



EPS

Escola Politècnica
Superior

Treball final de grau

Estudi: Grau en Enginyeria Electrònica Industrial i Automàtica

Títol: Detecció i generació d'ones acústiques subaquàtiques

Document: 1. Memòria

Alumne: Jordi Gratacós Prunell

Director/Tutor: Marc Carreras Pérez

Departament: Arquitectura i Tecnologia de Computadors

Àrea: ATC

Convocatòria (mes/any): setembre/2013

1. INTRODUCCIÓ	4
1.1. Antecedents.....	4
1.2. Objecte	4
1.3. Especificacions i abast	4
2. EXPLICACIÓ DEL PROBLEMA.....	5
2.1. Fonaments teòrics	5
2.1.1. Emissor acústic	5
2.1.2. Receptors acústics	6
2.2. Tècniques de filtratge	7
2.2.1. Passa baix.....	7
2.2.2. Passa alt.....	7
2.2.3. Passa banda.....	8
2.2.4. Notch	8
2.3. Tècniques d'adquisició del temps	8
2.3.1. Mètode de comptatge	9
2.3.2. Anàlisi del domini freqüencial.....	9
2.3.3. Correlació Creuada.....	12
2.4. Metodologia d'obtenció de l'angle d'emissió	14
3. HARDWARE ANALÒGIC	15
3.1. Els hidròfons.....	16
3.2. Amplificació	17
3.2.1. Amplificació	17
3.2.2. Alimentació	17
3.2.3. Ample de banda (Bandwidth).....	18
3.2.4. Slew Rate	18
3.3. Components del circuit	18
3.3.1. Amplificadors operacionals	18
3.3.2. Amplificadors d'instrumentació	19
3.3.3. Potenciòmetres digitals.....	19
3.4. Possibles configuracions	20
3.4.1. Amplificador operacional amb resistència.....	20

3.4.2.	Amplificador operacional amb potenciòmetre	20
3.4.3.	Amplificador d'instrumentació amb potenciòmetre	20
3.5.	Disseny final amplificació etapa entrada hidròfons.....	21
3.6.	Disseny amplificació emissor acústic	21
4.	Hardware Digital	23
4.1.	Introducció	23
4.1.1.	D'analògic a digital	23
4.1.2.	De digital a analògic	24
4.1.3.	Tècniques de mostrejat	24
4.2.	El microcontrolador.....	25
4.2.1.	Característiques	25
5.	DISSENY DEL PROGRAMA	27
5.1.	Configuració dels mòduls dsPIC	27
5.1.1.	Mòdul UART	27
5.1.2.	Mòdul I ² C.....	28
5.1.3.	Mòdul ADC	28
5.1.4.	Comptadors.....	28
5.2.	Estructura del procés.....	28
6.	PROVES REALITZADES	30
6.1.	LabVIEW	30
6.1.1.	Hardware.....	30
6.1.2.	Programa.....	32
6.1.3.	Problemes	32
6.1.4.	Conclusions.....	32
6.2.	Microcontrolador.....	33
6.2.1.	Hardware.....	33
6.2.2.	Problemes	34
6.2.3.	Conclusions.....	34
7.	POSSIBLES APLICACIONS.....	35
8.	RESUM DEL PRESSUPOST.....	36
9.	CONCLUSIONS	37
10.	RELACIÓ DE DOCUMENTS	38

11. BIBLIOGRAFIA.....	39
12. GLOSSARI	41
A. CODI DEL PROGRAMA	43

1. INTRODUCCIÓ

L'objectiu d'aquest projecte és crear un sistema independent per un AUV (Autonomous Underwater Vehicle) que sigui capaç d'escoltar i localitzar un so subaquàtic, i de la mateixa manera, generar-lo. Aquest sistema envia una trama amb la informació de l'angle a l'AUV, permetent aquest viatjar fins al so. Aquest projecte inclou el coneixement i la implementació de diferents àrees d'electrònica. El sistema utilitza tant hardware analògic com digital, juntament amb càlculs matemàtics per computar la informació entrant.

1.1. Antecedents

En el centre de recerca en robòtica submarina de la UdG es desitja desenvolupar un sistema per a la localització d'un emissor acústic subaquàtic que permeti la localització d'emissors i alhora que actuï d'emissor actiu.

1.2. Objecte

L'objecte d'aquest projecte és el disseny, la construcció i la verificació d'un sistema electrònic que permeti la localització d'un emissor acústic subaquàtic i que pugui actuar també com a emissor actiu.

1.3. Especificacions i abast

El sistema utilitzarà varis hidròfons per a adquirir les ones acústiques d'un emissor i detectar-ne la diferència de fase. Mitjançant aquest procediment, es podrà determinar la direcció o localització de l'emissor, depenent del nombre d'hidròfons utilitzats. També es desitja que el sistema desenvolupat pugui actuar com a emissor actiu, és a dir, que generi les ones acústiques després d'haver rebut un senyal acústic. La utilització d'emissors actius permet desenvolupar sistemes de localització basats en rang. El software de la placa permetrà la detecció de la diferencia de fases dels senyals adquirits amb els hidròfons, la determinació de la procedència de l'emissor acústic i la generació d'ones acústiques actuant tot el sistema com a emissor actiu. L'abast del projecte contindrà el disseny, construcció, programació i verificació de la placa electrònica. Es realitzaran experiments de validació en les instal·lacions del laboratori i en el mar. La placa té la concepció de prototipus i només es validarà en distàncies curtes.

2. EXPLICACIÓ DEL PROBLEMA

En aquest capítol s'explicarà la naturalesa del problema i diferents tècniques que s'han utilitzat per aconseguir el resultat desitjat.

El problema es basa en detectar la direcció d'una font emissora E mitjançant diferents receptors que estan ficats en una mateixa plataforma, en aquest cas, per simplificar la figura H1 i H2, en un medi aquàtic. L'emissor E emet una ona a una determinada freqüència que es propaga en totes les direccions (circumferències de color lila). A mesura que va passant el temps l'ona es va propagant fins a arribar al receptor H2 amb un temps t_2 i al cap d'un temps t_1 aquesta arriba al receptor H1. Mitjançant la diferència d'aquests temps i aplicant mètodes matemàtics explicats a continuació, es pot determinar l'angle de l'emissor respecte la plataforma o robot. En la figura següent es pot veure l'ona que ha enviat l'emissor acústic i la que reben els receptors.

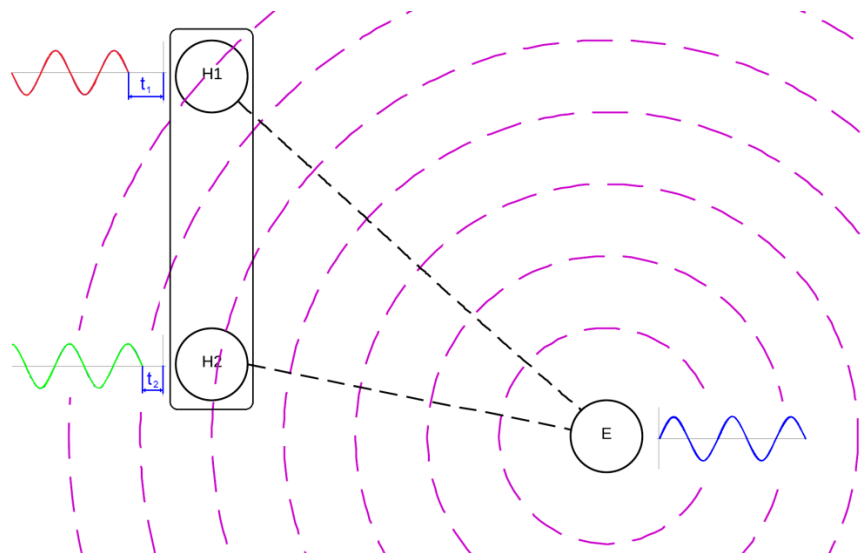


Figura 1. Diagrama explicatiu del problema

2.1. Fonaments teòrics

Els sons produïts per animals marins, processos naturals i activitats produïdes pels éssers humans omplen l'oceà. L'aigua és un medi efectiu per la transmissió del so, animals marins i persones usen el so com una eina per trobar objectes, navegar i comunicar-se sota l'aigua.

2.1.1. Emissor acústic

L'emissor acústic projecta al medi un senyal, en aquest cas és un senyal sinusoidal de cinc milisegons de duració i tres mil hertz de freqüència i el va repetint de forma periòdica cada segon. Aquesta és la freqüència utilitzada, però es pot canviar per qualsevol altre dins dels rangs que admet el sistema explicats més endavant.

2.1.2. Receptors acústics

Els receptors acústics solen ser hidròfons, aquests són explicats en més detall a l'apartat tres punt u. Aquests, per un correcte funcionament, han d'estar separats com a màxim una distància màxima igual a la meitat de la longitud d'ona. Aquesta es pot escriure com:

$$\lambda = \frac{c}{f} \quad (\text{Eq. 1})$$

On λ és la longitud d'ona, c és la velocitat del so a l'aigua i f la freqüència de l'ona.

La restricció es pot provar ja que per determinar la diferència de temps d'arribada de l'ona als dos hidròfons, hem de limitar-la a $\Delta t \leq \frac{1}{2f}$. En el domini temporal és de bon entendre, si s'observa la figura següent, es pot veure que hi ha una ambigüitat a l'estimar la diferència de temps entre les dues senyals degut a la periodicitat del senyal.

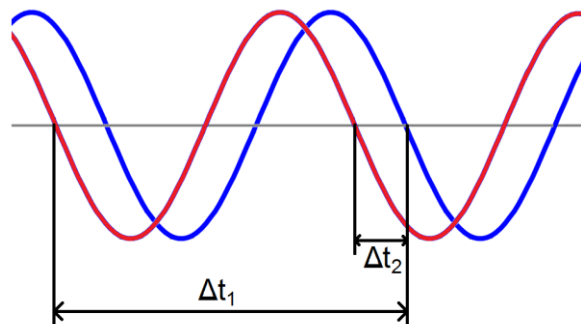


Figura 2. Dos senyals sinusoidals de mateixa freqüència però diferent fase

Si la relació espai temps es té en compte, podem escriure l'equació de la diferència de temps com:

$$\Delta t = \frac{\Delta d}{c} \leq \frac{1}{2 \cdot f} \quad (\text{Eq. 2})$$

I tenint en compte l'equació de la longitud d'ona, aquesta ens queda simplificada com:

$$\Delta d \leq \frac{\lambda}{2} \quad (\text{Eq. 3})$$

On Δt és la diferència de temps entre les dues fases, Δd és la diferència entre els dos receptors, c és la velocitat del so, f la freqüència del senyal i λ la longitud d'ona. Així d'aquesta manera demostrem la distància màxima de separació entre dos hidròfons.

2.2. Tècniques de filtratge

Un dels requeriments del sistema és que aquest sigui capaç de trobar senyals amb una determinada freqüència. Per fer això és necessari un filtre que elimini tots els senyals de freqüències no desitjades. Existeixen diferents tipus de filtres, en els subapartats següents s'explicarà cadascun d'ells i es discutirà la seva habilitat per treballar en el sistema

2.2.1. Passa baix

El passa baix és un tipus de filtre que permet passar totes les freqüències més baixes que la definida. Això vol dir que si una freqüència és lo suficientment baixa passarà a través del filtre, sinó serà atenuada. Aquest tipus de filtre és molt útil quan només s'espera tenir sorolls d'alta freqüència. El filtre no és apte pel sistema ja que en el mar tenim sorolls tant d'alta com de baixa freqüència.

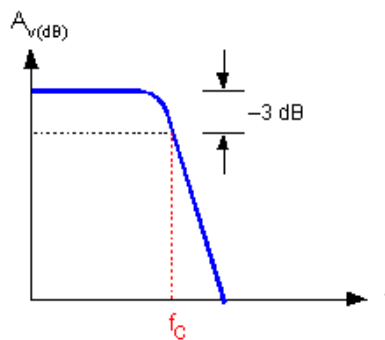


Figura 3. Filtre pas baix

2.2.2. Passa alt

El passa alt és un tipus de filtre que permet passar totes les freqüències més altes que la definida. Això vol dir que si una freqüència és lo suficientment alta passarà a través del filtre, sinó serà atenuada. Aquest tipus de filtre seria molt útil si el nostre soroll només vingués d'una sola font, tal com passa al corrent altern (font de cinquanta hertz). El filtre no funcionarà ja que en el mar tenim sorolls tant d'alta com de baixa freqüència.

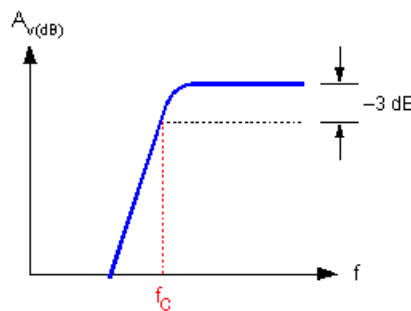


Figura 4. Filtre pas alt

2.2.3. Passa banda

El passa banda és un tipus de filtre que permet passar totes les freqüències d'una banda determinada. Les freqüències que es troben dins d'aquesta banda poden passar, en canvi les altres queden atenuades. Aquest filtre és el més adequat pel sistema, ja que si creem una banda en la qual s'hi trobi la freqüència del nostre emissor, tots els sorolls i freqüències no desitjades les eliminarem, quedant-nos pràcticament només amb la de l'emissor.

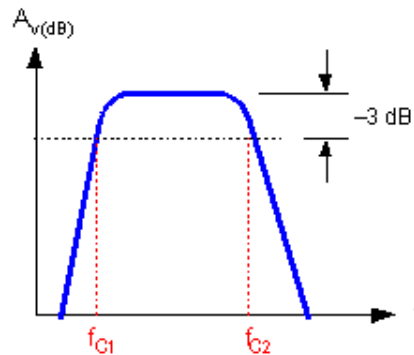


Figura 5. Filtre pas banda

2.2.4. Notch

El filtre notch és un tipus de filtre que permet passar totes les freqüències menys les d'una banda determinada. Les freqüències que es troben dins d'aquesta banda les atenua, en canvi les altres poden passar. Aquest filtre és útil si es necessita eliminar una freqüència determinada, però en el nostre cas és l'acció oposada la que volem dur a terme.

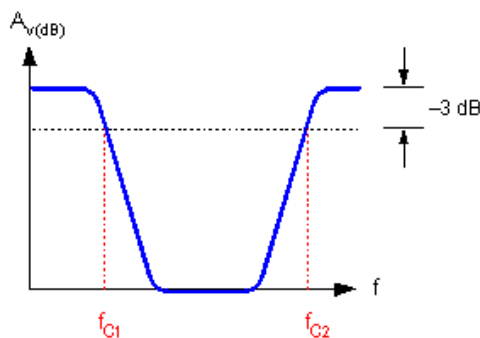


Figura 6. Filtre Notch

2.3. Tècniques d'adquisició del temps

Adquirir de forma sincronitzada el senyal de l'emissor per a cada hidròfon requereix hardware especialitzat i diferents tècniques. S'han buscat i estudiat diferents alternatives per poder resoldre aquest problema. Cadascun d'aquests mètodes té els seus punts forts i febles basats en la complexitat, precisió i adaptabilitat.

2.3.1. Mètode de comptatge

Un dels mètodes més simples implementats és fer servir un comptador d'alta velocitat per mesurar les diferències de temps de cada hidròfon. Aquest mètode utilitza la sortida analògica de cada hidròfon i aquest és comparat amb un voltatge de referència que dispara un comptador. Un dels rellotges del microcontrolador és l'encarregat per incrementar els comptadors quan el comparador s'activa. La sortida de cada comparador ha d'iniciar un registre amb el valor de temps actual. Després de que tots els comparadors s'hagin activat, aquests registres seran interpretats pel processador i s'implementaran les rutines per la localització de l'ona acústica.

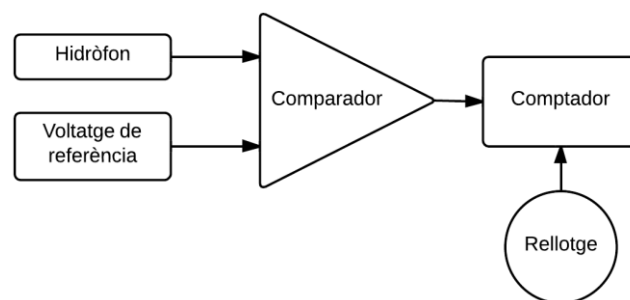


Figura 7. Diagrama de blocs del mètode per comptatge

El mètode explicat és senzill, tot i així és sensible en ambients sorollosos. Si hi ha soroll que no és provinent de l'emissor acústic, però té la suficient magnitud per activar el comparador, obtindríem un fals positiu. Aquest temps està basat en la condició de que cada element del sistema és el mateix, inclosos els hidròfons. En el món real aquesta afirmació és falsa, ja que cada element té unes toleràncies donades pels fabricants i la suma de totes aquestes toleràncies al final del sistema obtindríem mesures diferents, cosa que afectaria el resultat final. Degut a l'exposat anteriorment el mètode de comptatge no és el més adequat pel nostre problema.

2.3.2. Anàlisi del domini freqüencial

Com que una de les especificacions conegudes és la freqüència de l'emissor acústic, la utilització de l'anàlisi del domini freqüencial pot ser útil per filtrar freqüències no desitjades i obtenir una alta fiabilitat dels temps de l'hidròfon. Si disposem els hidròfons de manera que cadascun està dins de la meitat de la longitud d'ona (λ) de la freqüència de l'emissor, l'anàlisi de fase pot ser calculat per aconseguir la informació dels temps del sistema. Aquesta tècnica introdueix una aproximació diferent al problema i a la transformació dels paràmetres d'entrada per una solució més acurada.

La transformació de les dades des del domini del temps al domini freqüencial és el primer pas perquè les dades puguin ser analitzades. Això s'aconsegueix mitjançant la transformada de Fourier, per ser més específics la transformada ràpida de Fourier (FFT). Per tal de poder analitzar en el domini freqüencial amb hardware digital, el que primer s'ha de fer és capturar el senyal continu analògic tant en voltatge com amb temps i passar-lo a valors discrets digitals. Això s'aconsegueix mitjançant un convertidor analògic digital (ADC). Això requereix que el sistema necessiti filtres que atenuïn les freqüències més altes, més enllà del que l'hidròfon pot escoltar. Per tal de dur a terme l'anàlisi en la següent figura podem veure de forma esquemàtica el hardware necessari.

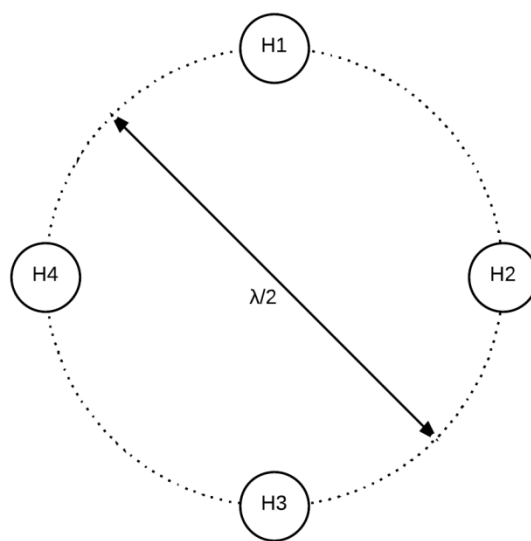


Figura 8. Exemple de configuració per a quatre hidròfons que mantenen entre ells una distància màxima de λ mitjos

Al fer servir aquest sistema dependent del medi, on la magnitud de l'emissor acústic depèn de la distància de la font, el senyal rebut no és un valor constant, i pot incrementar de tal manera que saturi els amplificadors. Per això, l'amplificador necessita configurar-se dinàmicament, quan més gran és la distància entre l'emissor i el receptor més gran és l'atenuació que rep el senyal. Però quan el sistema està a prop de l'emissor, més petita és i per tant llegeix valors més alts, i s'ha d'aconseguir que l'amplificador no es saturi, això ho aconseguirem disminuint i augmentant el guany d'aquest.

Un cop el senyal és correctament amplificat, filtrat, i digitalitzat, és el moment de processar-lo. Assegurant-nos que el nombre de dades que tenim és una potència de dos (requeriment per la FFT), podem aplicar la FFT a les dades adquirides. El resultat ens dona dues components per cada freqüència, la part real i la imaginària. Es calcula la potència de les dues de la següent forma:

$$P_i = \sqrt{\Re_i^2 + \Im_i^2} \quad (\text{Eq. 4})$$

On P és la potencia, $\Re$ és la part real, i $\Im$ és la part imaginària en la freqüència i .

Després de calcular la potencia de cadascuna de les freqüències específiques, la freqüència amb una amplitud més gran hauria de ser la freqüència de l'emissor acústic.

$$f_i = i \cdot \frac{f_s}{N} \quad (\text{Eq. 5})$$

On f_i és la freqüència a l'índex i , f_s és la freqüència de mostreig, i N el número de mostres. En aquesta freqüència, la fase de cada hidròfon pot ser calculada per la següent formula:

$$\phi = a \tan \frac{\Im}{\Re} \quad (\text{Eq. 6})$$

On ϕ és la fase de cada senyal de l'hidròfon en radiants. Després de calcular cada fase per separat, aquestes necessiten ser referenciades a un hidròfon en particular per poder calcular la diferencia de temps d'arribada. Aquesta serà considerada com a 0, mentre que la resta s'hauran d'ajustar en referencia a aquesta. Un cop calculades, es pot procedir a trobar la diferencia de temps.

$$t_{x,y} = \frac{\phi_{x,y}}{2\pi} \cdot \frac{\lambda}{v} \quad (\text{Eq. 7})$$

I a partir de la equació numero 7 obtenim la diferència de temps en funció de l'angle i la freqüència del senyal.

$$t_{x,y} = \frac{\phi_{x,y}}{2\pi} \cdot \frac{1}{f} \quad (\text{Eq. 8})$$

Aquests resultats poden ser usats en els mètodes de càlcul per la localització de la direcció de la font emissora. Aquest mètode sembla robust gracies al filtratge del soroll a diferents freqüències, però és un requisit que el senyal captat sigui el més proper a una senoide i això no és veritat en la pràctica. Els errors augmenten quan calculem la FFT a partir d'un senyal que no és purament sinusoidal degut als acoblaments en el domini freqüencial. Si la

freqüència del senyal no és una de les múltiples freqüències de la transformada de Fourier esdevindrà un desbordament als components de la freqüència adjacent, dificultant el procés per trobar la fase.

2.3.3. Correlació Creuada

En el processament del senyal, el procés que mesura la diferència entre dues ones desfasades entre elles es coneix amb el nom de correlació creuada. Per a funcions discretes, la correlació creuada es defineix com:

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f^*[m] \cdot g[n + m] \quad (\text{Eq. 9})$$

On $f * g[n]$ és la correlació creuada en el punt n , f és el primer senyal, f^* és el complexa conjugat del senyal f i g és el segon senyal. La correlació creuada també es pot calcular mitjançant la fórmula de la convolució però amb un senyal invertit. On N és la suma de les dimensions dels dos vectors.

$$(f * g)[n] = \sum_{m=0}^{N-1} f[m] \cdot g[n - m] \quad (\text{Eq. 10})$$

Si agafem el senyal que ens donen dos hidròfons que són separats a una determinada distància. Agafant com a base que el senyal de l'emissor acústic s'està propagant a través de l'aigua a una determinada velocitat, un dels hidròfons començarà a rebre primer que l'altre, no obstant la forma de l'ona rebuda serà exactament la mateixa. La única diferència entre els dos senyals serà la atenuació de l'amplitud del segon hidròfon relatiu al primer. La correlació creuada és un molt bon mètode per aquests tipus de senyals que només estan desfasats entre ells.

Per explicar-ho una mica més clarament, partirem amb una configuració de dos hidròfons. Si els hidròfons estan en línia i paral·lels amb l'emissor acústic, el senyal rebut per cada receptor estarà en fase amb una diferència de temps zero segons per la correlació creuada. Tal com es pot veure a la següent figura, l'emissor E emet una ona, i al cap de t_1 segons arriba alhora als dos receptors, si sobreposem les dues ones rebudes, aquestes tindran un desfasament zero entre elles. Això és degut a que la distància entre els dos hidròfons i l'emissor és la mateixa.

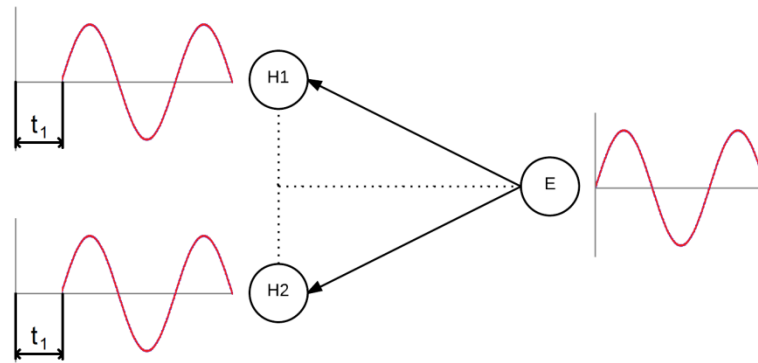


Figura 9. Mostra la configuració de dos hidròfons paral·lels amb l'emissor acústic

Per l'altre extrem, es tindran els dos hidròfons i l'emissor en línia. El senyal es propagarà a través de l'aigua i serà rebut per l'hidròfon més proper i després pel següent. Això produirà la màxima diferència de temps amb aquesta configuració calculada per la correlació creuada. Aquest exemple s'il·lustra en la figura següent. L'emissor E emet una ona, aquesta viatja a través del medi i al cap de t_1 segons arriba al receptor H1, l'ona emissora continua viatjant, i al cap de t_2 segons arriba al receptor H2. Per tant en aquest cas si sobreposem les dues ones rebudes pels receptors, veurem que hi ha un desfasament entre elles equivalent a t_2 segons.

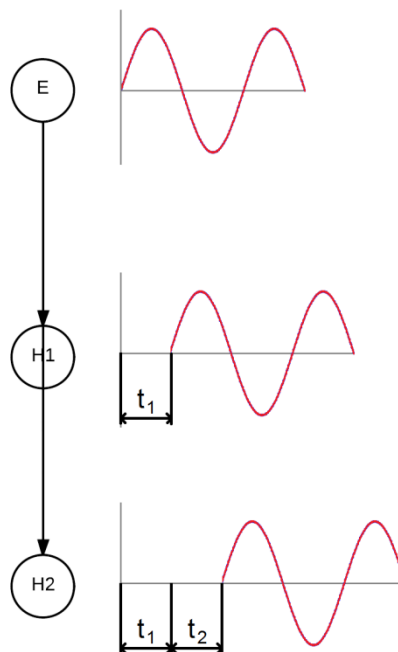


Figura 10. Mostra la configuració dels dos hidròfons i l'emissor acústic en sèrie

Utilitzar la correlació creuada per obtenir el temps per a cada senyal provinent dels hidròfons és un mètode molt robust, ja que no és dependent del senyal que ha de ser un sinus pur,

sinó que pot ser qualsevol tipus d'ona arbitrària. Aquesta també té les seves limitacions, amb el senyal discretitzat, la resolució més alta està basada en la freqüència de mostreig del sistema, per tant, quan més alta sigui la freqüència de mostreig més resolució obtindrem. Aquest s'ha de recolzar amb un sistema que tingui la suficient memòria per capturar la gran quantitat de dades provinents dels quatre hidròfons.

2.4. Metodologia d'obtenció de l'angle d'emissió

Un cop es té la diferència de temps t entre les dues ones, l'últim pas a fer és calcular l'angle de la font emissora, això es pot aconseguir mitjançant trigonometria. A la figura següent es pot veure una ona sonora que és rebuda per dos hidròfons, l'ona primer és rebuda per l'hidròfon dos, i al cap de t segons és rebuda per l'hidròfon u. A la part de sota hi ha el senyal que ha rebut cadascun dels hidròfons en funció del temps.

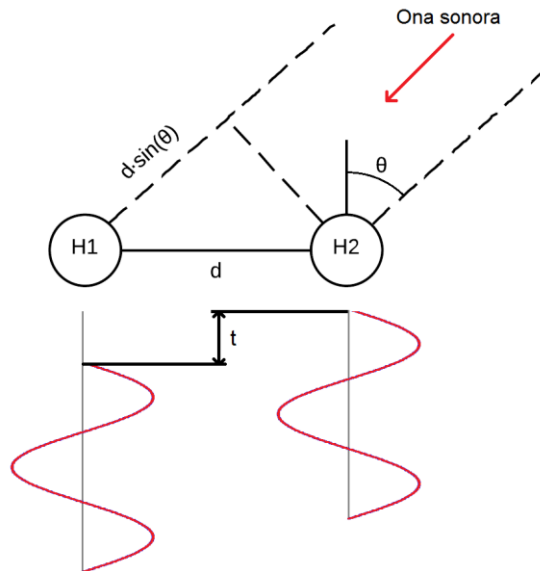


Figura 11. Estimació de l'angle de l'emissor

On t és la diferència de temps d'arribada de l'ona entre els dos hidròfons, c és la velocitat del so a l'aigua, d és la distància de separació entre els dos hidròfons i θ l'angle de procedència de la font emissora. Aquest el podem calcular de la següent manera:

$$\theta = \arcsin\left(\frac{t \cdot c}{d}\right) \quad (\text{Eq. 11})$$

Així d'aquesta manera trobem l'angle de la font emissora

3. HARDWARE ANALÒGIC

L'electrònica és la part essencial d'aquest projecte, aquesta s'agrupa en l'analògica i la digital per formar una placa electrònica. La placa és capaç de captar els senyals provinents dels hidròfons (receptors), amplificar-los perquè el microcontrolador pugui efectuar els seus càlculs i transmetre les dades al AUV. També és capaç de generar una ona per ser transmesa per l'hidròfon.

S'ha escollit efectuar el disseny amb una placa electrònica, ja que ha de ser capaç d'integrar-se en l'AUV, una de les possibilitats que es van tenir en compte era comprar una placa d'adquisició comercial que es comunicés per PCI-e o USB. Aquesta possibilitat va quedar descartada perquè al tractar-se d'un equip prefabricat, no es podia configurar amb les especificacions necessàries, és a dir, no duia amplificació per les entrades i sortides analògiques, això feia que l'aparell a més de les dimensions d'aquest encara s'hi hagués d'afegir una placa electrònica fent-lo més gran del necessari, dificultant-ne la integració amb l'AUV. Un dels altres motius és que havia de ser el més reduït possible, és a dir, ha de ser possible posar-lo dins una caixa estanca i llençar-lo al mar o acoblar-lo a algun vehicle o submarinista. I per últim, que fos un sistema barat.

En la següent figura podem veure el diagrama general de la placa, i com s'estructura. La placa la componen els blocs que estan dins el rectangle ratllat. El primer element que trobem és un interruptor, aquest serveix per escollir si es vol escoltar o generar un senyal per l'hidròfon u. Seguidament trobem les etapes amplificadores, la de sota és la d'entrada, mentre que la de sobre és la de sortida, aquesta veiem que està unida a un bloc anomenat DAC, aquest és el convertidor digital analògic, i finalment el bloc amb el nom μC és el microcontrolador encarregat de processar les dades i enviar els resultats al AUV.

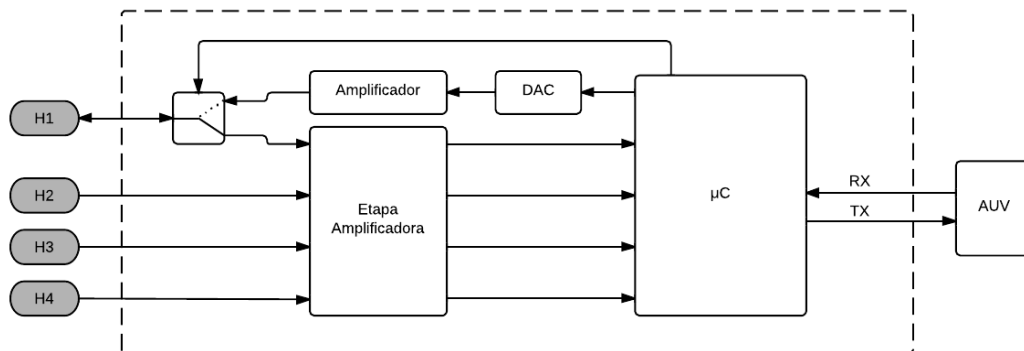


Figura 12. Blocs de configuració de la placa

3.1. Els hidròfons

Un hidròfon és un micròfon especialment dissenyat per al seu funcionament sota l'aigua i així poder escoltar els sons subaquàtics. La majoria dels hidròfons es basen en un transductor piezoelèctric que genera electricitat quan percep un canvi de pressió sobre ell. Alguns transductors poden servir com a emissor, però no a la seva màxima capacitat.



Figura 13. Exemple d'hidròfon comercial H1a de la marca Aquarian Audio

Per tenir un coneixement major sobre aquests elements en la següent imatge es pot observar el model electrònic d'aquest a la següent figura. Destacar la resistència R1, aquesta la recomana el fabricant per descarregar la càrrega acumulada en el piezoelèctric, i alhora per fer un filtre pas baix mitjançant el condensador C_s .

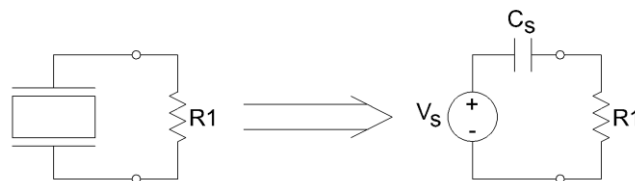


Figura 14. Model electrònic d'un hidròfon piezoelèctric

La formula del filtre pas baix bé donada pel fabricant, però si apliquem la formula general dels filtres podem determinar la freqüència de tall que volem a partir de la resistència escollida.

$$f_c = \frac{1}{2\pi RC} = \frac{1}{0.000000157R} \quad (\text{Eq. 12})$$

On f_c és la freqüència de tall, C és la capacítància de l'hidròfon i R és la resistència escollida.

L'elecció dels hidròfons és molt important. Aquests són els únics sensors que reben els senyals de l'emissor acústic de l'aigua. Les característiques més importants que s'han de comprovar són el rang de freqüències que pot captar, la sensibilitat, el guany, la profunditat en la que poden treballar i les seves dimensions. Seguidament es farà una breu explicació del model que m'ha proporcionat el laboratori, l'hidròfon H1a.

L'hidròfon H1a, el fabrica una empresa anomenada Aquarian Audio, aquest és el model que m'ha proporcionat el laboratori, per poder realitzar les proves. Aquests, amb les proves realitzades, no han presentat cap problema, tot i així per futures aplicacions que necessitin de més profunditat caldria buscar un altre hidròfon, ja que aquest només aguanta fins a vuitanta metres. La taula següent mostra les característiques bàsiques d'aquest.

Freqüència	Sensibilitat	Guany	Profunditat	Dimensions
1Hz fins 100kHz	-190dB a 4dB	0	80m	25x46mm

Taula 1. Especificacions de l'hidròfon H1a

3.2. Amplificació

Els següents subapartats discutiran les diferents característiques que s'han de tenir en compte per la tasca d'amplificació.

3.2.1. Amplificació

Per la localització de l'emissor, el sistema tindrà una entrada de l'ordre de milivolts. Aquest voltatge per poder ser posteriorment tractat, s'haurà d'amplificar unes cinc centes vegades per tal d'aprofitar tota l'entrada del convertidor analògic digital del microcontrolador. Els encarregats de fer aquesta tasca són els amplificadors operacionals, aquests han de ser molt acurats, ja que no poden introduir sorolls o ombres en els senyals entrants provinents dels hidròfons.

3.2.2. Alimentació

La idea principal és que el sistema dissenyat s'incorpori en un dels robots del laboratori, aquests treballen amb unes bateries de Li-ion (Lithium-ion) a vint-i-quatre volts. Un dels problemes és que els amplificadors operacionals treballen amb voltatges positius i negatius alhora, això implica que s'ha d'introduir un regulador de voltatge. Aquest té una entrada de cinc volts, i unes sortides de més cinc volts i menys cinc volts. S'ha escollit alimentar la placa a cinc volts i no vint-i-quatre perquè hi ha diferents elements en el robot que treballen a cinc volts, i així d'aquesta manera no fa falta un altre convertidor. En els esquemes podem observar que hi ha uns elements que tenen com a referència dos coma cinc volts, aquests s'ha optat per aconseguir-los amb un regulador de voltatge i no amb un divisió de tensió, ja que un cop el sistema estigui encapsulat, aquest haurà de resistir temperatures de fins a setanta graus centígrads, i això fa que les variacions de les resistències a causa del calor es descontrolin, en canvi l'integrat ens assegura el voltatge desitjat amb temperatures elevades.

3.2.3. Ample de banda (Bandwidth)

Aquest és un altre atribut important de cares a un sistema per captar senyals a diferents freqüències. Components com els amplificadors operacionals s'han de buscar amb un ample de banda determinat, en aquest cas un ample que com a mínim sigui igual al rang de freqüències de l'hidrófon. Si l'amplificador no te prou ample de banda els senyals s'atenuaran.

3.2.4. Slew Rate

L'slew rate d'un amplificador operacional és l'habilitat que te aquest de canviar el seu voltatge en un determinat temps. Un slew rate elevat permetrà a l'operacional reaccionar davant de senyals ràpids, cosa que és necessari en el cas de la localització de l'emissor acústic. També és important perquè un cop està establert l'ample de banda, l'slew rate establirà les limitacions de l'amplificador. Aquest paràmetre és molt important en l'aplicació perquè el senyal ha de ser el més precís possible.

3.3. Components del circuit

Els subapartats següents es discutirà sobre els components en que alhora de fer la placa es van tenir en compte per tal de poder completar la tasca

3.3.1. Amplificadors operacionals

Un amplificador operacional és un mòdul de circuits electrònics que amplifica la senyal d'entrada mitjançant l'ajuda d'elements externs. Pot actuar com a integrador o com a derivador. En el nostre cas es necessita que actuï com a amplificador, això es pot dur a terme amb unes quantes resistències, aquestes amplifiquen el senyal sense introduir molt de soroll. Aquests amplificadors són usats per moltes aplicacions, així que hi ha molts fabricants que disposen de diferents models.

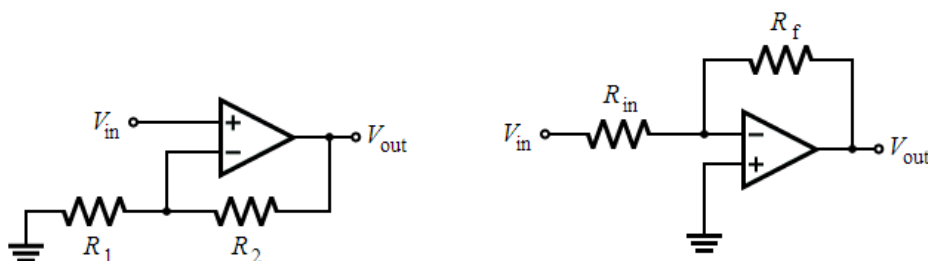


Figura 15. Diferents configuracions per amplificar amb amplificadors operacionals, a l'esquerra l'amplificador no inversor, i a la dreta l'amplificador inversor

3.3.2. Amplificadors d'instrumentació

Aquests són una variació dels anteriors. Són utilitzats en instrumentació electrònica per amplificar senyals molt dèbils que provenen de sensors. Es pot modificar el seu guany de forma analògica, mitjançant la variació d'una única resistència externa o de forma digital, programant un guany en funció d'entrades digitals. Els senyals d'entrada, en els amplificadors normalment poden tenir un mode diferencial i un mode comú. L'amplificador d'instrumentació està dissenyat per obtenir un guany de l'entrada diferencial i reduir el màxim possible el mode comú.

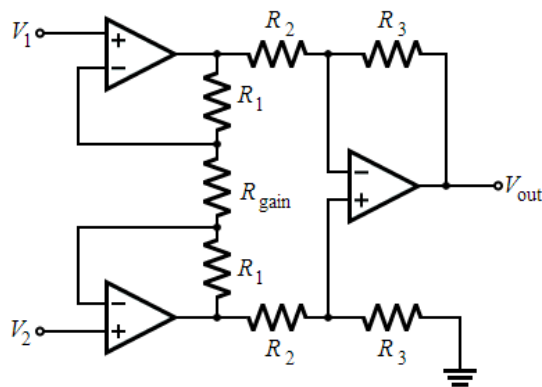


Figura 16. Amplificador d'instrumentació

En la figura anterior es pot veure l'esquema d'un amplificador d'instrumentació, el seu guany es regula mitjançant la resistència R_{gain} . S'ha de tenir en compte que tota aquesta configuració a part de la resistència R_{gain} , està encapsulada dins d'un integrat i R_1, R_2 i R_3 depenent del fabricant són de diferents valors. La fórmula pel càlcul del guany és la següent:

$$V_{\text{out}} = \left(1 + \frac{2 \cdot R_1}{R_{\text{gain}}} \right) \frac{R_3}{R_2} (V_2 - V_1) \quad (\text{Eq. 13})$$

3.3.3. Potenciòmetres digitals

Un potenciòmetre digital és com un potenciòmetre rotatiu convencional, excepte que enlloc de moure un dispositiu mecànic per canviar-li la resistència, se li envien missatges a través d'un protocol de comunicació amb el valor de resistència que volem i canvia la seva resistència a partir d'aquest paràmetre.

Per entendre el seu funcionament, a la següent figura hi ha un petit diagrama que explica el funcionament d'un potenciòmetre digital de sis bits. Mitjançant el protocol de comunicació li

enviem un valor, i aquest tanca l'interruptor associat a aquest número. Per exemple, si li enviem un valor de seixanta-tres el potenciòmetre estarà situat al seu màxim valor, en canvi si li enviem un zero, estarà situat al valor mínim.

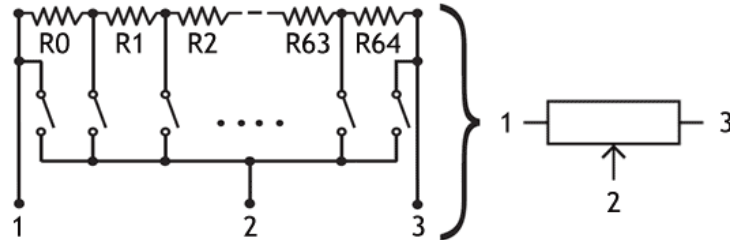


Figura 17. Estructura interna d'un potenciòmetre digital de sis bit

3.4. Possibles configuracions

En aquest apartat es discutiran les possibles configuracions per dur a terme l'amplificació desitjada. Aquestes configuracions estan elaborades amb els elements descrits en els apartats anteriors.

3.4.1. Amplificador operacional amb resistència

Aquesta és la configuració més senzilla, composta per un amplificador operacional i dues resistències com a mínim. Els avantatges d'aquesta configuració són que sempre tindrem un guany constant i res l'alterarà. Per contra, en aquesta configuració no hi ha una forma fàcil de canviar el guany, per canviar-lo s'ha de dessoldar la resistència i posar en el seu lloc la nova resistència que ens proporciona el guany desitjat, fent aquesta configuració molt limitada per la aplicació.

3.4.2. Amplificador operacional amb potenciòmetre

Aquesta configuració és similar a la anterior, però enlloc de dues resistències n'hi ha una i un potenciòmetre. Aquest potenciòmetre per motius logístics no pot ser mecànic, perquè sinó un cop es posi la placa dins del AUV no es podrà modificar el guany, per aquest motiu ha de ser un potenciòmetre digital. Els avantatges d'aquesta configuració és que disposem de guany variable d'una forma senzilla. Per contra, el disseny de la placa es complica una mica més, així com la programació d'aquesta.

3.4.3. Amplificador d'instrumentació amb potenciòmetre

Aquesta configuració és semblant a l'anterior, però aquesta simplifica el nombre de components passius de la placa, cosa que redueix les dimensions d'aquesta. Els avantatges d'aquesta configuració és que l'amplificador d'instrumentació és molt més estable que un

amplificador operacional perquè ha estat construït amb el propòsit de la reducció del soroll i limitar les distorsions d'altres senyals. Per contra aquesta configuració pateix dels mateixos punts febles que l'anterior.

3.5. Disseny final amplificació etapa entrada hidròfons

Un cop vistes totes les possibles configuracions i possibilitats, al final s'ha escollit l'amplificador d'instrumentació amb un potenciòmetre digital. Aquesta és sens dubte la configuració més estable i acurada pel control del guany. A la figura següent es pot veure l'amplificació per un sol hidròfon.

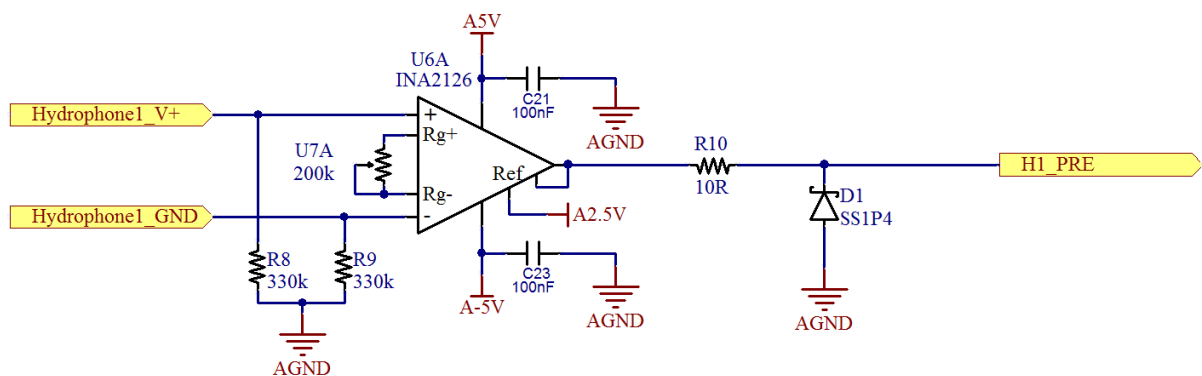


Figura 18. Disseny final analògic receptor

Destacar que a la pota Ref de l'amplificador d'instrumentació hi ha dos coma cinc volts, això és degut a que el microcontrolador no permet voltatges negatius, i s'ha optat per pujar el nivell de la senyal a la meitat del màxim que pot adquirir. El díode zener és per evitar que voltatges negatius arribin al microcontrolador. Per voltatges més grans que cinc volts no ens n'hem de preocupar, ja que l'amplificador es saturarà i com a molt tindrà una sortida de cinc volts.

3.6. Disseny amplificació emissor acústic

La generació mitjançant un hidròfon es pot aconseguir introduint un senyal sinusoïdal amb la capacitat de fer ressonar l'element piezoelèctric de l'hidròfon. Les proves realitzades amb un voltatge màxim d'un volt pic a pic i una ona sinusoïdal de dos mil hertz es podia escoltar l'ona emesa. Al tractar-se d'un element pensat per l'adquisició i no la generació, no s'ha volgut introduir més voltatge, ja que el fabricant no diu les especificacions com a emissor i no es volia destruir l'hidròfon. Per futures aplicacions que es vulgui emetre a grans distàncies, caldrà canviar aquest per un element pensat per la generació.

Pel que fa a la part de generació de l'ona, aquesta un cop generada amb el convertidor digital analògic, màxim de zero a un volt, la passem per un condensador electrolític per treure-li la part de contínua, i finalment la passarem per un seguidor de voltatge per donar-li més potència, ja que el convertidor digital analògic no te prou força per fer ressonar l'element piezoelèctric de l'hidròfon.

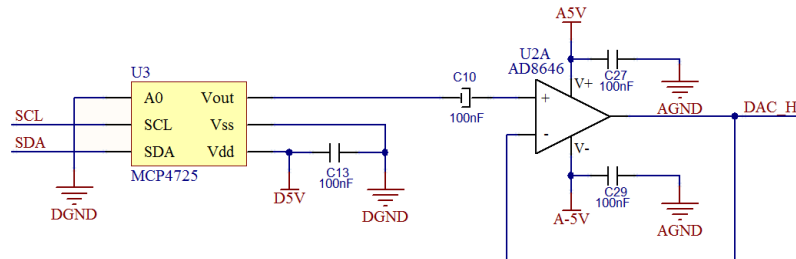


Figura 19. Disseny final analògic emissor

4. Hardware Digital

L'elecció del hardware digital juntament amb l'analògic és una de les parts més importants del projecte, ja que hi ha diferents components que fan quasi el mateix, combinacions entre aquests són favorables, mentre que d'altres no ho son. El nostre projecte requereix d'un rang de freqüència que pugui abastar des d'un hertz fins a cent mil hertz, amb un voltatge d'operació de cinc volts.

4.1. Introducció

Es tenen quatre hidròfons per capturar el senyal analògic, i aquests al ser elements passius no fan cap feina de processament. Seria molt complicat fer que el hardware analògic processés les dades provinents dels hidròfons. Per tal de poder processar les dades, primer s'han de convertir a digital, això s'aconsegueix mitjançant un convertidor analògic digital, i finalment ja es poden passar les dades al microprocessador per ser processades.

4.1.1. D'analògic a digital

Per la captura del senyal, aquest s'ha de convertir d'analògic a digital, l'element que permet aquest pas és un ADC (convertidor analògic digital). En el mercat hi ha milers de tipus de convertidors, tots tenen especificacions diferents, velocitats de mostrejat més altes o més baixes, en aquest cas s'ha optat per escollir un microcontrolador especialitzat en el processament digital de senyals (DSP), que té diverses entrades analògiques i una velocitat de mostreig més que respectable. Alhora té la habilitat de capturar quatre entrades en el mateix instant de temps.

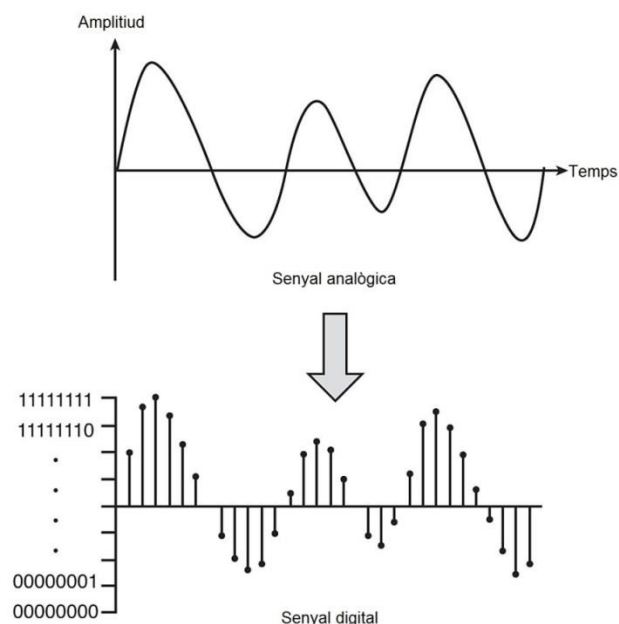


Figura 20. Exemple de conversió d'analògic a digital

Degut al mostrejat dels senyals el màxim retard entre dos hidròfons adjacents és de:

$$\delta = \frac{F_s \cdot d}{c} \quad (\text{Eq. 14})$$

On F_s és la freqüència de mostreig, d és la distància entre els dos hidròfons i c és la velocitat del so a l'aigua.

4.1.2. De digital a analògic

Per la conversió de digital a analògic hi ha diferents metodologies, tals com resistències en paral·lel connectades a les potes del microcontrolador. Però la millor de totes és mitjançant un convertidor digital a analògic (DAC). Aquest mitjançant un protocol de comunicació rep un valor, i el converteix en un voltatge constant a la seva sortida. Si es repeteix varies vegades per segon aquesta operació, podem treure per la sortida un senyal sinusoidal, que ens servirà per poder generar ones subaquàtiques.

4.1.3. Tècniques de mostrejat

El primer que es té en compte és que s'ha de fer servir el teorema de Nyquist per mostrejar la senyal. Aquest teorema assegura que es podrà reconstruir el senyal mostrejat a partir de les mostres sempre i quan la freqüència de mostreig sigui dues vegades més gran que la màxima freqüència que volem mostrejar. Però s'aconsella que a nivells pràctics aquesta sigui com a mínim unes deu vegades més gran si es vol treballar sense incerteses, ja que d'aquesta manera s'asseguren com a mínim deu punts per període.

Tot i així aquest no ens assegura una perfecte reconstrucció del senyal. Si no es mostra el senyal amb prou freqüència es pot donar el cas que es produeixi l'aliasing i no es pugui capturar el senyal correctament. Per exemple, si la freqüència més alta és de trenta cinc mil hertz, aleshores s'ha de capturar com a mínim a una freqüència de setanta mil hertz per aconseguir un resultat favorable. En la figura següent podem veure un clar exemple de com la freqüència de mostrejat influeix en la ona reconstruïda, on f és la freqüència del senyal a mostrejar. En el cas c, es pot observar el fenomen d'aliasing.

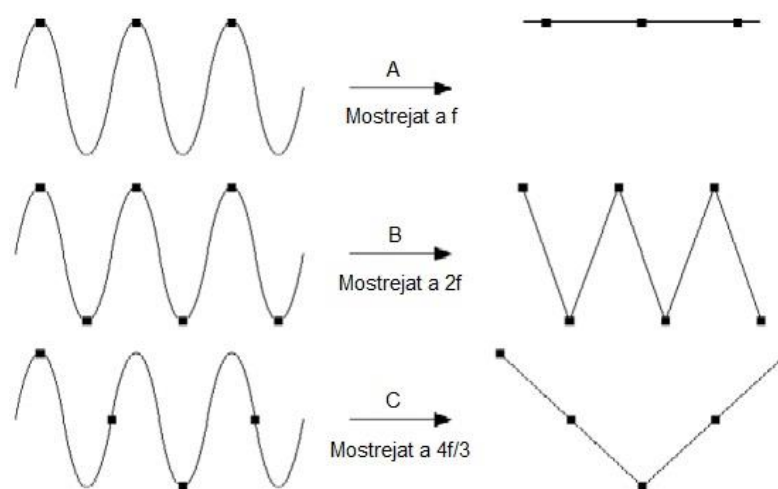


Figura 21. Mateix senyal per a diferents freqüències de mostreig i la seva reconstrucció en el pla digital

4.2. El microcontrolador

En aquest cas és el cervell de la operació, l'encarregat d'adquirir les dades analògiques i convertir-les a digital per posteriorment processar-les i obtenir un resultat. Per tal d'efectuar aquesta tasca s'ha escollit un dsPIC de la marca microchip. Per ser més concrets s'ha optat pel model dsPIC30F4011. Aquesta família és com un MCU (microcontrolador) però amb funcions de DSP i això s'anomena DSC (digital signal controllers)

4.2.1. Característiques

Les característiques principals d'aquesta família és que tenen un rang d'alimentació comprès entre dos coma cinc volts i cinc coma cinc volts en corrent continu, un rang de temperatura interna que va des de menys quaranta graus a cent vint-i-cinc graus centígrads. El rendiment d'aquest arriba als trenta MIPS quan el valor d'alimentació té un valor entre quatre coma cinc volts i cinc coma cinc volts.

Respecta a l'arquitectura de la CPU d'aquesta família, tenen un nucli RISC amb arquitectura Harvard millorada. Com a suport central de la informació té un banc de setze registres de setze bits cadascun i disposa d'un bus de dades de setze línies i un d'instruccions de vint-i-quatre. Per potenciar la velocitat de les operacions aritmètiques complexes posseeix un motor DSP que conté un multiplicador per hardware ràpid de disset per disset bits, dos acumuladors de quaranta bits i un registre de desplaçament robust.

Recurs	Rang de valor
Memòria del programa FLASH	48.000 bytes
Memòria de dades SRAM	2.048 bytes

Recurs	Rang de valor
Memòria de dades EEPROM	1.024 bytes
Temporitzadors de 16 bit	5
Comparador/PWM	4
10bit- ADC 1Msps	9 canals
UART	2
SPI	1
I ² C	1
CAN	1

Taula 2. Especificacions bàsiques del dsPIC30F4011

El dsPIC, tal com s'ha comentat anteriorment, porta un convertidor analògic digital de deu bits amb quatre sample and Hold (S&H) d'entrada. Aquest pot arribar a una velocitat d'adquisició d'un milió de mostres per segon, però aquesta velocitat és divideix entre el nombre d'entrades que adquirim. Consta de nou entrades analògiques, i permet la forma d'adquisició de forma seqüencial o simultània. Aquesta habilitat ens permetrà adquirir alhora el que passa a cada canal, això s'aconsegueix activant els quatre S&H de quatre canals i posteriorment convertir-los a digital amb el ADC. Aquesta habilitat és molt important en la aplicació, ja que necessitem adquirir tots els valors alhora i sense introduir retards entre ells.

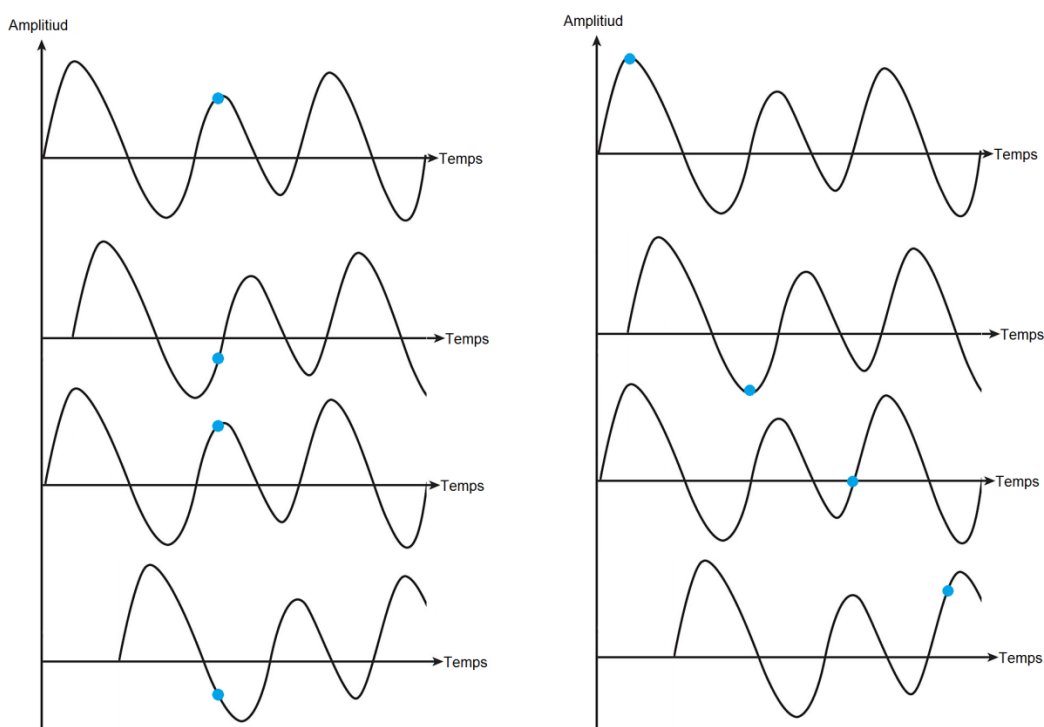


Figura 22. Exemple d'adquisicions. A l'esquerra la simultània i a la dreta la seqüencial, en blau els punts que es digitalitzaran

5. DISSENY DEL PROGRAMA

En aquest capítol s'explicarà com està programat el microcontrolador dsPIC. Podem desglossar el programa en dues parts, la primera en la que es configuren els mòduls del dsPIC, i la segona en la que es realitza tot el procés. A la següent figura podem veure el diagrama de flux general del programa.

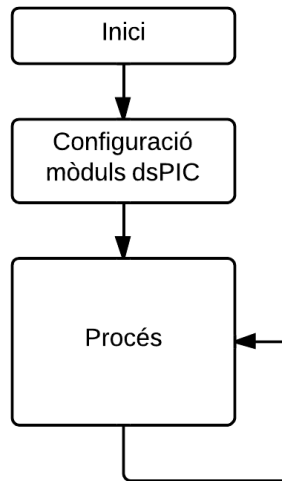


Figura 23. Diagrama de flux general del programa

5.1. Configuració dels mòduls dsPIC

En aquest apartat s'explicarà quins mòduls s'utilitzen i quin és l'ordre per configurar-los. A la figura següent es pot veure el diagrama de flux de la configuració dels mòduls.

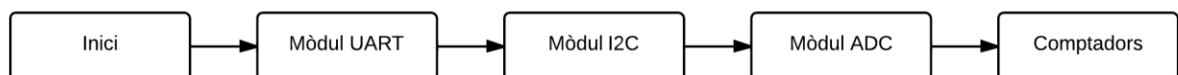


Figura 24. Diagrama de flux de la configuració dels mòduls

5.1.1. Mòdul UART

Aquest mòdul és l'encarregat de gestionar les comunicacions del dsPIC, en aquest cas es fa servir el protocol de comunicació sèrie RS232. La comunicació s'ha establert en dinou-mil dos-cents bits per segon. Per tal d'optimitzar la comunicació s'ha implementat en forma d'interruptió i mitjançant uns buffers rotatius. Cada cop que es produeix una interrupció, es llegeix el valor i es guarda dins el buffer, i posteriorment es llegeix en el programa. En cas que no es llegeixi, un cop el buffer és ple, es sobreescriu. El mateix per les interrupcions d'escriptura.

5.1.2. Mòdul I²C

Aquest mòdul és l'encarregat de comunicar-se amb el potenciòmetres digitals i el convertidor analògic digital. La velocitat d'aquest s'ha establert a quatre-cents mil bauds per segon. La podem determinar mitjançant els datasheets dels dos elements i escollint la màxima velocitat de l'element més lent, en aquest cas el potenciòmetre digital. En l'estàndard de l'I²C aquesta velocitat l'anomenen fast.

5.1.3. Mòdul ADC

Aquest mòdul és l'encarregat de convertir les dades que es troben en l'entrada analògica i convertir-les en digital. La seva implementació s'ha fet manualment a nivell de registres, ja que el compilador que feia servir no tenia la opció d'adquisició simultània. Per aquest mòdul si que s'ha activat la seva interrupció. Aquesta s'ha configurat de tal manera que el mòdul llegeix i converteix els quatre canals alhora, i un cop te les dades disponibles, activa el flag d'interrupció.

5.1.4. Comptadors

Aquests també anomenats temporitzadors són els encarregats de mesurar la diferència de temps d'arribada entre ones. S'ha configurat de tal manera que la temporització duri com a molt vuit milisegons, tres milisegons més del que dura el puls de l'emissor.

5.2. Estructura del procés

En aquest apartat és on hi ha el programa encarregat d'adquirir, trobar les diferències de temps entre els hidròfons i calcular els angles. Per simplificar el programa s'han fet dos diagrames de flux, un que és la part d'adquisició que consta de diferents passos, i l'altre és el procés general del programa.

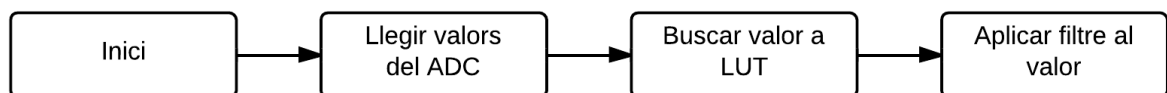


Figura 25. Diagrama de flux de la part d'adquisició de dades

En la figura anterior podem veure el mètode de captura de les dades, primer es llegeixen del convertidor analògic digital, seguidament es busca el seu valor real mitjançant una LUT (look up table) i finalment s'aplica el filtre, un cop tractada la dada s'envia per processar. L'ús de la taula és per estalviar operacions, ja que aquesta busca el valor real equivalent al valor donat pel convertidor analògic digital.

El programa es vasa en el sistema de comptatge explicat en el capítol dos, però s'ha fet de forma totalment digital. En la figura següent podem veure el diagrama de flux del programa.

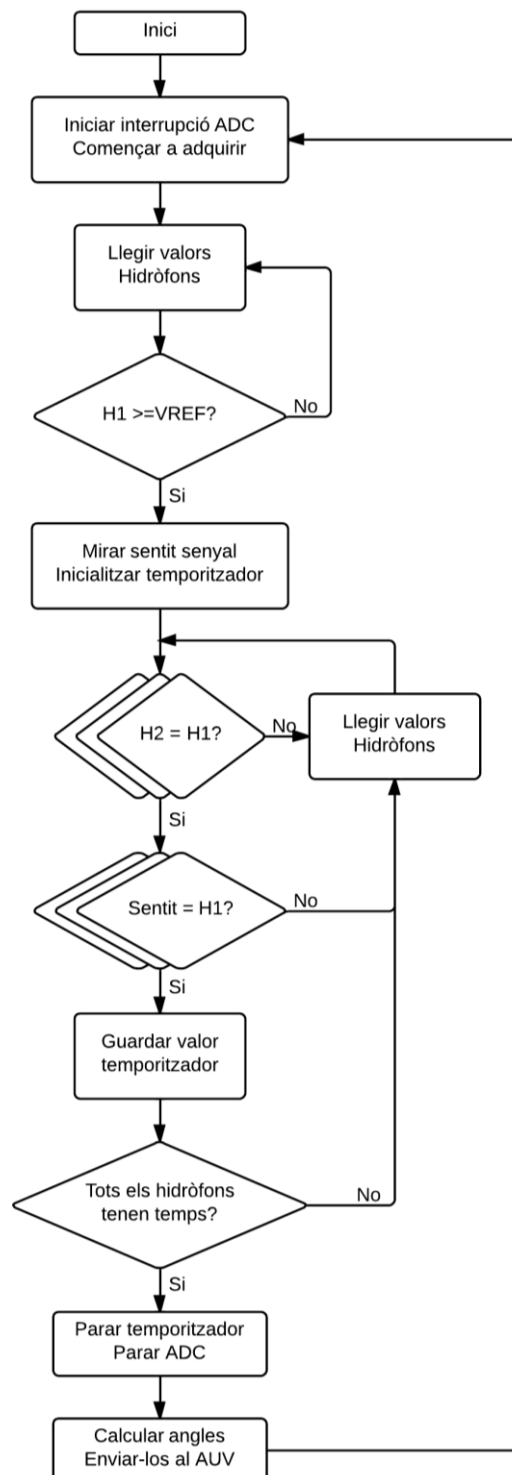


Figura 26. Diagrama de flux del programa

6. PROVES REALITZADES

En aquest capítol s'explicarà el procés evolutiu que ha tingut el projecte des del seu inici fins al final. Cada pas que s'ha fet per aconseguir el final, ha passat per una sèrie de proves i dificultats que s'explicaran a continuació.

Primer de tot, es va tenir una presa de contacte amb els elements hidròfons, durant un dia es va veure com funcionaven i com reaccionaven al so. D'aquest primer contacte es va veure clarament amb l'oscil·loscopi que el senyal s'havia d'amplificar unes cent vegades com a mínim per poder-hi treballar còmodament.

6.1. LabVIEW

Les primeres proves es van realitzar amb el programari LabVIEW, ja que és un programa que s'ha utilitzat a algunes assignatures. Aquesta primera prova es va efectuar amb un ordinador.

6.1.1. Hardware

Per tal de poder emetre dins de l'aigua, el laboratori em va subministrar un altaveu aquàtic i un amplificador d'àudio. Mitjançant un reproductor mp3 amb el so de l'ona es podia emetre aquest dins l'aigua.

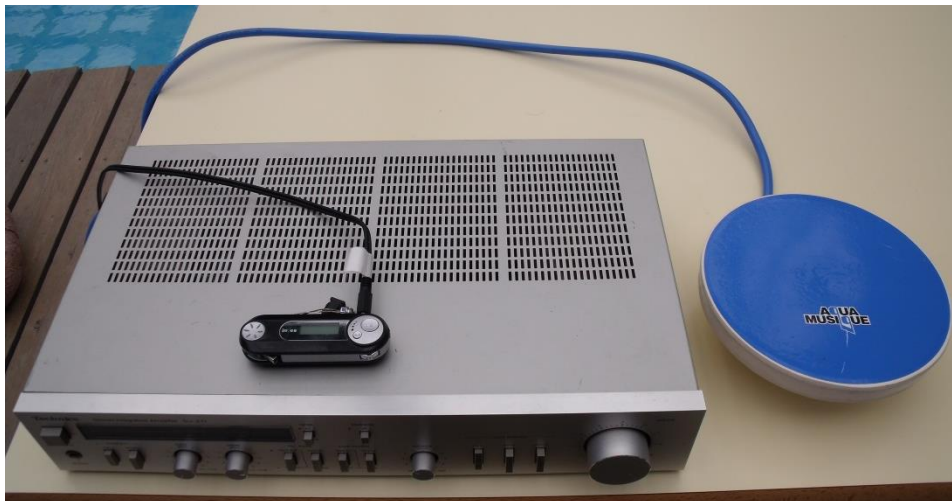


Figura 27. Amplificador i altaveu aquàtic

La segona part del hardware és la plataforma dels hidròfons, aquesta es va dissenyar en un panell plàstic i amb l'ajuda d'un transportador d'angles es van marcar tots els angles amb increments de deu graus. Al mig d'aquesta plataforma s'hi va posar el suport dels hidròfons, aquest ens permet girar sense moure la plataforma i alhora subjectar correctament. La

distància entre els dos hidròfons és de set coma dos centímetres. Aquesta separació es va determinar després de fer un seguit de proves, ja que si estaven massa junts, s'interferien entre ells i no rebien correctament. Aquesta separació ens permet captar correctament freqüències que van des d'un hertz (rang mínim que pot captar l'hydròfon), fins a setze mil quatre-cents vuitanta nou hertz (calculat a partir de l'equació dos).

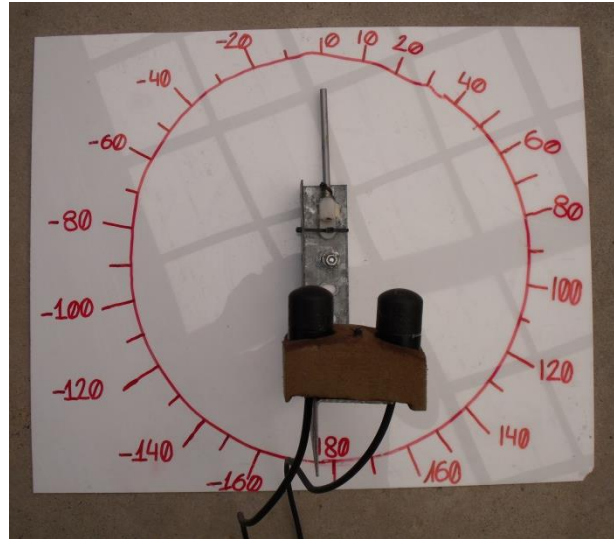


Figura 28. Plataforma per la subjecció dels hidròfons i la mesura dels angles

Per aquesta prova es va crear una petita placa amplificadora capaç de captar el senyal de dos hidròfons, amplificar-los i treure'ls amb una sortida d'àudio estèreo, que va connectada a l'entrada d'àudio de l'ordinador.

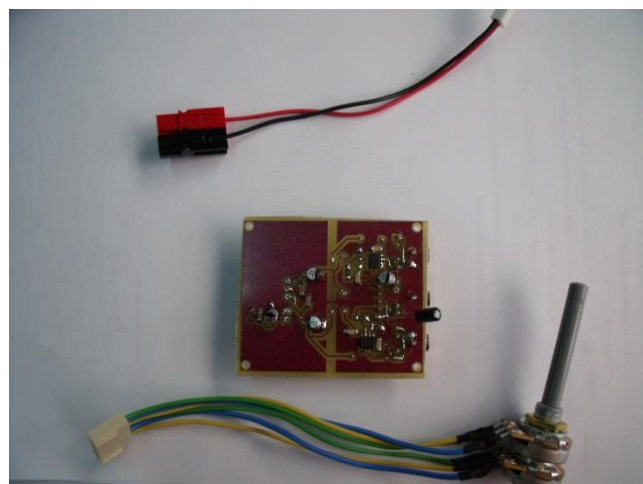


Figura 29. Placa pre-amplificadora, amb dues entrades mono i una sortida estèreo

6.1.2. Programa

Amb el LabVIEW es va crear un programa que permetia obtenir l'angle de procedència de l'emissor acústic. El que es feia en aquest era agafar els senyals del canal estèreo i separar-los amb el canal dret i l'esquerra, s'hi aplicava un filtre passa banda i es calculava l'angle de l'emissor.

6.1.3. Problemes

Al realitzar les proves en un espai reduït, la piscina del laboratori, les ones acústiques rebotaven per les parets d'aquesta, dificultant el programa per determinar l'angle. Alhora d'adquirir les dades es feia mitjançant la targeta de so de l'ordinador, aquesta mostrejava com a molt a noranta-sis mil hertz, i introduïda soroll de l'alterna i un offset en els dos canals, apart que hi havia acoblament de senyals entre un canal i l'altre.

6.1.4. Conclusions

Les mesures dels angles que es van aconseguir van ser força aproximades, amb errors elevats, això s'atribueix a la qualitat de la conversió d'analògic a digital. Un cop comprovat que el sistema era viable de construir, es va dissenyar una aplicació amb un microcontrolador i una part amplificadora, tot això a la mateixa placa. D'aquesta manera s'espera resoldre el problema de l'adquisició.

En la taula següent podem veure una comparació dels angles reals amb els adquirits amb el LabVIEW, es van captar els tres-cents seixanta graus, però en la taula només es mostren des de menys cinquanta fins a cinquanta graus, ja que els valors restants són amb molt més error.

Angle real	Angle calculat
-50	-68
-40	-50
-30	-42
-20	-30
-10	-18
0	1
10	15
20	22
30	39
40	62
50	75

Taula 3. Comparació d'angles reals amb els angles calculats mitjançant el LabVIEW

6.2. Microcontrolador

A partir de les experiències anteriors, el que es va proposar va ser encapsular tot lo anterior, és a dir, un sistema que permetés adquirir i generar les ones subaquàtiques amb una sola placa, i alhora enviés la informació dels angles a l'ordinador de l'AUV.

6.2.1. Hardware

El hardware és el mateix que en l'apartat anterior, amb excepció de la placa. Aquesta cobreix la part d'amplificació, generació i processament del senyal. S'han fet dues versions d'aquesta placa. En la primera versió, la part d'amplificació s'efectua amb tres amplificadors operacionals per hidròfon, dos per amplificar el senyal i un tercer per sumar-hi un offset. La següent figura mostra la primera versió.

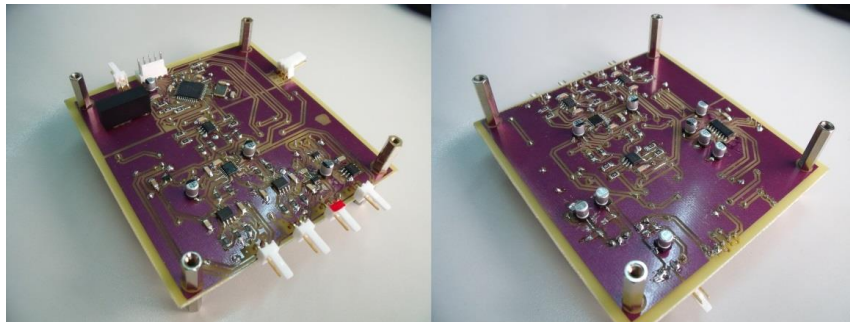


Figura 30. Placa d'adquisició, generació i processament del senyal provinent de l'emissor acústic

Un dels majors problemes que es van tenir amb aquesta versió van ser les interferències i sorolls. Això és degut a l'amplificació amb tres etapes en mode comú. Això dificultava l'adquisició ja que es tenien valors amb pics molt pronunciats i dificultaven la detecció de l'angle. Per solucionar aquest problema el que es va fer va ser crear una nova placa simplificant la part d'amplificació. Es van substituir els tres amplificadors operacionals per un únic amplificador d'instrumentació capaç d'amplificar i sumar un offset a la sortida. A la imatge següent podem veure la placa final per les dues cares. Simplifica el disseny anterior afavorint la incorporació d'aquesta dins del AUV.

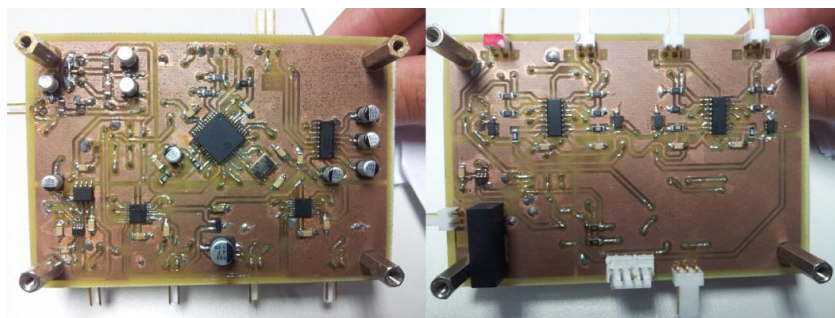


Figura 31. Placa Final

6.2.2. Problemes

Es va començar a programar el microcontrolador amb el mètode de la correlació creuada, ja que amb diferents proves efectuades amb el matlab s'havia vist que és el més eficaç i menys sensible a sorolls i pertorbacions. Un cop entès el principi, es va crear un algorisme i finalment es va implementar amb el microcontrolador. En el moment de l'elecció del microcontrolador no es va tenir en compte que aquest mètode consumiria tants recursos impossibilitant la seva implementació, així que finalment es va implementar el mètode del comptador, tal com s'explica en el capítol número dos, però per software.

Un dels majors problemes que s'han tingut és que no es poden veure les dades que processem amb el microcontrolador, això és degut a la interfície de comunicació RS232. Adquirim dades amb el dsPIC a una freqüència d'un Msps i la màxima velocitat del port sèrie admesa pel compilador és de trenta vuit mil quatre-cents bauds. Això fa que no es pugui veure què passa, dificultant la verificació i comprovació del programa.

La programació d'aquest ha estat complicada degut a problemes del compilador, ja que no implementava les funcions desitjades o estaven mal implementades, finalment s'han implementat a nivell de registre. Una de les més remarcables ha estat la configuració del convertidor analògic digital perquè adquirís de forma simultània i a la velocitat desitjada.

6.2.3. Conclusions

Al haver implementat una nova forma d'amplificació s'ha reduït considerablement el soroll, tot i així encara es pot apreciar, però gràcies al filtre digital s'ha pogut eliminar. En el moment de l'entrega de la memòria el sistema no s'ha pogut acabar d'implementar. En l'actualitat es disposa d'una versió del codi que estima l'angle verdader amb alguns errors que cal depurar. S'espera que durant el transcurs de l'estiu es pugui depurar el codi fent que totes les dades obtingudes siguin correctes.

7. POSSIBLES APLICACIONS

Un cop realitzat el sistema, aquest obre les portes a possibles aplicacions que es poden crear gràcies a aquest, un clar exemple podria ser la localització de caixes negres. Aquestes estan dissenyades amb un pinger que quan toca l'aigua comença a emetre senyals acústics a una determinada freqüència. Sabent aquesta freqüència el sistema pot trobar la localització de la caixa negra, facilitant-ne la recuperació.

Una altra aplicació seria un transponder, aquest constaria de dos plaques, una en una posició coneguda, i l'altre en una desconeguda. El seu funcionament és emetre una ona acústica des de la localització coneguda. El que està submergit la rep i respon amb la mateixa ona. L'emissor al rebre l'ona i sap l'angle d'on ha vingut i a partir del temps de vol sap la distància, ja que sabent la velocitat del so sota l'aigua i el temps que ha trigat a arribar des de que s'ha transmès es pot determinar la posició de l'element. Aquest sistema a efectes pràctics pot servir per la localització de l'AUV a les missions, monitorització de submarinistes des de la barca, etc.

8. RESUM DEL PRESSUPOST

La creació de la placa per la detecció i generació d'ones acústiques subaquàtiques puja a la quantitat mil cinc-cents cinquanta tres amb vint-i-nou euros, sense IVA.

9. CONCLUSIONS

En aquest projecte primer es va començar mirant la viabilitat d'aquest mitjançant dos hidròfons i calculant l'angle mitjançant LabVIEW. Un cop es va veure que era viable, es va dissenyar una placa electrònica que efectua la part d'amplificació i càlcul de l'angle. D'aquesta placa es va dissenyar una altra versió per eliminar els problemes trobats en la primera.

El sistema és capaç d'adquirir les ones acústiques provinents dels hidròfons i amplificar-les correctament. Així com l'estimació de l'angle de l'emissor amb uns errors que s'esperen depurar durant el transcurs de l'estiu. Pel que fa a la generació d'ones acústiques no s'ha pogut dur a terme per falta de temps, era la part més senzilla i s'ha volgut dedicar més temps a la part del càlcul de l'angle. Per tant l'objectiu del projecte ha estat parcialment cobert, tot i que l'abast ha anat més enllà del previst degut a la complexitat del disseny i control de la placa microcontrolada.

Com a treball futur, es proposa canviar de microcontrolador per un de més potent i versàtil que permeti el mètode de la correlació creuada i veure les dades en temps real. Aquest podria ser un raspberry Pi, aquest funciona amb Linux i s'hi pot instal·lar ROS (Robot Operating System), que és el sistema en que actualment funcionen els robots del laboratori. Es podria provar el sistema en diferents entorns, per veure com reacciona, i si la lectura de l'angle és correcta. També es podria provar el sistema amb quatre hidròfons i diferents configuracions per veure com afecten aquestes a l'adquisició de les dades, i consegüentment al càlcul de l'angle.

Jordi Gratacós Prunell

Graduat en Enginyeria Electrònica Industrial i Automàtica

Banyoles, 21 de juliol de 2013

10. RELACIÓ DE DOCUMENTS

El projecte consta de cinc documents: memòria, plànols, plec de condicions, estat d'amidaments i pressupost.

11. BIBLIOGRAFIA

ANALOG DEVICES, 24MHz Rail-to-Rail Amplifiers with Shutdown Option. Norwood 2009. (CD ROM)

ANALOG DEVICES, Single/Dual, +15V/±5V, 256-Position, I²C-Compatible Digital Potentiometer. Norwood 2009. (CD ROM)

CARLES POUS, Processament digital del senyal I i II. UdG. Girona 2013. (CD ROM)

JOHN PIKE, Underwater Acoustics (<http://www.fas.org/man/dod-101/sys/ship/acoustics.htm>, 23 d'abril de 2013)

MAXIM, Low-Voltage, Dual-Supply, SPDT Analog Switch with Enable. Sunnyvale 2003. (CD ROM)

MICROCHIP, 12-Bit Digital-to-Analog Converter with EEPROM Memory in SOT-23-6. West Chandler Blvd. 2007. (CD ROM)

MICROCHIP, dsPIC30F4011/4012 Data Sheet. West Chandler Blvd., 2010. (CD ROM)

MICROCHIP, Reference manual dsPIC30f family 10bit A/D Converter. West Chandler Blvd. 2007 (CD ROM)

MICROCHIP, Reference manual dsPIC30f family I2C. West Chandler Blvd. 2009 (CD ROM)

MICROCHIP, Reference manual dsPIC30f family Timers. West Chandler Blvd. 2005 (CD ROM)

MICROCHIP, Reference manual dsPIC30f family UART. West Chandler Blvd. 2004 (CD ROM)

MOHAMAD FAIZAL, Underwater Acoustic Transducer: Receive Sensitivity and Transmit Response (http://urrg.eng.usm.my/index.php?option=com_content&view=article&id=115:underwater-acoustic-transducer-receive-sensitivity-and-transmit-response&catid=31:articles&Itemid=70, 20 d'abril de 2013)

MOHD IKHWAN, hydrophone: designs and principles (http://urrg.eng.usm.my/index.php?option=com_content&view=article&id=69:hydrophone-designs-and-principle&catid=31:articles&Itemid=70, 5 d'abril de 2013)

RAMÓN PALLÁS ARENY, Sensors and signal Conditioning. New York. 2001

ROBERT J. URICK, Principles of underwater sound. Peninsula Publishing. Los Altos. 1983.

SITI SARAH SAMSURI, Underwater Acoustic: Speed of Sound, c :[m/s] in Sea Water (http://urrg.eng.usm.my/index.php?option=com_content&view=article&id=256:underwater-acoustic-speed-of-soundc--ms-in-sea-water-&catid=31:articles&Itemid=70, 5 d'abril de 2013)

SOUND IN THE SEA, What are common underwater sounds? (<http://www.dosits.org/science/soundsinthesea/commonsounds/>, 5 d'abril de 2013)

TEXAS INSTRUMENTS, CMOS voltatge reference Dallas 2003. (CD ROM)

TEXAS INSTRUMENTS, INA126/INA2126. Dallas 2005. (CD ROM)

VISHAY, High Current Density Surface Mount Schottky Barrier Rectifiers, D-95100 Germany 2012. (CD ROM)

12. GLOSSARI

ADC: Analog Digital Converter

AUV: Autonomous Underwater Vehicle

CAN: Controller Area Network

CPU: Central Processing Unit

DAC: Digital Analog Converter

DSC: Digital Signal Controller

DSP: Digital Signal Processing

EEPROM: Electrically Erasable Programmable Read-Only Memory

FFT: Fast Fourier Transform

I²C: Inter-Integrated Circuit

Li-ion: Lithium-ion

LUT: Look up table

MCU: Microcontroller

MIPS Millions Instruction Per Second

Msp/s: Mega samples per second

PCI-e: Peripheral Component Interconnect Express

PWM: Pulse-Width Modulation

RISC: Reduced Instruction Set Computer

SMD: Surface-Mount Device

SOT: Small Outline Transistor

SPI: Serial Peripheral Interface bus

SRAM: Static Random Access Memory

SSOP: Shrink Small-Outline Package

TQFP: Thin Quad Flat Package

UART: Universal Asynchronous Receiver/Transmitter

USB: Universal Serial Bus

A. CODI DEL PROGRAMA

```
#include <SoundLocator.h>
#include <inputRaw.h>
#include <outputRaw.h>
#include <math.h>

/*****
*
* CONFIGURACIONS
*
*****/
//CONFIGURACIÓ DEL MODUL UART I DELS SEUS REGISTRES
#use rs232(UART1A,baud=19200,parity=N,bits=8)
#word U1TXREG = 0x210 //UART1 Transmit Register
#word U1RXREG = 0x212 //UART1 Receive Register
#word U1STA = getenv("SFR:U1STA") //UART1 Status and Control Register

//DEFINICIÓ DEL REGISTRES DEL TIMER 2
#word T2CON = 0x0110
#bit TCS = T2CON.1
#bit T32 = T2CON.3
#bit TCKPS0 = T2CON.4
#bit TCKPS1 = T2CON.5
#bit TON = T2CON.15
#word TMR2 = 0x0106

//DECLARACIÓ DELS REGISTRES DEL MODUL ADC
#word ADCON1 = getenv("SFR:ADCON1") //A/D Control Register 1
#bit ADON = ADCON1.15
#word ADCON2 = getenv("SFR:ADCON2") //A/D Control Register 2
#word ADCON3 = getenv("SFR:ADCON3") //A/D Control Register 3
#word ADCHS = getenv("SFR:ADCHS") //A/D Input Channel Select Register
#word ADPCFG = getenv("SFR:ADPCFG") //A/D Port Configuration Register
#word ADCSSL = getenv("SFR:ADCSSL") //A/D Input Scan Select Register
#word ADCBUF0 = getenv("SFR:ADCBUF0") //ADC Data Buffer 0
#word ADCBUF1 = getenv("SFR:ADCBUF1") //ADC Data Buffer 1
#word ADCBUF2 = getenv("SFR:ADCBUF2") //ADC Data Buffer 2
#word ADCBUF3 = getenv("SFR:ADCBUF3") //ADC Data Buffer 3

//CONFIGURACIÓ DEL MODUL I2C
#use i2c(Master,Fast,sda=PIN_F2,scl=PIN_F3)

//DEFINICIÓ DE FUNCIONS
void configuracio();
float32 filterH1(float32 in);
float32 filterH2(float32 in);
unsigned int16 ad5282_read_val(unsigned int8 adr);
void ad5282_write_val(unsigned int8 adr, int1 pot, unsigned int8 data);
float32 tempsDelay(unsigned int16 temp);
float32 angleEmissor(float32 temp, float32 soundVel, float32 dist);

//DEFINICIÓ DE CONSTANTS DEL PROGRAMA
#define POT_STEP 784.31 //ve donat per 200000/255 resistència/nºpassos
//const float32 VREF = 0.065;
float32 VREF = 0.5;
const float32 SOUNDVEL = 1435.00; //1550.00; quan es vagi al mar canviar-ho
const float32 DISTH = 0.047;
const float32 Tcy = 0.000000125;
const unsigned int8 AD5282_1_ADDR = 0b00101101; //POT H3-H4
const unsigned int8 AD5282_2_ADDR = 0b00101110; //POT H1-H2
const float32 ADC_LUT[1024] =
{
    -2.50000,-2.49511,-2.49022,-2.48534,-2.48045,-2.47556,-2.47067,-2.46579,
    -2.46090,-2.45601,-2.45112,-2.44624,-2.44135,-2.43646,-2.43157,-2.42669,
    -2.42180,-2.41691,-2.41202,-2.40714,-2.40225,-2.39736,-2.39247,-2.38759,
    -2.38270,-2.37781,-2.37292,-2.36804,-2.36315,-2.35826,-2.35337,-2.34848,
    -2.34360,-2.33871,-2.33382,-2.32893,-2.32405,-2.31916,-2.31427,-2.30938,
    -2.30450,-2.29961,-2.29472,-2.28983,-2.28495,-2.28006,-2.27517,-2.27028,
    -2.26540,-2.26051,-2.25562,-2.25073,-2.24585,-2.24096,-2.23607,-2.23118,
    -2.22630,-2.22141,-2.21652,-2.21163,-2.20674,-2.20186,-2.19697,-2.19208,

```

-2.18719,-2.18231,-2.17742,-2.17253,-2.16764,-2.16276,-2.15787,-2.15298,
-2.14809,-2.14321,-2.13832,-2.13343,-2.12854,-2.12366,-2.11877,-2.11388,
-2.10899,-2.10411,-2.09922,-2.09433,-2.08944,-2.08456,-2.07967,-2.07478,
-2.06989,-2.06500,-2.06012,-2.05523,-2.05034,-2.04545,-2.04057,-2.03568,
-2.03079,-2.02590,-2.02102,-2.01613,-2.01124,-2.00635,-2.00147,-1.99658,
-1.99169,-1.98680,-1.98192,-1.97703,-1.97214,-1.96725,-1.96237,-1.95748,
-1.95259,-1.94770,-1.94282,-1.93793,-1.93304,-1.92815,-1.92326,-1.91838,
-1.91349,-1.90860,-1.90371,-1.89883,-1.89394,-1.88905,-1.88416,-1.87928,
-1.87439,-1.86950,-1.86461,-1.85973,-1.85484,-1.84995,-1.84506,-1.84018,
-1.83529,-1.83040,-1.82551,-1.82063,-1.81574,-1.81085,-1.80596,-1.80108,
-1.79619,-1.79130,-1.78641,-1.78152,-1.77664,-1.77175,-1.76686,-1.76197,
-1.75709,-1.75220,-1.74731,-1.74242,-1.73754,-1.73265,-1.72776,-1.72287,
-1.71799,-1.71310,-1.70821,-1.70332,-1.69844,-1.69355,-1.68866,-1.68377,
-1.67889,-1.67400,-1.66911,-1.66422,-1.65934,-1.65445,-1.64956,-1.64467,
-1.63978,-1.63490,-1.63001,-1.62512,-1.62023,-1.61535,-1.61046,-1.60557,
-1.60068,-1.59580,-1.59091,-1.58602,-1.58113,-1.57625,-1.57136,-1.56647,
-1.56158,-1.55670,-1.55181,-1.54692,-1.54203,-1.53715,-1.53226,-1.52737,
-1.52248,-1.51760,-1.51271,-1.50782,-1.50293,-1.49804,-1.49316,-1.48827,
-1.48338,-1.47849,-1.47361,-1.46872,-1.46383,-1.45894,-1.45406,-1.44917,
-1.44428,-1.43939,-1.43451,-1.42962,-1.42473,-1.41984,-1.41496,-1.41007,
-1.40518,-1.40029,-1.39541,-1.39052,-1.38563,-1.38074,-1.37586,-1.37097,
-1.36608,-1.36119,-1.35630,-1.35142,-1.34653,-1.34164,-1.33675,-1.33187,
-1.32698,-1.32209,-1.31720,-1.31232,-1.30743,-1.30254,-1.29765,-1.29277,
-1.28788,-1.28299,-1.27810,-1.27322,-1.26833,-1.26344,-1.25855,-1.25367,
-1.24878,-1.24389,-1.23900,-1.23412,-1.22923,-1.22434,-1.21945,-1.21457,
-1.20968,-1.20479,-1.19990,-1.19501,-1.19013,-1.18524,-1.18035,-1.17546,
-1.17058,-1.16569,-1.16080,-1.15591,-1.15103,-1.14614,-1.14125,-1.13636,
-1.13148,-1.12659,-1.12170,-1.11681,-1.11193,-1.10704,-1.10215,-1.09726,
-1.09238,-1.08749,-1.08260,-1.07771,-1.07283,-1.06794,-1.06305,-1.05816,
-1.05327,-1.04839,-1.04350,-1.03861,-1.03372,-1.02884,-1.02395,-1.01906,
-1.01417,-1.00929,-1.00440,-0.99951,-0.99462,-0.98974,-0.98485,-0.97996,
-0.97507,-0.97019,-0.96530,-0.96041,-0.95552,-0.95064,-0.94575,-0.94086,
-0.93597,-0.93109,-0.92620,-0.92131,-0.91642,-0.91153,-0.90665,-0.90176,
-0.89687,-0.89198,-0.88710,-0.88221,-0.87732,-0.87243,-0.86755,-0.86266,
-0.85777,-0.85288,-0.84800,-0.84311,-0.83822,-0.83333,-0.82845,-0.82356,
-0.81867,-0.81378,-0.80890,-0.80401,-0.79912,-0.79423,-0.78935,-0.78446,
-0.77957,-0.77468,-0.76979,-0.76491,-0.76002,-0.75513,-0.75024,-0.74536,
-0.74047,-0.73558,-0.73069,-0.72581,-0.72092,-0.71603,-0.71114,-0.70626,
-0.70137,-0.69648,-0.69159,-0.68671,-0.68182,-0.67693,-0.67204,-0.66716,
-0.66227,-0.65738,-0.65249,-0.64761,-0.64272,-0.63783,-0.63294,-0.62805,
-0.62317,-0.61828,-0.61339,-0.60850,-0.60362,-0.59873,-0.59384,-0.58895,
-0.58407,-0.57918,-0.57429,-0.56940,-0.56452,-0.55963,-0.55474,-0.54985,
-0.54497,-0.54008,-0.53519,-0.53030,-0.52542,-0.52053,-0.51564,-0.51075,
-0.50587,-0.50098,-0.49609,-0.49120,-0.48631,-0.48143,-0.47654,-0.47165,
-0.46676,-0.46188,-0.45699,-0.45210,-0.44721,-0.44233,-0.43744,-0.43255,
-0.42766,-0.42278,-0.41789,-0.41300,-0.40811,-0.40323,-0.39834,-0.39345,
-0.38856,-0.38368,-0.37879,-0.37390,-0.36901,-0.36413,-0.35924,-0.35435,
-0.34946,-0.34457,-0.33969,-0.33480,-0.32991,-0.32502,-0.32014,-0.31525,
-0.31036,-0.30547,-0.30059,-0.29570,-0.29081,-0.28592,-0.28104,-0.27615,
-0.27126,-0.26637,-0.26149,-0.25660,-0.25171,-0.24682,-0.24194,-0.23705,
-0.23216,-0.22727,-0.22239,-0.21750,-0.21261,-0.20772,-0.20283,-0.19795,
-0.19306,-0.18817,-0.18328,-0.17840,-0.17351,-0.16862,-0.16373,-0.15885,
-0.15396,-0.14907,-0.14418,-0.13930,-0.13441,-0.12952,-0.12463,-0.11975,
-0.11486,-0.10997,-0.10508,-0.10020,-0.09531,-0.09042,-0.08553,-0.08065,
-0.07576,-0.07087,-0.06598,-0.06109,-0.05621,-0.05132,-0.04643,-0.04154,
-0.03666,-0.03177,-0.02688,-0.02199,-0.01711,-0.01222,-0.00733,-0.00244,
0.00244,0.00733,0.01222,0.01711,0.02199,0.02688,0.03177,0.03666,0.04154,
0.04643,0.05132,0.05621,0.06109,0.06598,0.07087,0.07576,0.08065,0.08553,
0.09042,0.09531,0.10020,0.10508,0.10997,0.11486,0.11975,0.12463,0.12952,
0.13441,0.13930,0.14418,0.14907,0.15396,0.15885,0.16373,0.16862,0.17351,
0.17840,0.18328,0.18817,0.19306,0.19795,0.20283,0.20772,0.21261,0.21750,
0.22239,0.22727,0.23216,0.23705,0.24194,0.24682,0.25171,0.25660,0.26149,
0.26637,0.27126,0.27615,0.28104,0.28592,0.29081,0.29570,0.30059,0.30547,
0.31036,0.31525,0.32014,0.32502,0.32991,0.33480,0.33969,0.34457,0.34946,
0.35435,0.35924,0.36413,0.36901,0.37390,0.37879,0.38368,0.38856,0.39345,
0.39834,0.40323,0.40811,0.41300,0.41789,0.42278,0.42766,0.43255,0.43744,
0.44233,0.44721,0.45210,0.45699,0.46188,0.46676,0.47165,0.47654,0.48143,
0.48631,0.49120,0.49609,0.50098,0.50587,0.51075,0.51564,0.52053,0.52542,

0.53030,0.53519,0.54008,0.54497,0.54985,0.55474,0.55963,0.56452,0.56940,
0.57429,0.57918,0.58407,0.58895,0.59384,0.59873,0.60362,0.60850,0.61339,
0.61828,0.62317,0.62805,0.63294,0.63783,0.64272,0.64761,0.65249,0.65738,
0.66227,0.66716,0.67204,0.67693,0.68182,0.68671,0.69159,0.69648,0.70137,
0.70626,0.71114,0.71603,0.72092,0.72581,0.73069,0.73558,0.74047,0.74536,
0.75024,0.75513,0.76002,0.76491,0.76979,0.77468,0.77957,0.78446,0.78935,
0.79423,0.79912,0.80401,0.80890,0.81378,0.81867,0.82356,0.82845,0.83333,
0.83822,0.84311,0.84800,0.85288,0.85777,0.86266,0.86755,0.87243,0.87732,
0.88221,0.88710,0.89198,0.89687,0.90176,0.90665,0.91153,0.91642,0.92131,
0.92620,0.93109,0.93597,0.94086,0.94575,0.95064,0.95552,0.96041,0.96530,
0.97019,0.97507,0.97996,0.98485,0.98974,0.99462,0.99951,1.00440,1.00929,
1.01417,1.01906,1.02395,1.02884,1.03372,1.03861,1.04350,1.04839,1.05327,
1.05816,1.06305,1.06794,1.07283,1.07771,1.08260,1.08749,1.09238,1.09726,
1.10215,1.10704,1.11193,1.11681,1.12170,1.12659,1.13148,1.13636,1.14125,
1.14614,1.15103,1.15591,1.16080,1.16569,1.17058,1.17546,1.18035,1.18524,
1.19013,1.19501,1.19990,1.20479,1.20968,1.21457,1.21945,1.22434,1.22923,
1.23412,1.23900,1.24389,1.24878,1.25367,1.25855,1.26344,1.26833,1.27322,
1.27810,1.28299,1.28788,1.29277,1.29765,1.30254,1.30743,1.31232,1.31720,
1.32209,1.32698,1.33187,1.33675,1.34164,1.34653,1.35142,1.35630,1.36119,
1.36608,1.37097,1.37586,1.38074,1.38563,1.39052,1.39541,1.40029,1.40518,
1.41007,1.41496,1.41984,1.42473,1.42962,1.43451,1.43939,1.44428,1.44917,
1.45406,1.45894,1.46383,1.46872,1.47361,1.47849,1.48338,1.48827,1.49316,
1.49804,1.50293,1.50782,1.51271,1.51760,1.52248,1.52737,1.53226,1.53715,
1.54203,1.54692,1.55181,1.55670,1.56158,1.56647,1.57136,1.57625,1.58113,
1.58602,1.59091,1.59580,1.60068,1.60557,1.61046,1.61535,1.62023,1.62512,
1.63001,1.63490,1.63978,1.64467,1.64956,1.65445,1.65934,1.66422,1.66911,
1.67400,1.67889,1.68377,1.68866,1.69355,1.69844,1.70332,1.70821,1.71310,
1.71799,1.72287,1.72776,1.73265,1.73754,1.74242,1.74731,1.75220,1.75709,
1.76197,1.76686,1.77175,1.77664,1.78152,1.78641,1.79130,1.79619,1.80108,
1.80596,1.81085,1.81574,1.82063,1.82551,1.83040,1.83529,1.84018,1.84506,
1.84995,1.85484,1.85973,1.86461,1.86950,1.87439,1.87928,1.88416,1.88905,
1.89394,1.89883,1.90371,1.90860,1.91349,1.91838,1.92326,1.92815,1.93304,
1.93793,1.94282,1.94770,1.95259,1.95748,1.96237,1.96725,1.97214,1.97703,
1.98192,1.98680,1.99169,1.99658,2.00147,2.00635,2.01124,2.01613,2.02102,
2.02590,2.03079,2.03568,2.04057,2.04545,2.05034,2.05523,2.06012,2.06500,
2.06989,2.07478,2.07967,2.08456,2.08944,2.09433,2.09922,2.10411,2.10899,
2.11388,2.11877,2.12366,2.12854,2.13343,2.13832,2.14321,2.14809,2.15298,
2.15787,2.16276,2.16764,2.17253,2.17742,2.18231,2.18719,2.19208,2.19697,
2.20186,2.20674,2.21163,2.21652,2.22141,2.22630,2.23118,2.23607,2.24096,
2.24585,2.25073,2.25562,2.26051,2.26540,2.27028,2.27517,2.28006,2.28495,
2.28983,2.29472,2.29961,2.30450,2.30938,2.31427,2.31916,2.32405,2.32893,
2.33382,2.33871,2.34360,2.34848,2.35337,2.35826,2.36315,2.36804,2.37292,
2.37781,2.38270,2.38759,2.39247,2.39736,2.40225,2.40714,2.41202,2.41691,
2.42180,2.42669,2.43157,2.43646,2.44135,2.44624,2.45112,2.45601,2.46090,
2.46579,2.47067,2.47556,2.48045,2.48534,2.49022,2.49511,2.50000};

//DEFINICIÓ DE VARIABLES GLOBALS

//int8 parameter[16]; //Per ajudar a enviar dades pel port RS232

int8 estat = 0;

unsigned int8 VAL_POT = 128;

float32 H1_ant = 0;

float32 H1_Vmax = 0;

float32 H1_Vmin = 0;

float32 H2_ant = 0;

float32 H2_Vmax = 0;

float32 H2_Vmin = 0;

unsigned int16 H2_temp = 0;

unsigned int8 ERROR = 0.1;

int1 pujaBaixaH1;

int1 pujaBaixaH2;

float32 yt1,yt2,yt3,yt4,xt2,xt3,xt4;

float32 yt1h2,yt2h2,yt3h2,yt4h2,xt2h2,xt3h2,xt4h2;

/*****

* INTERRUPCIONS *

*****/

//INTERRUPTIÓ QUE S'ACTIVA QUAN ES DETECTEN DADES EN EL PORT RS232 (RX)

#INT_RDA

```

void RDA_isr(void)
{
    putInputRaw(U1RXREG);    //Guardem la dada rebuda del port RS232
}

//INTERRUPCió QUE S'ACTIVA QUAN EL BUFER DE TRANSMISIó DEL RS232 ESTÀ LLIURE
#define INT_TBE
void TBE_isr(void)
{
    if(outputRawEmpty())    //Si tenim el buffer de transmissió buit...
    {
        disable_interrupts(INT_TBE);    //desactivem la interrupció de transmissió
    }
    else
        //Si te dades...
    {
        U1TXREG = getFromOutputRaw();    //Transmetem la dada
    }
}

//INTERRUPCió QUE S'ACTIVA QUAN EL ADC HA ACABAT LA CONVERSIó
#define INT_ADC1
void ADC1_isr(void)
{
    float32 H1_act = 0;
    float32 H2_act = 0;
    //float32 H3_act = 0;
    //float32 H4_act = 0;

    H1_act = filterH1(ADC_LUT[ADCBUF0]);
    //printf("H1:%f\n\r",H1_act);
    //H4_act = ADC_LUT[ADCBUF1];
    //printf("H4:%f\n\r",H4_act);
    //H3_act = ADC_LUT[ADCBUF2];
    //printf("H3:%f\n\r",H3_act);
    H2_act = filterH2(ADC_LUT[ADCBUF3]);
    //printf("H2:%f\n\r",H2_act);

    //MÀQUINA D'ESTATS
    switch (estat)
    {
        case 1:    //Comprovem que el valor d'H1 passi un valor de
referencia

                if(H1_act >= VREF)
                {
                    if(H1_act < H1_ant)
                    {
                        pujaBaixaH1 = 1;
                    }
                    else
                    {
                        pujaBaixaH1 = 0;
                    }

                    H1_ant = H1_act;

                    //printf("PBH1:%d\n\r",pujaBaixaH1);
                    //printf("H1_ant:%e\n\r",H1_ant);

                    TMR2 = 0;
                    enable_interrupts(INT_TIMER2);    //Habilitem el
temporitzador 2

                    TON = 1;
                    estat = 2;
                }

                H1_ant = H1_act;
                H2_ant = H2_act;

```

```

        break;

    case 2:        //Mirem si el valor d'H1 i d'H2 coincideixen

        //printf("H2_act:%e\n\r",H2_act);
        //printf("if(%e    >=    %e)=%d\n\r",H2_act,H1_ant,(H2_act    >=
H1_ant));

        if(H2_act >= H1_ant)
        {
            H2_ant = H2_act;
            H2_temp = TMR2;        //Llegim el valor acutal del
temporitzador

            estat = 3;
        }

        H2_ant = H2_act;

        break;

    case 3:        //Comprovem el sentit dels hidròfons

        if(H2_act < H2_ant)
        {
            pujaBaixaH2 = 1;
        }
        else
        {
            pujaBaixaH2 = 0;
        }

        //printf("PBH2:%d\n\r",pujaBaixaH2);

        if(pujaBaixaH1 == pujaBaixaH2)
        {
            estat = 4;
        }
        else
        {
            estat = 2;
        }

        break;
    }

    //COMPROVEM ELS VALORS MÀXIMS I MÍNIM DE CADA HIDRÒFON
    if(H1_act > H1_Vmax)
    {
        H1_Vmax = H1_act;
    }
    else
    {
        if(H1_act < H1_Vmin)
        {
            H1_Vmin = H1_act;
        }
    }

    if(H2_act > H2_Vmax)
    {
        H2_Vmax = H2_act;
    }
    else
    {
        if(H2_act < H2_Vmin)
        {
            H2_Vmin = H2_act;
        }
    }

```

```

    }
}

//INTERRUPCIÓ DEL TIMER 2
#int_TIMER2
void TIMER2_isr(void)
{
    TMR2 = 0;
    estat = 0;
}

/*****
*                                     PROGRAMA PRINCIPAL                                     *
*****/

void main()
{
    configuracio();

    //DEFINICIÓ DE VARIABLES LOCALS
    float32 AngleH12;
    int8 caracter;
    float32 H2_tempDelay;

    printf ("INICIALITZANT...\n\r");

    //ad5282_write_val(AD5282_1_ADDR,0,VAL_POT);
    //ad5282_write_val(AD5282_1_ADDR,1,VAL_POT);

    ad5282_write_val(AD5282_2_ADDR,0,VAL_POT);
    ad5282_write_val(AD5282_2_ADDR,1,VAL_POT);

    //INICI DEL BUCLE INFINIT
    while(true)
    {
        switch (estat)
        {
            case 0:          //Inicialitzem el convertidor ADC

                ADON = 1;
                enable_interrupts(INT_ADC1);

                estat = 1;

                break;

            case 4:          //Calculem el valor de l'angle

                disable_interrupts(INT_TIMER2);
                disable_interrupts(INT_ADC1);
                ADON = 0;
                TON = 0;
                H2_tempDelay = tempsDelay(H2_temp);
                printf("T2:%Ld,H2D:%e\n\r",H2_temp,H2_tempDelay);
                angleH12 = angleEmissor(H2_tempDelay, SOUNDVEL, DISTH);
                printf("AH12:%f\n\r\n\r",angleH12);

                estat = 0;

                break;

        }

    }

    //COMPROVACIÓ QUE NO HI HAGI OVERRUN ERROR
    if (U1STA & 0x02)      //Si hi ha overrun error en el port RS232...
    {
        U1STA = 0x80;      //Netejem el CREN bit
    }
}

```

```

        U1STA = 0x90;        // Posa a 1 el bit CREN i SPEN
    }
}

/*****
*
*          FUNCIO DE CONFIGURACIO
*
*****
*   Aquesta funcio ens permet fer la configuració dels diferents mòduls del
*   dsPIC. Aquests són el mòdul UART, I2C i l'ADC
*****
void configuracio()
{
    //Inicialització dels buffers d'emmagatzematge rotatius
    inputRawInit();
    outputRawInit();

    //Inicialització dels potenciómetres a un determinat valor VAL_POT
    ad5282_write_val(AD5282_1_ADDR,0,VAL_POT);
    ad5282_write_val(AD5282_1_ADDR,1,VAL_POT);
    ad5282_write_val(AD5282_2_ADDR,0,VAL_POT);
    ad5282_write_val(AD5282_2_ADDR,1,VAL_POT);

    //Configuració del modul ADC i dels seus registres
    ADPCFG = 0b111111110000111;
        /*111111110000111 AN15-AN7 & AN2-AN0 in digital mode
        -----0000--- AN3,AN4, AN5 & AN6 in analog mode*/
    ADCON1 = 0b0000000011101100;
        /* -X-XXX-----X----- Unimplemented, set to 0
        0----- ADON: A/D Operating Mode bit
        --0----- ADSIL: Stop in Idle Mode bit
        -----00----- FORM<1:0> Data Output Format bits
        -----111----- SSRC<2:0> Conversion Trigger Source Select bits
        -----1----- SIMSAM: Simultaneous Sample Select bit
        -----1-- ASAM: A/D Sample Auto-Start bit
        -----0-- SAMP: A/D Sample Enable bit
        -----X DONE: A/D Conversion Status bit (Read only)*/
    ADCHS = 0b0000000000100110;
        /*XXXXXXX----- MUX B input selections, not used
        -----0x----- CH123NA<1:0> MUX A CH1-CH3 negative input is Vref-
        -----1----- CH123SA: MUX A CH1 positive input is AN3, CH2
        positive input is AN4, CH3 positive input is AN5
        -----0----- CH0NA: MUX A CH0 negative input is VREF-
        -----0110 CH0SA<3:0> MUX A CH0 positive input is AN6*/

    ADCSSL = 0b0000000000000000; //No channels participate in input scan
    ADCON3 = 0b0000110000000100;
        /*XXX-----X----- Unimplemented, set to 0
        ---01100----- SAMC<4:0> Auto-Sample Time bits
        -----0----- ADRC: A/D Conversion Clock Source bit
        -----000100 ADCS<5:0> A/D Conversion Clock Select bits*/
    ADCON2 = 0b0110001100000000;
        /*---XX---X----- Unimplemented, set to 0
        011----- VCFG<2:0> Voltage Reference Configuration bits
        -----0----- CSCNA: Scan Input Selections for CH0+ S/H Input for
MUX A Input Multiplexer Setting bit
        -----11----- CHIPS<1:0> Selects Channels Utilized bits
        -----X----- BUFS: Buffer Fill Status bit
        -----0000-- SMPI<3:0> Sample/Convert Sequences Per Interrupt
Selection bits
        -----0-- BUFM: Buffer Mode Select bit
        -----0 ALTS: Alternate Input Sample Mode Select bit*/

    ADON = 0; //Posar a 1 per conversió ADC

    //Configuració de les interrupcions de cada mòdul
    setup_timer1(TMR_DISABLED);
    //Configuració del timer 2

```

```

TCKPS0 = 0;          //Timer Input Clock Presale Select bits 0
TCKPS1 = 0;          //Timer Input Clock Presale Select bits 1
T32 = 0;             //32-bit Timer Mode Select bits
TCS = 0;             //Timer Clock Source Select bit
TON = 0;             //Timer On Control bit
TMR2 = 0;            //Temps d'overflow = 8.191875ms
setup_timer4(TMR_DISABLED | TMR_DIV_BY_1 ,0);
setup_timer3(TMR_DISABLED | TMR_DIV_BY_1 ,0);
setup_timer5(TMR_DISABLED | TMR_DIV_BY_1 ,0);
enable_interrupts(INT_RDA);
disable_interrupts(INT_TBE);
disable_interrupts(INT_ADC1);
disable_interrupts(INT_EEPROM);
enable_interrupts(INT_TIMER2);
}

/*****
*                               FUNCIÓ DE FILTRATGE                               *
*****/
*   Aquesta funció ens permet filtrar el senyal d'entrada dels hidròfons   *
*   a partir d'un filtre Butterworth de X ordre                           *
*****/
float32 filterH1(float32 in)
{
    float32 yt;

    yt = 3.97450921470395*yt1-5.9350134244133876*yt2+3.9463557967817047*yt3-
0.98588336209461136*yt4+0.000025087639167625506*in-
0.000050175278335251013*xt2+0.000025087639167625506*xt4;

    yt4 = yt3;
    yt3 = yt2;
    yt2 = yt1;
    yt1 = yt;
    xt4 = xt3;
    xt3 = xt2;
    xt2 = in;

    return yt;
}

float32 filterH2(float32 in)
{
    float32 yth2;

    yth2 = 3.97450921470395*yt1h2-5.9350134244133876*yt2h2+3.9463557967817047*yt3h2-
0.98588336209461136*yt4h2+0.000025087639167625506*in-
0.000050175278335251013*xt2h2+0.000025087639167625506*xt4h2;

    yt4h2 = yt3h2;
    yt3h2 = yt2h2;
    yt2h2 = yt1h2;
    yt1h2 = yth2;
    xt4h2 = xt3h2;
    xt3h2 = xt2h2;
    xt2h2 = in;

    return yth2;
}

/*****
*                               FUNCIÓ LLEGIR POTENCIOMETRE                               *
*****/
*   Aquesta funció ens permet llegir els valors de resistència dels potenci- *
*   ometres indicant la adreça, tenir en compte que per llegir el pot1 avans *
*   s'ha d'haver escrit en el pot1 i de la mateixa forma pel pot2.         *
*****/
unsigned int16 ad5282_read_val(unsigned int8 adr)

```

```

{
    unsigned int8 read = 0b00000001;
    unsigned int8 data;

    adr <= 1;
    adr |= read;
    i2c_start();
    i2c_write(adr);
    data = i2c_read();
    i2c_stop();

    return data;
}

/*****
 *          FUNCIO ESCRIURE POTENCIOMETRE
 *****/
*   Aquesta funció ens permet escriure un valor de resistència en els potenci-
*   ometres, indicant la adreça d'aquest, el pot que volem escollir i finalment
*   el valor de resistència que hi volem escriure (0-255)
*****/
void ad5282_write_val(unsigned int8 adr, int1 pot, unsigned int8 data)
{
    unsigned int8 instr;

    adr <= 1;
    adr &= 0xFE;
    instr = pot;
    instr <= 7;
    instr &= 0x80;

    i2c_start();
    i2c_write(adr);
    i2c_write(instr);
    i2c_write(data);
    i2c_stop();
}

/*****
 *          FUNCIO PER DETERMINAR EL TEMPS
 *****/
*   Aquesta funció el que fa és un cop aconseguida la difereència entre els one
*   transforma el valor actual del timer per un valor real de temps
*****/
float32 tempsDelay(unsigned int16 stepTim)
{
    int8 prescaler = 1;    //Valor del prescaler del temporitzador
    float32 delay = 0;

    delay = Tcy*prescaler*stepTim;

    return delay;
}

/*****
 *          FUNCIO QUE PERMET CALCULAR L'ANGLE A PARTIR DEL DELAY DE TEMPS
 *****/
*   Aquesta funció se li passa el valor de temps de delay entre hidròfons i
*   retorna el valor de l'angle real que hi ha entre aquests i l'emissor
*   acústic. Els paràmetres que se li passen són el delay de temps entre hidrò-
*   fons, la velocitat del so a l'aigua i la distància entre els hidròfons
*****/
float32 angleEmissor(float32 temp, float32 soundVel, float32 dist)
{
    float32 angle = 0;
    float32 tempo = 0;
    tempo = (temp*soundVel)/dist;
    angle = asin(tempo);
}
    
```

```
    return angle;
}

INPUTRAW.H

#ifndef INPUTRAW_H
#define INPUTRAW_H

#define DATA_ELEMENTS 16

int8 inputRaw[DATA_ELEMENTS];
int8 inputRawElements;
int8 inputRawWritePointer;
int8 inputRawReadPointer;

void inputRawInit ();
void putInputRaw (int8 data);
int8 getFromInputRaw ();
int1 inputRawEmpty ();
int1 inputRawFull ();

void inputRawInit ()
{
    inputRawElements = 0;
    inputRawWritePointer = 0;
    inputRawReadPointer = 0;
}

int1 inputRawFull ()
{
    if (inputRawElements == DATA_ELEMENTS)
    {
        return 1;
    }
    else
    {
        return (0);
    }
}

int1 inputRawEmpty ()
{
    if (inputRawElements == 0)
    {
        return 1;
    }
    else
    {
        return (0);
    }
}

// puts a new Data into the input Raw
void putInputRaw (int8 data)
{
    if ( !inputRawFull () )
    {
        inputRaw[inputRawWritePointer] = data;
        inputRawWritePointer++;
        if ( inputRawWritePointer == DATA_ELEMENTS )
        {
            inputRawWritePointer = 0;
        }
        inputRawElements ++;
    }
}

int8 getFromInputRaw ()
```

```

{
    int8 dataRead;

if ( !inputRawEmpty () )
{
    dataRead = inputRaw[inputRawReadPointer];
    inputRawReadPointer++;
    if ( inputRawReadPointer == DATA_ELEMENTS )
    {
        inputRawReadPointer = 0;
    }
    inputRawElements --;
    return (dataRead);
}
else
{
    return (0xFF);
}
}

```

```

#endif

```

```

OUTPUTRAW.H

```

```

#include <string.h>

```

```

#ifndef OUTPUTRAW_H

```

```

#define OUTPUTRAW_H

```

```

#define DATAO_ELEMENTS 32

```

```

#if DATAO_ELEMENTS > 255

```

```

    int16 outputRawElements;

```

```

    int16 outputRawWritePointer;

```

```

    int16 outputRawReadPointer;

```

```

#else

```

```

    int8 outputRawElements;

```

```

    int8 outputRawWritePointer;

```

```

    int8 outputRawReadPointer;

```

```

#endif

```

```

int8 outputRaw[DATAO_ELEMENTS];

```

```

void outputRawInit ();

```

```

void putOutputRaw (int8 data);

```

```

int8 getFromOutputRaw ();

```

```

int1 outputRawEmpty ();

```

```

int1 outputRawFull ();

```

```

void outputRawInit ()

```

```

{

```

```

    outputRawElements = 0;

```

```

    outputRawWritePointer = 0;

```

```

    outputRawReadPointer = 0;

```

```

}

```

```

int1 outputRawFull ()

```

```

{

```

```

    if (outputRawElements == DATAO_ELEMENTS)

```

```

    {

```

```

        return 1;

```

```

    }

```

```

    else

```

```

    {

```

```

        return (0);

```

```

    }

```

```

}

```

```

int1 outputRowEmpty ()
{
    if (outputRowElements == 0)
    {
        return 1;
    }
    else
    {
        return (0);
    }
}

// puts a new Data into the input Row
void putOutputRaw (int8 data)
{
    if ( !outputRowFull () )
    {
        outputRaw[outputRawWritePointer] = data;
        outputRawWritePointer++;
        if ( outputRawWritePointer == DATAO_ELEMENTS )
        {
            outputRawWritePointer = 0;
        }
        outputRowElements ++;
    }
}

int8 getFromOutputRaw ()
{
    int8 dataRead;

    if ( !outputRowEmpty () )
    {
        dataRead = outputRaw[outputRawReadPointer];
        outputRawReadPointer++;
        if ( outputRawReadPointer == DATAO_ELEMENTS )
        {
            outputRawReadPointer = 0;
        }
        outputRowElements --;
        return (dataRead);
    }
    else
    {
        return (0xFF);
    }
}

void putOnOutputRaw(int8 *str)
{
    int16 i;

    for(i = 0; i<strlen(str); i++)
    {
        putOutputRaw(str[i]);
    }
}

void putOnOutputRawL(int8 *str, int16 len)
{
    int16 i;

    for(i = 0; i<len; i++)
    {
        putOutputRaw(str[i]);
    }
}

```

```
#endif

SOUNDLOCATOR.H

#include <30F4011.h>

#FUSES NOWDT          //No Watch Dog Timer
#FUSES WPSA8          //Watch Dog Timer PreScalar A 1:8
#FUSES WPSB1          //Watch Dog Timer PreScalar B 1:1
#FUSES HS              //High speed Osc (> 4mhz for PCM/PCH) (>10mhz
for
PCD)
#FUSES CKSFSM          //Clock Switching is enabled, fail Safe clock
monitor is enabled
#FUSES NOBROWNOUT     //No brownout reset

#device CONST=ROM
#device *=16

#use delay(clock=32000000)
#use fixed_IO(E_outputs = PIN_E0,PIN_E0,PIN_E1,PIN_E1,PIN_E2,PIN_E2)
```