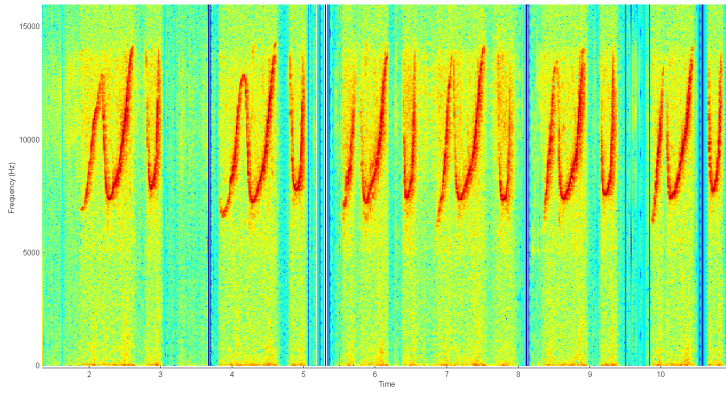


Figur 26: Frekvensanalys av Vickys vissling

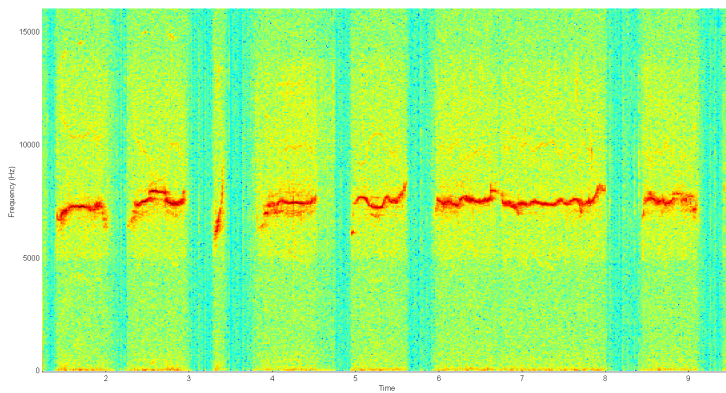


Figur 27: Frekvensanalys av Lunas vissling

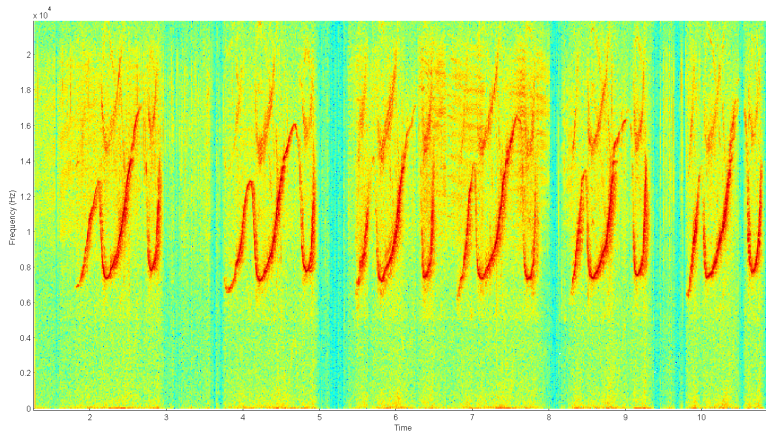
Figur 28 till 31 är inspelningar via TC4013 i ett vattenbad. Signalen i detta fallet är utlagd via D/A-konvertern. I figur 28 och 29 visar ljudfiler där samplingsfrekvens konverteras ner till 32kHz medan ljuden i figur 30 och 31 har behållit den ursprungliga samplingsfrekvensen på 44,1kHz, det är alltså dessa två visslingar som användes i den slutgiltiga prototypen.



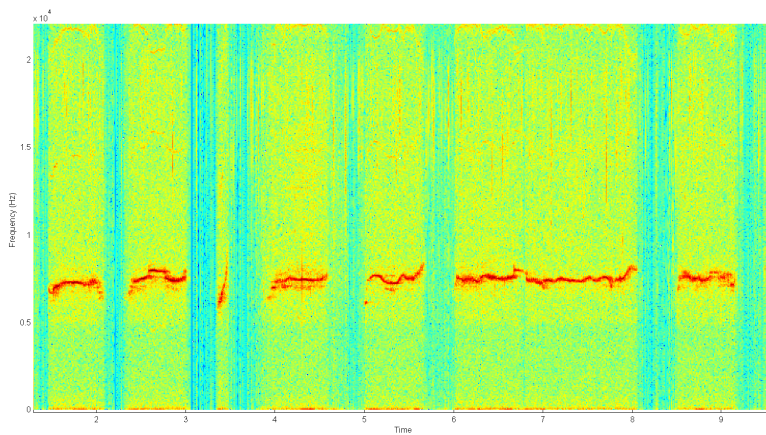
Figur 28: Inspelning av Vicky's vissling via hydrofon



Figur 29: Inspelning av Lunas vissling via hydrofon



Figur 30: Inspelning av Vickys vissling via hydrofon



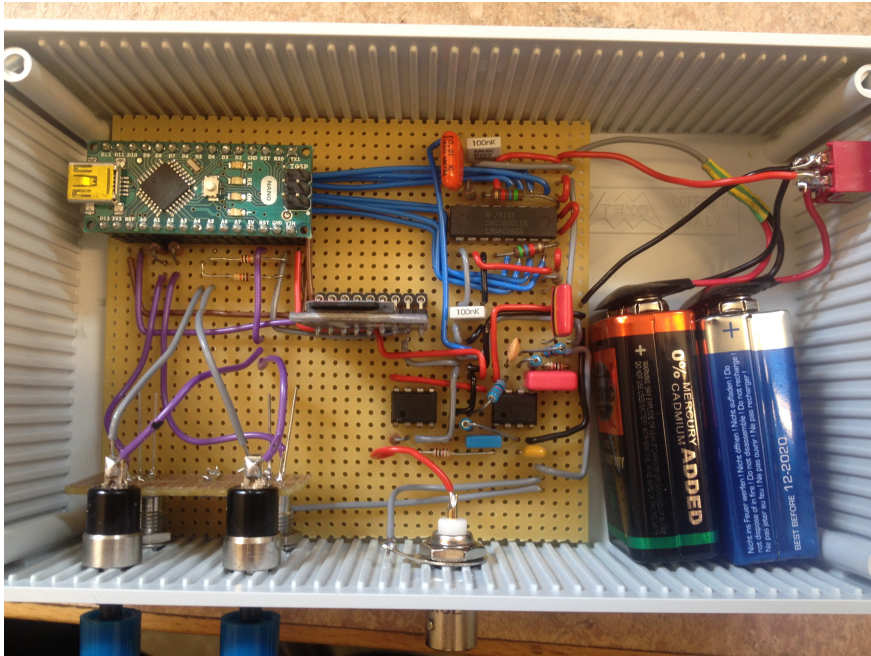
Figur 31: Inspelning av Lunas vissling via hydrofon

3.4 Prototyp

Efter att ha använt en kopplingsplatta under större delen av projektet gjordes en prototyp genom att montera komponenterna på ett veroboard. Utöver detta monterades också två knappar för att välja vissling, en strömbrytare och två lysdioder. Ena lysdioden var röd och markerade om systemet var påslaget, den andra lysdioden var grön och markerade om en vissling sändes. Se figur 32. Om nya visslingar önskas att läggas till görs detta genom att ta ut SD-kortet och kopiera in de nya ljuden. En BNC hona monterades för att ansluta en givare. Två 9V batterier används för att skapa en spänningsmatning på $\pm 9V$. Se gärna kretsschemat i appendix.



Figur 32: Färdig prototyp

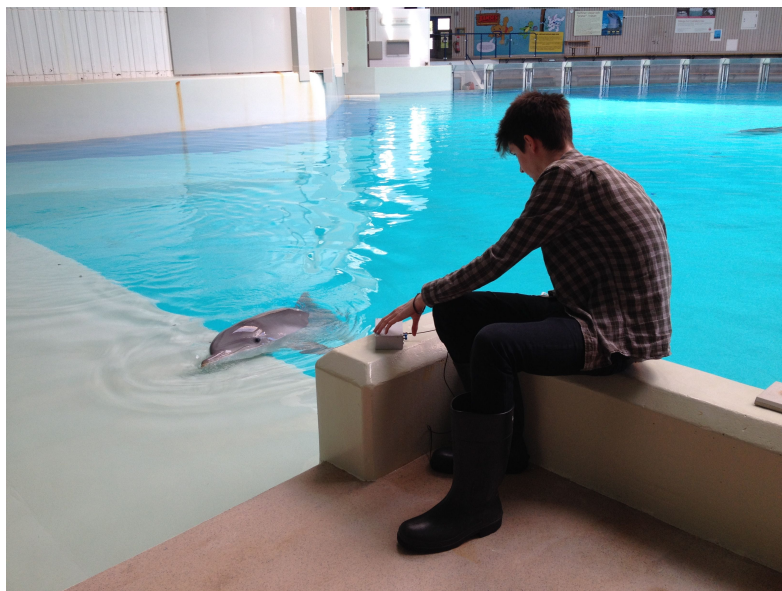


Figur 33: Kretsen i prototypen

3.5 Sluttest i Kolmården

I slutet av projektet gjordes en resa till Kolmården för att testa systemet. Testerna gick till så att de två delfinerna Vicky och Luna fick simma runt i den större bassängen (Kolmårdens delfinarium består av tre bassänger, en där shower visas, en mindre bassäng där veterinärer och skötare kan komma närmare djuren samt den större, s.k Lagunen, där testerna gjordes). Givaren sänktes ner i vattnet och riktades mot antingen Vicky eller Luna, och deras korresponderande signaturvissling spelades upp med en viss repetition. Deras beteende studerades under tiden testet genomfördes av Mats Amundin. Vicky och Lunas respons på signalerna var, visade att de var klart medvetna om visslingarna som skickas ut men de höll sig på ett betryggande avstånd. Delfinerna i Kolmården lever i en skyddad miljö och kan agera misstänksamt mot nya föremål eller ljud. Trots detta ses expe-

rimentet som lyckat då en viss respons erhöles. Det var förväntat att varje delfin måste tränas att betrakta sin egen signaturvissling som inkallnings-signal. Förhoppningen är att denna inläring kommer att ske snabbare än om man valt arbiträra ljud.



Figur 34: Författaren i full fart med tester. Fotograf Monica Almqvist

4 Diskussion

Valet av arduino som plattform för projektet hade sina begränsningar när utsignalen var i PWM-format. Framst p.g.a. att de ursprungliga ljudfilerna samplingsfrekvens fick konverteras ner till 32kHz och därmed att brytfrekvensen för lågpassfiltreringen fick läggas på 16kHz för att undvika vinkning. Detta var inte optimalt då visslingsfrekvenserna spänner upp till 20kHz. Det innebär att frekvensinnehållet i utsignalen blir ofullständig. Detta kan ses i Vickys signaturvissling, figur 28. De högre frekvenserna som syns i originalfilen (jämför med figur 8) finns ej med. Även i figur 27 kan en intressant störning ses, dessa ”vågor” som ses är till följd av PWM metoden. Med ett annat val av plattform hade detta bekymmer kunnat undvikas. När signalen lades ut via D/A omvandlaren uppkom dock inte detta problem. Detta då port-manipulering använts, vilket tillåter högre hastigheter på input/output-portarna jämfört med PWM. Alltså kunde samplingsfrekvensen 44,1kHz behållas men filerna fick fortfarande konverteras ner från 16-bitar till 8-bitar. Det finns dock ett problem med att använda 44,1kHz som samplingsfrekvens då arduinos klocka på 16MHz måste skalas med en s.k prescaler. Denna prescaler beräknas ut genom ekvation 4.1.

$$prescaler = 16 * 10^6 / (freq * bit) - 0,5 \quad (4.1)$$

Där $16 * 10^6$ är arduinos klockhastighet i Hz, $freq$ är önskad frekvens och bit är antal önskade bitar. Om $freq$ sätts till 44,1kHz och bit sätts till 8 fås $prescaler = 44,85$. Det närmsta heltal som arduinon avrundar detta till är 45. Detta innebär att frekvensen inte kommer att vara exakt 44,1kHz utan ungefär 43,9kHz. Detta är naturligtvis ej optimalt men är vad som har använts. Det är känt att delfiner ej härmar varandras visslingar helt perfekt så detta anses trots allt vara ett godkänt resultat.

Under testet i Kolmården fungerade prototypen mer som en delfin-skrämma än en inkallningsenhet, men enligt tränarna är det dock inget större problem att lära delfinerna att betrakta sina signaturvisslingar som

inkallningssignal.

4.0.1 Framtida arbete

Det var som sagt problematiskt att mäta den exakta källstyrkan då detta är definierat på 1 meters avstånd. Detta löstes genom att bortse från transmissionsförlusterna ekvation 2.8. En mätning på exakt 1 meters avstånd är önskvärt för framtida arbete för att få ett korrekt värde på källstyrkan.

För att tränarna i Kolmården i framtiden ska ha en produkt de kan använda dagligen är det flera förbättringar och problem som måste lösas. En av de större utmaningarna är att välja givare. Då de flesta givare ej har samma transmissionskaraktäristika egenskaper så måste förstärkningen bestämmas utefter vald givare. Detta åstadkomes lättast genom att förbereda filerna digitalt i matlab. Samt att använda en linjär förstärkning i själva kretsen i 5kHz-20kHz-området. Givaren bör då väljas efter dess transmissions känslighet och en lämplig kandidat är SRD HS/70. Utöver detta så måste användaren kunna välja mellan fler signaturvisslingar än vad som är möjligt med denna prototyp (2 stycken). Ett förslag från Mats Amundin är en vridkontakt med flera steg men andra lösningar bör också begrundas då det ska vara användarvänligt att hantera enheten med våta fingrar. Enheten måste också vara helt vattentät. För att det ska hända måste kretsen i första hand minimeras. Författaren har också tankar på att inkludera andra sensorer, t.ex en fuktsensor som varnar om den vattentäta behållaren skulle läcka in vatten. Valet med arduino som plattform har fungerat relativt väl, fördelarna är att den är lätt att programmera och det går snabbt att undersöka om kodningen fungerar. Den har dock visat sig ha sina begränsningar vilket jag hoppas framgår i diskussionen. För en vidareutveckling av arbetet bör också arduino som plattform begrundas ytterligare.

5 Appendix

5.1 Arduino-koden

```
#ifndef cbi
#define cbi(sfr, bit) (_SFR_BYTE(sfr) &= ~_BV(bit))
#endif
// Defineras för setting register bit
#ifndef sbi
#define sbi(sfr, bit) (_SFR_BYTE(sfr) |= _BV(bit))
#endif

#include <SD.h>
#include <SPI.h>

Sd2Card card;
float t=0;
int chipselect=10;
int freq=44100;
float bitval=8;
File tempfile;
volatile unsigned long readings,counter;

volatile boolean aready,readit; //readit, variabel om file read
    pågår. aready om bufa eller b ska läsas
volatile boolean playing=false;
#define BUF_SIZE 256
uint8_t bufa[BUF_SIZE];
uint8_t bufb[BUF_SIZE];
volatile uint16_t bufcount;

void setup(){
    delay(1000);
```



```

if(!SD.begin()){
    return;
}
DDRD=0xFF;
pinMode(chipselect,OUTPUT);
pinMode(A0,OUTPUT);
pinMode(A1,OUTPUT);
byte setocroa=(16000000/(freq*bitval))-0.5; //måste vara
mindre än 256
cli();//inaktiverar interrupts
// timer2 interrupt
TCCR2A = 0;// TCCR2A register sätts till 0
TCCR2B = 0;// samma för TCCR2B
TCNT2 = 0;// räknaren sätts till 0
// compare match register sätts

OCR2A=setocroa;
// sätter på CTC mode
TCCR2A |= (1 << WGM21);
// sätter CS21 bit för 8 prescaler
TCCR2B |= (1 << CS21);
sbi(TIMSK2,OCIE2A); // tillåter interrupt på timer 2

sei();//tillåter interrupts
aready=true;
readit=false;
bufcount=0;
counter=0;
card.init(SPI_FULL_SPEED, chipselect);
digitalWrite(A1,HIGH);

}

```

```

ISR(TIMER2_COMPA_vect){ //interrupt rutin timer 2

if(playing){
  if (counter<readings) {
    if(aready) {
      PORTD=bufa[bufcount];
    } else {
      PORTD=bufb[bufcount];
    }

    counter++;
    bufcount++;
    if (bufcount==BUF_SIZE){
      if(readit==false){ // läser ej fil
        bufcount=0;
        aready = ! aready;
        readit=true;
      } else {
        // fil läsning pågår
        counter--;
        bufcount--;
      }
    }
  } else {
    cli();
    PORTD=128;
    playing=false;
    tempfile.close();
    digitalWrite(A0,LOW);
    sei();
  }
}
}

```

```

void loop(){
  if(readit){
    if (! aready){
      // läs in till bufa
      tempfile.read(bufa, BUF_SIZE);
    } else {
      // läs in till bufb
      tempfile.read(bufb, BUF_SIZE);
    }
  }
  readit=false;
}

if(analogRead(A4)==LOW){
  tempfile=SD.open("/soundfile/LUNA44.WAV");
  tempfile.seek(44);
  readings=tempfile.size();
  counter=0;
  playing=true;
  readit=true;
  digitalWrite(A0,HIGH);

}

if(analogRead(A3)==LOW){
  tempfile=SD.open("soundfile/VICKY44.WAV");
  readings=tempfile.size();
  tempfile.seek(44);
  counter=0;
  playing=true;
  readit=true;
  digitalWrite(A0,HIGH);
}

}

```


Referenser

- [1] Cranford, T. Amundin, M. Krysl, P. 2015: Sound Production and Sound Reception in Delphinoids. In: Dolphin Communication & Cognition: Past, Present & Future", edited by Denise Herzing & Christine Johnson. MIT press.
- [2] Whitlow. W.L. Au. och Mardi C. Hastings, Principles of Marine Bioacoustics. 2008.
- [3] "Wav header." <http://www.codeproject.com/Articles/22361/Play-and-Visualize-WAV-Files-using-Managed-Direct>
- [4] IBM Corporation and Microsoft Corporation, "Multimedia programming interface and data specifications 1.0," August 1991. <http://www-mmssp.ece.mcgill.ca/Documents/AudioFormats/WAVE/Docs/riffmci.pdf>
- [5] Linear Technology, "Ltspice iv, programvara." <http://www.linear.com/designtools/software/>
- [6] G. Pavan, "Seawave - sound emission analyzer wave edition, programvara." <http://www-3.unipv.it/cibra/seawave.html>
- [7] Waveshare electronics, "Micro sd storage board," 2014. <http://www.wvshare.com/product/Micro-SD-Storage-Board.htm>
- [8] SanDisk, "Sandisk secure digital card, product manual," 2003. <http://www.circlemud.org/jelson/sdcard/SDCardStandardv1.9.pdf>
- [9] T. Kugelstadt, "Active filter design techniques," 2008. <http://www.ti.com/lit/ml/sloa088/sloa088.pdf>
- [10] David M. Alter, "Using pwm output as a digital-to-analog converter," 2008. <http://www.ti.com/lit/an/spraa88a/spraa88a.pdf>

[11] Texas instruments, “Dac0800/dac0802 8-bit digital-to-analog converters,” 1999.

<http://www.ti.com/lit/ds/symlink/dac0800.pdf>

[12] Teledyne Reson, “Tc-4013 datablad.”

<http://www.teledyne-reson.com/products/hydrophones/tc-4013/>

[13] Texas instrument, “Tl07x low-noise jfet-input operational amplifiers,” 2014.

<http://www.ti.com/lit/ds/symlink/tl074.pdf>

[14] Whitlow. W.L.Au, The Sonar of Dolphins. 1993.

[15] Lewis Loflin.

http://www.bristolwatch.com/PIC18F2550/PIC18F2550_pwm.htm