

Llicència GPL

Pot copiar i distribuir el Programa (o un treball basat en ell), segons s'especifica en l'apartat 2, com a codi objecte o en format executable segons els termes dels apartats 1 i 2, suposant que a més compleixi una de les següents condicions:

1. Acompanyar-ho amb el codi font complet corresponent, en format electrònic, que ha de ser distribuït segons s'especifica en els apartats 1 i 2 d'aquesta llicència en un mitjà habitualment utilitzat per a l'intercanvi de programes, o
2. Acompanyar-ho amb una oferta per escrit, vàlida durant almenys tres anys, de proporcionar a qualsevol tercera part una còpia completa en format electrònic del codi font corresponent, a un cost no major que el de realitzar físicament la distribució de la font, que serà distribuïda sota les condicions descrites en els apartats 1 i 2 anteriors, en un mitjà habitualment utilitzat per a l'intercanvi de programes, o
3. Acompanyar-ho amb la informació que va rebre oferint distribuir el codi font corresponent. (Aquesta opció es permet només per a distribució no comercial i només si vostè va rebre el programa com a codi objecte o en format executable amb tal oferta, d'acord amb l'apartat 2 anterior).

UNIVERSITAT OBERTA DE CATALUNYA

Enginyeria Tècnica de Telecomunicació especialitat Telemàtica

DISSENY D'UN SISTEMA DE FILTRAT D'ÀUDIO

Alumne: Marc Güell Cargol
Dirigit per: Asier Ibeas Hernández
CURS 2009-10 (Febrer)

A la meva família,

Agraïments,

Al director Asier Ibeas per la seva ajuda i suport

Resum

Tant el medi transmissor com els equips d'enregistrament o reproducció de so introdueixen components de soroll d'alta freqüència als senyals. En aquest treball de final de carrera (TFC), s'ha dissenyat i implementat un sistema de filtrat d'àudio encaminat a filtrar aquestes components d'alta freqüència. Donat que l'oïda humana no pot percebre sons de més de 20 kHz, s'ha considerat aquest límit com a freqüència màxima a mantenir en la senyal.

S'ha començat estudiant el senyal problema a través del seu espectre de freqüències simulat mitjançant la transformada discreta de Fourier (DFT, en anglès). Una vegada identificades les components d'alta freqüència a atenuar, s'han estudiat les diferents opcions de filtre passabaix.

Inicialment, s'ha valorat la possibilitat del disseny de filtres analògics de Butterworth o Chebyshev, o de filtres digitals de tipus IIR (*Infinite Impulse Response*) basats en els primers. Tanmateix, malgrat assolir les especificacions en magnitud, mitjançant aquest filtre no s'obté una fase lineal en la banda de pas. Per això, s'ha realitzat un disseny de filtre digital tipus FIR (*Finite Infinite Response*) que compleix estrictament amb les especificacions i presenta una fase lineal en la banda de pas. S'ha simulat el comportament d'aquest filtre amb el senyal problema per tal d'assegurar el seu correcte funcionament.

A continuació, s'ha implementat aquest últim disseny en llenguatge C i compilat per un microcontrolador de l'empresa Microchip. S'han realitzat proves de simulació mitjançant Stimulus del programa MPLAB.

En definitiva, s'ha dissenyat un filtre passabaix de tipus FIR per acondicionar una senyal d'àudio que posteriorment s'ha implementat en un microcontrolador de Microchip.

Índex de la memòria

1 Plantejament del projecte.....	7
1.1 Introducció.....	7
1.1.1 Àudio digital	7
1.1.2 Filtres.....	8
1.2 Objectius	10
1.3 Material.....	11
1.4 Planificació	12
1.4.1 Etapes del projecte.....	13
ETAPA I: Determinació de les especificacions del filtre.....	13
ETAPA II: Discutir les diferents opcions de disseny.....	13
ETAPA III: Disseny del filtre	13
ETAPA IV: Comprovació del resultats.....	13
ETAPA V: Implementació del filtre.....	13
ETAPA VI: Finalització de la documentació.....	13
1.4.2 Temporització i diagrama de Gantt	14
1.4.3 Incidències i riscos.....	15
2 Determinació de les especificacions del filtre	16
2.1 Estudi de el senyal	16
2.2 Especificacions del filtre.....	18
3 Disseny del filtre.....	19
3.1 Disseny analògic.....	20
3.2 Disseny digital.....	25
3.2.1 Disseny del filtre IIR.	27
3.2.2 Disseny del filtre FIR.....	30
4 Comprovació dels resultats.....	33
5 Implementació del filtre.....	35
5.1 Introducció als microcontroladors.....	35
5.2 Els microcontroladors PIC de Microchip.....	35
5.3 Desenvolupament d'un sistema mitjançant microcontroladors PIC.....	37
5.4 Elecció del dsPIC30F6010A de Microchip.....	38
5.5 Implementació del filtre en un dsPIC30F6010A de Microchip.....	39
5.7 Implementació del filtre en codi C.....	40
5.8 Comprovació dels resultats.....	42
6 Valoració econòmica.....	45
7 Conclusions.....	46
7.1 Resultats obtinguts.....	46
7.2 Treballs futurs.....	46
8 Glossari.....	47
9 Bibliografia.....	49
10 Annexos.....	50
10.1 Annex I: Filtres analògics (Butterworth).....	50
10.2 Annex II: Especificacions del microcontrolador.....	51
10.3 Annex III: Codi C de la implementació del chip.....	55

Índex de figures

Figura 1. Esquema de la conversió d'una senyal analògica a digital.....	7
Figura 2. Corbes d'amplificació ideals dels filtres mencionats.....	8
Figura 3: Esquema general del projecte.....	12
Figura 4: Diagrama de Gantt.....	14
Figura 5: Senyal problema centrada en la zona d'interès (20,000 Hz). En l'eix d'absisses es representen freqüències i en l'eix d'ordenades magnitud.....	17
Figura 6: Representació del filtre normalitzat.....	20
Figura 7: Resposta en freqüència d'un filtre de Butterworth.....	21
Figura 8: Representació del guany en funció de la freqüència prop del final de la zona de transició. Només el filtre d'ordre 63 (blau) permet un guany per sota de 0.05, el filtre d'ordre 62 (vermell) no permet complir amb les especificacions.....	22
Figura 9: Resposta en freqüència d'un filtre de Chebyshev.....	23
Figura 10: Resposta del filtre analògic de Chebishev de tipus I i ordre 20.	27
Figura 11: Magnitud de resposta del filtre digital dissenyat.....	28
Figura 12: Espectre de el senyal sense filtrar.....	28
Figura 13: Espectre de el senyal filtrada.....	29
Figura 14: Resposta en magnitud i fase del filtre tipus FIR dissenyat.....	32
Figura 15: Espectre del senyal de sortida.....	33
Figura 16: Variació de la fase en funció de la freqüència.....	34
Figura 17: Exemple d'un PIC de Microchip Technology.....	36
Figura 18: Pins disponibles en el dsPIC30F6010A de Microchip.....	39
Figura 19: Diagrama de flux del programa.....	41
Figura 20: Configuració d'Stimulus per simular el mòdul ADC.....	42
Figura 21: Configuració de la sortida mitjançant el simulador UART1.....	43

Índex de taules

Taula 1: Temporització detallada de les diferents tasques del projecte.....	14
Taula 2: Especificacions del filtre a dissenyar.....	18
Taula 3: Diferents microcontroladors proposats.....	38
Tabla 4: Comparació dels resultats del disseny i de la implementació.....	43
Taula 5: Resum de la valoració econòmica.....	45

1 Plantejament del projecte

Dintre el marc del TFC es procedirà al disseny i implementació d'un filtre per condicionar una senyal d'àudio proposada. Es començarà fent un estudi espectral de el senyal, a continuació es realitzaran diferents propostes de filtres, tant analògics com digitals. Finalment, es realitzarà la implementació del filtre en un sistema adequat.

Amb ànim de posar en context el projecte es realitzarà una breu introducció als filtres i als sistemes d'àudio digital.

1.1 Introducció

En aquest primer apartat s'introduirà breument l'àudio digital, i el filtratge per tal d'acondiconar senyals.

1.1.1 Àudio digital

L'àudio digital utilitza senyals digitals per la reproducció del so. El processament del senyal inclou diferents etapes de conversions d'analògic a digital, de digital a analògic, de transformació, d'emmagatzematge o de transmissió. A diferència dels sistemes analògics, els digitals són discrets en el temps i en els nivells. El senyal analògic es mostra a una freqüència determinada, i cada valor de mostreig s'emmagatzema en un número de bits determinat (figura 1; veure com el senyal analògic es discretitza, cada punt de mostreig correspon amb una de les línies verticals, cada valor digital permès coincideix per cada línia horitzontal). El número de valors permesos depèn del número de bits emprats. Degut a aquesta naturalesa discreta, és pot fer el tractament de el senyal digital sense distorsió. És a dir, petites perturbacions introduïdes dins els diferents passos del tractament no es transmeten degut a que es permeten només determinats nivells. A més, els sistemes digitals implementen sistemes de detecció i correcció d'errors de gran eficiència. Per això, el tractament digital s'utilitza massivament en enregistrar, manipular i distribuir el so. Mentre que els circuits analògics són més susceptibles a soroll i distorsió intrínseca, en els digitals, essencialment aquest problemes es redueixen a les etapes analògiques.

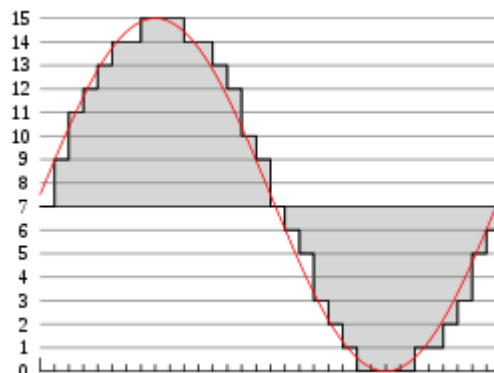


Figura 1. Esquema de la conversió d'una senyal analògica a digital

Els sistemes d'àudio digitals fan un ús intensiu de diferents filtres. Ja sigui per tal d'implementar funcionalitats com pot ésser controlar els tons greus i aguts del estèreo. O bé, per tal d'eliminar sorolls no desitjats que disminueixen la qualitat en què percebem el senyal. Aquest soroll està causat per senyals d'alta freqüència que poden tenir diferents origen. Per una banda, poden ser injectats al dispositiu d'àudio des de fons externes. Per exemple, a degut a imperfeccions mecàniques dels micròfons. Per l'altra, també pot aparèixer soroll causat pels transitoris de canvi de fase dels semiconductors dels components electrònics. L'ànim de qualsevol equip de so és proporcionar una bona qualitat sonora, per això es fa ús de filtres com a part del tractament digital per tal d'eliminar aquests components no desitjats.

Com s'indica en el títol d'aquest TFC, es dissenyarà un sistema de filtrat d'àudio. Aquest haurà de ser capaç d'eliminar els components d'alta freqüència no desitjats d'un senyal adjuntada en l'enunciat. Donat que es tracta d'un sistema de filtrat d'àudio, només té sentit conservar les freqüències que la oïda humana és capaç de percebre (inferiors a 20 kHz). Per tant, es dissenyarà un filtre passabaix per tal d'eliminar qualsevol component de freqüència superior a 20 kHz.

1.1.2 Filtres

En processament de senyals, un filtre és un dispositiu que modifica un component no desitjat d'un senyal, modificant tant la seva amplitud com la seva fase.

Un filtre es descriu segons la seva funció de transferència. Aquesta representa de forma matemàtica la forma en què el senyal aplicat canvia en amplitud i fase en travessar el filtre. Segons la transformació que realitzen a el senyal, els filtres es poden classificar en diferents tipus (veure figura 2):

- Filtre passabaix. Deixa passar inalterades les freqüències baixes però atenua les altres.
- Filtre passalt. Deixa passar inalterades les freqüències altes però atenua les baixes.
- Filtre passabanda. Deixa passar inalterades únicament un conjunt de freqüències dintre un interval determinat i atenua la resta.
- Filtre banda eliminada. Deixa passar inalterades totes les freqüències excepte aquelles compreses dintre un cert interval.

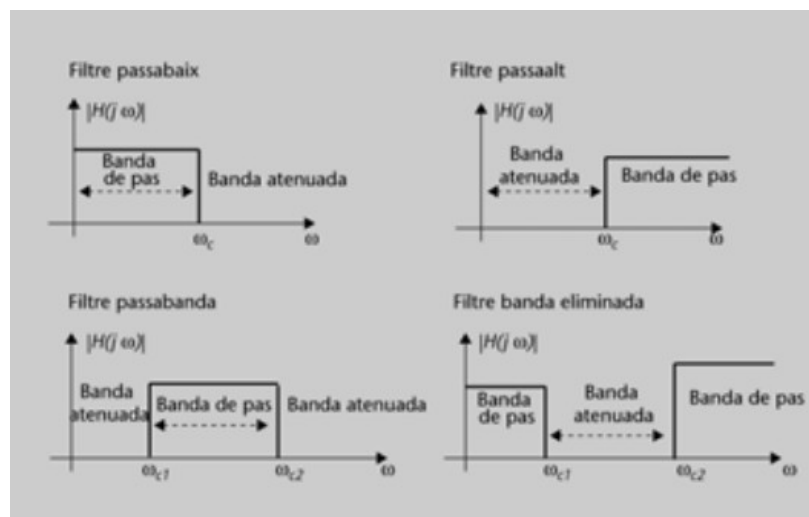


Figura 2. Corbes d'amplificació ideals dels filtres mencionats.

Els filtres es poden construir mitjançant un gran nombre de tecnologies. Moltes vegades, la mateixa funció de transferència es pot implementar de diferents maneres. Els més freqüents es comenten breument a continuació:

- Els filtres analògics consisteixen de una combinació de resistències, bobines i condensadors. A més, poden també incloure elements actius com amplificadors operacionals. Aquests filtres són utilitzats per senyals en temps continu.
- Els filtres digitals són la implementació d'un algorisme matemàtic que correspon a la funció de transferència desitjada. Normalment s'implementen mitjançant programari sobre un microprocessador o directament en maquinari digital. Aquests s'empren per senyals de naturalesa discreta o contínues mostrejades.

La transmissió de senyals és un aspecte clau en el nostre dia a dia. Tots fem ús intensiu de telèfons mòbils, equips de so, etc. El so que percebem sol haver passat per un medi hostil, com també per diferents elements electrònics que el transformen. Per això, és essencial tenir eines per recuperar les parts del senyal desitjades.

1.2 Objectius

L'objectiu d'aquest TFC és dissenyar un sistema de filtrat d'àudio que permeti eliminar un component no desitjat d'alta freqüència en una senyal de prova. Per aconseguir aquest objectiu es pretén:

- Analitzar l'espectre del senyal problema obtenint la seva distribució espectral i les bandes associades al soroll.
- Analitzar les diferents alternatives de filtres disponibles i escollir la més adequada.
- Dissenyar el filtre triat i comprovar el seu funcionament mitjançant una simulació.
- Implementar el filtre dissenyat, ja sigui un circuit electrònic, o bé un algorisme matemàtic programat en un PIC.
- Dur a terme les simulacions adequades sobre el disseny final per tal d'assegurar que el filtre es comporta com dicten les especificacions.

Cada una d'aquestes parts contribueix a assolir l'objectiu final i constitueixen fites del projecte verificables i constatables amb el seu corresponent resultat final. Fer aquest desglossament de l'objectiu últim permet, a més, no fer una valoració binària del resultat del projecte, és a dir, no classificar-lo només com d'èxit o fracàs sinó que es pot graduar quines coses s'han aconseguit i quins no. També fan una mica més fàcil la distribució del projecte en tasques.

1.3 Material

El material que es preveu necessari és el següent. Es separa en necessitats de programari i de maquinari:

Programari

- Software específic:
 - Scilab. Llenguatge de programació d'alt nivell lliure per el càlcul científic. S'emprarà per analitzar el senyal problema i dissenyar els filtres.
 - Matlab. Llenguatge de programació d'alt nivell propietari per el càlcul científic. S'emprarà per analitzar el senyal problema i dissenyar els filtres de manera complementària a Scilab.
 - MAX+PLUSII. Entorn per tal de dissenyar i simular circuits integrats.
 - Pspice. Programa per tal de simular circuits analògics. S'utilitzarà per simular filtres analògics.
 - MPLAB IDE. És un entorn per tal de seleccionar microcontroladors, realitzar la programació, compilació i simulació.
 - Camtasia. Programari per la captura i edició de vídeos.
- Software d'espectre general: OpenOffice 3.0, Ubuntu 8.04

Maquinari: Punt de treball estàndard de la UOC

- Portàtil IBM-ThinkPad T60, Intel Core 2 T7200, 4GB RAM
Connexió a Internet tipus ADSL

1.4 Planificació

En aquest apartat es realitza la planificació general del projecte. Es comença dividint el projectes en etapes, que es corresponen amb unitats funcionals del projecte. Una vegada definides es temporitzen mitjançant un taules i diagrames de Gantt.

Es comença fent un estudi de el senyal de partida, i així s'especifica les propietats necessàries pel filtre (Etapa I). A continuació, s'estudien les diferents opcions de disseny i es raona quin tipus de disseny és el més adequat (Etapa II). Una vegada decidit, es realitza el disseny en l'etapa III, i en la IV es comprova que compleix amb les especificacions. Una vegada vist el bon funcionament, es procedeix a la seva implementació (Etapa V). Finalment, es finalitza la presentació i la memòria que s'haurà anat confeccionant de manera progressiva al llarg de totes les fases. A la figura 3 podem es pot veure un esquema general del projecte de manera gràfica. Es comença determinant les especificacions, i s'acaba simulant la implementació.

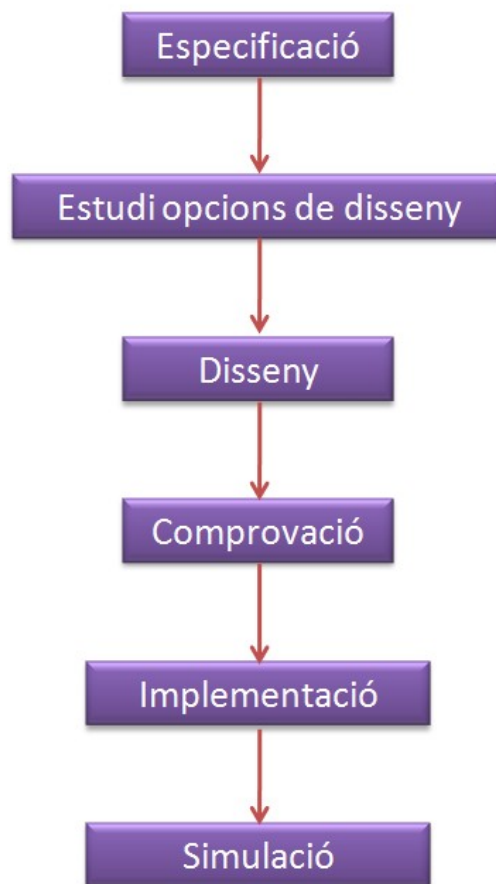


Figura 3: Esquema general del projecte

1.4.1 Etapes del projecte

Per tal de poder desenvolupar el projecte cal finalitzar les següents 6 etapes compostes d'una sèrie de tasques.

ETAPA I: Determinació de les especificacions del filtre

- Tasca 1.1: S'estudiarà el senyal problema i la part que se'n vol eliminar mitjançant la transformada discreta de Fourier. Es farà ús del software Scilab i el fitxer signal.dat.
- Tasca 1.2: Es determinarà el tipus de filtre que és adequat per realitzar la transformació desitjada.

ETAPA II: Discutir les diferents opcions de disseny

- Tasca 2.1: S'analitzaran els diferents avantatges i desavantatges de cada tipus de filtre
- Tasca 2.2: Es decidirà si es dissenya un filtre analògic o un de digital i de quin tipus serà.

ETAPA III: Disseny del filtre

- Tasca 3.1: S'escollirà un procediment de disseny i es realitzarà el disseny.
- Tasca 3.2: L'obtenció de l'equació que descriu la funció de transferència del filtre.

ETAPA IV: Comprovació del resultat

- Tasca 4.1: Es realitzaran simulacions del filtre dissenyat mitjançant el software Scilab sobre el senyal signal.dat

ETAPA V: Implementació del filtre

- Tasca 5.1: En cas d'escollir-se un filtre analògic, es dibuixarà l'esquema electrònic del dispositiu mitjançant el software Pspice. En cas d'escollir un filtre digital, s'escollirà un microprocessador PIC per implementar-lo.
- Tasca 5.2: En cas d'escollir-se un filtre analògic, es dissenyarà la placa de circuit imprès mitjançant el software MAX+plus II. En cas d'escollir un filtre digital, caldrà programar el microprocessador escollit.
- Tasca 5.3: En qualsevol dels dos casos caldrà realitzar simulacions per tal de comprovar que el disseny es comporta com esperem.

ETAPA VI: Finalització de la documentació

- Tasca 6.1: Finalització de la memòria mitjançant el software OpenOffice 3.0. Cal destacar que a mesura que es finalitzin les etapes anteriors s'anirà documentant el treball fet. Per tant, en l'etapa VI es considera la revisió i edició final dels apartats 1-6, juntament amb generació de la documentació relacionada amb el tancament del projecte.
- Tasca 6.2: Preparar la presentació per el debat virtual mitjançant el software OpenOffice 3.0.

1.4.2 Temporització i diagrama de Gantt

A continuació (Taula 1) es presenta la temporització detallada prevista per dur a terme les tasques descrites en l'apartat anterior.

TASCA	DATES	TEMPS	COMENTARIS
1,1	8-14 Març	6 hores	PAC1 (9 de març)
1,2	15-21 Març	6 hores	
2,1	29-31 Març	3 hores	
2,2	1-4 Abril	3 hores	
3,1	5-7 Abril	3 hores	
3,2	8-11 Abril	3 hores	
4,1	19-22 Abril	6 hores	
5,1	23-25 Abril	6 hores	PAC2 (20 d'abril)
5,2	26 Abril- 2 Maig	6 hores	
5,3	3-23 Maig	6 hores	
6,1	10 Maig-7 Juny	24 hores	PAC3 (18 de maig), Memòria (7 de juny)
6,2	8-13 Juny 21-23 Juny	6 hores	Presentació virtual (23 de juny)

Taula 1: Temporització detallada de les diferents tasques del projecte.

En la figura 4 que es mostra a continuació es mostra el diagrama de Gantt del projecte. Es recullen totes les tasques de la taula 1 representant la seva temporització. En la figura 4 es poden veure de manera gràfica les dates associades amb cada tasca.

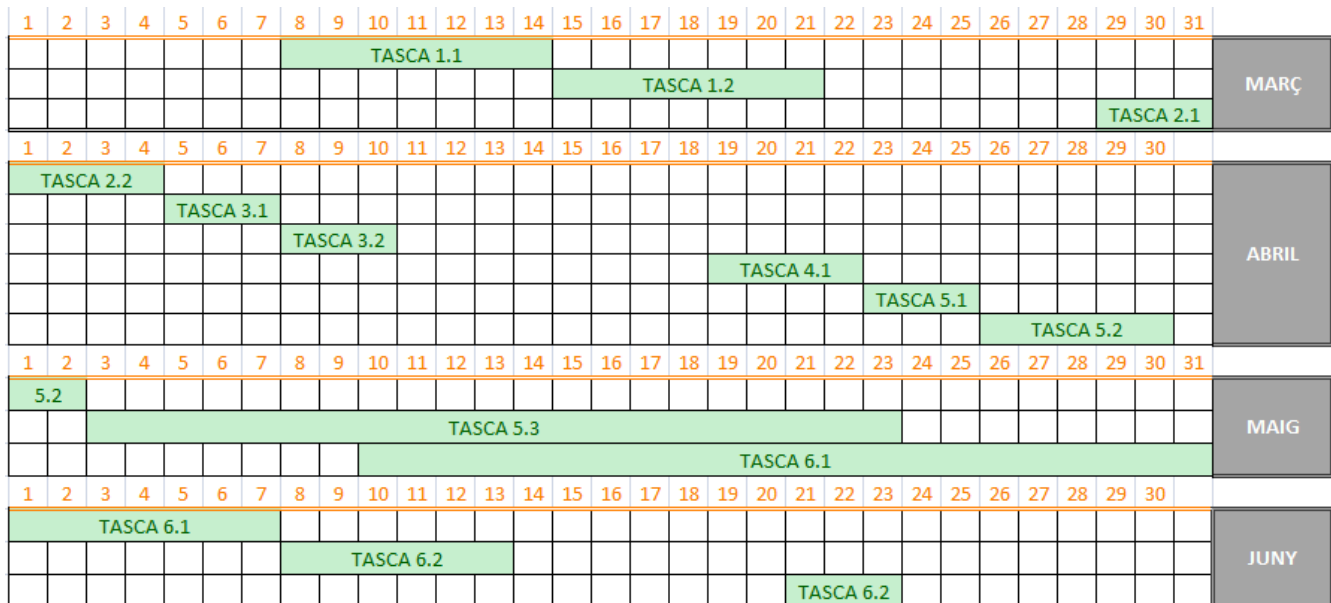


Figura 4: Diagrama de Gantt

1.4.3 Incidències i riscos

Es presenten a continuació dos grups de circumstàncies que podrien alterar el calendari proposat. El primer grup, fa referència a fets imprevisibles que poden passar o no. El segon, recull diferents fets previsibles que no permetran desenvolupar el projecte en certes dates.

RISC	PLA DE CONTINGÈNCIA
Excés de feina en altres assignatures	S'intentarà realitzar una planificació acurada. Tanmateix, en cas de requerir més temps del previst es pot demanar vacances de la feina.
Imprevistos laborals	Intentaria negociar la possibilitat de endarrerir algun viatge inesperat o preveure les dates finals d'entrega de projectes.
Falta de coneixement en l'ús del programari	Buscar temps, fins i tot demanant vacances de la feina per tal de poder formar-me adequadament.
INCIDÈNCIA	PLA DE CONTINGÈNCIA
Viatges de feina	No és possible incloure les dates del 22-28 de març a la planificació degut a un viatge de feina.

El projecte ha transcorregut de forma pausada i sense masses complicacions. Tanmateix, ha calgut realitzar un parell d'ajustos en la planificació.

Inici-20 abril (Entrega PAC2)

La planificació del TFC ha funcionat de manera idèntica a la prevista fins a la tasca 4.1 (Taula 1). Un viatge de inesperat i urgent de feina ha fet que no es pogués dedicar el temps previst a l'esmentada tasca. Ha estat necessari reprogramar aquesta part en la setmana següent. La planificació global del projecte no va patir canvis.

20 abril (Entrega PAC2)-18 maig (Entrega PAC3)

Durant aquesta etapa a grans trets es van poder assolir quasi tots els punts compromesos fins al punt 6.1. Tanmateix, fou necessari acabar de detallar el punt 5.2 on calgué resoldre un aspecte concret del codi del programa C d'implementació del codi. En concret, fou necessari ajustar el ritme del mostreig de senyals. Aquest problema, va requerir el canvi de microcontrolador, i per tant, reescriure el codi escrit en la part 5.2, i evaluar-ne altre cop els resultats en la part 5.3. Aquest fet ha forçat solapar la tasca 5.3 amb la tasca 6.1 durant uns 13 dies.

2 Determinació de les especificacions del filtre

Per tal de realitzar el disseny del filtre cal seguir diferents passos. En primer lloc, és necessari analitzar el senyal per tal de poder determinar les especificacions del filtre. Bàsicament, cal dur a terme l'anàlisi freqüencial per tal de poder determinar la freqüència de tall i la tolerància del filtre.

A continuació, es procedeix a estudiar les diferents opcions de disseny. Tant l'analògica com la digital, i es discuteix les avantatges i desavantatges de cadascuna. Finalment, es procedeix amb un disseny i se'n comproven els resultats. Una vegada s'ha comprovat que el disseny funciona bé, es realitza la implementació. En aquest apartat es descriurà la metodologia i els resultats de l'anàlisi de el senyal.

2.1 Estudi de el senyal

Per tal de dur a terme la determinació de les especificacions del filtre en primer lloc es procedeix a estudiar el senyal problema mitjançant la transformada discreta de Fourier (DFT, en anglès) (1). Qualsevol senyal discreta periòdica permet una descomposició en sèrie de Fourier o representació freqüencial. Aquesta és una descomposició espectral, on es representa com un sumatori de diferents termes, cadascun d'ells amb una freqüència determinada que representen cadascun dels harmònics de el senyal. En l'equació 1 es mostra la transformada DFT i la antitransformada DFT. Un senyal discret no periòdic té un espectre continu però la DFT en dóna un aproximació. Per això, és una eina habitual en el processament de senyal.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}kn} \quad k = 0, \dots, N - 1 \text{ Transformada DFT}$$

Equació 1: Transformada DFT

$$x_k = \frac{1}{N} \sum_{n=0}^{N-1} X_n e^{\frac{2\pi i}{N}kn} \quad n = 0, \dots, N - 1 \text{ Antitransformada DFT}$$

Equació 2: Antitransformada DFT

En el cas de la transformada s'acostuma a representar el valor absolut de X_k . Aquest valor s'interpreta com la magnitud de el senyal en una determinada freqüència.

Es fa ús del software Scilab per tal de representar l'espectre de el senyal problema (figura 5). Aquest

software realitza el càlcul mitjançant la implementació d'un algoritme FFT (*Fast Fourier Transform* (2)). Aquest algoritme utilitza les propietats de simetria i periodicitat per tal de descompondre recursivament la transformada a realitzar, en altres de més senzilles fins aconseguir transformades de 2 elements on el valor n (equació 1) pot prendre els valors de 0 i 1. D'aquesta manera s'aconsegueix simplificar el càlcul.

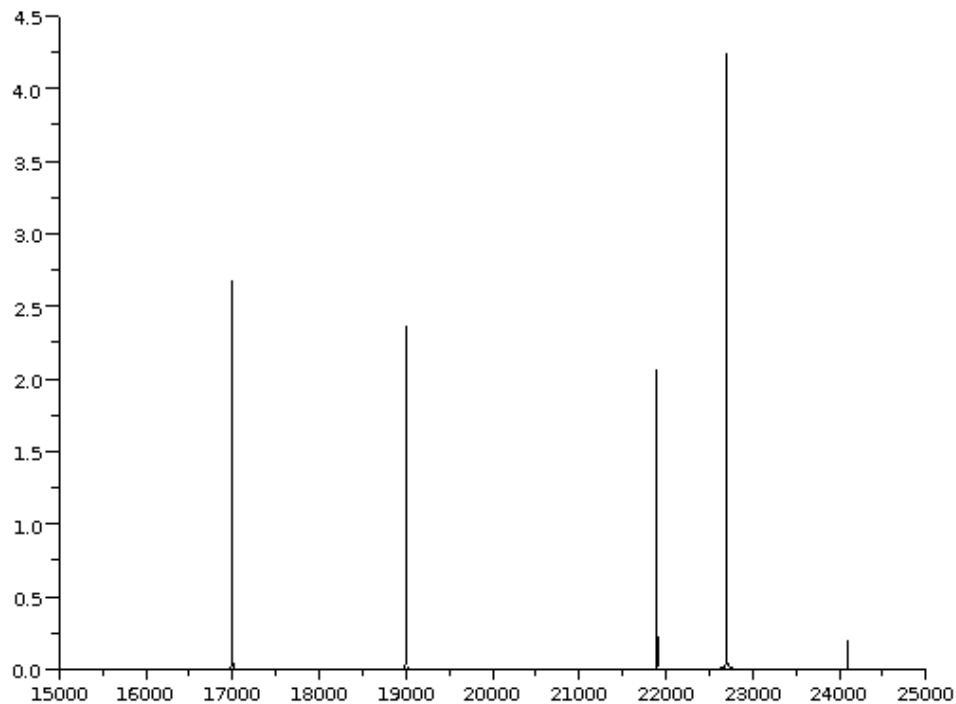


Figura 5: Senyal problema centrada en la zona d'interès (20,000 Hz). En l'eix d'absisses es representen freqüències i en l'eix d'ordenades magnitud.

La senyal problema oscil·la entre -20 i 20 unitats, i presenta una freqüència de mostreig de 220 kHz. Com s'ha comentat en l'apartat 1, que els components de freqüència superior a 20 kHz són considerats soroll i caldria eliminar-los. En la figura 5, es representa l'espectre de el senyal en la zona de 20 kHz i presenta components de freqüència superior a aquest valor. Per tant, seria desitjable construir un filtre passa-baix amb freqüència de tall al voltant de 20 kHz. Tanmateix, el disseny d'un filtre ideal d'aquestes característiques no és possible i cal deixar un marge de tolerància. Un filtre ideal seria un que té guany constant 1 i 0 a les bandes de pas i d'atenuació, respectivament, i fase zero o lineal. Aquest no és físicament realitzable. Desafortunadament, és impossible implementar aquest filtre ideal ja que a la pràctica, la resposta a l'impuls és infinitament llarga en el temps i el filtre no seria causal. En generar la sortida, en un moment donat t , caldria fer servir el senyal en $t+h$, és a dir en un futur. Cal recordar que es coneix com a sistema causal aquell que només depèn de les entrades presents o passades. Els sistemes realitzables no poden funcionar amb entrades futures.

En la realitat es treballa en filtres reals, els quals són aproximacions dels ideals. Per exemple, si es considera un filtre passabaix ideal, s'obté un guany de 1 per totes les freqüències per sota la de tall i

un 0 per les superiors. Només és possible treballar en aproximacions a aquesta resposta ideal. Aquesta aproximació a la funció ideal la proporcionen determinades funcions matemàtiques que tenen propietats que es corresponen amb les especificacions. Per això, en les especificacions es deixa una certa amplitud de tolerància.

2.2 Especificacions del filtre

Examinant la l'espectre mostrat en la figura 5, es pot comprovar que caldria dissenyar un filtre amb una freqüència de tall de 20.5 kHz. Per una banda, caldria situar a 19.5 kHz l'inici de la zona de transició ja que en aquest punt es situa el senyal útil més gran. Així la última freqüència de pas no està atenuada. Per l'altra, és necessari situar a 21.5 kHz el final de la zona de transició ja que en aquest valor es troba el component de menor freqüència de soroll (veure figura 5). D'aquesta manera es garanteix que la primera freqüència de soroll sigui adequadament atenuada. Es decideix permetre una tolerància del 5% tant en la banda de tall com en la de pas. La tolerància es defineix en la la banda de pas com l'atenuació de el senyal, i en la banda de tall com l'invers de l'atenuació. En la taula 2 es mostren les especificacions del filtre.

<u>Magnitud</u>	<u>Valor</u>
Freqüència de tall:	20.5 kHz
Zona de transició:	De 19.5 kHz a 21.5 kHz
Tolerància de la banda de pas:	5%
Tolerància de la banda de tall:	5%

Taula 2: Especificacions del filtre a dissenyar.

3 Disseny del filtre

Una vegada determinades les especificacions del filtre cal procedir a estudiar les diferents opcions de disseny. En primer lloc, s'estudia la possibilitat de l'ús d'algun filtre de tipus analògic. S'estudiarà l'ordre mínim necessari pels filtres de Butterworth, Chebyshev i Bessel. Posteriorment, en l'apartat 3.2 s'estudiarà l'alternativa digital.

Com s'ha comentat anteriorment, els filtres analògics consisteixen d'una combinació de resistències, bobines i condensadors. Aquests serien filtres passius, ja que només atenuen el senyal amb major o menor grau. Tanmateix, també poden incloure elements actius com amplificadors operacionals. Aquests últims s'anomenen actius, ja que poden incorporar guany en alguna part de el senyal. Per tal de dissenyar filtres amb unes determinades especificacions s'han proposat diferents mètodes: Butterworth, Chebyshev o Bessel. En canvi, els filtres digitals solen consistir d'un convertidor analògic a digital, un microprocessador, i un convertidor digital a analògic. El software que s'executa en el microprocessador implementa el filtre digital mitjançant les operacions necessàries. N'existeixen dos tipus, el IIR (*Infinite Impuls Response*) i els FIR (*Finite Impulse Response*). Es veuran en els apartats 3.1 i 3.2 les diferents opcions de disseny.

En alguns casos, els filtres digitals poden resultar lleugerament més cars que l'equivalent analògic degut a la seva major complexitat. En canvi, pot fer possible o més pràctic alguns dissenys que són impossibles o poc pràctics en versió analògica. A més, presenten alguns avantatges funcionals davant els filtres analògics. En primer lloc, els filtres digitals són molt més estables davant canvis en la temperatura i en el temps. No estan tan exposats a les no linealitats. Els filtres analògics consisteixen de components imperfectes que poden canviar les seves propietats amb la temperatura i el temps. Aquest problema es veu fortament augmentat quan l'ordre del filtre és més alt. En segon lloc, aquest tipus de filtres es poden dissenyar, provar i implementar en qualsevol ordinador. En tercer lloc, són programables. En qualsevol moment, pot implementar-se algun canvi sense afectar els circuits, ja que el seu comportament ve determinat per un programa emmagatzemat a la memòria. En quart lloc, els filtres digitals permeten tractar freqüències baixes més eficaçment. A més, gràcies a què cada vegada es desenvolupen processadors més ràpids també són competitius a freqüències altes, on anteriorment només els analògics podien treballar. En cinquè lloc, és possible crear filtres adaptatius. Finalment, els últims models de processadors DSP poden combinar combinacions complexes de filtres en paral·lel o en sèrie. D'aquesta manera, es simplifiquen els circuits respecte els que s'obtindrien mitjançant els seus equivalents analògics. Per això poden resultar més econòmics i amb una millor relació senyal soroll (3).

3.1 Disseny analògic

En el procés del disseny d'un filtre es segueixen diferents etapes (4). El primer pas, consisteix en normalitzar les especificacions. És a dir, transformar les freqüències de manera que la freqüència de tall sigui 1 rad/s. Seguidament, s'obté el filtre passabaixos normalitzat. En la figura 6 es mostra la representació en freqüència del filtre normalitzat que compleix amb les especificacions den la taula 2 normalitzades. Finalment, cal aplicar una transformació en les freqüències per tal de passar a les especificacions reals.

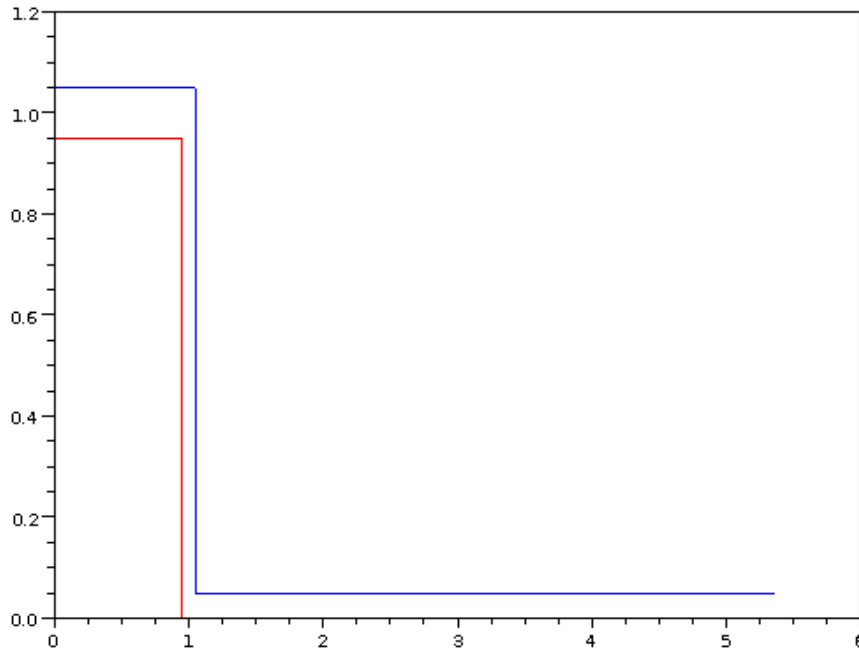


Figura 6: Representació del filtre normalitzat.

En el projecte actual, no cal convertir el filtre a passabaix. Tanmateix, com s'ha comentat en l'apartat anterior no és possible assolir un filtre ideal, bo i així, és possible acostar-s'hi mitjançant aproximacions a la funció ideal no realitzable. Per tal de dissenyar filtres amb unes determinades especificacions s'han proposat diferents mètodes:

- Filtres Butterworth (4)

El filtre de Butterworth és un dels filtres electrònics més bàsics. El seu disseny matemàtic està basat en produir la resposta més plana possible fins a la freqüència de tall. La seva funció de transferència es mostra en l'equació 3, on n és l'ordre del filtre, ω_c és la freqüència de tall, ω la freqüència, $H(j\omega)$ és la resposta en freqüència del filtre i G_0 és el guany a freqüència 0 que normalment és 1. Específicament, per filtres passabaix es pot fer un disseny sense normalitzar fent ús de l'equació 3.

$$G^2(\omega) = |H(j\omega)|^2 = \frac{G_0^2}{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}$$

Equació 3: Filtre de Butterworth

En la figura 7 es mostra la resposta en freqüència d'un filtre de Butterworth:

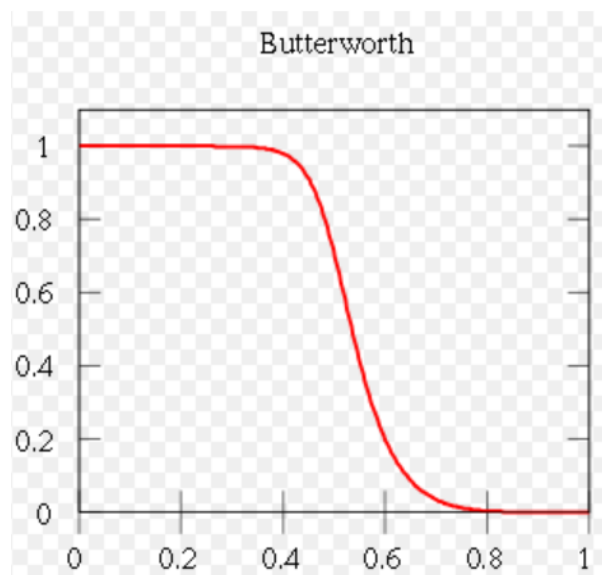


Figura 7: Resposta en freqüència d'un filtre de Butterworth

En la equació 4 s'estima l'ordre mínim (n) necessari en aquest tipus de filtre per tal d'assolir les especificacions de l'apartat anterior (taula 2). Per tal de poder realitzar aquesta estimació es manipula l'equació 4. Es substitueix $|G(\omega)|$ per $1/A$ (atenuació, sense unitats), ω/ω_0 per f , i s'aïlla n . El valor A val 0.05, el valor f 21.5/20.5 i n és el grau obtingut. En l'annex 1 es mostra el desenvolupament detallat.

$$n = \frac{\log_{10}(A^2 - 1)}{2\log_{10}f}$$

$$n = \frac{\log_{10}\left(\frac{1}{0.05}^2 - 1\right)}{2\log_{10}\left(\frac{21.5}{20.5}\right)}$$

$$n = 62.87$$

Equació 4: Deducció del grau necessari.

Com s'ha vist l'ordre és 63, massa elevat per dur a terme una implementació satisfactòria. Els filtres analògics passius consisteixen de elements imperfectes. Quan més gran és el grau, més gran és el nombre de components i l'acumulació dels seus errors. A més, tampoc resulta pràctic dur a terme filtres amb tants components, ja que el preu i la mida augmenta considerablement (4).

En la figura 8, es mostra que l'ordre 63 és el mínim que assoleix les especificacions. Per exemple, el d'ordre 62 no compleix amb les especificacions (vermell).

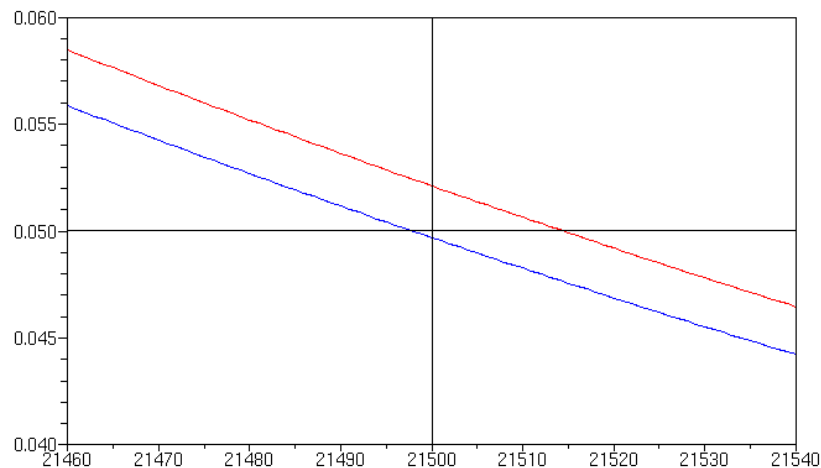


Figura 8: Representació del guany en funció de la freqüència prop del final de la zona de transició. Només el filtre d'ordre 63 (blau) permet un guany per sota de 0.05, el filtre d'ordre 62 (vermell) no permet complir amb les especificacions.

- Filtres Chebyshev (4)

Filtres electrònics relacionats amb els de Butterworth. Aquests es diferencien en què aconseguen una

caiguda més pronunciada en freqüències baixes degut a què permeten un arrissat (veure figura 9). N'hi ha de dos tipus: els de tipus I (equació 5), que només tenen pols, i els de tipus II, que presenten zeros i pols. Els dos tipus es diferencien en què, el primer presenta arrissat en la banda de pas i el segon presenta arrissat en la banda de tall (4).

$$G_n(\omega) = |H_n(j\omega)| = \frac{1}{\sqrt{1 + \varepsilon^2 T_n^2\left(\frac{\omega}{\omega_0}\right)}}$$

Equació 5: Filtre de Chebyshev tipus I

En la figura 9 es mostra la resposta en freqüència d'un filtre de Chebyshev:

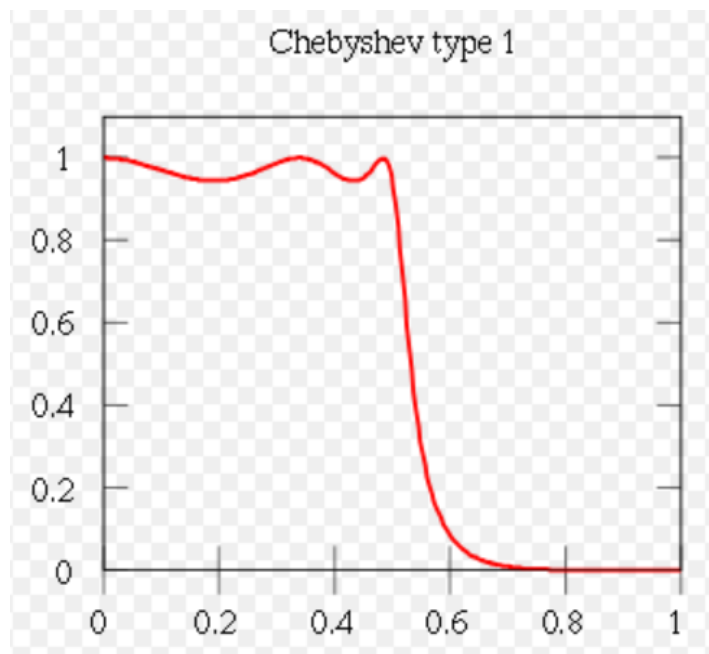


Figura 9: Resposta en freqüència d'un filtre de Chebyshev

Cal destacar que en les equacions de Chebyshev apareix el paràmetre ε . Aquest té a veure amb l'arrissat, que correspon amb les oscil·lacions que presenta el filtre Chebyshev tipus I en la banda de pas, i el de tipus II en la banda de tall.

Per altra banda, també apareix T_n que és el polinomi de Chebyshev d'ordre enèsim.

El filtre de Butterworth és caracteritzat per la seva senzillesa. En canvi, la seva resposta és menys sobtada que Chebyshev però no presenta arrissat. En les figures 9 i 10, es mostren la resposta de dos filtres d'ordre 5, un de tipus Chebyshev i l'altre de tipus Butterworth. Es pot observar molt clarament, la presència d'arrissament en el primer, i la menor pendent prop del punt d'inflexió del segon.

Com s'ha fet anteriorment per Butterworth, s'estima l'ordre mínim necessari per assolir les especificacions de la taula 2. (equació 6). Aquesta equació es dedueix de manera similar a la realitzada per Butterworth. Es substitueix $|G(\omega)|$ per $1/A$ (atenuació), ω/ω_0 per f , i s'aïlla n .

$$n = \frac{\log_{10}\left(\sqrt{\frac{A^2-1}{\epsilon^2}} + \sqrt{\frac{A^2-1}{\epsilon^2} - 1}\right)}{\log_{10}(f^2 + \sqrt{f^2 - 1})}$$

$$n = \frac{\log_{10}\left(\sqrt{\frac{1^2}{0.05^2} - 1} + \sqrt{\frac{1^2}{0.05^2} - 1}\right)}{\log_{10}\left(\frac{21.5^2}{20.5^2} + \sqrt{\frac{21.5^2}{20.5^2} - 1}\right)}$$

$$n = 19.25$$

Equació 6: Deducció de l'ordre mínim per Chebyshev

En aquest cas l'ordre mínim, és un xic inferior. Tanmateix, segueix sent massa alt per poder realitzar la implementació d'una manera eficient. Per altra banda, no s'ha intentat realitzar el disseny mitjançant un filtre de Bessel ja que hagués estat més difícil complir amb les especificacions (3).

Donat l'alt ordre requerit en les diferents alternatives analògiques, no sembla raonable fer ús d'un filtre analògic. Els filtres digitals són molt més versàtils per tal de processar senyals. Cal recordar que un filtre digital utilitza un processador per implementar una funció de transferència desitjada. A més, els filtres digitals presenten altres avantatges comentats a l'inici de l'apartat 3.

3.2 Disseny digital

El disseny de filtres digitals és un tema molt extens. Encara que els filtres es poden calcular de manera senzilla, els aspectes pràctics del seu disseny són un camp d'intensa recerca.

Hi ha dues grans categories de filtres digitals: els filtres recursius o IIR (*Infinite Impulse Response*) i els filtres no recursius o FIR (*Finite Impulse Response*). Els primers s'obtenen de discretitzar filtres analògics i els segons, són filtres directament discrets (3).

Els FIR no tenen memòria d'ells mateixos en instants anteriors. En canvi, els IIR sí en tenen ja que la sortida depèn d'ella mateixa en instants anteriors. Aquest comportament es pot veure en l'equació 8; la resposta és independent de la sortida en temps passats, en canvi en l'equació 7 la sortida depèn de la sortida en temps passats. És a dir, en el cas dels filtres FIR no es requereix realimentació mentre que pels IIR sí se'n requereix, ja que presenten memòria de el senyal de sortida. La realimentació pot incrementar la complexitat del càlcul.

$$y[n] = - \sum_{k=1}^N a_k y[n-k] + \sum_{k=0}^M b_k x[n-k]$$

Equació 7: Resposta a l'impuls unitari d'un filtre IIR

$$y[n] = \sum_{i=0}^N b_i x[n-i].$$

Equació 8: Resposta a l'impuls unitari d'un filtre FIR

Avantatges dels filtres FIR:

- Sempre són estables. Al no haver-hi recursivitat, no poden entrar en inestabilitat.
- Poden presentar una desfase perfectament lineal, el que implica que el filtre no introdueix distorsió de fase. Aquesta es produeix quan la resposta de fase no és lineal al llarg del marge de freqüències d'interès. D'aquesta manera el retard al llarg del filtre no és sempre el mateix. La falta de distorsió de fase és una avantatge crítica dels filtre tipus FIR

Avantatges dels filtres IIR:

- Poden dissenyar-se a partir de prototips analògics, transformant resultats. Per això és possible partir de especificacions i tècniques de disseny analògiques.
- Calen menys coeficients que un filtre FIR per les mateixes especificacions.
- La sensibilitat de la sortida per efecte d'arrodoniments i truncaments és menor en els filtres IIR.

En els filtres digital no és tan crític l'ordre que presenten. Donat que s'implementen mitjançant software, l'ordre es trasllada en augmentar lleugerament el càlcul matemàtic i l'emmagatzemament de paràmetres i dades. En canvi, com s'ha vist anteriorment un ordre elevat en filtres analògics comporta problemes en la seva construcció i funcionament.

Com a primera opció de disseny, es decideix fer un disseny IIR. Aquest tipus es aconsellable per dissenys on no es puguin preveure problemes d'estabilitat, per filtres d'ordre molt elevat, i en casos on es puguin aprofitar les especificacions basades en aproximacions analògiques. Donat que es té un disseny analògic, s'emprarà inicialment com a base del disseny IIR.

3.2.1 Disseny del filtre IIR.

Una manera de dissenyar filtres IIR és a partir d'aproximacions discretes de filtres analògics. Partint d'un filtre analògic estable i casual, el filtre obtingut mitjançant l'aproximació també hauria de ser-ho.

El mètode de la aproximació de la integral es porta a la correspondència entre els plans s i z mitjançant el que es coneix com la transformació bilineal (equació 9).

$$\begin{aligned} z &= e^{sT} \\ &= \frac{e^{sT/2}}{e^{-sT/2}} \\ &\approx \frac{1 + sT/2}{1 - sT/2} \end{aligned}$$

*Equació 9:
Aproximació
bilineal*

La transformació bilineal és la millor per convertir filtres analògics en digitals (3). En aquest cas, es partirà d'un filtre analògic de Chebyshev d'ordre 20 ja que compleix amb les especificacions (veure equació 5).

La resposta d'aquest filtre es mostra en la figura 10.

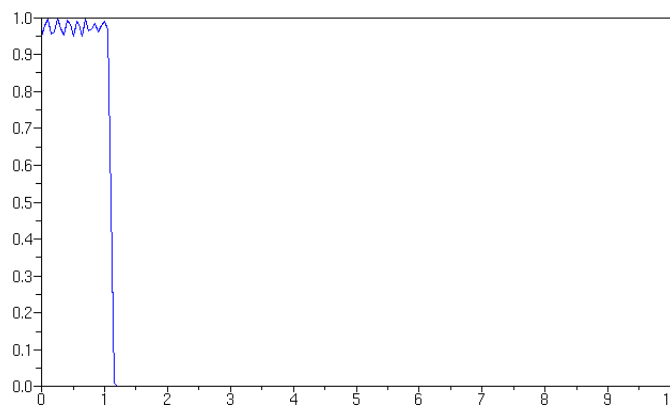


Figura 10: Resposta del filtre analògic de Chebyshev de tipus I i ordre 20.

A continuació es fa ús de l'aproximació bilineal per tal de deduir el filtre digital. La seva resposta es mostra en la figura 11. També s'afegeix en color verd el canvi de la fase.

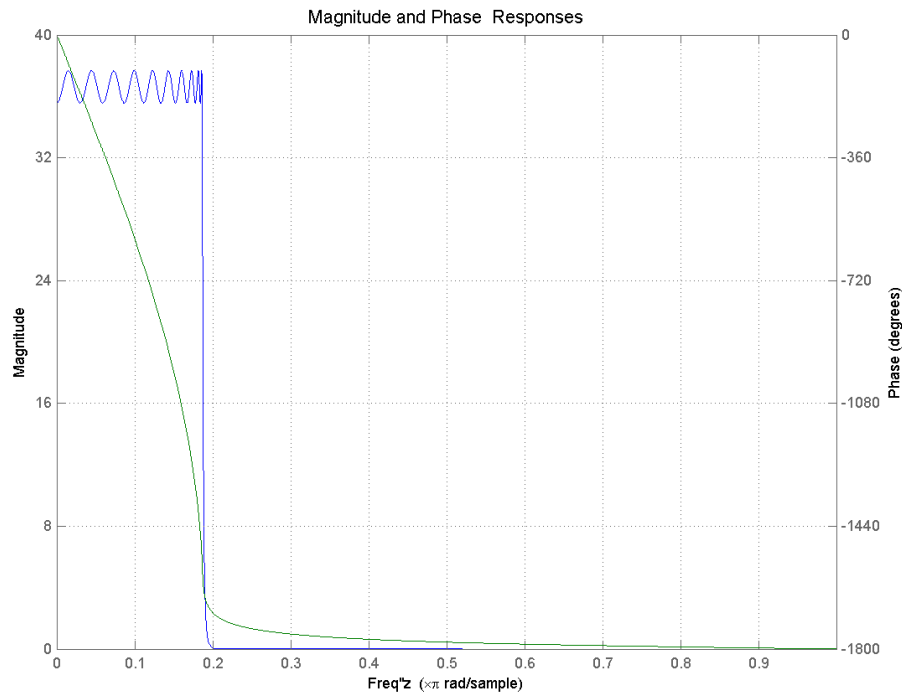


Figura 11: Magnitud de resposta del filtre digital dissenyat.

Per una banda, es compara l'atenuació diferencial de freqüències. L'atenuació compleix estrictament amb les especificacions inicials. En la banda de pas l'atenuació és menor al 5%, i en la banda de tall l'atenuació és per sobre 95%. Es pot comprovar visualment aquest fet comparant les figures 12 i 13, on es mostra l'espectre de el senyal d'entrada i del de la de sortida. En la figura 13, les freqüències superiors a 21500 Hz ha estat fortament atenuades.

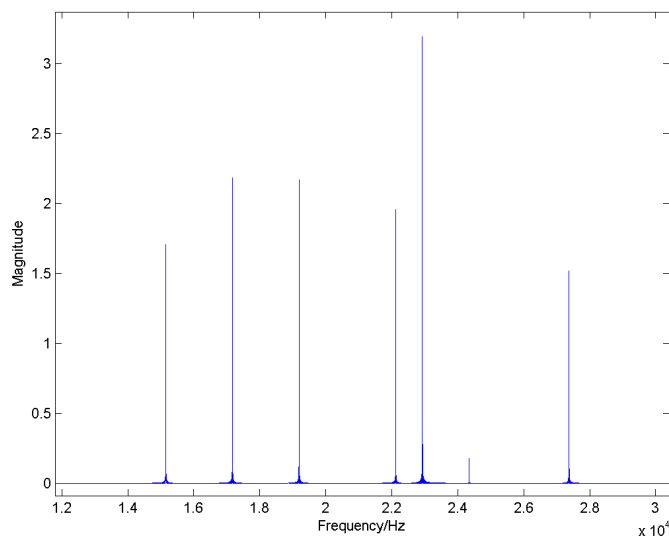


Figura 12: Espectre de el senyal sense filtrar

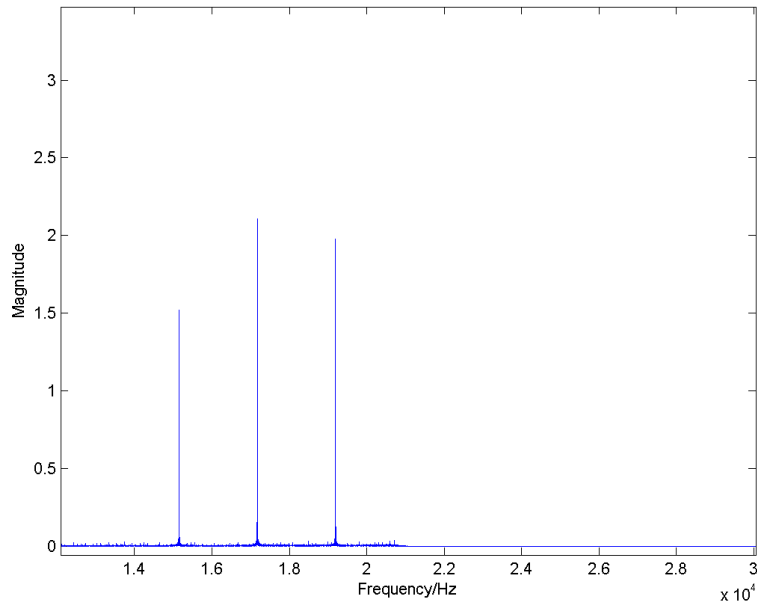


Figura 13: Espectre de el senyal filtrada

Des del punt de vista de les especificacions el filtre compleix estrictament. En canvi, malgrat aquest aspecte positiu el disseny no oferiria un bon rendiment, ja que apareix distorsió de fase en la banda de freqüència. Aquest aspecte es pot detectar en la figura 11 (línia verda), es detecta que la fase no és lineal en la banda de pas (freqüència normalitzada menor a 0.1864). Per això, cal realitzar un disseny FIR.

3.2.2 Disseny del filtre FIR

Com s'ha comentat anteriorment una altra possibilitat pel disseny de filtres digitals és el tipus FIR. Per tal de realitzar el disseny s'emprarà el mètode de la finestra lliscant (3). Aquest mètode es basa en obtenir la resposta l'impuls del filtre digital com la transformada inversa de la funció de transferència desitjada. Es trasllada les especificacions al domini temporal utilitzant la transformada inversa de Fourier (equació 10).

$$h_d[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\Omega}) \cdot e^{j\Omega n} d\Omega$$

Equació 10: Resposta a l'impuls desitjada

La resposta a l'impuls sol ser de duració infinita però truncant per la longitud escollida, es tindria el disseny acabat. No obstant, degut al truncament apareixen sobre impulsos i arrissats. Truncar la resposta a l'impuls equival a multiplicar-la per una finestra rectangular de longitud N. Aquests efectes es poden reduir escollint adequadament la finestra amb què s'efectua el truncament. Escollir aquesta finestra és un compromís entre l'arrissat afegit i el pendent del filtre. Un augment de la banda de transició es tradueix en una disminució de l'arrissat, i a l'inrevés.

Existeixen diferents tipus de finestres: rectangular, triangular, de Hamming, de Blackman o de Kaiser. Cada finestra té les seves avantatges i inconvenients, per tant, cal avaluar en cada cas quina és la més indicada. La finestra de tipus Kaiser (5), és programable i la seva forma es pot controlar a través del seu paràmetre β (equació 11). En aquest cas s'utilitza aquesta ja que permet utilitzar unes equacions per tal d'estimar l'ordre del filtre amb gran precisió i el paràmetre β que controla l'arrissat (equacions 11 i 12). Aquestes equacions permeten estimar aquests paràmetres utilitzant l'atenuació en dB de la banda de tall, l'ordre del filtre i l'amplada de la zona de transició. Mitjançant aquestes fórmules s'eviten les iteracions de prova i error.

$$\omega_n = \begin{cases} \frac{I_0(\beta \sqrt{1 - (2n/M - 1)})}{I_0\beta}, & 0 \leq n \leq M \\ 0, & \text{altrament} \end{cases}$$

Equació 11: Finestra de Kaiser, on I_0 és la funció de Bessel modificada d'ordre 0

En aquest cas s'utilitzarà un filtre d'ordre 246, amb una finestra de tipus Kaiser i $\beta=3.3953$. Per tal d'estimar l'ordre i el paràmetre β , s'empren les fórmules empíriques comentades anteriorment (Equacions 11 i 12). On α és l'atenuació en dB en la banda de tall, n l'ordre del filtre i $\Delta\omega$ l'amplada de la regió de transició (3).

La representació matemàtica d'aquest filtre d'ordre 246 consta de 247 paràmetres, i per estimar la sortida en cada punt caldran les 247 últimes mostres (recordar l'equació 8).

$$\beta = \begin{cases} 0.1102(\alpha - 8.7), & \alpha > 50 \\ 0.5842(\alpha - 21)^{0.4} + 0.07886(\alpha - 21), & 50 \geq \alpha \geq 21 \\ 0, & \alpha < 21 \end{cases}$$

Equació 12: Fórmules empíriques per tal d'estimar el paràmetre β

$$n = \frac{\alpha - 7.95}{2.285(\Delta\omega)}$$

*Equació 13:
Estimació de
l'ordre del
filtre*

Es realitza la implementació del filtre utilitzant les especificacions acabades de deduir i s'obté la resposta en fase i en freqüència que es mostra en la figura 14. Tant la resposta en magnitud com en fase semblen satisfactòries. La fase és lineal en la banda de pas, i s'aconsegueix un filtre amb alt pendent i poc arrissat. En l'apartat 4 s'analitzaran en detall els resultats.

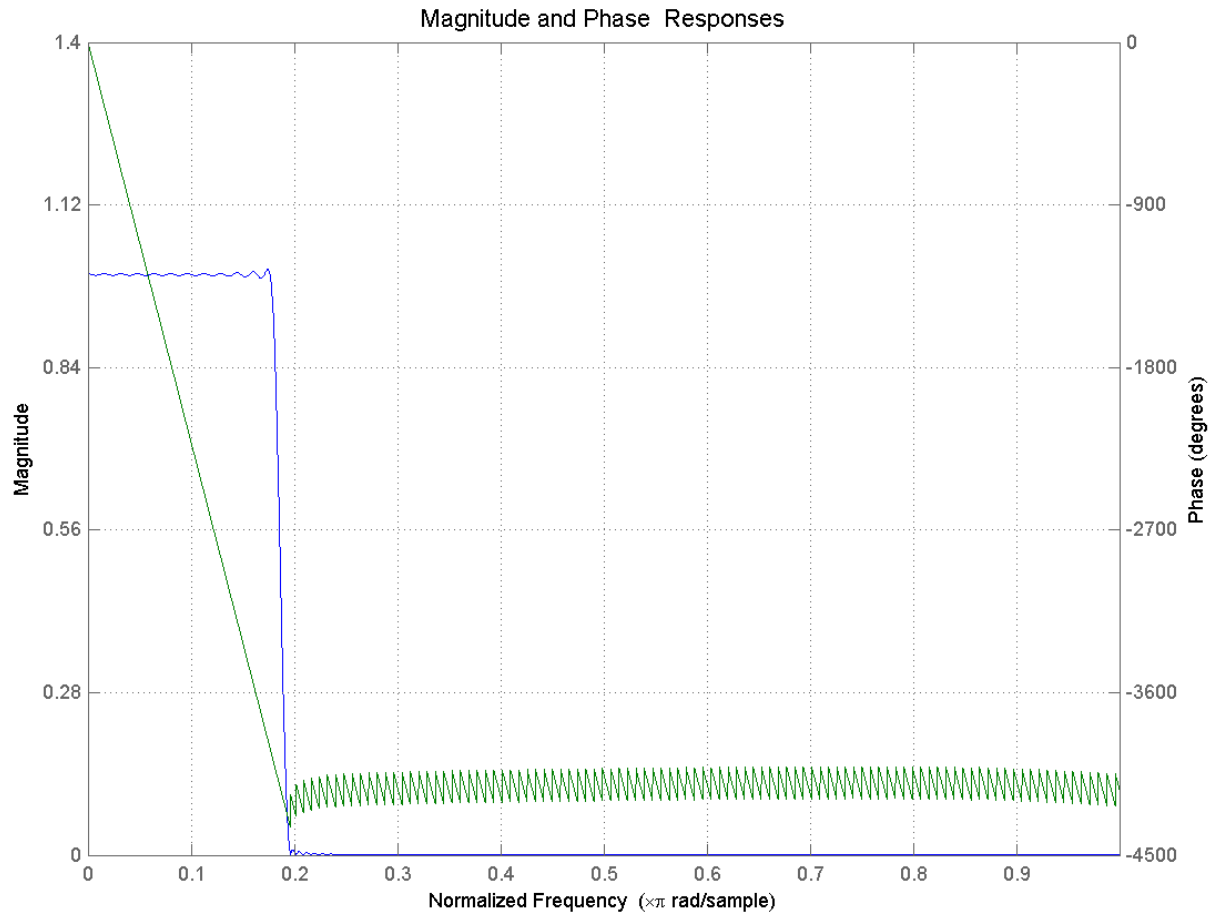


Figura 14: Resposta en magnitud i fase del filtre tipus FIR dissenyat.

4 Comprovació dels resultats

En primer lloc, es procedeix a comparar l'espectre de el senyal d'entrada amb el de la de sortida. Si es compara la figura 12 amb la figura 15, es pot comprovar que el filtre compleix amb les especificacions. Per una banda, es compara l'atenuació diferencial de freqüències. L'atenuació, una vegada més compleix estrictament amb les especificacions inicials. En la banda de pas l'atenuació és menor al 5%, i en la banda de tall l'atenuació és per sobre 95% (figura 15).

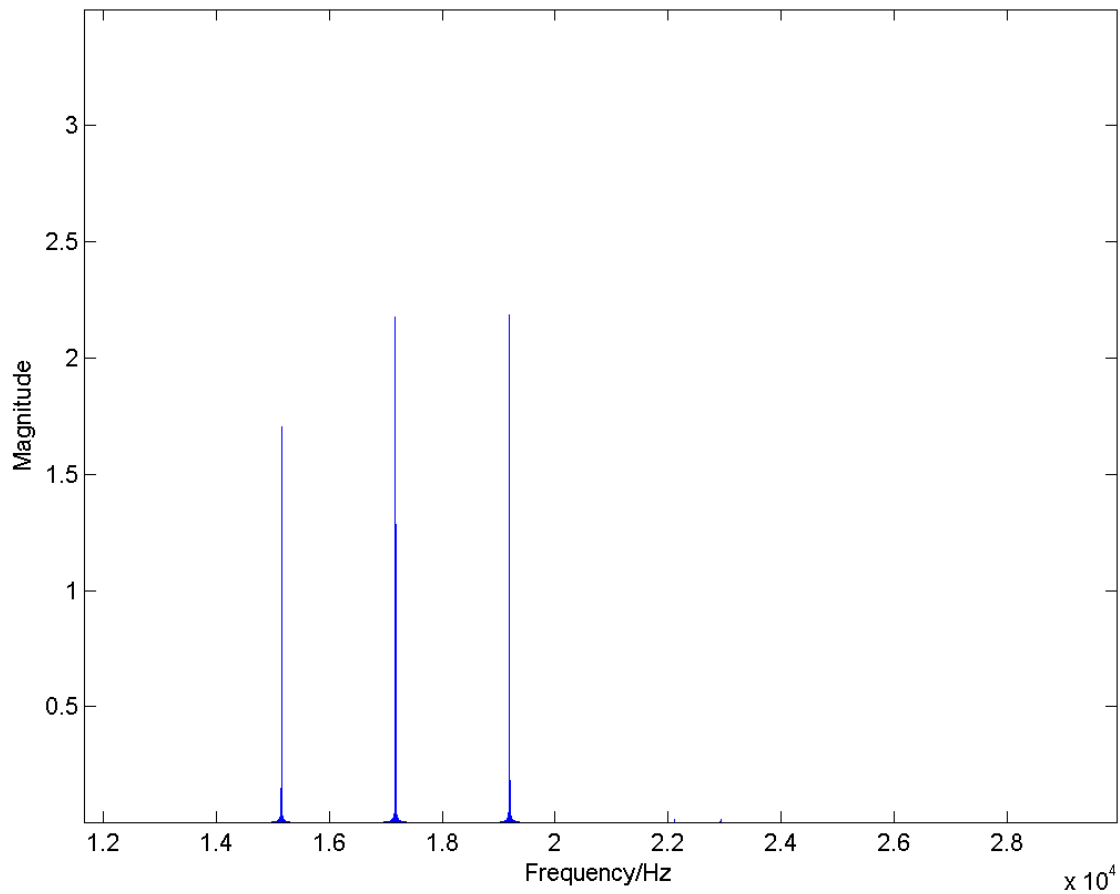


Figura 15: Espectre del senyal de sortida

A més, aquest filtre presenta una resposta en fase estrictament lineal. Per tant, es procedirà a implementar aquest disseny FIR. En la figura 16 es pot comprovar el comportament totalment lineal en la banda de pas de la fase (freqüència normalitzada menor a 0.1864).

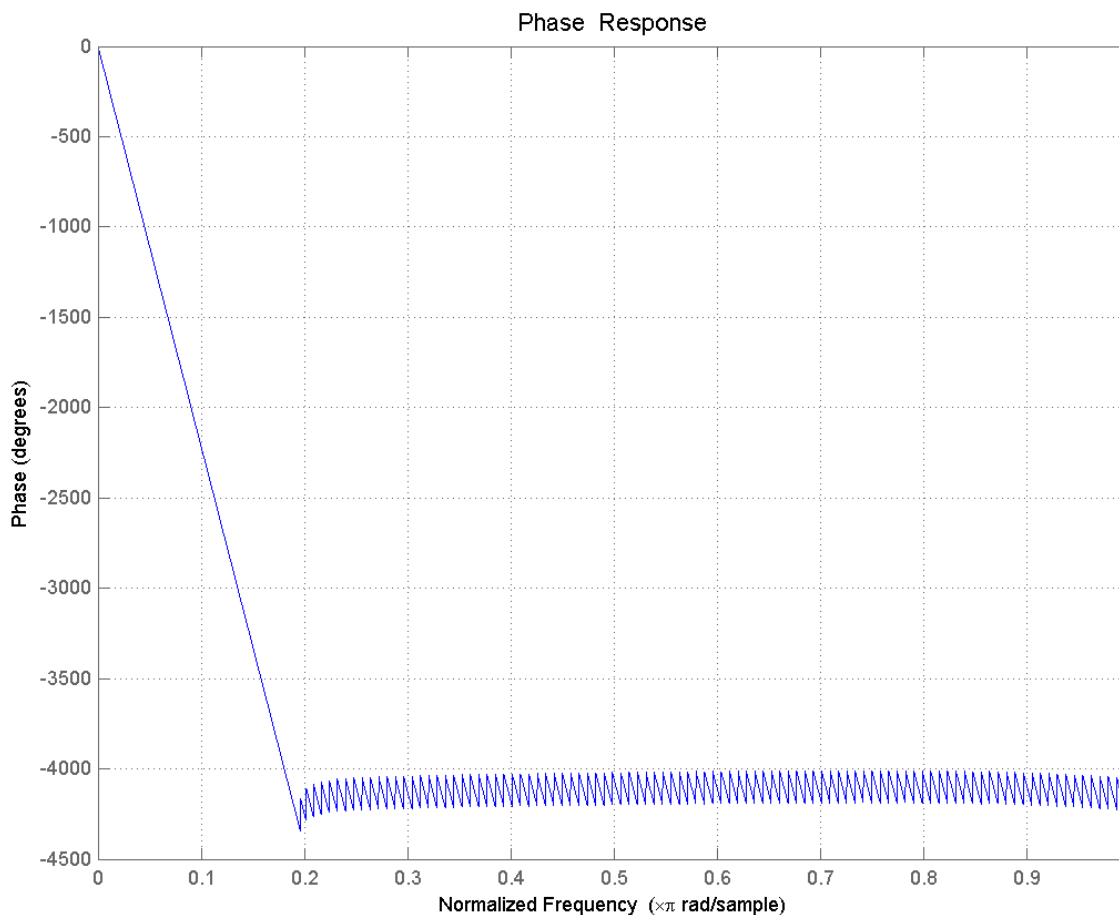


Figura 16: Variació de la fase en funció de la freqüència

Una vegada aconseguit el disseny d'un filtre que compleix amb les especificacions, ja només cal implementar-lo. Dels 5 objectius definits en la pàgina 8, ja se'n han assolit 3. S'ha analitzat l'espectre del senyal problema, s'ha estudiat les diferents possibilitats de disseny, s'ha realitzat i comprovat un disseny satisfactori.

Seguidament, es procedirà a implementar el filtre en un microcontrolador i es realitzaran simulacions per assegurar que funciona correctament.

5 Implementació del filtre

Donat que s'ha escollit un sistema digital per tal de dissenyar el filtre, cal realitzar la implementació mitjançant un microcontrolador que es programarà adequadament. Més concretament, s'ha escollit un microcontrolador tipus PIC (6) fabricat per l'empresa Microchip Technologies, el qual es programarà fent ús del llenguatge de programació C en l'entorn MPLAB IDE.

5.1 Introducció als microcontroladors

Un microcontrolador és un sistema destinat a gestionar el funcionament d'un procés. Aquests estan formats per diferents parts. A continuació es presenten les quatre parts principals:

- Un processador, que és l'element essencial que s'encarrega d'adreçar la memòria per llegir la instrucció que cal executar, descodificar-la i activar els senyals necessaris per a la seva execució. Es solen catalogar segons la complexitat de les instruccions que poden executar. Els tipus CISC (*Complex Instruction Set Computing*) permeten executar instruccions complexes que sovint requereixen més d'un cicle de rellotge, mentre que els RISC (*Reduced Instruction Set Computing*) permeten executar instruccions simples amb un únic cicle de rellotge.
- La memòria, que pot està dividida en dues parts, la que emmagatzema les dades i la que emmagatzema les instruccions. Si les instruccions i les dades es troben integrades en una única memòria es parla d'arquitectures Von Neumann, mentre que si es troben en dues memòries independents es parla d'arquitectures Harvard.
- Els ports d'entrada i sortida. Aquests permeten comunicar el microcontrolador amb la resta de dispositius dels sistema global.
- El rellotge, que marca el ritme de funcionament del microcontrolador i sincronitza els diferents elements.

Adicionalment, els microcontroladors porten altres prestacions com són *timers*, *watchdogs*, convertidors A/D o D/A, moduladors d'amplada de polsos i busos de comunicacions amb diversos protocols.

5.2 Els microcontroladors PIC de Microchip

A principis dels 90 Microchip technology va començar a oferir una família de petits controladors anomenats PIC (*peripheral interface controller*). L'arquitectura fou dissenyada a partir d'un conjunt d'instruccions reduït. Els xips normalment corren fins a 40 Mhz i executen una instrucció cada cicle de màquina (que correspon a quatre cicles de rellotge), excepte els salts de programa que consumeixen només dos cicles. La seva principal característica és la simplicitat.

El processador és de tipus RISC i disposa d'un conjunt molt limitat d'instruccions, unes poques desenes que permeten optimitzar l'estructura. Arquitectura està fonamentada en bancs de registres, és a dir, tots els elements del sistema tals com ports d'entrada i sortida, temporitzadors, convertidors, etc..estan

mapats en registres físics. Presenten arquitectura tipus Harvard, que permet adreçar separatament les instruccions de les dades. També implementen tècniques de segmentació en l'execució de les instruccions, que permeten estar llegint una instrucció mentre se n'executa una altra. Aquests tipus de microcontroladors no només són atractius per la seva simplicitat, també estan disponibles amb una gran varietat de models amb múltiples recursos accessoris. A més, es disposa un elevadíssim nombre de programes de suport tal com compiladors en C, simuladors, i emuladors.

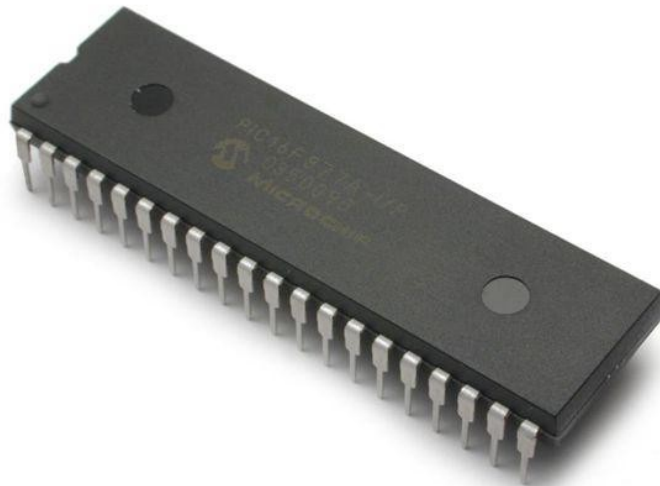


Figura 17: Exemple d'un PIC de Microchip Technology

5.3 Desenvolupament d'un sistema mitjançant microcontroladors PIC

Per tal d'implementar un sistema amb un microcontrolador cal fer un seguit de passos:

1. Donat que s'implementa una solució a baix nivell, cal conèixer el funcionament del PIC que s'utilitzarà. Cal fixar-se en les especificacions de la memòria de dades i de programa, registres de funcions especials, les instruccions s'executen a una velocitat més lenta que la freqüència de l'oscil·lador intern utilitzat, etc...
2. Una vegada analitzat el dispositiu, es pot entrar en el desenvolupament de l'aplicació. Primerament, cal escriure el codi que s'ha d'executar en qualsevol editor de textos. Tant es pot usar llenguatge de l'assemblador com llenguatges de més alt nivell com C. En la present pràctica s'emprarà aquest últim. A continuació es compilarà el codi i es gravarà a la memòria del microcontrolador. L'empresa Microchip facilita aquest programari per fer la conversió. A més, l'empresa també facilita a través d'Internet el programa de simulació MPLAB-SIM, que permet visualitza en l'ordinador com s'aniran executant les diferents instruccions i així poder-ne validar el funcionament. Finalment, es pot fer ús d'un programa gravador per tal d'introduir el programa a la memòria del PIC.

5.4 Elecció del dsPIC30F6010A de Microchip

Per tal de dur a terme el projecte cal un convertidor analògic a digital i un microcontrolador. S'ha optat per un microcontrolador amb un mòdul convertidor A/D per dues raons. Per una banda, per simplicitat. Així s'han pogut emprar les llibreries del compilador C30 de MPLAB. Aquest és un compilador optimitzat els microprocessador de la família PIC30. Per l'altra, d'aquesta manera s'assegura eficàcia i fiabilitat.

L'elecció del microcontrolador s'ha centrat en la família dsPIC30. Aquests presenten un excel·lent balanç entre preu i característiques. A més, com s'ha comentat, tenen un compilador propi, el C30 de MPLAB, i una àmplia documentació, ja que són productes àmpliament emprats.

Concretament s'ha escollit el dsPIC30F6010A. S'ha escollit aquest PIC per diferents raons. En primer lloc, té una freqüència de processador alta, 40Mhz, més de 200 vegades més alta que la de mostreig que és de 220kHz. En segon lloc, els canals de conversió analògic a digital són de 10 bits i amb una capacitat de mostreig de fins a 1 milió de mostres per segon. Per tant, permet representar 1024 valors diferents el marge de el senyal que oscil·la entre -20 i +20 unitats. Aquest número de bits permet assolir una precisió de 0.04 unitats. En tercer lloc, s'ha considerat la quantitat de memòria necessària per processar el senyal. Donat que el filtre és de ordre 246 i cada mostra ocupa 10 bits, cal tenir capacitat per emmagatzemar 2460 bits o 308 bytes de dades. A més, també cal emmagatzemar 247 paràmetres amb precisió float que ocupen 16 bits cadascun. Per tant, cal 494 bytes més. Només en paràmetres i dades calen 802 bytes.

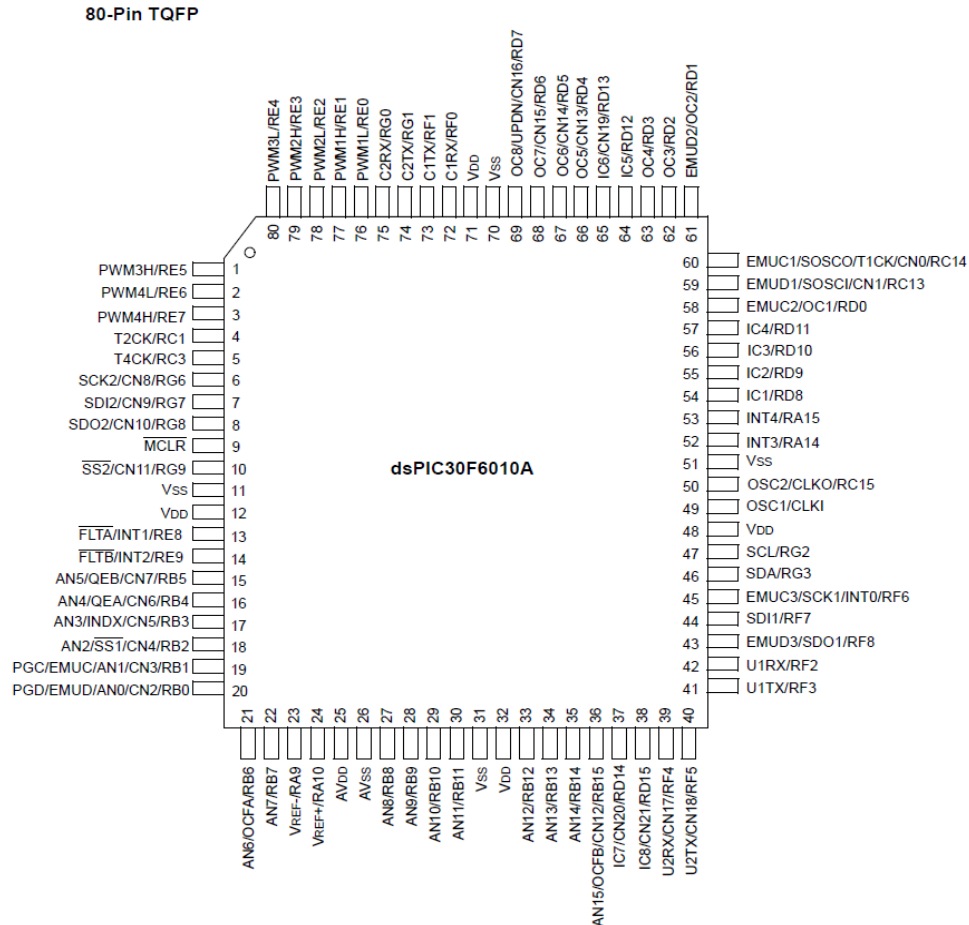
Abans de realitzar la implementació en aquest microcontrolador de la família dsPIC30, s'havia intentat la implementació en un de la família PIC18, concretament el PIC18F45J10. Aquest, microcontrolador que resultava significativament més econòmic (veure taula 3) permetia complir la majoria d'especificacions. Presenta suficient memòria, convertidor A/D de 10 bits i velocitat de processador alta. Tanmateix, la freqüència màxima de mostreig del convertidor A/D no permetia assolir els 220 kHz necessaris. Per això, durant les simulacions fou necessari descartar-lo.

Família	Model	Preu
PIC18	PIC18F45J10	1.51\$
dsPIC30F	dsPIC30F6010A	7.36\$

Taula 3: Diferents microcontroladors proposats

5.5 Implementació del filtre en un dsPIC30F6010A de Microchip

El dsPIC30F6010A pertany a la família dels dsPIC30. Aquest microprocessador presenta una arquitectura RISC de 16 bits optimitzada per programes escrits en C que pot treballar fins a 30 milions d'instruccions per segon (MIPS). La memòria de programa és de tipus flash i té capacitat per 144 kbytes, la memòria RAM és de 8192 bytes. A nivell d'entrada sortida, entre d'altres està equipat en 16 canals de conversió analògic-digital de 10 bits cadascun amb capacitat de mostreig de 1 milió de mostres per segon. Les especificacions exactes del pic escollit es troben en l'annex 1.



5.6

Figura 18: Pins disponibles en el dsPIC30F6010A de Microchip

5.7 Implementació del filtre en codi C

Com s'ha comentat en l'apartat anterior s'ha implementat el filtre FIR en un programa en C (7). El codi del programa es mostra en forma de diagrama de flux en la figura 19 i es troba detallat en l'annex 2. Per tal d'adquirir el senyal d'entrada, es fa ús del canal0 del convertidor analògic a digital. En canvi, per el senyal de sortida es fa ús dels pins RC0, RC1, RC2, RC3, RC4,RC5, RC6 i RC7 per codificar en 8 bits el senyal de sortida.

El programa fa ús de 3 llibreries. Per una banda, les generals *stdio.h* i *stdlib.h* per processament de text i entrades i sortides, respectivament. Per l'altra, es fa ús d'una llibreria específica per pics de la sèrie ds30.

Al començament del programa s'inclou també els paràmetres del filtre i la definició de les funcions, tant que actualitzen el valor de el senyal d'entrada després de cada mostreig, com de càlcul de el senyal de sortida en funció dels paràmetres i el senyal d'entrada.

A continuació, ja es defineix la funció *main*. Inicialment, configura els pins del port C i s'activa el convertidor analògic a digital. Seguidament, s'entra en una estructura *while* que realitza la mesura, la conversió d'unitats, el càlcul de el senyal de sortida i finalment es mostra pel port C el resultat, després de cada mostreig.

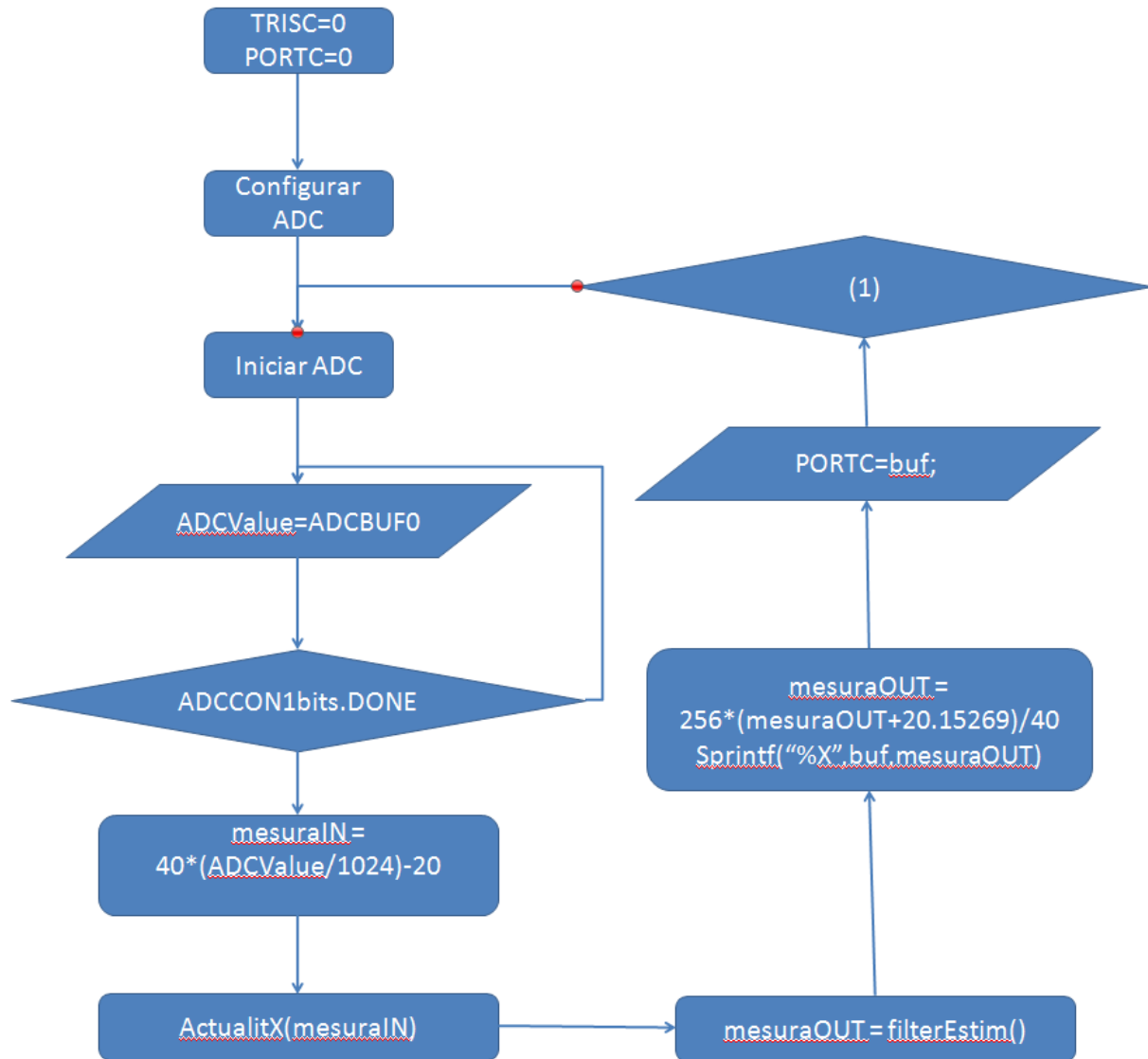


Figura 19: Diagrama de flux del programa.

En la figura 19 es veuen els cicles que descriu el programa, després de configurar la sortida i el mòdul ADC, s'entra en el cicle de mesura i processament. En cada cicle, s'inicia la lectura i s'espera que acabi. Posteriorment, es transforma a escala decimal i s'apliquen les rutines *ActualitX* i *filterEstim*. Aquestes actualitzen el vector que conté el senyal d'entrada, i calculen el senyal de sortida. Tant la rutina *ActualitX* com *filterEstim* es troben definides al fitxer principal *main.c*, però fora la rutina *main*. Finalment, el senyal es mostra a la sortida mitjançant el port C.

5.8 Comprovació dels resultats

Per tal de comprovar que el codi que s'ha escrit produeix els resultats esperats es procedeix a utilitzar l'eina MPLAB-SIM. Per una banda, es farà ús de l'eina Stimulus per tal de simular el mostreig de el senyal d'entrada. Per l'altra, s'utilitzarà el simulador UART1 per tal de recollir els valors el senyal de sortida en un fitxer. El programari Stimulus permet verificar els algorismes implementats en els microcontroladors. En concret permet injectar valors en el codi del programa escrit, o recollir-ne en un fitxer. Sobretot, s'utilitza per tal de simular senyals de hardware i permet dues modes de funcionament, síncrona o asíncrona. En aquest cas, interessa simular l'acció de mostreig. Per fer-ho, es configura Stimulus de manera que en cada operació de lectura per part del mòdul de conversió analògic a digital, es llegeixi d'un fitxer de dades el valor del senyal simulat (figura 20). Aquest fitxer conté el senyal problema. Com es pot comprovar en la figura 20, Stimulus es configura indicant el fitxer que conté el senyal (Data Filename); al lloc de memòria o registre on cal guardar el valor llegit, en aquest cas el *buffer* de ADC; i, el fet que desencadena la lectura que coincideix amb l'operació de lectura.

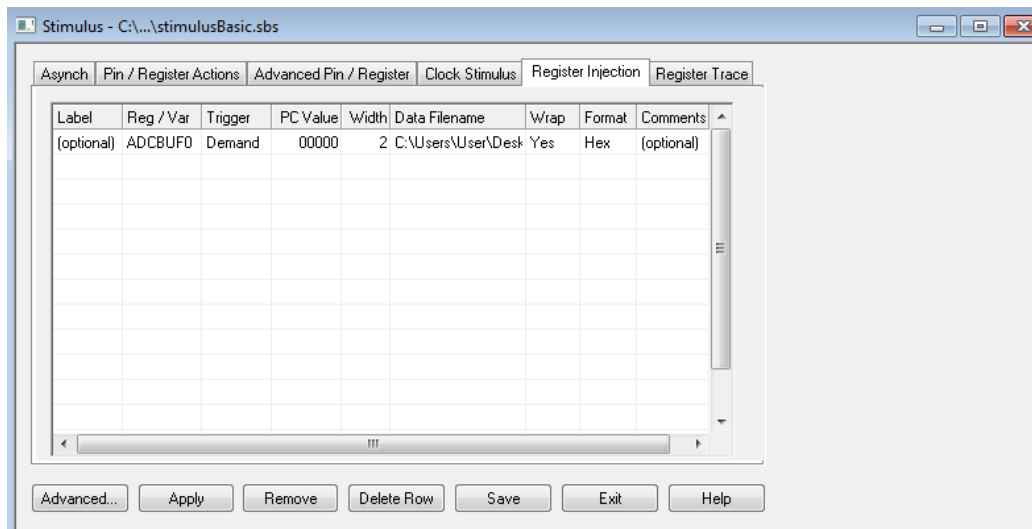


Figura 20: Configuració d'Stimulus per simular el mòdul ADC

Per tal d'avaluar la sortida es fa servir el simulador UART1. D'aquesta manera, el text de la comanda *printf()* es dirigeix a un fitxer de sortida (figura 21).

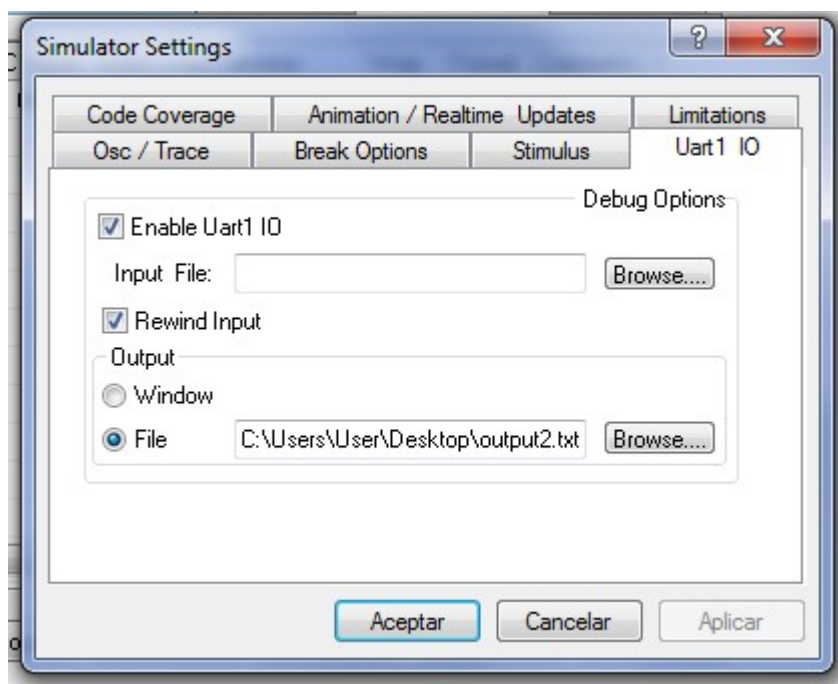


Figura 21: Configuració de la sortida mitjançant el simulador UART1

S'ha fet córrer el programa fent ús dels dos simuladors mencionats anteriorment. Així, s'ha pogut comprovar el correcte funcionament del programa, i l'error introduït degut a efectes numèrics. L'error relatiu mig introduït per el pas en el microcontrolador és del 0.11%. En la taula 4, es mostren els 10 últims valors generats pel disseny, i els simulats. Com es pot comprovar, l'error és sempre molt baix.

Disseny		Implementació		Error relatiu/%
INPUT	OUTPUT	INPUT	OUTPUT	
9.85	-5.08	9.85	-5.07	0.20
8.08	-5.53	8.09	-5.53	0.14
-1.70	-5.18	-1.72	-5.17	0.09
-1.38	-4.24	-1.36	-4.24	0.02
-4.21	-3.14	-4.22	-3.14	0.07
-14.67	-2.36	-14.68	-2.36	0.19
-16.46	-2.32	-16.48	-2.32	0.21
-16.52	-3.24	-16.52	-3.25	0.12
-16.18	-5.07	-16.17	-5.07	0.04
0.00	-7.44	0.00	-7.44	0.01

Tabla 4: Comparació dels resultats del disseny i de la implementació

Una vegada vist que la implementació dóna bons resultats, el pas següent seria gravar el programa a la memòria del microcontrolador fent ús del programari ofert per Microchip. Aquest microcontrolador ja

estaria a punt per dur a terme la tasca de filtratge requerida en el problema.

6 Valoració econòmica

Una vegada finalitzat el projecte és possible realitzar-ne un balanç econòmic. Per tal de dur a terme aquest projecte calgué emprar diferents recursos humans i materials.

Per una banda, fou necessari invertir unes 90 hores d'un enginyer en formació. Inicialment, es van planificar 78 hores però degut a la necessitat de repetir la implementació en un pic de gamma superior fou necessari afegir 12 hores més. Aquestes hores es corresponen a 11.25 jornades laborals. Si s'assumeix un salari d'uns 1600 € bruts per mes (20 jornades), corresponen 900 € associats als recursos humans.

Per l'altra, s'empraren diferents recursos materials. A nivell de software, es va fer servir una combinació de software lliure amb software propietari. Dins el software propietari es va fer ús de Matlab i Camtasia. La llicència d'estudiant de Matlab costa uns 89 \$, i uns 29 \$ per la toolbox de Filtres. La llicència de Camtasia costa uns 299 \$. A nivell de hardware s'ha utilitzat un portàtil IBM-ThinkPad T60, Intel Core 2 T7200, 4GB RAM de preu 1200 €. També ha calgut una connexió ADSL a Internet que costa 60 € i 3 € de despeses de papereria.

Si s'assumeix que les llicències s'han comprat exclusivament per aquest projecte i que s'assigna un 5% del valor del hardware i de la connexió ADSL, el cost total puja fins a 1,313.05 €.

	Import	Comentari
Recursos humans	900.00 €	11.25 jornades
Llicències software	74.17 €	Matlab
	24.17 €	Filtres toolbox
	249.17 €	Camtasia
Hardware	60.00 €	Ordinador (5%)
Internet	3.00 €	Connexió internet (5%)
Papereria	3.00 €	Bolígraf, esborranys
TOTAL	1,313.50 €	

Taula 5: Resum de la valoració econòmica

7 Conclusions

S'ha assolit el disseny i implementació d'un filtre que encaixa en les especificacions de la taula 2. Per una banda, s'ha emprat un disseny de tipus FIR seguint el mètode de la finestra lliscant. S'ha requerit un filtre d'ordre 246, amb una finestra de tipus Kaiser i $\beta=3.3953$. Anteriorment, s'havia realitzat un disseny IIR basat en un filtre de Chebyshev. Malgrat assolir les especificacions de la taula 2, no s'aconseguia una fase lineal en la banda de pas. Per l'altra, s'ha realitzat la implementació en un programa C instal·lat en un microcontrolador model dsPIC33F6010A de Microchip.

7.1 Resultats obtinguts

El resultat del treball ha estat molt satisfactori, s'han assolit tots els objectius marcats. A partir de l'estudi mitjançant la DFT de la senyal problema, es van deduir les especificacions. A continuació, s'han plantejat les diferents opcions possible i s'ha realitzat el disseny. Es va comprovar que els alts ordres requerits descartaven qualsevol disseny analògic. En primera instància, s'intentà utilitzar un dels dissenys analògics realitzats per dissenyar un filtre digital de tipus IIR. Malgrat que aquest filtre compleix estrictament les especificacions, presenta distorsió de fase en la banda de freqüència. Per tant, fou necessari dissenyar un filtre digital de tipus FIR. Aquest últim, a més de complir amb les especificacions presentava una fase totalment lineal.

Aquest últim disseny és un excel·lent candidat per dur a terme la implementació en un PIC. Per fer-ho es codificà en llenguatge C, la funció de transferència del filtre. En primer lloc, es va intentar en un pic de la família PIC18. No obstant, degut a la incapacitat de realitzar el mostreig a una freqüència suficientment alta calgué canviar-lo per un de la família dsPIC30 amb capacitat de mostreig de fins a 1 milió de mostres per segon. A més, s'ha pogut comprovar el bon comportament d'aquesta darrera implementació mitjançant l'eina MPLAB-SIM.

Per tal d'assegurar el bon funcionament del filtre han estat essencials les etapes de comprovació de resultats i simulació. Per una banda, la comprovació del disseny IIR va permetre descobrir que presenta distorsió de fase. Per l'altra, les simulacions mitjançant MPLAB-SIM obligaren descartar el pic de la família PIC18 ja que el mostreig no assolia suficient freqüència.

7.2 Treballs futurs

En el treball s'ha realitzat el disseny i implementació en un microprocessador d'un filtre digital de tipus FIR. A partir d'aquí es podria realitzar el muntatge físic d'aquest dispositiu. En primer lloc, es gravaria el programa a la memòria i es col·locaria en un suport físic adequat. Després de connectar-lo a una font d'alimentació adequada i en un circuit adequat, el filtre ja estaria llest pel seu ús.

8 Glossari

Amplificador operacional: Component electrònic actiu dels circuits elèctrics que presenta una entrada no inversora, una inversora i una sortida, a més de l'alimentació. La tensió de sortida és la diferència entre les entrades i multiplicada pel guany de llaç obert.

Bobina: Component electrònic passiu dels circuits elèctrics, que a causa del fenomen de l'autoinducció emmagatzema energia en forma de camp magnètic creat pel pas de corrent elèctric.

Convertidor analògic-digital: És un dispositiu electrònic que té la capacitat de convertir un senyal analògic en un altra de digital. És a dir es realitza la discretització d'una senyal contínua.

Compilador: Programa que tradueix el codi d'un llenguatge informàtic com pot ser C, a un altra llenguatge informàtic. Típicament, es tradueix des d'un llenguatge d'alt nivell a un de baix nivell normalment assemblador.

Condensadors: Component electrònic passiu dels circuits elèctrics que emmagatzema energia en el camp elèctric que estableix entre un parell de conductors els quals estan carregats però amb càrregues elèctriques oposades.

Diagrama de Gantt: Eina de planificació del treball que representa les activitats que s'han de realitzar i quan s'han de realitzar.

Filtre: Enginy que permet atenuar de manera selectiva determinades components freqüencials d'un determinat senyal.

Finestra de Kaiser: Finestra molt propera a la ideal utilitzada en el processament de senyals digitals.

FIR (*Finite Impulse Response*): Tipus de filtre digital en què la resposta impulsional de la sortida tindrà un nombre finit de termes no nuls.

IIR (*Infinite Impulse Response*): Tipus de filtre digital en què la resposta impulsional de la sortida tindrà un nombre infinit de termes no nuls, és a dir, mai torna al repòs.

Llenguatge assemblador: Llenguatge de programació de baix nivell. Implementa una representació simbòlica del codi binari i altres constants necessàries per programar en una arquitectura del processador concreta. Presenta una estreta relació amb l'arquitectura en què s'executa el programa.

Llenguatge C: Llenguatge de programació més utilitzat per desenvolupar sistemes operatius i altres tipus de programari bàsic. És un llenguatge de mitjà nivell que té moltes característiques de baix nivell.

Memòria flash: Memòria de tipus EEPROM que permet esborrar posicions de memòria amb una operació programable. Està implementada en xips que mantenen el seu contingut sense necessitat de tensió d'alimentació.

Microchip: Empresa de desenvolupa, fabrica i distribueix components electrònics. Especialment microprocessadors.

Microntrolador: microprocessador especialitzat a controlar equips electrònics.

MPLAB IDE: Entorn de desenvolupament i implementació de programes per pics. Integra editor de textos, compiladors, diferents eines de simulació, i programari per gravar els programes a la memòria dels pics.

Pic: Microprocessador fabricat per l'empresa Microchip.

Representació espectral: Descomposició d'un senyal en els seus components freqüencials.

Transformada discreta de Fourier (DFT, en anglès): És una operació que permet obtenir la representació d'un senyal en el domini freqüencial. El senyal resultant és continu i de variable real.

9 Bibliografía

1. Oppenheim. Signals and Systems. Prentice Hall; 1983.
2. Walker JS. Fast Fourier transforms. CRC Press; 1996.
3. Albertí EB. Procesado digital de señales: fundamentos para comunicaciones y control. Edicions UPC; 2005.
4. Paarmann LD. Design and analysis of analog filters. Springer; 2001.
5. Kaiser, J.F. Nonrecursive Digital Filter Design Using the I_0 -sinh Window Function. IEEE Symp. Circuits and Systems 1974;:20-23.
6. Huang H. PIC microcontroller: an introduction to software and hardware interfacing. Cengage Learning; 2005.
7. Barnett RH, O'Cull L, Cox S. Embedded C programming and the microchip PIC. Cengage Learning; 2004.

10 Annexos

10.1 Annex I: Filtres analògics (Butterworth)

Dedució de l'ordre mínim en un filtre de Butterworth:

$$G^2(\omega) = |H(j\omega)|^2 = \frac{G_0^2}{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}$$

Es substitueix $|G(\omega)|$ per $1/A$ (atenuació, sense unitats), ω/ω_0 per f , i s'aïlla n .

$$\frac{1}{A^2} = \frac{1}{1 + f^{2n}}$$

$$A^2 - 1 = f^{2n}$$

$$\log_{10}(A^2 - 1) = n2\log_{10}f$$

$$n = \frac{\log_{10}(A^2 - 1)}{2\log_{10}f}$$

$$n = \frac{\log_{10}\left(\frac{1}{0.05^2} - 1\right)}{2\log_{10}\left(\frac{21.5}{20.5}\right)}$$

$$n = 62.87$$

10.2 Annex II: Especificacions del microcontrolador

Especificacions recollides de la documentació específica de la documentació del fabricant (Microchip).

High-Performance Modified RISC CPU:

- Modified Harvard architecture
- C compiler optimized instruction set architecture with flexible Addressing modes
- 83 base instructions
- 24-bit wide instructions, 16-bit wide data path
- 144 Kbytes on-chip Flash program space (Instruction words)
- 8 Kbytes of on-chip data RAM
- 4 Kbytes of nonvolatile data EEPROM
- Up to 30 MIPS operation:
 - DC to 40 MHz external clock input
 - 4 MHz-10 MHz oscillator input with PLL active (4x, 8x, 16x)
 - 7.37 MHz internal RC with PLL active (4x, 8x, 16x)
- 44 interrupt sources:
 - 5 external interrupt sources
 - 8 user selectable priority levels for each interrupt source
 - 4 processor trap sources
- 16 x 16-bit working register array

DSP Engine Features:

- Dual data fetch
- Accumulator write-back for DSP operations
- Modulo and Bit-Reversed Addressing modes
- Two, 40-bit wide accumulators with optional saturation logic
- 17-bit x 17-bit single-cycle hardware fractional/integer multiplier
- All DSP instructions single cycle
- ± 16 -bit single-cycle shift

Peripheral Features:

- High-current sink/source I/O pins: 25 mA/25 mA
- Timer module with programmable prescaler:
 - Five 16-bit timers/counters; optionally pair 16-bit timers into 32-bit timer modules
- 16-bit Capture input functions
- 16-bit Compare/PWM output functions
- 3-wire SPI modules (supports 4 Frame modes)

- I2CTM module supports Multi-Master/Slave mode and 7-bit/10-bit addressing
- 2 UART modules with FIFO Buffers
- 2 CAN modules, 2.0B compliant (dsPIC306010A)
- 1 CAN module, 2.0B compliant (dsPIC306015)

Motor Control PWM Module Features:

- 8 PWM output channels:
 - Complementary or Independent Output modes
 - Edge and Center-Aligned modes
- 4 duty cycle generators
- Dedicated time base
- Programmable output polarity
- Dead-Time control for Complementary mode
- Manual output control
- Trigger for A/D conversions

Peripheral Features:

- High-current sink/source I/O pins: 25 mA/25 mA
- Timer module with programmable prescaler:
 - Five 16-bit timers/counters; optionally pair 16-bit timers into 32-bit timer modules
- 16-bit Capture input functions
- 16-bit Compare/PWM output functions
- 3-wire SPI modules (supports 4 Frame modes)
- I2CTM module supports Multi-Master/Slave mode and 7-bit/10-bit addressing
- 2 UART modules with FIFO Buffers
- 2 CAN modules, 2.0B compliant (dsPIC306010A)
- 1 CAN module, 2.0B compliant (dsPIC306015)

Motor Control PWM Module Features:

- 8 PWM output channels:
 - Complementary or Independent Output modes
 - Edge and Center-Aligned modes
- 4 duty cycle generators
- Dedicated time base
- Programmable output polarity
- Dead-Time control for Complementary mode
- Manual output control
- Trigger for A/D conversions

Quadrature Encoder Interface Module Features:

- Phase A, Phase B and Index Pulse input
- 16-bit up/down position counter
- Count direction status
- Position Measurement (x2 and x4) mode

- Programmable digital noise filters on inputs
- Alternate 16-bit Timer/Counter mode
- Interrupt on position counter rollover/underflow

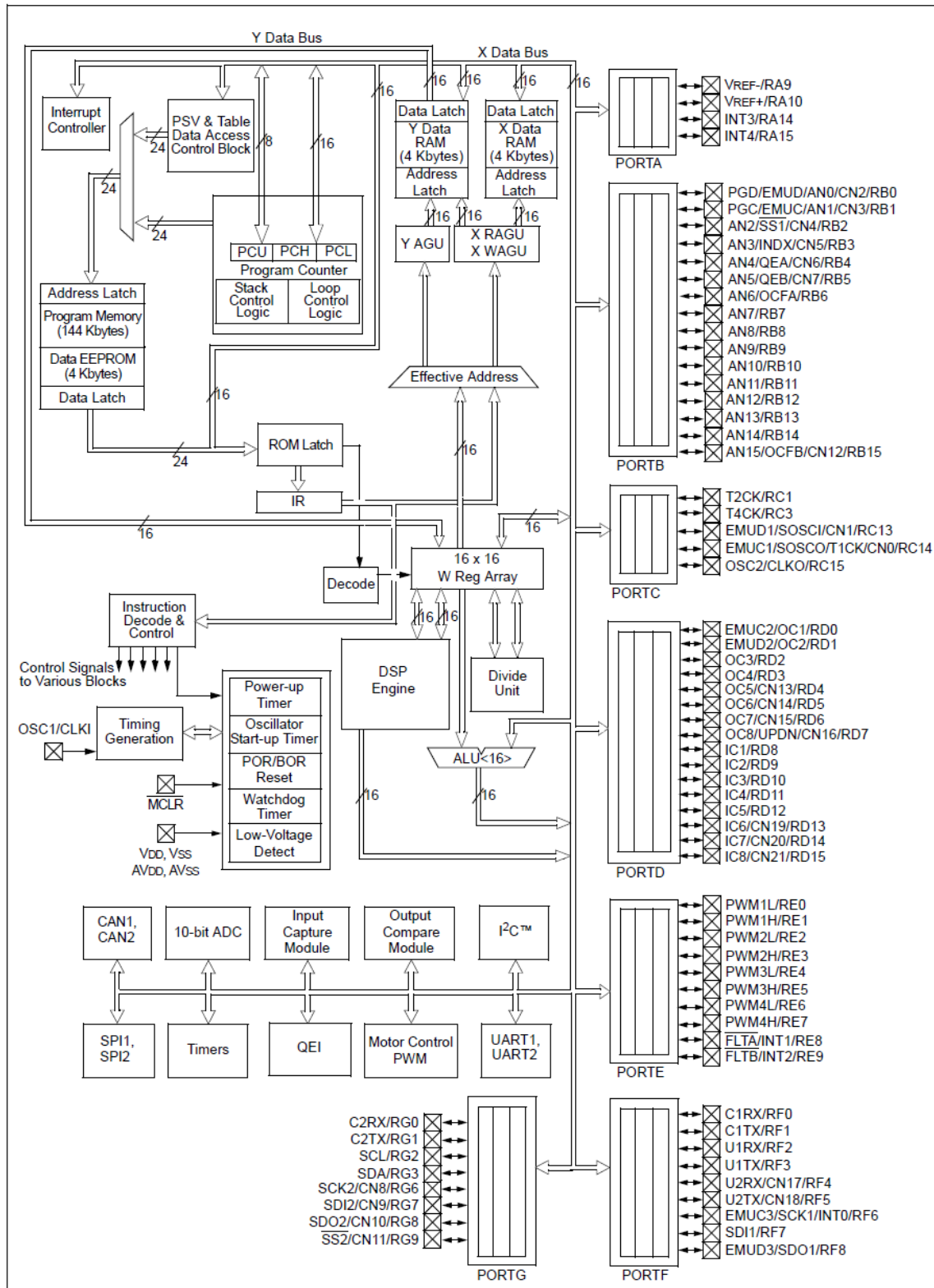
Analog Features:

- 10-bit Analog-to-Digital Converter (ADC) with 4 S/H Inputs:
 - 1 Msps conversion rate
 - 16 input channels
 - Conversion available during Sleep and Idle
- Programmable Brown-out Reset
- Phase A, Phase B and Index Pulse input
- 16-bit up/down position counter
- Count direction status
- Position Measurement (x2 and x4) mode
- Programmable digital noise filters on inputs
- Alternate 16-bit Timer/Counter mode
- Interrupt on position counter rollover/underflow

Analog Features:

- 10-bit Analog-to-Digital Converter (ADC) with 4 S/H Inputs:
 - 1 Msps conversion rate
 - 16 input channels
 - Conversion available during Sleep and Idle
- Programmable Brown-out Reset

Diagrama de blocs corresponent al dsPIC30F6010A de Microchip.



10.3 Annex III: Codi C de la implementació del chip

```
#include "p30fxxxx.h"

#include <stdio.h>

#include <stdlib.h>

float mesuralN, mesuraOUT;

float          params[247]={0.000092046461,0.000297853690,0.000421033670,0.000408350690,0.000249596040,-0.000014168377,-
0.000300065010,-0.000508899030,-0.000559054860,-0.000417372530,-
0.000115605350,0.000254793150,0.000569666920,0.000713243960,0.000619891770,0.000301720970,-0.000148996060,-
0.000585577850, -0.000854069290,

-0.000846822050,-0.000544092400,-0.000028007950,0.000538200110,0.000961082780,0.001082380700,0.000837059930,
    0.000283040790, -0.000409907900, -0.001011208400, -0.001305467200, -0.001168620700, -0.000617623460,
    0.000185535740, 0.000980070500, 0.001490220700, 0.001520015000, 0.001026707800,

0.000145900080, -0.000843608410,-0.001606992400,-0.001865724800,-0.001497685700,-0.000589534380,      0.000579877080,
    0.001623685200, 0.002173902000, 0.002009755900, 0.001143161500, -0.000170907080, -0.001507364800, -
0.002407196400, -0.002533693400, -0.001796025800, -0.000395516590,

0.001226011000, 0.002523895000,0.003032009800,0.002527971000,0.001124302800,-0.000750229880, -0.002479220100, -
0.003459422700, -0.003308972400, -0.002011867500, 0.000054708559, 0.002226542400, 0.003763452700, 0.004099001100,
    0.003045431400, 0.000880865300, -0.001718148500, -0.003884815400, -0.004848050500,

-0.004202887900,-0.002071783900,0.000905003440,0.003757034000,0.005495960600,0.005453293200, 0.003529742900,
    0.000265430610, -0.003304235300, -0.005971322600, -0.006757978800, -0.005266645000, -0.001855055200,
    0.002435118200, 0.006188007900, 0.008072243900, 0.007302669300, 0.003948311400, -0.001028777700,

-0.006036116900,-0.009347537900,-0.009682971000,-0.006681264500,-0.001098060400,0.005359446100, 0.010534001000,
    0.012514591000, 0.010309923000, 0.004271506100, -0.003896844500, -0.011583198000, -0.016059568000, -
0.015392765000, -0.009185909600, 0.001108725700, 0.012450875000, 0.021023318000, 0.023393035000,

0.017695153000,0.004475647500,-0.013099530000,-0.029795912000,-0.039534402000,-0.036995774000, -0.019182668000,
    0.013500638000, 0.056981984000, 0.104192310000, 0.146546420000, 0.175881030000, 0.186363640000,
    0.175881030000, 0.146546420000, 0.104192310000, 0.056981984000, 0.013500638000, -0.019182668000, -
0.036995774000, -0.039534402000, -0.029795912000, -0.013099530000,

0.004475647500,0.017695153000,0.023393035000,0.021023318000,0.012450875000,0.001108725700, -0.009185909600, -
0.015392765000, -0.016059568000, -0.011583198000, -0.003896844500, 0.004271506100, 0.010309923000, 0.012514591000,
    0.010534001000, 0.005359446100, -0.001098060400, -0.006681264500, -0.009682971000, -0.009347537900, -
0.006036116900, -0.001028777700,

0.003948311400,0.007302669300,0.008072243900,0.006188007900,0.002435118200, -0.001855055200, -0.005266645000, -
0.006757978800, -0.005971322600, -0.003304235300, 0.000265430610, 0.003529742900, 0.005453293200, 0.005495960600,
    0.003757034000, 0.000905003440, -0.002071783900, -0.004202887900, -0.004848050500, -0.003884815400, -
0.001718148500, 0.000880865300, 0.003045431400,
```



```

{
    float res=0;
    int i=0;
    for(i=0;i<247;i++)
    {
        res=res+xN[246-i]*params[i];
    }
    return res;
}

void actualitX(float nouPunt)
{
    int i=0;
    for(i=0;i<247;i++)
    {
        xN[i]=xN[i+1];
    }
    xN[246]=nouPunt;
}

int master,ADCValue;

int main (void)
{

    /*output pins configuration*/
    TRISC=0; /* control the direction I/O of the pins on the port*/

```

```

/*all 8 pins are output*/

PORTC=0;

/*iniciar el conversor*/

ADPCFG=0xFFFB; //all portB=digital;RB12 analog

ADCON1=0x00E0; //internal counter ends sampling and starts converting

ADCHS=0x0002; //connect RB12/AN12 as CH0 input

ADCSSL=0;

ADCON3bits.SAMC = 1;

ADCON3bits.ADCS = 22; //TAD 378ns @ 30 MIPS (0.5*TAD/Tcy-1)

ADCON2=0x0000;

ADCON1bits.ADON=1;

while (1)
{
//      char buf;

      ADCValue=0;

      ADCON1bits.SAMP=1;

      while(!ADCON1bits.DONE);

      ADCValue=ADCBUF0;

      mesuralN=ADCValue/0x0001;

      mesuralN = 40*(mesuralN/1024)-20.15269; // 1024 estados para 40 unidades

      printf("%f\t",mesuralN);

      actualitX(mesuralN); //se actualiza el valor de el senyal de entrada

      medidaOUT=filterEstim(); //se calcula el valor de el senyal de salida

```

```
printf("%f\n",mesuraOUT); //se muestra el resultado, asi con el MPLAB SIM sera facil comprobar el resultado
```

```
mesuraOUT=256*(mesuraOUT+20)/40; //se pasa a notacion de 8 bits
```

```
sprintf("%X",buf,mesuraOUT); //escribe con hexadecimal los valores de output del puerto C
```

```
PORTC=buf;
```

```
}
```

```
}
```