



Escola Politècnica Superior
d'Enginyeria de Vilanova i la Geltrú

UNIVERSITAT POLITÈCNICA DE CATALUNYA

PROJECTE FI DE CARRERA

**TÍTOL: Programació del microprocessador ColdFire MCF54455 de
Freescale per l'adquisició i tractament de dades sísmiques**

AUTOR: Normandino Carreras Pons

TITULACIÓ: Enginyeria en Automàtica i Electrònica Industrial

DIRECTOR: Antoni Mànuel Làzaro

DEPARTAMENT: 710, Enginyeria electrònica

DATA: 4 Juliol 2011

PROJECTE FI DE CARRERA

RESUM (màxim 50 línies)

El grup de recerca SARTI de la UPC està desenvolupant una sèrie d'equips de sísmica marina anomenats OBS (de l'anglès "Ocean Bottom Seismometer"), equips que han de ser robustos i fiables, a més, també han de tenir la capacitat de treballar en entorns hostils de fins a 6000 metres de profunditat i de forma totalment autònoma.

L'OBS s'ha d'enfonsar a l'oceà amb la finalitat que vagi a parar al fons marí, on dipositarà un geòfon. Aquest sensor és l'encarregat de transformar les vibracions sísmiques del fons marí, provocades pel moviment de les plaques tectòniques, en variacions de tensió. Alhora, l'equip porta incorporat un hidròfon per enregistrar les ones de pressió del agua de l'entorn marí.

El present projecte consisteix en la realització d'un sistema d'adquisició de dades sísmiques, que presenta un baix consum per tal d'aconseguir gran autonomia. Aquest sistema d'adquisició de dades és l'element principal de l'OBS, dissenyat a partir del microprocessador ColdFire MCF54455 de Freescale, el qual, s'encarregarà del control i la gestió de tot l'equip. Un cop s'obtenen les dades sísmiques, inicialment, aquestes passen per una placa que transforma el senyal analògic a digital i posteriorment es comprimeix i s'emmagatzema en el microprocessador.

En el codi desenvolupat en el projecte, hi trobarem la inicialització del microprocessador a partir de diferents registres accessibles, i mitjançant la programació de funcions bàsiques pels diferents perifèrics interns. Apareixen, també, funcions més específiques començant per la configuració de la placa analògic-digital, la lectura de les dades des de la placa d'adquisició, la compressió de les dades i finalment l'emmagatzematge de les dades a la targeta de memòria. Per acabar, cal tenir en compte les funcions corresponents al menú d'accés a l'usuari a nivell de terminal remot.

Paraules clau (màxim 10):

OBS	Adquisició	sismòmetre	Analògic-digital
microprocessador	ColdFire	C	PCB

ÍNDIX

1. INTRODUCCIÓ.....	2
1.1. <i>MÈTODES SÍSMICS.....</i>	<i>2</i>
1.2. <i>QUE ÉS UN OBS?.....</i>	<i>7</i>
1.3. <i>EL GEÒFON I L'HIDRÒFON.....</i>	<i>10</i>
1.4. <i>EL MICROPROCESSADOR MCF54455.....</i>	<i>11</i>
1.5. <i>OBJECTIUS.....</i>	<i>12</i>
2. DESENVOLUPAMENT DE L'ELECTRÒNICA DEL SISTEMA.....	13
2.1. <i>PLACA MICROPROCESSADOR.....</i>	<i>14</i>
2.2. <i>PLACA D'ALIMENTACIÓ</i>	<i>22</i>
2.3. <i>PLACA D'ADQUISICIÓ I CONVERSIÓ D'ANALÒGIC A DIGITAL</i>	<i>24</i>
2.4. <i>PLACA BACKPLANE</i>	<i>25</i>
3. PROGRAMA DE CONTROL	26
3.1. <i>FLUX DE L'APLICACIÓ</i>	<i>27</i>
3.2. <i>FUNCIONS BÀSIQUES I CONFIGURACIÓ.....</i>	<i>28</i>
3.3. <i>COMUNICACIÓ I CONFIGURACIÓ DE LA PLACA ANALÒGIC DIGITAL.....</i>	<i>31</i>
3.4. <i>COMPRESSIÓ DE LES DADES.....</i>	<i>36</i>
3.5. <i>EMMAGATZEMATGE DE LES DADES.....</i>	<i>38</i>
3.6. <i>FUNCIONS PEL SISTEMA DE FITXERS.....</i>	<i>42</i>
3.7. <i>MENÚ DE CONFIGURACIÓ.....</i>	<i>44</i>
4. MANUAL D'USUARI	49
5. PROVES I RESULTATS	51
5.1. <i>APARELLS UTILITZATS</i>	<i>51</i>
5.2. <i>PROVES REALITZADES</i>	<i>52</i>
5.3. <i>RESULTATS.....</i>	<i>61</i>
5.4. <i>CARACTERÍSTIQUES DE LES PLAQUES</i>	<i>65</i>
6. CONCLUSIONS.....	67
7. BIBLIOGRAFIA	68
8. ANNEX	69

1. INTRODUCCIÓ

El present projecte descriu el disseny i posta en marxa d'un sistema d'adquisició hardware i d'un software per l'adquisició i tractament de dades sísmiques. El sistema d'adquisició és l'element principal de l'equip d'un sismòmetre marí, anomenat en anglès OBS (Ocean Bottom Seismometer). El programa desenvolupat, està destinat al control de tot el sistema de l'OBS, i s'executa sobre un microprocessador ColdFire MCF54455 de Freescale. S'ha triat aquest microprocessador per les seves altes prestacions, especialment el baix consum, que ens permet un alt grau d'autonomia.

Les tasques principals proposades a realitzar en el microprocessador són: la inicialització del microprocessador a partir de diferents registres accessibles, i mitjançant la programació de funcions bàsiques pels diferents perifèrics interns, la programació de funcions més específiques començant per la configuració de la placa analògic-digital, la lectura de les dades des de la placa d'adquisició, la compressió de les dades i finalment l'emmagatzematge de les dades a la targeta de memòria, i finalment les funcions corresponents al menú d'accés a l'usuari a nivell de terminal remot.

1.1. MÈTODES SÍSMICS

Els mètodes sísmics han jugat històricament un paper clau en l'ampliació del coneixement de l'estructura i les propietats físiques dels diferents nivells que conformen l'interior de la terra. En línies generals, aquests mètodes permeten estimar la geometria dels nivells interns i el camp de velocitat de propagació de les ones sísmiques generades per fonts naturals (sismes) i/o artificials (explosions) en una zona determinada; a partir dels sismogrames registrats en una sèrie de sensors sísmics. Aquestes imatges estructurals han permès, a la vegada, comprendre millor els processos d'interacció entre el mantell i la litosfera, responsables en últim terme de la tectònica de plaques.

Alguns dels seus efectes més notables (activitat sísmica i vulcanològica, producció d'hidrocarburs, etc.) es produeixen principalment en els límits convergents de les plaques tectòniques (zones de subducció) que es localitzen majoritàriament en els límits entre oceans i continents. Per tant resulta indispensable investigar els marges actius mitjançant experiments combinats terra-mar per a comprendre l'origen i la naturalesa dels processos que els afecten.

La tomografia sísmica ha jugat un paper clau en l'ampliació dels nostres coneixements de la dinàmica de la Terra i de la seva estructuració interna. Possiblement sigui el millor mètode, a falta de la testificació directa, per a la determinació de paràmetres físics de les roques i materials constituents. Aquests paràmetres s'obtenen de forma indirecta a partir de les dades sísmiques i requereixen del coneixement dels temps recorreguts pel front de les pertorbacions acústiques (ones sísmiques) i de les seves trajectòries. Per això, és bàsic el registre d'aquests fronts d'ona amb diferents azimuts mitjançant sensors sísmics (geòfons). La resolució depèn de la cobertura i número de sensors susceptibles de registrar les pertorbacions i els seus temps d'arribada.

A terra, la resolució i cobertura és adequada en l'àmbit regional, i és relativament fàcil augmentar-la en casos específics, i mitjançant la disposició d'estacions sísmiques portàtils (p.e. per estudis de microsismicitat). No obstant, és important senyalar que la dinàmica de la litosfera, associada a la interacció de les plaques tectòniques passa en major part en les conques i marges oceànics (més del 99% de l'activitat tectònica - volcànica de la Terra prové dels marges continentals i zones d'acreció oceànica), tot i això la distribució d'estacions sísmiques marines està lluny de les equivalents a la terra, si bé sigui per la complexitat tecnològica que requereix el medi, ja sigui per les dificultats d'accés al registre de les mateixes o simplement per qüestions logístiques. Per això, avançar en aquests aspectes redundarà, sens dubte, en un major i millor coneixement dels processos que intervenen en la dinàmica de la litosfera terrestre.

Els instruments més utilitzats per a l'estudi dels processos sísmics submarins són els sismòmetres de fons oceànic (OBS), instruments autònoms que es mantenen fixes i immòbils durant l'experiment. Els sismòmetres marins es disposen en el fons marí, on registren sèrie temporals de dades (front d'ona acústica), realitzant-se la recuperació (del fons marí) dels mateixos mitjançant alliberadors acústics. A terra, els sensors equivalents (estacions sismològiques) van començar a utilitzar-se a principis del segle XX, però a nivell marí el seu desenvolupament i utilització és molt més recent. Els OBS començaren a desenvolupar-se a la dècada dels 70, i a partir de llavors la seva tecnologia no ha parat d'evolucionar, la seva utilització en l'àmbit de les Ciències de la Terra s'estén ràpidament i són cada vegada més nombrosos els grups de diversos països que intenten desenvolupar i perfeccionar els seus propis instruments.

De moment, tot i les activitats de recerca i desenvolupament realitzades, l'autonomia, la fiabilitat i el número d'OBS disponible per la comunitat científica internacional són encara inferiors a les de les estacions terrestres, però els progressos realitzats en aquests sentit són ràpids i evidents. Un dels objectius principals és el d'obtenir imatges estructurals d'alta resolució en els marges actius, per a poder localitzar amb precisió la sismicitat. Això permetrà determinar els mecanismes focals dels sismes, dotant de components dinàmiques a les imatges purament estructurals proporcionades per les dades sísmiques actives, i poder identificar les estructures tectòniques (falles actives, zona sismogènica inter-plaques) i comprendre els processos de deformació activa, generadores dels grans terratrèmols que ataquen periòdicament aquestes regions.

En l'àmbit nacional, excepte el grup SARTI (Sistemes de Adquisició Remota i Tractament de la Informació) de la UPC (Universitat Politècnica de Catalunya) amb col·laboració del CSIC (Consell Superior d'Investigacions Científiques) i la UPNa (Universitat Pública de Navarra), no existeix cap altre grup que estigui treballant en el desenvolupament d'un equipament similar, encara que aquest no és el cas en l'àmbit internacional. Així les grans institucions en el camp de la investigació marina, tant europees (IFREMER i Géosciences Azur a França, GEOMAR a Alemanya) com nord-americanes (Woods-Hole, SCRIPPS, Lamont-Doherty, Universitat de Texas) i japoneses (JAMSTEC), tenen equips de característiques similars.

En uns casos els sensors són hidròfons, i en altres són geòfons de 3 components, en la majoria dels casos de curt període, amb una freqüència natural de 4.5 Hz, i en alguns casos de banda ampla, fins a 240s en el cas dels SCRIPPS. Els dos grans centres d'investigació marina a EEUU, Woods-Hole Oceanographic Institution (Massachusetts), i SCRIPPS de San Diego, són pioners en el desenvolupament tecnològic de sismòmetres marins profunds. El projecte OBSIP de la NSF a Estats Units, per exemple, finança actualment la construcció d'un parc Nacional de més de 200 OBS de curt període (4.5Hz) i de banda ampla per a la utilització de la comunitat científica en experiments de monitoratge, estudi i vigilància dels processos sísmics i de l'estructura de l'escorça terrestre. En altres països com França o Japó existeixen iniciatives similars.

L'objectiu central d'aquest projecte és millorar la autonomia, dimensions, prestacions, etc., en una paraula el disseny de la electrònica d'adquisició dels sismòmetres marins, a partir dels resultats parcials aconseguits pels investigadors del grup SARTI en els projectes Sensual (2001-2003) i Sigsensual (2004-2007 Transmissió de Senyal en els Sensors Submarins Autònoms Lleugers CTM2004-04510-C03-00) finançats pel Ministeri de Ciència i Tecnologia.

La sísmica apareix inicialment com la ciència que estudia el comportament destructiu dels terratrèmols, i més tard, constata que l'anàlisi de les ones produïdes en els sismes conté una informació molt bona pel coneixement de l'estructura interna de la Terra, localitzant reserves minerals, aigua i hidrocarburs.

La sísmica seria una ciència diferent sense instruments. Els majors avanços realitzats en aquest àmbit van ser al voltant del any 1900 i des de llavors, gràcies a la innovació en la sensibilitat dels sismògrafs i altres equips, els terratrèmols podrien ser localitzats. A posteriori, la importància de les mesures de precisió del moviment real de la Terra es va convertir en evident per a l'estudi de l'atenuació de les ones sísmiques.

L'habilitat per a trobar la localització d'un terratrèmol i calcular la magnitud immediatament, ens porta a dos requeriments bàsics d'instrumentació: mesurar el temps en el precís instant i determinar la relació de dependència amb la freqüència entre la mesura i el moviment real de la Terra. Principalment es poden detectar amb la sísmica:

- a) Límits de capes
- b) Falles
- c) Porus reomplerts (com petroli)

L'exploració sísmica activa es basa en la generació, mitjançant dispositius mecànics, de fronts d'ona acústics en el rang de 5 Hz a 100Hz. Aquest front d'ones viatja a través del subsòl, trobant canvis en les diferents capes geològiques de la Terra, provocant ecos (reflexions) que viatgen per la superfície, on els sismòmetres que utilitzen transductors electromecànics (geòfons) detectaran i convertiran a senyals elèctriques, que convenientment amplificades i filtrades, seran tractades mitjançant programes de processat digital.

Per a realitzar l'exploració sísmica s'utilitzen dos tipus de tècniques: la reflexió i la refracció, basades en la detecció, registre i estudi de les ones que es generen en les interfases de les diferents capes, constituïdes per materials diferents, quan un ona generada artificialment arriba a aquest punt.

1.1.1. Sísmica activa

Depenent de la disposició relativa de les fonts i els sensors, els mètodes sísmics es classifiquen en mètodes de reflexió i de refracció. En la sísmica de reflexió, els sensors estan situats pròxims a les fonts ($10\text{-}10^3$ m) i es disposen generalment al llarg d'una línia (l'*streamer*, en el cas dels experiments marins), que es desplaça solidàriament amb les fonts (únicament artificials). Donada la seva geometria, en aquest tipus d'experiments es registren bàsicament les ones reflectides als diferents nivells de profunditat. Aquest mètode, àmpliament utilitzat a nivell científic i industrial (prospecció d'hidrocarburs), permet obtenir imatges estructurals d'alta resolució dels nivells menys profunds de la litosfera (escorça). El límit ve determinat principalment per la pèrdua d'energia de les ones reflectides i transmeses a cada interfase. D'altra banda, el temps de trajecte de les diferents reflexions depèn tant de la velocitat de propagació com de la profunditat dels reflectors, amb lo que resulta complicat determinar ambdós paràmetres per separat a partir únicament d'aquestes dades.

D'altra banda, en la sísmica de refracció els sensors es poden localitzar en un ampli rang de distàncies de les fonts ($\sim 10^3\text{-}10^6$ m), permetent registrar no únicament les ones reflectides sinó també les refractades en els diferents nivells. Aquestes fases contenen informació directa de la velocitat de propagació de les ones generades per les fonts (tant actives com passives) i, per tant, de les propietats físiques d'aquests nivells. Per tant, en aquest cas és més senzill determinar la seva velocitat sísmica i la profunditat a partir del temps de trajecte de les ones refractades. Un problema fonamental de la sísmica de refracció, és el *tradeoff* existent entre la resolució de les imatges obtingudes i la profunditat de l'objecte que es vulgui caracteritzar. Quan més profund sigui l'objecte, més s'han de separar les fonts i els receptors, i a la vegada menor és la resolució de la imatge obtinguda. La màxima profunditat de propagació de les fases ve determinada per la separació entre fonts i receptors. La geometria de cada experiment, per tant, s'estableix en funció de la profunditat i les dimensions de l'objecte que es vol caracteritzar (des de l'escorça fins el mantell profund).

Sísmica de Reflexió:

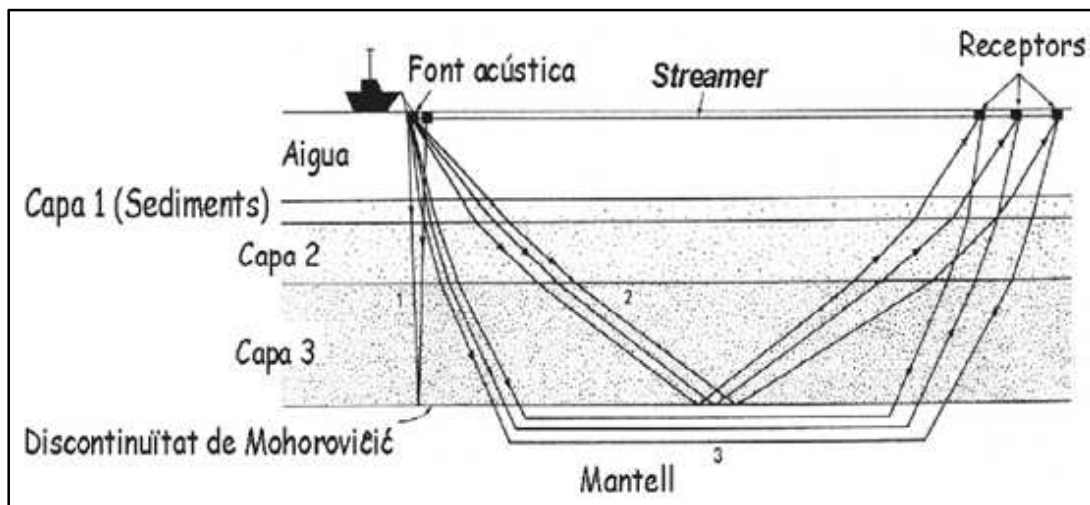


figura 1.1 Sistema de reflexió

La sísmica de reflexió és un mètode que utilitza l'emissió d'ones des d'un vaixell oceanogràfic, ones que incideixen en el fons marí des on descendiran cap a l'interior del subsòl, fins que topin amb un canvi de medi, on es generarà una ona reflectida. En figura 1.1 es mostra un diagrama del comportament del front d'ones acústiques.

Registrat amb múltiples hidròfons l'ona directa, l'ona reflectida, mesurant el temps entre l'arribada de les dues ones i la diferència d'energia entre elles, es poden determinar la profunditat de les diferents capes litològiques i les propietats de cadascuna d'aquestes.

Es genera l'ona directa a partir d'una font d'ones sísmiques proporcionada per canons d'aire comprimit situats a bord d'un vaixell oceanogràfic; l'ona directa es propaga en el medi superior amb la velocitat uniforme v_1 . L'ona reflectida es genera per la reflexió de l'ona directa incident en la interfase entre medi 1 i medi 2 i es propaga amb la velocitat v_1 . Una porció de l'ona incident en la interfase entre medi 1 i medi 2 passa per la interfase i es refracta. L'ona refractada es propaga en el segon medi amb la velocitat v_2 .

A través de les dades entregades per les reflexions sísmiques es poden construir l'horitzó de reflexió que correspon a un canvi de materials. Per exemple diferents estrats o falles tectòniques. Comentar per últim que la distància entre la font generadora d'ones acústiques i el geòfon és petita comparada amb la profunditat de la reflexió.

Sísmica de refracció:

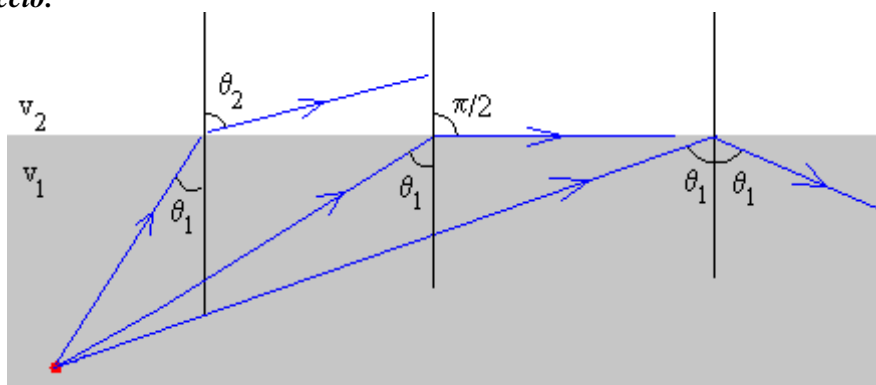


figura 1.2 Angle crític. Reflexió total

A la figura 1.2, observem que a mesura que s'incrementa l'angle d'incidència θ_1 , l'angle de refracció augmenta fins que es fa igual a $\pi/2$. Si es torna a incrementar l'angle d'incidència, l'ona incident es reflexa en el primer medi.

Si $v_1 > v_2$ l'angle $\theta_1 > \theta_2$ el raig refractat s'acosta a la normal
 Si $v_1 < v_2$ l'angle $\theta_1 < \theta_2$ el raig refractat s'allunya de la normal

En el segons cas, per un angle límit θ_c l'angle de refracció és $\theta_2 = \pi/2$

L'angle límit és aquell angle incident per el que el raig refractat emergeix tangent a la superfície de separació entre els dos medis.

Si l'angle d'incidència és major que l'angle límit, el sinus de l'angle de refracció resulta major que la unitat. Això indica, que les ones que incideixen amb un angle major que el límit no passen al segon medi, sinó que son reflectides totalment a la superfície de separació.

La sísmica de refracció es basa en la generació d'ones sísmiques mitjançant canons d'aire comprimit ubicats a bord d'un vaixell oceanogràfic; fent-les incidir sobre una interfície entre capes litològiques amb un angle crític. L'ona refractada que es generarà, es desplaçarà per la capa litològica inferior, paral·lelament a la frontera entre capes, fins que tornarà a la capa superior on serà registrada. A la figura 1.3 podem veure un diagrama del mètode de sísmica de refracció.

Registrant aquesta ona refractada amb múltiples geòfons, juntament amb el temps i la distància que ha estat viatjant, podem aproximar la velocitat d'una ona en una capa litològica concreta i d'aquesta forma determinar les propietats geològiques d'aquesta capa. La discontinuïtat de Mohorovic (Moho) divideix la superfície i el mantell terrestres i dins de les conques oceàniques aquesta discontinuïtat es troba localitzada a uns 6 o 7 km i en els continents a uns 40 km. La velocitat de propagació s'incrementa amb l'edat geològica i la profunditat.

ESTRAT	VELOCITATS	GRUIX
CAPA 1	SEDIMENTS	<500 m.
CAPA 2	$5,07 \pm 0,63 \text{ km/s}$	$1,71 \pm 0,75 \text{ km.}$
CAPA 3	$6,69 \pm 0,26 \text{ km/s}$	$4,86 \pm 1,42 \text{ km}$
MANTELL	$8,13 \pm 0,24 \text{ km/s}$	-

taula 1.1 Relació entre les velocitats de propagació i els gruixos

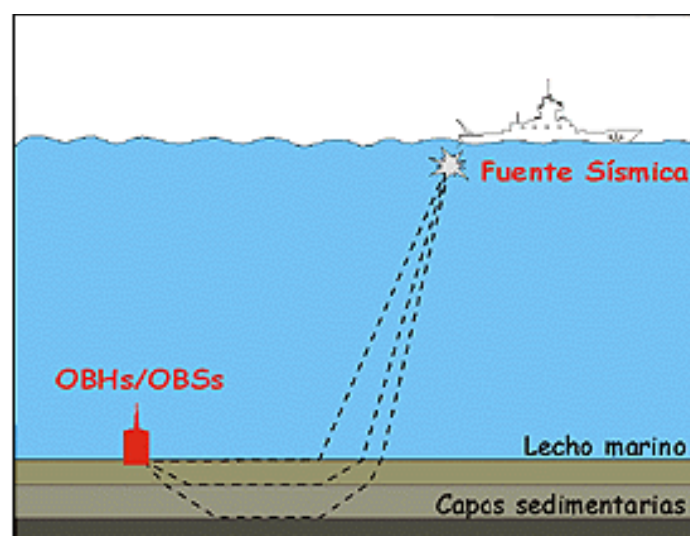


figura 1.3 Procediment esquemàtic de la sísmica de refracció

En la generació artificial de ones sísmiques s'utilitza energia de baixa freqüència entre 1 i 25 Hz. Les freqüències dominants es troben entre 5 i 20 Hz.

Per a la detecció de ones refractades seran necessaris geòfons de freqüència natural per sota de 5 Hz, sensibles a vibracions entre 5 i 100 Hz.

S'ha de destacar que la distància entre la font generadora de ones acústiques i el geòfon és gran comparada amb la profunditat de refracció, viatjant les ones horitzontalment.

A diferència de la sísmica de reflexió, en la sísmica de refracció els sensors són autònoms i es mantenen fixos i immòbils durant l'experiment. A terra, els sensors (estacions sísmiques) van començar a utilitzar-se al principi del segle XX, però a nivell marí el seu desenvolupament i utilització és molt més recent. Els sismòmetres de fons oceànic (OBS Ocean Bottom Siesmometers) van començar a desenvolupar-se en la dècada dels 70, i a partir de llavors la seva tecnologia no ha parat d'evolucionar, la seva utilització en l'àmbit de les Ciències de la Terra s'estén ràpidament i són cada vegada més nombrosos els grups de diversos països que intenten desenvolupar i perfeccionar els seus propis instruments. De moment, l'autonomia, la fiabilitat, i el número d' OBS disponible per la comunitat científica internacional són encara inferiors a les de les estacions terrestres, però els progressos en aquest sentit són ràpids i evidents. En la pròxima dècada s'espera disposar d'una cobertura similar d'OBS equivalents a les estacions terrestres. Un dels objectius principals és el d'obtenir imatges estructurals d'alta resolució en els marges actius, per a poder localitzar amb precisió la sismicitat, i amb ell, identificar les estructures tectòniques (falles actives) i comprendre els processos de deformació activa generadores dels grans terratrèmols que afecten periòdicament aquestes regions.

1.2. QUE ÉS UN OBS?

Un OBS consisteix en un equip de mesura que permet registrar les vibracions sísmiques del fons marí. Aquestes senyals són provocades pel moviment de les plaques sísmiques o/i de explosions, tant naturals com artificials. Per tal d'adquirir aquestes dades, l'OBS està preparat per incorporar dos tipus de sensors, un geòfon de fins a tres eixos ortogonals entre si per enregistrar les vibracions del subsòl, i un hidròfon per les vibracions procedents de la columna d'aigua.

El mode de funcionament és totalment autònom, degut a que la seva font d'alimentació prové de bateries que porta incorporades. Per tal de començar a captar dades, el sistema es deixa anar a l'oceà, i per gravetat baixa fins que fa contacte amb el fons marí i així començarà a captar dades. Un cop finalitzat el temps previst per l'experiment, es generà un senyal acústic per mitjà d'un transductor acústic, provocant l'activació de l'alliberador mecànic (figura 1.4) que deixa anar el llast fent possible que l'equip torni a sortir a superfície per la seva pròpia flotabilitat.



figura 1.4 Alliberador mecànic “*deepsea release unit*” de GURALP SYSTEMS

A grans trets, els sismòmetres marins, inclouen una estructura principal hermètica per encabir l'electrònica, un llast per fer-lo baixar a les grans profunditats marines, els dos sensors (geòfon i hidròfon) per captar les dades, l'alliberador acústic per interpretar el senyal d'alliberament, el transductor acústic per rebre el senyal d'alliberament (figura 1.5), l'alliberador electromecànic per deixar el llast i començar l'ascens, el flotador per un cop alliberat el llast tenir flotabilitat positiva, el sistema d'emmagatzematge de dades on s'inclou l'electrònica de control, una font d'alimentació amb les seves bateries, i d'un sistema d'il·luminació per trobar l'equip durant la nit un cop finalitzat l'ascens.



figura 1.5 Transductor acústic “*PET191E*” de IXSEA

Les dades recollides per l'instrument, poden proporcionar informació sobre les sub-capes del fons de l'oceà fins a una profunditat de 40 km sota el fons de l'oceà. L'exactitud dels resultats depèn directament de l'estabilitat de la temperatura del vidre utilitzat com la principal base de temps de l'equip. En sismologia passiva, el sismòmetre recull les vibracions del fons marí causades per una font natural (terratrèmol), on l'objectiu és determinar la magnitud i la localització de l'activitat.

En els experiments de refracció activa, una sèrie de OBS es despleguen en el fons del mar i una font artificial (canons d'aire comprimit) és arrossegada pel vaixell oceanogràfic a fi de generar senyals acústics de tant en tant durant l'experiment. El senyal generat es propaga cap al fons del mar, així com a través del subsòl sent reflectit i transmès per les diferents sub-capes del fons de l'oceà. Aquests senyals són recollits pels sensors de l'OBS, marcats en temps i emmagatzemats en una targeta Compact Flash de memòria, després del processat de les dades en el laboratori, es pot obtenir un mapa del llit marí a una profunditat de 40 km sota el fons marí, i informació sobre l'ample i el material de cada capa. En aquest cas, el paràmetre que proporciona aquesta informació és la velocitat del so a través de diferents capes, velocitat que s'estima pel coneixement exacte del temps transcorregut des de la generació d'un senyal acústic per la font artificial i el registre de les dades en l'OBS. Se sap que la velocitat del so en la columna d'aigua és de 1500 m/s aproximadament [10].

Mentrestant, els instants de tret dels canons d'aire comprimit són controlats per un sistema GPS (Global Positioning System) ubicat al vaixell [11], l'OBS submergit no té accés a aquesta senyal de sincronització durant tot l'experiment. El rellotge de l'OBS es sincronitza amb el senyal GPS, abans del seu desplegament i la deriva del temps del rellotge es calcula després de la recuperació de l'OBS utilitzant el mateix senyal. En l'etapa de processament de senyals, les marques de temps es corregeixen en el cas que la deriva de temps de l'OBS és lineal durant l'experiment.

A les imatges que tenim a continuació podem observar una sèrie de sismòmetres marins en diferents formats. La primera d'aquestes, la figura 1.6, correspon al MiniDOBS (Mini-Digital Ocean Bottom Seismometer) desenvolupat l'any 1996 per la universitat de Cambridge juntament amb Carrack Measurement Technology, i que posteriorment ha estat modificat pel grup de recerca SARTI (Sistemes d'Adquisició Remota i Tractament de la Informació) de la UPC. A la figura 1.7, tenim un sismòmetre marí construït l'any 2005 per IRD-GeoAzur amb la idea de poder acoblar geòfons de majors dimensions i un increment del temps d'autonomia. L'any 2002, GEOMAR, van crear un disseny amb la idea de obtenir un OBS de reduïdes dimensions, a la figura 1.8 podem observar un dels seus sistemes.



figura 1.6 Estructura MiniDOBS i electrònica UPC-SARTI



figura 1.7 Hippocampe de IRD-Geoazur



figura 1.8 K/MT 562 de Geomar

1.3. EL GEÒFON I L'HIDRÒFON

El geòfon és l'encarregat de convertir els senyals físics del moviment produït per les vibracions transmeses pel terra, en variacions de tensió diferencials, les quals són captades per la targeta d'adquisició de dades analògic-digital (figura 1.9). El geòfon que utilitzarem en l'actual OBS, correspon amb un dispositiu dissenyat amb tres sensors de SM-6 amb una sensibilitat de $28,8 \text{ V/ms}^{-1}$.



figura 1.9 Geòfon utilitzat en l'OBS de SARTI-UPC

L'hidròfon consisteix en un dispositiu encarregat de transformar les vibracions acústiques transmeses pel medi marí, en variacions de tensió al igual que el geòfon. Aquests senyals serveixen per obtenir més informació sobre l'entorn marí. En el cas de d'aquest dispositiu, l'OBS desenvolupat per SARTI-UPC, incorporarà un hidròfon HTI-90-U de Hightech amb una sensibilitat de 1778 V/Bar (figura 1.10).



figura 1.10 Hidròfon utilitzat en l'OBS de SARTI-UPC

1.4. EL MICROPROCESSADOR MCF54455

El microprocessador MCF54455 utilitzat en el projecte, està basat en la versió 4 del nucli de ColdFire amb una velocitat de fins a 410 Dhrystone2.1 MIPS (Megainstruccions per segon) tenint una velocitat de sistema de 266MHz. Algunes de les característiques necessàries a disposar consisteixen en la comunicació SPI, la comunicació per port sèrie, els ports GPIO (port de propòsit general), la comunicació ATA per la connexió amb la Compact Flash i els temporitzadors. Una de les prestacions més importants que ofereix aquest microprocessador, consisteix en el baix consum que pot treballar aquest microprocessador i el marge de freqüències en el que pot treballar. A la figura 1.11 es pot veure un diagrama de blocs del dispositiu.

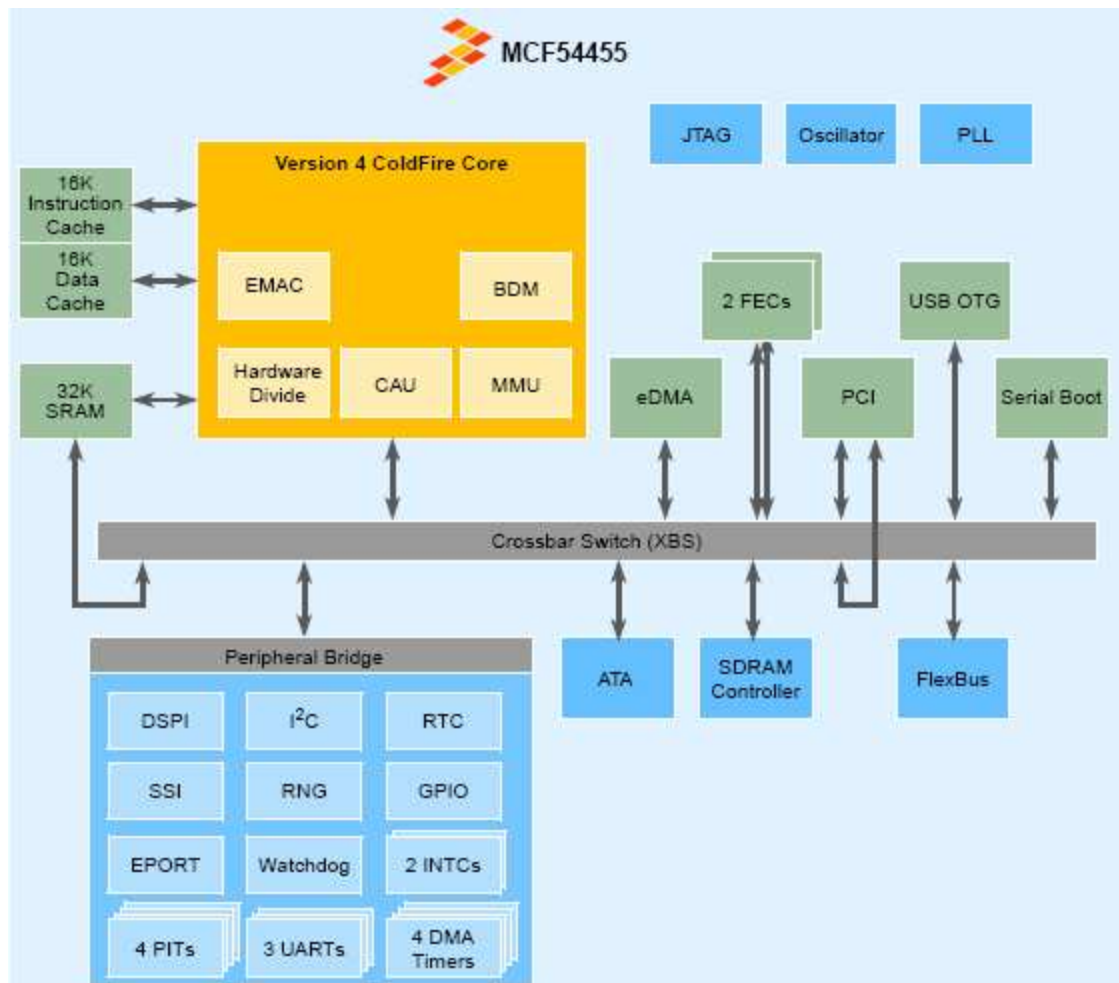


figura 1.11 Diagrama de blocs del MCF54455

El microprocessador pot treballar fins a 266MHz, freqüència que pot ser modificada a diferents valors segons dos modes de funcionament, el “PLL mode” i el “LIMP mode”. Cada un dels modes de funcionament, permeten diferents freqüències de treball, el primer d'aquest correspon al grup de les altes freqüències de treball que oscil·len entre els 50 i els 266 MHz, en el segon dels modes correspon al de les freqüències baixes que poden oscil·lar entre els 40 MHz (referència màxima del cristall d'entrada) i els 500 Hz (per una referència mínima del cristall d'entrada de 16MHz), provocant una reducció en el consum considerable.

Un altre dels elements dels que disposa el microprocessador consisteix en el mòdul EPORT, encarregat de gestionar algunes interrupcions de les que disposa el microprocessador i que són accessibles des dels pins exteriors.

Un dels avantatges que anteriorment s'ha esmentat respecte aquest microprocessador, consisteix en el baix consum en que pot treballar. Un dels mètodes per treballar en baix consum, és el de fer-ho a una freqüència baixa fent ús del mode *LIMP*, que a part de reduir-ne el consum també deshabilita algunes de les controladores del sistema. A part d'aquest mètode per reduir el consum, el microprocessador també pot treballar en quatre modes de funcionament diferents que són el RUN, el WAIT, DOZE i el STOP.

Un cop s'entra en algun d'aquest modes, tret del RUN que consisteix en el funcionament normal, el sistema para alguns dels rellotges de treball, segons el mode que s'hagi seleccionat, i per poder-ne sortir s'ha de generar una interrupció del nivell que prèviament s'hagi programat. En el cas del mode WAIT, consisteix en parar el rellotges de la CPU i de la memòria, fent que tots els perifèrics segueixin treballant normalment. En el cas del mode DOZE, també es paren els rellotges de la CPU i de la memòria, com també ho fan els dels perifèrics que s'hagin programat. Finalment, en el cas del STOP, tots els rellotges del sistema es paren fins que entra una interrupció del nivell programat.

Una de les característiques necessàries pel desenvolupament de l'aplicació consisteix en el protocol de comunicacions SPI (serial peripheral interface). Aquest protocol està compost per un mestre i per un o més esclaus. En el nostre cas tenim el mestre compost pel microprocessador i un sol esclau compost per el mòdul d'adquisició de dades analògiques a digitals. En el cas de l'esclau, està compost per solament 4 línies de comunicació, una per enviar les dades, una per rebre les dades, una pel senyal de rellotge i la darrera per habilitar l'esclau. En el cas del mestre tindrà les mateixes línies de comunicació que un esclau i tantes línies addicionals com altres esclaus tinguem. En el nostre cas només està compost per les 4 línies que estan connectades amb l'únic esclau que tenim.

1.5. OBJECTIUS

El present projecte ha estat realitzat en el grup de recerca SARTI de la UPC, on s'està desenvolupant des de fa 10 anys sismòmetres marins totalment autònoms i de llarga durada. El programari de la placa de control forma totalment part del desenvolupament de la darrera versió de l'OBS construït per SARTI-UPC. L'objectiu del present projecte consisteix en realitzar una aplicació en C, enfocada al microprocessador ColdFire MCF54455 de Freescale, pel control i gestió de tots els recursos, com també el desenvolupament de tot el procés d'adquisició de dades sísmiques. Així mateix s'ha desenvolupat i construït una placa de control basada en el microprocessador ColdFire MCF54455, sobre la que corre el programa desenvolupat.

2. DESENVOLUPAMENT DE L'ELECTRÒNICA DEL SISTEMA

El punt de partida pel desenvolupament d'un nou sistema d'adquisició de dades sísmiques marines, comença a partir del disseny desenvolupat amb anterioritat dins del grup de recerca de SARTI-UPC [1], en aquest sistema que tenim representat a la figura 1.6, l'electrònica està distribuïda a l'interior d'una esfera de cristall. En el nou OBS dissenyat, la nova electrònica està ubicada dins d'uns cilindres d'alumini, comportant un redisseny de les plaques base.

A la figura 2.1 es mostra el diagrama de blocs de la versió prèvia al sistema de control de l'electrònica d'adquisició dels OBS desenvolupats per SARTI-UPC amb anterioritat. Aquest sistema es basa en un datalogger TT8 que integra un microcontrolador MC68332 de Motorola (<http://www.onsetcomp.com/products/manuals/tt8-manset>). El microcontrolador incorpora la interfície de comunicació SPI, la qual utilitza per comunicar-se amb la placa d'adquisició de dades. Aquesta placa d'adquisició, també ens permet convertir les dades adquirides des dels quatre canals de que disposa, d'analògic a digital, abans de ser enviades pel port SPI.

El datalogger també disposa d'una RTC (Real TimeClock), la qual és en realitat un microcontrolador PIC que té aquesta funció i un convertidor de 8 canals de 12 bits. A part d'això, el sistema també disposa de 1MB de memòria RAM i de memòria Flash, per executar l'aplicació. Com a sistema d'emmagatzematge de les dades adquirides, s'utilitza una targeta de memòria de tipus Compact Flash de 1GB la qual esta ubicada en un altre placa que està connectada amb el datalogger. I finalment, aquest també disposa d'una connexió sèrie RS232 per comunicar-se amb l'ordinador i poder tant configurar el sistema com arrancar-lo.

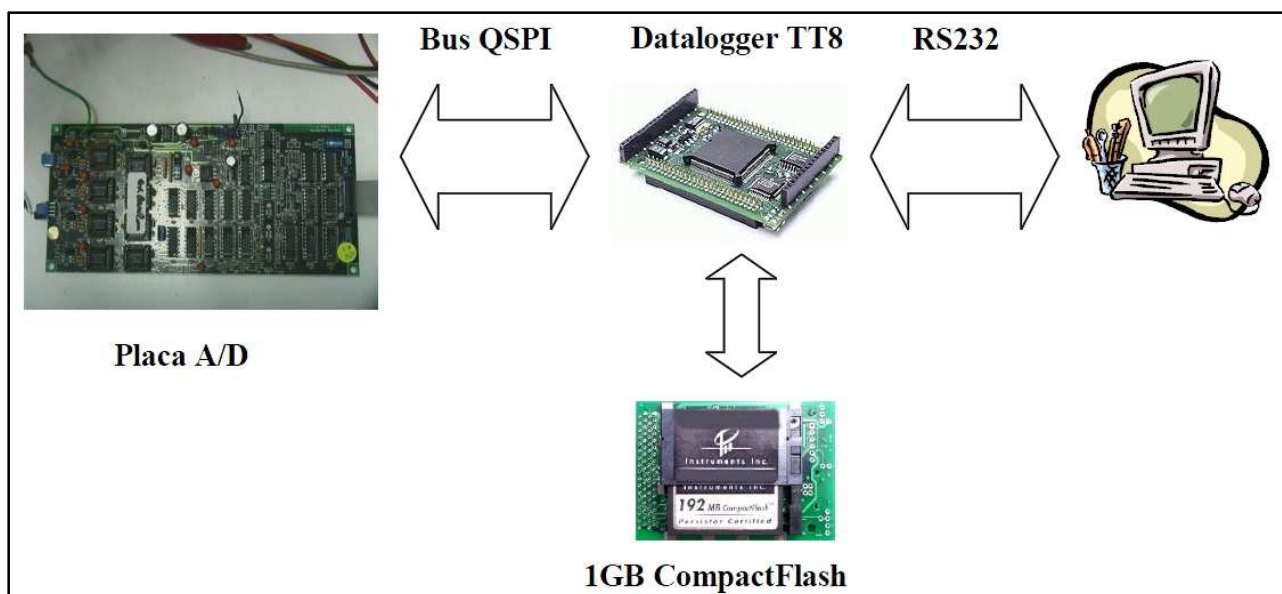


figura 2.1 Diagrama de blocs de la versió prèvia del OBS desenvolupat per SARTI-UPC

Abans de fer el nou disseny hardware-software i per tal de compactar tot el sistema, millorar el rendiment de treball i augmentar les prestacions, es va generar una nova aplicació a partir del codi anterior [2], fent els canvis adequats i generant nou codi. A la figura 2.2, hi ha la imatge de la placa d'avaluació utilitzada per realitzar aquestes proves i per desenvolupar un nou codi en l'adquisició de dades sísmiques.

Aquesta placa avaluadora, disposa de totes les prestacions del microprocessador ColdFire MCF54455 disponibles [3], però algunes d'aquestes parts, desapareixeran en el nou prototip. Per exemple, el port PCI, és un dels elements sense possibilitat de ser utilitzat en el disseny de la nova placa, un altre element dels elements innecessaris, és la FPGA que incorpora la placa d'avaluació per gestionar algunes de les possibilitats de la placa o per realitzar proves de programació. Per tant, a l'hora de generar el nou disseny de la placa del microprocessador, s'havia de realitzar una selecció de les parts desitjades i treure les que no ho eren.



figura 2.2 Placa d'avaluació M54455EVB de Freescale

Pel desenvolupament del nou disseny de la placa del microprocessador presentada en aquest treball de fi de carrera, es va partir de l'esquemàtic de la placa avaluadora [5] i es va dibuixar de nou l'esquemàtic del que seria la nova placa del microprocessador, per mitjà de l'aplicació Altium2008. En alguns casos, s'havien de modificar alguns integrats per ajustar-se a les necessitats que teníem, com per exemple va succeir en el cas de l'integrat de la font d'alimentació que adaptava els 3,3V de la tensió d'entrada a les tensions necessàries del microprocessador, que en la placa d'avaluació, disposava de més tensions derivades del port PCI o de la FPGA.

2.1. PLACA MICROPROCESSADOR

Les parts necessàries que havíem de tenir en compte a l'hora de realitzar la nova placa del microprocessador, eren les següents:

- Circuit d'alimentació, per adaptar els 3,3V a 1,8V i 1,5V per les diferents parts del circuit.
- Circuit de la memòria flash
- Circuit de la memòria SDRAM
- Circuit de la nova memòria SRAM
- Un port ethernet
- Circuit per connectar el BDM i per la configuració de arrancada del microprocessador
- Port sèrie
- Connectors usb

- Port amb pins GPIO no utilitzats
- Connector per la comunicació SPI amb la placa d'adquisició de dades
- Circuit per les senyals de rellotge
- Port ATA per la comunicació amb la Compact Flash

Circuit d'alimentació:

En el circuit d'alimentació, existia com a punt de partida dos integrats per generar 4 tensions diferents de les quals una era la mateixa d'entrada de 3,3V per estabilitzar-la i un altre era utilitzada per la FPGA. En aquest cas només era necessari convertir els 3,3V a dues tensions diferents, la de 1,8V i la de 1,5V, i per tant amb un sol integrat era suficient per realitzar aquesta tasca. El circuit integrat que s'ha utilitzat per realitzar aquesta conversió és el *MC34717EP* que ens permet treure les dues tensions necessàries.

Memòria flash:

El la circuit de la placa de avaluació, es disposava de dues memòries flash col·locades adequadament en el Flexbus, una de 512 kB i l'altre de 16 MB. En el nostre cas, no calia tenir-ne dues ja que el consum general de la placa final hauria resultat superior, i per tant només s'ha procedit a tenir-ne una de sola, i s'ha seleccionat la més gran de les dues que correspon a una memòria de 16 Mb amb la referència *28F128J3*.

Memòria SDRAM:

Originàriament, a la placa d'avaluació hi ha quatre integrats SDRAM del tipus DDR2 formant una sola memòria de 256 MB. En el cas de la nova placa del microprocessador porta un sol integrat, fent que el consum sigui menor i que tinguem un total de 64 MB de memòria. El circuit integrat utilitzat en aquesta part del circuit com a memòria DDR2 SDRAM, correspon al *MT47H32M16*. Com que a l'hora de treballar a baixa freqüència la controladora d'aqueta memòria deixa de funcionar, es planifica per ficar-la igualment en el disseny de la nova placa amb la intensió de tenir-la encara que no serà utilitzada, degut a que es dissenya la incorporació d'una nova memòria SRAM que si permetrà treballar a baixa freqüència.

Nova memòria SRAM:

Com s'ha vist en el circuit de la memòria SDRAM, la problemàtica de poder utilitzar aquella memòria a baixes freqüències, va conduir a incorporar una nova memòria en un bus diferent de l'anterior, la qual si permetia treballar a baixa freqüència gràcies a no utilitzar la controladors de la SDRAM. Aquesta memòria és la *CY62167EV30* que està connectada directament al Flexbus al igual que la memòria flash, i té una capacitat de 2 MB de memòria. Les connexions amb el microprocessador, són les mateixes que hi ha amb la memòria flash per estar utilitzant el flexbus, i el que ens diferencia l'un de l'altre, correspon al pin selector (el *CS0* i el *CS1*).

Port ethernet:

La placa d'avaluació ve preparada amb dos ports ethernet amb la possibilitat de donar-li altres utilitats als pins, és a dir per ser configurats per altres funcions com poden ser els GPIO. En el cas del pins corresponents al segon, s'han configurat per treballar com a ports GPIO i ser utilitzats en el port ATA, en canvi s'ha mantingut el primer dels dos ports ethernet per tal de poder disposar d'un d'ells a la nova placa d'adquisició. En aquest cas, s'ha canviat el circuit integrat que hi havia originàriament a la placa d'avaluació per un altre que només gestiona un sol port, aquest integrat és el *DP83848i*. A les següents figures (figura 2.3 i figura 2.4) tenim l'esquema corresponent al port ethernet generat per la nova placa del microprocessador.

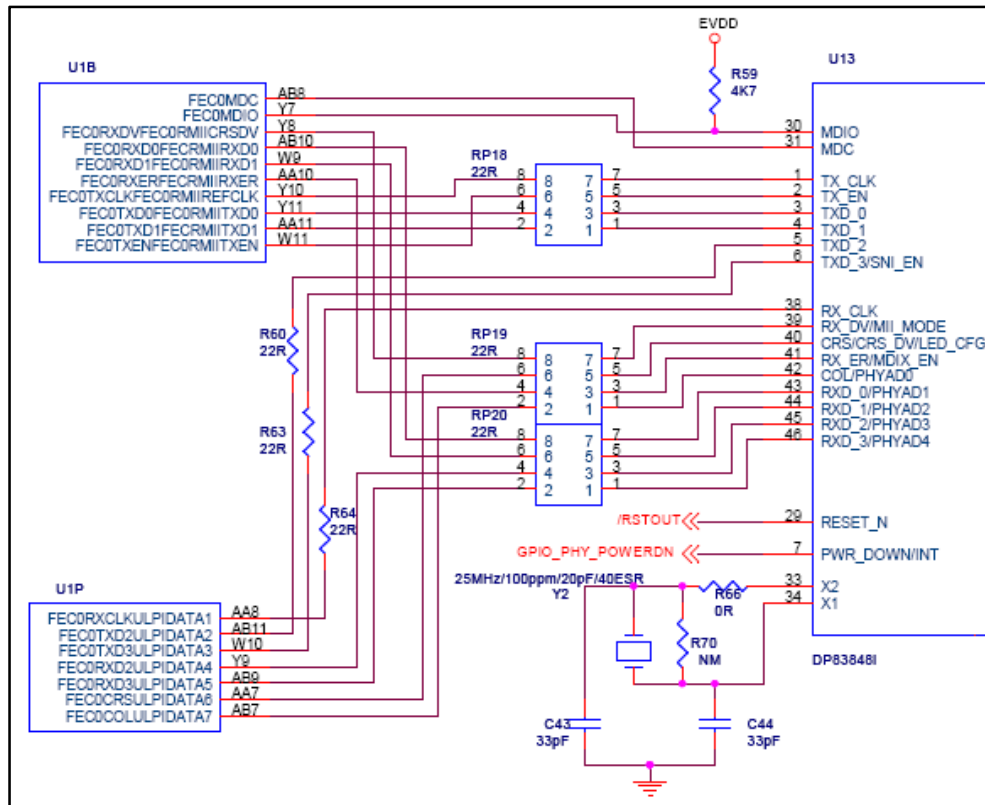


figura 2.3 Part 1 de l'esquema del port ethernet

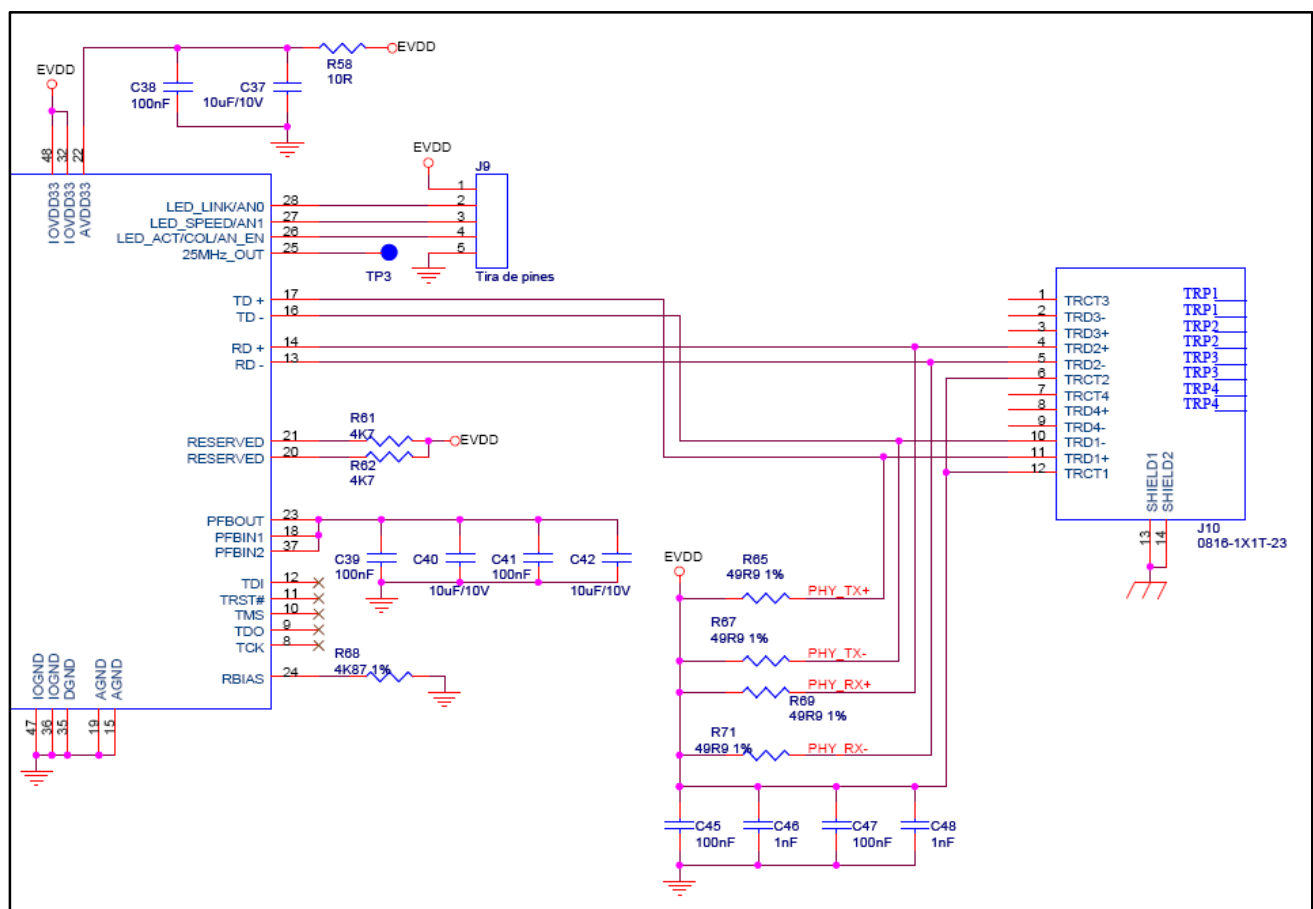


figura 2.4 Part 2 de l'esquema del port ethernet

Circuit pel BDM i per la configuració de arrancada del microprocessador:

Aquesta part del circuit ens hem basat completament en la versió de la placa d'avaluació i a més s'ha comprat un aparell per poder realitzar la comunicació entre el PC i el microprocessador per carregar les aplicacions. Aquest dispositiu és el USB-ML-CF de ColdFire [9], i el circuit realitzat consisteix en el corresponent connector per connectar aquest dispositiu amb el microprocessador i dels diferents components passius necessaris. A la figura següent (figura 2.5) podem veure el dispositiu del que s'ha comentat anteriorment per comunicar els dos elements.

En referència a la part de la configuració de arrancada, es disposa d'una sèrie de vuit microinterruptors amb els corresponents components, i connectat al microprocessador per mitja del flexbus i així configurar el sistema al començament. Aquesta part només s'utilitza en el moment del inici del sistema, un cop iniciat, aquest no l'utilitza més.



figura 2.5 Dispositiu per programar el microprocessador

Port sèrie:

Per la comunicació entre la placa del microprocessador i l'ordinador, un cop arrancat el sistema, es utilitza el port sèrie, amb aquest es transmeten les dades referents al menú del sistema i permet la configuració per part de l'usuari. En aquest circuit hem utilitzat l'integrat ST3232, situat entre el microprocessador i el connector del port sèrie. Per aprofitar aquest connector, i degut a que els connectors utilitzats sota l'aigua són molt més cars, s'ha fet ús d'un dels pins d'aquest connector per fer-hi passar el senyal del GPS en direcció a un pin GPIO (*GPIO_T2IN*). A la figura 2.6 podem veure com ha quedat la connexió del connector del port sèrie referent al que s'ha comentat.

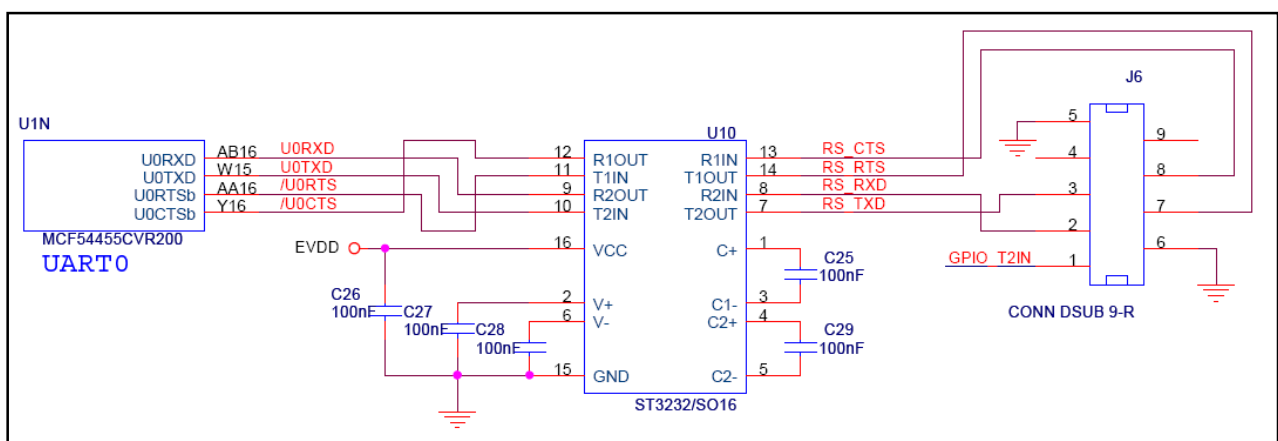


figura 2.6 Circuit port sèrie

Connectors usb:

En el circuit dels connectors USB, de la nova placa del microprocessador, tenim dos tipus diferents, l'USB de tipus A i el de tipus B, els quals per defecte, estan desconnectats. En el cas del tipus A, esta preparat per que un cop es tingui que utilitzar, s'hagi d'alimentar el dispositiu que s'hi vol connectar de forma externa, de manera que no consumeixi energia del sistema, i en el cas dels pins de dades, estan connectats al microprocessador amb normalitat. Per altre banda, el connector de tipus B, utilitza l'integrat *CP2102*, el qual converteix la comunicació del segon port sèrie RS232 del microprocessador, en la comunicació pel port USB de tipus B. A la figura 2.7 podem veure els dos esquemes en representació als corresponents circuits dels ports USB.

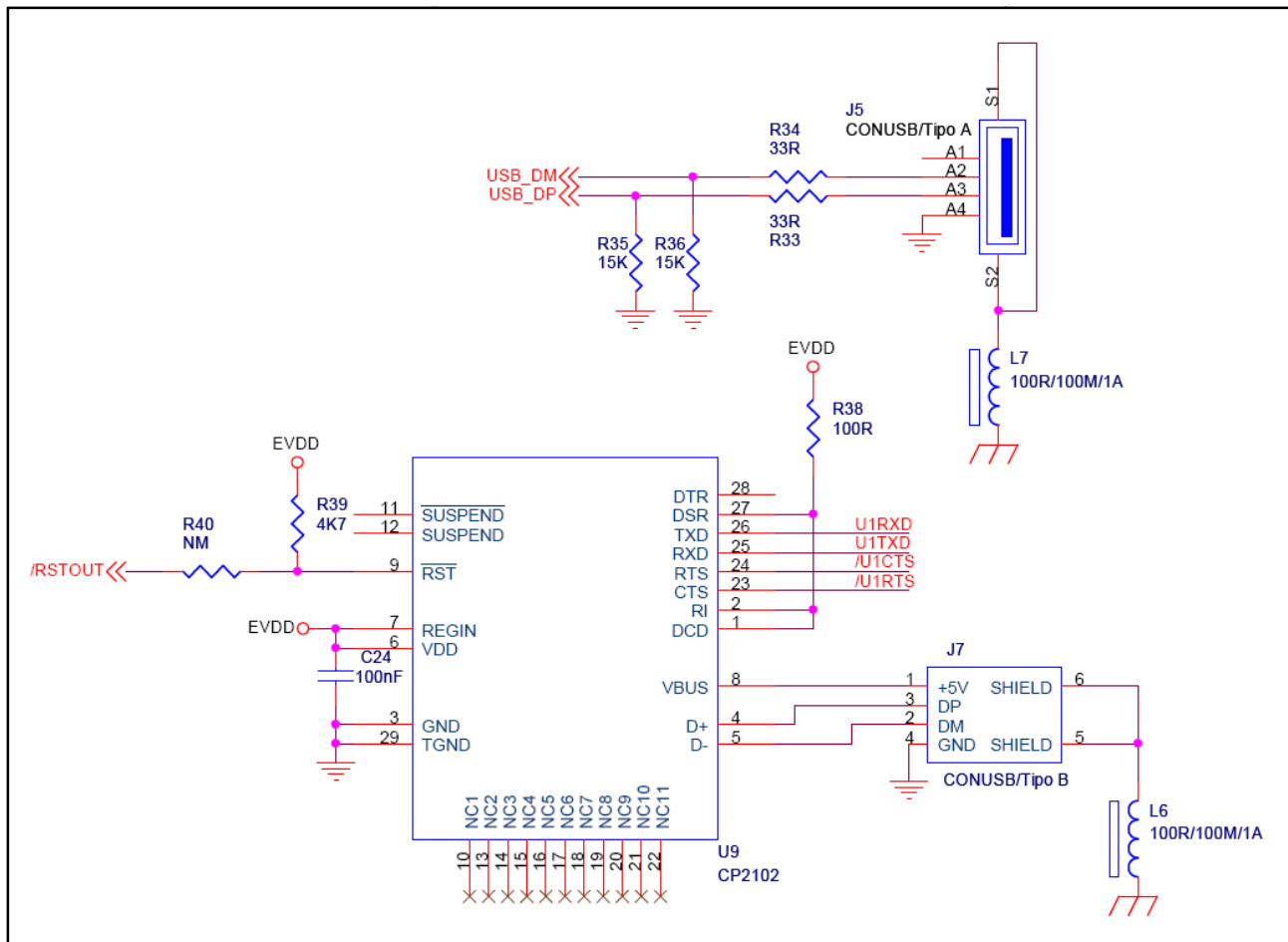


figura 2.7 Esquema dels dos circuits creats per l'USB

Port GPIO:

A l'hora de passar del circuit de la placa d'avaluació al circuit de la nova placa del microprocessador, s'han utilitzat diferents ports configurats com a GPIO per la realització de diferents parts, com són per exemple els pins de comunicació amb el port ATA que s'explica més endavant, o com per exemple el pin GPIO utilitzat per deshabilitar l'integrat del port ethernet. Amb els pins amb possibilitat de ser configurats com a GPIO i els quals no han estat utilitzats en cap altre part de l'esquema, s'han reunit tots creant un connector addicional. A més, també s'ha aprofitat aquest port per afegir alguns pins d'interrupcions i les tensions d'alimentació. A la figura 2.8 tenim el port amb els diferents pins que han quedat accessibles.

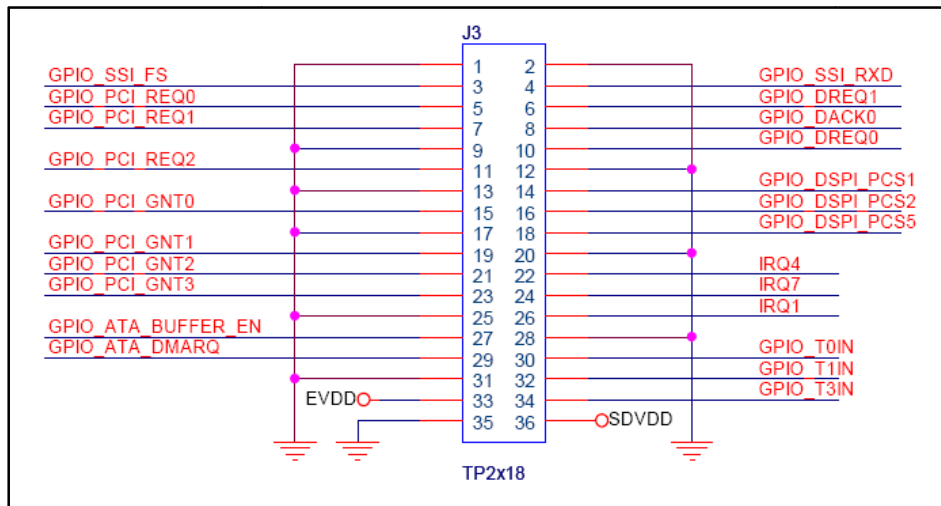


figura 2.8 Esquema del port GPIO

Comunicació amb la placa d'adquisició de dades:

Per realitzar la comunicació des de la placa d'avaluació amb la placa d'adquisició de dades, es disposava d'un port que tenia accessibles les corresponents línies utilitzades en el protocol de comunicació SPI a més de l'accés a la interrupció necessària per la comunicació. En el cas de les senyals de freqüència, eren connectades exteriorment a partir de generadors de funcions. En el disseny de la nova placa, aquestes senyals de freqüència procedeixen de la mateixa placa del microprocessador, pel que també flueixen per mitjà del mateix connector de comunicació. Les senyals utilitzades en aquest port són les següents:

DSPI_SIN: Senyal d'entrada en la comunicació SPI

DSPI_SOUT: Senyal de sortida en la comunicació SPI

DSPI_SCK: Senyal de rellotge de la comunicació SPI

CLK_ADC_32_768: Freqüència de treball del filtre de la placa d'adquisició

IRQ3: Interrupció al sistema per indicar que les dades estan llestes per ser enviades

DSPI_PCS0: Habilitació del dispositiu esclau amb el que es desitja comunicar-se dins del protocol SPI

CLK_ADC_SYNC: Freqüència de mostreig de les dades

Circuit per les senyals de rellotge:

La senyal de rellotge utilitzada en el circuit de la placa d'avaluació, correspon a un circuit que ens genera un rellotge de 33,33MHz que alimenten el microprocessador i totes les freqüències externes al microprocessador. En el cas de la nova placa del microprocessador, aquesta senyal procedeix d'un altre circuit, una placa creada per generar un senyal de rellotge més precís i més fiable. Aquesta placa és la placa de seascan inc. amb el model *SISMTB v.4.0* (figura 2.9) [6], que ens genera una freqüència de 125Hz, la qual per mitjà d'un PLL (*CS2300CP*) es converteixen en una freqüència de 32,768 MHz, freqüència que s'utilitza tant per alimentar el sistema com per alimentar el filtre de la placa d'adquisició.



figura 2.9 Placa seascan generadora de rellotge

A més, a partir d'aquesta freqüència generada i per mitjà d'altres integrats, es generen les diferents freqüències que poden ser utilitzades dins la placa d'adquisició com a freqüència de mostreig i la freqüència de la RTC del sistema que és de 32,768kHz. Aquests integrats són el 74LV4060, el 74HC4024 i el 74HC4051, en el cas dels dos primers s'encarreguen de dividir la freqüència d'entrada a freqüències menors, i en el cas del tercer, consisteix en un multiplexor per seleccionar una de les freqüències creades com a freqüència de sortida per ser utilitzada com a senyal de mostreig. A la següent imatge (figura 2.10) podem observar com estan connectats els tres integrats utilitzat per seleccionar la freqüència de mostreig. Aquesta selecció es realitza amb l'ajuda de tres senyals de control GPIO, els quals es poden observar connectats en els pins 9, 10 i 11 de l'integrat U8.

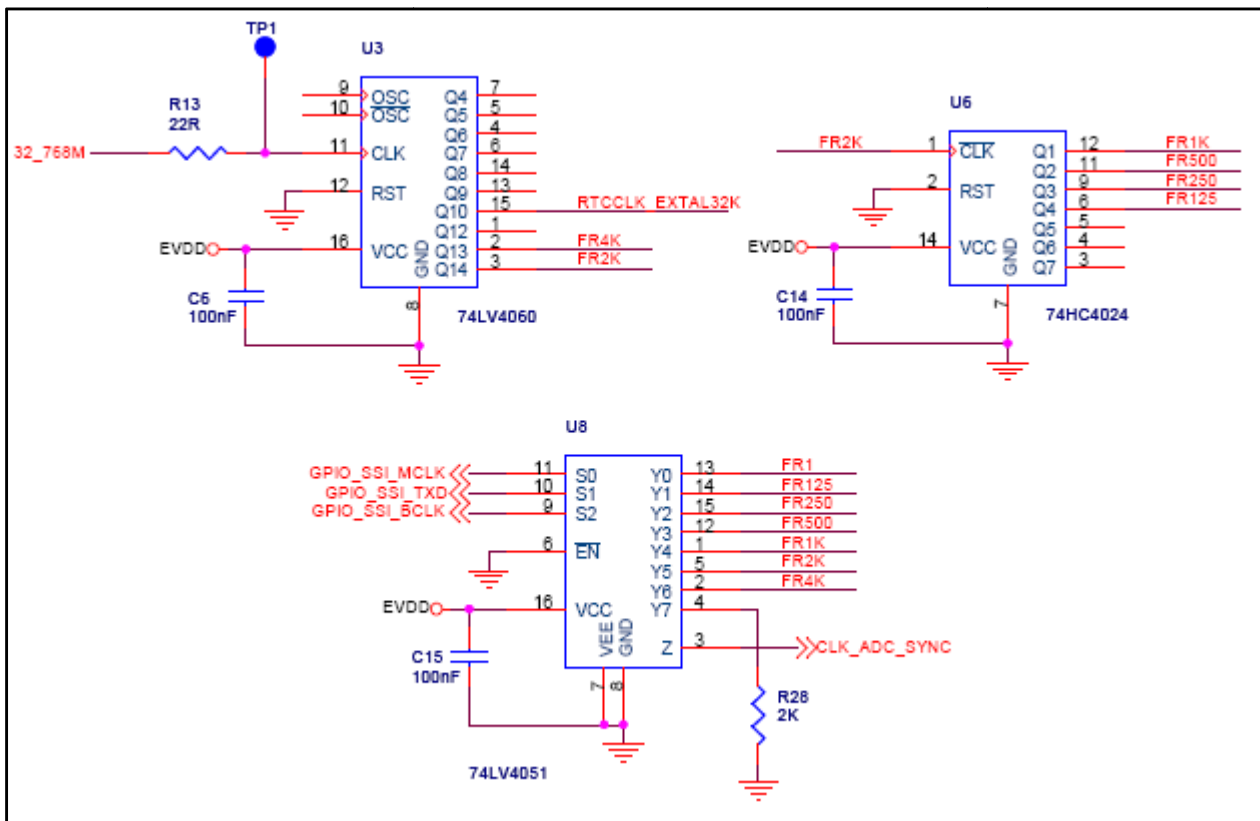


figura 2.10 Esquema selecció freqüència de mostreig

Port ATA per la comunicació amb la Compact Flash:

A l'hora d'emmagatzemar les dades obtingudes des de la placa d'adquisició, a la placa d'avaluació, s'utilitza una targeta Compact Flash que està connectada per mitja d'un altre targeta que ens converteix el connector ATA al connector de la targeta de memòria. En el cas de la nova placa del microprocessador, s'ha ficat directament el connector de la compact flash amb el que s'ha aconseguit eliminar la conversió dels connectors. A més del connector de la targeta de memòria, també s'ha col·locat el buffer entre el connector i el microprocessador, deixant l'esquema com el que podem veure a la figura 2.11 i la figura 2.12.

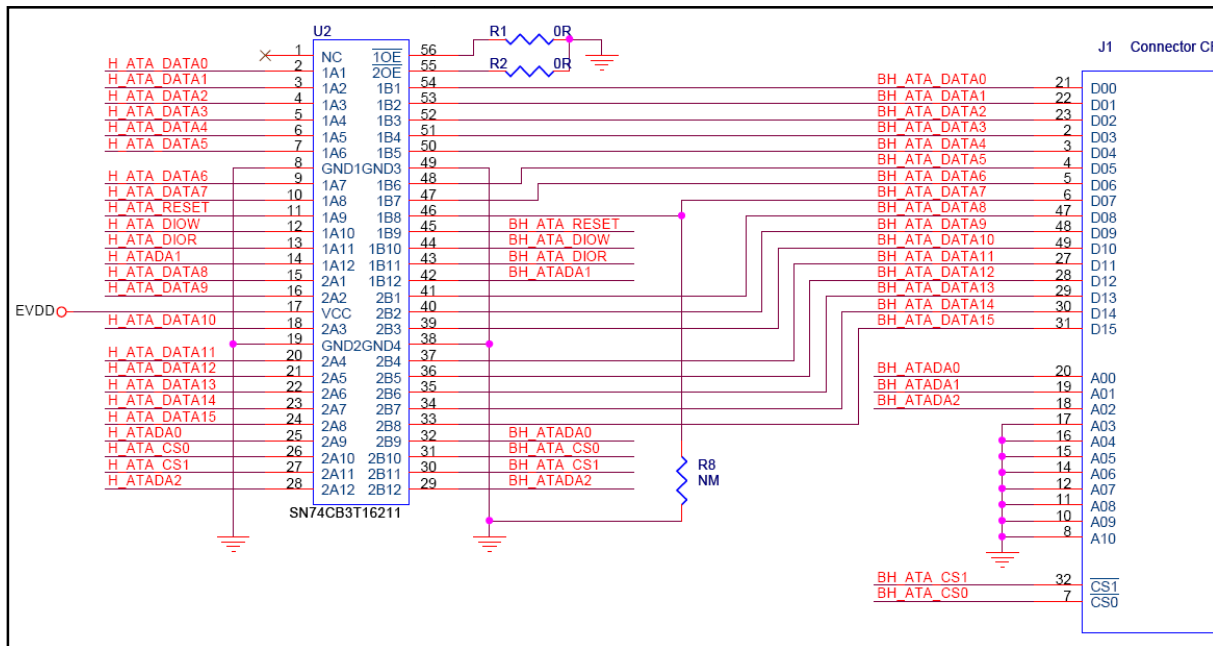


figura 2.11 Part 1 de l'esquema de comunicació amb la Compact Flash

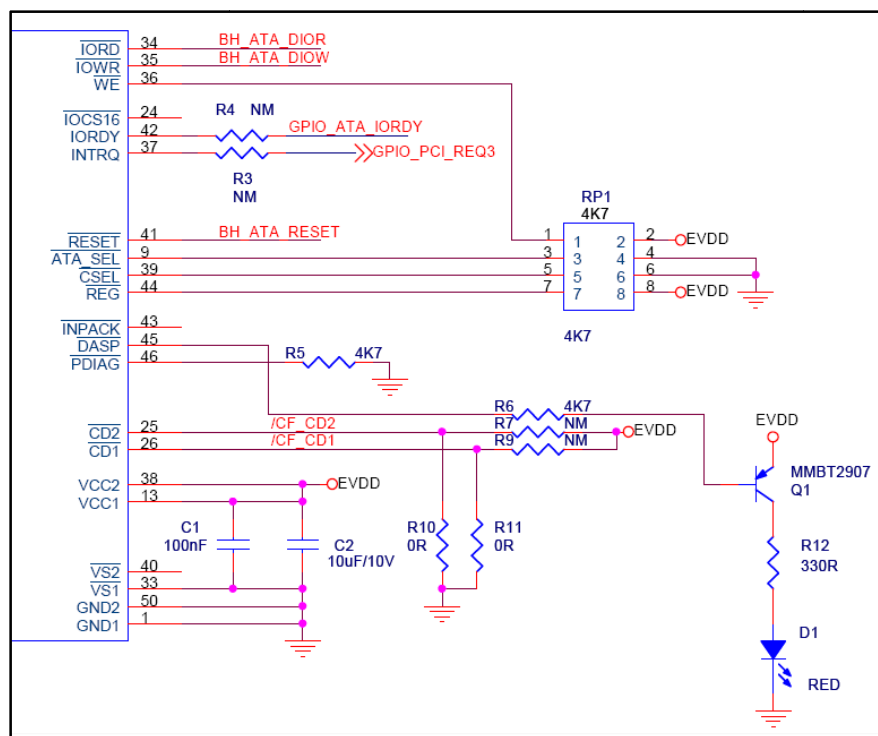


figura 2.12 Part 2 de l'esquema de comunicació amb la Compact Flash

Amb aquest circuit com a resultat, queda tot preparat per treballar amb el nostre codi de comunicació per mitjà dels ports en configuració GPIO, com també per treballar amb la controladora del port ATA per les possibles funcionalitats futures. El problema que s'explicarà posteriorment en el capítol 5.2 *proves realitzades*, està en que quan es treballa a baixes freqüències, prova que no es pot realitzar amb la placa d'avaluació, el temps que tarda una part del software en accedir a la compact flash, és massa elevat amb el que també es veurà que s'han de realitzar canvis a nivell de hardware.

Un cop aconseguit per complert l'esquema de tot el que es necessitava, es genera el primer prototip, la primera versió de la placa del microprocessador en PCB, la qual podem veure a la figura 2.13 (la placa multicapa va ser fabricada per una empresa externa). Amb aquesta nova placa, tenim un consum inferior al de la placa d'avaluació degut a la eliminació de totes aquelles parts innecessàries i a més s'ha afegit una memòria SRAM que treballa a baixa freqüència, amb la que podem treballar amb un rellotge de treball menor.

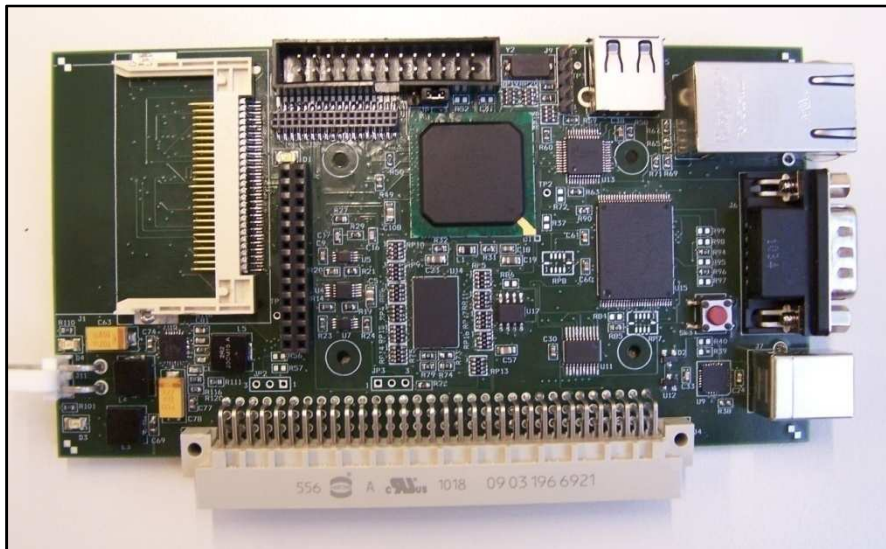


figura 2.13 Nova placa desenvolupada del projecte del nou OBS desenvolupat a SARTI-UPC

2.2. PLACA D'ALIMENTACIÓ

Juntament amb el nou sistema de processat, també s'ha incorporat una nova placa d'alimentació que ens permet subministrar a tot el sistema les tensions i corrents necessàries. Per tant, per alimentar el nostre sistema, es necessita una alimentació de 3.3V per alimentar la placa del microprocessador, i tres alimentacions de 3.3V, 2.5V i -2.5V per alimentar la placa de conversió de dades Analògic a Digital. En el cas de les dues tensions de 3.3V, és necessari generar-les de forma independent entre si per tal d'evitar interferències, principalment en el cas de la placa d'adquisició i conversió d'Analògic a Digital, per la seva delicades i per requerir d'unes especificacions de soroll més ajustades. La font d'entrada al sistema consisteix en un paquet de bateries que ens proporcionaran 3.6 V.

En el cas de l'esquema creat per la placa d'alimentació, s'han creat quatre circuits per cada una de les tensions a generar. Entre aquests quatre circuits, es segueix la mateixa estructura en tots els casos, seleccionant els components adequats per cada situació. A la figura 2.14 podem veure l'exemple d'un dels esquemes creats per generar la placa d'alimentació DC-DC.

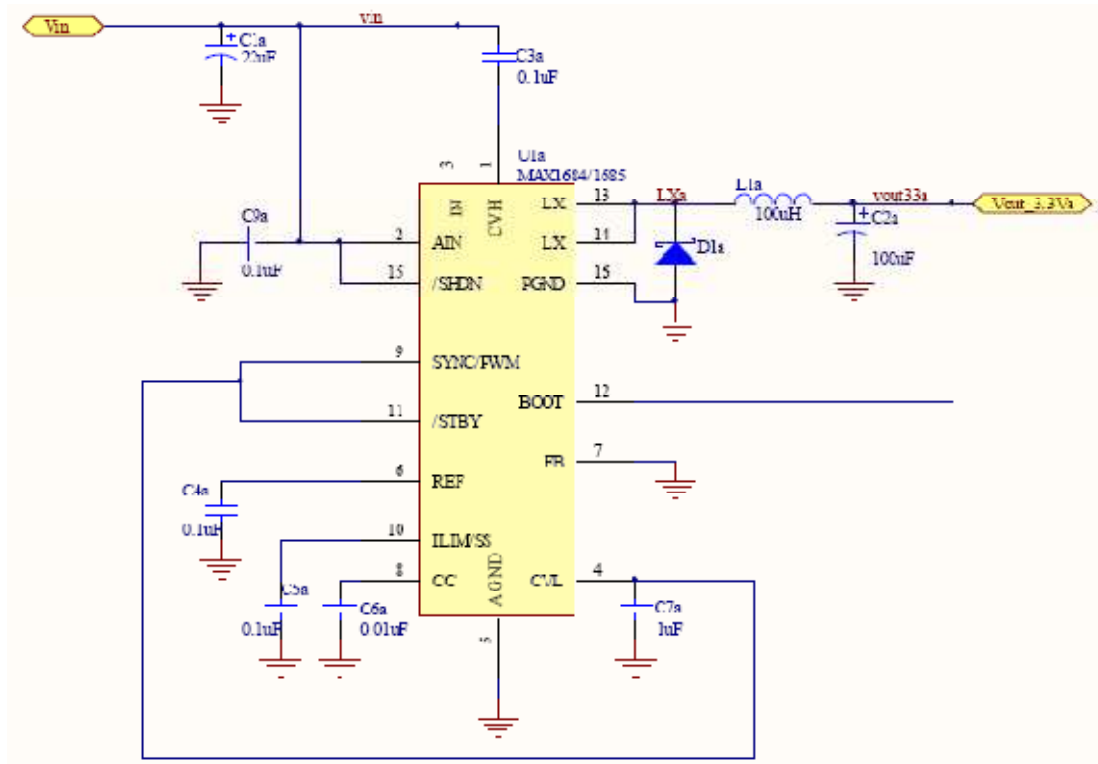


figura 2.14 Esquema per generar els 3,3V destinats a la placa del microprocessador

A la figura 2.15 hi tenim la imatge de la placa d'alimentació creada per alimentar tot el sistema. Si ens centrem en els connectors d'entrada i sortida que hi ha a la placa, a la part de la dreta de la imatge, tenim tres connectors per les diferents alimentacions, i a la part de baix tenim un connector per poder encastar el sistema en una placa comuna de *backplane* per ajuntar les tres plaques del sistema, la del microprocessador, la d'adquisició i la d'alimentació.

En referència als connectors de les alimentacions de la dreta, en primer lloc, a la part superior, tenim el connector amb la tensió d'entrada des de les bateries. En el següent connector, situat al mig d'aquests tres, tenim el connector per alimentar la placa del microprocessador. I finalment, el connector de baix a la dreta (amb quatre pins), correspon a l'alimentació de la placa d'adquisició amb les tres alimentacions necessàries per treballar i amb la corresponent referència de massa.

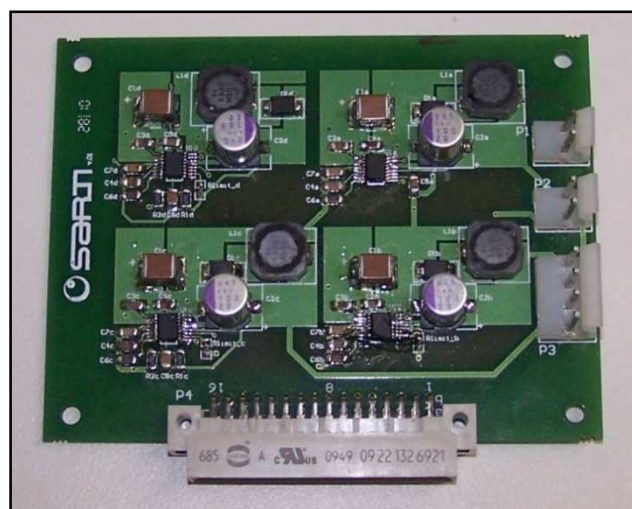


figura 2.15 Placa d'alimentació DC-DC del nou OBS de SARTI-UPC

2.3. PLACA D'ADQUISICIÓ I CONVERSIÓ D'ANALÒGIC A DIGITAL

El mòdul pel qual les dades passen des del sensor al microprocessador, és la placa de conversió de dades analògic a digital, on s'han realitzat petits canvis respecte al disseny anterior [12], a la figura 2.16 podem veure la versió modificada d'aquesta placa. La diferència que s'ha realitzat referent a la versió anterior, consisteix en la eliminació d'uns buffers al mig d'algunes línies de comunicació i la modificació del connector per ser inserit juntament amb la resta de les plaques en una de comuna (placa backplane).

La placa es pot dividir en tres blocs contigus i dependents, el primer consisteix en la part amplificadora del senyal, el segon està destinat al modulador d'aquest, i finalment, el tercer dels blocs, és on es realitza el filtratge del senyal resultant, juntament amb l'enviament al mòdul principal. En el primer bloc, on s'amplifica el senyal, s'utilitzen els integrats CS3301 i CS3302 pel geòfon i per l'hidròfon respectivament. Aquest integrats eleven el senyal dels sensors als nivells adequats de treball, per ser processats posteriorment per la resta de la placa. Llavors, en el segon bloc, es modula el senyal per mitjà del CS5372. Aquest integrat rep el senyal analògic amplificat i el digitalitza per ser tractat pel tercer i últim dels blocs. En la fase de modulador es fa us d'un integrat que està especialment dissenyat per aplicacions geofísiques i de sonar. La darrera de les fases, està composta pel CS5376, component que realitza el filtratge del senyal i l'enviament de les dades per mitjà del protocol de comunicacions SPI (serial peripheral interface) [7].

A la imatge de la figura 2.16, tenim la placa d'adquisició, on a la part dreta de la il·lustració, es poden veure els connectors blaus, per on posteriorment es connectaran els diferents sensors. A la cantonada inferior dreta, hi ha el connector blanc d'alimentació, i el connector gran de baix, correspon al connector de comunicacions per on es transmeten les dades en direcció al microprocessador.



figura 2.16 Mòdul d'adquisició de dades i convertidor d'analògic a digital

2.4. PLACA BACKPLANE

Per tenir totes les plaques unides, tenim la placa *Backplane* que podem observar a la figura 2.17. En aquesta placa hi ha tres simples connectors, dels quals els dos grans corresponen amb la placa del microprocessador i la placa d'adquisició. Les dades transferides entre aquestes dues plaques, ho fan a través d'aquests connectors i de la placa *backplane*. En el cas del connector petit, consisteix en un suport per la placa d'alimentació.

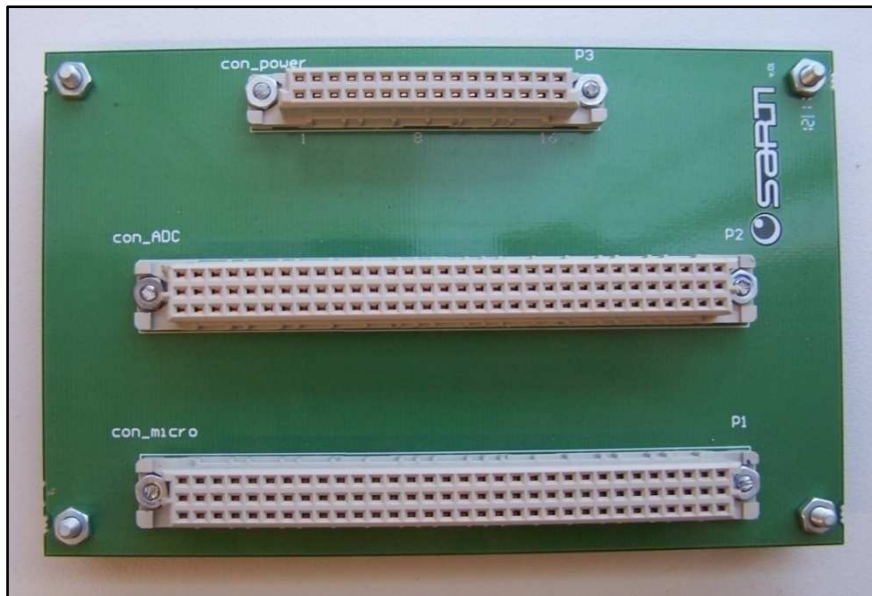


figura 2.17 Placa Backplane per unir les tres plaques

3. PROGRAMA DE CONTROL

A grans trets, l'estructura genèrica que segueix l'aplicació de control desenvolupada consisteix en una inicialització dels diferents registres del sistema i dels seus propis perifèrics [4]. Un cop inicialitzada la part general, passem a la interacció de l'usuari amb l'aplicació per mitjà d'un menú de configuració i així poder configurar les diferents parts més específiques que més endavant s'explicaran. I finalment, quan l'usuari ho ha configurat tot, s'inicia la adquisició de les dades. Aquesta darrera part la podem dividir en dos grans parts, l'adquisició de les dades per mitjà d'una interrupció i la compressió de les dades que ho realitza en els instants que l'aplicació no està adquirint dades.

El que volem arribar a obtenir, és una aplicació realitzada en C, que ens adquireixi dades sísmiques, que ens permeti adquirir-les de forma continua sense la pèrdua de cap valor, i juntament amb la informació del temps transcorregut en l'adquisició de les dades. En la figura 3.1, podem veure un esquema que representa els passos generals en que es divideix l'aplicació.

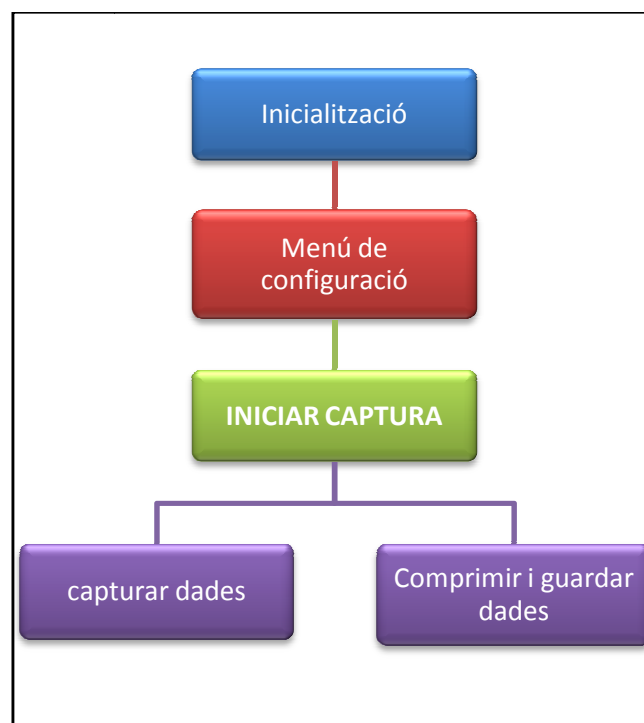


figura 3.1 Esquema general de l'estructura de l'aplicació

Per algunes d'aquestes parts de l'esquema de la figura 3.1, el microprocessador es comunica amb diferents perifèrics i parts del sistema. En la part del menú de configuració, és necessari la connexió del sistema a un ordinador per mitjà d'un port sèrie, el qual a de tenir instal·lat una aplicació per comunicar-se pel port com és l'HyperTerminal de Microsoft. En la part de la captura de dades, és on es comunica el microprocessador amb la placa Analògic-Digital per mitjà del protocol SPI, i per tant necessita estar connectat a aquest mòdul. I per acabar, la part de comprimir i guardar utilitza la targeta compact flash, on residiran les dades fins que l'usuari procedeixi a descarregar-les.

3.1. FLUX DE L'APLICACIÓ

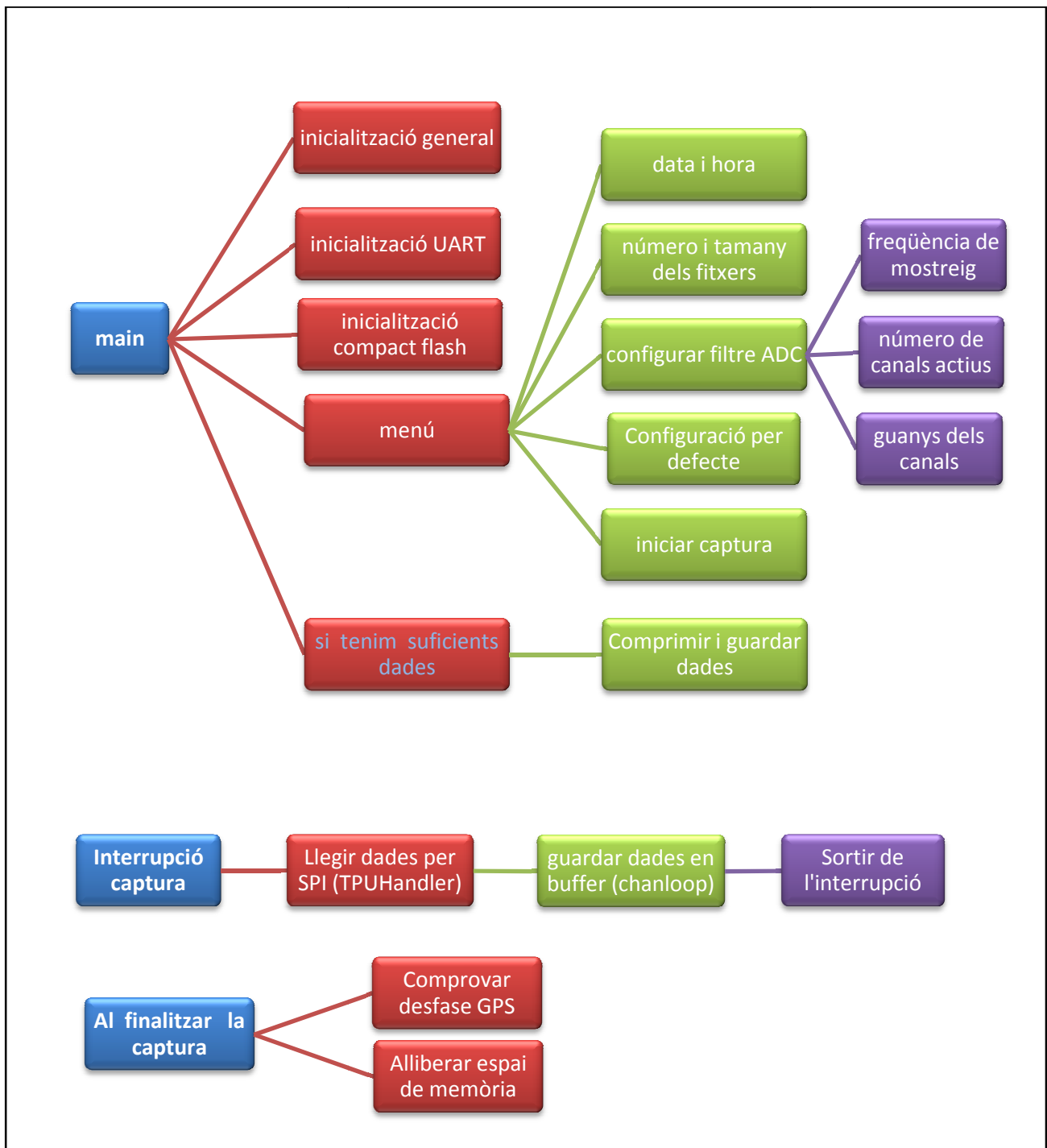


figura 3.2 Esquema general del flux de l'aplicació

Per tal de desenvolupar el programa han estat creades una sèrie de funcions que és detallaran en els següents apartats i que es poden agrupar en els següents apartats:

- Funcions bàsiques i configuració
- Comunicació i configuració de la placa analògica-digital
- Compresió de les dades

- Emmagatzematge de les dades
- Funcions pel sistema de fitxers
- Menú de configuració

En cada un d'aquests apartats hi trobarem una primera taula en forma de resum de les funcions que el representen i a continuació una explicació una mica més extensa sobre el seu funcionament.

3.2. FUNCIONS BÀSIQUES I CONFIGURACIÓ

Les funcions que es descriuen a continuació, són les encarregades de configurar el microprocessador, com també funcions d'ús general, al mateix temps, hi tenim descrita la funció general “*main*”, la qual ha d'existir obligatòriament en tota aplicació en C.

FUNCIÓ	DESCRIPCIÓ
main	Funció principal del sistema, és la primera funció que s'executa
general_init	Configuració general del microprocessador i del seus registres
inicializar_uart	Configuració del port sèrie amb la configuració desitjada
uart_tx	Envia un caràcter pel port sèrie
enviar_text	Envia frases senceres (matriu de caràcters) pel port sèrie
mcf5xxx_set_handler	Assignació d'una funció per cada interrupció habilitada
asm_set_ipl	Habilitació o deshabilitació de les interrupcions segons paràmetre
mcf5xxx_irq_enable	Executa la funció <i>asm_set_ipl</i> amb valor 0 per habilitar
mcf5xxx_irq_disable	Executa la funció <i>asm_set_ipl</i> amb valor 7 per deshabilitar
DelayMiliSecs	Espera de 'x' milisegons segons paràmetre
int_DMA_timer0	Rutina a la interrupció per comptar intervals d'1 milisegon (per les funció DelayMiliSecs)
int_DMA_timer2	Rutina a la interrupció per comptar intervals de 10 milisegon (per les funcions de l'ATA)
int_IRQ3	Rutina a la interrupció per captar dades des de la placa Analògic-Digital
Low_power_mode_IN	Configuració del baix consum dels perifèrics

taula 3.1 Funcions bàsiques i configuració

int **main** (void)

Funció principal del programa que s'utilitza per iniciar l'aplicació. En aquesta funció és on es crida a les diferents funcions per configurar el sistema i per executar l'aplicació en si mateixa. Inicialment crida a les funcions de configuració com és la d'inicialitzar els diferents registres del microprocessador, la de configurar el port de comunicació sèrie, la de configurar els pins de comunicació amb la Compact Flash, i finalment accedeix a la funció de menú on es configuraran altres parts del sistema. Un cop finalitzada la configuració del sistema entrem a un bucle sense fi que correspon a l'execució normal de l'aplicació.

Dins d'aquest bucle existeixen tres possibles casos, el primer cas és que no passi res i que el sistema estigui en espera a que entri una interrupció per adquirir dades, el segon cas correspon a que tinguem el buffer de dades preparat i que el podem anar a comprimir i guardar cridant a una funció de compressió, i per acabar, el tercer cas només s'executarà al finalitzar l'adquisició, cas en el que s'alliberarà espai de memòria i parará l'adquisició.

void **general_init** (void)

En aquesta funció és on realitzem la configuració de diferents registres del sistema. A continuació llistem les diferents parts a les que ataquem en aquesta funció amb una breu descripció del que consisteixen:

- *Configuració pel “limp mode”*. Mode de funcionament que redueix la freqüència de funcionament a nivells baixos.
- *Configuració dels ports d'entrada i sortida*. Configuració dels ports de entrada i sortida de propòsit general (GPIO). En aquest punt només es configuren els corresponents a ús general.
- *Configuració placa seascan per seleccionar la freqüència de mostreig*. Per mitjà de tres pins de sortida connectats a la placa de rellotge, es configura la freqüència de mostreig que tindrà la placa Analògic-Digital.
- *Desactivar elements del microporocessador per reduir-ne el consum*. Desactivació de diferents elements del microprocessador que no s'utilitzen per tal de reduir-ne el consum general del sistema.
- *Configuració del EPORT per habilitar les interrupcions*. Configuració de la interrupció d'entrada per ser utilitzada en l'adquisició de dades, comunicació entre el microprocessador i la placa d'adquisició analògic-digital
- *Configuració de la RTC*. Indicar el microprocessador quina freqüència estem aplicant a l'entrada corresponent al senyal de la RTC (*real timeclock*)
- *Configuració dels temporitzadors (DMA Timer)*. A partir de la freqüència dels sistema, es configuren 3 temporitzadors diferents, dos de propòsit general i un específic per la lectura i escriptura en la compact flash. Els temporitzadors programats són de 1 ms, 10 us i 10 ms, on el darrer d'aquests correspon al que anteriorment havíem definit per la comunicació amb la compact flash.
- *Configuració de les interrupcions del microprocessador*. En aquest punt es configuren totes les interrupcions del microprocessador, no només les que gestiona el EPORT, sinó que també la resta de les que disposa. El que realment fem consisteix en habilitar les interrupcions desitjades, donar prioritats a aquestes i definir quina funció haurà d'accedir el sistema quan entri cada una de les interrupcions.

void **inicializar_uart** (char canal, unsigned int systemClockKHz, unsigned int baudRate)

La inicialització de la uart consisteix bàsicament en declarar els corresponents pins per aquest ús i definir els diferents paràmetres del protocol de comunicació sèrie (sense paritat, 8bits de dades, sense control de flux i 1 bit de stop). Per definir la velocitat de transmissió del port, a partir de la variable d'entrada *baudrate* es realitza un càlcul per definir els valors a introduir en el corresponent registre.

$$\text{Valors del registre} = \frac{\text{Rellotge del sistema}}{(32 * \text{velocitat de transmissió})}$$

- *canal*: Seleccionar el número del port sèrie que es vol utilitzar per la comunicació.
- *systemClockKHz*: Freqüència de rellotge del sistema per tal de poder calcular i configurar correctament la velocitat de transmissió del port sèrie.
- *baudRate*: Velocitat de transmissió desitjada pel port sèrie.

void **uart_tx**(char canal, char datoTx)

Funció per enviar un caràcter pel port sèrie. El que realitza bàsicament és una comprovació de la disponibilitat del port de sortida i posteriorment un enviament de caràcter escrivint aquest en el registre corresponent.

- *canal*: Seleccionar el número del port sèrie que es vol utilitzar per la comunicació.
- *datoTX*: Caràcter a transmetre pel port sèrie de tipus Byte.

void **enviar_texto** (char canal, char dades[])

Aquesta funció realitza la mateixa finalitat que l'anterior, però en aquest cas en lloc d'enviar un sol caràcter, envia una sèrie de caràcters situats en una matriu passada com a paràmetre a la funció.

- *canal*: Seleccionar el número del port sèrie que es vol utilitzar per la comunicació.
- *datoTX*: Conjunt de caràcters a transmetre pel port sèrie de tipus matriu de Bytes.

void **mcf5xxx_set_handler** (int vector, void (*handler) (void))
asm void **asm_set_ipl**(int)
void **mcf5xxx_irq_enable** (void)
void **mcf5xxx_irq_disable** (void)

Aquestes quatre funcions són específiques del microprocessador, i per tant ja estaven programades. La primera de totes (*mcf5xxx_set_handler*) correspon a la funció que es crida per definir quina funció ha d'executar el microprocessador quan entri una interrupció, és a dir que vincula cada una de les interrupcions amb la seva funció corresponent i s'ha de cridar tantes vegades com interrupcions habitem al sistema.

A la segona funció (*asm_set_ipl*), correspon a una funció escrita en ensamblador, que pel cas que ens correspon a nosaltres, habilitarà les interrupcions si li passem com a paràmetre un 0, i les desactivarà si li passem un 7. Aquesta funció és utilitzada en les dues darreres funcions (*mcf5xxx_irq_enable* i *mcf5xxx_irq_disable*), i el que realitzen és una crida a la funció ensamblador amb el corresponent paràmetre a 0 o 7.

- *vector*: Número associat a la interrupció
- (**handler*): Nom de la funció que s'ha d'executar per atendre a la interrupció

void **DelayMiliSecs** (unsigned int milisec)
__declspec(interrupt) void **int_DMA_timer0** (void)

Pel cas en que necessitem un retard en el sistema, també s'ha dissenyat una funció (*DelayMiliSecs*) que activa la interrupció del temporitzador 0 prèviament programat, per tal de que passi el temps desitjat abans de continuar. Llavors, un cop passat aquest temps, el temporitzador es torna a parar i surt de la funció. Mentre el sistema està en espera a que passi el temps desitjat, el que està realitzant la concretament en l'execució de l'aplicació, és l'increment d'una variable cada milisegon i comparar aquest valor amb el que l'usuari li ha passat com a paràmetre. Com que la interrupció (*int_DMA_timer0*) es cridada cada milisegon, aquest és el valor mínim que pot tenir la interrupció d'espera.

- *Milisec*: Número de milisegons per realitzar un retard en el sistema dins de l'execució del programa

`__declspec(interrupt) void int_DMA_timer2 (void)`

Per tal de poder escriure en el sistema de fitxer de la compact flash, s'ha utilitzat el protocol "FAT", i per aquest motiu ha estat necessària l'implementació d'un altre temporitzador, que en un dels seus usos consisteix en el control del temps màxims en l'escriptura del sistema per tal que aquest no es quedi bloquejat, com un *watchdog*. L'altre funcionalitat que també té aquest temporitzador, consisteix en el simple ús d'espera d'un temps predefinit. Aquest temporitzador està configurat per treballar a intervals de 10 milisegons, per tant aquest serà el temps més petit que se li podrà demanar a la funció.

A diferència de l'altre temporitzador explicat anteriorment, aquest no està compost per una funció que activa el temporitzador i que està en espera que passi el temps, si no que a partir de la mateixa filosofia, realitza aquesta funció dins del codi extern, tant la part d'arrancar el temporitzador, com la part de control de la variable de comptatge. L'avantatge d'aquest temporitzador respecte a l'altre consisteix en que en aquest cas dins del bucle de espera, els sistema realitza altres funcions i en el cas de la funcionalitat de *watchdog* no acaba de comptar si a finalitzat la tasca que estava realitzant. I per acabar, un altre diferència respecte el primer temporitzador és que en aquest cas la variable es decrementa, fet que en el cas anterior era a l'inrevés.

`__declspec(interrupt) void int_IRQ3 (void)`

Aquesta funció és la que dona resposta a la rutina de servei a la interrupció principal del sistema. És la interrupció que s'activarà a la freqüència de mostreig del sistema. Aquest valor és configurable segons el menú de configuració i dependrà directament d'una de les sortides físiques de la placa d'adquisició Analògic-Digital. Per la funcionalitat que té aquesta funció i pel requeriment de no poder perdre cap de les mostres, aquesta interrupció queda configurada com la més prioritària de totes.

Quan aquest interrupció és executada, la seva funcionalitat és limita a cridar a una funció denominada *TPUHandler*, on és realitzen les corresponents tasques d'adquisició. Més endavant explicarem en que consisteix aquesta funció i totes les que precedeixen a aquesta.

`void Low_power_mode_IN (char mode)`

Amb aquesta funció s'aconsegueix canviar el mode de treball del microprocessador, des del mode normal (*RUN*) o els modes de baix consum (*WAIT*, *DOZE*, *STOP*). Amb aquesta funció es configuren alguns registres per definir els perifèrics que es desitgen configurar en baix consum i altres paràmetres com el nivell de la interrupció que ens permetrà sortir d'aquest estat. Per reduir els consums amb aquesta funció, el que es fa concretament és deshabilitar els rellotges dels diferents mòduls del sistema en funció del mode a treball.

- *mode*: Paràmetre per indicar en quin mode es vol ficar a treballar el sistema. Existeixen quatre opcions que són el *RUN*, el *WAIT*, el *DOZE* o el *STOP*

3.3. COMUNICACIÓ I CONFIGURACIÓ DE LA PLACA ANALÒGIC DIGITAL

En aquest apartat podem trobar-hi les funcions destinades a la configuració del mòdul analògic a digital, com també les funcions destinades a l'adquisició de les dades. Per realitzar la comunicació amb aquest mòdul, s'utilitza el protocol de comunicació SPI, (protocol explicat breument en l'apartat 1.4).

FUNCIÓ	DESCRIPCIÓ
dspl_tx	Envio des de 4 fins a 16 bits pel port SPI
dspl_config	Configuració del port SPI
dspl_rx_data	Recepció de dades pel port SPI
filter_config	Configuració de diferents opcions de la placa Analògic a Digital segons paràmetre
AD_config_default	Configuració de les opcions de la placa analògic a digital segons valors per defecte
AD_config	Configuració de les opcions de la placa analògic a digital segons valors passats com a paràmetre
buf_setup	Configuració dels buffers de sistema per la adquisició de dades
TPUHandler	Captació de dades de temps cada segon i crida a la funció <i>chan_loop</i>
chan_loop	Emmagatzematge de les dades en els buffers i crida a les funcions per comprimir dades si és necessari

taula 3.2 Funcions de comunicació i configuració de la placa analògic a digital

void **dspl_tx**(int tx_data, char cont, char ctas)

Aquesta funció s'encarrega d'escriure pel port SPI les dades ubicades en el paràmetre *tx_data*. Per tal de realitzar aquesta funció, el que realment realitza la funció és una escriptura en un dels registres del microprocessador, un registre compost per 32 bits que podem diferenciar en dos tipus de dades. Un primer grup, compost pels 16 bits de menor pes, està destinat a les dades que realment es desitgen enviar, i el segon grup de bits que corresponen als 16 de major pes del registre, estan destinades a la configuració del propi enviament.

Al realitzar la crida aquesta funció, no s'espera que aquesta retorni cap mena de valor, però sí que son necessaris que se li passin tres variables com a paràmetres:

- *tx_data*: Com ja s'ha comentat anteriorment, aquest paràmetre correspon als 16 bits de dades que es desitja enviar. Aquest paràmetre està compost per 16 bits però no és necessari l'enviament de tots els bits, això és per què per configuració d'un altre registre, és pot configurar el número de bits que es voldrà enviar en cada transmesa. Aquest valor podrà oscil·lar entre 4 i 16bits.
- *cont*: Aquest paràmetre indica si immediatament després de cada enviament, se'n ha de realitzar un altre o no, és a dir que si per exemple en la nostra aplicació volem fer un enviament de 16 bits 8 vegades seguides, cridarem a la funció 7 vegades amb aquest paràmetre a '1' i una darrera vegada amb el paràmetre a '0'.
- *ctas*: En la configuració del port SPI, podem tenir fins a 8 configuracions de comunicació diferents per tal de poder tenir fins a 8 esclaus diferents. Per tal de seleccionar quina d'aquestes configuracions utilitzarem en l'enviament, seleccionarem el que prèviament haguem configurat. En el nostre cas, com que només tenim un mestre i un esclau, aquest paràmetre el tindrem sempre per la configuració destinada a l'esclau 0.

void **dspl_config**(char fmsz, char cpol, char cpha)

La funció *dspl_config*, correspon a la de configuració del protocol SPI. En aquesta s'hi escriuen fins a quatre registres diferents. El primer d'aquests registres és per la configuració del multiplexat dels pins d'entrada i sortida. Com molts dels pins del microprocessador, els pins destinats a la comunicació del port SPI, poden ser utilitzats per diferents funcionalitats, com podria ser per exemple de propòsit general. En el nostre cas,

configurem tots els pins pel protocol SPI tret dels destinats a habilitar més d'un esclau, degut a que només en tenim un i per tant no són necessaris.

El registre **DPSI_MCR** (*module configuration register*), és el segon dels registres que configurem dins d'aquesta funció. Aquest està pensat per la configuració de diferents paràmetres associats directament amb les operacions de funcionament del SPI. Una de les característiques més rellevants que es configuren dins d'aquesta funció, és el bit que determina que el microprocessador serà el mestre en la comunicació, degut ha que també podria arribar a treballar com esclau.

En el cas del tercer registre que escrivim en aquesta funció, es correspon al **DSPI_CTARn** (*clock and transfer attributes register*). En el microprocessador existeixen fins a 8 registres com aquest, que corresponen a les diferents configuracions que podem tenir emmagatzemades. En el nostre cas, com que només tenim un esclau, només accedim al primer registre de configuració. En aquest seleccionem entre d'altres coses la freqüència que tindrà el nostre rellotge de comunicació SPI, el número de bits que tindrà cada paquet enviat, l'estat en que es trobarà el pin corresponent al clock quan no s'utilitzi i també l'ordre en que s'enviaran les dades, és a dir si primer seran les de major pes o les de menor pes. Per tal de configurar el rellotge de comunicació, seguirem la següent equació:

$$SCK \text{ baud rate} = \frac{f_{SYS/2}}{PBR} \times \frac{1 + DBR}{BR}$$

SCK baud rate: Freqüència de comunicació del SPI

f_{SYS}: Freqüència principal del microprocessador

PBR: Tenim quatre valors possibles [2 3 5 7]

DBR: Només pot tenir el valor '1' o '0'

BR: Fins a 16 valors disponibles segons la taula 3.3.

BR	Baud Rate Scaler Value	BR	Baud Rate Scaler Value
0000	2	1000	256
0001	4	1001	512
0010	6	1010	1024
0011	8	1011	2048
0100	16	1100	4096
0101	32	1101	8192
0110	64	1110	16384
0111	128	1111	32768

taula 3.3 Valors disponibles pel registre BR del rellotge SPI

Per últim, el darrer dels registres que accedim, correspon al **DPSI_SR**. Aquest registre indica alguns dels estats en que es troba la comunicació SPI, com és entre d'altres si la transmissió s'ha realitzat, o si algun buffer ha tingut un desbordament o que estigui ple. En tot cas, en aquests registres només hi podem escriure un '1' per resetejar-ne el seu estat. Si hi escrivim un '0', el sistema no realitzarà cap acció. En la nostra funció, el que realitzem és la escriptura a '1' a tots els bits accessibles per tal de començar des de zero.

Els paràmetres que li passem com a referència a la funció són els que a continuació nomenem i ataquen únicament al registre CTAR per la configuració del rellotge i les característiques de la transmissió.

- *fmsz*: Número que indica la quantitat de bits que es transmetran per cada trama. Aquest valor pot oscil·lar entre 4 i 16. En el cas de l'aplicació actual, s'utilitzen dos tipus de comunicació, un per la configuració de la placa analògic a digital i un altre per la transferència de les dades. En el cas de la configuració, la comunicació és realitza amb 8 bits, i en el cas de la comunicació amb les dades aquesta és fa amb 16 bits.
- *cpol*: Aquest paràmetre pot tenir dos valors possibles ('1' o '0'). El que es configurem amb aquest, és l'estat en que es troba el rellotge de la comunicació SPI en estat inactiu. Pes un estat baix és un '0' i per un estat alt és '1'.
- *cpha*: Per indicar el mode de funcionament de les dades respecta el rellotge utilitzem aquest registre. En aquest indicarem si volem que el sistema agafi les dades pel flanc de pujada i les modifiqui en el de baixada (ficant un '0') o les modifiqui en el de pujada i agafi les dades en el de baixada (amb un '1').

void **dspl_rx_data** (int envios)

El mètode que utilitza el nostre microprocessador per rebre les dades per SPI, consisteix en realitzar prèviament un enviament amb el registre de dades en blanc i amb la mateixa senyal de rellotge, pel canal de recepció, arriben les dades de lectura que s'emmagatzemen en un buffer de tipus FIFO. En la part que ens interessa a nosaltres, hem de realitzar una lectura de 4 canals, corresponents als tres del sismòmetre i un del hidròfon, i cada un d'aquests canals disposa de 32 bits d'informació, és a dir que tenim un total de 128 bits a llegir cada instant de mostreig. Com que la capacitat màxima de comunicació del nostre sistema és de 16 bit per cada enviament, haurem de realitzar 8 lectures del port d'entrada.

Llavors simplement, aquesta funció crida a la que hem explicat anteriorment (*dspl_tx*) per realitzar el número de enviaments desitjats segons el paràmetre. Per fer-ho, s'utilitza un bucle for de manera que s'envien totes les trames menys una que es realitza a fora del bucle, per tal d'indicar que hem acabat la transmissió. Un cop surt d'aquesta funció, ja tenim les dades preparades en un buffer que rep el nom de DSPI_RXFRn per tal de ser llegides.

- *envios*: Número de trames a llegir seguides, on cada trama és del número de bits que s'hagi configurat amb la funció *dspl_config*.

void **filter_config** (char reg, char data_H, char data_M, char data_L)

Aquesta funció està composta a grans trets, per la comunicació amb la placa analògic a digital per mitja del protocol SPI, i per tal de configurar les diferents opcions de la placa d'adquisició segons els paràmetres desitjats. La funció està creada per tal de donar suport a la funció *AD_config_default* i *AD_config* i així reunir en una sola funció els diferents registres a accedir. El seu funcionament bàsic consisteix en una sèrie d'enviaments per SPI en funció del registre al qual es vol accedir i amb les dades adequades.

- *reg*: Adreça del registre al qual volem accedir de la configuració de la placa analògic a digital.
- *data_H*: Dades dels 8 bits de major pes per escriure en el registre
- *data_M*: Dades dels 8 bits del mig per escriure en el registre
- *data_L*: Dades del 8 bits de menor pes per escriure en el registre

void **AD_config_default** (void)

Aquesta funció és la que cridem per realitzar la configuració de la placa analògic a digital amb els paràmetres configurats per defecte. Com hem comentat en la descripció de l'anterior funció, aquí fem ús de la funció *filter_config* per enviar les trames de configuració. La funció segueix tres parts en la seva execució, en primer lloc es configura la comunicació SPI per enviar i rebre 8 bits per trama fent ús de la funció *dsapi_config*, en segona instància es realitza una sèrie d'enviaments amb el mateix protocol i fent ús de la funció que hem comentat anteriorment (*filter_config*) per configurar la placa de forma predeterminada, i finalment es procedeix a configurar la comunicació SPI per enviar i rebre 16bits per trama, configuració que ens servirà per la transmissió de les dades d'adquisició.

Entre altres dades a configurar, amb aquesta funció indiquem els següents paràmetres per defecte:

- Treballarem amb els quatre canals d'adquisició de la placa
- Treballarem a 250Hz de mostreig
- Aplicarem una guany de '1' a tots els canals

void **AD_config**(char config_H,char config_M,char config_L,char filt_cfg_H,char filt_cfg_M,char filt_cfg_L)

Al igual que a la funció que s'ha explicat anteriorment, aquesta funció és exactament igual a la que s'ha explicat anteriorment (*AD_config_default*) però en aquest cas la configuració no és per defecte, si no que les dades que s'han de ficar en els diferents registres venen determinades pel programa quan es crida a la funció. Aquestes dades que li passem com a paràmetre queden generades amb les funcions del menú de configuració, que a partir de les dades desitjades, es generen les variables necessàries.

Per mitjà de les funcions del menú que explicarem en l'apartat 3.7, donarem valors a aquests paràmetres amb el que podrem configurar els següents paràmetres:

- Freqüència de mostreig (125, 250, 500, 1000 Hz)
- Número de canals a utilitzar (1, 2, 3 o 4 canals)
- Guany pel canal 1 de l'hidròfon (x1, x2, x4, x8, x16, x32, x64)
- Guany pel canal 2,3,4 del Geòfon (x1, x2, x4, x8, x16, x32, x64)
- *config_H*: Dades dels 8 bits de major pes per escriure en el registre CONFIG de la configuració del Hardware.
- *config_M*: Dades dels 8 bits del mig per escriure en el registre CONFIG de la configuració del Hardware.
- *config_L*: Dades dels 8 bits de menor pes per escriure en el registre CONFIG de la configuració del Hardware.
- *filt_cfg_H*: Dades dels 8 bits de major pes per escriure en el registre FILT_CFG de la configuració del filtre.
- *filt_cfg_M*: Dades dels 8 bits del mig per escriure en el registre FILT_CFG de la configuració del filtre.
- *filt_cfg_L*: Dades dels 8 bits de menor pes per escriure en el registre FILT_CFG de la configuració del filtre.

void **buf_setup** (void)

En aquesta funció és on es reserva l'espai de memòria pels diferents buffers temporals que s'utilitzaran en l'aplicació, i per inicialitzar els punters de lectura i escriptura a aquests buffers.

void **TPUHandler** (void)

Quan es cridava a la rutina d'atenció a la interrupció provocada pel mòdul d'adquisició de dades, el que es realitzava en aquesta era una crida a aquesta funció. El que és realitza el primer cop que s'executa aquesta correspon al descart de les primeres dades, per eliminar un possible error en les dades. En el cas de que aquestes primeres dades estiguin eliminades, el que realitza la funció consisteix en una lectura per SPI de les dades adquirides per mitja de la funció *dspi_rx_data* i en el suposat cas que hagi passat 1 segon, es pren una referència de temps. Finalment, un cop tenim les dades en el buffer de lectura del SPI, es crida a la funció *chan_loop* per utilitzar les dades i ficar-les en el corresponent buffer d'adquisició.

void **chan_loop** (void)

Com hem comentat en la descripció de la funció anterior, en aquesta funció llegim les dades del buffer de lectura del SPI per guardar-les en el corresponent buffer de treball. Dins cada paquet de dades rebudes de cada canal, a part de les dades corresponents a l'adquisició, hi trobem les dades que ens indiquen l'estat, com és per exemple el canal del que corresponen les dades. Llavors en aquesta funció eliminem aquestes dades que no es corresponen a la informació pròpia de l'adquisició i guardem la informació de l'adquisició en el corresponent buffer. Al mateix temps quan passa un segon de la adquisició, es realitza una comprovació de les dades per assegurar-nos de que no s'han perdut dades.

3.4. COMPRESSIÓ DE LES DADES

En aquest apartat tenim les funcions destinades a la compressió de les dades adquirides. A partir de les dades obtingudes de la placa d'adquisició, aquestes es comprimeixen per obtenir més espai a la Compact Flash. Alhora que es guarden les dades, també es guarda la informació del temps. De les cinc funcions que hi ha dins d'aquest grup, hi ha una funció principal que gestiona el flux del procés de compressió i les altres quatre que es distribueixen les diferents tasques a realitzar.

FUNCIÓ	DESCRIPCIÓ
pgmtime	Generació del valor de temps actual
store_data	Guardar dades a la memòria Flash
make_time_entry	Crear paquets de temps relacionats amb cada segon de adquisició i guardar en Flash
make_data_entry	Comprimeix les dades en paquets i els guarda a la memòria Flash
pc_buf_to_pac	Comprova les dades i crida a les altres funcions per comprimir i guardar els paquets de temps i de dades.

taula 3.4 Funcions per la compressió de les dades

PTIME *pgmtime (time_t t, PTIME *pt)

Aquesta funció ens permet generar la trama de temps actual, és una funció destinada a complementar la funció *make_time_entry*, la qual serà l'encarregada de cridar-la per generar els paquets de temps. Bàsicament la seva funcionalitat consisteix en convertir les dades en format de segons des de 01/01/1970 a les 00:00:00h al format de dia, hora i segons.

- *t*: Número de segons transcorreguts des del 01/01/1970 a les 00:00:00h, aquest paràmetre el podem trobar dins d'un dels buffers d'adquisició.
- *pt*: Punter de la variable on es guardaran les dades referents a la conversió realitzada dins de la funció.

voidstore_data(BYTE pt[PAC_BYTES])

Aquí serà on s'emmagatzemaran les dades a la Compact Flash. En aquesta funció crearem el corresponent fitxer cridant a la funció *f_open*, com també cridarem a la funció *f_write* que ens escriurà a la compact flash. El nom que generem pel nostre fitxer correspon al número del dia de l'any més l'hora i els minuts actuals, i finalment seguit per l'extensió ".ii" llavors el format quedaria d'aquesta forma "DDHMMM.ii".

- *pt*: Punter on es troben les dades per ser emmagatzemades en la Compact Flash

void make_time_entry (struct ring_buf *this_sec, time_t t, short s)

En aquesta funció creem els paquets de temps relacionats amb cada segon de l'adquisició i guardem a la Compact Flash aquesta informació cridant a la funció *store_data*.

- *this_sec*: Aquesta variable determina si les dades que es pretenen guardar a continuació consistiran en les dades pròpiament de l'adquisició (aportant la informació referent el temps associat al segment de dades), o per un altre costat correspon a una pèrdua de dades (valor NULL en la variable).
- *t*: Marca de l'instant que s'ha trobat una pèrdua de dades. Aquesta variable només s'utilitza en els casos de pèrdues de dades.
- *s*: Marca que ens indica el número de la mostra que s'ha perdut.

void make_data_entry (register unsigned long *dptr, short count, short channel, short gain, short wordlength)

Aquesta funció és l'encarregada de comprimir les dades i de crear els paquets que s'emmagatzemaran en la Compact Flash. Per comprimir aquestes dades, s'eliminen una sèrie de bits corresponents a zeros a l'esquerra fent que les dades ocupin menys espai en la memòria. A part també és creen les corresponents capçaleres amb la informació pertinent del canal, guany i longitud de les dades.

- *dptr*: Punter amb l'adreça on estan situades les dades amb la informació a comprimir i guardar.
- *count*: Número de dades que es poden comprimir dins d'un paquet de dades.
- *channel*: Número del canal al qual corresponen les dades que s'estan processant
- *gain*: Guany que s'ha utilitzat en l'adquisició de les dades.
- *wordlength*: Longitud de les dades a emmagatzemar.

`int pc_buf_to_pac (void)`

Per acabar el grup de funcions de compressió de dades, tenim la funció principal a la qual es crida externament per realitzar la compressió de les dades. El procés que realitza es compon bàsicament de la lectura de les dades a cada segon, el càlcul de quantes dades de la mateixa dimensió es poden comprimir en paquets i un cop calculat, cridar a la funció *make_data_entry* per comprimir-los. A part d'aquest procés, també realitza un control de la pèrdua de dades, que en cas afirmatiu, el programa avisaria amb un missatge per pantalla. I quan aquesta finalitza del tot, simplement retorna un '1' per indicar-ho.

3.5. EMMAGATZEMATGE DE LES DADES

El nostre microprocessador incorpora una controladora pel port ATA, on hi tenim connectada la targeta d'emmagatzematge Compact Flash. El problema que ens trobem és que en baixar la freqüència de treball del microprocessador, aquesta controladora és un dels elements que deixen de treballar en el sistema. Per aquest motiu, a partir de programari lliure, s'han afegit al sistema unes funcions que ens permeten escriure en una targeta de memòria a partir del sistema de fitxers FAT. Aquestes funcions es divideixen en dos grups diferents, les de nivell del sistema de fitxers i les de baix nivell. A la següent figura 3.3 hi podem trobar un esquema amb aquest dos grups de funcions.

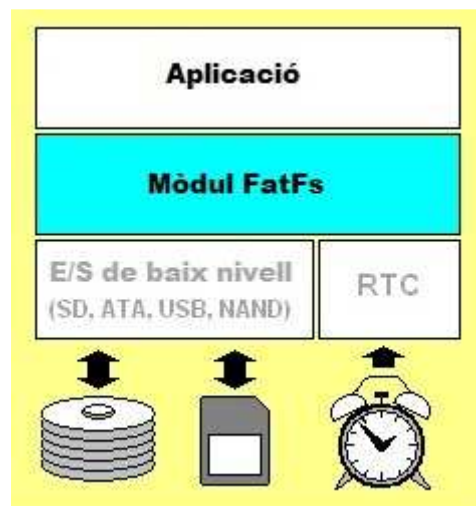


figura 3.3 Esquema de l'estructura que segueix l'accés a la CF

Les funcions corresponents al nivell del sistema de fitxers les explicarem en el següent apartat (3.6 Funcions pel sistema de fitxers). Les funcions de baix nivell, són les que s'expliquen en aquest apartat les quals permeten llegir i escriure en la Compact Flash a nivell baix.

FUNCIÓ	DESCRIPCIÓ
inicializar_ATA	Configurar els pins com a propòsit general per ser utilitzat per la comunicació de la Compact Flash
read_ata	Llegir registres de control de la Compact Flash
write_ata	Escriure en els registres de control de la Compact Flash
read_part	Llegir un registre concret amb informació de la Compact Flash
wait_data	Esperar a la Compact Flash estigui disponible
disk_initialize	Inicialització dels diferents registres del sistema i dels registres de la Compact Flash
disk_read	Llegir el número de sectors de la Compact Flash desitjat
disk_write	Escriure el número de sectors de la Compact Flash desitjat
disk_ioctl	Llegir dades de configuració de la Compact flash o de la configuració de l'aplicació
get_fattime	Retorna l'hora actual del sistema pel sistema de fitxers Fat32

taula 3.5 Funcions per l'emmagatzematge de les dades

void **inicializar_ATA** (void)

En aquesta funció es configuren els registres corresponents a l'ús dels pins utilitzats en la comunicació amb la Compact Flash. Aquests pins han de ser configurats com a propòsit general per tal de reproduir les accions de la controladora de l'ATA amb funcions externes al microprocessador. Els registre a configurar estan compostats per 5 registres, dos de dades i tres de configuració.

En el cas dels registres corresponents a les dades, s'utilitzen els 8 bits de cada un d'aquests, per tant, tenim un bus de dades de 16bits en total. En canvi, en el cas dels registres corresponent a la configuració ho tenim repartit en tres registres diferents encara que solament s'utilitzen un total de 8 bits de configuració. En un d'aquest registre hi tenim el bit de *reset*, en un altre hi tenim dos bits per definir si es pretén escriure o llegir, i en el tercer dels registres hi tenim els bits de configuració com és el *chip select* o els bits que determinen el tipus d'accés (accés per transferir dades o accés per accedir a la configuració).

Registres de dades:

DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
DD15	DD14	DD13	DD12	DD11	DD10	DD9	DD8

Registres de control:

-	-	-	-	-	RESET	-	-
-	-	DIOR	DIOW	-	-	-	-
-	-	-	CS1	CS0	DA2	DA1	DA0

static BYTE **read_ata** (BYTE reg)

Aquesta funció ens serveix per llegir dels registres interns de la Compact Flash. Per realitzar aquesta acció, li hem de passar el valor de l'adreça del registre al qual volem accedir i per mitja dels registres de control podem llegir la informació desitjada. Dos exemples dels registres a accedir són el *status* per saber-ne l'estat actual de la Compact Flash o el registre d'*error* per saber si existeix algun error. La funció retorna el valor de la lectura realitzada.

- *reg*: Adreça del registre al qual volem accedir.

static void **write_ata** (BYTE reg, BYTE dat)

Al igual que la funció anterior, en aquesta funció pretenem accedir als registres interns de la Compact Flash per tal de escriure el valor desitjat. Com a paràmetres li passem la informació referent de on volem escriure i del valor que volem escriure. En aquest cas, la funció no retorna cap valor, però en algunes accions per les quals requereixen de la utilització d'aquesta funció, poden estar complementades per la de llegir amb la finalitat de comprovar un correcte accés al registre per mitjà del registre *error*.

- *reg*: Adreça del registre al qual volem escriure o accedir.
- *dat*: Informació que volem escriure en el registre especificat.

static void **read_part** (BYTE *buff, BYTE ofs, BYTE count)

En aquesta funció ens encarreguem de llegir informació referent a la configuració de la Compact Flash. Correspon a l'accés d'un byte o d'una sèrie de bytes amb informació determinada segons la posició desitjada dins d'un registre.

- *buff*: Punter on hi escriurà la informació obtinguda en la lectura del registre.
- *ofs*: Offset del *word* (2 Bytes) a partir del qual ha de començar a llegir
- *count*: Número de *words* a llegir corresponents a la informació desitjada.

static BOOL **wait_data** (void)

Esperar a que la Compact Flash estigui disponible per seguir amb la transferència. En el cas de passar més de 10 segons, saltaria fora del bucle i marcaria error en l'accés a la targeta. En cas de estar disponible abans d'aquest temps, el sistema torna al punt on estava quan s'havia cridat a la funció. El paràmetre que la funció retorna és un '1' en cas de ser tot correcte o un '0' en cas contrari.

DSTATUS **disk_initialize** (BYTE drv)

Aquí és on inicialitzem i configurem la Compact Flash per començar a treballar amb ella. En primer lloc cridem a la funció *inicialitzar_ATA* per tal de configurar els pins com a propòsit general, amb la finalitat de utilitzar-los en el procés d'accés a la Compact Flash. Posteriorment realitzem un *reset* a nivell de software de la memòria i comprovem el seu estat, i finalment configurem la targeta per ser utilitzada en el mode de funcionament desitjat per mitja de la funció *write_ata* per accedir al registre de configuració.

DRESULT **disk_read** (BYTE drv, BYTE *buff, DWORD sector, BYTE count)

Amb la finalitat d'accedir a la informació de la Compact Flash, utilitzem aquesta funció. En primer lloc escriu en els registres corresponents de la Compact Flash (amb la funció *write_ata*) per tal de preparar-la per poder llegir tants sectors com es desitgin. En segon lloc, es procedeix a llegir els 512Bytes del primer sector i en el cas de voler llegir un altre sector, es comprovarà l'estat de la Compact Flash i es llegiran els 512 Bytes del següent sector. Finalment, si tot ha estat correcte es retorna un '0', en cas d'haver-hi un error es retornaria el número corresponent a l'error.

- *drv*: Número del dispositiu físic a accedir (en el nostre cas només en tenim 1 i aquest paràmetre valdrà sempre 0)
- *buff*: Punter a la variable on es guardaran les dades llegides.
- *sector*: Adreça interna de la Compact Flash on volem llegir les dades.
- *count*: Número total de sector a llegir.

DRESULT **disk_write** (BYTE drv, const BYTE *buff, DWORD sector, BYTE count)

A l'hora d'escriure es segueix un procediment molt similar al seguit en la lectura de la memòria. En primer lloc s'escriuen als registres de la Compact Flash com es feia abans, però en aquest cas amb la configuració per escriure en lloc de llegir. Després s'escriuen els sectors als quals s'hi vol accedir, fent la comprovació de l'estat de la targeta de memòria entre cada una de les escriptures dels diferents sectors. I per acabar comprova que hagi acabat la darrera escriptura i retorna un '0' en cas de finalitzar correctament (com en el cas anterior, si hi ha un error, retornarà el valor numèric corresponent a l'error).

- *drv*: Número del dispositiu físic a accedir (en el nostre cas només en tenim 1 i aquest paràmetre valdrà sempre 0)
- *buff*: Punter a la variable on hi tenim les dades que s'escriuran a la memòria Compact Flash.
- *sector*: Adreça interna de la Compact Flash on volem llegir les dades.
- *count*: Número total de sector a llegir.

DRESULT **disk_ioctl** (BYTE drv, BYTE ctrl, void *buff)

Aquesta és una de les funcions que requereix el sistema de fitxers per tal de llegir dades de configuració de la Compact flash o de la configuració de l'aplicació. En aquesta crida a la funció de *read_part* per llegir la informació de la memòria o retorna els valors definits per software d'alguns paràmetres preestablerts, com per exemple la capacitat que té un sector, el qual retorna el número 512.

- *drv*: Número del dispositiu físic a accedir (en el nostre cas només en tenim 1 i aquest paràmetre valdrà sempre 0)
- *ctrl*: Paràmetre o informació a la qual es desitja accedir.
- *buff*: Punter a la variable on hi escriurem la informació obtinguda

DWORD **get_fattime** (void)

Aquesta és un altre de les funcions que requereix el sistema de fitxers per tal d'obtenir l'hora del sistema i poder-la utilitzar en el protocol del FAT. Aquesta funció no requereix cap paràmetre d'entrada degut a que llegeix la informació directament de la RTC del sistema i retorna una trama amb la informació completa de la data i hora actual. Aquesta variable que retorna, està composta per 32 bits en els quals la informació queda repartida de la següent forma:

BITS	DADA	POSSIBLES VALORS
31-25	any	0-127 (<i>organitzat des de 1980</i>)
24-21	mes	1-12
20-16	dia	1-31
15-11	hores	0-23
10-5	minuts	0-59
4-0	segons	0-29

taula 3.6 Trama per les dades de temps pel sistema de fitxers FAT

3.6. FUNCIONS PEL SISTEMA DE FITXERS

Com hem comentat en l'apartat anterior, aquestes funcions formen part del sistema de fitxers FAT [8] i al ser programari lliure, aquesta és la part de les funcions que no s'han de modificar. Les funcions que s'han explicat en l'apartat anterior, seran cridades per aquestes funcions que s'explicaran breument a continuació. De totes les funcions disponibles, només s'explicaran les funcions utilitzades en aquesta aplicació amb una breu descripció de la seva funcionalitat.

FUNCIÓ	DESCRIPCIÓ
f_mount	Muntar el sistema de fitxers Fat
f_open	Obrir un fitxer nou
f_read	Llegir d'un fitxer obert
f_write	Escriure en un fitxer obert
f_close	Tancar un fitxer

taula 3.7 Funcions pel sistema de fitxers

FRESULT **f_mount** (BYTE vol, FATFS*fs)

Funció per muntar i desmuntar el sistema de fitxers en el disc de destí, en el nostre cas en la Compact Flash. El paràmetre que aquesta funció retorna consisteix en el resultat de realitzar l'operació, un zero si és correcte i un número indicador de l'error en cas d'haver-n'hi algun.

- *vol*: Número del dispositiu físic a accedir (en el nostre cas només en tenim 1 i aquest paràmetre valdrà sempre 0)
- *fs*: Punter a una nova estructura del dades pel sistema de fitxers

FRESULT **f_open** (FIL*fp, const XCHAR*path, BYTE mode)

En aquesta funció, podrem obrir o crear un fitxer dins del sistema de fitxers. Per tal de poder escriure o llegir d'un fitxer, haurem de cridar a aquesta funció per tal de tenir el fitxer accessible. En cas de realitzar-se tot correctament, la funció ens retornarà un zero, en cas contrari ens retornarà el número equivalent a l'error.

- *fp*: Punter a la estructura del fitxer que es vol crear o que es vol obrir
- *path*: Punter a la adreça on es vol crear o obrir el fitxer i el nom que tindrà el fitxer.
- *mode*: Diferents modes en els quals es desitja treballar amb el fitxer, com per exemple podria ser un fitxer de només lectura.

FRESULT **f_read** (FIL*fp, void*buff, UINT btr, UINT*br)

Funció per llegir les dades del fitxer que prèviament s'ha d'haver obert. Al igual que en totes les funcions d'aquest apartat, la funció ens retornarà un zero si no existeix cap error en la lectura, o el número corresponent a l'error en cas contrari.

- *fp*: Punter a la estructura del fitxer on es vol accedir per llegir.
- *buff*: Punter a la variable on es guardaran les dades llegides.
- *btr*: Número de Bytes a llegir
- *br*: Punter a la variable que ens indica el número de Bytes que s'han llegit.

FRESULT **f_write** (FIL*fp, const void*buff, UINT btw, UINT *bw)

Al igual que en la funció anterior, aquesta funció en serveix per accedir a les dades, però en aquest cas és per tal d'escriure en el sistema de fitxers. Si tot és correcte ens retornarà un zero i en cas contrari el valor corresponent a l'error.

- *fp*: Punter a la estructura del fitxer on es vol accedir per escriure.
- *buff*: Punter a la variable que conté les dades que es volen escriure.
- *btw*: Número de Bytes a escriure
- *bw*: Punter a la variable que ens indica el número de Bytes que s'han escrit.

FRESULT **f_close** (FIL *fp)

La darrera de les funcions que utilitzem pel sistema de fitxers, consisteix en una funció per tancar el fitxer al qual s'ha accedit. Si no es tanca el fitxer, les dades que s'haguessin escrit en aquest es perdrien. Per aquesta funció, al igual que fins ara, ens retornarà un zero si esta tot correcte o el número corresponent a l'error en cas d'haver-n'hi algun.

- *fp*: Punter a la estructura del fitxer que es vol tancar. Aquesta estructura ha d'haver estat creada prèviament amb la funció *f_close*.

3.7. MENÚ DE CONFIGURACIÓ

Les funcions representades en aquest apartat, estan creades per realitzar una transferència de dades pel port sèrie en els dos sentits per interactuar amb l'usuari, i així poder definir adequadament les variables i registres de sistema, com també poder configurar els diferents paràmetres que es necessiten per realitzar l'adquisició de dades. Amb les diferents funcions que tenim, podrem configurar l'hora del sistema, configurar la capacitat dels fitxers d'adquisició i configurar la placa analògic a digital. A part de tot això, un cop configurat tots els paràmetres, l'usuari serà qui donarà inici a l'adquisició. En els diferents menús que tenim, les opcions que té l'usuari per interactuar estan limitades al que se li demana, és a dir que si per exemple tenim 4 opcions per elegir, el sistema no respondrà si aquest li envia un cinquena opció.

FUNCIÓ	DESCRIPCIÓ
menu	Menú principal del sistema
time_menu	Menú principal per les dades de temps
menu_fecha	Ajustar la data actual del sistema
menu_hora	Ajustar l'hora actual del sistema
flanco_GPS	Definir flanc de sincronització per l'hora del sistema
sincronizar_GPS	Sincronitzar l'hora introduïda amb el senyal de GPS
comparar_GPS	Calcular el retard de l'hora del sistema amb el senyal del GPS
storage_menu	Definir els paràmetres d'emmagatzematge
dimension_max_CF	Determina el número màxim i la dimensió dels fitxers segons les dimensions de la Compact Flash
menu_dimension_fichero	Definir les dimensions dels fitxers a crear
menu_numero_fichero	Definir el número màxim de fitxers que generarà
config_placa_adc	Configuració dels diferents paràmetres de la placa d'adquisició

taula 3.8 Funcions pel menú de configuració

void **menu** (void)

Aquesta és la funció principal des de la qual es comença la interacció amb l'usuari per tal de configurar el sistema. En primer lloc, s'envia la informació de les opcions que té l'usuari pel mitjà del port sèrie, i aquest gràcies a una aplicació com és per exemple l'hiperterminal de Microsoft, interactuarà enviant la opció desitjada. Un cop seleccionada l'opció desitjada, la funció cridarà a les altres funcions específiques de menú en funció de l'acció a realitzar.

El menú principal mostrat per pantalla té les següents opcions:

```

1 ----- TIME MENU
2 ----- STORAGE MENU
3 ----- A/D CONFIG
4 ----- DEFAULT CONFIG
5 ----- START ACQUISITION

```

En les opcions 1, 2 i 3, es criden a les funcions *time_menu*, *storage_menu* i *config_placa_adc* respectivament. En la opció 4, es defineixen les variables per defecte, s'informa a l'usuari de quina és la configuració per defecte que s'està configurant, es crida a la funció *AD_config_default* per configurar la placa d'adquisició i finalment es configuren els buffers d'adquisició amb la funció *buf_setup*. I per acabar, la opció 5 provoca la sortida del bucle del menú.

void **time_menu** (void)

L'adquisició no és possible sense tenir una base de temps correcta i per tant és necessari tenir ben definida l'hora del sistema. En aquesta funció tenim representat un altre llistat d'opcions on podem definir l'hora del sistema o comprovar quin error hi ha entre l'hora del sistema i el senyal del GPS. L'estructura que segueix en aquest cas és el mateix de la funció principal *menu*.

Les opcions mostrades en aquest cas quedarien de la següent forma:

```
1 ----- Set time
2 ----- Compare time
3 ----- Quit
```

La primera de les opcions ens cridarà a una sèrie de funcions per tal definir si treballem amb flanc positiu o amb flanc negatiu del GPS i per introduir l'hora del sistema, aquestes funcions són *flanco_GPS*, *menu_fecha* i *menu_hora*. En el cas de la segona opció, cridarem a dues funcions, la mateixa d'abans per definir el tipus de flanc i una altre per calcular el retard, aquestes funcions són *flanco_GPS* i *comparar_GPS* respectivament. Per finalitzar, la tercera de les opcions que tenim, és per sortir del bucle del menú de temps i així tornar al menú principal, encara que en qualsevol de les altres opcions, després de cridar a les corresponents funcions, també sortirem del menú de temps.

void **menu_fecha** (void)

En aquest menú, l'usuari no té que elegir entre cap mena d'opcions, si no que ha de introduir els valors de la data actual en el format que se li demana. En el suposat cas que l'usuari introdueixi un valor erroni com per exemple podria ser el mes 15, el sistema respondrà amb un missatge per que torni a introduir les dades de nou. El primer missatge que sortirà en cridar aquesta funció serà el següent:

Enter current date (ddmmyyyy):

On 'dd' correspon al dia, 'mm' correspon al mes i 'yyyy' correspon a l'any. Llavors un cop introduïdes les dades de la data, es procedeix a guardar aquesta informació dins dels registres de la RTC.

void **menu_hora** (void)

Així com passa en la funció anterior per introduir la data, en aquesta funció s'estructura de la mateixa forma però amb la finalitat d'introduir l'hora actual. Per ajustar correctament l'hora, l'usuari ha d'introduir el minut següent a l'actual i els segons a zero, per què després el programa no arrancarà la RTC fins que arribi el flanc del GPS que es produeix a cada minut. Llavors un cop introduïdes les dades de l'hora, es crida a la funció *sincronizar_GPS* amb aquestes informació i aquesta funció s'encarregarà de guardar les dades als corresponents registres, esperar el flanc del GPS i arrancar la RTC. El missatge amb el que es demana a l'usuari que introdueixi l'hora actual és la següent:

Enter current time (hhmmss):

On 'hh' és l'hora, 'mm' són els minuts i 'ss' són els segons.

void flanco_GPS (void)

Aquesta funció es crida des del menú de temps abans d'introduir les dades per determinar si el flanc que llegirem del GPS és positiu o negatiu. Aquí només es defineix com s'ha de llegir el senyal del GPS i després aquesta informació serà utilitzada en altres funcions on realment s'estigui llegint la informació del GPS, com ara *sincronizar_GPS* o *comparar_GPS*. En aquest cas, com que només tenim dues opcions, ens queda representat pel següent menú, on només podrem elegir entre el primer o el segon.

```
1 ----- Risingedge
2 ----- Fallingedge
```

void sincronizar_GPS (unsigned short hhmss[6])

Per sincronitzar l'hora del sistema amb el GPS, en primer lloc es guarden les dades en els corresponents registres de la RTC a continuació s'espera el flanc del GPS i finalment s'activa la RTC. Per tal de realitzar la sincronització, tenim dues possibilitats diferents segons la opció que s'hagi triat dins la funció *flanco_GPS*. En el cas de llegir del GPS per flanc de pujada, haurem de detectar quan aquest senyal està a '0' i esperar a que passi a '1', en cas contrari passarà igual però a l'inversa, de 1 a 0.

- *hhmss*: Dades corresponents al l'hora del sistema que es vol col·locar dins de la RTC

void comparar_GPS (void)

Quan finalitza l'adquisició, s'ha de comprovar l'error que existeix entre l'hora del sistema i el senyal del GPS per tal d'eliminar l'error que es podria produir a l'hora de tractar les dades. En aquesta funció, al igual que teníem en la funció anterior (*sincronizar_GPS*) tenim dues possibilitats segons si esperem el flanc de pujada o de baixada. Llavors, el que fem és esperar a que passi la RTC per zero o que arribi el flanc del GPS, i arranquem el temporitzador per començar a comptar els milisegons de retard que podria haver-hi. Finalment el temporitzador deixarà de comptar quan detecti el senyal que li falta, és a dir que si per exemple detectem primer que la RTC a passat per zero, després esperaríem a rebre el pols de la RTC per parar el comptador.

void storage_menu (void)

A l'hora d'emmagatzemar les dades a la Compact Flash, hem de decidir si volem un número concret de dades o si preferiblement volem la capacitat màxima de la memòria. En el primer dels casos podríem dir que està més enfocat a la realització de les proves, degut a que en l'adquisició final serà més convenient configurar la màxima capacitat. El menú que es mostra a continuació reflecteix les possibilitats que té l'usuari a l'hora de seleccionar una opció o un altre.

```
1 ----- Enter Manuallyvalues
2 ----- Maximumcapacity for 4MB
3 ----- Quit
```

En el cas d'accedir a la primera opció, aquesta funció cridarà a les funcions *menu_numero_ficheros* i *menu_dimension_fichero*. En la primera funció determinarà el número de fitxers a crear i la segona les dimensions que tindran cada un d'aquests. En el cas de la segona opció, saltarà a la funció *dimension_max_CF* on a partir de la capacitat de la memòria generarà tants fitxers de 4MB com sigui possible. Per últim, el darrer cas, com en altres funcions ens permet sortir del bucle d'aquest menú per tornar al menú principal.

void menu_numero_ficheros (void)

Quan accedim a aquesta funció, per pantalla ens apareixerà un missatge per tal que l'usuari introdueixi el número de fitxers que el sistema crearà de forma manual, i amb uns marges preestablerts. El missatge que apareixerà serà el següent:

Enter number of files [1 to 200]:

Un cop introduït el número de fitxers a generar, sortirà de la funció per continuar en el punt que havia estat cridada.

void menu_dimension_fichero (void)

Un cop introduït el número de fitxers que es desitgen en l'adquisició, el sistema cridarà a aquesta funció per tal que l'usuari introdueixi les dimensions de cada fitxer. Aquestes dimensions només podran ser entre quatre possibles valors que estan preestablerts i els quals es mostraran a l'usuari per què en seleccioni un amb el següent menú.

*1 ----- 4 MB
2 ----- 2 MB
3 ----- 1 MB
4 ----- 512 kB*

void dimension_max_CF (void)

Si en el menú de la funció *storage_data* es selecciona la segona opció, el sistema saltarà fins aquesta funció per tal que l'usuari pugui indicar quina és la capacitat màxima de la targeta introduïda per mitja de diferents opcions i de manera acotada a valors estàndards de memòries. El menú que es mostrarà serà el següent:

*1 ----- 64 MB
2 ----- 128 MB
3 ----- 256 MB
4 ----- 512 MB
5 ----- 1 GB
6 ----- 2 GB*

Un cop seleccionada la capacitat màxima de la nostra memòria, el sistema guardarà a la variable corresponent el número de fitxers màxim que el sistema crearà per 4MB de capacitat cada un.

char config_placa_adc (void)

Quan en el menú principal seleccionem la opció de configurar la placa d'adquisició, es crida a aquesta funció que ens anirà demanant els diferents paràmetres necessaris per configurar correctament la placa d'adquisició. En primer lloc ens demanarà que introduïm el valor de la freqüència de mostreig, després el número de canals amb els que volem treballar, a continuació el valor de guany del hidròfon i finalment el guany del sismòmetre sempre i quant haguem activat més d'un canal.

Els diferents menús que apareixeran per tal de que l'usuari introdueixi els paràmetres necessaris i de forma pautaada, són els que a continuació queden reflectits.

Frequència de mostreig:

- 1 ----- 125 Hz
- 2 ----- 250 Hz
- 3 ----- 500 Hz
- 4 ----- 1000 Hz

Número de canals actius:

- 1 ----- 1 Channel (*Channel 1 enable*)
- 2 ----- 2 Channels (*Channels 1 and 2 enable*)
- 3 ----- 3 Channels (*Channels 1, 2 and 3 enable*)
- 4 ----- 4 Channels (*Channels 1, 2, 3 and 4 enable*)

Guany de l'hidròfon:

- 1 ----- x1
- 2 ----- x2
- 3 ----- x4
- 4 ----- x8
- 5 ----- x16
- 6 ----- x32
- 7 ----- x64

Guany del geòfon:

- 1 ----- x1
- 2 ----- x2
- 3 ----- x4
- 4 ----- x8
- 5 ----- x16
- 6 ----- x32
- 7 ----- x64

A mesura que l'usuari va introduint els valors que se li demanen, es van guardant en les corresponents variables per un cop introduïts tots els paràmetres es pugui cridar a les funcions *AD_config* per configurar la placa d'adquisició i *buf_setup* per generar els buffers de l'adquisició.

4. MANUAL D'USUARI

En aquest apartat explicarem lleugerament la interfície d'usuari realitzada per mitjà de la comunicació del port sèrie. La interfície està basada per mitjà de menús que s'envien per aquest port i la interacció de l'usuari amb el mateix. En el nostre cas, hem fet ús de l'aplicació de Microsoft denominada HiperTerminal per comunicar-nos amb el sistema. Un cop iniciem el sistema, es necessari introduir l'hora actual, amb la finalitat que el sistema tingui les dades de temps fiables i per arrancar la RTC d'aquest. A continuació es poden introduir les dades de forma manual com també seleccionar la configuració per defecte. Finalment hem d'arrancar l'adquisició amb la darrera de les opcions per tal que el sistema comenci a adquirir dades. A continuació (figura 4.1) tenim un mapa dels diferents menús que ens anirem trobant segons les opcions elegides:

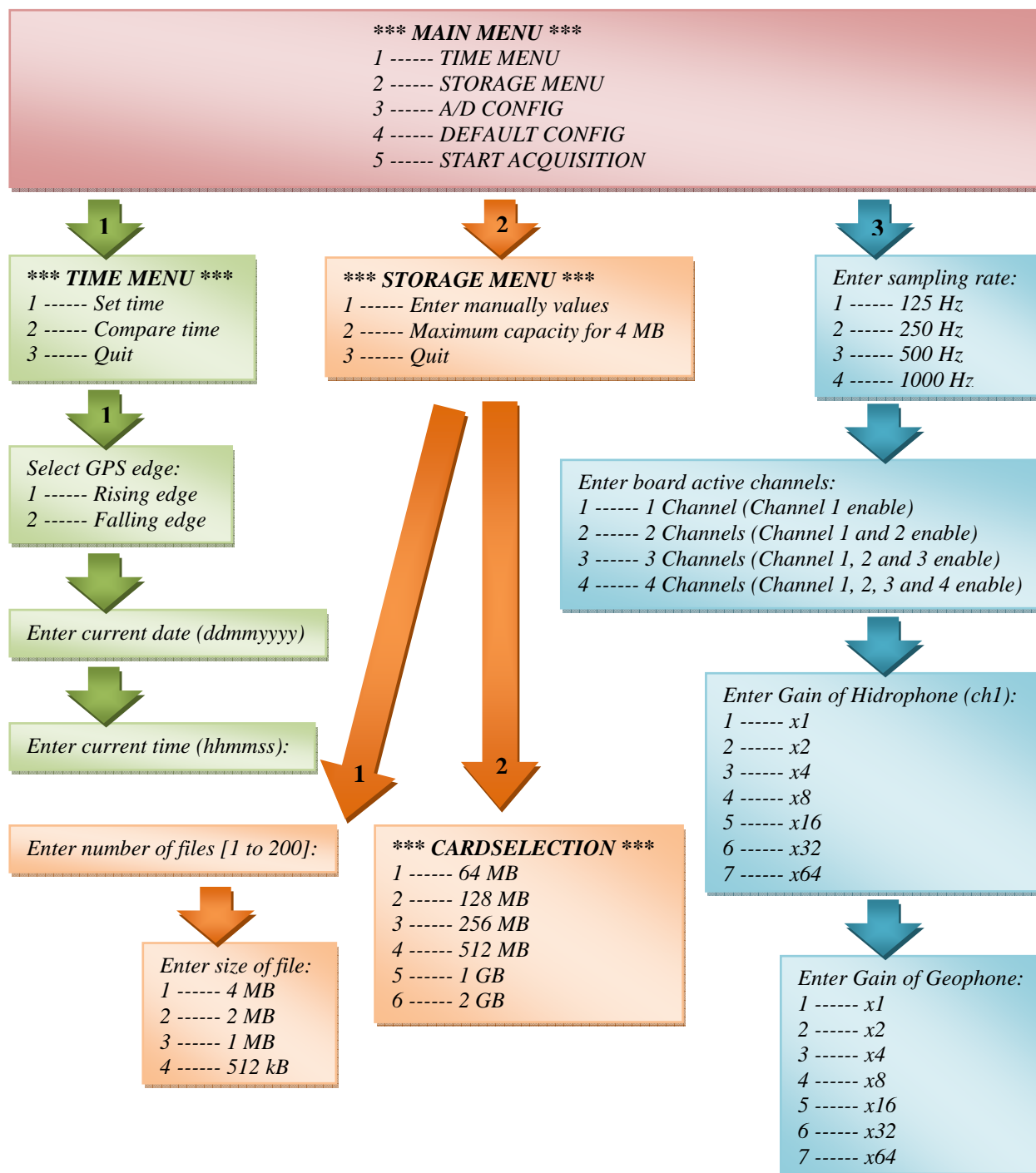


figura 4.1 Diagrama de blocs de la estructura del menú

Quan arribem al final d'una de les línies del menú, es retorna al menú principal automàticament. Referent a la configuració per defecte que tenim en l'opció 4 del menú principal, és la més adequada alhora de realitzar les proves, però posteriorment aquesta configuració s'adequarà a les necessitats del sistema. I finalment, en la opció 5 iniciarem l'adquisició de les dades.

La configuració per defecte actual que hi ha programada és la següent:

- 4 canals habilitats
- 250 Hz de freqüència de mostreig
- Guany de "1" per tots els canals
- Creació d'1 sol fitxer
- Fitxer de 512 kB

5. PROVES I RESULTATS

5.1. APARELLS UTILITZATS

A continuació tenim un llistat dels aparells utilitzats en les proves realitzades on s'han anat simulant diferents elements del sistema, i després, a la figura 5.1, un esquema de com s'han connectat aquests a la placa d'avaluació i al PC :

Font de tensió: Agilent E3631A

- Font que té tres sortides de tensió (valors màxims de 6V, 25V i -25V)

Generador de funcions: HP 33120A

- Generador dels 32,768 kHz per la RTC
- Simulació de la senyal de GPS amb un període de 1min.

Generador de funcions: Stanford research systems CG635

- Generador dels 32,768 MHz pel clock de la placa ADC

Generador de funcions: PROMAX GF-941

- Generador de 250 Hz de mostreig per l'adquisició de dades de la placa ADC

Generador de funcions diferencial: Stanford research systems DS360

- Simulació dels sensors. Generació d'un senyal diferencial de molt baixa distorsió

Oscil·loscopi: Tektronix TDS 220

- Per testejar diferents senyals

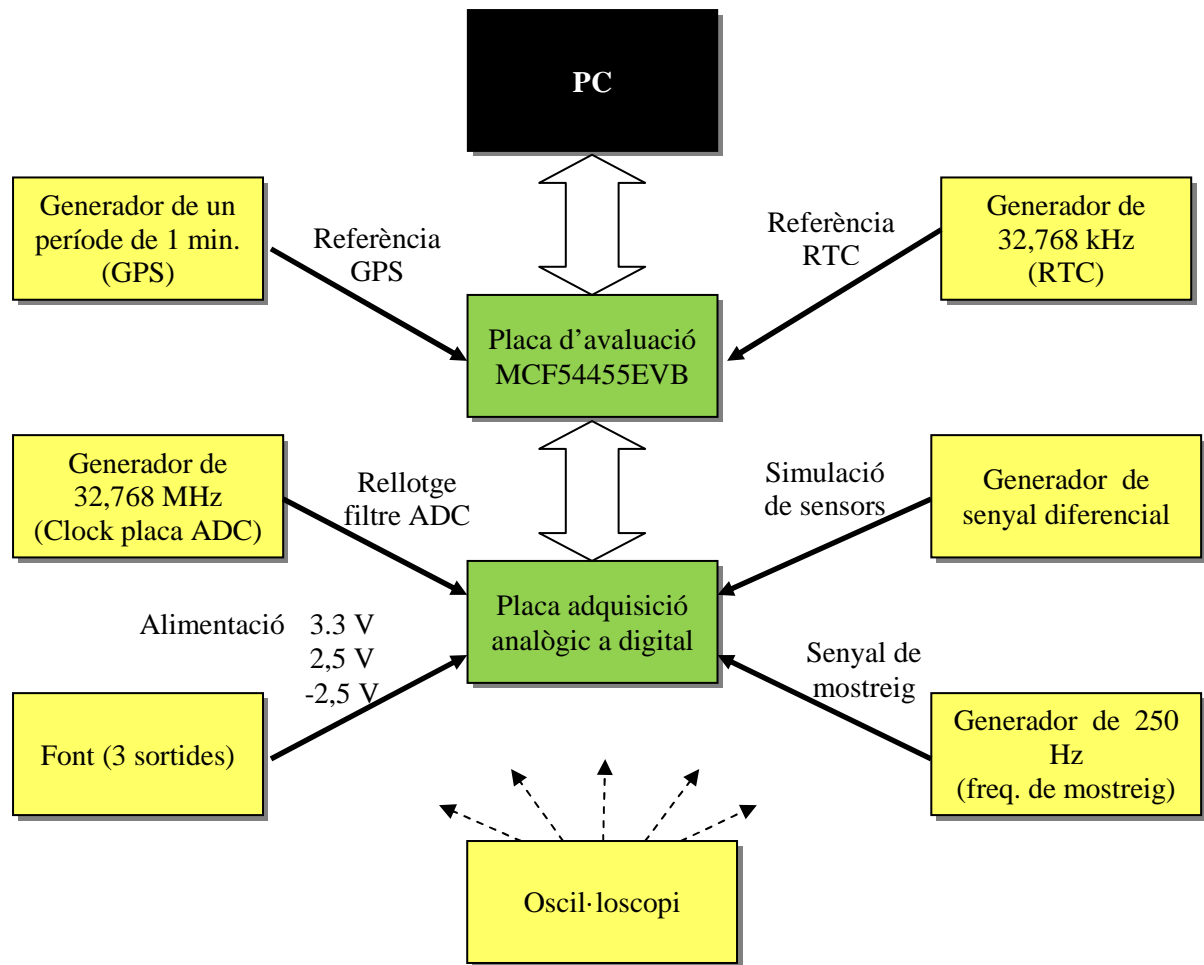


figura 5.1 Esquema de les connexions dels diferents aparells utilitzats

5.2. PROVES REALITZADES

Per tal de corroborar que la nova placa del microprocessador funciona correctament, es procedeix a migrar les diferents aplicacions específiques utilitzades en la placa de avaluació. Aquestes funcions específiques consisteixen en aplicacions encarregades de comprovar una part del sistema sense la necessitat de interferir amb un altre part. Un cop confirmat el seu bon funcionament, s'adapta la part del codi necessari i s'uneix a l'aplicació principal.

Les diferents proves que s'han realitzat per la verificació i per ser utilitzades posteriorment en l'aplicació final són:

- Configuració i utilització dels ports GPIO (Ports d'entrada i sortida de propòsit general)
- Configuració i utilització del port sèrie tant per escriure com per llegir.
- Ús de la RTC del sistema (Relloige de temps real)
- Utilització de les interrupcions del sistema.
- Creació de funcions per l'ús de retards dins de la nostra aplicació (Temporitzadors)
- Modificació de la freqüència de treball del sistema en valors d'alta freqüència (Ús de diferents freqüències per mitjà del PLL)
- Modificació de la freqüència de treball del sistema en valors de baixa freqüència (Ús de diferents freqüències per mitjà del mode LIMP)
- Configuració i utilització del bus SPI
- Configuració i utilització del port ATA per llegir i escriure en una memòria Compact Flash
- Treballar a baix consum

En primer lloc, es configura la freqüència de treball del sistema a la màxima, segons venia per defecte la placa de avaluació i així no dificultar el desenvolupament de les proves. Més endavant, una de les proves realitzades consisteix en el desenvolupament d'una aplicació destinada a modificar les freqüències de treball del microprocessador, tant en alta freqüència com en baixa freqüència. A continuació donarem unes pinzellades referents a les proves realitzades.

Ports GPIO:

Cada pin del microprocessador, té una o més funcionalitats, i en alguns d'aquests tenen la opció de ser configurats com a ports de propòsits general. Per aquest motiu, alhora d'utilitzar un port com a ús general, cal modificar dos registres diferents, un primer registre per definir que l'ús serà el de propòsit general, i un segon registre per indicar si serà d'entrada o de sortida. Finalment, a l'hora d'utilitzar aquest pin, es tindrà que accedir a un tercer o quart registre en funció si s'ha configurat com a sortida o com entrada.

El resultat de l'aplicació del correcte funcionament dels ports GPIO, l'hem observat connectant l'oscil·loscopi en els pins de sortida i observant com el senyal a la sortida anava canviant correctament. En el cas de la utilització dels pins com a entrada, se li aplica una tensió de 3,3 volts al pin corresponent i per mitjà de l'aplicació, podem observar com el valor d'aquest pin canvia.

Port sèrie:

Per la comunicació pel port sèrie, primer de tot ens hem d'assegurar que els pins utilitzats en el microprocessador estan configurats per aquesta tasca, degut a que aquests tenen la possibilitat de ser configurats per diferents utilitats. A continuació tenim una sèrie de registres que ens permeten configurar el protocol de comunicació pel port sèrie, és a dir definir quants bits s'enviaran de dades per cada trama, determinar el bit de paritat i el bit de stop, definir si hi ha control de flux i finalment indicar els bits per segons dels enviaments.

Llavors, un cop tenim definits els diferents registres de configuració, tenim uns altres registres per tal de realitzar l'enviament de les dades. Un registre del tipus buffer per enviar les dades, i un altre registre per rebre-les. A part d'això, també tenim un bit per informar si l'enviament s'ha realitzat i un altre bit per saber si existeixen dades en el buffer de recepció.

Per realitzar les proves amb aquesta aplicació, es disposa de la placa del microprocessador, d'un cablejat sèrie i de l'ordinador de programació, a la figura 5.2 podem observar un esquema de la connexió. A l'ordinador, hem utilitzat un aplicació que ens permet comunicar-nos pel port sèrie. Aquesta aplicació és el HiperTerminal de Microsoft. Amb els resultats finals podem observar com la placa del microprocessador ens envia un missatge i en escriure amb l'ordinador, la placa ens retorna el caràcter enviat com a confirmació.



figura 5.2 Esquema de la connexió per la prova del port sèrie

RTC:

El sistema porta incorporat un rellotge en temps real. Aquest rellotge ens permet tenir en tot moment l'hora actual, element necessari a l'hora d'adquirir dades, tenir una bona base de temps. La RTC està composta per una sèrie de registres on es guarda la informació referent a l'hora actual, registres que podem modificar per actualitzar o simplement llegir-ne el seu valor. Un cop actualitzada la informació del registre, només cal indicar a quina freqüència està el cristall d'entrada per mitjà d'un altre registre i finalment arrancar la controladora de la RTC per què comenci a comptar.

En l'obtenció de resultats, s'han introduït dades en els diferents registres de temps i per pantalla s'han anat imprimint aquestes cada cert temps, amb el que s'ha pogut veure el l'evolució del temps.

Interrupcions:

Per verificar que les interrupcions funcionen correctament en la placa d'avaluació, existien dos botons d'accés extern, els quals anaven connectats al microprocessador per mitjà d'una interrupció d'entrada. En aquest cas la comprovació consistia en accionar el botó i veure com un led de la placa de avaluació s'encenia i s'apagava. En el cas de la placa del microprocessador nova, no es disposava d'aquest botó per comprovar la funcionalitat de les interrupcions però existia l'accés a algunes interrupcions en un dels connector inserits per ús general.

Per configurar qualsevol de les interrupcions s'han de definir prèviament diferents paràmetres. Aquests corresponen a la definició del nivell de prioritat de cada interrupció (valor que pot anar de 1 a 7), la rutina al servei de la interrupció que s'ha d'executar per cada una de les interrupcions activades i finalment la creació de la màscara de les interrupcions per definir quines es podran activar i quines no. Un cop configurats tots els registres, es poden habilitar les interrupcions per començar a generar peticions.

Finalment, alhora de veure'n els resultats obtinguts d'aquesta aplicació, s'ha imprès per pantalla un text cada cop que el pin d'entrada s'havia activat.

Temporitzadors:

Per generar els temporitzadors, cal definir els paràmetres de comptatge i aprofitar l'apartat d'interrupcions del que ja es disposa. En aquesta aplicació és creen unes funcions que compten el valor que prèviament s'emmagatzema en el registre corresponent i cada cop que finalitza aquest comptatge salta una de les interrupcions. Per aprofitar aquesta funcionalitat, es crea una funció amb la idea de arrancar el temporitzador des de zero i incrementar una variable cada cop que salta la interrupció (cada milisegon), amb el que s'aconsegueix un retard controlat amb una variació mínima de 1 milisegon.

Com a resultat es pot observar com al cridar a aquesta funció amb la definició de la variable a per exemple 5000 milisegons, el sistema quedava parat esperant que aquest temps passés.

Rellotge en mode PLL:

El sistema treballa per defecte en mode PLL, que consisteix en un mòdul del microprocessador que permet elevar la freqüència de l'entrada fins a una freqüència de treball màxima de 266 MHz. Per defecte el sistema treballa amb aquesta freqüència, però aquesta pot ser modificada a nivell de software i de hardware.

En el cas del hardware, consisteix en una sèrie de microinterruptors destinats a la configuració d'inici (també existeixen altres mètodes per modificar aquest valor, com és iniciar amb valors per defecte, però nosaltres hem optat pel mètode extern), en els quals ens permeten augmentar la freqüència d'entrada fins a nivells superiors. En el cas de la part de software, es tracta de modificar uns registres de configuració que ens

redueixen la freqüència per ajustar-la als valors adequats. A la figura 5.3, tenim l'esquema de com està connectat el cristall d'entrada i la resta de mòduls del sistema.

Les freqüències de treball que podem optar amb aquesta configuració, dependrà de la freqüència d'entrada que tinguem. En el cas de la placa del microprocessador, tenim dos possibles orígens de la freqüència d'entrada, una que prové des del cristall de quars incorporat a la mateixa placa i l'altre des d'una placa externa amb un senyal de rellotge més precís. Si per exemple agafem l'origen de la placa externa, que serà la utilitzada en l'aplicació general, tindrem una freqüència d'entrada de 32,768 MHz. Tenint en compte algunes restriccions que indicarem posteriorment i que hem de tenir present a l'hora de configurar els registres, tindrem com a resultat un determinat rang de freqüències per elegir.

Fórmula per determinar la freqüència principal del sistema:

$$f_{sys} = f_{REF} * \frac{PFDR}{OUTDIV1 + 1}$$

F_{sys} : Freqüència principal del microprocessador

F_{REF} : Freqüència d'entrada al sistema

PFDR: Valor configurat per hardware

OUTDIV1: Primer divisor del registre destinat per la freqüència principal

- Les restriccions que tenim a l'hora de seleccionar un valor en els registres, consisteix en la diferència de velocitat a que ha de treballar cada mòdul referent a la freqüència principal. A continuació, primer podem observar el paràmetre del registre en el que actuem i a continuació la restricció a tenir en compte:

OUTDIV1 $\rightarrow f_{SYS}$
OUTDIV2 $\rightarrow f_{SYS}/2$
OUTDIV3 $\rightarrow f_{SYS}/4$ o 8

- Un altre de les restriccions és que la freqüència màxima de treball del sistema està en aproximadament 266 MHz.
- Els valors del PFDR oscil·laran entre les següents possibilitats: 6, 8, 10, 12, 16, 18, 20, 24.
- Possibles valors del registre OUTDIV1: 1, 2, 3

Finalment, el rang de freqüències que podrem tenir en aquest cas i tenint present totes les restriccions, oscil·laran entre els 49,152 MHz i els 262,144 MHz.

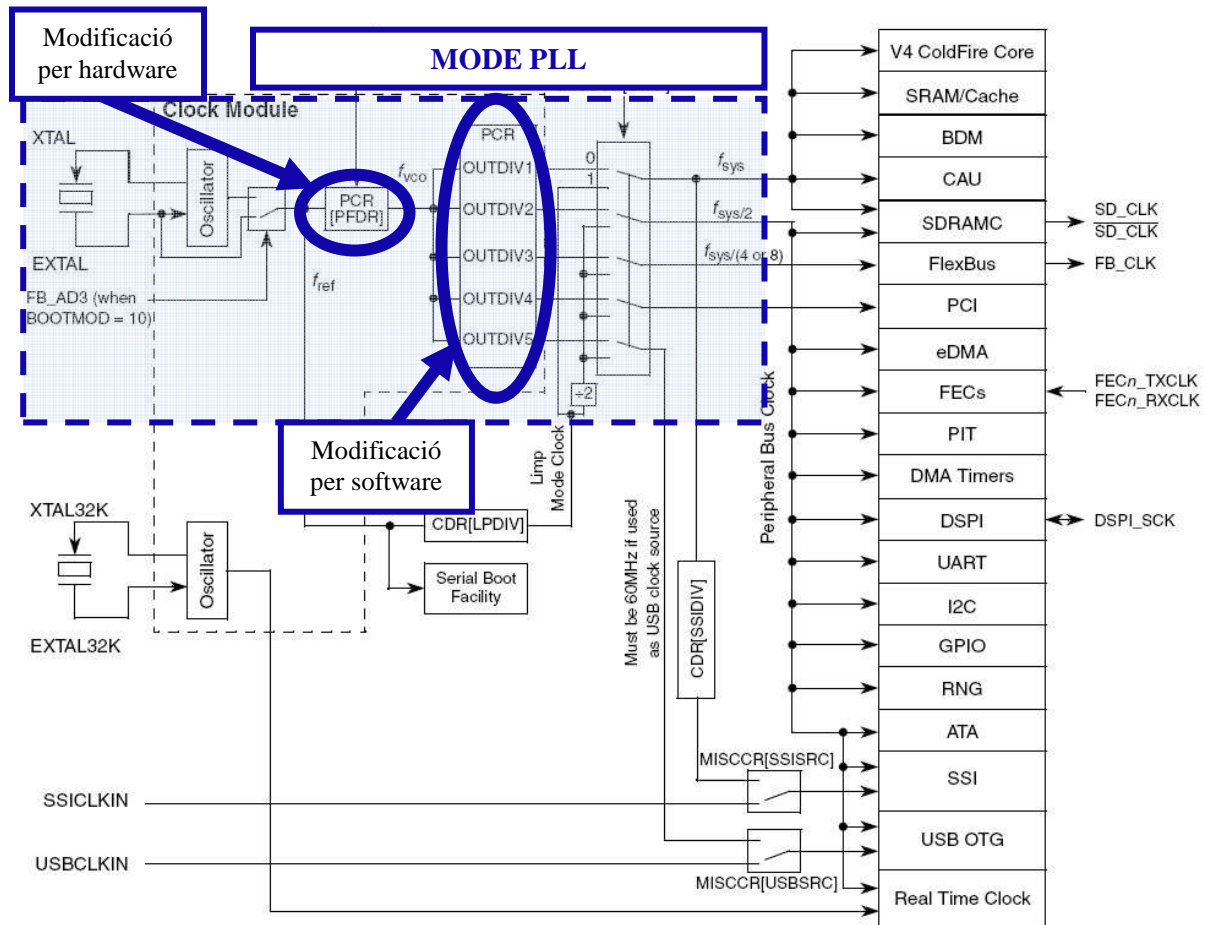


figura 5.3 Estructura de les connexions del rellotge de sistema pel mode PLL

Relotge en mode LIMP:

A l'hora de treballar en el mode LIMP, la freqüència general baixa a valors inferiors o iguals a la freqüència d'entrada del sistema, amb el que algunes de les controladores d'aquest deixen de funcionar com és la del mòdul USB, la dels ports Ethernet, la del port PCI, la del ATA i també la memòria SDRAM. Amb la pèrdua d'aquesta darrera controladora de memòria, es perdia l'execució del programa amb el que no es podia provar en la placa d'avaluació al 100%. Llavors, la solució adoptada en la nova placa del microprocessador, ha estat la incorporació d'una segona memòria SRAM controlada per mitja del FlexBus, el qual no deixa de funcionar a baixa freqüència.

Per treballar en el mode Limp, hem de modificar un dels registres del sistema que ens permeten dividir la freqüència d'entrada entre una potència de 2. A la figura 5.4 podem veure l'estructura de com està connectada la senyal d'entrada al sistema amb els mòduls d'aquests, passant per entre mig pel registre que redueix la freqüència de treball. A continuació tenim l'equació que es segueix per determinar la freqüència de rellotge principal del sistema, i la resta de freqüències correspondran a la meitat de la freqüència principal que haguem configurat.

$$f_{sys} = \frac{f_{REF}}{2^{LPDIV}}$$

f_{sys} : Freqüència principal del microprocessador

f_{REF} : Freqüència d'entrada al sistema

LPDIV: Valor a modificar per registre per obtenir la freqüència desitjada

El valor de LPDIV pot tenir diferents valors compresos entre 0 i 15, amb això i que la freqüència d'entrada és de 32,768 MHz com en les proves del PLL, tindrem una freqüència principal compresa entre 1 KHz i 32,768MHz.

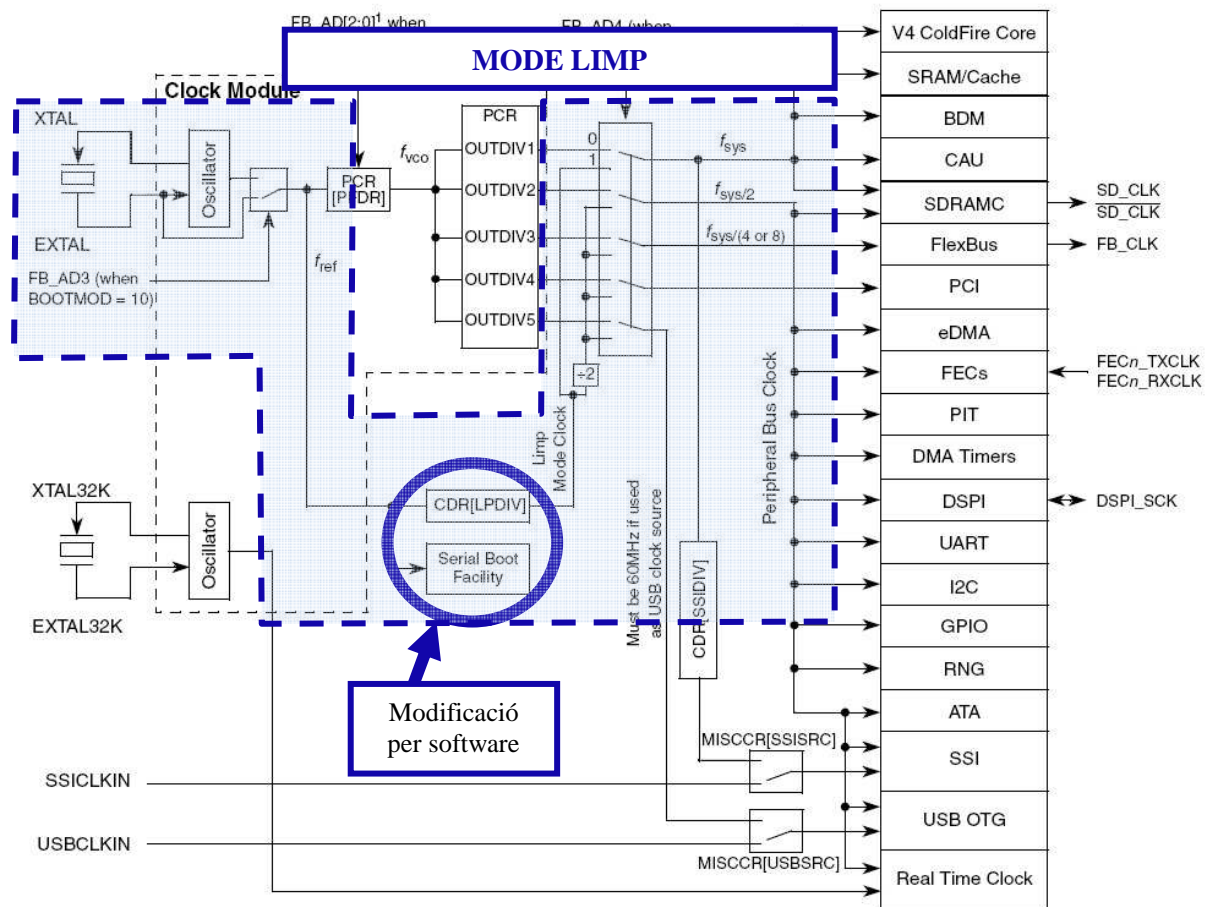


figura 5.4 Estructura de les connexions del rellotge de sistema pel mode LIMP

Port SPI:

El port SPI, és el port destinat a l'adquisició de les dades, degut a que la comunicació entre el microprocessador i la placa d'adquisició es realitza per mitjà d'aquest. Per realitzar les primeres proves amb aquesta aplicació, un cop configurat el protocol amb els corresponents registres, es procedeix a realitzar diversos enviaments pel port i amb l'ajuda d'un oscil·loscopi comprovar que l'enviament es realitza correctament.

Una comprovació posterior de que funciona correctament aquest port de comunicació, va consistir en connectar en port de SPI amb un sensor de temperatura i humitat de SENSIRION amb el model sht11 (figura 5.5), el qual es comunicava per SPI. El ser un senzill dispositiu, aquest treballa com esclau dins del protocol de comunicació, i quan el microprocessador enviava una petició de dades, aquest li tornava el valor de la temperatura i la humitat mesurades a l'ambient. Per tal de saber si realment aquestes dades eren correctes, un cop llegides s'imprimien per pantalla per poder-ne veure el resultat.

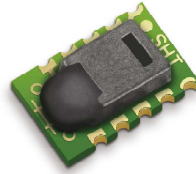


figura 5.5 Sensor humitat i temperatura *sht11*

A continuació, un cop comprovat el funcionament del port SPI, el següent pas va ser comunicar la placa d'adquisició de dades analògic a digital amb el microprocessador, per tal de configurar i iniciar l'adquisició. Aquesta prova es va realitzar juntament amb altres parts de l'aplicació, com és el port sèrie amb el qual enviàvem les dades adquirides o les interrupcions per informar que existien dades. A l'altre costat del port sèrie, hi teníem l'ordinador amb una aplicació de Labview per llegir i mostrar per pantalla les dades adquirides. A la figura 5.6 tenim una imatge de l'aplicació utilitzada.

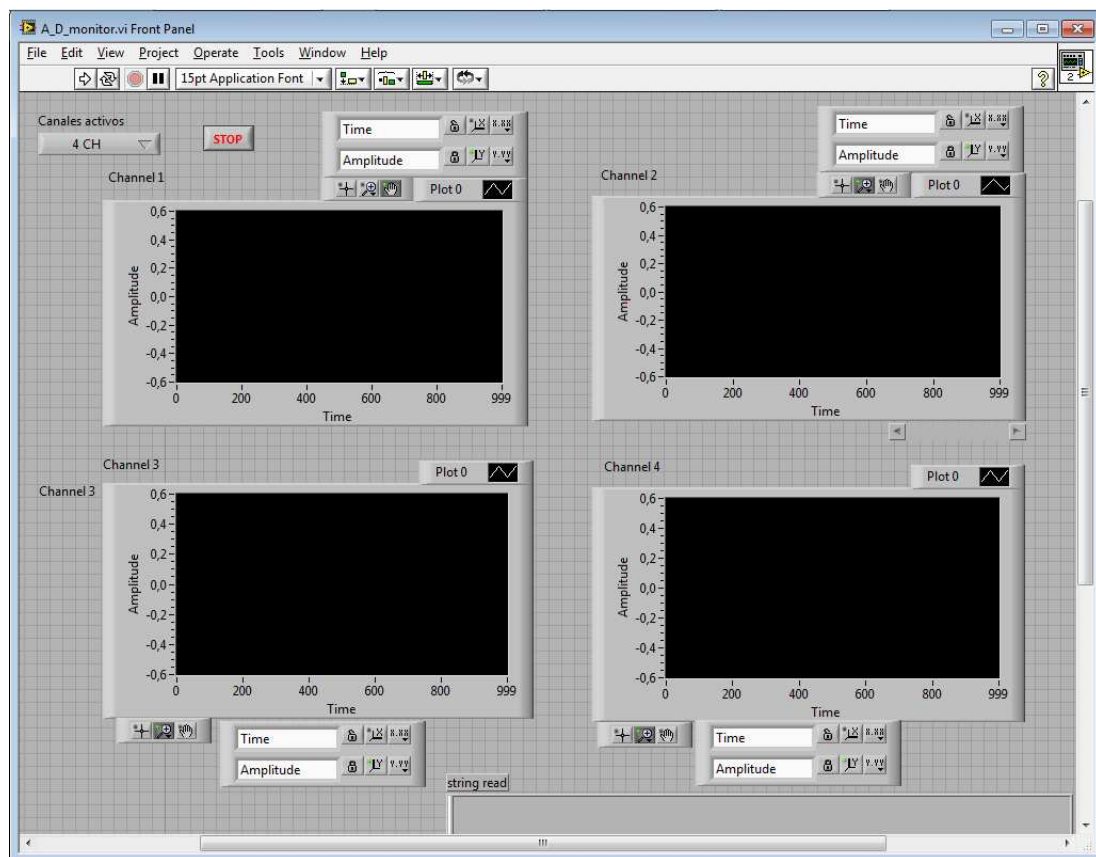


figura 5.6 Aplicació per visualitzar les dades enviades pel port sèrie

El muntatge del sistema utilitzat per realitzar les proves d'adquisició i visualitzar-les en Labview, consistia en connectar un generador de funcions a l'entrada dels quatre canals de la placa d'adquisició, connectar aquesta ala placa del microprocessador per mitjà del port SPI i connectar la placa del microprocessador amb l'ordinador per mitjà del port sèrie. Amb tot això, la placa d'alimentació del sistema aportant les diferents tensions a cada una de les plaques d'adquisició.

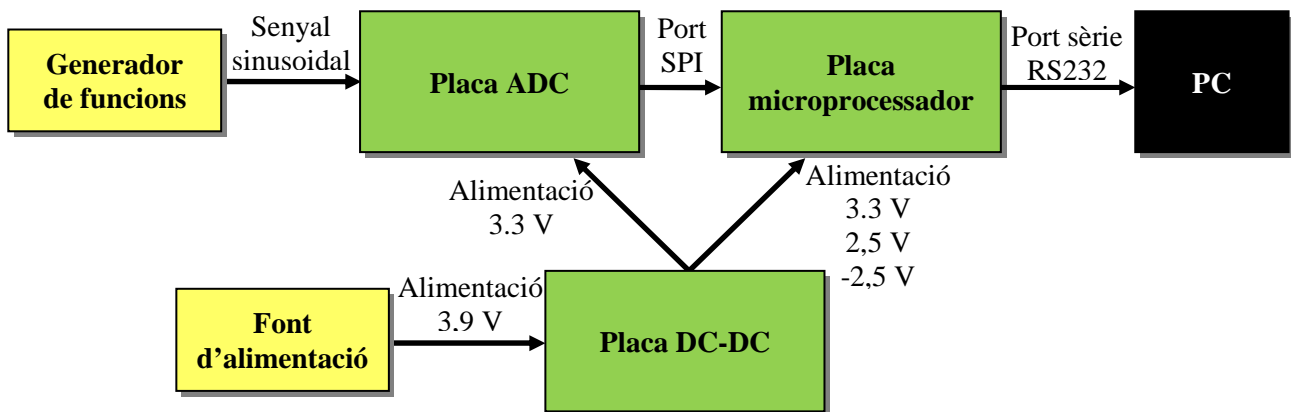


figura 5.7 Esquema del muntatge de la prova de comunicació SPI

Un cop comprovat el bon funcionament de la comunicació del port SPI, també quedava verificat que la configuració de la placa d'adquisició de dades, amb els valors per defecte, eren correctes. El següent pas que procedeix ara, un cop es tenen les dades correctes, consisteix en l'emmagatzematge de les dades en la targeta Compact Flash. Aplicació que aconseguim en comprovar el bon funcionament del port ATA que explicarem a continuació.

Port ATA:

La part del port ATA és el que dona accés a la comunicació amb la Compact Flash. En aquesta part, és on s'han realitzat les proves de comunicació amb la targeta de memòria on aniran emmagatzemades les dades de l'adquisició. Les proves que s'han fet consisteixen bàsicament en llegir i escriure dades en un fitxer de la memòria. Per realitzar aquestes proves, s'han fet ús de les funcions pel sistema de fitxers *FAT*.

Per realitzar les primeres proves de comunicació amb la targeta de memòria, es va fer ús de la placa d'avaluació i una petita placa per adaptar el connector del port ATA al de la targeta Compact Flash. Amb això teníem dos reptes a resoldre, primer que al treballar a baixa freqüència la controladora ATA deixava de funcionar i segon que necessitàvem treballar en algun sistema de fitxers concret. Per treure solucions a aquests problemes, es va optar per treballar amb els ports d'entrada i sortida de propòsit general (GPIO) i amb uns codis de programari lliure per llegir i escriure en el sistema de fitxers *FAT*.

Amb aquest programari lliure hi ha una sèrie de funcions creades per ser utilitzades, i unes altres funcions que han de ser creades per l'usuari per tal d'adaptar-ho al hardware que es tingui i al microprocessador que es tingui. A la figura 3.3 de l'apartat 3.5 *Emmagatzematge de les dades*, podem veure un esquema de com està estructurada l'accés a les diferents funcions. Les funcions de la part FatFs corresponen a les funcions que no hem de modificar i en el cas de la part d'E/S de baix nivell, és on tenim les funcions específiques del hardware.

Inicialment com que es treballava amb la placa d'avaluació, es tenia com a sortida els pins corresponents a la controladora del port ATA, però configurant aquests com a pins GPIO. Per aquest motiu, a l'hora de llegir o escriure una dada en el corresponent port de sortida, es va crear una sèrie de funcions per redirigir la informació de cada bit de les dades al corresponent pin de sortida per què els bits no estaven correlatius. Finalment aquesta aplicació funciona correctament, podem veure com s'escriuen les dades en un fitxer i també podem llegir-ne la informació, però a baixa freqüència no es podia provar a conseqüència de que la controladora de la memòria SDRAM no funcionava.

Un cop es tenia l'aplicació de comunicació, es va procedir a la incorporació d'aquesta part en l'aplicació general, per tal de capturar dades i emmagatzemar-les dins de la memòria externa. Es configurava la placa d'adquisició, s'iniciava una nova adquisició i es guardaven les dades dins de la Compact Flash. Per tal de comprovar si les dades guardades són correctes o no, s'utilitza una aplicació realitzada en Matlab, que descomprimeix les dades del fitxer creat en l'adquisició i per cada canal, les representa en un gràfic. En l'apartat 5.3 *resultats*, tenim una representació d'aquest resultat obtinguts satisfactòriament.

Una de les modificacions noves que es van incorporar en la nova versió de la placa del microprocessador, era la d'una nova memòria SRAM connectada en el flexbus en lloc del bus de la SDRAM, amb el que ja era possible baixar la freqüència i realitzar les noves proves de llegir i escriure en la Compact Flash a freqüències baixes. La primera prova realitzada consistia en baixar la freqüència de treball a valors baixos de per exemple els 32,768 MHz i realitzar lectura i escriptura de la Compact Flash. La segona prova, un cop comprovat que seguia funcionant l'aplicació de comunicació, es va procedir a provar l'aplicació general, però en aquest cas la comunicació era massa lenta. S'escribia més ràpid en els buffers que no pas llegir-ne, pel que aquest s'omplia i l'aplicació perdia les dades.

La solució adoptada per aquest entretemps, va correspondre en tornar a l'aplicació anterior i a partir d'un bit GPIO, comprovar els temps de les funcions destinades a l'escriptura. Amb això vam veure que tardava molt en escriure les dades i que s'havien de seguir dos camins, modificar les funcions per fer-les més òptimes i eliminar les funcions de redireccionament fent una petita modificació a nivell de hardware. Si mirem la figura 5.8, podem veure la modificació realitzada a nivell hardware. Aquesta modificació va consistir en eliminar un buffer que hi havia entre el microprocessador i la targeta de memòria per realitzar el redireccionament de pins per mitjà de pistes, aconseguint que l'escriptura al bus de dades fos directament al registre corresponent.

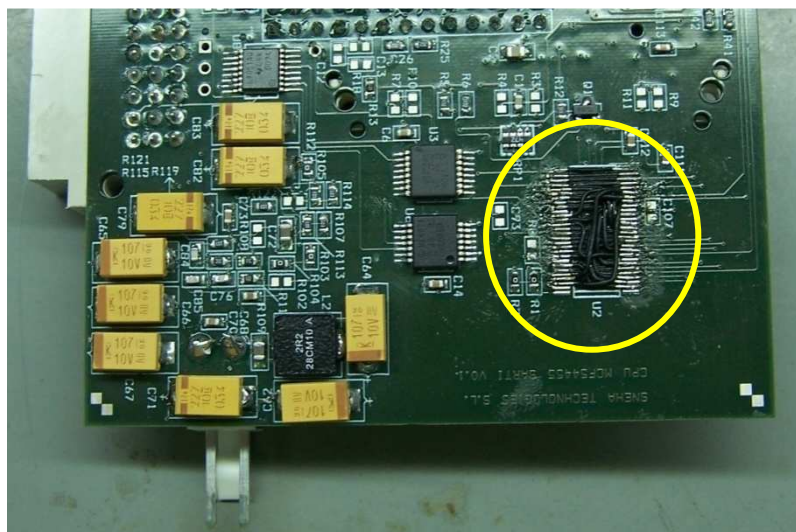


figura 5.8 Modificació de la placa micro, eliminació del buffer

Un cop realitzada aquesta modificació a nivell hardware i eliminant les funcions de redireccionament, es va poder observar com el temps que tardava en escriure un buffer de 32768 Bytes i amb una freqüència de treball de 16,284 MHz, passava de ser de 9 segons a ser de 4,5 segons, que encara no era suficient. El següent pas va ser la modificació per fer les funcions més òptimes, donant més importància a la velocitat de l'execució de l'aplicació que en les dimensions que poguéssin tenir aquesta.

La modificació havia estat realitzada pel bits de dades, però en el cas dels bits de control aquesta modificació no era possible. El que sí era possible, era que en cada moment que es volia escriure en els pins de control, mirar exactament que es volia escriure i accedir a aquests registres de forma directe, és a dir mirant cada cas en particular. Finalment, amb aquesta modificació, altres petites modificacions i mantenint el rellotge i buffer com en el cas anterior, es va aconseguir reduir els temps fins als 500 ms.

Finalment, quan ja no es podien realitzar més modificacions que aportessin una millora considerable, es va procedir a la comprovació del funcionament de les noves funcions amb l'aplicació general, amb el que es va observar que a 16,384 MHz encara no era suficient però en canvi a 32,768 MHz sí que ho era, i que amb aquesta freqüència de treball es podia treballar correctament.

Baix consum:

Una de les formes que permet portar tot el sistema a un estat de baix consum, és l'aplicació del mode *limp* per configurar el rellotge del sistema, que a part de reduir la freqüència de treball, també deshabilita algunes de les controladores, amb el que es redueix el consum general del sistema. Amb aquesta configuració ja queda reduït gran part del consum del sistema, però encara tenim l'opció de reduir-lo una mica més, aplicant una de les funcions descrites anteriorment com a *Low_power_mode_IN*. Aquesta funció ens permet deshabilitar els rellotges destinats als perifèrics, com també deixar el sistema sota mínims en qüestió de senyals de rellotge es refereix. En el nostre cas ho utilitzem durant els períodes d'inactivitat entre adquisicions, fent que el sistema consumeixi una mica menys. En el següent apartat 5.3 *resultats* tenim els resultats de consums obtinguts en diferents proves realitzades.

5.3. RESULTATS

A continuació (figura 5.9) tenim una imatge de tot el conjunt de plaques connectades per realitzar les diferents proves d'adquisició, juntament amb el connector de BDM i el port sèrie connectats. En aquesta imatge podem observar els dos cables de color marró i rosa que corresponen a la tensió d'alimentació general, els cables vermell i negre corresponents a la alimentació de la placa del microprocessador, i per últim els quatre cables de color blau, lila, groc i verd que corresponen a la alimentació de la placa d'adquisició.

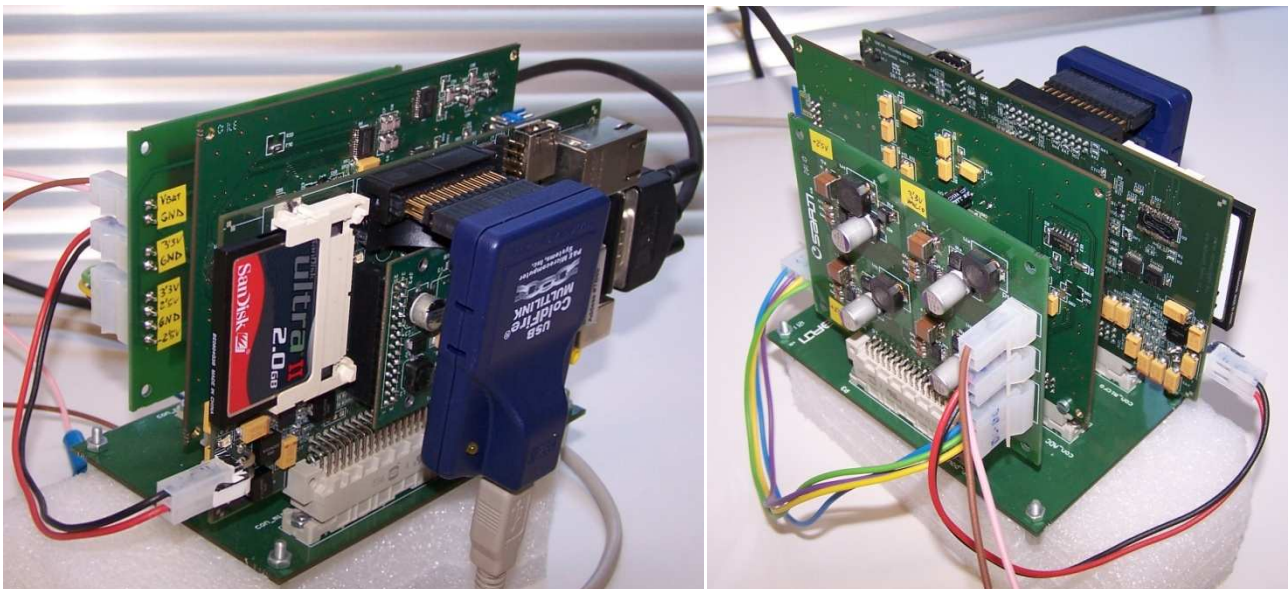


figura 5.9 Conjunt de plaques connectades al backplane

Una de les proves realitzades i explicades en l'apartat anterior (5.2 *proves realitzades*), consistia en adquirir un senyal sinusoidal a partir d'un generador de funcions, omplir quatre buffers (pels corresponents quatre canals) i un cop es tenien aquests quatre plens, enviar-ho tot pel port sèrie. A l'altre costat del port sèrie, hi tenim l'ordinador amb una aplicació creada en Labview, amb la qual, es llegeixen les dades i es visualitzen per pantalla. A la següent imatge (figura 5.10) podem visualitzar una mostra de una de les adquisicions amb aquesta aplicació. En el cas del generador de funcions, teníem un senyal de 2 Hz i una amplitud de 1 V pic a pic.

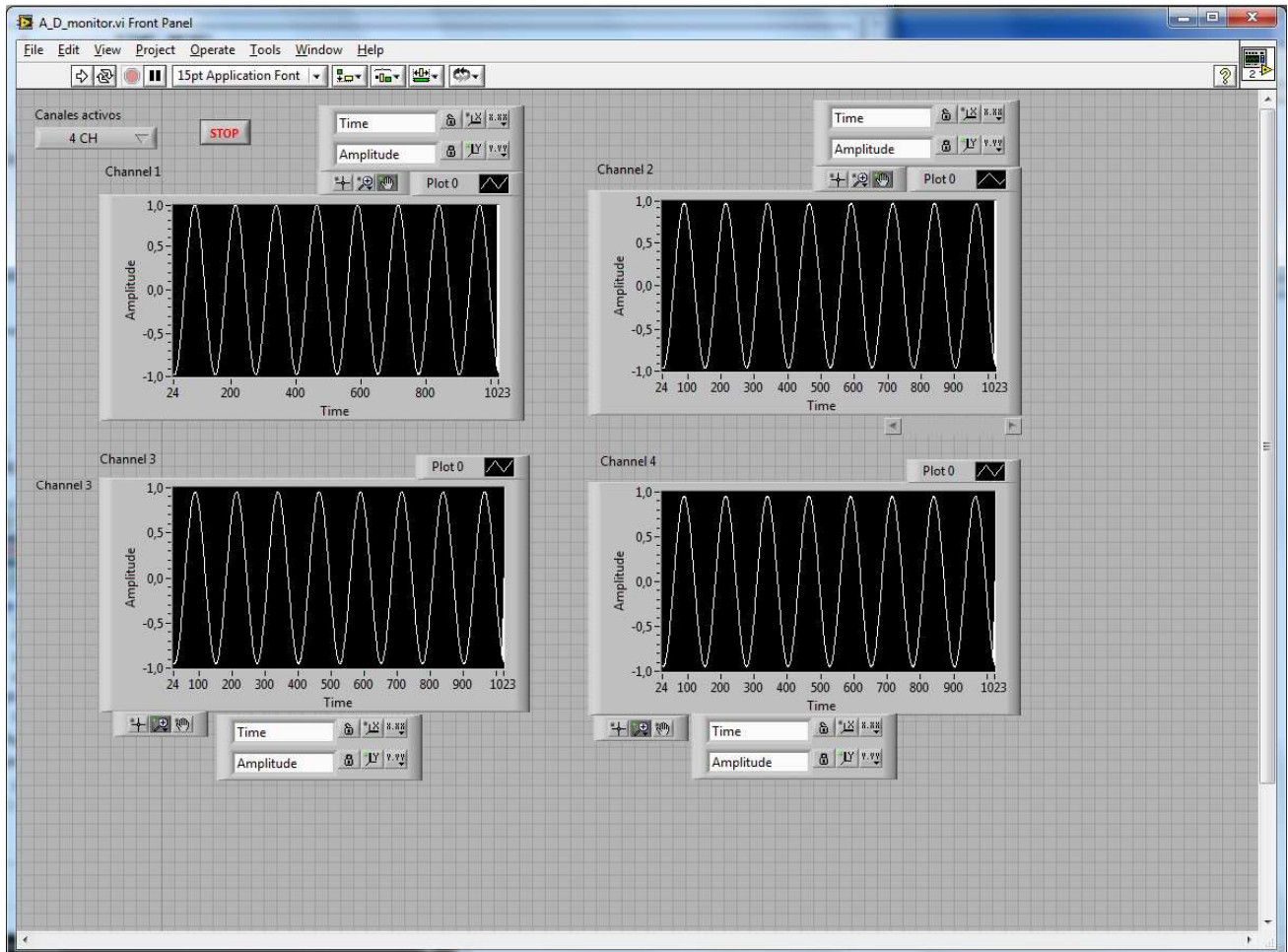
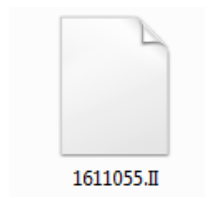


figura 5.10 Mostra dels resultats obtinguts en una adquisició, visualitzat amb Labview

Un dels resultats finals obtinguts, on podem observar el bon funcionament de tot el sistema correspon a la verificació de que els fitxers que es generen en una de les adquisicions, es realitzen satisfactòriament. A la figura 5.11, podem veure el resultat de descomprimir i representar gràficament les dades extretes dels fitxers que es generen per mitjà d'una aplicació creada en Matlab. A part de la representació correcta de les funcions, també podem observar com les funcions referents a les dades temporals funcionen correctament, degut a que en el títol superior de la finestra, podem veure la data en que es va realitzar l'adquisició.

El senyal que teníem a l'entrada d'aquesta adquisició, consistia en un senyal sinusoidal de 4 Hz i una amplitud de 5 V pic a pic. En el cas del fitxer, era de 4MB i la configuració de la placa d'adquisició era la d'adquirir amb els 4 canals actius, amb una freqüència de mostreig de 250 Hz i un guany de 1 per cada canal.



Fitxer generat amb l'aplicació principal
 amb el que s'han representat les dades
 dels gràfics de la figura 5.11.

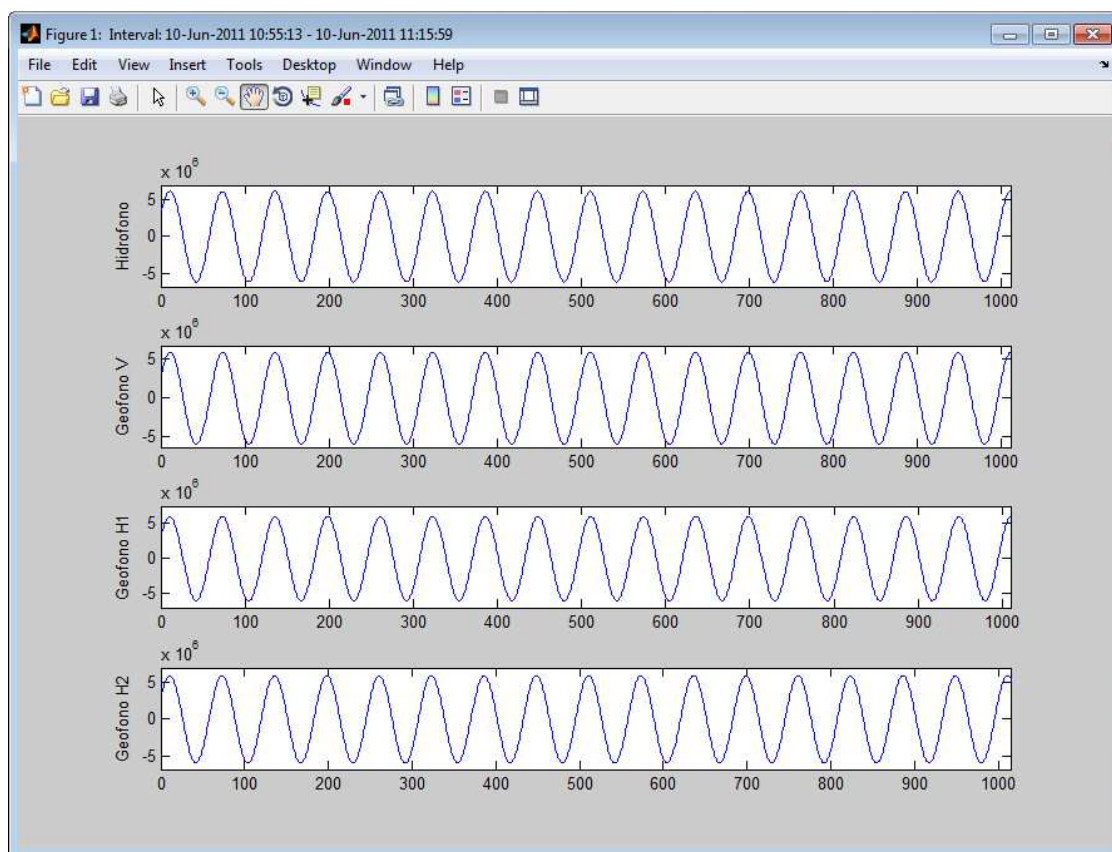


figura 5.11 Mostra dels resultats obtinguts en una adquisició, visualitzat amb el Matlab

Una de les característiques principals del sistema, està en el control del consum requerit per aquest, fer que sigui el menor possible i per sota dels 1000mW és un dels objectius finals a aconseguir per poder augmentar l'autonomia del sistema al màxim. A continuació tenim un llistat de diferents proves realitzades amb les corresponents dades de consum existents, en primer lloc veurem el consum de la placa d'avaluació, la qual gestionava alguns perifèrics innecessaris per la nostra aplicació, i després diferents dades referents a la nova placa del microprocessador.

- 1- Placa d'avaluació alimentada independent i treballant a 266,64 MHz, més la placa d'adquisició alimentada mitjançant la placa d'alimentació DC-DC:

Placa d'avaluació	$I = 926\text{mA}$	$V = 3,3\text{V}$	\rightarrow	$P = 3056\text{mW}$
Placa Adquisició + placa DC-DC	$I = 55\text{mA}$	$V = 6\text{V}$	\rightarrow	$P = 330\text{ mW}$

$$\text{Potència total} = 3056\text{mW} + 330\text{ mW} = \mathbf{3386\text{mW}}$$

- 2- Placa micro alimentada independent, utilitzant la memòria SRAM (la nova memòria incorporada) i treballant a 262,144 MHz, més la placa d'adquisició alimentada mitjançant la placa d'alimentació DC-DC:

Placa micro	I = 370mA	V = 3,3V	→	P = 1221mW
Placa adquisició + placa DC-DC	I = 55mA	V = 6V	→	P = 330 mW

Potència total = 1221mW + 330 mW = **1551mW**

- 3- Tot el sistema alimentat per mitjà de la placa DC-DC, utilitzant la memòria SRAM (per treballar en baixes freqüència) i treballant a 262,144 MHz:

Tot el sistema	I = 409mA	V = 3,9V	→	P = 1595mW
----------------	-----------	----------	---	------------

Potència total = **1595mW**

- 4- Tot el sistema alimentat per mitjà de la placa DC-DC, utilitzant la memòria SRAM (per treballar en baixes freqüència), deshabilitant Ethernet, USB, RNG, I2C i treballant a 262,144 MHz:

Tot el sistema	I = 378mA	V = 3,9V	→	P = 1474mW
----------------	-----------	----------	---	------------

Potència total = **1474mW**

- 5- Tot el sistema alimentat per mitjà de la placa DC-DC, utilitzant la memòria SRAM (per treballar en baixes freqüència) i treballant a 98,304 MHz:

Tot el sistema	I = 308mA	V = 3,9V	→	P = 1201mW
----------------	-----------	----------	---	------------

Potència total = **1201mW**

- 6- Tot el sistema alimentat per mitjà de la placa DC-DC, utilitzant la memòria SRAM (per treballar en baixes freqüència) i treballant en mode *limp* a 32,768 MHz:

Tot el sistema	I = 248mA	V = 3,9V	→	P = 967mW
----------------	-----------	----------	---	-----------

Potència total = **967mW**

- 7- Tot el sistema alimentat per mitjà de la placa DC-DC, utilitzant la memòria SRAM (per treballar en baixes freqüència), deshabilitant Ethernet i treballant en mode *limp* a 32,768 MHz:

Tot el sistema	I = 225mA	V = 3,9V	→	P = 878mW
----------------	-----------	----------	---	-----------

Potència total = **878mW**

- 8- Tot el sistema alimentat per mitjà de la placa DC-DC, utilitzant la memòria SRAM (per treballar en baixes freqüència), deshabilitant Ethernet, cridant a la funció *Low_power_mode_IN* en els instants que no adquireix i treballant en mode *limp* a 32,768 MHz:

Tot el sistema $I = 215\text{mA}$ $V = 3,9\text{V}$ \rightarrow $P = 839\text{mW}$

Potència total = 839mW

5.4. CARACTERÍSTIQUES DE LES PLAQUES

Placa del microprocessador:

Tensió d'alimentació	3,3 V
Relloige del sistema	1 kHz (mín.) 262,144 (màx.)
Memòries	Flash - 64 MB SDRAM - 16 MB SRAM - 2 MB
Funcionalitats	<ul style="list-style-type: none"> - Comunicació amb Compact Flash - Comunicació SPI - Comunicació i2c - Port sèrie RS232 - Port Ethernet - Port USB tipus A - Port USB tipus B - RTC (real time clock) - Pin d'entrada senyal GPS - 4 Temporitzadors - 4 interrupcions externes - 20 línies externes de tipus GPIO

taula 5.1 Característiques placa micro

Placa adquisició i convertidor analògic a digital:

Tensió d'alimentació	3,3 V 2,5 V -2,5 V
Relotge del sistema	32,768 MHz
Freqüències de mostreig	125 Hz 250 Hz 500 Hz 1000 Hz
Canals de mostreig	Fins a 4 canals
Guanys pel senyal adquirida	x1 x2 x4 x8 x16 x32 x64

taula 5.2 Característiques placa adquisició analògic a digital

Placa d'alimentació:

Tensió d'entrada màxima	11,5 V
Tensions de sortida	3,3 V 3,3 V 2,5 V -2,5 V
Corrent màxima de sortida per cada font	1A

taula 5.3 Característiques placa alimentació

Placa backplane:

Aquesta placa és d'ús exclusiu entre les plaques del microprocessador i de la placa ADC degut a que aquest és el canal de comunicació per SPI entre les dues plaques.

6. CONCLUSIONS

Com hem pogut veure en el transcurs d'aquest treball, hem iniciat les activitats a partir de l'electrònica de l'OBS ja desenvolupada pel grup de recerca SARTI-UPC al voltant del microprocessador Motorola MC68332 incorporat al datalogger TT8 fabricat per Onse Computer Corporation. Per facilitar la manipulació dels equips i donar major fiabilitat, es va decidir canviar les estructures mecàniques dels sismòmetres marins, passant de les esferes de vidre a cilindres d'alumini que suporten 600 atmosferes de treball. Aquests canvis, amb les necessitats de bolcar les dades de cada actuació sense obrir els equips, conjuntament amb augmentar prestacions com la autonomia, han provocat un nou disseny de la electrònica dels OBS.

El nou disseny aquí presentat s'ha focalitzat en la utilització del microprocessador ColdFire MCF54455, per les seves altes prestacions i baix consum. El primer pas ha estat validar el programa en una plataforma d'avaluació M54455EVB de Freescale. Tot seguit s'ha dissenyat, construït, i validat el nou sistema d'adquisició i control basat en el ColdFire MCF54455. No solament s'ha migrat a un nou sistema de control, si no que s'han reduït els consums respecte els 1000mW anteriors i per tant s'ha augmentat l'autonomia del nou disseny. Amb la realització d'aquest nou sistema, s'han creat les diferents plaques noves per gestionar i adquirir les dades sísmiques, s'han migrat algunes de les funcions que es podien aprofitar de la versió anterior, com també s'han creat noves funcions. I finalment s'han realitzat les corresponents proves d'adquisició per verificar i confirmar el bon funcionament de tot el sistema un cop encastat en un sol conjunt.

En la realització del codi s'han tingut presents molts petits detalls que no podien passar desapercebuts, petits detalls que podrien arribar a fer que l'aplicació s'executés erròniament, al ser una aplicació de grans dimensions, amb més de 8000 línies de codi i al tenir múltiples funcions (unes 55 funcions) distribuïdes en diferents documents, s'havia de tenir cura de per exemple les variables globals, les quals són cridades en diferents llocs, o com també de l'estructura que es segueix en l'execució, és a dir que cada part del codi ha de tenir unes prioritats concretes, en aquest cas l'adquisició és la part més important i la que s'ha d'executar davant de qualsevol altre part del codi.

En referència a la creació de les plaques del nou disseny, s'han passat pels diferents processos, des del disseny dels diferents esquemes, com a la creació dels diferents PCB's. S'han utilitzat diferents datasheets per treure'n la informació de com es connecten els elements i com crear els esquemes, i s'han extret les recomanacions de com crear el corresponent PCB. Un cop creades les plaques, també s'han realitzat les diferents proves de funcionament, la comprovació de que tota la feina realitzada fins el moment era correcta.

Un cop realitzades les proves prèvies de funcionament, resta realitzar proves de llarga durada per verificar la consistència i la robustesa del sistema, com també el muntatge de l'electrònica amb la resta del sistema, creant així el nou OBS pel grup de recerca SARTI-UPC per adquisicions de dades sísmiques marines. Finalment, l'objectiu principal de tot plegat, queda reduït a que el funcionament de l'equip en el medi marí sigui completament satisfactori segons les especificacions desitjades, i que l'adquisició de les dades sísmiques en el fons marí siguin correctes.

7. BIBLIOGRAFIA

- [1] SharamShariat-Panahi. Aportaciones a la caracterización y diseño de los sistemas de registro y almacenamiento de datos de los sismómetros de fondo marino (OBS). Desembre 2007.
- [2] SharamShariatPanahi. Manual de programación OBS-CTVG. Septembre 2003
- [3] MCF54455EVB User's Manual. Revisió 4. Gener 2008
- [4] MCF54455 Reference Manual. Revisió 3. Maig 2008
- [5] MCF54455 Schematics. Revisió 2. Juliol 2007
- [6] SEASCAN INC. Low-power, high-accuracy time base module (model #SISMTB ver. 4.0). Gener 2005
- [7] Cirrus Logic CS5376A. Low Power Multi-Channel Decimation Filter. Setembre 2003
- [8] FatFs Generic FAT File System Module. http://elm-chan.org/fsw/ff/00index_e.html . Abril 2011
- [9] USB-ML-CF, ColdFire Multilink RevD. Technical Summary. Document #PE3346. Versió 1.1. 2010
- [10] S. Salon, A. Crise, P. Picco, E. de Marinis, O. Gasparini, "Sound speed in the Mediterranean Sea: an analysis from a climatological data set", *Proc. EGU 2003*, pp 833-846, Apr 2003.
- [11] J.Sorribas, J.del Río, S.Shariat-Panahi, C.Dickel, A.Mànuel, E.Trullols. Development of a Synchronization Trigger for the Spanish Oceanographic Ships Based on an Embedded Real-time Java System. *IEEE Trans on Instrumentation and Measurement*. Vol. 58, Pp 3323-3334. Septembre 2009
- [12] ShahramShariat-Panahi, Francisco Alegria, Antoni Mànuel.. Design and Test of a High Resolution Acquisition System for Marine Seismology *IEEE I&M Magazine*. Vol. 12 N°2, Pp 8-15, Abril 2009 ISSN 1094-6969

8. ANNEX

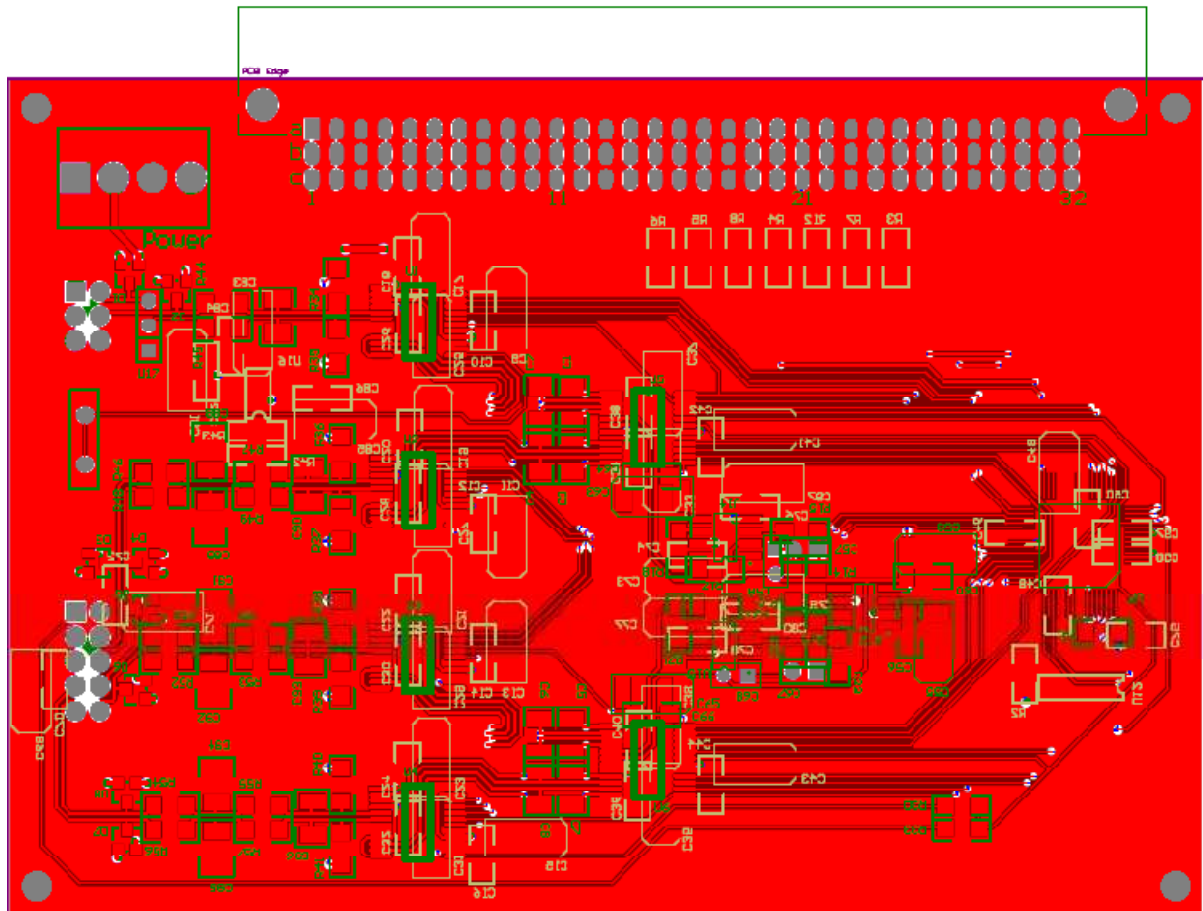


figura 8.1 Placa d'adquisició analògic a digital

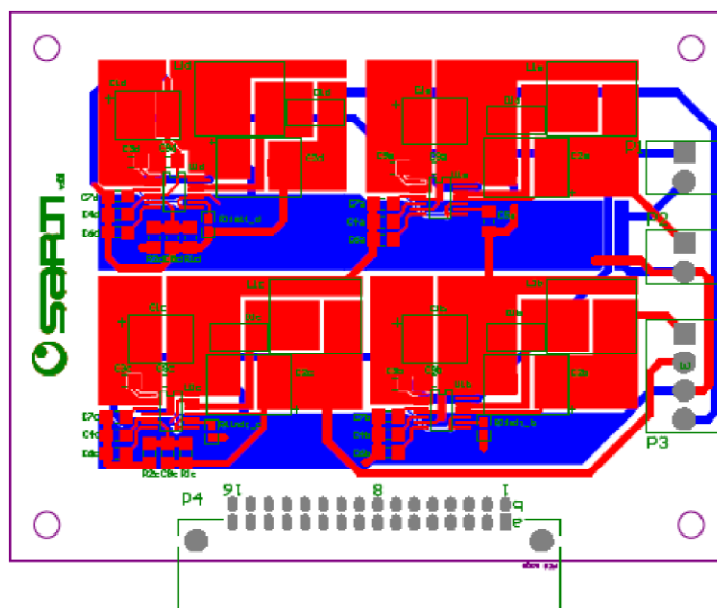


figura 8.2 Placa d'alimentació DC-DC

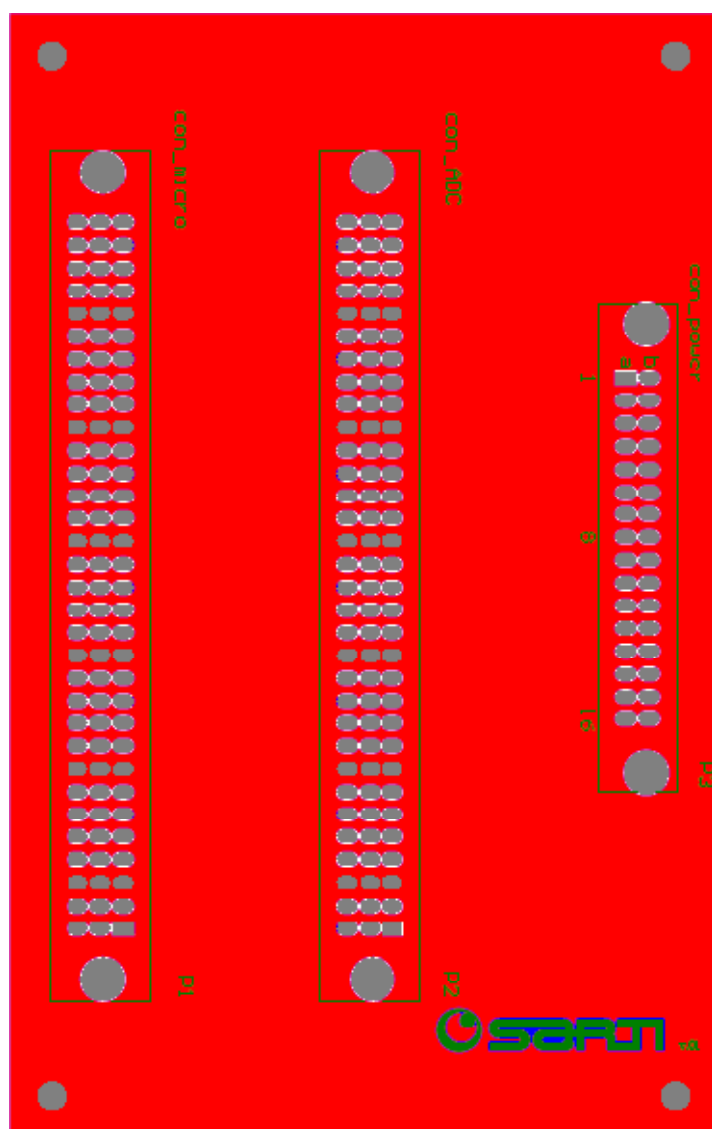


figura 8.3 Placa backplane