



Escola Politècnica Superior
de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

PROJECTE DE FI DE CARRERA

TÍTOL DEL PFC: Espectroradiòmetre intel·ligent basat en dsPIC

TITULACIÓ: Enginyeria de Telecomunicació (segon cicle)

AUTOR: Sergi Pons Freixes

DIRECTOR: Jaume Piera Fernández

DATA: 23 de febrer de 2006

Títol: Espectroradiòmetre intel·ligent basat en dsPIC

Autor: Sergi Pons Freixes

Director: Jaume Piera Fernández

Data: 23 de febrer de 2006

Resum

L'objectiu d'aquest projecte és el disseny i implementació de tota la circuiteria i programari d'una sonda oceanogràfica.

Com a part del projecte de investigació SAMPLER, aquesta sonda ha de ser capaç de prendre dades de pressió i signatures espectrals a diferents profunditats, i enviar-les cap a un PC o PDA.

S'ha realitzat tota la part pertanyent a l'electrònica de la sonda, inclòs la comunicació amb l'exterior, com a culminació d'antics treballs similars, sent aquest el primer projecte que es podrà provar en entorns aquàtics reals. També s'ha desenvolupat íntegrament tot el programari, utilitzant programari lliure, ja que aporta beneficis al camp de la investigació i la recerca.

Ha primat la facilitat d'ús de cara a l'usuari de la sonda, amb una organització modular tant del *hardware* com del *software*, de manera que tot i ser un model funcional es puguin crear noves versions millorades o amb més prestacions amb facilitat.

Title: Espectroradiòmetre intel·ligent basat en dsPIC

Author: Sergi Pons Freixes

Director: Jaume Piera Fernández

Date: February, 23rd 2006

Overview

The target of this project is the design and implementation of the whole circuitry and software of an oceanographic probe.

As a part of SAMPLER investigation project, this probe has to be able to take pressure and spectral data at different depths, and send it to a PC or PDA.

The whole electric board has been developed, including communication, as a culmination of old similar projects, being this project the first one that will be able to be tested on real aquatic environments. Also, the software has been developed, using free software, ideal for the investigation field.

A main feature is the simplicity and usability, with a modular organization of the software and hardware design. So, future improvements and modifications could be applied easily.

ÍNDIX

CAPÍTOL 1. INTRODUCCIÓ	1
1.1.Entorn: projectes VARITEC i SAMPLER.....	1
1.2.Abast i objectius.....	1
1.3.Antecedents.....	3
1.3.1.Fase I: TFC Disseny d'un sistema de mesures espectralradiomètriques.....	3
1.3.2.Fase II: TFC Disseny d'un espectralradiòmetre.....	4
CAPÍTOL 2. ESPECIFICACIONS	5
2.1.Especificacions generals.....	5
2.2.Part física: electrònica i mecànica.....	6
2.3.Part de programació: dsPIC i PC.....	7
CAPÍTOL 3. DESCRIPCIÓ DEL HARDWARE	9
3.1.Sensor de pressió.....	9
3.1.1.Característiques tècniques.....	9
3.1.2.Muntatge realitzat.....	10
3.1.3.Calibració del sensor.....	11
3.2.L'espectralradiòmetre.....	12
3.2.1.Característiques tècniques.....	12
3.2.2.Calibració.....	14
3.3.Acondicionament dels senyals.....	15
3.3.1.L'amplificador operacional Rail-to-Rail.....	16
3.3.2.Característiques tècniques.....	16
3.3.3.Muntatge realitzat.....	17
3.3.4.Filtre antialiasing.....	18
3.4.Digitalització de les dades.....	19
3.4.1.Característiques tècniques.....	19
3.4.2.Control i Comunicació SPI.....	19
3.4.3.Muntatge realitzat.....	20
3.5.Comunicació RS-485.....	21
3.5.1.Característiques tècniques.....	22
3.5.2.Muntatge realitzat.....	22
3.6.El microcontrolador dsPIC.....	23
3.6.1.Característiques tècniques.....	23
3.6.1.Característiques generals.....	23
3.6.2.Ports E/S.....	24
3.6.3.Perifèric SPI.....	25
3.6.4.Perifèric UART.....	25
3.5.Sistema general – Placa final.....	25
3.5.1.Creació de components.....	25
3.5.2.Disseny del Esquemàtic.....	27
3.5.3.Disseny del PCB.....	28

3.5.4. Configuració de la fresadora.....	28
CAPÍTOL 4. SOFTWARE.....	29
4.1. Programari del dsPIC.....	29
4.1.1. Organització del codi i dependències.....	29
4.1.2. Explicació i comentaris.....	31
4.2. Programari del PC.....	33
4.2.1. Organització del codi i dependències.....	33
4.2.2. Explicació i comentaris.....	34
CAPÍTOL 5. ASPECTES MEDIAMBIENTALS.....	37
5.1. Concepció del projecte.....	37
5.2. Construcció del projecte.....	37
5.3. Explotació.....	37
5.4. Desmantellament.....	38
CAPÍTOL 6. CONCLUSIONS.....	39
6.1. Estat actual.....	39
6.2. Desenvolupament futur.....	39
6.2.1. Alternatives a l'espectroradiòmetre.....	40
6.2.2. Sistemes de xarxes de sensors.....	40
BIBLIOGRAFIA.....	42
ANNEX.....	43
A.1. El pont de Wheatstone.....	43
A.2. Algorisme PID.....	44
A.3. Pressupost i material.....	45
A.4. Placa PCB.....	46
A.5. Codi PC.....	49
A.5.1. ssgrafic01c.py.....	49
A.6. Codi dsPIC.....	52
A.6.1. main.c.....	52
A.6.2. ad977.c.....	54
A.6.3. hamamatsu.c.....	55
A.6.4. init_basic.c.....	56
A.6.5. max485.c.....	57
A.6.6. spi.c.....	57
A.6.7. timernou.c.....	58
A.6.8. uart.c.....	58

ÍNDIX DE FIGURES I TAULES

Fig. 1.1: Diagrama de blocs del funcionament de la sonda.....	2
Fig. 1.2: Possible desplaçament de la sonda. L'adquisició es realitza durant l'ascens.....	2
Fig. 1.3: Primer prototip de prova de l'espectroradiòmetre.....	3
Fig. 1.4: Segon prototip, amb dimensions adaptades a la sonda.....	4
Fig. 3.1: Vista externa del sensor de pressió.....	9
Taula 3.1: Sortida del sensor amb una excitació de 4 mA.....	10
Fig. 3.2: Esquema del interior del sensor de pressió.....	10
Fig. 3.3: Resultat de la calibració del sensor de pressió. La gràfica superior mostra les mesures obtingudes (+), i la recta de regressió calculada (línia vermella). La gràfica inferior indica la diferència entre la mesura i la sortida ideal.....	11
Fig. 3.4: Visió del interior del sensor de llum. S'aprecia com arriba la llum a través de la fibra òptica, i es descompon al reflectir-se al substrat. Cada component va a parar a un fotodíode diferent.....	12
Fig. 3.5: Esquema intern del espectroradiòmetre i encapsulat.....	13
Fig. 3.6: Cronograma de les senyals de control de l'espectroradiòmetre. El temps entre pols consecutius de ST ens marca el temps d'integració. La sortida de cada fotodíode correspon a un cicle de rellotge.....	14
Fig. 3.7: Nivell de sensibilitat de l'espectroradiòmetre per cada longitud d'ona. 15	
Fig. 3.8: Calibració del sensor de pressió. A l'eix d'abscisses hi ha el número de fotodíode, i en el d'ordenades les longituds d'ona. L'objectiu d'aquestes mesures és que es correspon cada díode amb una longitud d'ona concreta, de manera que es reparteixi l'espectre uniformement.....	15
Fig. 3.9: Patinatge de l'amplificador AD8603.....	17
Taula 3.2: Característiques elèctriques del AD8603 @ $V_s = 5V$	17
Fig. 3.10: Esquema de muntatge de l'amplificador.....	18
Fig. 3.11: Encapsulat del convertidor A/D.....	19
Fig. 3.12: Cronograma general de tot el procés d'adquisició d'una mostra de l'A/D.....	20
Fig. 3.13: Cronograma de la comunicació SPI de l'A/D. Els senyals R/C i DCLK son generats pel dsPIC, i BUSY i DATA per l'AD977.....	20
Fig. 3.14: Connexions i elements necessaris per la part analògica del convertidor. D'aquesta manera es defineix un marge a V_{in} de 0 a 5V.....	21
Fig. 3.15: Connexions necessàries entre l'AD977 i el dispositiu mestre (dsPIC).	21
Fig. 3.16: Esquema del encapsulat del MAX485. Es pot observar com internament els pins DE i RE habiliten /deshabiliten la recepció/transmissió....	22
Fig. 3.17: Esquema de la comunicació RS-485. Es pot observar com es trenca el cable i s'afegeixen dos resistències com a terminadors per tal de millorar el rendiment del sistema.....	22
Fig. 3.18: Encapsulat del dsPIC. Es pot observar el gran nombre de ports E/S sortida de que disposa, i que molts d'ells tenen nomenclatures especials ja que donen accés directe a perifèrics de comunicació concrets.....	24
Fig. 3.19: Esquema de la placa principal del sistema. Inclou el dsPIC, la comunicació RS-485, tota l'etapa del sensor de pressió i l'alimentació.....	26
Fig. 3.20: Esquema de la placa secundària del sistema. Inclou l'espectroradiòmetre i tota la seva etapa d'aquisició.....	27
Fig. 3.21: Diagrama de tot el sistema. Es mostren els principals components i	

les comunicacions/interaccions entre ells.....	27
Fig. 4.1: Diagrama de dependències i funcions del programa del dsPIC.....	30
Fig. 4.2: Exemple de l'ajust automàtic mitjançant l'algorisme PID. A cada iteració disminueix l'error.....	31
Fig. 4.3: Diagrama de flux del funcionament del dsPIC.....	32
Fig. 4.4: Diagrama de dependències i classes del programa del PC.....	34
Fig. 4.5: Captures de pantalla del programa de control de la sonda en diferents sistemes operatius i en diferents estats d'operació.....	35
Fig. 4.6: Diagrama de flux del funcionament del programa del PC.....	36
Fig. A.1: Pont de Wheatstone per mesurar una resistència.....	43
Fig. A.2: Representació gràfica de tres adquisicions contigües de l'espectroradiòmetre.....	45
Taula A.3: Pressupost aproximat de la sonda.....	46
Fig. A.3: En verd la cara bottom i en vermell la cara top de la placa principal. .	47
Fig. A.4: En verd la cara bottom i en vermell la cara top de la placa secundària.	48
Fig. A.5: Fotografia de la cara top de les dues plaques. Es poden observar el dsPIC i l'espectroradiòmetre.....	49

CAPÍTOL 1. INTRODUCCIÓ

1.1. Entorn: projectes VARITEC i SAMPLER

Per tal de poder entendre la finalitat i utilitat d'aquest projecte, és necessari comentar prèviament el projecte d'investigació VARITEC, i el subprojecte que en forma part SAMPLER.

El projecte VARITEC coordina els subprojectes RETOC i SAMPLER, amb la intenció d'abordar l'estudi multiescala de la relació entre la variabilitat de la turbulència i l'estructura i la dinàmica de l'ecosistema coster del mediterrani noroccidental. Per poder complir aquest objectiu, s'han de realitzar moltes tasques i molt diverses, des de la creació de sistemes per realitzar les mesures fins el desenvolupament de programari de tractament de dades.

La part tractada en aquest document és la referent al subprojecte SAMPLER (liderat per la UPC): el desenvolupament d'un sistema d'adquisició i anàlisi de dades oceanogràfiques de Microestructura física i perfils hiperespectrals. Un dels objectius del projecte general és avaluar l'efecte de la turbulència en la distribució dels organismes planctònics. Per avaluar aquest efecte cal prendre mesures relacionades amb la turbulència (microestructura física) i mesures que permetin identificar la presència d'organismes planctònics. Molts dels organismes planctònics tenen diferents propietats òptiques i per això es plantegen les mesures de l'espectre lumínic amb alta resolució (hiperespectral) com una mesura que permetrà caracteritzar la distribució d'aquests organismes. Per tant, una de les quals tasques principals es porta a terme en aquest projecte: el desenvolupament d'una sonda per l'adquisició de dades de microestructura i control d'un espectroradiòmetre.

Com a dades de microestructura s'entenen les mesures de conductivitat, pressió i temperatura, pressos al mateix temps i amb una resolució vertical de l'ordre de mil·límetres. Les mesures espectroradiomètriques han de ser a escala fina, resolució vertical de l'ordre de centímetres. Aquests paràmetres han de permetre identificar i caracteritzar les estructures físiques i biològiques del medi.

1.2. Abast i objectius

Un cop ja s'ha definit el marc que envolta el projecte, és necessari establir quin serà el seu abast. Intentar realitzar tota les tasques definides el l'apartat anterior en un temps de quatre mesos és inviable si es volen obtenir uns resultats satisfactoris i fiables. Per aquesta raó, aquest projecte és només una part de tot el procés de desenvolupament de la sonda. Tot i això, és el disseny més realista i complet que s'ha fet fins al moment, fins al punt de que està preparat un cop finalitzat per realitzar proves de camp en entorns reals, fet inèdit fins al moment.

L'esquema de la sonda completa es pot observar a la Fig. 1.1. L'objectiu és que

s'arribin a completar tots els blocs, no de manera definitiva, però sí amb un nivell tal que els canvis necessaris siguin trivials. Per aquest motiu, es treballarà tot l'aspecte de la comunicació, l'adquisició de dades de l'espectroradiòmetre i d'un sensor de pressió. Afegir la resta de sensors de temperatura i conductivitat comporta només duplicar (amb petites modificacions) la part del sensor de pressió, i modificar el protocol de comunicacions amb l'exterior de la sonda perquè inclogui les noves dades i siguin enviades a la freqüència que pertorqui.

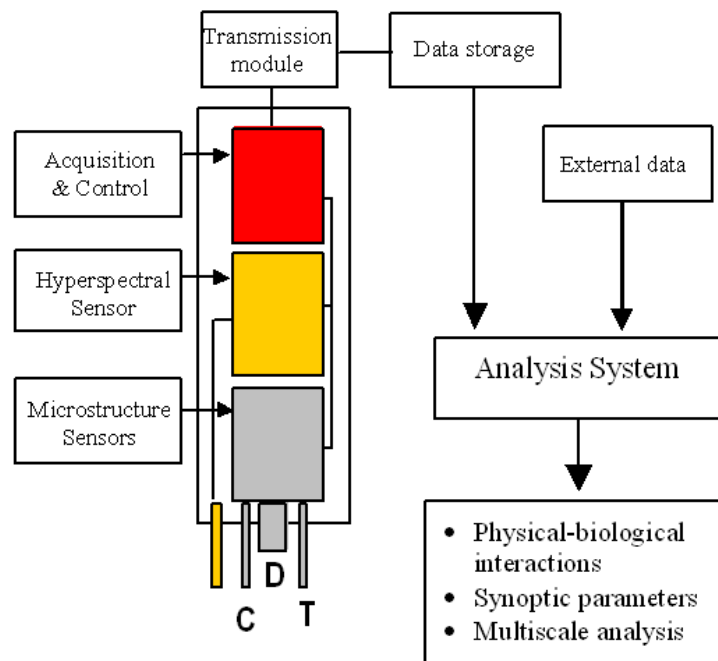


Fig. 1.1: Diagrama de blocs del funcionament de la sonda.

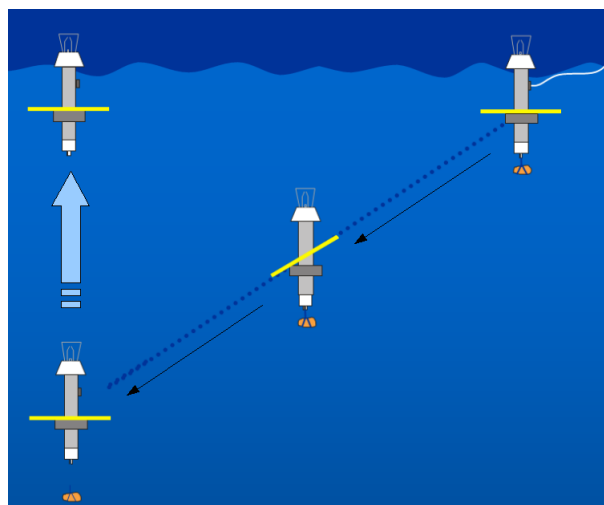


Fig. 1.2: Possible desplaçament de la sonda. L'adquisició es realitza durant l'ascens.

A la Fig. 1.2, es pot veure el possible recorregut que seguirà la sonda per tal d'adquirir les dades. Es fa arribar fins a la profunditat desitjada mitjançant un disc basculant (que permet mantenir la posició vertical i l'estabilitat) i un pes. Un cop s'ha arribat al punt establert, s'allibera el pes i es modifica la direcció del disc, de manera que la sonda comença a ascendir per flotabilitat dels seus elements cap a la superfície. Durant aquest trajecte vertical és quan es realitza la presa de mesures útils pels projectes. S'ha optat per que durant l'ascens s'enviïn les dades cap a la superfície, de manera que puguin ser emmagatzemades en un PC o PDA, o consultades *in-situ* si fos necessari.

Al Capítol 2 es fa un anàlisi més exhaustiu i acurat de tots els objectius que ha de complir el projecte. Degut a que alguns aspectes s'han tractat més en profunditat en alguns projectes anteriors, al següent apartat es fa una menció d'aquests i la informació que s'hi pot trobar.

1.3. Antecedents

1.3.1. Fase I: TFC Disseny d'un sistema de mesures espectralradiomètriques

Aquest projecte es va realitzar durant els mesos de setembre de 2003 fins el gener de 2004, per Manel Merchán, i l'autor d'aquest document, Sergi Pons. Els objectius van ser una primera presa de contacte amb els microcontroladors PIC (concretament un PIC18) i amb l'espectroradiòmetre. Es va crear un sistema que es pot considerar l'embrió de la sonda. El microcontrolador controlava i mostrejava les dades amb un AD intern de 10 bits, i era capaç de guardar-les en una EEPROM externa o d'enviar-les cap al PC.

A més a més, es va realitzar una interfície amb Matlab que permetia el visualitzat de les dades i el seu emmagatzemament, al mateix temps que mitjançant el port sèrie del PC es comunicava amb el microcontrolador per donar-li les ordres corresponents.

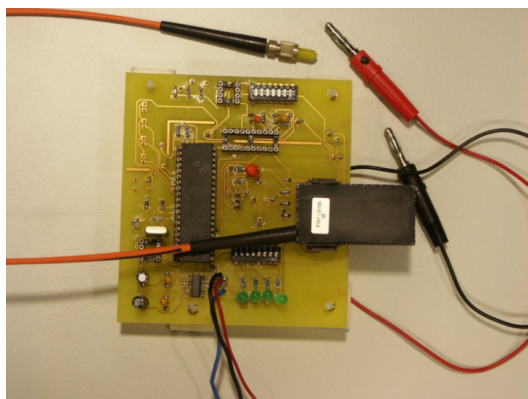


Fig. 1.3: Primer prototip de prova de l'espectroradiòmetre.

En tot moment va ser considerat un prototip de laboratori, i no es van tenir en compte aspectes mecànics o d'adaptació al medi aquàtic. A la Fig. 1.3 es pot observar la placa dissenyada.

1.3.2.Fase II: TFC Disseny d'un espectroradiòmetre

Durant els mesos de febrer fins setembre de 2005, Maribel Pérez i Núria Pujol van seguir el desenvolupament del projecte. Es va reemplaçar el PIC18 per un dsPIC30F6012, que oferia més potència de càlcul i sobretot més capacitat de connexió. L'espectroradiòmetre va ser substituït per un nou model, més senzill de controlar i amb una millor resposta freqüencial. La EEPROM va ser desestimada, ja que es va decidir que no seria utilitzada en la sonda real. En canvi, es va afegir un AD extern (controlat també pel dsPIC), per tal d'augmentar la resolució de les dades fins a 16 bits. Aquesta vegada, la placa realitzada ja es va fer pensant en introduir-la en la sonda, respectant unes mides concretes (veure Fig. 1.4).

Respecte la part del programari de control de la sonda i representació de dades, es va millorar en el sentit de simplificació i optimització del codi en Matlab. A més, es va afegir una funcionalitat que permetia captar les dades de manera continua, i regulava el temps d'integració del l'espectroradiòmetre (veure apartat 3.2) de manera automàtica.

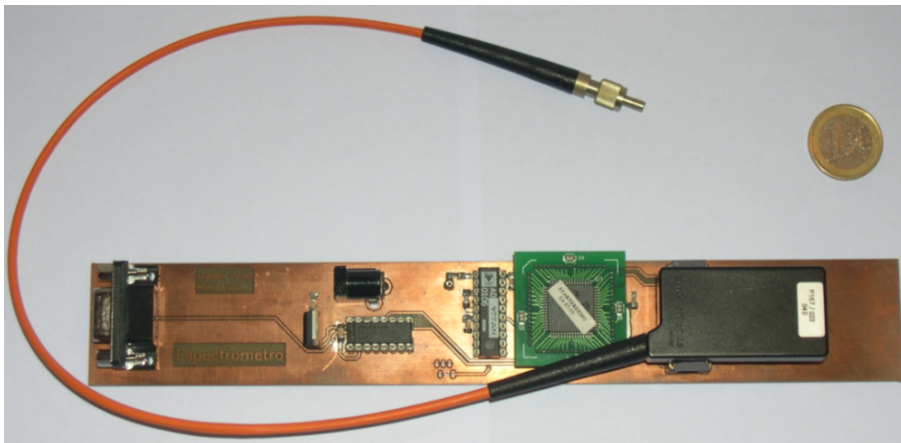


Fig. 1.4: Segon prototip, amb dimensions adaptades a la sonda.

CAPÍTOL 2. ESPECIFICACIONS

2.1. Especificacions generals

Abans de començar a dissenyar i implementar les diferents parts del projecte, s'han de tenir clars tots els requisits que ha de complir. Per tant, en aquest primer apartat es fa un llistat dels requeriments imposats al projecte:

- Requisits de la placa:
 - Ha de ser capaç d'introduir-se i acoblar-se al tub estanc de PVC del que ja disposa el departament de TSC.
 - Ha de ser capaç d'adquirir les dades de l'espectroradiòmetre S8378 – 256N/SPL de la marca HAMAMATSU, amb resolució de 16 bits.
 - Ha de ser capaç d'adquirir les dades del sensor de pressió PAA-10 de la marca KELLER, que ja es troba acoblat al tub estanc de PVC, amb resolució de 16 bits.
 - La major part de lògica de control i processament dels senyals adquirits l'ha de realitzar un microcontrolador dsPIC30F6012.
 - La font d'alimentació ha de ser externa, mitjançant el cable que es connecta al tub estanc de PVC.
 - Les dades s'han d'enviar a un PC a través del cable que es connecta al tub estanc de PVC.

- Requisits del programari:
 - La placa ha de ser completament controlable des d'un PC de manera externa, sense necessitat d'accedir físicament a la sonda, exceptuant el cable pel qual han de circular les dades.
 - S'han d'adquirir les dades de forma continua durant períodes determinats de temps, en temps real.
 - No és necessària una representació/visualització en temps real de les dades obtingudes.
 - Les dades s'han d'emmagatzemar en el PC, per poder consultar-les posteriorment.

A partir d'aquesta llista inicial, en els següents apartats es prenen una sèrie de decisions per tal de concretar encara més l'abast del projecte. En un projecte és vital tenir tots els objectius a complir llistats de manera simple i clara, ja que qualsevol dubte que aparegui posteriorment pot implicar haver de fer canvis en els dissenys i implementacions anteriors, resultant en un cost de temps molt elevat. I no només això, sinó que aquesta llista de punts a complir serveix al mateix temps per estructurar les diferents tasques a realitzar, la seva interrelació i independència.

Ja es poden divisar dos grans blocs de treball diferenciats, un completament físic i electrònic, i un altre més abstracte, de programació i informàtica. Tot i així, tots dos aspectes estan intrínsecament relacionats, sobretot en el cas del dsPIC. Això comporta, com es deia en el paràgraf anterior, que es prenguin totes les decisions abans de fer res, ja que és molt probable que l'apartat

mecànic condicioni a la programació i viceversa. A continuació, doncs, se'n realitza un anàlisi més exhaustiu.

2.2.Part física: electrònica i mecànica

El major condicionant que es troba en aquest apartat és la reutilització del tub de PVC estanc. Condiciona tant les mides de la placa com la situació dels seus elements, i l'elecció d'alguns d'ells. Afortunadament es disposa de complet accés físic al tub de PVC, cosa que permet estudiar-lo amb detall i no trobar-nos amb cap sorpresa. A més del tub en si, s'ha de preveure la connexió amb el sensor de pressió i el cable de dades i alimentació, ambdós fixats a un extrem del tub i amb una longitud limitada. Un disseny antic utilitzava tres plaques diferents, interconnectades entre elles, per tal implementar tota l'electrònica. Degut a la miniaturització dels components i la substitució de tecnologia analògica per digital, en aquest cas només se'n necessitaran dos.

Els altres grans condicionants són els sensors dels quals s'ha d'adquirir les dades, i la utilització del dsPIC. Afortunadament, degut a l'experiència prèvia en el TFC comentat a l'apartat 1.3.1 i als coneixements adquirits treballant com a becari al departament de TSC, es disposa d'un *Know-How* que permet fer una ràpida visualització del sistema en general.

Per una banda, el dsPIC té capacitat de processament i de connectivitat suficient com per fer-se càrrec de tot el control de la sonda i la comunicació amb l'exterior. Té potència per fer càlculs amb temps real i modificar els paràmetres de funcionament del sistema per tal que l'adquisició de dades sigui òptima, com per exemple variant el temps de integració del sensor de llum (veure apartat 3.2.1). Tampoc es troba limitació en els protocols de comunicació a seguir, donant més flexibilitat.

Respecte els sensors, l'espectroradiòmetre ha estat testejat anteriorment al departament de TSC, i la principal dificultat és referent a la seva mida i fer la connexió de la fibra de vidre que incorpora amb l'exterior de la sonda. Aquesta es farà mitjançant el col·lector de llum *fiber-optic plane irradiance collector* de HOBI Labs que ha adquirit el departament expressament pel projecte. El sensor de pressió amaga en principi més incògnites, però el fabricant mostra que es tracta simplement d'un pont de Wheatstone. El seu funcionament doncs ja ha estat analitzat en assignatures de la universitat, i per tant es pot abordar amb seguretat.

Sobre la connexió amb el PC, degut a la gran distància del cable, es desprecia el protocol RS-232 utilitzat anteriorment i s'implementarà el RS-485. El fet de basar-se en senyals diferencials ofereix a més a més una tolerància a interferències superior. S'ha de tenir en compte, doncs, que serà necessari algun dispositiu en el PC que suporti també aquest protocol, ja sigui una placa PCI, PCMCIA o un adaptador en el port sèrie convencional.

La llista de requisits resultant de la part *hardware* és:

- Realitzar el disseny de tota la circuiteria en dos plaques separades, interconnectades entre elles, amb les característiques:
 - Mides de 18,2 x 3,5 cm.
 - La separació entre elles ha de ser de 20 mm.
 - El connector amb el cable extern ha d'estar a 11,8 cm d'un dels marges.
 - El connector amb el sensor de pressió ha d'estar a 9 cm d'un dels marges, que ha de ser el mateix que el connector amb el cable extern.
- L'alimentació de tot el sistema ha de ser de 5V, menys pel sensor de pressió que serà de 12V.
- Utilitzar conversors AD externs, que es comuniquin amb el dsPIC mitjançant el protocol SPI.
- Ajustar i amplificar l'entrada de senyal als conversors per tal d'aprofitar tot el marge dinàmic.
- Utilitzar el protocol RS-485 per la comunicació amb l'exterior.

2.3.Part de programació: dsPIC i PC

Com es comenta en l'apartat anterior, ja es disposa de coneixement del funcionament amb detall del dsPIC. Això permet que moltes de les seves funcionalitats no hagin de ser programades de nou, sinó que es poden reutilitzar llibreries ja creades i testejades. Per tant, més que centrar-se en el control dels perifèrics del dsPIC, es poden focalitzar els esforços en la lògica de control, protocol de comunicació amb el PC, i tot el que conforma la intel·ligència de la sonda. Com a plataforma de desenvolupament del dsPIC, s'utilitzarà el IAR Embedded Workbench for dsPIC de IAR Systems, ja que el departament encara en disposa de llicència.

Un dels requisits inicials és que la sonda ha de ser capaç d'adquirir dades de forma continua durant un període de temps concret. Mitjançant una reunió amb el director de projecte, s'ha acordat que l'inici de l'adquisició serà manualment, i aquesta es detindrà quan hagi adquirit un nombre determinat de mostres o hagi arribat a certa profunditat, o el que és el mateix, registri una pressió concreta.

La part del programari del PC és una de les més innovadores respecte el que s'havia fet anteriorment. Fins ara s'havia estat lligat al Matlab de MathWorks, però un objectiu possible però no obligatori era canviar a un entorn més lliure. El fet que de no necessitar un programa propietari basat en llicències el fa molt més atractiu per a la investigació i l'ensenyança. A més a més, tot i que en un principi només es defineix que la interfície home-sonda és un PC, és interessant contemplar que pugui ser un aparell més pràctic d'utilitzar en un vaixell com una PDA, ja que és l'escenari real d'utilització de la sonda.

Seguint amb el programari del PC, tot i que no és necessari que sigui capaç de representar les dades obtingudes per pantalla, si que ha de ser el màxim de senzill i fàcil d'utilitzar, de manera que l'ús d'una interfície gràfica prima per sobre de la línia de comandes. A més, serà més fàcil d'utilitzar en dispositius

portàtils com es deia en el punt anterior.

Per motius personals, s'ha decidit transformar aquest objectiu possible en objectiu obligatori, i desenvolupar íntegrament tot el codi destinat a utilitzar en el PC, utilitzant programari lliure.

D'aquestes decisions, la llista d'objectius ha complir resultant és:

- El dsPIC ha d'implementar la major part de intel·ligència de control de la sonda, de manera que no sigui necessària per fer-ho una comunicació continua amb el PC.
- Per la programació del dsPIC utilitzar l'entorn de desenvolupament IAR Embedded Workbench for dsPIC de IAR Systems.
- La programació del PC:
 - Ha d'estar basada en sistemes de programari lliure, sense cap necessitat de programes propietaris.
 - Ha de ser multiplataforma, preveient una futura implementació en altres sistemes com PDAs.
- La interfície amb l'usuari ha de ser gràfica, senzilla i intuïtiva, permetent realitzar totes les funcions de la sonda:
 - Adquirir les dades des d'un instant iniciat manualment, fins un nombre determinat de mostres o una profunditat prefixada.
 - Guardar les dades en un fitxer de text pla.

CAPÍTOL 3. DESCRIPCIÓ DEL HARDWARE

En aquesta part del projecte es tracta tot el referent als components electrònics, la circuiteria i altres elements físics de la sonda. Alguns d'ells ja han estat analitzats exhaustivament en publicacions anteriors (veure apartat 1.3), de manera que aquí només se'n farà una breu menció. Es centrarà en els components més novedosos i el fet que s'ha d'integrar en un tub de PVC estanc que havia format part d'una sonda real.

3.1. Sensor de pressió

El sensor de pressió PAA-10 del fabricant KELLER ja es trobava incorporat a l'estructura de la sonda. En aquesta nova fase del projecte encara es considera que compleix amb les expectatives desitjades i, com el seu funcionament es correcte, es seguirà utilitzant.

3.1.1. Característiques tècniques

Aquest element és bàsicament un transductor de pressió piezoresistiu. La mesura de la pressió es fa mitjançant un xip de silici piezoresistiu d'alta sensibilitat, protegit de les influències ambientals mitjançant una carcassa d'acer inoxidable segellat amb un diafragma concèntricament acanalat. La carcassa està omplerta d'oli de silicona per assegurar la correcta transferència de pressió del diafragma cap al component sensor. Totes les parts de metall en contacte amb el medi de pressió estan fetes d'acer inoxidable 316L, i els elements de la carcassa soldats i ajuntats al buit. El seu circuit equivalent (Fig. 3.2) és un pont de Wheatstone (veure Annex A.1 per més informació sobre aquest tipus de circuit), per tant la sortida serà un diferència de potencial.



Fig. 3.1: Vista externa del sensor de pressió.

Segons el fabricant, internament segueix l'esquema de la Fig. 3.2, amb les característiques següents:

- Pressió absoluta, 0 Bar en el buit.
- Resistència del pont @25°C: 3500Ω±20%
- Alimentació constant de 4mA nominals, 5mA màxim.
- Resolució típica del 0.25% del marge dinàmic (FS).
- Offset@25°C menor de 20 mV, corregible amb una resistència externa (R5).
- Temperatures de funcionament: De -30°C a 100°C.
- Durabilitat de 100x106 cicles de Full Scale@25°C.

Taula 3.1: Sortida del sensor amb una excitació de 4 mA.

Pressió (Bar)	0.1	0.2	0.5	1	2	5	10	20
Sortida típica (mV)	85	135	250	400	600	900	900	900

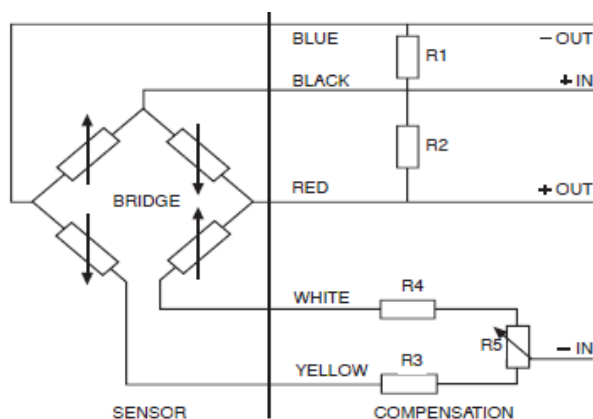


Fig. 3.2: Esquema del interior del sensor de pressió.

3.1.2. Muntatge realitzat

El fabricant especifica totes les dades sobre el comportament del sensor a 4 mA, però tot i això és possible treballar amb una alimentació menor. En aquest projecte, per qüestions d'alimentació de la resta del sistema, no es pot generar aquesta corrent de 4 mA. Al treballar amb una bateria externa de 15 V, mitjançant un regulador de tensió es pot obtenir una alimentació de 12 V. Combinats amb els 3500 Ω aproximats del pont (variables en funció de la pressió), es presenta uns valors de corrent pròxims als 3.4 mA. Com a resultat, es té una disminució del marge dinàmic (valor de tensió mínima i màxima de sortida més propers), i per tant menys resolució (un canvi en la pressió genera un canvi més petit en la tensió de sortida).

3.1.3. Calibració del sensor

Tot i disposar de les especificacions del fabricant, donat que no es treballa amb un corrent constant (el que es té constant és la tensió), i que el sensor ha estat sense utilitzar-se durant molt de temps, ha estat necessari una comprovació del seu funcionament. Al mateix temps, s'ha aprofitat per calibrar el sensor amb les característiques del sistema, de manera que es pugui obtenir els paràmetres del seu funcionament, necessaris per implementar la part d'acondicionament de senyal i la seva digitalització.

El procés s'ha realitzat generant pressió amb un PC6 Pressure Calibrator de SI Pressure Instruments, i mesurant la sortida generada. Els resultats de les mesures es poden comprovar a les gràfiques de la Fig. 3.3. Els rangs de pressió pels quals s'ha testejat el sistema han estat els equivalents als que es veurà sotmesa la sonda en l'aplicació real, o sigui, de 0 a 50 metres sota l'aigua.

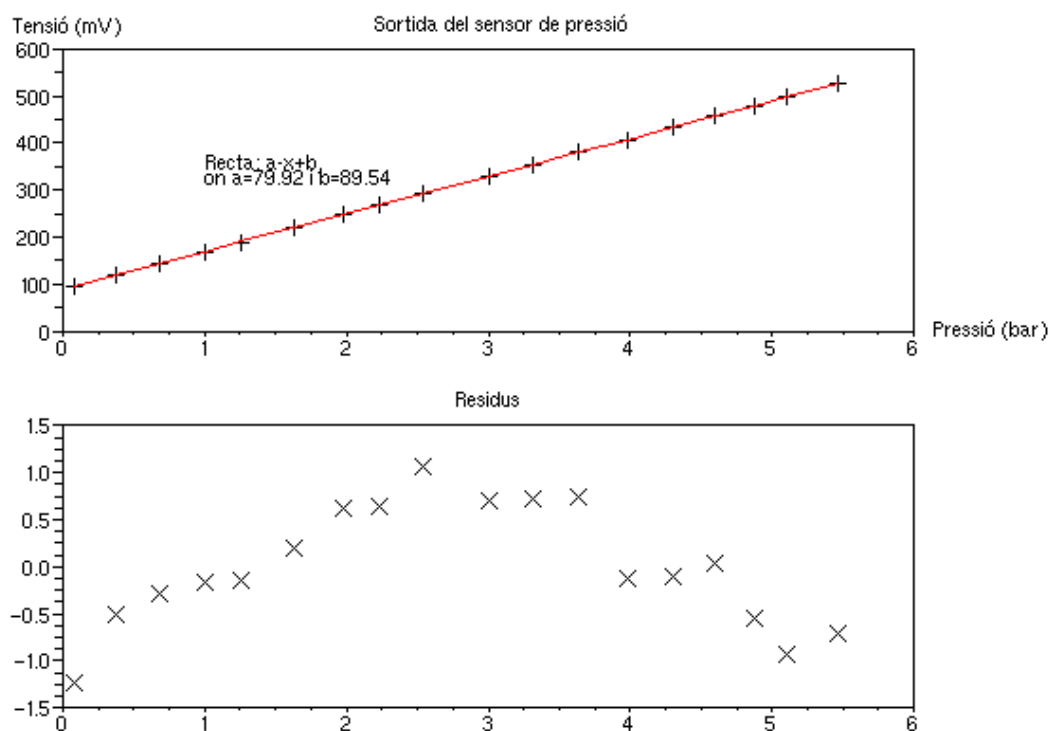


Fig. 3.3: Resultat de la calibració del sensor de pressió. La gràfica superior mostra les mesures obtingudes (+), i la recta de regressió calculada (línia vermella). La gràfica inferior indica la diferència entre la mesura i la sortida ideal.

Es pot caracteritzar el comportament del sensor amb una recta, on cada increment de pressió de 1 Bar genera una variació a la sortida aproximada de 79.92 mV, o sigui, una relació de 79.92 mV/Bar, amb un desplaçament positiu (*offset*) de 89.54 mV. El marge de treball serà de 0 a 5 Bar, i per tant s'obté una sortida de 89.54 a 489.14 mV. No s'ha pogut mesurar la resolució mínima del

sensor, ja que en el muntatge realitzat per fer la calibració no era 100% estanc i estable davant dels canvis de pressió i no permetia controlar variacions petites. L'efecte d'un *offset* no és rellevant, ja que mitjançant *software* es corregirà. El que es farà és tenir en compte quant ha variat la pressió des del punt que fixi l'usuari com a pressió 0.

3.2.L'espectroradiòmetre

L'altre sensor del qual es desitja adquirir les dades és el S8378 – 256N/SPL de HAMAMATSU. Aquest component és un espectroradiòmetre, o sigui, mitjançant el qual podem la firma espectral a partir de la llum captada. S'ha estat utilitzant en el departament de TSC durant els últims mesos, així com estudiant el seu comportament i calibració. En els projectes comentats a l'apartat 1.3 es pot trobar molta informació al respecte, de manera que aquí només es farà un resum de les característiques més importants.

3.2.1.Característiques tècniques

El model concret a estudiar en aquesta fase del projecte, és un sensor d'imatge lineal basat en tecnologia CMOS. Capta la llum mitjançant una fibra òptica, que a la sonda esta connectada a un difusor. Aquesta llum es condueix fins a l'interior del sensor, on mitjançant un substrat de cristall es descompon en les diferents longituds d'ona que la componen. El cristall està dissenyat de tal forma que cada component, formada per unes longituds d'ona concretes, es absorbidada per un díode. En funció de l'excitació de cada díode, es pot quantificar la intensitat lumínica d'aquella component (o sigui, d'aquella part concreta de l'espectre).

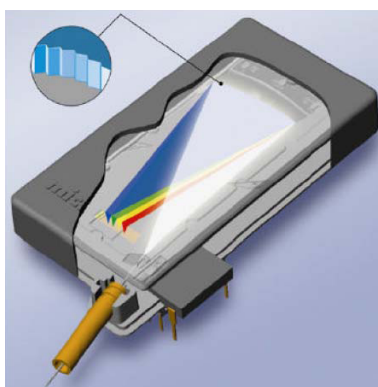


Fig. 3.4: Visió del interior del sensor de llum. S'aprecia com arriba la llum a través de la fibra òptica, i es descompon al reflectir-se al substrat. Cada component va a parar a un fotodíode diferent.

Aquest funcionament es pot caracteritzar per dos paràmetres. Un, és el rang de longitud d'ona a mesurar, i amb quin nombre de díodes. Els valors en aquest cas serien de 200-1000nm i 256 díodes, obtenint que cada díode capta un ample de 3.125 nm. L'altre paràmetre és el temps d'exposició. Com més temps està exposat un díode a una intensitat concreta, més es va excitant fins que es pot arribar a saturar.

Un cop ja s'ha vist com funciona internament el sensor, a continuació es detalla com s'hi interactua, que li és necessari per funcionar i com ofereix les dades. Tot i que l'encapsulat (Fig. 3.5) presenta un gran nombre de potes, només en són necessàries un grup reduït d'aquestes. L'alimentació es limita a 5V (amb un consum de potència de 25 mW) i una pressa de terra i per tant no tenim cap problema amb la resta de la placa. Les línies de control (veure Fig. 3.6) són Vg, CLK, ST, i les dades obtingudes EOS, Video i Vref.

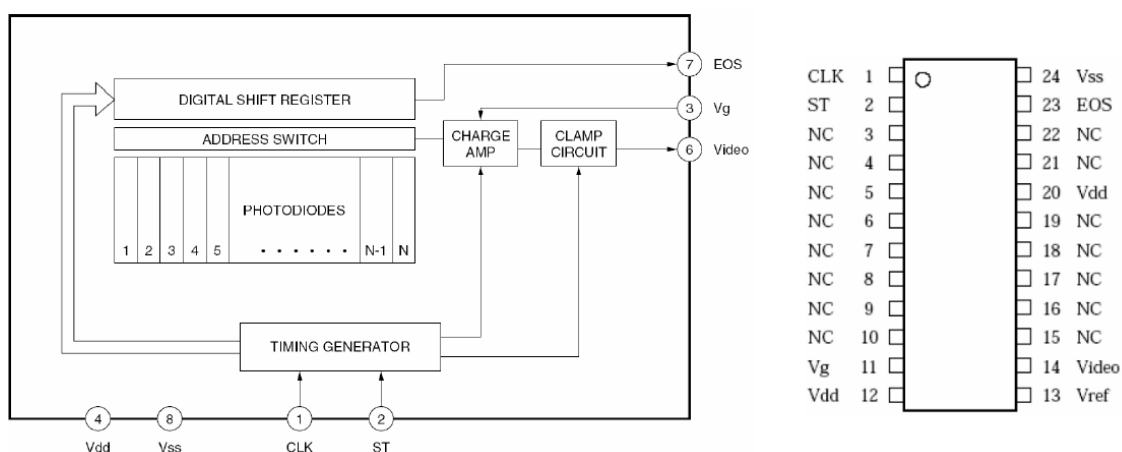


Fig. 3.5: Esquema intern del espectroradiòmetre i encapsulat.

- Vg (Gain Selection Voltage): Amb aquesta línia, es selecciona si la senyal de Video ha d'estar preamplificada (5V) o no (0V). Realment, la senyal de Video (que correspon la sortida dels díodes com veurem més endavant) sempre està preamplificada, però es selecciona si aquest guany ha de ser elevat o baix.
- CLK (Clock): És el senyal de rellotge que controla tot el procés d'adquisició de dades. Per tant, tota la resta de senyals (ST, EOS i Video) depenen d'ell. També condiciona el temps d'exposició dels fotodíodes, en funció de la durada de cada cicle de rellotge. El rang de freqüències possibles oscil·la entre 100Hz i 500kHz
- ST (Start Pulse): Senyal que indica l'inici d'una adquisició, coordinada amb un cicle de rellotge.
- EOS (End of Scan): Senyal que indiqui el fi de l'adquisició, o sigui, que ja s'ha generat la senyal de tots els díodes del sensor.
- Video: Senyal formada per les diferents sortides dels fotodíodes. A cada cicle de rellotge a partir de l'Start Pulse, el senyal agafa l'amplitud corresponent a un díode concret. La impedància de sortida és de 1kΩ.

- **Vref (Reference Voltage):** Aquest senyal és un nivell de tensió que ens indica quina sortida s'obtindria al canal de Video si es dongués el cas de fosc absoluta, o sigui, els fotodíodes rebessin excitació nula.

S'ha de concretar, però, que el temps durant el qual està exposat un fotodíode no correspon a la duració de un cycle de rellotge, si no al temps que es tarda entre dues adquisicions consecutives. En el nostre cas, correspon a 256 cycles, un per díode.

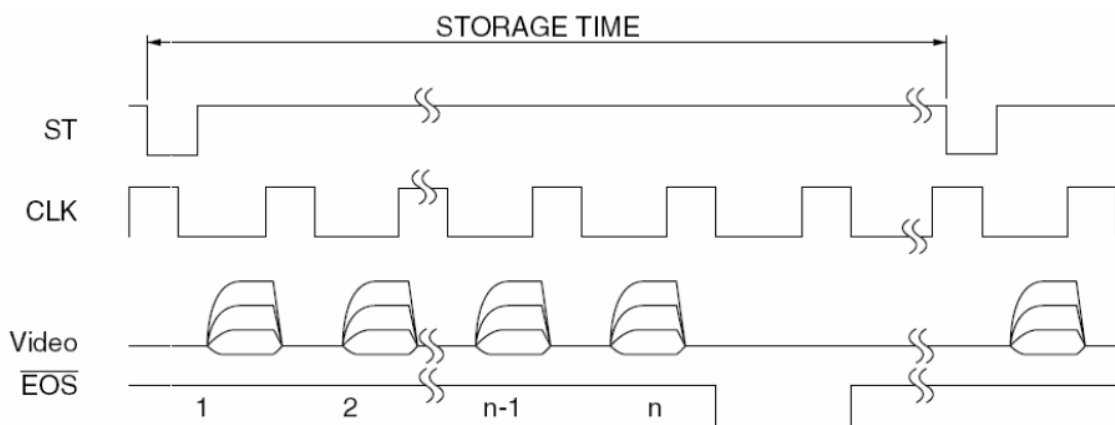


Fig. 3.6: Cronograma de les senyals de control de l'espectroradiòmetre. El temps entre pols consecutius de ST ens marca el temps d'integració. La sortida de cada fotodíode correspon a un cycle de rellotge.

3.2.2. Calibració

A partir del que s'ha comentat a l'apartat anterior, es pot prendre consciència que aquest l'espectroradiòmetre és més complex que altres tipus de sensors. Per aquest motiu és molt important una bona calibració del component, i un bon estudi previ de les seves reaccions davant diferents estímuls.

Afortunadament, amb les dades que ofereix el fabricant, i el testeig que se li va realitzar en el TFC de la fase II (apartat 1.3.2), no és necessari repetir tots aquests processos. Les dades rellevants que s'han de contemplar són:

- La sortida obtinguda, la senyal de Video, està compresa en un marge de 1V (nivell de fosc màxima) a 4,2V (saturació dels fotodíodes). Notar que, per tant, a la línia Vref també hi haurà 1V.
- Hi ha diferències de sensibilitat (resposta espectral) entre els fotodíodes (Fig. 3.7). O sigui, obtenim més excitació a unes longituds d'ona que a unes altres.
- Els fotodíodes es distribueixen l'espectre de forma aproximadament uniforme i lineal (Fig. 3.8).

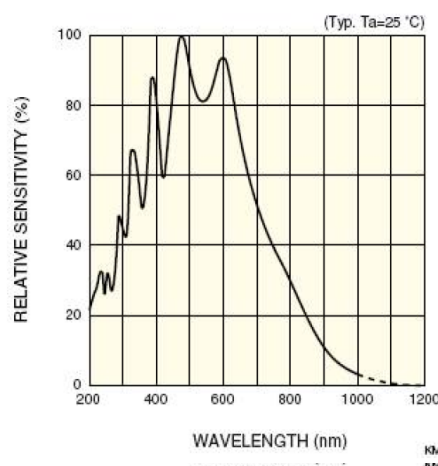


Fig. 3.7: Nivell de sensibilitat de l'espectroradiòmetre per cada longitud d'ona.

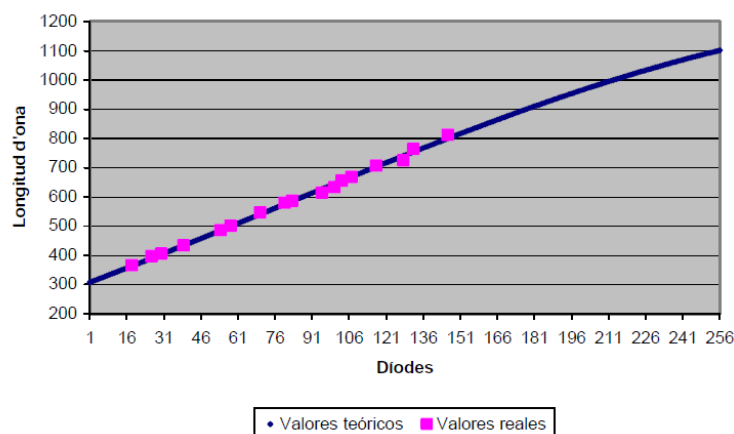


Fig. 3.8: Calibració del sensor de pressió. A l'eix d'abscisses hi ha el nombre de fotodíode, i en el d'ordenades les longituds d'ona. L'objectiu d'aquestes mesures és que es correspon cada díode amb una longitud d'ona concreta, de manera que es reparteixi l'espectre uniformement.

3.3.Acondicionament dels senyals

En els apartats 3.1 i 3.2 s'anuncia que les sortides dels sensors estan formades per senyals elèctrics que varien el seu voltatge. Aquests senyals s'han de digitalitzar per tal de poder-se processar posteriorment, amb la qual cosa seran necessaris conversors analògics-digitals.

Si es vol aconseguir la màxima eficiència en el procés de digitalització, s'ha d'aconseguir que el marge dinàmic de les senyals a quantificar sigui el mateix que el marge d'entrada del A/D. Si és superior, estarem perdent informació la partir del nivell màxim, i si és inferior la quantificarem amb menys resolució.

Els A/Ds utilitzats, els quals es detallen en el següent apartat, disposen d'un marge d'entrada de 0 a 5V. Els dos senyals provinents dels sensors estan compreses dins d'aquests marges sent inferiors a 5V en el seu màxim, i superiors a 0V en el seu valor mínim. Per tant, per ajustar-los és necessari realitzar dos operacions: ajustar el nivell d'*offset* i amplificar.

3.3.1.L'amplificador operacional Rail-to-Rail

S'han estudiat diferents maneres de porta a terme les operacions comentades, eliminació d'*offset* i amplificació, de la manera més compacta possible. Una possible solució es la d'utilitzar un amplificador d'instrumentació.

En cas d'optar per aquest tipus de component, la circuiteria per acondicionar els senyals es simplificava de sobremanera. Per realitzar les dues operacions a la vegada, només és necessari introduir el senyal amb la informació i el senyal de referència (el valor d'*offset*) per una entrada diferencial en un integrat, i regular el seu guany amb una sola resistència.

Malauradament, l'estructura interna d'aquest element impedeix la seva utilització en el projecte. En un procés d'acondicionament de senyal típic, sempre s'elimina primer l'*offset* i després s'amplifica, ja que si no el procés d'amplificat afecta a aquest *offset*, i s'obtenen valors de tensió elevats que que costen més de corregir. L'esquema intern és una cascada de tres amplificadors, que realitza aquestes operacions en l'ordre invers al comentat. Així, doncs, si a la sortida es desitja una tensió màxima de 5V, a l'interior de l'integrat poden aparèixer tensions superiors a aquest valors. S'arriba a la conclusió que és necessari doncs una alimentació del sistema superior a la sortida en el seu valor màxim, fet que trenca amb les especificacions de la circuiteria de la sonda (exceptuant el sensor de pressió).

Desestimada aquesta opció, s'ha utilitzat finalment un amplificador operacional *rail-to-rail*, concretament el AD8603 d'Analog Devices. Aquest tipus d'amplificadors es caracteritza perquè tant accepta tant a l'entrada com a la sortida valors equivalents a la seva alimentació, que és just el que es busca en el projecte. A més a més, destaca per produir poc soroll (CMOS) i poc consum.

3.3.2.Característiques tècniques

L'AD8603 és un amplificador simple (un amplificador per integrat) *rail-to-rail*, caracteritzat per una tensió d'*offset* i un nivell de soroll molt baixos, a més de una baixa corrent de bias d'entrada i un mínim consum.

La característica de ser rail-to-rail, tant a l'entrada com a la sortida, el fa idoni per aplicacions amb limitacions d'alimentació, on la font d'energia sigui senzilla i no es puguin obtenir alimentacions bipolars. L'encapsulat TSOT-23 comporta un volum mínim.

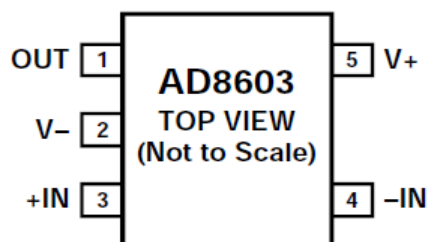


Fig. 3.9: Patillatge de l'amplificador AD8603.

Taula 3.2: Característiques elèctriques del AD8603 @ $V_s = 5V$.

Marge de tensions d'alimentació	1.8 – 5 V
Marge de tensions de sortida	4.97V – 16 mV
Marge de tensions d'entrada	-0.3 – +5.2 V
Corrent bias d'entrada	0.2 pA
Corrent offset d'entrada	0.1 pA
CMRR	100 dB
Tensió d'offset d'entrada	40 μ V
Impedància de sortida	36 Ω
Slew Rate @$R_L=10 K\Omega$	0.1V/ μ s
GBP @$R_L=10 K\Omega$	316 kHz
PSRR	100 dB

3.3.3.Muntatge realitzat

Si es pretén que l'amplificador amplifiqui la senyal i al mateix temps elimini l'*offset* de la senyal d'entrada, és necessària una combinació de resistències com la mostrada a la Fig. 3.10. A partir de l'anàlisi del circuit, s'arriba a les següents conclusions:

- A l'entrada V1 hi ha el senyal d'interès a tractar.
- A l'entrada V2 hi ha un senyal amb el valor d'*offset*, o sigui, el valor mínim possible a V1.
- A la sortida Vo obtenim la diferència de tensió entre V1 i V2, multiplicada pel guany G.
- El guany ve determinat per la relació $G=R_2/R_1$.
- Per tal d'eliminar l'*offset* del senyal V1, a partir de V2, s'han de complir les relacions de resistències $R_1=R_3$ i $R_2=R_4$.

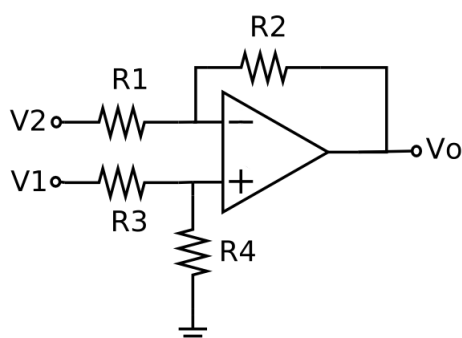


Fig. 3.10: Esquema de muntatge de l'amplificador.

Per tant, en el cas del sensor de pressió, es té:

$$G = \frac{V_{O_{max}} - V_{O_{min}}}{V1 - V2_{max}} = \frac{5 \text{ V}}{(489.14 - 89.54) \cdot 10^{-3} \text{ V}} = 12.51 \quad (3.1)$$

$$R_2 = R_4 = 1200 \Omega$$

$$R_1 = R_3 = R_2 \div G = 1200 \Omega \div 12.51 = 95.92 \Omega \approx 100 \Omega$$

I en l'espectroradiòmetre:

$$G = \frac{V_{O_{max}} - V_{O_{min}}}{V1 - V2_{max}} = \frac{5 \text{ V}}{(4.2 - 1) \text{ V}} \approx 1.56 \quad (3.2)$$

$$R_2 = R_4 = 2400 \Omega$$

$$R_1 = R_3 = R_x \div G = 2400 \Omega \div 1.56 = 1538.46 \Omega \approx 1500 \Omega$$

3.3.4. Filtre antialiasing

Una funció present en molts sistemes d'adquisició és el filtre antialiasing. L'objectiu és complir el teorema de Nyquist, de manera que no es mostregi una senyal amb components freqüencials massa elevades.

En el disseny exposat en aquest document, en el cas de l'espectroradiòmetre no és necessari degut al tipus de sortida que dona. En el cas del sensor de pressió i en els futurs sensors de conductivitat i temperatura si que és necessari. Al ser senyals a baixes freqüències, es pot utilitzar un filtre amb una freqüència de tall molt baixa, eliminant molt del soroll.

No s'ha implementat en la placa actual, però es té en compte en el pròxim model on ja hi haurà tots els sensors.

3.4. Digitalització de les dades

Un cop les dades provinents dels sensors abarquen tot el marge dinàmic disponible, de 0 a 5V, es pot procedir a la seva digitalització. Degut a que es desitja un nivell de precisió de 16 bits, no tenir limitacions de velocitat i a la familiaritat amb el component per haver-lo utilitzat anteriorment, s'escull l'AD977 d'Analog Devices per dur a terme aquesta tasca.

Tot i que existeix la possibilitat de multiplexar els senyals abans d'entrar a l'A/D o just després de ser processats, abans d'enviar-los cap al dsPIC, es desestima aquesta opció ja que el dsPIC disposa de canals suficients per comunicar-se amb dos A/D a la vegada. En cas d'afegir més canals de dades en el futur, la multiplexació sí que haurà de contemplar-se.

3.4.1. Característiques tècniques

L'AD977 és un convertidor analògic/digital de 16 bits, gran velocitat i baix consum que treballa amb una alimentació de 5V. Té una taxa de *throughput* de 100 kSPS (milers de mostres per segon) i s'entrega calibrat de fàbrica per minimitzar tots els errors de linealitats. Permet treballar amb marges bipolars de ± 10 V, ± 5 V i ± 3 V, així com marges unipolars de 0 a 10 V, 0 a 5 V (el cas del projecte) i 0 a 4 V.

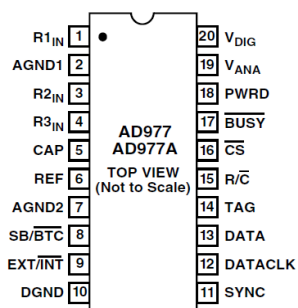


Fig. 3.11: Encapsulat del convertidor A/D.

3.4.2. Control i Comunicació SPI

Tot el procés d'inici de mostreig es fa mitjançant les línies d'entrada R/C i CS. Un cop finalitza el procés, el qual s'indica amb la línia de sortida BUSY, les dades en format digital són entregades a 100 kSPS mitjançant la comunicació SPI, *Serial Peripheral Interface*. Aquest tipus de comunicació es caracteritza per utilitzar només dues línies, una de rellotge i una de dades, i permet més de dos dispositius en un mateix bus. Segueix una estructura de mestre-esclau, on el mestre és l'encarregat de general el senyal de rellotge. Es pot veure de forma gràfica el procés general a la Fig. 3.12, i el detall de la comunicació SPI a la Fig. 3.13.

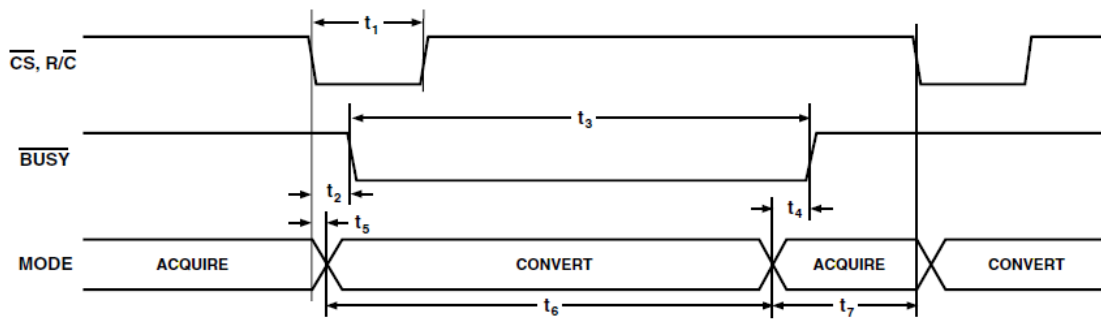


Fig. 3.12: Cronograma general de tot el procés d'adquisició d'una mostra de l'A/D.

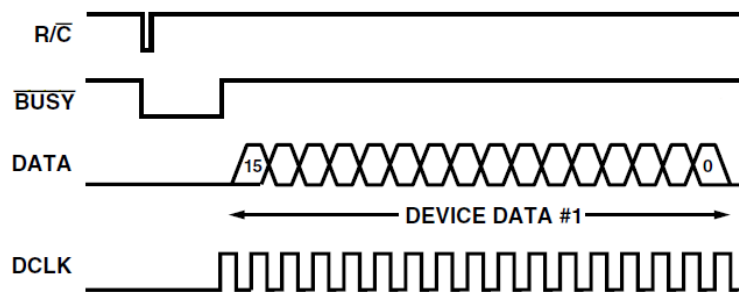


Fig. 3.13: Cronograma de la comunicació SPI de l'A/D. Els senyals R/C i DCLK son generats pel dsPIC, i BUSY i DATA per l'AD977.

3.4.3. Muntatge realitzat

Per una banda, es té les connexions referent a l'entrada de dades, o sigui la part analògica. Aquesta s'ha de configurar pel marge dinàmic desitjat, que és de 0 a 5 V (recordar que es pot configurar per diferents marges, apartat 3.4.1). Per això s'han de connectar les resistències i condensadors pertinents, tal com mostra la Fig. 3.14.

Per la part referent a la sortida digital i altres senyals de control, bàsicament s'han connectat totes al dsPIC, menys la senyal de RC que es manté a nivell baix permanentment. Això és degut a que la pota RC funciona amb conjunció amb CS, i només té sentit treballar amb les dues per separat si hi haguessin diferents A/Ds compartint el mateix bus (CS serviria per seleccionar quin AD volen utilitzar, i RC per iniciar l'adquisició).

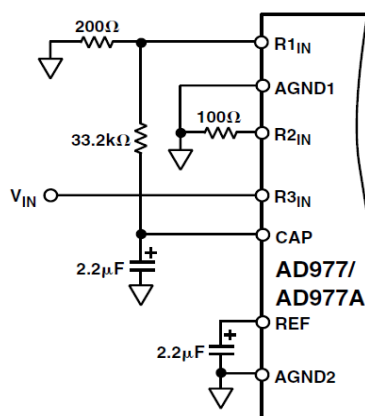


Fig. 3.14: Connexions i elements necessaris per la part analògica del convertidor. D'aquesta manera es defineix un marge a V_{in} de 0 a 5V.

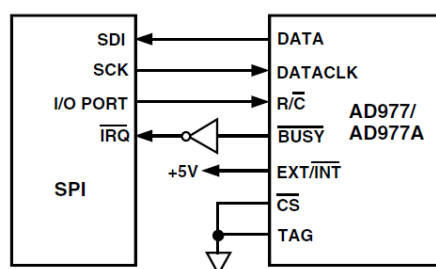


Fig. 3.15: Connexions necessàries entre l'AD977 i el dispositiu mestre (dsPIC).

3.5. Comunicació RS-485

Com s'ha comentat anteriorment, degut a la distància entre la sonda i la interfície de control, fins a 50 metres, es desestimava en aquest projecte l'ús del RS-232 que s'havia mantingut en treballs anteriors (apartat 1.3). Un altre estàndard de comunicacions molt semblant és el RS-485, que es pot veure com un RS-232 millorat, més robust a interferències i que permet una major longitud del cable (fins a 1200 metres), així que serà l'utilitzat en aquesta implementació.

Pel que respecta al dsPIC i al PC, no suposa grans problemes aquest canvi de tecnologia, ja que es segueix utilitzant una UART (*Universal Asynchronous Receiver-Transmitter*) en cada un dels extrems de la comunicació. A la placa, doncs, s'hi instal·larà un MAX485 de Maxim, i al PC/PDA una targeta PCMCIA.

3.5.1. Característiques tècniques

El MAX485 és un *transceiver* de baix consum per comunicacions *half-duplex* RS-485. L'*Slew Rate* del controlador no està limitat, permetent arribar a velocitats de fins a 2.5 Mbps. Opera amb una alimentació de 5 V.

Disposa de un pin pel qual entrar les dades a transmetre en nivells TTL (DI), i un altre pel qual llegir les rebudes (DO), també en TTL. Dos pins (A i B) són els encarregats de transmetre la informació en els nivells del estàndard RS485. Com que es treballa en *half-duplex*, realitzen les funcions tant de transmissió com de recepció. Amb dos pins més (DE i RE) es controla si està funcionant en mode receptor o transmissor.

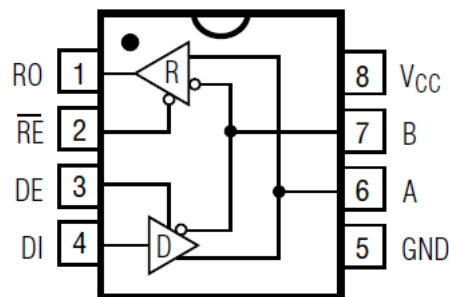


Fig. 3.16: Esquema del encapsulat del MAX485. Es pot observar com internament els pins DE i RE habiliten /deshabiliten la recepció/transmissió.

3.5.2. Muntatge realitzat

Els pins d'entrada i sortida de dades TTL (DI i DO), es connecten als perifèrics UART del dsPIC. El pins de control (DE i RE), degut a que un és complement de l'altre, es poden curtcircuitar, i connectar-los amb un pin E/S del dsPIC per tal de controlar el seu estat. Les línies de transmissió RS-485 es connecten al cable de sortida cap a l'exterior de la sonda.

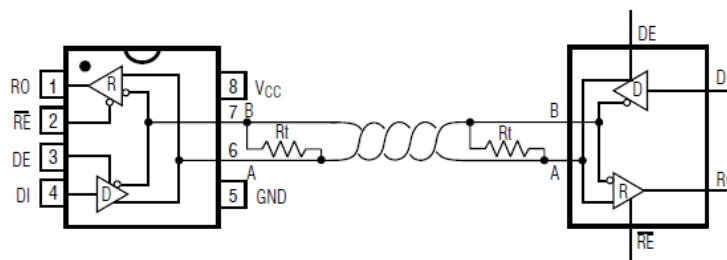


Fig. 3.17: Esquema de la comunicació RS-485. Es pot observar com es trenca el cable i s'afegeixen dos resistències com a terminadors per tal de millorar el rendiment del sistema.

Per la part del PC, la targeta PCMCIA adquirida (una PM-120 del fabricant BrainBoxes) presenta un connector DB9, que s'ha d'adaptar al connector de la sonda. Només s'ha de destacar que al treballar en *half-duplex* és necessari curtcircuitar els pins Rx+ amb Tx+ i Rx- amb Tx- de la targeta, tal com indica el fabricant en les especificacions.

Per últim, en les comunicacions RS-485 es recomana posar resistències amb valor de 120 Ω com a terminadors de les línies, tal com es pot observar a la Fig. 3.17.

3.6.El microcontrolador dsPIC

El dsPIC30F6012 de Microchip és el cervell de la sonda. Ha de gestionar el control dels sensors, com l'adquisició de dades i les comunicacions amb l'exterior. Afortunadament, disposa de prou potència de càlcul (inclòs una DSP) i perifèrics integrats per dur a terme totes les tasques simultàniament.

3.6.1.Característiques tècniques

Les capacitats d'aquest dispositiu són enormes, fins i tot en aquest projecte (el més ambiciós dels comentats a l'apartat 1.3) no se n'utilitza tot el potencial. Com ha estat analitzat anteriorment, es procedeix a només remarcar les funcionalitats realment rellevants pel nostre sistema.

3.6.1.Característiques generals

Per tal de poder funcionar, el dsPIC només necessita una alimentació de 5 V i un provinent d'un oscil·lador (RC, XTAL o altres). El model utilitzat es presenta en un encapsulat TQFP de 64 pins, acoblat a un sòcol PLCC de 68 pins, cosa que permet la fàcil inserció/extracció del component.

- Velocitat de procés de fins a 30 MIPS amb un rellotge de 40 MHz. En el projecte serà de 4 MIPS amb un rellotge de 16 Mhz.
- Instruccions de fins a 24 bits, amb camí de dades de 16 bits.
- Memòria de programa Flash de 144 kB, memòria de dades RAM de 8kB, i memòria auxiliar EEPROM de 4 kB.
- DSP amb velocitats d'execució d'instruccions d'un sol cicle per instrucció.

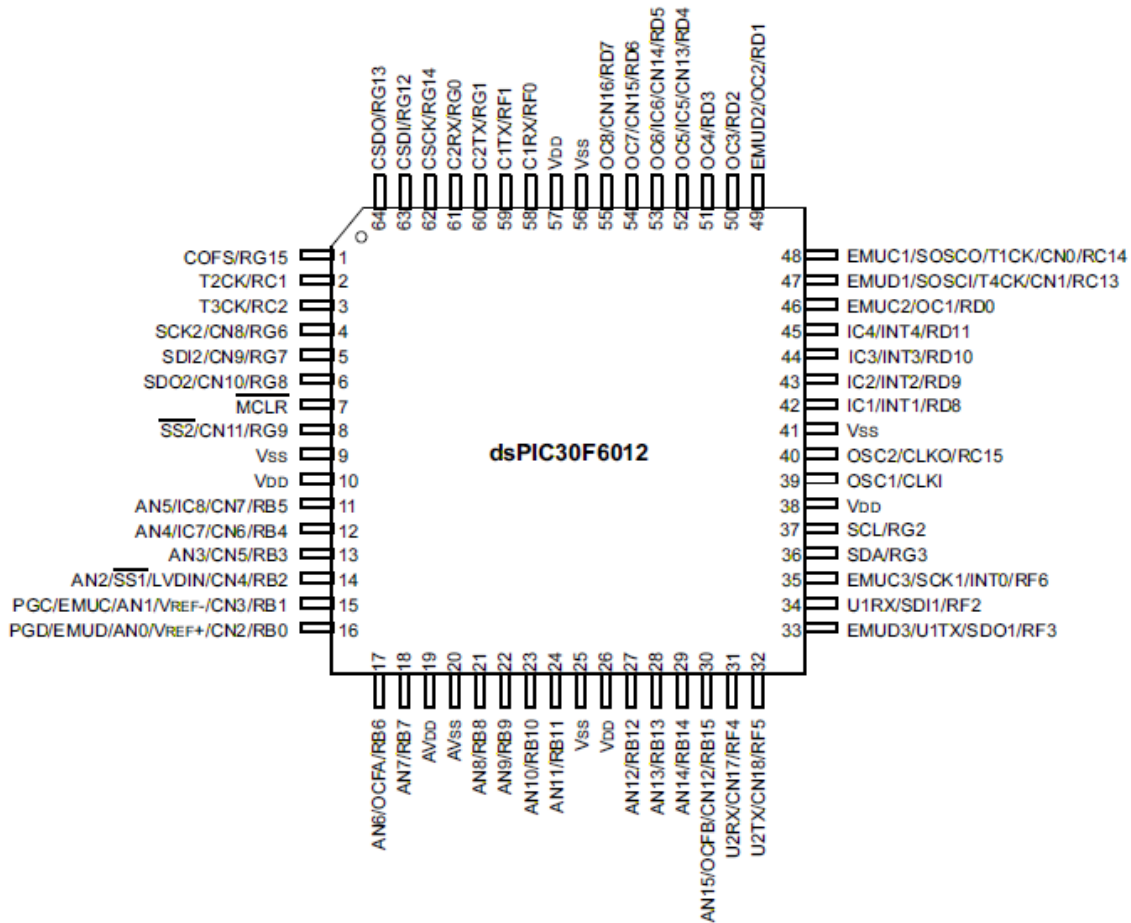


Fig. 3.18: Encapsulat del dsPIC. Es pot observar el gran nombre de ports E/S sortida de que disposa, i que molts d'ells tenen nomenclatures especials ja que donen accés directe a perifèrics de comunicació concrets.

3.6.2. Ports E/S

Moltes de les línies del microcontrolador poden ser utilitzades com a pins d'entrada o sortida de dades controlades per l'usuari. Són línies amb nivells de tensió TTL, i per tant es compatibles amb tots els components descrits anteriorment. Hi ha 5 ports diferents, de 16 pins cada un, però molts d'ells són compartits amb els perifèrics interns del dsPIC. D'aquesta manera, s'ha d'escollir amb precaució quins s'utilitzen, per no utilitzar el mateix recurs a dos finalitats diferents, resultant en incompatibilitats de funcionament.

S'utilitzaran per:

- 3 línies per comunicar-se amb l'espectroradiòmetre, generant les senyals necessàries de control.
- 2 línies per comunicar-se amb els convertors A/D, generant les senyals de control que no corresponen al protocol SPI.
- 1 línia per comunicar-se amb el MAX485, per controlar si està en recepció o en transmissió.

3.6.3.Perifèric SPI

El dsPIC incorpora ja dos mòduls que permeten la comunicació transparent amb el protocol SPI, sense haver-se de preocupar de generar la senyal de rellotge ni fer coincidir l'enviament de dades amb els flancs del rellotge. S'habilitaran els dos mòduls, per tal de rebre les dades dels dos conversors A/D. Com s'ha especificat a l'apartat 3.4.2, cada comunicació SPI només necessita dues línies.

3.6.4.Perifèric UART

De la mateixa manera que posseeix mòduls per SPI, també disposa de dos mòduls UART. Aquests mòduls permeten comunicacions amb protocols RS-232 o RS-485 amb facilitat. Tot i així, les senyals segueixen estan limitades als nivells TTL, i per tant és necessari un *transceiver* (com el MAX485, apartat 3.5) per la comunicació. Cal destacar que el un dels mòduls UART comparteix pins amb un dels mòduls SPI, de manera que en el projecte només disposen realment d'un dels mòduls. En aquest cas, utilitzem dues línies per la comunicació, una de transmissió i una altre de recepció.

3.5.Sistema general – Placa final

Per la realització del disseny de tota l'electrònica del sistema s'ha utilitzat el programa P-CAD 2004 de l'empresa Altium. Tot el procés ha estat condicionat pels requisits del projecte, les característiques dels components utilitzats, i les particularitats del tub de PVC on ha d'anar allotjada la placa.

El programa està format per diferents mòduls, de manera que s'ha pogut realitzar el procés íntegrament sense necessitar programari addicional.

3.5.1.Creació de components

El P-CAD incorpora de fàbrica unes llibreries on ja hi ha declarades les característiques de components d'alguns fabricants. En aquestes llibreries, s'hi pot trobar l'esquema del component, el *pattern* (que equival a la representació real i física del component, o sigui l'espai que ocuparà a la placa i la mida dels seus pins), o ambdues coses.

Malauradament, per molts dels components utilitzats en el projecte no existia cap llibreria, de manera que ha estat necessària la seva creació. D'altres s'han hagut de modificar, per tal de que, per exemple, fossin després més fàcils de soldar. Tots aquest processos s'han pogut fer amb eines pròpies del P-CAD, com el P-CAD 2004 Pattern Editor i el P-CAD 2004 Library Executive.

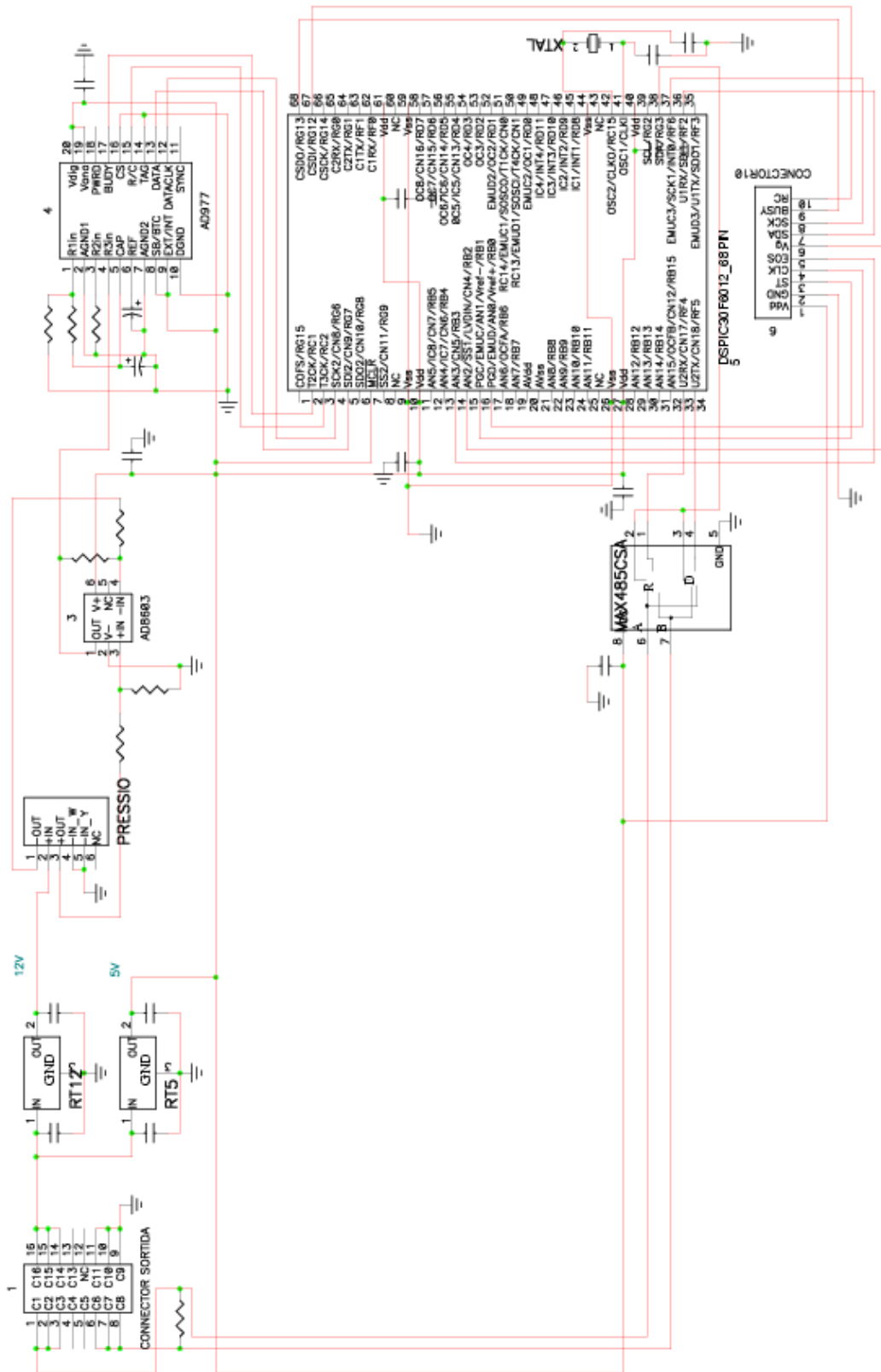


Fig. 3.19: Esquema de la placa principal del sistema. Inclou el dsPIC, la comunicació RS-485, tota l'etapa del sensor de pressió i l'alimentació.

3.5.2. Disseny del Esquemàtic

Un cop ja s'han definit els components que s'utilitzaran, es procedeix a la creació de l'esquemàtic amb la utilitat P-CAD 2004 Schematic. L'objectiu d'aquesta fase és realitzar les connexions del sistema, de manera que de forma abstracte es pugui comprovar la compatibilitat dels components entre ells.

Elimina la necessitat d'haver d'utilitzar llapis i paper per fer un esboç del sistema a més de les facilitats que ofereix, com per exemple, no oblidar de fer cap connexió. És un pas intermig abans de procedir al disseny del PCB. Un cop finalitzat, s'exporta en un fitxer que s'anomena *netlist*, necessari per la següent etapa. A les Fig. 3.19 i Fig. 3.20 es poden observar els esquemàtics de les plaques. De manera comparativa, a la Fig. 3.21 es mostra un diagrama de blocs dels components i les connexions.

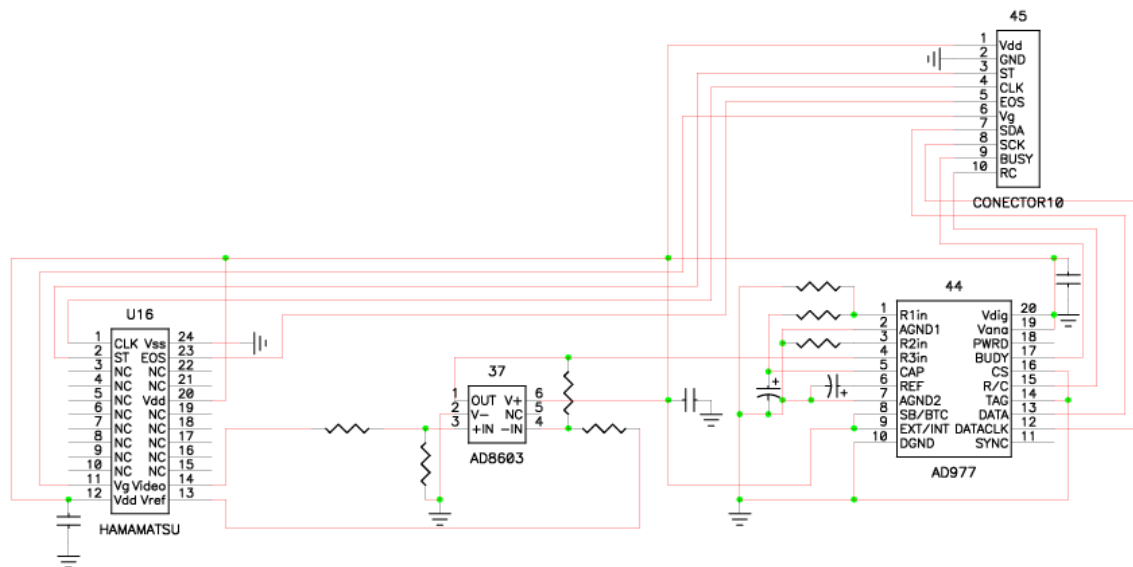


Fig. 3.20: Esquema de la placa secundària del sistema. Inclou l'espectroradiòmetre i tota la seva etapa d'adquisició.

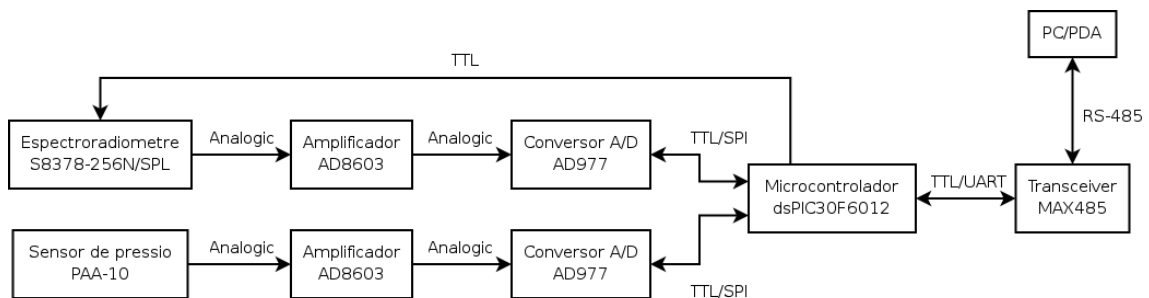


Fig. 3.21: Diagrama de tot el sistema. Es mostren els principals components i les comunicacions/interaccions entre ells.

3.5.3. Disseny del PCB

En aquest etapa es defineix tot el referent a la representació física i real a la placa. El resultat serà el que es passarà a la fresadora per tal de tallar i definir la placa, sense aplicar-hi cap correcció. Es du a terme amb l'aplicació P-CAD 2004 PCB.

Primer de tot, s'ha d'importar el fitxer *netlist* creat en l'etapa anterior. Això crearà tots els components que hi hagi definits i ens indicarà les connexions entre els diferents pins. Les marques que indiquen connexions són merament orientatives, per saber quina pota de cada component va amb quina. Per tant, les tasques que s'han de realitzar són:

- Definició de la mida de la placa.
- Col·locació dels components dins d'aquesta. Això inclou tant encarament com a quina cara estan situats, la superior (*top*) o la inferior (*bottom*).
- Definició de les línies entre *pads*. S'ha d'escollir per on passen exactament, la seva amplitud, per quina cara, i s'hi canvien de cara mitjançant una via.
- Definició dels plans de massa i la seva connexió amb les presses de massa dels components.

A les Fig. A.3 i Fig. A.4 es poden observar els PCB resultants del procés. Un cop finalitzat, és necessari exportar el fitxer en format Gerber, que es processarà amb el programa Circuit Camp i s'enviarà a la fresadora.

3.5.4. Configuració de la fresadora

Abans d'enviar els fitxers directament a la fresadora, és necessari definir la mida de les broques de que es disposa, en quin cas s'utilitzarà cada una, i quin recorregut han de realitzar exactament per la placa per tal d'extreure el coure corresponent.

CAPÍTOL 4. SOFTWARE

En aquest capítol es pretén donar a conèixer el funcionament de tot el programari introduït als diferents elements del sistema. Això inclou tant el microcontrolador dsPIC com al PC. En tots dos casos es farà una descripció del llenguatge utilitzat, les eines necessàries pel seu desenvolupament, i una explicació del codi escrit.

Respecte al codi, s'ha pensat que la millor manera per entendre'l no era mitjançant l'explicació de les diferents línies de codi, sinó mitjançant diagrames de blocs els quals facin referència a les diferents funcions que es puguin trobar. Així el lector es pot fer una idea general del funcionament del programa i la seva complexitat, sense necessitat de llegir cap línia de codi ni entendre com funciona el llenguatge de programació utilitzat. Per qualsevol informació més detallada, es pot consultar el codi íntegre als Annexs A.5 i A.6, amb els comentaris corresponents per facilitar-ne la comprensió.

4.1. Programari del dsPIC

Un dels canvis més importants en aquest projecte respecte els anteriors és l'increment de la independència de la sonda respecte l'usuari. Objectius com que la sonda decideixi quan ha de parar de prendre mostres a partir d'uns criteris predefinitos o ser capaç d'autoajustar el temps d'integració de l'espectroradiòmetre (apartat 3.2.1) dinàmicament comporten un augment de la "intel·ligència" del microcontrolador. Aquesta intel·ligència o capacitat de prendre decisions per si sol equival a la implementació de noves rutines de programació.

A més d'afegir noves funcionalitats, s'ha analitzat el codi antic que es podia reaprofitar, de manera que en alguns casos fins i tot s'ha modificat per millorar-ne el rendiment, les funcionalitats, o fer-lo més robust a errors com el cas del control de la comunicació amb la UART.

4.1.1. Organització del codi i dependències

En el cas del dsPIC, l'estructura i organització dels arxius és la típica de un programa en llenguatge C. En diferents fitxers es defineixen funcions que fan feines concretes. A més a més, per cada fitxer es crea un fitxer de capçalera (extensió .h), que s'utilitza quan es fan enllaços entre ells. En un fitxer es defineix la funció principal, de manera que de forma jeràrquica anirà utilitzant totes les funcions declarades en els altres fitxers de manera que actuïn com un tot.

A la Fig. 4.1 hi ha representada en forma d'arbre tots els fitxers desenvolupats i les dependències entre ells. Com es pot observar, el programa principal es troba en el fitxer main.c. S'ha estructurat el codi de manera que a cada llibreria hi hagi les funcions per treballar amb un perifèric o component concret.

D'aquesta manera, en cas de que es modifiqui el *hardware* en un futur, només serà necessari eliminar/afegir la llibreria corresponent a aquest *hardware*. han estat totes creades des de zero pel projecte, o a modificant les del projecte comentat a l'apartat 1.3.2. Un cas particular és la llibreria `io30f6012.h`, que és una llibreria que ja proporciona l'entorn de desenvolupament IAR Embedded Workbench, el qual permet accedir des del llenguatge C a registres i bits concrets del dsPIC.

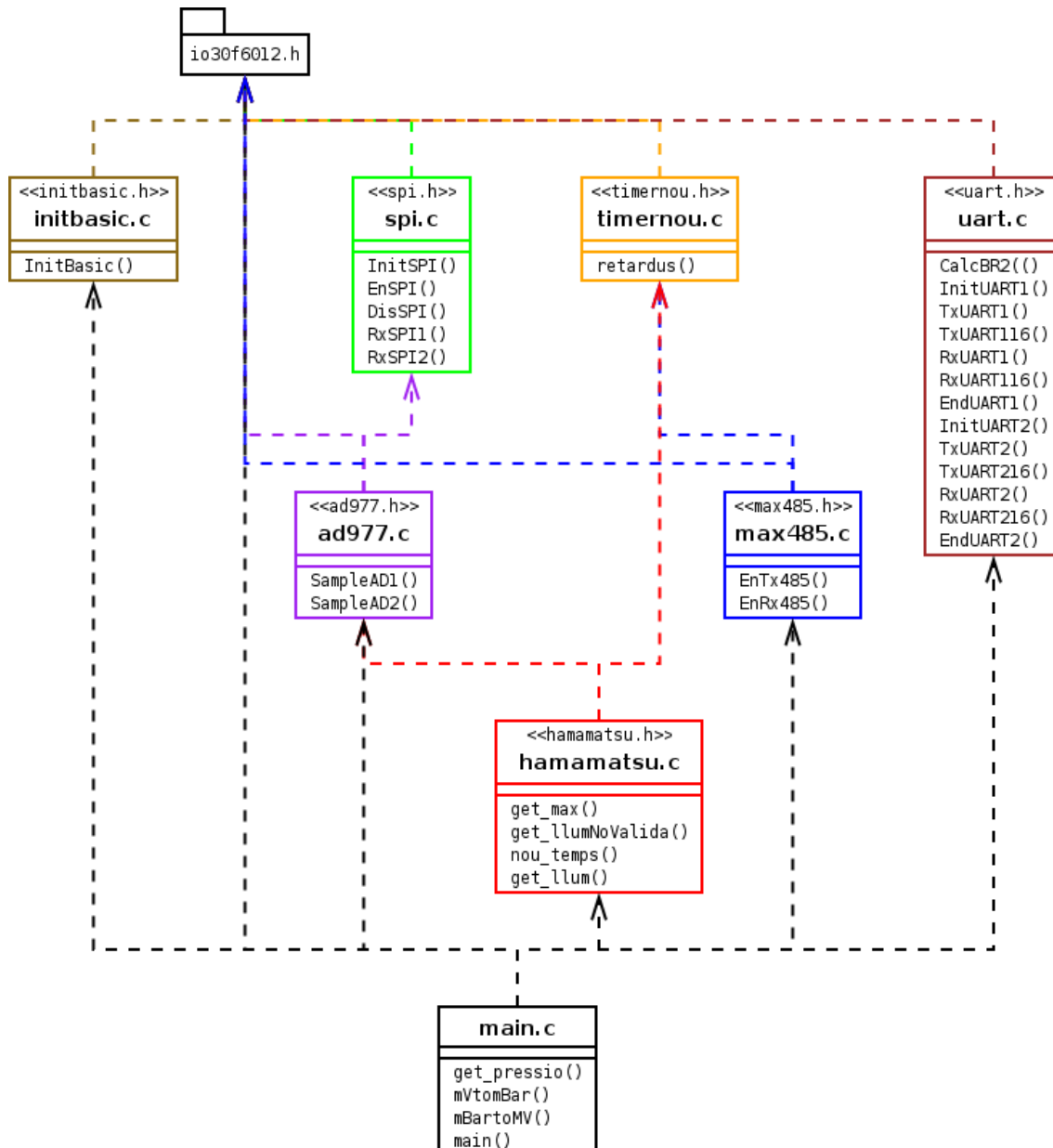


Fig. 4.1: Diagrama de dependències i funcions del programa del dsPIC.

4.1.2. Explicació i comentaris

A la Fig. 4.3 es troba un diagrama de flux que mostra el funcionament del programa. Un fet a destacar és que té inici (l'activació de la sonda), però no té fi, ja que roman sempre esperant ordres de control. No s'entra amb detall d'algunes accions, com per exemple tots els subprocessos que comporta prendre una mostra del sensor de pressió, ja que augmentaria de manera innecessària l'esquema general. I tot i així, aquestes subtasques són el suficientment senzilles com per entendre com funcionen a partir de l'estudi del codi, ja que han estat consegüentment documentades i comentades per facilitar la tasca de comprensió.

Un dels processos més complexes, és potser el d'adquisició de dades de l'espectroradiòmetre. No el procés d'adquisició en si, ja que es tracta de generar els senyals de control comentats a l'apartat XX i mostrejar en els instants concrets, si no en el càlcul automàtic del temps d'integració. Aquest es pot trobar a la llibreria hamamatsu.h, on hi ha declarades totes les funcions referents al control de l'espectroradiòmetre.

Bàsicament, és una adaptació de l'aproximació de l'algorisme PID que es va implementar en el projecte de l'apartat 1.3.2. En aquell cas, el càlcul el realitzava el Matlab, i enviava el nou temps d'integració al dsPIC, mentre que ara el calcula dinàmicament sense cap intervenció externa. La finalitat és ajustar el temps d'integració dels díodes, per tal que el senyal obtingut (la representació de l'espectre) ocupi tot el marge dinàmic disponible, però sense arribar a saturar la sortida.

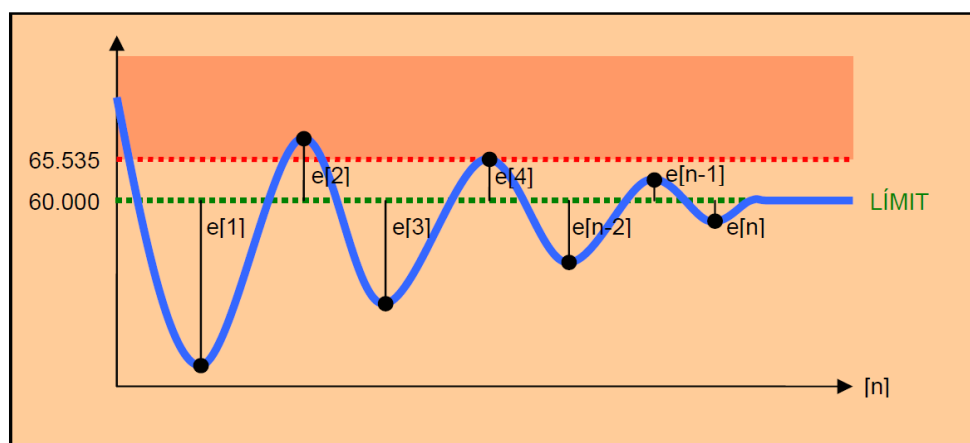


Fig. 4.2: Exemple de l'ajust automàtic mitjançant l'algorisme PID. A cada iteració disminueix l'error.

L'objectiu és que el nivell màxim del senyal sigui igual a un nivell prefixat com a òptim. En funció de la diferència entre el valor màxim real i el valor màxim òptim, es disminueix o s'augmenta el temps d'integració. A més a més, aquest nivell de correcció és directament proporcional a l'error, tal com es pot veure a la Fig. 4.2. Informació més detallada sobre el seu fonament matemàtic es pot trobar a l'annex A.2.

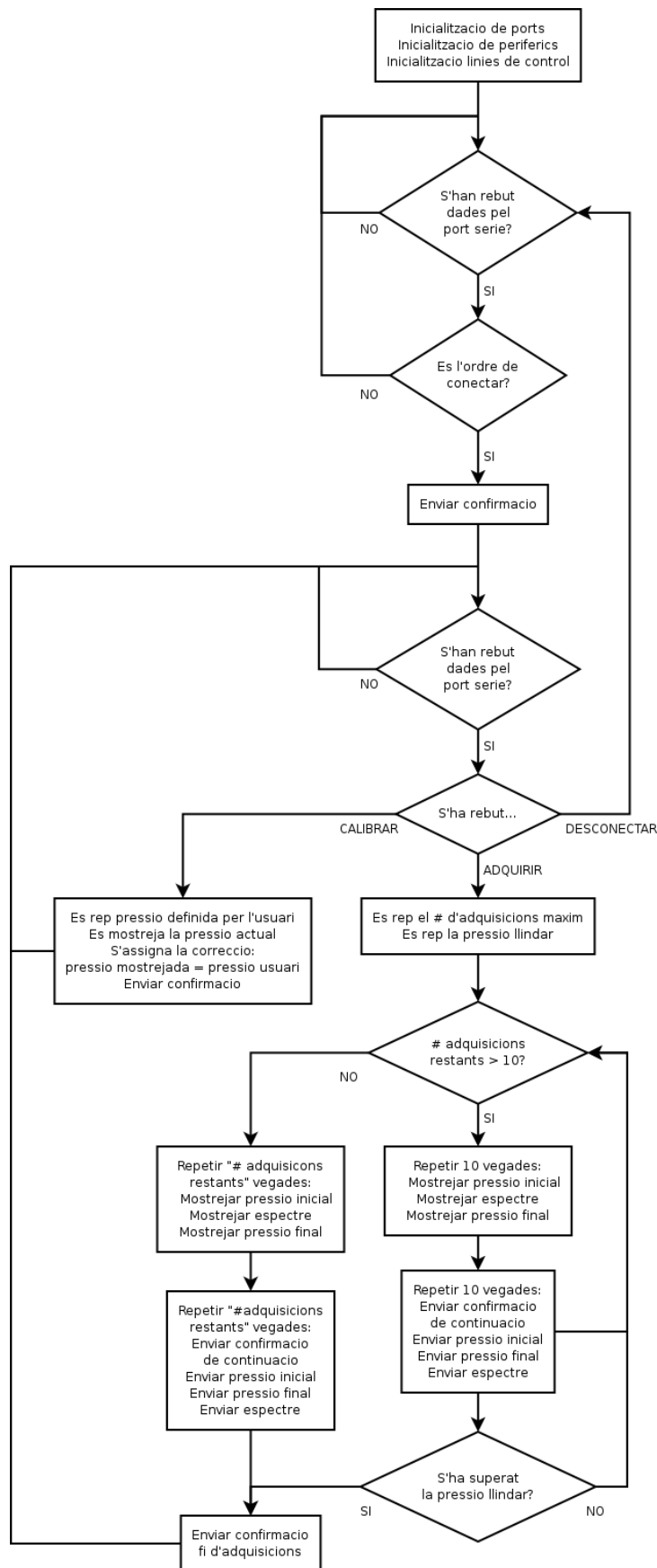


Fig. 4.3: Diagrama de flux del funcionament del dsPIC.

4.2.Programari del PC

Un del objectius recomanats però no obligatori era la de canviar completament tot el programari del PC utilitzat fins el moment, o sigui, Matlab. Els motius d'aquesta decisió són bàsicament dos:

- Matlab és un programa propietari basat en llicències. Això implica que no només es necessària una còpia del programa, sinó haver pagat per una llicència en cada equip on es vulgui utilitzar. A més a més, no es té disponible el codi font de tot el programa, de manera que no es pot optimitzar o modificar per tal d'ajustar-se millor als objectius de l'usuari. Per últim, el fabricant (Mathworks) pot introduir canvis en futures versions del programa que no siguin del tot compatibles amb versions anteriors.
- En un programa que ha de ser utilitzat en proves de camp prima la senzillesa i rapidesa d'utilització. És molt més útil una PDA on tenim un programa que ens mostra l'estat de la sonda mitjançant colors i ens permet controlar-la amb quatre botons, que no un programa en un portàtil a pantalla completa en el qual ens hem de desplaçar per menús i ens mostra totes les dades en temps real. Al cap i a la fi, el important en aquests moments és captar i emmagatzemar les dades, ja que el processat i visualitzat d'aquestes es farà posteriorment al laboratori.

Per aquestes raons, i també per les creences pròpies de que el model de programari lliure és idoni per entorns d'educació i investigació, s'ha decidit implementar tot el programari amb el llenguatge Python.

Python és habitualment comparat a TCL, Perl, Scheme, Java i Ruby. Actualment, Python es desenvolupa com un projecte de codi obert, administrat per la Python Software Foundation. L'última versió estable del llenguatge és actualment (gener de 2006) la 2.4.

Python permet dividir el programa en mòduls reutilitzables des d'altres programes Python. Incorpora una gran col·lecció de mòduls estàndard que poden utilitzar com a base dels programes. També hi ha mòduls que proporcionen capacitats d'E/S de fitxers, crides al sistema, *sockets* i fins i tot interfícies GUI com Tk. Python és un llenguatge interpretat, fet que estalvia un temps considerable en el desenvolupament del programa, doncs no és necessari compilar ni enllaçar. L'interpret es pot utilitzar en mode interactiu, característica que facilita experimentar amb característiques del llenguatge, escriure programes d'un sol ús o provar funcions durant el desenvolupament del programa. El principal objectiu que persegueix aquest llenguatge és la facilitat, tant de lectura com de disseny.

4.2.1.Organització del codi i dependències

A diferència del codi del dsPIC, aquest està tot comprés en un únic fitxer. S'utilitzen també llibreries, però no són de desenvolupament propi, si no que ja

venen incloses amb el propi intèrpret o són de lliure distribució. Un altre diferència important és que Python permet programació orientada a objectes, amb característiques com definicions de classes. A la Fig. 4.4 hi ha una representació de les dependències, tan de les llibreries com de la classe definida en el programa principal, amb tots els objectes i funcions que la componen.

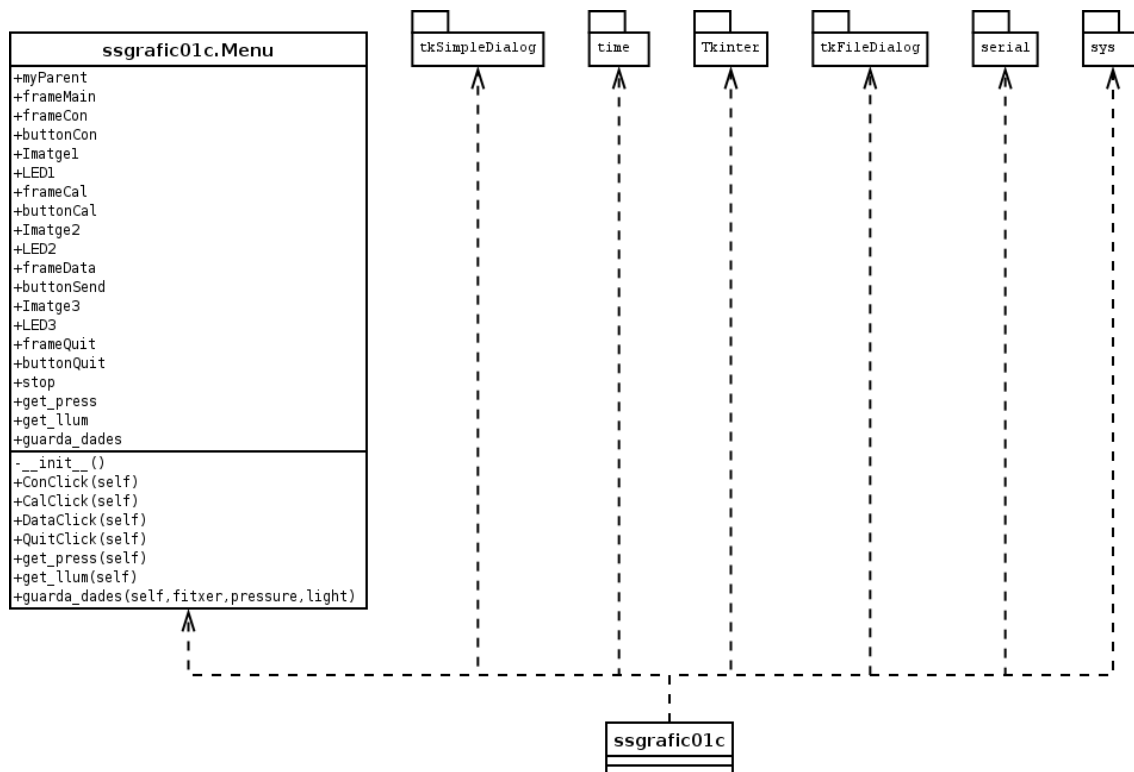


Fig. 4.4: Diagrama de dependències i classes del programa del PC

4.2.2. Explicació i comentaris

Per tal de comprendre el funcionament del programa, s'ha plasmat en el diagrama de flux de la Fig. 4.6. Al tractar-se de un programa a una interfície gràfica, el seu comportament es basa en events (la interacció del usuari) per prendre algunes decisions. Totes les accions que no estan contemplades, com per exemple pulsar el botó de calibrar abans del de connectar, és perquè s'ha dissenyat el programa de manera que no permeti realitzar accions “no possibles”, inhabilitant/habilitant parts de l'interfície dinàmicament. No es comenta en profunditat tot el referent al control d'errors en la comunicació, que es basa en comprovacions mitjançant bytes coneguts, ja que complicaria el diagrama de forma innecessària.

A la Fig. 4.5 hi ha una captura del programa, on es pot observar com s'ha optat per la minimització i simplicitat d'ús. Tot es realitza mitjançant botons, es dona a entendre quines accions té disponibles l'usuari mitjançant l'habilitació/inhabilitació d'aquests, i s'informa de l'estat de les accions

mitjançant uns LEDs virtuals,. A més a més, aquests LEDs serveixen per informar d'errors en la comunicació.

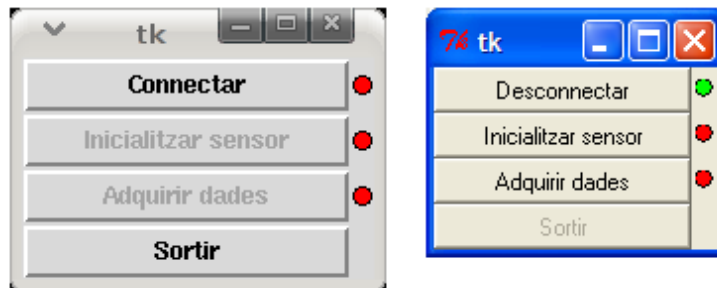


Fig. 4.5: Captures de pantalla del programa de control de la sonda en diferents sistemes operatius i en diferents estats d'operació.

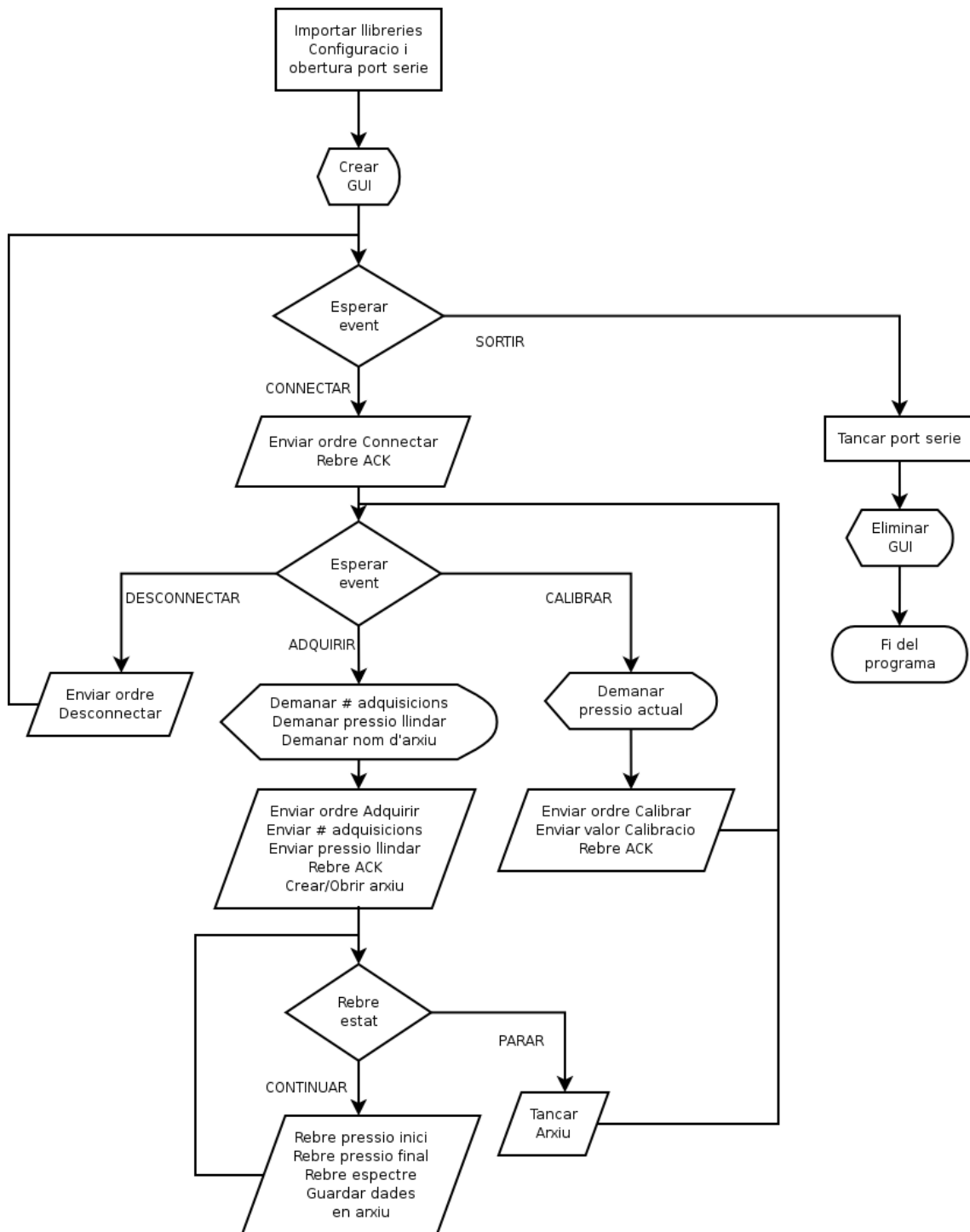


Fig. 4.6: Diagrama de flux del funcionament del programa del PC.

CAPÍTOL 5. ASPECTES MEDIAMBIENTALS

A continuació es fa un anàlisi de diferents aspectes mediambientals i la seva relació amb el projecte. Cal recordar que l'objectiu del projecte ja és de per si purament mediambiental, ja que és una eina que servirà per realitzar estudis científics del litoral Mediterrani.

5.1. Concepció del projecte

- Alguns dels components utilitzats han estat reciclats d'altres projectes, de manera que s'ha minimitzat l'adquisició de nous components. Tenir aquestes consideracions en el disseny permet reduir costos, tant econòmics com mediambientals.
- L'ús i malbaratament de l'energia també afecta negativament el nostre entorn, de manera que un altre tret característic del disseny és l'elecció de components de baix consum.
- El disseny en dues plaques separades afegeix una modularitat al sistema, de manera que es pot variar la funcionalitat sense haver-lo de substituir tot sencer.
- Fins ara, la utilització de Matlab en el PC comportava que aquest hagués de disposar d'una bona potència de càlcul. Utilitzar un nou programari més lleuger per controlar la sonda permet utilitzar-la amb ordinadors més antics, augmentant-ne les possibilitats de reciclatge.

5.2. Construcció del projecte

- S'ha intentat provar en sistemes reutilitzables (com protoboards) les diferents parts del sistema, en comptes de fer diferents versions de plaques PCB que després s'hagin de desestimar.
- Degut a la minimització de l'espai, alguns components estan soldats directament a la placa i la seva reutilització és difícil. Tot i això, els components més costosos i complexos com el dsPIC o l'espectroradiòmetre estan allotjats en sòcols, de manera que és possible reutilitzar-los en futurs sistemes.

5.3. Explotació

- Tot i utilitzar la sonda en un medi aquàtic, les repercussions són mínimes ja que tots els sensors incorporats són passius.
- No s'emet cap tipus de pol·lució ni radiació al utilitzar-lo, excepte l'alimentació. Aquesta es genera mitjançant bateries, amb totes les repercussions corresponents del seu us. Una optimització seria la d'utilitzar bateries reciclades.

5.4.Desmantellament

- Tot i ser una placa dissenyada per un fi molt concret, es pot utilitzar en el futur per testejar aspectes com comunicacions RS-485, o l'adquisició de dades d'un A/D. En certa manera, es podria reciclar com una placa Demo del dsPIC.
- L'estructura de PVC de la sonda es pot reutilitzar, ja que la unió amb la placa és mitjançant cargols i connectors, no hi ha cap soldadura ni fixació permanent.

CAPÍTOL 6. CONCLUSIONS

Per finalitzar, es fa un comentari sobre l'estat actual del projecte. S'analitzen quins objectius s'han complert, si ha estat de forma satisfactòria, i quines consideracions s'han de tenir en compte de cara a un futur. A més a més, es plantegen unes reflexions sobre els sistemes d'adquisicions de dades intel·ligents, i un possible canvi de rumb que hauria de prendre el desenvolupament del projecte.

6.1.Estat actual

Si partim dels objectius especificats en el capítol 2, es pot considerar que el projecte s'ha finalitzat de forma satisfactòria. S'ha desenvolupat dues plaques que es poden inserir en la sonda, capaces de comunicar-se amb un dispositiu extern com un PC, i enviar-hi les dades precedents dels sensors exigits. També s'ha completat el desenvolupament del programari per controlar la sonda, i funciona en diferents sistemes operatius.

L'únic aspecte no completat de forma esperada és el referent a la programació de l'algorisme que calcula automàticament el temps d'integració. Per causes que no ha donat temps a determinar, en ocasions l'algorisme no arriba a convergir, resultant en temps d'integració inadequats. Per tal de solventar aquest problema, és necessari depurar el codi en temps real durant l'execució, de manera que es puguin observar totes les variables implicades i poder discernir quins càlculs no es realitzen correctament.

Malauradament, no hi ha hagut prou temps com per portar a terme la part més mecànica de tot el procés. Restava adaptar el sistema de fixació de la sonda a les plaques actuals, col·locar el difusor de llum a l'estructura i adaptar el cable de la sonda a les bateries i el connector DB9. Són tots processos molt mecànics, com perforar i cargolar les plaques, de baixa dificultat, però que necessiten un temps addicional.

Però, això no ha impedit que es comprovés dins del possible el correcte funcionament del sistema. S'ha pogut simular la sortida del sensor de pressió amb els valors obtinguts a la calibració de l'apartat 3.1.3 i l'espectroradiòmetre a l'aire lliure tal com es va fer en els projectes de l'apartat 1.3, i el funcionament ha estat l'esperat.

Per tant, es pot concloure que l'únic obstacle que resta per tenir la sonda operativa són unes hores empalmant cables amb connectors i acoblant les plaques al interior de la sonda.

6.2.Desenvolupament futur

A arrel de la feina realitzada, s'han millorat els coneixements tan a nivell pràctic com teòric de les tecnologies tractades en el projecte. Per aquesta raó es volen

fer dues apreciacions sobre el futur de la sonda, referents a l'espectroradiòmetre i l'arquitectura de funcionament.

6.2.1. Alternatives a l'espectroradiòmetre

Respecte l'espectroradiòmetre, és qüestionable la utilització d'un sensor CMOS en el projecte actual. Primer de tot, s'ha de recordar que l'objectiu és prendre mostres a nivell de centímetres. Segons els estudis fets, amb el sensor escollit i el baix nivell de lluminositat que hi ha en el medi aquàtic, és necessari un temps de 2-3 segons per obtenir una lectura adient. Aquests dos factors comporten que la velocitat de desplaçament de la sonda hagi de ser molt reduïda. A més a més, caracteritzar una zona extensa comportaria molt de temps, i això comporta que les condicions del medi al principi del procés siguin diferents respecte les del final. Propietats del medi com les mesurades, poden haver canviat de forma significativa en un temps de dos minuts.

Una alternativa és la substitució de l'espectroradiòmetre per un sensor CCD. Aquests han disminuït enormement el seu preu els últims anys, de manera que tot i que durant el projecte comentat a l'apartat 1.3.1 es van desestimar, avui en dia ja es poden tenir en compte. A més a més, alguns d'aquests espectroradiòmetres basats en CCD entreguen les dades ja digitalitzades, simplificant el procés d'adquisició de dades.

6.2.2. Sistemes de xarxes de sensors

Consultar documentació i bibliografia referent al l'adquisició remota de dades ha servit per descobrir un tipus de tecnologies en auge. S'està fent referència a les xarxes de sensors i actuadors.

S'està tendint cap a un tipus de sistemes formats per diferents components, autònoms però amb capacitat de connectar-se amb d'altres i treballar conjuntament de forma transparent. I això tant en xarxes de sensors formades per sondes o boies com en sistemes com ROVs i AUVs.

Per exemple, es realitza un mòdul format per un sensor concret i un microcontrolador. Un altre mòdul, amb un pistó i un microcontrolador. Programats de la manera corresponent, es pot aconseguir que tots els mòduls es comuniquin per exemple via bus CAN i amb el mateix protocol. Llavors, es poden connectar n mòduls a un mateix bus, controlat per un microcontrolador que dirigeixi tot el sistema. S'obté així una estructura jeràrquica, on cada microcontrolador fa una feina específica, de forma més simple i eficient.

Fins i tot, un sistema de varis mòduls, es podria connectar a un altre sistema semblant, utilitzant un nou mòdul com a *gateway* entre ells. El resultat és obtenir una xarxa intel·ligent, que fins i tot seria capaç d'augmentar o disminuir la seva mida (i per tant les seves prestacions) de manera dinàmica.

Aquest és un objectiu molt més ambiciós que el de la sonda dissenyada, però es pot tenir en compte en dissenys futurs, sobretot si aparegués la possibilitat d'utilitzar varies sondes simultàniament, o el numero de sensors en cada s'incrementés.

BIBLIOGRAFIA

- [1] Microchip Design Resources [en línia]. Pàgina web, URL
<http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=70>
- [2] Ferg, S., Thinking in Tkinter [en línia]. Pàgina web, URL
<http://www.ferg.org/thinking_in_tkinter/index.html>
- [3] Pilgrim, M., *Dive into Python*, 2004 [PDF].
- [4] P-CAD 2004 (2004) P-CAD 2004. PCB Design. P-CAD_2004_PCB_Users_Guide [PDF]
- [5] Microchip (2004). dsPIC30F Family Reference Manual. [PDF]
- [6] Manuel Merchán i Sergi Pons (2004). Director Gabriel Montoso i Jaume Piera. Disseny d'un sistema de mesures espectroradiomètriques. Treball final de carrera, Departament de Teoria de Senyals i de comunicacions, Universitat Politècnica de Catalunya.
- [7] Maribel Pérez i Nuria Pujol (2005). Director Jaume Piera. Disseny d'un espectroradiòmetre. Treball final de carrera, Departament de Teoria de Senyals i de comunicacions, Universitat Politècnica de Catalunya.
- [8] Donovan Parks (2002). Embedded Communication System for an Autonomous Underwater Vehicle. Work Team Report, Faculty of Engineering, University of Victoria.
- [9] J. Piera, R. Quesada, E. Torrecilla, I. Fernandez i S. Pons (2005). SAMPLER: An instrumentation project for studying the effect of turbulence in aquatic systems. Unitat de Tecnologia Marina (CMIMA-CSIC) i Departament de Teoria de Senyals i de comunicacions (UPC).
- [10] E. Torrecilla, J. Piera, R. Quesada, I. Fernandez i S. Pons (2005). Characterization of the water optical properties using hyperspectral sensors. Unitat de Tecnologia Marina (CMIMA-CSIC) i Departament de Teoria de Senyals i de comunicacions (UPC).

ANNEX

A.1.El pont de Wheatstone

A l'apartat 3.1.1 es comenta que el sensor de pressió internament funciona com un pont de Wheatstone. A continuació es fa un comentari sobre aquest tipus de circuits.

Un pont de Wheatstone és un instrument elèctric de mesura inventat per Samuel Hunter Christie en 1833, millorat i popularitzat per Xerris Wheatstone en 1843. S'utilitza per mesurar resistències desconegudes mitjançant l'equilibri dels braços del pont (Fig. A.1). Aquests estan constituïts per quatre resistències que formen un circuit tancat, sent una d'elles la resistència sota mesura.

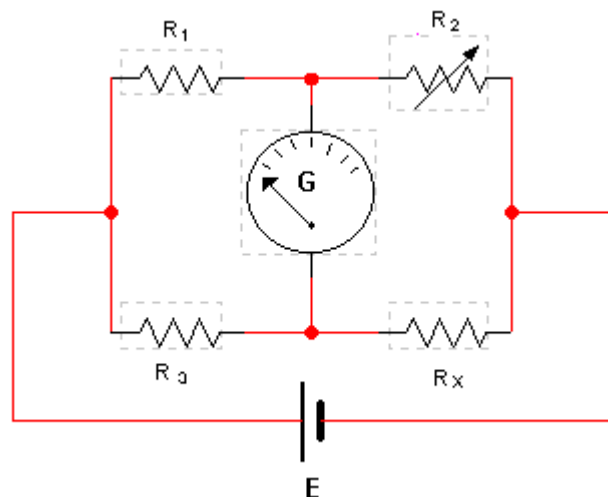


Fig. A.1: Pont de Wheatstone per mesurar una resistència.

A la Fig. A.1 veiem que, R_x és la resistència el valor de la qual volem determinar, R_1 , R_2 i R_3 són resistències de valors coneguts. a més la resistència R_2 és ajustable. Si la relació de les dues resistències del braç conegut (R_2/R_1) és igual a la relació de les dues del braç desconegut (R_x/R_3), el voltatge entre els dos punts mitjos serà nul i per tant no circularà corrent alguna entre aquests dos punts.

A més a més, el pont de Wheatstone s'utilitza en el camp dels sensors, com en el cas del sensor de pressió de l'apartat 3.1, amb resistències fixes i una que varia en funció del paràmetre a mesurar.

Un voltatge d'entrada continu, voltatge d'excitació, s'aplica entre el cim i el fons del diamant i el voltatge de sortida és mesura al centre. Quan la resistència que reacciona amb el paràmetre a mesurar el pont prèviament equilibrat ara es desequilibra. Això provoca una diferència de potencial al centre del pont,

proporcional al canvi en la resistència, per tant proporcional al canvi en la característica a mesurar.

A.2. Algorisme PID

En l'apartat 4.1.2 es comenta que s'ha programat una versió per al dsPIC de l'adaptació de l'algorisme PID desenvolupat en el projecte de l'apartat XX. Com a explicació matemàtica d'aquest, a continuació es fa una transcripció de la seva documentació:

L'algorisme utilitzat es basa en l'algorisme PID adaptat a les característiques del projecte. L'algorisme PID per a temps continu queda descrit per la fórmula A.1 on els paràmetres K_p , T_i i T_d caracteritzen l'evolució dels valors de sortida $u(t)$ en funció de l'error obtingut en el moment t i tenint en compte també l'error acumulat. Variant els paràmetres es pot variar l'estabilitat del sistema, i per cada cas s'hauran d'ajustar pels valors més adequats, segons la rapidesa en que la informació varia i la sensibilitat del sistema.

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (\text{A.1})$$

En el projecte, aquesta fórmula perd el significat, ja que es treballa en temps discrets (en mostres). Així que una possible traducció de la fórmula anterior aplicada al sistema seria la A.2.

$$u[n] = K_p(e[n]) + K_i \left(\sum_1^n e[n] \right) + K_d (e[n] - e[n-1]) \quad (\text{A.2})$$

Es pretén anar regulant el temps d'integració per aconseguir un temps òptim que permeti aprofitar al màxim el marge dinàmic. Així doncs, el valor de l'error ($e[n]$) serà la diferència entre el màxim absolut obtingut de l'adquisició de l'espectroradiòmetre i el límit marcat en el programa coma òptim, i el seu valor ha d'anar tendint a 0. L'evolució teòrica de l'amplitud relativa en funció del temps d'integració desitjat es pot veure a la gràfica 4.2.

S'ha de tenir en compte que l'algorisme PID pur és realment fàcil d'implementar quan es té caracteritzat el sistema al llarg del temps i la seva aplicació passa per implementar un filtre a l'entrada dels sistema. Això no es dona en el cas de la sonda, ja que l'únic que es pot caracteritzar és la proporció entre el temps d'integració i el valor màxim de la mostra equivalent al terme proporcional. Per tal d'aconseguir un millor funcionament s'ha introduït un nou terme corresponent al temps anterior i la resta de termes s'han anat regulant a mesura que s'han anat fent proves i realment es calcula mitjançant PID la

diferència que s'ha de sumar o restar al temps anterior. Cal remarcar que es tracta d'una primera aproximació de coeficients i és millorable.

Un exemple real de l'adaptació de les dades es troba a la figura A.X. Es tracta de la representació gràfica de tres adquisicions contigües, on s'observa la diferència de nivells.

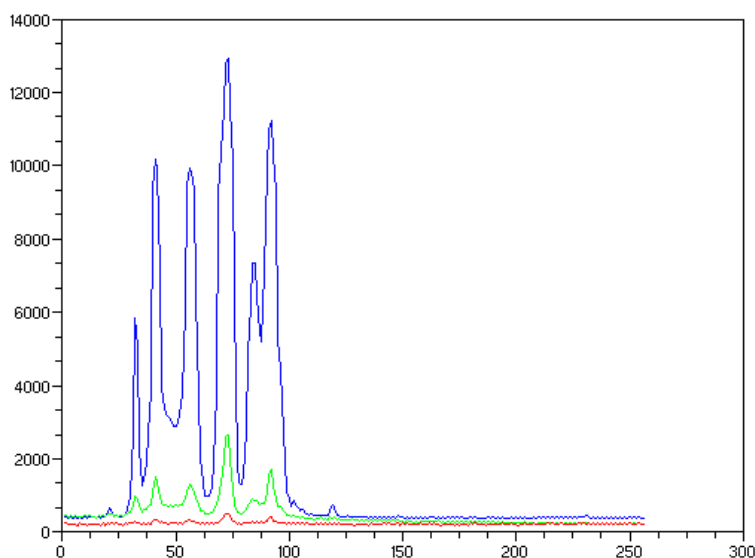


Fig. A.2: Representació gràfica de tres adquisicions contigües de l'espectroradiòmetre.

A.3. Pressupost i material

En aquest apartat es fa esment dels components que formen la sonda. L'objectiu és obtenir un preu total orientatiu, a partir del que hi ha exposat a la Taula A.3. No s'ha tingut en compte el tub de PVC, ni elements comuns de baix cost com resistències, condensadors, connectors, tirapins, etc.

Taula A.3: Pressupost aproximat de la sonda.

Component	Descripció	Unitats	Preu total (€)
Microchip dsPIC30f6012	Microcontrolador	1	16
Hamamtsu S8378-256N	Espectroradiòmetre CMOS	1	420
<i>HOBi Labs fiber-optic plane irradiance collector</i>	Col·lector de llum	1	250
Keller PAA-10	Sensor de pressió	1	140
Analog Devices AD8603	Amplificador Operacional rail-to-rail	2	2
Analog Devices AD977AN	Conversor A/D 16 bits	2	40
Maxim MAX485	Transceiver RS-485	1	1
STMicroelectronics L7805CV	Regulador de tensió 5V	1	1
STMicroelectronics L7812CV	Regulador de tensió 12V	1	1
CMAC HC49	Relotge de cristall 16 MHz	1	1
Total			872

A.4.Placa PCB

A les Fig. A.3 i Fig. A.4 es poden observar les plaques PCB, i a la Fig. A.5 una fotografia.

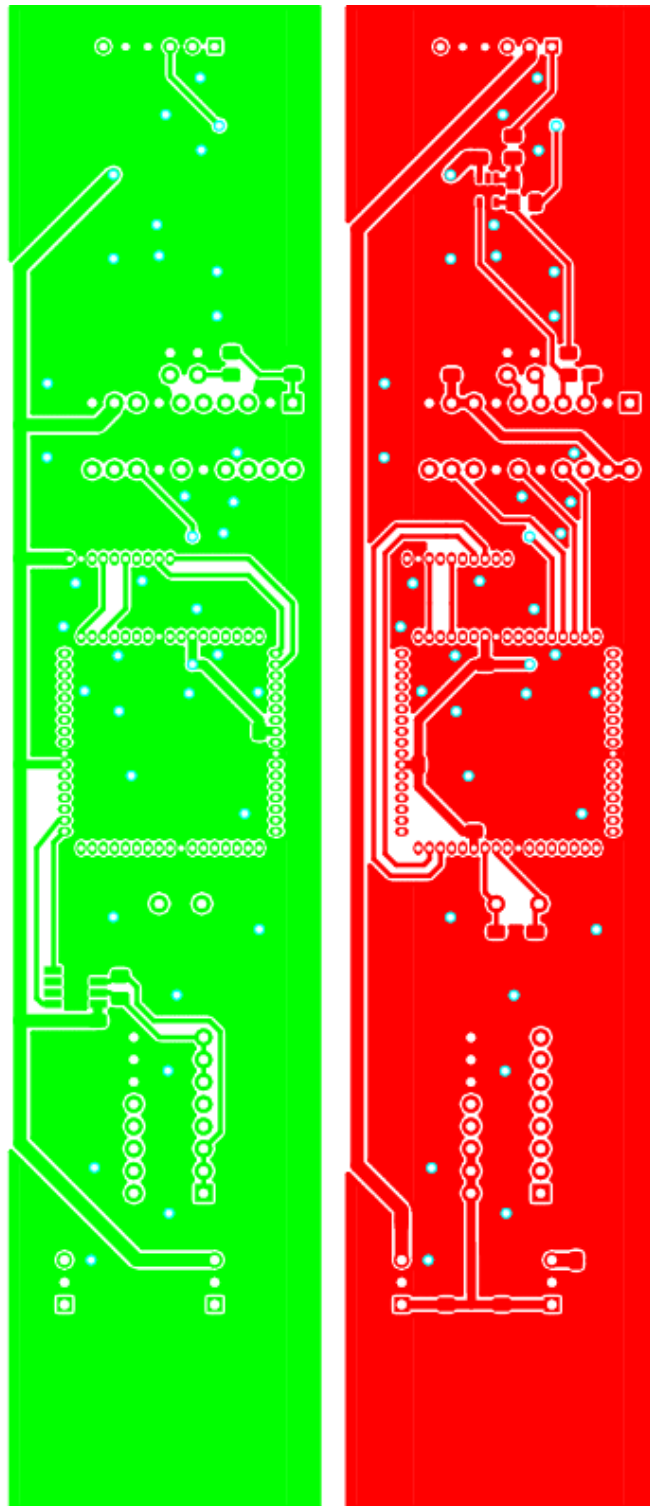


Fig. A.3: En verd la cara bottom i en vermell la cara top de la placa principal.

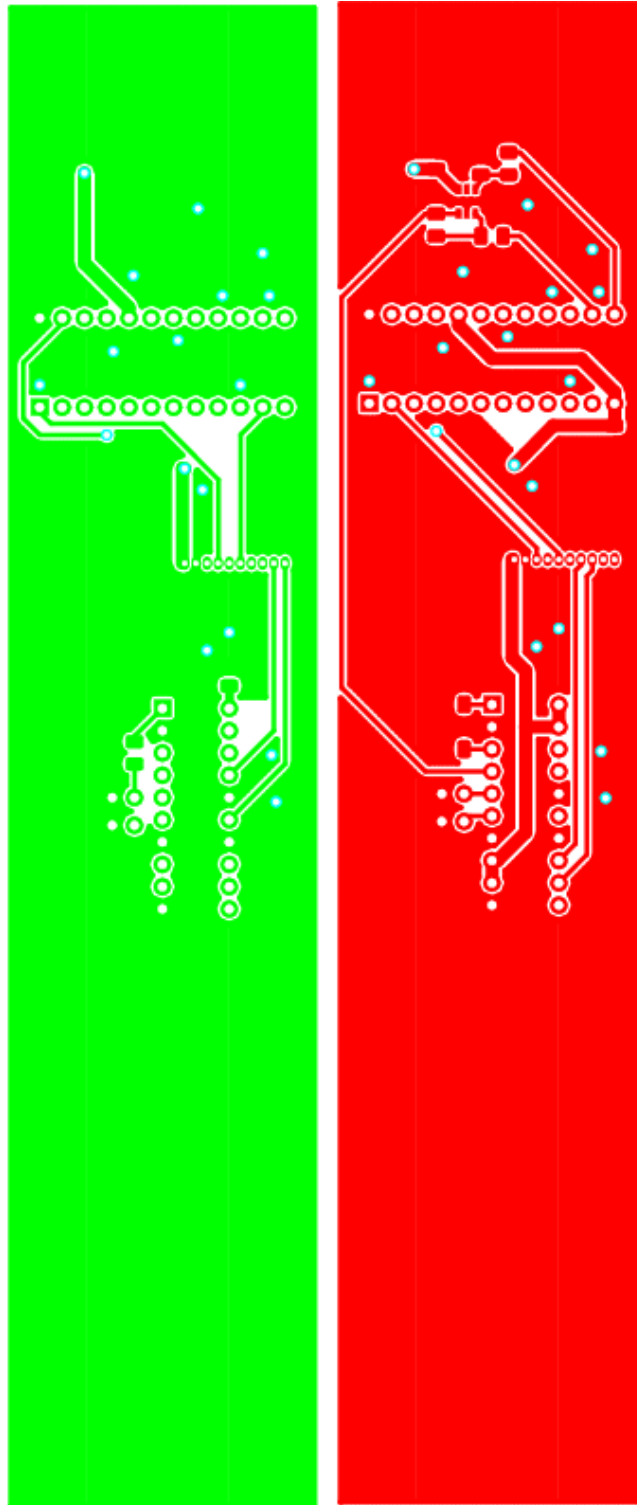


Fig. A.4: En verd la cara bottom i en vermell la cara top de la placa secundaria.

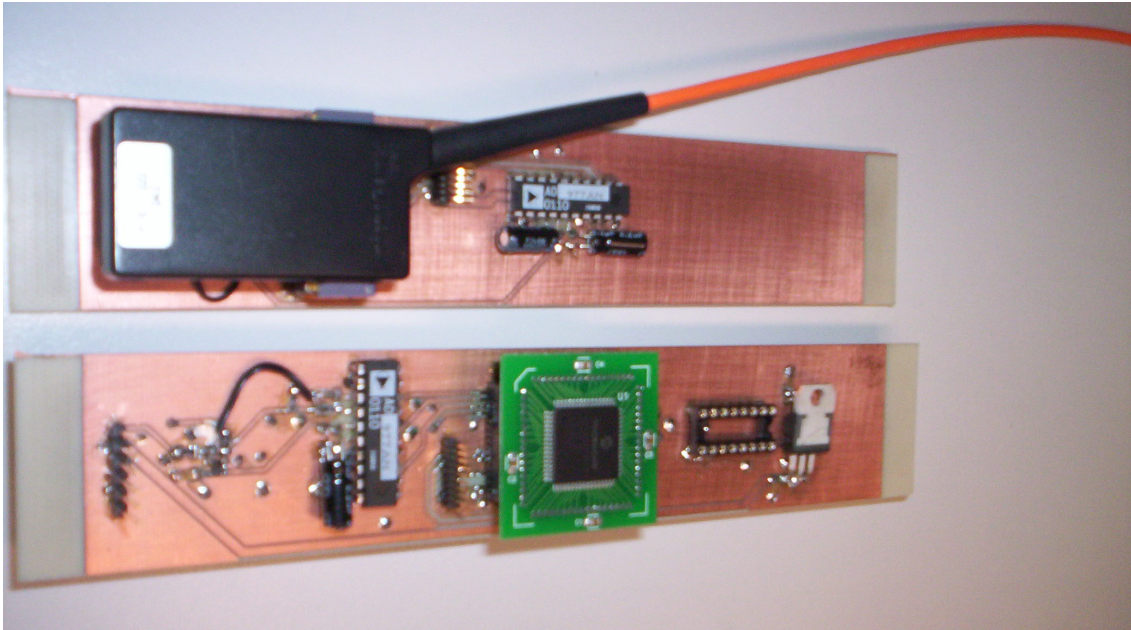


Fig. A.5: Fotografia de la cara top de les dues plaques. Es poden observar el dsPIC i l'espectroradiòmetre.

A.5.Codi PC

A continuació es mostra el codi del programa en Python desenvolupat.

A.5.1.ssgrafic01c.py

```
import serial
import time
import sys
if sys.platform=='Pocket PC':
    sys.path.append('\\\\Program Files\\\\Python\\\\Lib\\\\python23.zip\\\\lib-tk')
from Tkinter import *
import tkFileDialog
import tkSimpleDialog

### Paraules de control ###
connectp='c'
desconnectp='d'
getdatap='g'
calibrarp='i'
num_mostres=1

### Configuracio del port serie ###
if sys.platform=='win32':
    ser = serial.Serial(13, 9600)
else:
    ser = serial.Serial('/dev/ttyUSB0', 4800)
ser.flushInput() #Buidar buffer d'entrada
ser.flushOutput() #Buidar buffer de sortida

### Main class ###
class Menu:
    def __init__(self, parent):
        buttonWidth=20 #Variable que controla lamplada dels butons
```

```

self.myParent=parent

###Frame principal que ho engloba tot
self.frameMain=Frame(parent)
self.frameMain.pack()

###Frame de connexio
self.frameCon=Frame(self.frameMain)
self.frameCon.pack()

###Boto conectar
self.buttonCon=Button(self.frameCon)
self.buttonCon.configure(text="Connectar", command=self.ConClick,
width=buttonWidth)
self.buttonCon.pack(side=LEFT)

self.Imatge1=Canvas(self.frameCon, height=10, width=10)
self.LED1=Canvas.create_oval(self.Imatge1, 1,1,10,10, fill="red")
self.Imatge1.pack(side=RIGHT)

###Frame de calibrar
self.frameCal=Frame(self.frameMain)
self.frameCal.pack()

###Boto calibrar sensor
self.buttonCal=Button(self.frameCal)
self.buttonCal.configure(text="Inicialitzar sensor", state=DISABLED,
command=self.CalClick , width=buttonWidth)
self.buttonCal.pack(side=LEFT)

self.Imatge2=Canvas(self.frameCal, height=10, width=10)
self.LED2=Canvas.create_oval(self.Imatge2, 1,1,10,10, fill="red")
self.Imatge2.pack(side=RIGHT)

###Frame de dades
self.frameData=Frame(self.frameMain)
self.frameData.pack()

###Boto rebre dades
self.buttonSend=Button(self.frameData)
self.buttonSend.configure(text="Adquirir dades", state=DISABLED,
command=self.DataClick , width=buttonWidth)
self.buttonSend.pack(side=LEFT)

self.Imatge3=Canvas(self.frameData, height=10, width=10)
self.LED3=Canvas.create_oval(self.Imatge3, 1,1,10,10, fill="red")
self.Imatge3.pack(side=RIGHT)

###Frame de Sortir
self.frameQuit=Frame(self.frameMain)
self.frameQuit.pack(side=LEFT)
self.buttonQuit=Button(self.frameQuit)
self.buttonQuit.configure(text="Sortir", command=self.QuitClick,
width=buttonWidth)
self.buttonQuit.pack()

self.stop=0

### Event de pulsar el botó de conectar    ###
def ConClick(self):
    if self.Imatge1.itemcget(self.LED1, "fill")=="red":
        ser.write(connectp)
        received=ser.read(1)
        if received==connectp:
            self.Imatge1.itemconfigure(self.LED1, fill="green")
            self.buttonCon.configure(text="Desconnectar")
            self.buttonQuit.configure(state=DISABLED)
            self.buttonCal.configure(state=NORMAL)
            self.buttonSend.configure(state=NORMAL)
        else:
            self.Imatge1.itemconfigure(self.LED1, fill="yellow")
            self.buttonCon.configure(text="Error")
    else:
        ser.write(desconnectp)

```

```

self.Imatge1.itemconfigure(self.LED1, fill="red")
self.buttonCon.configure(text="Connectar")
self.buttonQuit.configure(state=NORMAL)
self.buttonCal.configure(state=DISABLED)
self.buttonSend.configure(state=DISABLED)

### Event de pulsar el botó de calibració ###
def CalClick(self):
    if self.Imatge2.itemcget(self.LED2, "fill")=="red":
        self.Imatge2.itemconfigure(self.LED2, fill="green")
        pres_actual=tkSimpleDialog.askinteger('Calibracio sensor',
'Pressio actual\nde la sonda (dBar):', initialvalue=0, minvalue=0, maxvalue=50)
        ser.write(calibrarp)
        ser.write(chr(pres_actual))
        received=ser.read(1)
        if received==calibrarp:
            self.Imatge2.itemconfigure(self.LED2, fill="red")
        else:
            self.Imatge2.itemconfigure(self.LED2, fill="yellow")
            self.buttonCal.configure(text="Error")
    else:
        self.Imatge1.itemconfigure(self.LED1, fill="red")
        self.Imatge2.itemconfigure(self.LED2, fill="red")
        self.Imatge2.itemconfigure(self.LED3, fill="red")
        self.buttonCal.configure(text="Inicialitzar sensor")
        self.buttonQuit.configure(state=NORMAL)
        self.buttonCal.configure(state=DISABLED)
        self.buttonSend.configure(state=DISABLED)

### Event de pulsar el botó d'adquisició ###
def DataClick(self):
    if self.Imatge3.itemcget(self.LED3, "fill")=="red":
        num_mostres=tkSimpleDialog.askinteger('Mostreig', 'Numero de
mostres\nmaxim a adquirir:', initialvalue=1, minvalue=1)
        press_min=tkSimpleDialog.askinteger('Pressio', 'Nivell de
pressio\nminim (dBar):', initialvalue=0, minvalue=0, maxvalue=50)
        nomfitxer=tkFileDialog.asksaveasfilename(defaulttextension='*.txt',
title="Save data as...", initialfile='dades.txt')
        self.Imatge3.itemconfigure(self.LED3, fill="green")
        self.buttonSend.configure(text="Adquirint dades...")
        self.buttonCon.configure(state=DISABLED)
        self.buttonCal.configure(state=DISABLED)
        ser.write(getdatap)
        ser.write(chr(num_mostres))
        ser.write(chr(press_min))
        received=ser.read(1)
        datafile=open(nomfitxer,'a')
        press_actual_end=-1
        while received==getdatap:
            press_actual_start=self.get_press()
            press_actual_end=self.get_press()
            llum_actual=self.get_llum()
            received=ser.read(1)
            self.guarda_dades(datafile, press_actual_start,
press_actual_end, llum_actual)
            num_mostres=num_mostres-1
        datafile.close()
        if (press_actual_end!=press_min & num_mostres!=0):
            print num_mostres
            self.Imatge3.itemconfigure(self.LED3, fill="yellow")
            self.buttonSend.configure(text="Error")
        else:
            self.Imatge3.itemconfigure(self.LED3, fill="red")
            self.buttonSend.configure(text="Adquirir dades")
            self.buttonCon.configure(state=NORMAL)
            self.buttonCal.configure(state=NORMAL)
    else:
        self.Imatge3.itemconfigure(self.LED3, fill="red")
        self.buttonSend.configure(text="Adquirir dades")
        self.buttonCon.configure(state=NORMAL)
        self.buttonCal.configure(state=NORMAL)

### Tancar el programa###
def QuitClick(self):

```

```

        ser.close()
        self.myParent.destroy()

###   Rebre les dades del sensor de pressio   ###
def get_press(self):
    byte1=ord(ser.read(1))
    byte2=ord(ser.read(1))
    datapress=byte1+byte2*256
    return datapress

###   Rebre les dades del espectroradiometre   ###
def get_llum(self):
    byte1=ord(ser.read(1))
    byte2=ord(ser.read(1))
    dosbytes=byte1+byte2*256
    llista=[dosbytes]
    for x in llista:
        if len(llista)<256:
            byte1=ord(ser.read(1))
            byte2=ord(ser.read(1))
            dosbytes=byte1+byte2*256
            llista.append(dosbytes)

    return llista

###   Guardar les dades en un fitxer de text   ###
def guarda_dades(self, fitxer, pressure1, pressure2, light):
    fitxer.write(str(pressure1))
    fitxer.write('\t')
    fitxer.write(str(pressure2))
    fitxer.write('\t')
    fitxer.write(str(light))
    fitxer.write('\n')
    return

### Main Program   ####
root=Tk()
programa=Menu(root)
root.mainloop()

```

A.6.Codi dsPIC

A continuació es mostra el codi del programa de la sonda, de tots els fitxers desenvolupats.

A.6.1.main.c

```

#include <io30f6012.h> //Necessaria per dsPIC30f6012
#include "init_basic.h" //Inicialitació basica del dsPIC
#include "max485.h" //Necessària pel MAX485
#include "uart.h" //Necessaria pel perifèric UART
#include "spi.h" //Necessaria pel perifèric SPI
#include "ad977.h" //Necessària pel AD977
#include "hamamatsu.h" //Necessària pel Hamamatsu

/*   Paraules de control de la comunicació sèrie   */
#define connectarp 'c'
#define calibrarp 'i'
#define mostrejarp 'g'
#define desconectarp 'd'

#define PT 10 //Numero de mostres maxm per paquet

/*   Adquisició d'una mostra de pressió   */
unsigned int get_pressio(void) {
    unsigned int pDig;
    pDig=SampleAD2();
    return pDig;
}

```

```

}

/* Conversió de pressió en mV a mBar */
unsigned int mVtomBar(float mV) {
    /*
     * 73.28mV = 1Bar
     * 73.28mv = 1000mBar
     * 1q = 76.29uV
     * mBar=#q*1.041145053
     */

    float mbar;
    mbar=mV*1.041145053;
    return mbar;
}

/* Conversió de pressió en mBar a mV */
unsigned int mBartomV (float mbar) {
    float mV;
    mV=mbar/1.041145053;
    return mV;
}

/* Funció Principal */
void main(void) {
    unsigned int control=0;
    unsigned short int desconectar;
    unsigned int pmBarReal;
    unsigned int pmVReal;
    unsigned int pmVActualS;
    unsigned int pmVMinim;
    signed int pmVInit=0;
    unsigned int llum[256];
    unsigned int num_mostres;
    unsigned int tIntegracio=1953; //1953us*2*256=1s
    //int tIntegracioFals=100; //100us*2*256=50ms
    signed int integral=0;
    signed int error=0;
    unsigned short int press_exit;
    unsigned int pSArray[10];
    unsigned int pEArray[10];
    unsigned int llArray[10][256];
    int i, j;

    InitBasic(); //Inicialitzem aspectes generals de la placa
    InitUART2(); //Inicialització del perifèric UART2
    EnRx485(); //MAX485 en mode rebre
    InitSPI(); //Inicialització dels perifèrics SPI
    EnSPI(); //Habilitació dels ports SPI

    while(1) {
        while(control!=conectarp) //Esperem a rebre ordre de connectar
            control=RxUART2();

        EnTx485();
        TxUART2(conectarp); //Conectem
        EnRx485();
        desconectar=0;
        while(desconectar==0) {
            control=RxUART2(); //Rebem una ordre

            switch(control) { //Comprobem ordre rebuda
                case(calibrarp): //Ordre de calibrar
                    pmBarReal=100*RxUART2();
                    pmVActualS=get_pressio(); //Rebem dBar que
                    //passem a mBar
                    pmVReal=mBartomV(pmBarReal);
                    pmVInit=pmVActualS-pmVReal;
                    EnTx485();
                    TxUART2(calibrarp);
                    EnRx485();
                    break;

                case(mostrejarp): //Ordre d'acquirir dades

```

```

press_exit=0;
num_mostres=RxUART2();
pmVMinim=mBartomV(100*RxUART2());
EnTx485();

/* Primers paquets */
while(num_mostres>PT && press_exit==0) {
    get_llumNoValida(); //Primera mostra no
                        //vàlida

/* Adquirim i enviem en paquets de 10 */
    for(i=0; i<PT; i++) {
        pSArray[i]=get_pressio();
        get_llum(llum, &tIntegracio,
&integral, &error);
        pEArray[i]=get_pressio();

        pSArray[i]=pSArray[i]-pmVInit;
        pEArray[i]=pEArray[i]-pmVInit;
        if(pEArray[i]>=pmVMinim)
            press_exit=1;
        for(j=0; j<256; j++)
            llArray[i][j]=llum[j];
    }

    for(i=0; i<PT; i++) {
        TxUART2(mostrejarp);
        TxUART216(mVtomBar(pSArray[i]));
        TxUART216(mVtomBar(pEArray[i]));
        for(j=0; j<256; j++)
            TxUART216(llArray[i][j]);
    }
    num_mostres=num_mostres-PT;
}

/* Ultims paquets */
get_llumNoValida(); //Primer mostra no vàlida
for(i=0; i<num_mostres; i++) {
    pSArray[i]=get_pressio();
    get_llum(llum, &tIntegracio,
&integral, &error);
    pEArray[i]=get_pressio();
    pSArray[i]=pSArray[i]-pmVInit;
    pEArray[i]=pEArray[i]-pmVInit;
    for(j=0; j<256; j++)
        llArray[i][j]=llum[j];
}

for(i=0; i<num_mostres; i++) {
    TxUART2(mostrejarp);
    TxUART216(mVtomBar(pSArray[i]));
    TxUART216(mVtomBar(pEArray[i]));
    for(j=0; j<256; j++)
        TxUART216(llArray[i][j]);
}

TxUART2(70);
EnRx485();
break;
case(desconectarp): //Ordre de desconectar
    desconectar=1;
    break;
}
}
}
}
}

```

A.6.2.ad977.c

```

#include <io30f6012.h>
#include "spi.h"

```

```

/* Mostreig del AD1 */
int SampleAD1(void) {
    int sample;
    RG12=0; //R/C AD1
    while(!RG13); //Esperem el BUSY
    RG12=1;
    sample=RxSPI1(); //Adquirim via SPI
    return sample;
}

/* Mostreig del AD2 */
int SampleAD2(void) {
    int sample;
    RC2=0; //R/C AD2
    while(!RC1); //Esperem el BUSY
    RC2=1;
    sample=RxSPI2(); //Adquirim via SPI
    return sample;
}

```

A.6.3.hamamatsu.c

```

#include <io30f6012.h>
#include "timernou.h" //Necessaria pel TIMER
#include "ad977.h"

#define MAX_OPTIM 53120 //Valor optim que hauria d'arribar el maxm de la llum
#define T_MAX 5859 //5859us*2*256=3s
#define T_MIN 50 //50us*2*256=25.6ms

/* Calcul del calor de llum màxim actual */
unsigned int get_max(unsigned int *data) {
    unsigned int current_max=0;
    int i;
    for(i=0; i<256; i++) {
        if(data[i]>current_max)
            current_max=data[i];
    }
    return current_max;
}

/* Adquisició rapida necessaria per posar el sensor a punt */
void get_llumNoValida(void){
    int i;
    RB1=0;
    retardus(50);
    RB0=0;
    retardus(50);
    RB1=1;
    retardus(100);
    RB0=1;
    for(i=1; i<256; i++) {
        retardus(100);
        RB0=0;
        retardus(100);
        RB0=1;
    }
    return;
}

/* Càlcul del nou temps d'integració */
unsigned int nou temps(float error, float integral, float derivada, float
temps_anterior) {
    float temps;

    if(error>=10)
        temps=(0.025*(error)+(temps_anterior)+ 0.001*(integral));
    //
    temps=(0.25*(error)+(temps_anterior)+ 0.01*(integral));
    else {
        if(error<=-10)

```

```

//          temps=((temps_anterior)-0.025*(-error)+0.001*(integral));
          temps=(0.25*(error)+(temps_anterior)+ 0.01*(integral));
      else temps=temps_anterior;
  }
  if(temps<T_MIN)
    temps=T_MIN;
  else {
    if(temps>T_MAX)
      temps=T_MAX;
  }
  return (unsigned int)temps;
}

/*      Adquisició d'un espectre complet      */
void get_llum(unsigned int *vector, unsigned int *temps, signed int *integral, signed
int *error_anterior) {
  unsigned int tempscicle;
  unsigned int llum_max;
  signed int error;
  unsigned int tempsnou;
  signed int derivada;
  int i;

  tempscicle=*temps*2;
  /*      Primera mostra, amb pols ST      */
  RB1=0;
  retardus(*temps);
  RB0=0;
  retardus(*temps);
  RB1=1;
  vector[0]=SampleAD1();
  retardus(*temps);
  RB0=1;

  /*      Les mostres restants      */
  for(i=1; i<256; i++) {
    retardus(tempscicle);
    RB0=0;
    retardus(*temps);
    vector[i]=SampleAD1();
    retardus(*temps);
    RB0=1;
  }

  llum_max=get_max(vector);
  error=MAX_OPTIM-llum_max;
  *integral=*integral+error;
  derivada=*error_anterior-error;
  tempsnou=nou_temps(error, *integral, derivada, *temps);
  *error_anterior=error;
  *temps=tempsnou;

  return;
}

```

A.6.4.init_basic.c

```

#include <io30f6012.h>

void InitBasic(void) {
  /*      Inicialització Ports I/O      */
  PORTB=0;
  TRISE=0x0000;
  PORTC=0;
  TRISC=0x0002; //RC1 (BUSY AD2) com a input
  PORTG=0;
  TRISG=0x2000; //RG13 (BUSY AD1) com a input

  /*      Inicialització sensor de llum */
  RB1=1; //ST
  RB0=1; //CLK
  RB2=0; //Vg (Control de guany)
}

```



```

    /*      Inicialització ADs      */
    RG12=1; //R/C AD1
    RC2=1; //R/C AD2
    return;
}

```

A.6.5.max485.c

```

#include <io30f6012.h>
#include "timernou.h"

/*      Conmutar a Mode Transmissió      */
void EnTx485(void) {
    retardus(1000);
    RG3=1;
    return;
}

/*      Conmutar a Mode Recepció      */
void EnRx485(void) {
    while(!U2STA_TRMT); //Esperem a acabar d'enviar tot
    RG3=0;
    return;
}

```

A.6.6.spi.c

```

#include <io30f6012.h>

/*      Inicialització del perifèric SPI      */
void InitSPI(void){
    /*      Habilitar les interrupcions      */
    /*      SPI1      */
    SPI1IF=0;
    SPI1IE=1;
    SPI1IP=000; //Sense prioritat
    /*      SPI2      */
    SPI2IF=0;
    SPI2IE=1;
    SPI2IP=000; //Sense prioritat

    /*      Configurar SPIxCON      */
    /*      SPI1CON      */
    SPI1CON_FRMEN=0; //Framed SPI desabilitada
    SPI1CON_DISSDO=1; //Només com a receptor
    SPI1CON_MODE16=1; //Transmissió de 16 bits
    SPI1CON_SMP=1; //Configuració de l'instant de mostreig. Veure grafiques
del manual
    SPI1CON_CKP=0;
    SPI1CON_CKE=0; //Configuració de la forma del 'dada clock'.Veure
grafiques del manual
    SPI1CON_SSEN=0; //Desabilitem el pin SS_negat. No l'utilitzarem
    SPI1CON_MSTEN=1; //Configurem la comunicació SPI com a master
    /*      SPI2CON      */
    SPI2CON_FRMEN=0; //Framed SPI desabilitada
    SPI2CON_DISSDO=1; //Només com a receptor
    SPI2CON_MODE16=1; //Transmissió de 16 bits
    SPI2CON_SMP=1; //Configuració de l'instant de mostreig. Veure grafiques
del manual
    SPI2CON_CKP=0;
    SPI2CON_CKE=0; //Configuració de la forma del 'dada clock'.Veure
grafiques del manual
    SPI2CON_SSEN=0; //Desabilitem el pin SS_negat. No l'utilitzarem
    SPI2CON_MSTEN=1; //Configurem la comunicació SPI com a master

    /*      Configuració dels rellotges. Partim de la base que FCY = 4MHz ( en MIPS )      */
    /*      SPI1      */
    SPI1CON_SPRE0=1;
    SPI1CON_SPRE1=1;
    SPI1CON_SPRE2=1;
    SPI1CON_PPRE0=0;

```

```

    SPI1CON_PPRES1=1; //SPI rate = 4MHz/1*16 = 250 KHz.
    /*      SPI2      */
    SPI2CON_SPRES0=1;
    SPI2CON_SPRES1=1;
    SPI2CON_SPRES2=1;
    SPI2CON_PPRES0=0;
    SPI2CON_PPRES1=1; //SPI rate = 4MHz/1*16 = 250 KHz.
    return;
}

/*      Habilitació Ports SPI      */
void EnSPI(void) {
    SPI1STAT_SPIEN=1;
    SPI2STAT_SPIEN=1;
    SPI2STAT_SPIROV=0;
    SPI2STAT_SPIROV=0;
    return;
}

/*      Deshabilitació Ports SPI      */
void DisSPI(void) {
    SPI1STAT_SPIEN=0;
    SPI2STAT_SPIEN=0;
    return;
}

/*      Recepció d'una mostra AD1      */
int RxSPI1(void) {
    int bytes;
    while(SPI1STAT_SPIRBF); //Esperem que hagi acabat d'enviar
    SPI1BUF = 0x0000; //Es necessari enviar per poder rebre
    (generar clock)
    while(!SPI1STAT_SPIRBF); //Esperem a rebre algo
    bytes=SPI1BUF;
    return bytes;
}

/*      Recepció d'una mostra AD2      */
int RxSPI2(void) {
    int bytes;
    while(SPI2STAT_SPIRBF); //Esperem que hagi acabat d'enviar
    SPI2BUF = 0x0000; //Es necessari enviar per poder rebre
    (generar clock)
    while(!SPI2STAT_SPIRBF); //Esperem a rebre algo
    bytes=SPI2BUF;
    return bytes;
}

```

A.6.7.timernou.c

```

#include <io30f6012.h>

//MIN 16us, MAX 65535us=65.5ms
void retardus(int us) {
    TMR1 = 0; // Clear Timer 1
    PR1 = (us*4)-4; // Configurem el periode (temps d'espera fins que salta la
    interrupcio)
    T1IF = 0; // Clear interrupt flag
    T1CON = 0x8000; // Fosc/4, 1:1 prescaler, Start TMR1
    while(T1IF == 0); //Esperem que salti la interrupció
    return;
}

```

A.6.8.uart.c

```

#include <io30f6012.h>

#define BAUDRATE 9600 //Velocitat en bps
#define FREQ 4000000 //Frequència en MIPS

/*      Calcul del BaudRate      */
unsigned int CalcBR2(void) {

```

```

    float br;
    br=((float)FREQ/(16.0*(float)BAUDRATE));
    return br;
}

/* Inicialització del perifèric UART1 */
void InitUART1(void) {
    U1MODE=0x8000; //Habilitem UART
    U1STA=0x0000; //Configuració per defecte: 8 bits, sense paritat, 1 bit stop
    U1BRG=CalcBR2();
    NSTDIS=1; //Disable nested interrupts
    U1STA_UTXEN=1; //Habilitar transmissió
    return;
}

/* Enviament d'un byte */
void TxUART1(int byte) {
    while(U1STA_UTXBF);
    U1TXREG=byte;
    return;
}

/* Enviament de dos bytes*/
void TxUART116(int bytes) {
    while(U1STA_UTXBF);
    U1TXREG=bytes;
    bytes>>=8;
    while(U1STA_UTXBF);
    U1TXREG=bytes;
    return;
}

/* Recepció d'un byte */
char RxUART1(void) {
    char byte;
    while(!U1STA_URXDA);
    byte=U1RXREG;
    return byte;
}

/* Recepció de dos bytes */
int RxUART116(void) {
    int temps;
    char byte1;
    char byte2;
    byte1=RxUART1();//Més pes
    byte2=RxUART1();//Menys pes
    temps=(byte1*256)+byte2;
    return temps;
}

/* Deshabilitació de UART1 */
void EndUART1(void) {
    U1MODE_UARTEN=0; //Deshabilitem UART
    return;
}

/* ----- UART2 ----- */
/* Inicialització del perifèric UART2 */
void InitUART2(void) {
    U2MODE=0x8000; //Habilitem UART
    U2STA=0x0000; //Configuració per defecte: 8 bits, sense paritat, 1 bit stop
    U2BRG=CalcBR2(); //Configurar BaudRate
    NSTDIS=1; //Disable nested interrupts
    U2STA_UTXEN=1; //Habilitar transmissió
    return;
}

/* Enviament d'un byte */
void TxUART2(int byte) {
    while(U2STA_UTXBF); //Esperem que hi hagi espai al buffer de Tx
    U2TXREG=byte;
    return;
}

```

```

/*      Enviament de dos bytes*/
void TxUART216(int bytes) {
    while (U2STA_UTXBF);
    U2TXREG=bytes;
    bytes>>=8;
    while (U2STA_UTXBF);
    U2TXREG=bytes;
    return;
}

/*      Recepció d'un byte      */
char RxUART2(void) {
    char byte;
    while (!U2STA_URXDA); //Esperem que hi hagi algun byte al buffer de Rx
    byte=U2RXREG;
    return byte;
}

/*      Recepció de dos bytes */
int RxUART216(void) {
    int temps;
    char byte1;
    char byte2;
    while (!U2STA_URXDA); //Esperem que hi hagi algun byte al buffer de Rx
    byte1=RxUART2(); //Més pes
    while (!U2STA_URXDA); //Esperem que hi hagi algun byte al buffer de Rx
    byte2=RxUART2(); //Menys pes
    temps=(byte1*256)+byte2;
    return temps;
}

/*      Deshabilitació de UART2      */
void EndUART2(void) {
    U2MODE_UARTEN=0; //Deshabilitem UART
    return;
}

```