UNIVERSITAT OBERTA DE CATALUNYA

Enginyeria tècnica de telecomunicacions

Especialitat en Telemàtica

Disseny d'un expansor d'entrades i sortides analògiques per ModBus RTU sobre RS-485

Alumne/a: Laura Solà i Sanmiquel

Dirigit per: Carlos Alberto Pacheco Navas

CURS 2012-2013 (Febrer)

A tota la meva família i amics pel seu suport incondicional.
Un agraïment molt especial al director d'aquest projecte, Carlos Pacheco, per la seva continuada ajuda durant tot el semestre.

Resum

La majoria de processos industrials en les línies de fabricació utilitzen sistemes que incorporen autòmates, sistemes encastats, adaptadors d'interfícies i equips similars. Aquests sistemes es connecten entre ells per actuar d'acord amb un algoritme de control programat en aquests equips.

Els medis de connexió d'equips són molt variats i normalment se n'utilitza més d'un. Una interfície molt utilitzada en el món industrial és el bus RS-485. Els protocols que s'utilitzen per transmetre la informació també són molt diversos, però un dels més coneguts és ModBus.

Aquest treball està basat en equips adaptadors d'interfícies, és a dir, en equips que fan un canvi de medi per permetre la lectura i l'escriptura de diferents paràmetres des d'un determinat medi físic, utilitzant una altra interfície.

En concret, l'objectiu del treball és dissenyar un sistema que permeti adquirir valors des de diferents entrades analògiques utilitzant una interfície RS-485.

Índex de Continguts

1. Introducció 6
2. Objectius 7
3. Planificació 8
3.1. Tasques 8
3.2. Fites 10
3.3. Planificació temporal 10
3.4. Diagrama de Gantt 12
3.5. Riscos i pla de contingència 14
3.6. Avaluació del material 14
4. Estudi del protocol ModBus RTU i la interfície RS-485 16
4.1. Protocol ModBus RTU 16
4.1.1. Funcions ModBus més rellevants 19
4.2. Interfície RS-485 19
5. Circuits d'adaptació d'entrades i sortides 21
5.1. Estudi del sistema 21
5.2. Disseny dels circuits electrònics 21
5.2.1. Adaptació del voltatge d'entrada 22
5.2.2. Adaptació de corrent a voltatge d'entrada 23
5.2.3. Adaptació del voltatge de sortida 26
6. Disseny de la font d'alimentació 28
6.1. Anàlisi de les tensions i corrents de la font 28
6.2. Disseny dels circuits electrònics 30
7. El mapa de memòria ModBus 35
7.1. Anàlisi del sistema 35
7.2. El mapa de memòria ModBus 36
8. El microprocessador 40
8.1. Selecció del microprocessador 40
8.2. El transductor RS-485 42
8.3. Programació del microcontrolador 43
9. Disseny de la PCB 51
10. Conclusions 56
11. Futures ampliacions 58
12. Bibliografia 59

Índex de Taules

Taula 1 – Llista de tasques i precedències 10
Taula 2 – Llistat de fites i dates de finalització 10
Taula 3 – Planificació temporal de les tasques i les fites 12
Taula 4 – Riscos i mitigacions identificats a l'inici del projecte 14
Taula 5 – Format del protocol ModBus RTU 16
Taula 6 – Funcions de ModBus 17
Taula 7 – Funcions de ModBus de l'equip 35
Taula 8 – Espai de referència ModBus 36
Taula 9 – Entrades ModBus 36
Taula 10 – Funcions ModBus dels registres 3x 36
Taula 11 – Sortides ModBus 37
Taula 12 – Funcions ModBus dels registres 4x 37
Taula 13 – Codis d'excepció ModBus 38
Taula 14 – Mapa de memòria ModBus 38
Taula 15 – Relació entre adreces i pins del microprocessador 41

Índex de Figures

Figura 1 – Diagrama de Gantt 13
Figura 2 – Diagrama de dades ModBus 17
Figura 3 – Diagrama de transaccions d'estat ModBus 18
Figura 4 – Topologia de xarxa multipunt RS-485 20
Figura 5 – Circuit d'adaptació de la tensió d'entrada 22
Figura 6 – Simulació de l'adaptació de la tensió d'entrada 23
Figura 7 – Circuit d'adaptació de corrent a tensió d'entrada 24
Figura 8 – Corrent a la font de tensió de 4.125 V 24
Figura 9 – Circuit d'adaptació de corrent a tensió d'entrada modificat 25
Figura 10 – Simulació de l'adaptació del corrent a tensió d'entrada 25
Figura 11 – Circuit adaptació tensió sortida 26
Figura 12 – Simulació de l'adaptació de la tensió de sortida 27
Figura 13 – Corrent a les fonts de 15 V (línia verda) i -15 V (línia groga) del circuit d'adaptació de voltatge d'entrada 28
Figura 14 – Corrent a les fonts de 15 V (línia groga) i -15 V (línia verda) del circuit d'adaptació de corrent d'entrada 29

Figura 15 – Corrent a les fonts de 15 V (línia verda) i -15 V (línia groga) del circuit d'adaptació de voltatge de sortida amb la màxima corrent de sortida..... 29
Figura 16 – Circuit d'adaptació DCDC de 15 V a 3.3 V..... 30
Figura 17 – Eficiència dels circuits DCDC de 15 V a 3.3 V 31
Figura 18 – Circuit d'adaptació DCDC de 15 V a -15 V 32
Figura 19 – Eficiència del circuit DCDC de 15 V a -15 V..... 32
Figura 20 – Circuit d'adaptació DCDC de la font d'alimentació a 15 V 33
Figura 21 – Eficiència del circuit DCDC de la font d'alimentació a 15 V..... 33
Figura 22 – Característiques principals del microcontrolador seleccionat..... 40
Figura 23 – Diagrama de pins del microcontrolador..... 41
Figura 24 – Mapa de memòria FLASH del microcontrolador 42
Figura 25 – Comunicació transductor Maxim 3483E 43
Figura 26 – Xarxa half duplex transductor Maxim 3483E..... 43
Figura 27 – Esquemàtic del circuit..... 53
Figura 28 – Layout de la PCB 54
Figura 29 – Layout de la cara superior de la PCB amb el pla de massa..... 55

1. Introducció

En aquest projecte es dissenya un expansor d'entrades i sortides analògiques amb el protocol ModBus RTU sobre la interfície RS-485. Es tracta d'un equip que pot llegir les dades recollides per un equip extern, com per exemple un sensor, passar-les a un format que pugui llegir un processador; i un cop fet el processat generar unes sortides altre cop analògiques perquè es puguin entendre des de l'exterior.

Es treballa amb equips adaptadors d'interfície, que fan un canvi de medi perquè es puguin llegir i/o escriure certs paràmetres d'un medi físic, utilitzant una altra interfície que sigui més còmoda. Tant el protocol ModBus com la interfície RS-485 són molt utilitzats en el món industrial.

Aquest expansor consta de diferents mòduls, que s'han anat dissenyant al llarg del projecte, tal i com indica la planificació proposada en l'apartat 3.3.

Les parts que componen aquest sistema expansor i els seus requeriments tècnics són els següents:

- Interfície A: considerem que és la interfície RS-485, que és d'entrada i de sortida. És la que implementa el protocol ModBus. Se n'ha de poder configurar la velocitat de transmissió, la paritat i l'identificador d'adreça de dispositiu per poder escriure a registres ModBus. Cal que estigui aïllada elèctricament.
- Interfície B: és la de les entrades i sortides analògiques, on es vol llegir i escriure.
- Adaptador d'entrades analògiques: circuit que adapta el senyal que es mesura de manera que el microcontrolador en pugui adquirir les seves mesures. Ha de tenir dues entrades analògiques per a mesurar tensions d'entre 0 V i 10 V, amb un mínim 10 bits de precisió, i també dues entrades més per mesurar corrent passiu de 4 mA a 20 mA.
- Adaptació de sortides analògiques: circuit que adapta el senyal de sortida del microcontrolador per generar la sortida. Ha de tenir dues sortides analògiques que permetin mesurar tensions d'entre 0 V i 10 V, també amb un mínim de 10 bits de precisió, i donant fins a 200 mA de corrent cadascuna.
- Circuit de control: que implementa el protocol ModBus.
- Condicionador de senyal o transductor RS-485: que condiciona el senyal del controlador a RS-485, per tal que es pugui tractar.
- Alimentació: adapta les tensions de la font d'alimentació segons les tensions necessàries per a cada component de l'expansor. Ha de poder-se alimentar amb qualsevol tensió contínua entre 12 V i 48 V.

A més, també es fa el disseny del *layout* de la PCB que inclou tots els circuits electrònics dissenyats, de manera que aquesta ja es pugui enviar a fabricar i muntar.

Bàsicament, el treball es desenvolupa en diferents tasques, cadascuna de les quals tracta un bloc concret del disseny de l'expansor, com ja s'ha esmentat anteriorment. És per això que aquesta memòria també s'organitza en diferents apartats, corresponents als diferents mòduls i estadis de disseny que es duen a terme.

2. Objectius

L'objectiu d'aquest projecte és dissenyar un equip que permeti adquirir valors de diferents entrades analògiques, i establir-ne posteriorment les seves sortides, també analògiques, utilitzant el protocol ModBus i la interfície RS-485.

En termes generals, l'objectiu final d'aquest projecte és comprendre i entendre, i per últim, posar en pràctica els continguts teòrics que s'han anat adquirint al llarg de la carrera d'Enginyeria Tècnica de Telecomunicacions. A la vegada, també es tracta de desenvolupar aquest projecte tal i com es gestionaria en una empresa. És a dir, tenint en compte la gestió del projecte com a tal, des de la seva bona planificació inicial (incloent la planificació temporal i l'estimació del temps que porta cada tasca), el seguiment, la gestió de riscos i de costos, les lliçons apreses, la identificació de documents entregables (fites), etc. A més cal fer un bon anàlisi dels resultats per tal de treure'n unes conclusions clares i concises. I finalment, convé proposar línies futures d'ampliació del treball.

Els principals objectius tècnics, ja concrets per aquest projecte en sí, són els següents:

- Dissenyar la font d'alimentació.
- Dissenyar els circuits d'adaptació d'entrades i sortides a nivell del microcontrolador per mesurar les entrades i generar les sortides analògiques.
- Definició del mapa de memòria ModBus.
- Elecció del processador i implementació del seu programa de control per a la correcta manipulació i processament dels senyals.
- Disseny del *layout* de la PCB que inclogui tots els circuits i components de l'expansor.

Aquests objectius per a ser mesurables hauran de complir les especificacions tècniques, que ja s'han descrit en la introducció.

L'assoliment de tots aquests objectius, tant els tècnics com els més generals, és clau per un bon desenvolupament del projecte.

3. Planificació

Expliquem a continuació com s'ha plantejat el desenvolupament d'aquest treball durant el semestre. A l'inici del projecte es van identificar les tasques necessàries, i les majors fites que calia abordar. Se'n va fer la planificació temporal en funció de l'esforç que es creia que cada tasca requeriria, i a la vegada es van fixar les dates d'entrega de les PACs. Cal esmentar que les dates d'entrega de les PACs 2 i 3 es van fixar segons es creia millor per una bona planificació de les tasques al llarg del semestre. Totes aquestes tasques i fites es presenten en un diagrama de Gantt per a una millor visualització, cosa que ha estat de gran ajuda per poder fer un bon seguiment del projecte al llarg del seu desenvolupament.

3.1. Tasques

En aquest apartat es presenta l'avaluació inicial que es va fer de totes les tasques que s'havien de dur a terme per tal de poder assolir els objectius finals del treball. A partir de les tasques identificades a alt nivell, es van identificar diferents subtasques per cadascuna d'elles, que es detallen a la taula de sota. En aquesta taula també s'indiquen les tasques que les precedeixen, és a dir, que no es podien començar fins que la tasca indicada s'hagués completat.

Es va considerar que, en principi, la majoria de tasques no es farien en paral·lel, tot i que en la majoria de casos és més una condició imposada per nosaltres, és a dir per una limitació dels recursos humans disponibles, que no pas que no sigui tècnicament possible. Tot i així, en les precedències que indiquem a sota, aquestes no s'han marcat com a tasques precedents, sinó només les que venen estrictament imposades per les pròpies tasques en sí. Ara bé, en el diagrama de Gantt si que es van incloure aquestes precedències addicionals per tal de poder planificar millor les tasques en el temps. A la taula de sota també s'hi afegeix una estimació de les hores que a l'inici del projecte es va pensar que cada tasca requeriria.

Cal dir que es va considerar que les entregues de les PACs 2 i 3 no són tasques en sí, ja que no formen part del projecte, sinó que es consideren com a fites. Aquestes fites les descriurem en l'apartat 3.2, i també s'incorporen a la planificació del projecte.

També esmentar que les hores dedicades a la instal·lació de programari i a aprendre com funciona el software pel disseny i les simulacions, estan incloses dins d'aquestes tasques.

Tasca	Precedents	Hores
1. Inici del projecte de final de carrera		-
1.1 Llegir enunciat del projecte		1
1.2 Trobada d'inici	1.1	2
2. Desenvolupament del Pla de Treball del Projecte		-
2.1 Lectura dels materials 'Treball de Final de Carrera'		20
2.2 Definició de les tasques del projecte	1.1	1
2.3 Fer la planificació temporal de les tasques identificades	2.2	1
2.4 Fer un anàlisi de riscos del projecte		1
2.5 Redacció del Pla de Treball	2.2	2
3. Estudi del protocol ModBus i la interfície RS-485		-
3.1 Buscar documentació apropiada pel nostre projecte del protocol ModBus i el bus RS-485, tenint en compte les funcions i opcions que	1.1	2

s'utilitzaran en aquest projecte		
3.2 Lectura de la documentació de manera detallada i assegurant-ne la seva comprensió. Aclariment de dubtes amb el consultor si és necessari.	3.1	10
3.3 Redacció del capítol corresponent a la memòria resumint aquest estudi.	3.2+1 dia	2
4. Disseny dels circuits d'adaptació d'entrades i sortides		-
4.1 Estudi del processador per esbrinar com adquirirà les entrades i generarà les sortides per conèixer els nivells que cal adaptar. Selecció dels components electrònics del circuit.		4
4.2 Disseny dels circuits analògics per mesurar les entrades i adaptar-les correctament per tal que el controlador en pugui adquirir el seu valor	4.1	10
4.3 Disseny dels circuits analògics que permetin al controlador generar les sortides requerides	4.1	10
4.4 Programació del codi del microprocessador (opcional)	4.2 i 4.3	-
4.5 Simulació dels circuits	4.2 i 4.3	5
4.6 Redacció del capítol de la memòria corresponent al disseny dels circuits d'adaptació d'entrades i sortides	4.1	2
5. Disseny de la font d'alimentació		-
5.1 Anàlisis dels consums necessaris d'entrada i sortida de la font		2
5.2 Selecció dels components electrònics adequats segons les tensions de cada un dels blocs	5.1	3
5.3 Disseny del circuit, tenint en compte l'eficiència de la font	5.2	10
5.4 Simulació de la font d'alimentació	5.3	5
5.5 Redacció de la documentació per a la memòria	5.1	2
6. Definició del mapa de memòria ModBus		-
6.1 Disseny del mapa de memòria: informació de cada posició, unitats, format, ordre de bytes, funcions per accedir-hi, etc		10
6.2 Generació de la documentació amb les especificacions ModBus de l'equip	6.1	10
6.3 Escriptura de la part de la memòria corresponent al mapa de memòria ModBus	6.1	2
7. Elecció del processador i implementació del programa de control		-
7.1 Selecció del processador per la implementació del protocol ModBus RTU		2
7.2 Programació de l'algorisme de control intern	7.1	10
7.3 Programació de l'algorisme de gestió del protocol de comunicacions ModBus	7.1 i 7.2	10
7.4 Validació dels programes mitjançant simulacions	7.2 i 7.3	5
7.5 Redacció de l'apartat de la memòria corresponent a aquesta tasca	7.1	2
8. Disseny del *layout* de la PCB		-
8.1 Disseny del *layout* de la PCB amb tots els circuits elèctrics de l'expansor		10
8.2 Redacció de la part de la memòria referent al *layout* de la PCB	8.1	2
9. Acabament de la Memòria i la presentació i debat final		-
9.1 Redacció de les conclusions del treball incloent una reflexió sobre els objectius, els problemes, solucions i les lliçons apreses, i fer una avaluació de les desviacions del calendari i dels riscos del projecte		2
9.2 Redacció de la part de la memòria referent a possibles ampliacions futures del projecte	9.1	2
9.3 Preparació de la presentació del projecte en Power Point	9.1	4
9.4 Gravació de la presentació	9.3	6

9.1 Respondre preguntes del tribunal - Debat	9.1, 9.2, 9.3, 9.4	2
TOTAL D'HORES PREVISTES		**174**

Taula 1 – Llista de tasques i precedències

3.2. Fites

A continuació s'exposen les fites que es va pensar que eren rellevants pel correcte desenvolupament del projecte. Bàsicament venen definides per les entregues que cal anar fent durant el semestre, però també s'hi va afegir la reunió inicial i els lliuraments dels esborranys previs que s'han enviat al consultor per a una revisió prèvia a l'entrega de cada una de les PACs.

Com ja s'ha comentat més amunt, els lliuraments de les PACs 2 i 3 es van fixar a l'inici del projecte. Es va creure adequat col·locar les entregues d'aquestes dues PACs de manera que quedessin espaiades en el temps durant el semestre. Per aquesta mateixa raó, es va planificar que la PAC 2 inclogués les tasques 3, 4 i 5, i que la PAC 3, inclogués les tasques 6, 7 i 8. La tasca 1 és una tasca prèvia d'inici del projecte, i la tasca 2 fa referència al Pla de Treball. La tasca 9 forma part de l'entrega final de la memòria i la presentació. També cal dir que es va considerar que es tardaria 6 setmanes a desenvolupar les tasques de la PAC 2, i 7 setmanes per a les tasques de la PAC 3, ja que es va preveure que serien lleugerament més complicades.

Es van establir les següents fites que es va creure que s'haurien de complir pel bon desenvolupament del projecte:

Fita	Data
Reunió inicial	22/09/2012
Entrega esborrany Pla de Treball	27/09/2012
Entrega Pla de Treball	02/10/2012
Entrega esborrany PAC 2	28/10/2012
Entrega PAC 2	03/11/2012
Entrega esborrany PAC 3	12/12/2012
Entrega PAC 3	19/12/2012
Entrega esborrany memòria	02/01/2013
Entrega esborrany presentació	04/01/2013
Entrega memòria i presentació	07/01/2013

Taula 2 – Llistat de fites i dates de finalització

3.3. Planificació temporal

S'exposen a continuació les diferents tasques que ja s'han explicat a l'apartat 3.1, i les fites que s'han llistat a l'apartat 3.2, amb la corresponent planificació temporal.

Tot i que es va intentar fer una planificació el més acurada possible segons la informació que es tenia del projecte als seus estadis inicials, es va pensar que molt possiblement aquesta podria canviar lleugerament a mesura que aquest evolucionés i s'anés entenent millor l'abast real de cada una de les tasques.

Per fer aquesta planificació es va tenir en compte que la disponibilitat canvia setmana a setmana, però sempre s'acostuma a tenir més temps durant els caps de setmana. Per tant, es va avaluar-la pensant que durant els caps de setmana s'hi dedicarien més

hores que durant la setmana. De manera genèrica es va considerar que es dedicarien onze hores setmanals al projecte, sis de les quals serien durant la setmana, i les cinc restants durant el cap de setmana. Així doncs, tenint en compte que la durada del projecte és d'aproximadament 15 setmanes, el total d'hores dedicades serien unes 165. Veiem que, més o menys, aquest nombre d'hores està en consonància amb les hores estimades en l'apartat 3.1. Tot i així, ja es va preveure que durant alguns períodes puntuals caldria posar-hi algunes hores més de les habituals.

Es va proposar la següent planificació, on s'hi inclouen les dates d'inici i de finalització de cadascuna de les tasques, a més de les fites que cal assolir. Les fites estan en cursiva per tal de diferenciar-les de les tasques.

Tasca o fita	**Data inici**	**Data final**
1. Inici del projecte de final de carrera	19/09/2012	22/09/2012
1.1 Llegir enunciat del projecte	19/09/2012	21/09/2012
1.2 Trobada d'inici	22/09/2012	22/09/2012
2. Desenvolupament del Pla de Treball del Projecte	23/09/2012	02/10/2012
2.1 Lectura dels materials 'Treball de Final de Carrera'	23/09/2012	02/10/2012
2.2 Definició de les tasques del projecte	23/09/2012	24/09/2012
2.3 Fer la planificació temporal de les tasques identificades	25/09/2012	25/09/2012
2.4 Fer un anàlisi de riscos del projecte	23/09/2012	25/09/2012
2.5 Redacció del Pla de Treball	25/09/2012	02/10/2012
Lliurament de l'Esborrany del Pla de Treball	*27/09/2012*	
Lliurament del Pla de Treball	*02/10/2012*	
3. Estudi del protocol ModBus i el bus RS-485	24/09/2012	03/10/2012
3.1 Buscar documentació del protocol ModBus i el bus RS-485	24/09/2012	27/09/2012
3.2 Lectura de la documentació	28/09/2012	02/10/2012
3.3 Redacció del capítol corresponent	29/09/2012	03/10/2012
4. Disseny dels circuits d'adaptació d'entrades i sortides	03/10/2012	18/10/2012
4.1 Estudi inicial del processador i selecció de components	03/10/2012	05/10/2012
4.2 Disseny dels circuits analògics per mesurar les entrades	06/10/2012	12/10/2012
4.3 Disseny dels circuits analògics que permetin al controlador generar les sortides	06/10/2012	12/10/2012
4.4 Programació del codi del microprocessador *(opcional)*	13/10/2012	18/10/2012
4.5 Simulació dels circuits	13/10/2012	18/10/2012
4.6 Redacció del capítol de la memòria corresponent	06/10/2012	18/10/2012
5. Disseny de la font d'alimentació	18/10/2012	03/10/2012
5.1 Anàlisis dels consums necessaris d'entrada i sortida de la font	18/10/2012	19/10/2012
5.2 Selecció dels components electrònics adequats	20/10/2012	21/10/2012
5.3 Disseny del circuit	22/10/2012	26/10/2012
5.4 Simulació de la font d'alimentació	27/10/2012	29/10/2012
5.5 Redacció de la documentació per a la memòria	20/10/2012	03/10/2012
Lliurament de l'esborrany de la PAC 2	*28/10/2012*	
Lliurament de la PAC 2	*03/11/2012*	
6. Definició del mapa de memòria ModBus	03/11/2012	13/11/2012
6.1 Disseny del mapa de memòria	03/11/2012	09/11/2012
6.2 Generació de la documentació amb les especificacions ModBus de l'equip	10/11/2012	13/11/2012
6.3 Escriptura de la part de la memòria corresponent	10/11/2012	13/11/2012
7. Elecció del processador i implementació del programa de control	14/11/2012	30/11/2012
7.1 Selecció del processador per la implementació del protocol	14/11/2012	17/11/2012

ModBus RTU		
7.2 Programació de l'algorisme de control intern	18/11/2012	22/11/2012
7.3 Programació de l'algorisme de gestió del protocol de comunicacions ModBus	23/11/2012	26/11/2012
7.4 Validació dels programes	27/11/2012	30/11/2012
7.5 Redacció de l'apartat de la memòria corresponent	18/11/2012	30/11/2012
8. Disseny del *layout* de la PCB	01/12/2012	19/12/2012
8.1 Disseny del layout de la PCB	01/12/2012	09/12/2012
8.2 Redacció de la part de la memòria	10/12/2012	19/12/2012
Lliurament de l'esborrany de la PAC 3	*12/12/2012*	
Lliurament de la PAC 3	*19/12/2012*	
9. Acabament de la Memòria i la presentació i debat final	19/12/2012	25/01/2013
9.1 Redacció de les conclusions del treball	19/12/2012	28/12/2012
9.2 Redacció de la part de la memòria referent a possibles ampliacions futures del projecte	29/12/2012	02/01/2013
9.3 Preparació de la presentació del projecte en Power Point	29/12/2012	04/01/2013
9.4 Gravació de la presentació	05/01/2013	07/01/2013
9.5 Respondre preguntes del tribunal - Debat	23/01/2013	25/01/2013
Lliurament de l'esborrany de la memòria	*02/01/2013*	
Lliurament de l'esborrany de la presentació	*04/01/2013*	
Lliurament de la memòria i la presentació	*07/01/2013*	

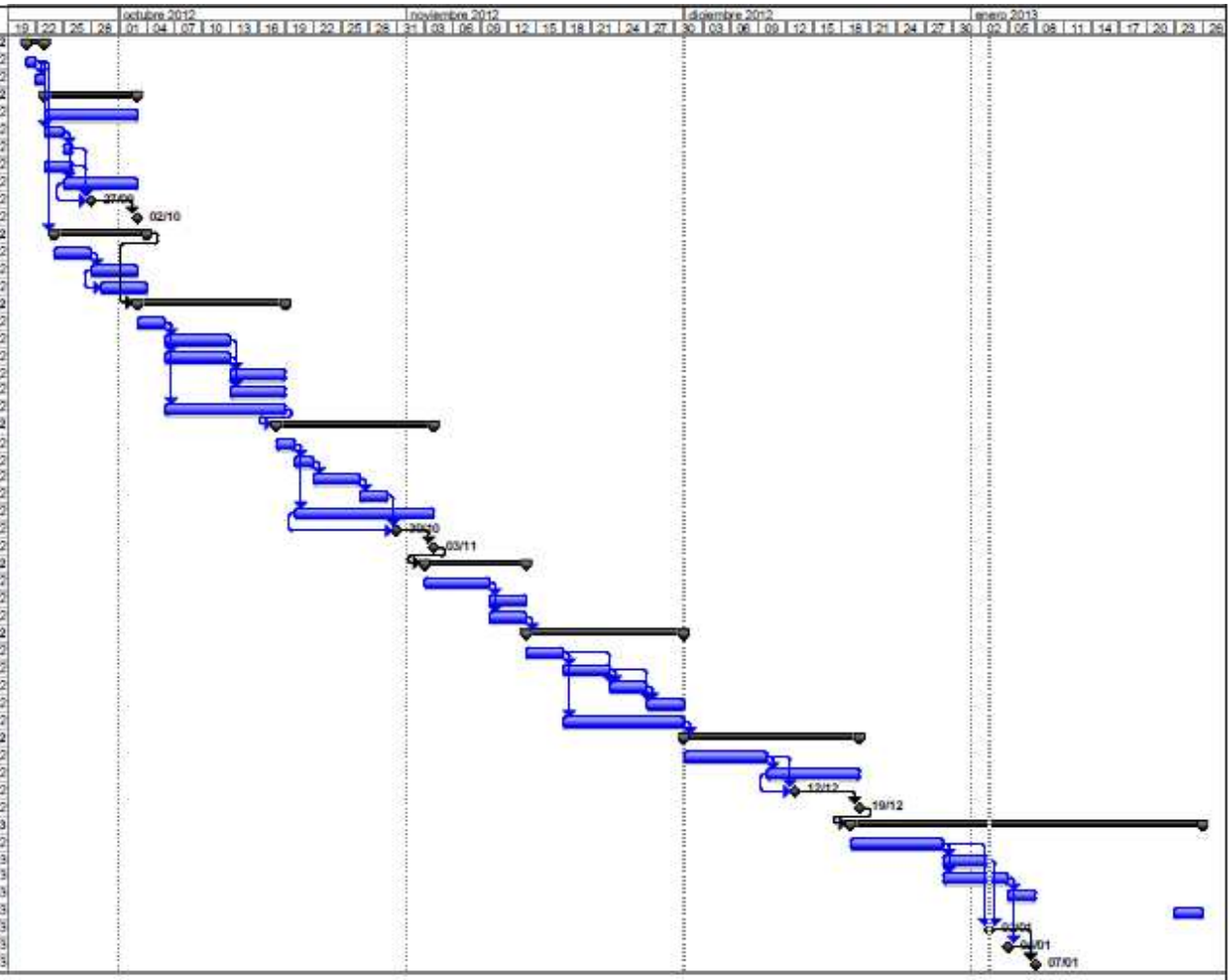
Taula 3 – Planificació temporal de les tasques i les fites

3.4. Diagrama de Gantt

A continuació es presenta el diagrama de Gantt que reflecteix totes les tasques i fites exposades en els apartats 3.1 i 3.2 amb les seves corresponents relacions de precedència, segons la planificació temporal proposada en l'apartat 3.3. Cal afegir, com ja s'ha indicat anteriorment, que en aquest diagrama s'hi han incorporat algunes relacions de precedència entre fites i tasques, o entre tasques i tasques, que no s'havien indicat a l'apartat 3.1. La raó és que en l'apartat 3.1 només s'han indicat les precedències estrictament avaluades des d'un punt de vista tècnic, és a dir, que una tasca no pot començar sense haver-ne acabat una altra perquè encara no s'ha fet la part anterior disseny, per exemple. En el diagrama de Gantt s'hi ha afegit altres precedències, marcades per nosaltres, per a una bona planificació estrictament temporal.

Es presenta el diagrama de Gantt en la Figura 1.

Id	Nombre de tarea	Duración	Comienzo	Fin
1	**Inici del Projecte**	**2 días?**	**vie 21/09/12**	**sáb 22/09/12**
2	Llegir enunciat del projecte	1 día?	vie 21/09/12	vie 21/09/12
3	Trobada d'Inici	1 día?	sáb 22/09/12	sáb 22/09/12
4	**Desenvolupament del Pla de T**	**10 días?**	**dom 23/09/12**	**mar 02/10/12**
5	Lectura dels materials 'Tret	10 días?	dom 23/09/12	mar 02/10/12
6	Definició de les tasques de	2 días?	dom 23/09/12	lun 24/09/12
7	Planificació temporal de les	1 día?	mar 25/09/12	mar 25/09/12
8	Anàlisis de riscos	3 días?	dom 23/09/12	mar 25/09/12
9	Redacció del Pla de Treball	8 días?	mar 25/09/12	mar 02/10/12
10	Lliurament de l'Esborrany del Pl	1 día?	jue 27/09/12	jue 27/09/12
11	Lliurament del Pla de Treball	1 día?	mar 02/10/12	mar 02/10/12
12	**Estudi del protocol ModBus i**	**10 días?**	**lun 24/09/12**	**mié 03/10/12**
13	Buscar documentació Modi	4 días?	lun 24/09/12	jue 27/09/12
14	Lectura documentació Mod	5 días?	vie 28/09/12	mar 02/10/12
15	Redacció capítol memòria	5 días?	sáb 29/09/12	mié 03/10/12
16	**Disseny dels circuits d'aptacio**	**16 días?**	**mié 03/10/12**	**jue 18/10/12**
17	Estudi inicial del processad	3 días?	mié 03/10/12	vie 05/10/12
18	Disseny dels circuits analòg	7 días?	sáb 06/10/12	vie 12/10/12
19	Disseny dels circuits analòg	7 días?	sáb 06/10/12	vie 12/10/12
20	Programació del microproc	6 días?	sáb 13/10/12	jue 18/10/12
21	Simulació dels circuits	6 días?	sáb 13/10/12	jue 18/10/12
22	Redacció capítol memòria	13 días?	sáb 06/10/12	jue 18/10/12
23	**Disseny de la font d'alimentac**	**17 días?**	**jue 18/10/12**	**sáb 03/11/12**
24	Anàlisi consums entrada i s	2 días?	jue 18/10/12	vie 19/10/12
25	Selecció components elect	2 días?	sáb 20/10/12	dom 21/10/12
26	Disseny circuit	5 días?	lun 22/10/12	vie 26/10/12
27	Simulació font alimentació	3 días?	sáb 27/10/12	lun 29/10/12
28	Redacció capítol memòria	15 días?	sáb 20/10/12	sáb 03/11/12
29	Lliurament esborrany PAC2	1 día?	mar 30/10/12	mar 30/10/12
30	Lliurament PAC 2	1 día?	sáb 03/11/12	sáb 03/11/12
31	**Definició del mapa de memòri**	**11 días?**	**sáb 03/11/12**	**mar 13/11/12**
32	Disseny del mapa de mem	7 días?	sáb 03/11/12	vie 09/11/12
33	Generació d'especificacion	4 días?	sáb 10/11/12	mar 13/11/12
34	Redacció capítol	4 días?	sáb 10/11/12	mar 13/11/12
35	**Elecció del processador i imp**	**17 días?**	**mié 14/11/12**	**vie 30/11/12**
36	Selecció processador	4 días?	mié 14/11/12	sáb 17/11/12
37	Programació algorisme de	5 días?	dom 18/11/12	jue 22/11/12
38	Programació de l'algorisme	4 días?	vie 23/11/12	lun 26/11/12
39	Validació dels programes	4 días?	mar 27/11/12	vie 30/11/12
40	Redacció capítol memòria	13 días	dom 18/11/12	vie 30/11/12
41	**Disseny del layout de la PCB**	**19 días?**	**sáb 01/12/12**	**mié 19/12/12**
42	Disseny layout PCB	9 días?	sáb 01/12/12	dom 09/12/12
43	Redacció capítol memòria	10 días?	lun 10/12/12	mié 19/12/12
44	Lliurament de l'esborrany PAC3	1 día?	mié 12/12/12	mié 12/12/12
45	Lliurament PAC3	1 día?	mié 19/12/12	mié 19/12/12
46	**Acabament memòria, present**	**34 días?**	**mié 19/12/12**	**vie 25/01/13**
47	Redacció conclusions proje	10 días?	mié 19/12/12	vie 28/12/12
48	Redacció possibles ampliac	5 días?	sáb 29/12/12	mié 02/01/13
49	Presentació projecte Powe	7 días?	sáb 29/12/12	vie 04/01/13
50	Gravació presentació	3 días?	sáb 05/01/13	lun 07/01/13
51	Respondre preguntes Debo	3 días?	mié 23/01/13	vie 25/01/13
52	Lliurament esborrany de la mem	1 día?	mié 02/01/13	mié 02/01/13
53	Lliurament esborrany presentac	1 día?	vie 04/01/13	vie 04/01/13
54	Lliurament memòria i presentaci	1 día?	lun 07/01/13	lun 07/01/13



Proyecto: Planning
Fecha: jue 03/01/13

Tarea · Progreso · Resumen · Tareas externas · Fecha límite
División · Hito · Resumen del proyecto · Hito externo

Figura 1 – Diagrama de Gantt

3.5. Riscos i pla de contingència

En aquest apartat es presenta l'estudi de riscos i el pla de contingències que a l'inici del projecte es van considerar rellevants pel seu bon desenvolupament.

A la taula següent es llisten cada un dels riscos identificats, amb la seva corresponent mesura de mitigació.

Risc	Mitigació
Estudi d'altres assignatures durant el mateix semestre – realització de PACs i estudi per l'examen	Cal fer una bona planificació de l'entrega de les PACs 2 i 3 de manera que no coincideixin (o estiguin suficientment espaiades) amb les entregues de les PACs de l'altra assignatura que es cursa a la vegada. Com que l'examen de l'altra l'assignatura és el dia 19 de gener, es pensa que no hauria de ser un problema per les entregues finals del projecte ja que hi ha uns dies de marge entre aquestes i la data d'examen.
Imprevistos laborals com viatges de feina, hores extres, etc	S'haurien de poder recuperar aquestes hores de feina posant més hores de dedicació no previstes, o bé si són moltes, negociant amb l'empresa poder agafar algun dia de festa per poder recuperar-les.
S'espatlla l'ordinador amb el què es treballa habitualment	Cal tenir més d'un ordinador disponible amb tot el software necessari instal·lat i a la vegada tenir còpies de seguretat de tota la feina que es fa a diari.
Compromisos personals: - Viatge del 12 al 14 d'Octubre - Visites de familiars del 28 al 30 de Setembre - Festes de Nadal	Establir el Pla de Treball de manera que aquest no es vegi afectat per tots els compromisos que ja es coneixen a priori.
Malaltia	Cal tenir suficient contingència en el Pla de Treball per si en cas de malaltia es perden alguns dies o hores.
El projecte és tècnicament molt més difícil del previst, i per tant porta moltes més hores de feina de les planificades	Com que no es podrà fer una replanificació temporal, o si més no, modificar-ne les fites, la única solució serà dedicar-hi més hores.

Taula 4 – Riscos i mitigacions identificats a l'inici del projecte

3.6. Avaluació del material

A l'inici del projecte es va fer una avaluació de les eines informàtiques que es pensava que es necessitarien pel desenvolupament del projecte.

Es va creure que les següents eines informàtiques són les que, com a mínim, s'utilitzarien:

- Microsoft Project – per a fer la planificació temporal i el correcte seguiment del projecte
- Microsoft Word – per a l'elaboració de la memòria i les diferents entregues

- Microsoft Excel – per realitzar una anàlisi dels costos del projecte
- TINA-TI de Texas Instruments i Schematics d'Orcad – per a les simulacions de circuits analògics
- MPLAB de Microchip – per a les simulacions de circuits digitals
- Eagle de CadSoft – pel disseny del layout de les PCBs

4. Estudi del protocol ModBus RTU i la interfície RS-485

En aquest apartat s'expliquen, de manera teòrica, el protocol ModBus RTU i la interfície RS-485, tenint en compte les funcions i opcions que són rellevants per aquest projecte.

4.1. Protocol ModBus RTU

El protocol ModBus és un protocol de comunicació màster/esclau en sèrie molt utilitzat en el món industrial, sobretot per a la programació de PLCs (*programmable logic controller*). Permet la comunicació entre molts dispositius connectats a la mateixa xarxa. Molts cops també s'utilitza per a connectar un ordinador supervisor amb una unitat RTU (*remote terminal unit*) per a control de supervisió, i sistemes d'adquisició de dades.

La majoria d'equips ModBus es comuniquen en sèrie per la capa física, amb RS-485. El protocol és de la capa 7 OSI, la capa d'aplicació.

Una de les variants més utilitzades de ModBus, és el ModBus RTU, que s'utilitza en comunicacions sèrie. Utilitza una representació de dades per protocol de comunicació compacta i binària. Segueix les dades i comandes amb un *checksum* de redundància cíclica com a mecanisme de control d'errors per assegurar la fiabilitat de les dades. Els missatges ModBus RTU són transmesos de manera contínua, i estan separats entre ells per períodes silenciosos (*idle*). A partir d'aquí, s'explica només aquesta variant del protocol.

Cada equip utilitzat en la comunicació utilitza una adreça única. A més, només el node assignat com a màster pot començar una comanda, que contindrà l'adreça ModBus de l'equip on anirà destinada. Només actua l'equip on va destinada la comanda, tot i que els altres la poden rebre; amb l'excepció de les comandes *broadcast*, enviades al node 0. Les comandes sempre tenen informació per a comprovar que aquesta ha arribat correctament.

El format del protocol ModBus RTU és com s'indica a la taula següent:

Nom	Llargada	Funció
Inici	3.5c *idle*	A l'inici sempre hi ha com a mínim 3.5 caràcters silenciosos
Adreça	8 bits	Adreça de l'equip de destí. Les adreces vàlides van de 1 a 247
Funció	8 bits	Indica la funció que el destinatari ha de fer, el seu codi va de 1 a 255
Dades	n*8 bits	Porta informació addicional a la funció si és necessari, i conté els codis d'excepció en cas d'errors
CRC	16 bits	Conté el valor calculat pel mètode CRC (*cyclic redundancy checking*)
Final	3.5c *idle*	Al final sempre hi ha com a mínim 3.5 caràcter silenciosos entre missatge i missatge

Taula 5 – Format del protocol ModBus RTU

El protocol ModBus defineix una unitat de dades de protocol simple, PDU (protocol data unit), independent de les capes de comunicació inferiors. El mapatge del protocol ModBus en busos específics o xarxes, pot introduir alguns camps addicionals a la uni-

tat de dades d'aplicació, ADU (*application data unit*) tal i com es pot veure en la figura següent:

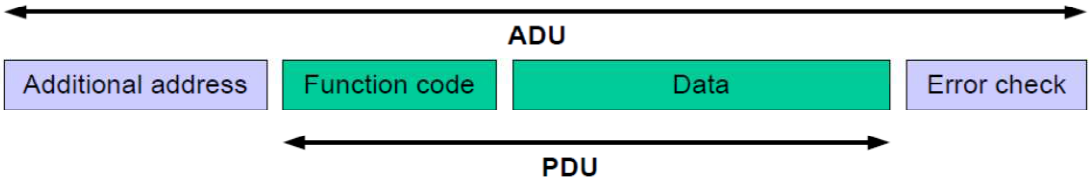


Figura 2 – Diagrama de dades ModBus

També cal afegir que la mida del PDU està limitada a 256 bytes, incloent l'adreça i el CRC (*cyclic redundancy check*).

Les funcions principals de ModBus es poden veure a la taula següent.

			Funció	**Codi**
Accés de dades	Accés a bits	Entrades discretes físi-ques	*Read Discrete Input*	02
		Bits interns	*Read Coils*	01
			Write Single Coil	05
			Write Multiple Coils	15
	Accés de 16 bits	Registres d'entrada física	*Read Input Register*	04
		Registres interns	*Read Holding Registers*	03
			Write Single Register	06
			Write Multiple Register	16
			Read/Write Multiple Regis-ter	23
			Mask Write Register	22
			Read FIFO Queue	24
	File Record Access		*Read File Record*	20
			Write File Record	21
	Diagnòstic		*Read Exception Status*	07
			Diagnostic	08
			Get Com Event Counter	11
			Get Com Event Log	12
			Report Slave ID	17
			Read Device Identification	43
	Altres		*Encapsulated Interface Transport*	43

Taula 6 – Funcions de ModBus

En aquest projecte no s'utilitzaran les funcions d'accés d'un sol bit, ja que es llegiran registres de dades de més d'un bit. S'expliquen les funcions més interessants pel desenvolupament d'aquest projecte a l'apartat 4.1.1.

Les funcions estan dividides entre funcions públiques, funcions definides per l'usuari i funcions reservades. Les funcions públiques tenen assignades els codis de l'1 al 64, del 73 al 99 i del 111 al 127. Aquestes funcions públiques estan ben definides i validades, són úniques i estan també ben documentades. Per les funcions definides per l'usuari, no es garanteix que el seu codi sigui únic. I les funcions reservades no es poden utilitzar públicament. Per altra banda, aquestes funcions també estan dividides entre funcions amb dades d'un sol bit o dades de 16 bits, i cadascuna d'elles pot ser d'entrada, només de lectura, o bé de lectura i escriptura.

La transmissió de dades es fa byte a byte, començant pel byte més significatiu.

Es presenta a continuació un diagrama d'estat que representa les transaccions ModBus des del receptor:

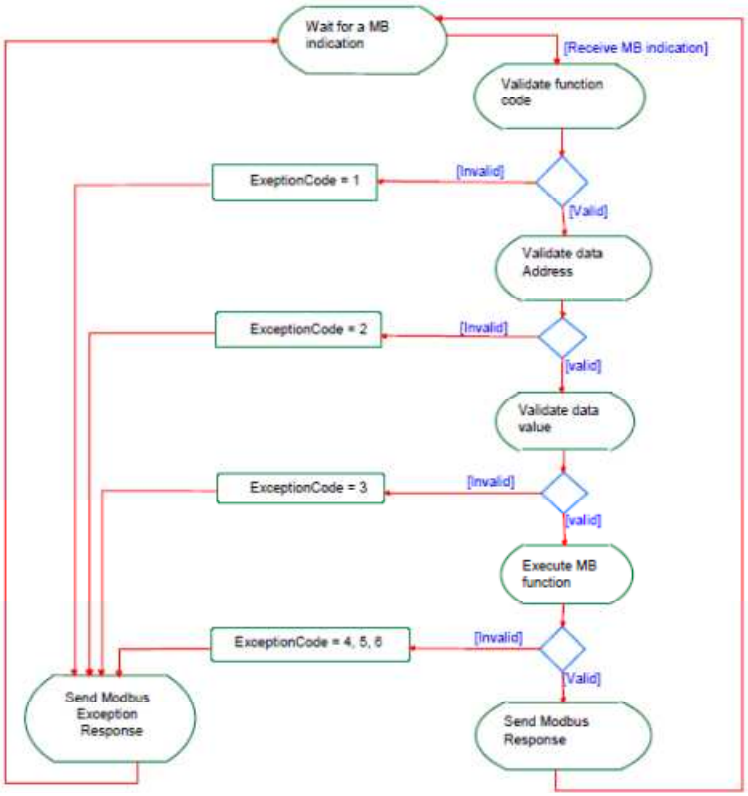


Figura 3 – Diagrama de transaccions d'estat ModBus

Com es pot veure en la figura superior, quan s'ha processat la petició pel servidor, es crea la resposta ModBus. Depenent del resultat del processament, la resposta serà positiva, i contindrà el codi de la funció de petició; o bé d'excepció, i contindrà el codi d'excepció. Si és d'excepció, com veiem a la figura superior, podrà ser degut a què troba un codi de la funció no vàlid, que donarà un codi d'excepció 1; o bé degut a l'adreça de dades, que retornaria un codi d'excepció 2; o per valors de dades incorrectes, amb un codi d'excepció 3; o per un error en l'execució de la funció ModBus, amb els codis d'excepció 4, 5 o 6.

ModBus defineix tres tipus de PDU, segons si són de pregunta, resposta o excepció: *MODBUS Request PDU*, *MODBUS Response PDU* i *MODBUS Exception Response* PDU. Es processa una resposta d'excepció quan el servidor no rep les dades per un error en la comunicació i no retorna cap resposta, i al cap d'un temps es processa un *timeout*, o bé quan el servidor rep la resposta però hi ha un error en la comunicació i tampoc respon i es processa un *timeout*, o bé quan el servidor rep les dades però no les pot processar. En tots tres casos, s'emet una resposta d'excepció que conté el codi de la funció original i el codi d'excepció. Aquests codis d'excepció es poden consultar a les referències donades a la bibliografia.

També cal esmentar les seves majors limitacions, que es llisten a continuació:

- Els tipus de dades que permet està limitat i per tant no suporta els objectes binaris gaire grans
- Pels nodes no existeix una manera estàndard de descriure un objecte de dades.
- Al ser un protocol màster/esclau, un equip no pot transmetre dades per excepció, sinó que el màster ha d'anar llegint de cada equip consecutivament, cosa que fa que requereixi molta amplada de banda i consumeixi molt temps de xarxa
- No té cap sistema de seguretat per a comandes no autoritzades o per intercepció de dades.

4.1.1. Funcions ModBus més rellevants

Les funcions que seran de més interès pel desenvolupament d'aquest projecte seran les següents:

- 03 – *Read Holding Registers*: s'utilitza per a la lectura dels continguts de blocs consecutius de registres d'un equip remot. La PDU de pregunta especificarà l'adreça d'inici i el nombre de registres. En la resposta, les dades estan agrupades en registres de 2 bytes.
- 04 – *Read Input Registers*: s'utilitza per llegir de 1 a 125 blocs consecutius de registres d'un equip remot. Les PDU de pregunta i resposta són com la funció 03.
- 06 – *Write Single Register*: s'utilitza per escriure un únic registre a un dispositiu remot. La PDU de pregunta especifica l'adreça del registre on es vol escriure. La resposta normal és un eco de la pregunta.
- 16 – *Write Multiple Registers*: s'utilitza per escriure un bloc de registres consecutius a un equip remot. Els registres tenen dos bytes cadascun. La resposta retorna el codi de la funció, l'adreça d'inici i la quantitat de registres que s'han escrit.

4.2. Interfície RS-485

RS-485 és un estàndard de comunicacions en bus de la capa física del model OSI que especifica la transmissió de dades bidireccional o *half duplex*. És molt utilitzat en protocols d'autòmats per control de sistemes industrials. Defineix les característiques elèctriques dels *drivers* i receptors en l'ús de sistemes digitals multipunt, però no estableix ni recomana cap protocol de comunicació. Les xarxes de comunicacions digitals que implementen RS-485 poden ser utilitzades per a llargues distàncies i en ambients sorollosos. La velocitat de transmissió de dades pot arribar a 35 Mbits/s per

distàncies de fins a 10 metres, i a 100 Kbits/s per distàncies de fins a 1200 m; cosa que el fa ideal per a transmissions a llargues distàncies.

Es transmet, normalment, a través d'un parell entrellaçat que admet fins a 32 estacions en un únic fil, suportant 32 transmissors i 32 receptors. En configuracions d'alta resistència pot suportar-ne fins a 256. La transmissió diferencial també permet diferents *drivers* en configuració multipunt. La instal·lació recomanada són nodes sèrie punt a punt (multipunt), és a dir, en línia o en bus, com es veu a la figura de sota.

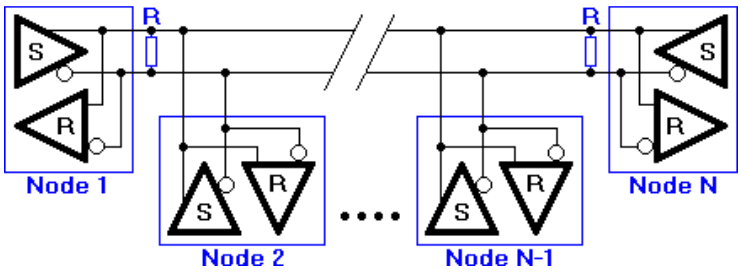


Figura 4 – Topologia de xarxa multipunt RS-485

Els equips connectats a RS-485 s'anomenen nodes, estacions o equips.

Tal com veiem a la figura anterior, al final del cable hi hauria d'haver una resistència per tal d'evitar reflexions i disminuir la sensibilitat del soroll elèctric. El seu valor hauria de ser igual a la impedància característica del cable, normalment d'uns 120 Ω per parells creuats. A la vegada, també s'hi posen resistències de polarització per tal de polaritzar els cables a un voltatge conegut de manera que els equips no el confonguin com a dades.

La configuració usual és la de màster esclau, on el màster és al centre dels cables, amb els esclaus als extrems del cable. Aquesta és la millor configuració ja que el màster és qui introdueix la polarització.

També és important dir que tot i que les dades es transmeten per un parell creuat, tots els transductors RS-485 interpreten el nivell de voltatge com la diferència respecte a un voltatge de referència, que normalment és el subministrament de potencial o el terra. Sense aquest voltatge de referència es podria interpretar malament la diferència de potencial.

La línia diferencial RS-485 té dos pins i un tercer de referència:

- B o - o TxD-/RxD- o pin inversor
- A o + o TxD+/RxD+ o pin no inversor
- SC o G o pin de referència

5. Circuits d'adaptació d'entrades i sortides

En aquest apartat s'expliquen els circuits que s'utilitzen per adaptar les entrades analògiques per tal que el microprocessador les pugui processar, i a la vegada els circuits que adapten les sortides analògiques del microprocessador per tal que es puguin entendre des de l'exterior.

L'adaptació de senyals analògics a digitals i de digitals a analògics no es fa en aquest apartat ja que el microprocessador que s'ha escollit ja té aquesta funcionalitat.

5.1. Estudi del sistema

El sistema objectiu d'aquest projecte ha de poder llegir les entrades analògiques provinents de l'exterior que es volen mesurar per tal que el microprocessador en pugui adquirir els seus valors amb un mínim de deu bits de precisió. El sistema té quatre entrades analògiques, dues de tensió, que prenen valors entre 0 V i 10 V; i dues de corrent, que prenen valors entre 4 mA i 20 mA.

A la vegada, també són necessaris els circuits d'adaptació de sortides, que modifiquen la sortida del microcontrolador per generar una sortida que es pugui llegir. El requeriment és tenir dues sortides analògiques que permetin generar tensions entre 0 V i 10 V i un corrent màxim de 200 mA; i un mínim de 10 bits de precisió.

Al buscar informació sobre microprocessadors, es pot veure que la majoria integren conversors de senyals analògics a digitals, i a la vegada, conversors de digital a analògic. Per tant, s'ha decidit procedir d'aquesta manera, i seleccionar, com veurem més endavant, un microprocessador que inclogui aquests conversors. Les consideracions a tenir en compte són que aquest haurà de tenir quatre canals d'entrada analògics i dos canals de sortida.

El què sí que es desenvoluparà en aquest apartat seran els circuits d'adaptació d'entrada de senyals analògics, de corrent i tensió, al voltatge d'entrada del microprocessador, que normalment serà de 0 V a V_{dd}, on V_{dd} és la tensió d'alimentació del microprocessador, que acostuma a ser de 3.3 V o de 5 V. Veurem més endavant que el microprocessador que nosaltres seleccionem té una font d'alimentació de 3.3 V. I també es dissenyen en aquest apartat els circuits d'adaptació de les sortides del microprocessador, que seran de 0 V a 3.3 V, a les tensions requerides a la sortida de 0 V a 10 V.

5.2. Disseny dels circuits electrònics

Per tal de fer l'adaptació de senyals analògics, ja siguin de diferents nivells de tensió o bé de corrent a tensió, necessitem amplificadors operacionals. Cal tenir en compte que hi ha multitud de components electrònics que fan les funcions desitjades, i en moltes ocasions escollir-ne un o altre dependria molt de l'aplicació del sistema, que en aquest cas no es coneix. Per exemple, si és una aplicació aeroespacial es buscarien components electrònics molt robusts i fiables i el cost no seria tant important, cosa que seria diferent si el sistema es dissenya per a d'altres aplicacions comercials on es buscaria components més moderns i potser amb funcions més avançades i molt econòmics, o bé si és pel sector mèdic, on també es buscaria un alt índex de fiabilitat, però els requeriments no serien tant alts com pel sector aeroespacial. En aquest treball s'intenten buscar components coneguts i fiables, i a la vegada també econòmics.

Pels adaptadors de les entrades analògiques s'ha escollit l'amplificador operacional LM324[1] de Texas Instruments, ja que és molt conegut, molt fiable i molt econòmic. Ara bé, per l'adaptador de tensió de les sortides, s'ha escollit l'amplificador operacional OPA551[2], ja que ha de poder tenir un alt corrent a la sortida, de 200 mA. N'hi ha molts d'altres que tenen un *slew rate* molt més elevat, però no es creu que sigui necessari per a la nostra aplicació.

També afegir, que s'ha considerat que totes les tensions d'alimentació dels amplificadors operacionals són de +/-15 V. Estrictament, només és un requeriment pel circuit d'adaptació de les sortides, com es pot veure del *datasheet*, però es creu que és millor escollir una tensió comuna per a tots ells ja que això simplifica el disseny de la font d'alimentació, i per tant també abarateix el cost.

Per tal d'assegurar que els circuits funcionen com es desitja, simulem aquests circuits dissenyats amb el programa TINA-TI de Texas Instruments. Totes aquestes simulacions s'han fet de manera que es cobreixi tot el rang de voltatge o corrent de l'entrada o sortida corresponent.

A continuació, es descriuen els diferents circuits d'adaptació i les seves simulacions.

5.2.1. Adaptació del voltatge d'entrada

Per a adaptar la tensió d'entrada s'ha de fer una divisió de la tensió de 0 V a 10 V, al rang d'entrada del microprocessador, que com ja s'ha dit és de 0 V a 3.3 V. Es presenta el disseny del circuit a continuació:

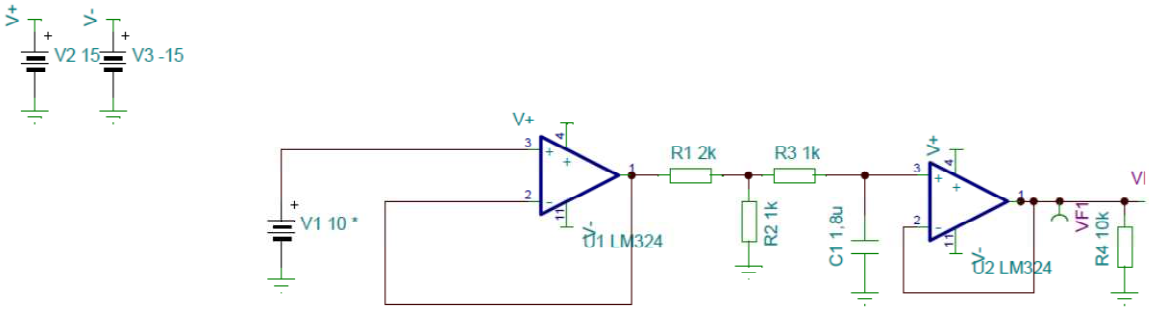


Figura 5 – Circuit d'adaptació de la tensió d'entrada

Les resistències R_1 i R_2 fan la divisió de tensió, sempre i quan hi hagi un seguidor de tensió que controli la impedància d'entrada cap al microprocessador; aquesta és la funció de l'amplificador operacional que hi ha a la part dreta del circuit. Sense aquest operacional, si tenim una impedància d'entrada petita al microprocessador, aquesta provocaria una caiguda de voltatge molt important, i per tant degradaria el senyal. També és interessant posar un altre seguidor de tensió abans de les dues resistències que divideixen la tensió, per tal d'independitzar la tensió d'entrada i el circuit del convertidor analògic a digital i no carregar l'entrada analògica. Els valors de les resistències, segons la llei d'Ohm, i tal i com es veu a continuació, hauran de complir la relació següent ja que el guany del circuit ha de ser de 1/3 aproximadament:

$$V_{out} = V_i \left(\frac{R_2}{R_1+R_2}\right)$$

On R_1 i R_2 són les dues resistències del divisor de tensió i V_i és la tensió d'entrada, anomenada V_1 al circuit de la figura superior.

[1] http://www.ti.com/lit/ds/symlink/lm324.pdf
[2] http://www.ti.com/lit/ds/symlink/opa551.pdf

A la vegada, cal dir que els valors d'aquestes resistències convindrà que estiguin en el rang dels kΩ, ja que valors més alts provocarien soroll tèrmic o també pèrdues de corrent, i valors més petits pot ser que no siguin adequats per l'aplicació que es dissenya. També cal tenir en compte que la resistència s'ha de poder trobar en el mercat. S'han escollit valors de resistències de 1 kΩ i 2 kΩ. Com que els valors de les resistències no són exactament els donats per l'equació superior, sempre tindrem un petit error, però en aquest cas els valors de les resistències utilitzades és un 98% del valor requerit, o sigui que pensem que l'error és menyspreable.

I per últim, s'hi ha afegit un filtre passabaixos per poder filtrar el soroll d'alta freqüència que pugui generar el microprocessador. Com podem veure en el circuit, hem col·locat aquest filtre entre el divisor de tensió i el segon seguidor de tensió. S'ha escollit un filtre passiu RC, és a dir, format per una resistència i un condensador. Tenint en compte que les freqüències d'aquest circuit no seran molt altes, escollim un freqüència de tall del filtre entre 50 Hz i 100 Hz. La relació entre els valors de la resistència i el condensador amb la freqüència de tall és:

$$f_c = \frac{1}{2\pi RC}$$

Així doncs, escollim una resistència de 1 kΩ i un condensador de 1.8 μF. Per tant, la freqüència de tall del filtre és de 88.4 Hz.

Per fer la simulació del circuit, fem un escombrat de la tensió d'entrada, que com ja s'ha vist, va de 0 V a 10 V. Es pot veure al gràfic de la Figura 6 que efectivament, a l'eix vertical, la tensió de sortida del circuit varia entre 0 V i 3.3 V pel rang de tensió d'entrada donat, representada a l'eix horitzontal.

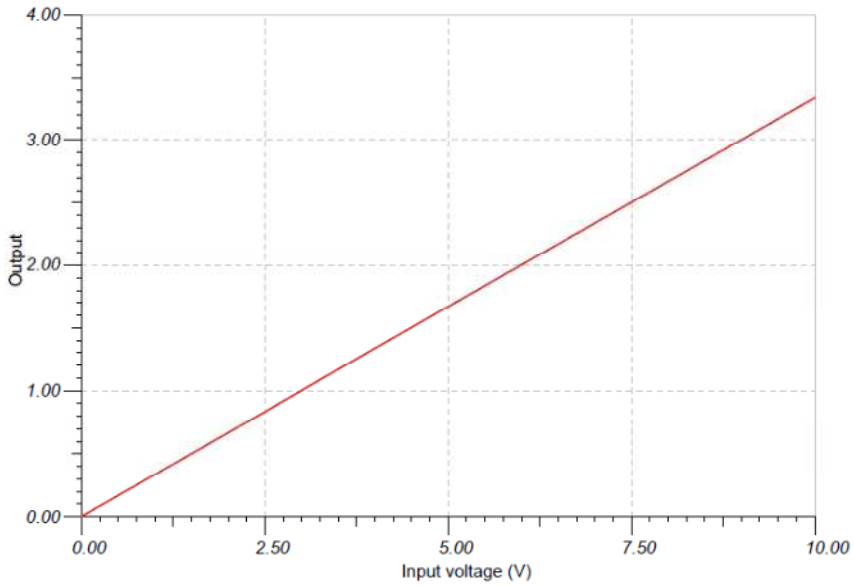


Figura 6 – Simulació de l'adaptació de la tensió d'entrada

5.2.2. Adaptació de corrent a voltatge d'entrada

En aquest cas, s'ha de fer una adaptació de corrent de 4 mA a 20 mA, al rang d'entrada del microprocessador, de 0 V a 3.3 V. Es presenta el disseny del circuit a continuació:

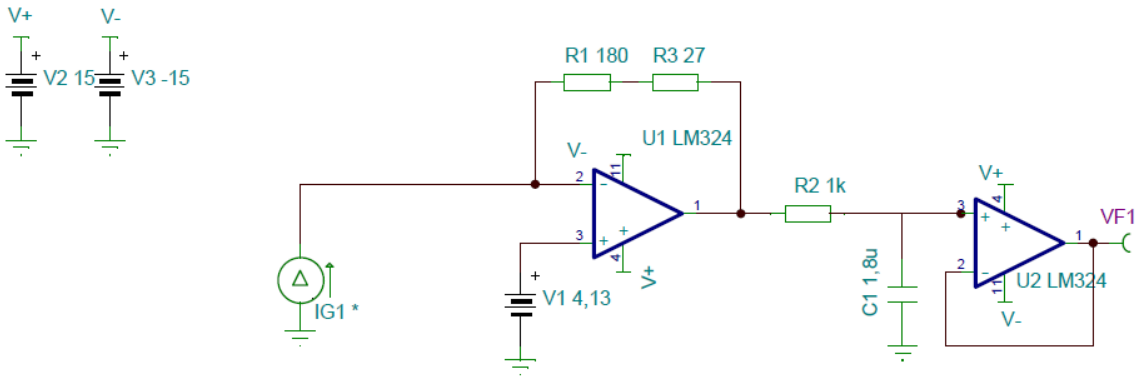


Figura 7 – Circuit d'adaptació de corrent a tensió d'entrada

En aquest cas, i assumint que l'amplificador és ideal, tindrem que segons les lleis d'Ohm, la tensió de sortida serà segons com segueix:

$$V_{out} = V_1 - IR$$

On, V_1 és el voltatge de la font de tensió, I és el corrent de la font de corrent, i R és el valor de la resistència de l'operacional que fa l'adaptació de corrent a tensió. Per aconseguir el voltatge de sortida desitjat, obtenim de l'equació superior que R = 206.25 Ω i que V_1 = 4.125 V. Per tal d'aconseguir el valor de la resistència més proper possible, decidim posar dues resistències en sèrie, R_1 i R_3, on la resistència total de les dues és de 207 Ω, ja que cal tenir en compte que el valor de la resistència ha d'estar disponible en el mercat..

També s'hi ha afegit un filtre passabaixos, exactament igual al que es descriu a l'apartat anterior, i també un seguidor de tensió, amb la mateixa funció que el que s'ha utilitzat per l'adaptador de la tensió d'entrada.

Analitzant millor aquest voltatge de 4.125 V i el corrent que demana, veiem que a l'entrada de l'amplificador operacional es requereix un corrent d'uns 52 nA, com podem veure a la figura de sota.

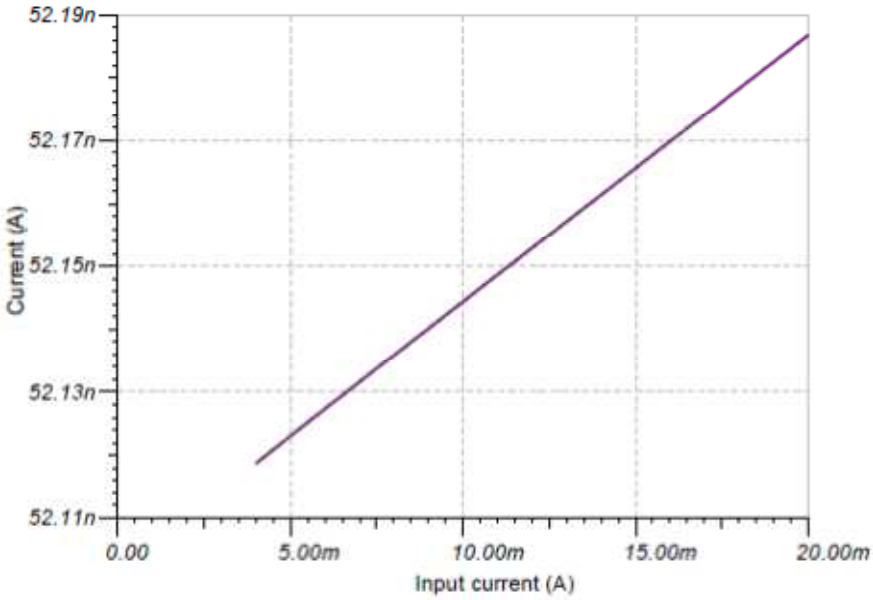


Figura 8 – Corrent a la font de tensió de 4.125 V

S'ha pensat que per unificar els voltatges, tenint en compte que aquest corrent és tant petit, la millor solució seria utilitzar un simple divisor de tensió amb resistències, per tal de què la tensió d'entrada sigui de 15 V. Així doncs, modifiquem el circuit presentat a la Figura 7 de manera que el voltatge d'entrada d'aquesta font sigui de 15 V. Aquesta tensió es dividirà utilitzant dues resistències, de la mateixa manera que s'ha fet a l'apartat 5.2.1 per l'adaptació del voltatge d'entrada. En aquest cas, escollim els valors de les resistències de 680 Ω i 1.8 kΩ. Cal dir que com que s'han hagut d'escollir valors de les resistències disponibles en el mercat, sempre tindrem un petit error associat, però en aquest cas es calcula que és només d'un 0.3%, ja que el valor teòric de la segona resistència era de 1.79 kΩ. Presentem aquest circuit final a la següent figura.

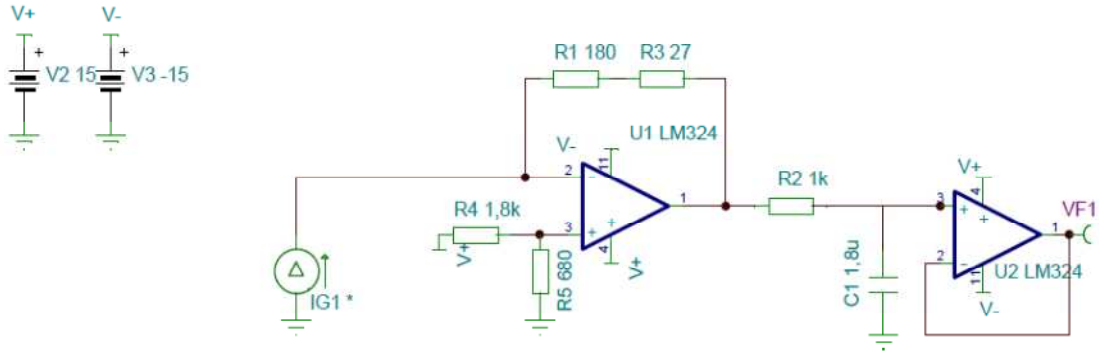


Figura 9 – Circuit d'adaptació de corrent a tensió d'entrada modificat

Per aquest circuit hem de comprovar que l'entrada de 4 mA a 20 mA es tradueix en una sortida del circuit com l'anterior, és a dir, una tensió entre 0 V i 3.3 V. Fixem-nos, com veiem en la simulació de la Figura 10, que en aquest cas tenim que quan el corrent d'entrada és 4 mA, la sortida és de 3.3 V, i per un corrent d'entrada de 20 mA, tenim una sortida de 0 V.

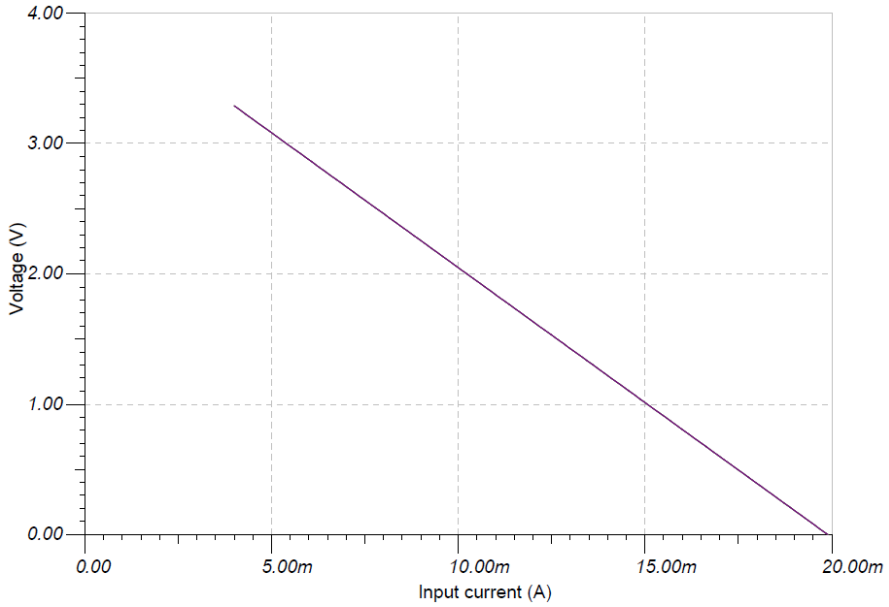


Figura 10 – Simulació de l'adaptació del corrent a tensió d'entrada

5.2.3. Adaptació del voltatge de sortida

Per últim, cal un circuit d'adaptació del voltatge de sortida del microprocessador. S'ha de fer una amplificació de la tensió de sortida de 0 V a 3.3 V del microprocessador a una escala de 0 V a 10 V, i tenint en compte que el corrent pot arribar a 200 mA. Es presenta el disseny del circuit a continuació:

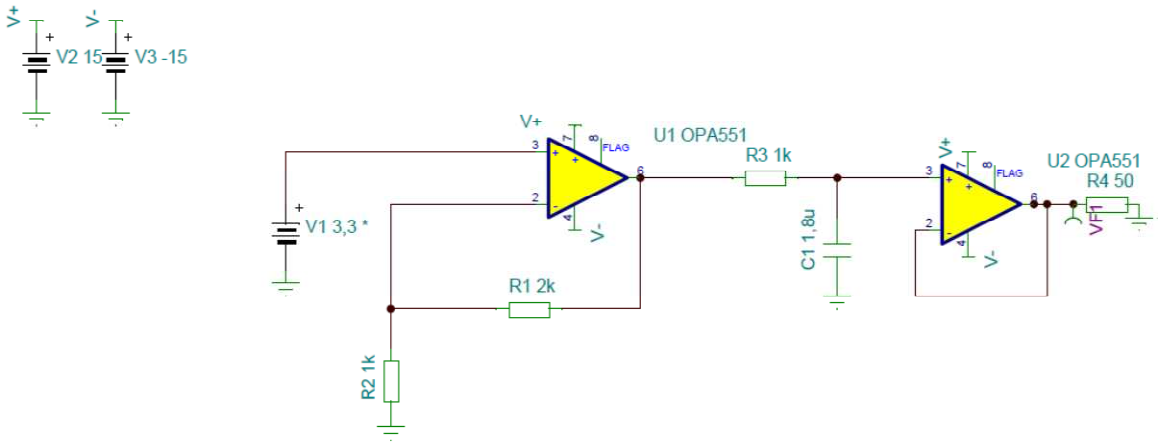


Figura 11 – Circuit adaptació tensió sortida

En aquest cas, el guany ha de ser de tres aproximadament, ja que es vol amplificar el senyal de 3.3 V a 10 V. Segons les lleis d'Ohm, la relació entre les tensions i les resistències del circuit és:

$$V_{out} = V_1 \left(\frac{R_1+R_2}{R_2}\right)$$

Per tant els valors d'aquestes resistències hauran de ser també iguals que als del circuit d'adaptació de tensió d'entrada, i els tornem a seleccionar altra vegada de 1 kΩ i 2 kΩ.

També s'hi afegeix un filtre passabaixos i un seguidor de tensió exactament iguals que en els dos adaptadors anteriors.

Finalment, es col·loca una resistència de 50 Ω a la sortida per comprovar que l'amplificador operacional pot suportar un corrent de 200 mA tal i com es demana.

I per últim, es fa la simulació del circuit, on tenim que l'entrada al circuit serà d'entre 0 V i 3.3 V, i la sortida d'aquest circuit d'adaptació és una tensió de 0 V a 10 V. Presentem el voltatge a la sortida en el gràfic següent.

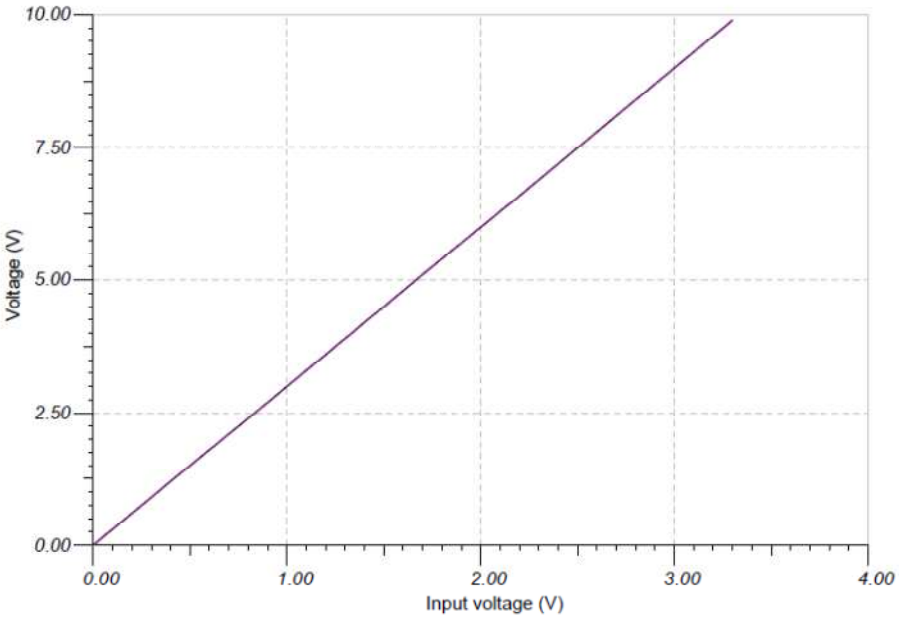


Figura 12 – Simulació de l'adaptació de la tensió de sortida

6. Disseny de la font d'alimentació

En aquest apartat es presenta el disseny de la font d'alimentació tenint en compte les tensions de cada un dels blocs del circuit i l'eficiència de la font. Es dissenya la font el més eficient possible per tal de reduir el consum d'energia de l'equip.

6.1. Anàlisi de les tensions i corrents de la font

En aquest apartat es fa un anàlisi dels consums necessaris d'entrada i sortida de la font segons les tensions requerides per cada un dels blocs de l'expansor. La condició que tenim és que la font s'ha de poder alimentar amb qualsevol tensió contínua d'entre 12 V i 48 V.

Els circuits que aquesta font d'alimentació haurà d'alimentar són els diferents amplificadors operacionals dels circuits d'adaptació d'entrades i sortides, a +/- 15 V, i el microprocessador i el transductor RS-485 a 3.3 V.

A la vegada, també cal considerar quina és la intensitat que consumeix cada circuit per tal de poder optimitzar el disseny de la font d'alimentació. Per això, utilitzant TINA-TI, s'han analitzat les intensitats pel rang de valors de les tensions d'entrada per a cada un dels circuits d'adaptació analògics. Convé remarcar que aquests corrents són els totals de cada una de les fonts del circuit d'adaptació, considerant ja tots els amplificadors operacionals i altres fonts que hi pugui haver (pel cas de l'adaptació de corrent a tensió). En presentem les gràfiques a continuació.

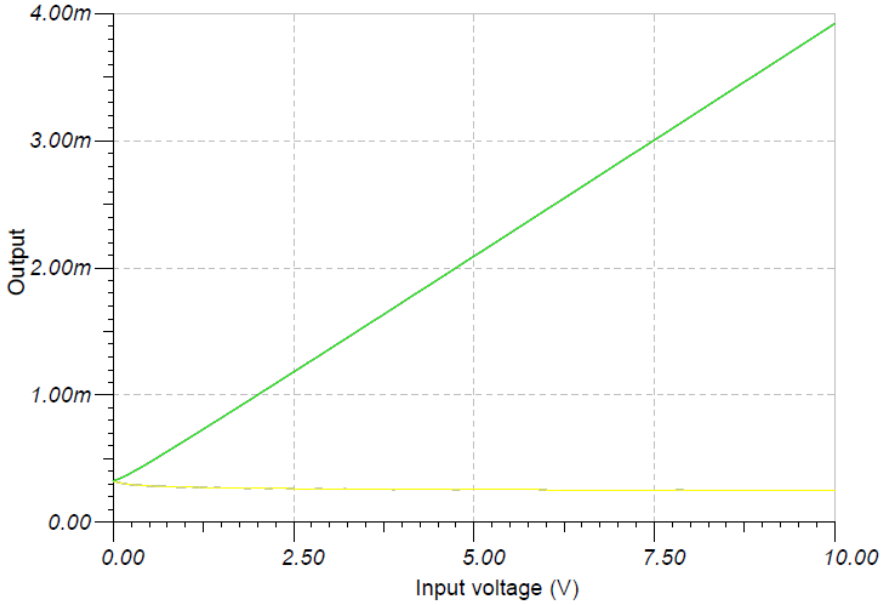


Figura 13 – Corrent a les fonts de 15 V (línia verda) i -15 V (línia groga) del circuit d'adaptació de voltatge d'entrada

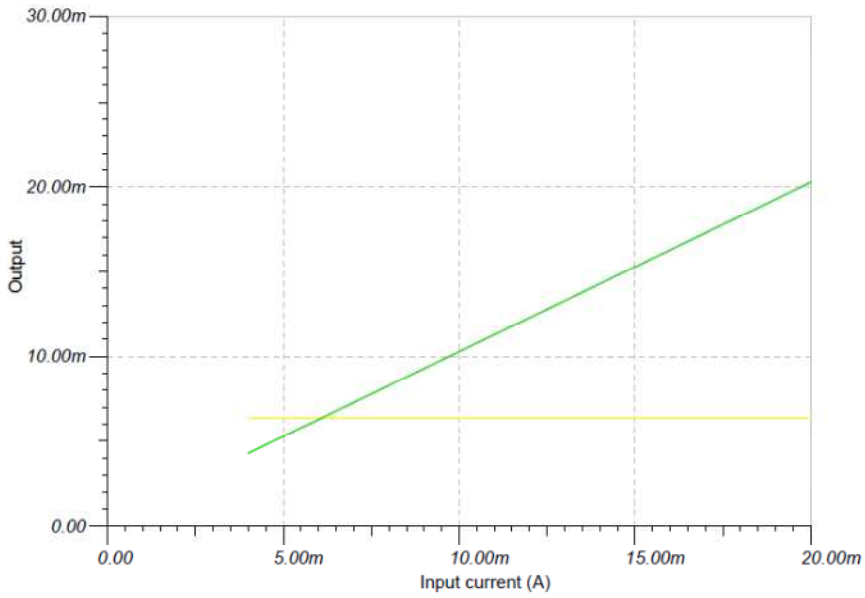


Figura 14 – Corrent a les fonts de 15 V (línia groga) i -15 V (línia verda) del circuit d'adaptació de corrent d'entrada

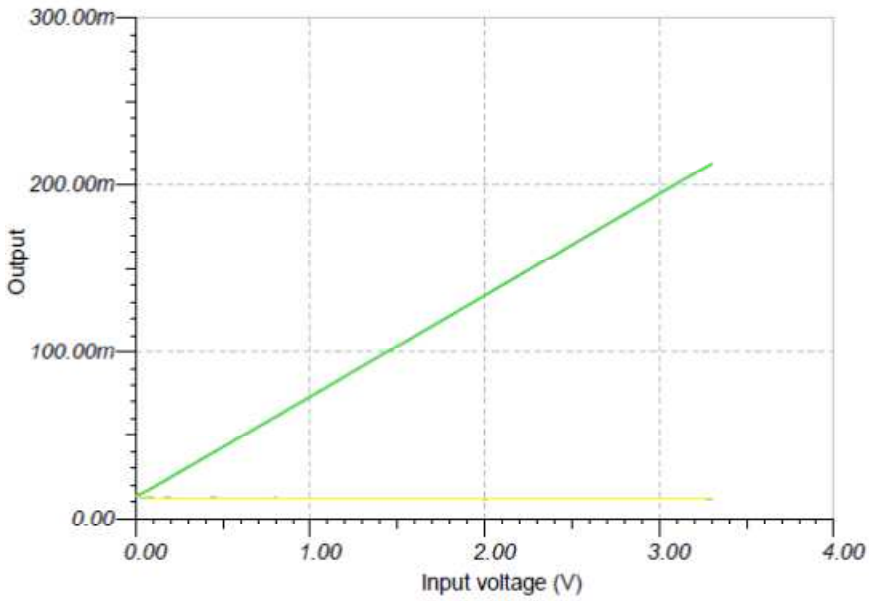


Figura 15 – Corrent a les fonts de 15 V (línia verda) i -15 V (línia groga) del circuit d'adaptació de voltatge de sortida amb la màxima corrent de sortida

També cal saber la corrent a l'entrada del microprocessador. Del seu *datasheet*[3] es pot veure que per a una velocitat de 10 MIPS, adequada pel nostre disseny, i a temperatura ambient, el màxim corrent és de 22 mA. I falta afegir el corrent màxim a l'entrada del transductor RS-485 escollit[4], que és de 2.2 mA. Per tant, el corrent a l'entrada del conversor de 15 V a 3.3 V haurà de ser de 24.2 mA com mínim.

[3] http://ww1.microchip.com/downloads/en/DeviceDoc/70292G.pdf
[4] http://datasheets.maximintegrated.com/en/ds/MAX3483E-MAX3491E.pdf

6.2. Disseny dels circuits electrònics

El disseny d'aquesta font d'alimentació consisteix en diferents parts que expliquem en els propers paràgrafs. Aquests dissenys i les corresponents simulacions s'han fet amb l'ajuda de l'eina 'WEBENCH Designer' de Texas Instruments. També presentem les gràfiques d'eficiència per a cada un dels circuits d'alimentació proposats.

Primer de tot, anem a explicar com es fa el disseny de la font d'alimentació en conjunt. Tenim una entrada de tensió contínua que pot prendre qualsevol valor entre 12 V i 48 V. Amb un regulador DCDC la convertim a 15 V. Aquests 15 V els utilitzarem per alimentar els voltatges de 15 V dels amplificadors operacionals, i també els DCDC *converters* que utilitzarem per passar aquesta tensió a -15 V per a l'alimentació negativa dels amplificadors operacionals, i també a 3.3 V per a l'alimentació del microprocessador i el transductor RS-485.

Comencem amb el disseny d'aquestes dues últimes fonts d'alimentació, ja que aquests ens dictaran el corrent requerit a la sortida del convertidor DCDC de 15 V. El primer d'ells consisteix en utilitzar els 15 V que s'han obtingut de la primera etapa, amb un convertidor del tipus *step-down*, per passar al voltatge de 3.3 V per al microprocessador i el transductor. S'han escollit convertidors del tipus *step-down* i no lineals perquè com que la diferència de tensió d'entrada i de sortida és força elevada, la potència que un convertidor lineal dissipa (P = I x (Vi – Vo)), on Vi és la tensió d'entrada i Vo la tensió de sortida, pot ser força alta. I si la corrent fos elevada caldria posar un dissipador.

Com ja s'ha comentat a l'apartat anterior, el màxim corrent a l'entrada del conversor de 15 V a 3.3 V ha de ser de 24.2 mA, cosa que s'ha de considerar pel disseny d'aquest conversor. Normalment es deixa un marge d'un 20% a un 30% al màxim corrent que es necessita a la sortida. En aquest cas, com que el corrent és molt petit, el marge pot ser fins i tot més gran. Escollim doncs un corrent a la sortida del conversor de 50 mA. S'ha escollit el xip TPS62125[5] ja que ofereix bona eficiència. Presentem aquest circuit en la figura següent:

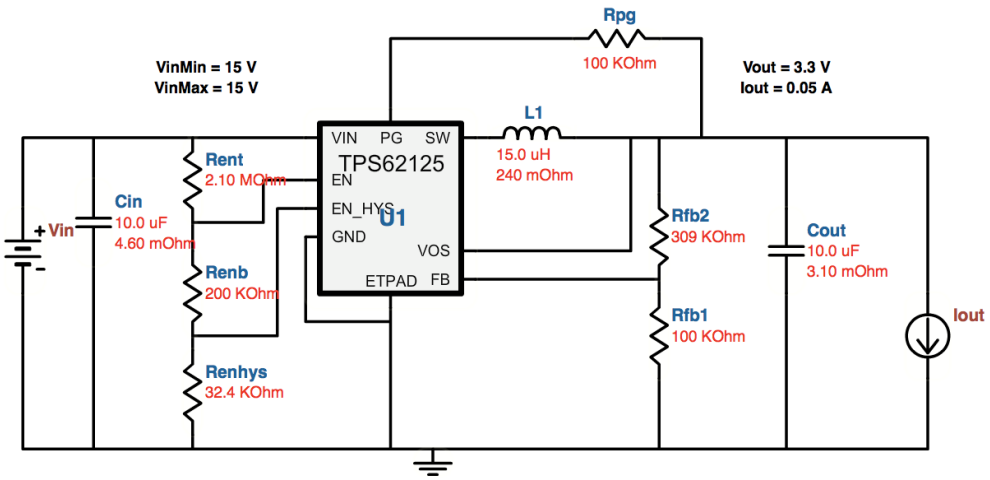


Figura 16 – Circuit d'adaptació DCDC de 15 V a 3.3 V

Podem veure a la Figura 17 que l'eficiència del xip seleccionat a aquesta etapa és molta alta, especialment per corrents elevats. L'eficiència mitjana del circuit és de

[5] http://www.ti.com/lit/ds/symlink/tps62125.pdf

86.735%. D'aquí en calcularem la intensitat d'entrada, per tal de saber la que aquest circuit demana al conversor de la primera etapa. Sabent que:

$$eficiència = \frac{P_{sortida}}{P_{entrada}} = \frac{V_S \cdot I_S}{V_e \cdot I_e}$$

Així doncs, aïllant-ne el corrent d'entrada, obtenim que aquest ha de ser de 12.68 mA.

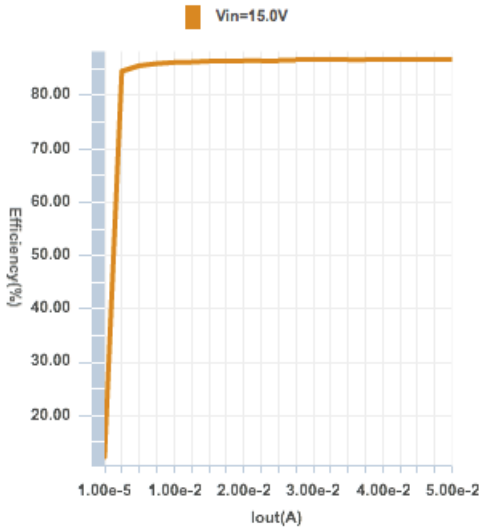


Figura 17 – Eficiència dels circuits DCDC de 15 V a 3.3 V

El segon pas, és obtenir el voltatge de -15 V. Després d'investigar diferents opcions, s'ha arribat a la conclusió que l'opció més fàcil i econòmica d'implementar és passar dels 15 V que ens dóna la sortida de la primera etapa als -15 V que necessitem pels amplificadors operacionals mitjançant un xip inversor de polaritat. En aquest cas, per saber el màxim corrent que es necessita a la sortida, utilitzarem les gràfiques que s'han presentat en l'apartat anterior. Com que els diferents dispositius estaran en paral·lel, cal sumar els corrents màxims que cadascuna de les fonts dels circuits d'adaptació ens demanen, tenint en compte que tenim dos circuits de cada tipus. Així doncs, utilitzant els cursors de l'eina TINA-TI per saber-ne els valors exactes, podem deduir que la intensitat a la font de -15 V ha de ser:

$$I = 2x(I_{adaptació\ corrent\ entrada} + I_{adaptació\ voltatge\ entrada} + I_{adaptació\ voltatge\ sortida}) = 2x(20.29\ mA + 323.31\ \mu A + 13.64\ mA) = 68.5\ mA$$

Altre cop, deixem un marge d'un 20% com a mínim i per tant escollim un corrent de 85 mA.

En aquest cas utilitzarem el circuit *step-down* TPS84259[6], que sembla ser que és un dels que té un millor compromís entre cost, eficiència i tamany del circuit. Cal dir que el tamany del circuit també serà molt important en moltes aplicacions, on la mida del dispositiu pugui ser decisiu pel disseny.

L'eficiència mitjana d'aquest circuit és d'un 84.87%. Com s'ha fet a l'apartat anterior, utilitzant aquesta eficiència i els voltatges d'entrada i sortida, i el corrent de sortida que necessitem, calculem que el corrent que aquest requereix a l'entrada és de 100.15 mA.

Presentem aquest circuit en la figura següent.

[6] http://www.ti.com/lit/ds/symlink/tps84259.pdf

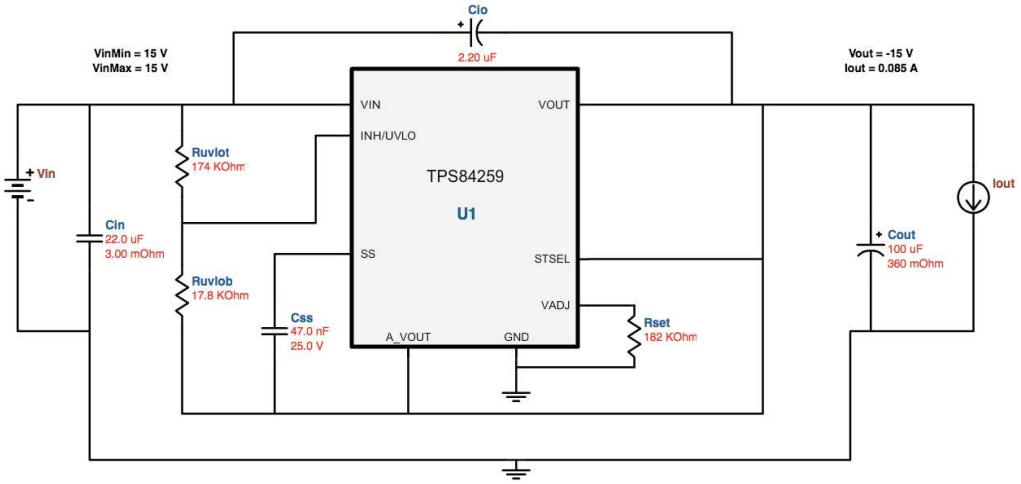


Figura 18 – Circuit d'adaptació DCDC de 15 V a -15 V[7]

Es pot veure a continuació la gràfica d'eficiència per a aquest circuit, on passa el mateix pel circuit d'alimentació anterior, l'eficiència també és superior quan la intensitat de sortida és més gran.

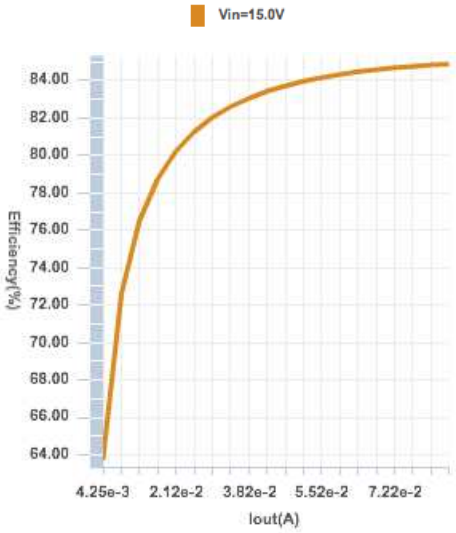


Figura 19 – Eficiència del circuit DCDC de 15 V a -15 V

I per últim, ja podem passar a dissenyar el circuit conversor de l'entrada que pot prendre qualsevol valor entre 12 V i 48 V donant una sortida de 15 V. En aquest cas, com que la tensió de sortida és més alta que el mínim voltatge d'entrada, necessitem que el regulador estigui en la configuració que es coneix com a SEPIC (*Single-Ended Primary-Inductor Converter*). A la vegada, també cal considerar el màxim corrent necessari a la font. Haurem de sumar tots els corrents que es demanen a la sortida d'aquest conversor, tenint en compte que a la seva sortida hi tindrem connectats els sis circuits d'adaptació d'entrades i/o sortides i el conversor DCDC a -15 V i a 3.3 V. Per tant, la intensitat que necessitarem a la sortida d'aquest circuit serà:

[7] Cal indicar que el condensador C_{out} hauria d'estar orientat al revés, però no es pot modificar, ja que és un error del simulador Webench.

$$
\begin{aligned}
I = \; & 2x(I_{adaptació\ corrent\ entrada} + I_{adaptació\ voltatge\ entrada} + I_{adaptació\ voltatge\ sortida}) \\
& + I_{conversor\ a-15V} + I_{conversor\ a\ 3.3V} \\
& = 2x(6.34\ mA + 3.92\ mA + 212.86\ mA) + 100.15\ mA + 12.68\ mA \\
& = 559.07\ mA
\end{aligned}
$$

També deixem un marge d'un 20% al consum de la font. Per tant, farem el disseny per un corrent de sortida de 0.67 A.

Cal dir que, altre cop, s'ha valorat que el circuit sigui el màxim eficient possible, tenint en compte que el preu no es dispari, intentant arribar a un compromís entre els dos paràmetres. Aquest disseny té una eficiència d'un 88.826% de mitjana.

Segons aquests criteris, s'ha escollit el regulador d'alta eficiència de Texas Instruments LM3481[8]. Es pot veure a la figura següent el circuit que hem dissenyat per passar d'una tensió contínua de 12 a 48 V a 15 V:

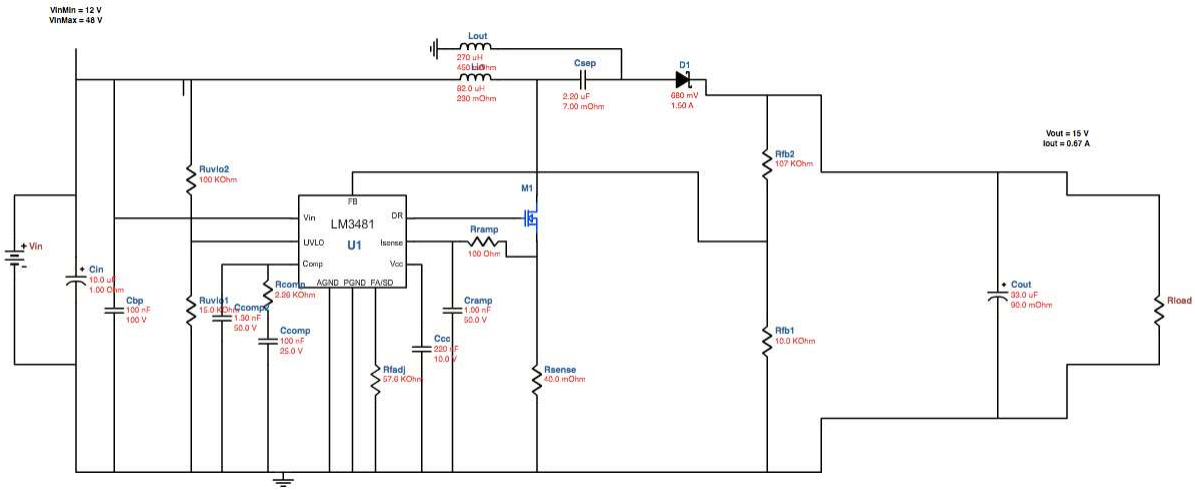


Figura 20 – Circuit d'adaptació DCDC de la font d'alimentació a 15 V

Presentem en la figura següent l'eficiència per aquesta font en funció de la intensitat de sortida.

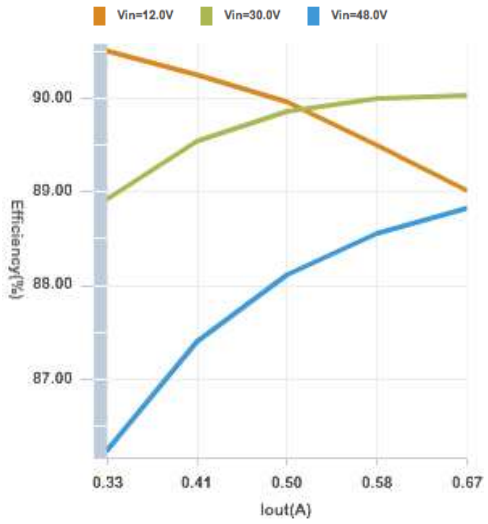


Figura 21 – Eficiència del circuit DCDC de la font d'alimentació a 15 V

[8] http://www.ti.com/lit/ds/symlink/lm3481.pdf

Veiem que per a aquesta primera etapa, l’eficiència augmenta a mesura que la corrent de sortida augmenta per a voltatges d’entrada de 30 V i 48 V; ara bé aquesta eficiència disminueix per voltatges d’entrada de 12 V. Per tant, idealment voldríem un alt corrent de sortida i un voltatge d’entrada el més proper possible a 30 V, que és quan tenim una major eficiència.

7. El mapa de memòria ModBus

En aquest apartat es detallen les especificacions ModBus per a l'equip. Aquestes especificacions cobreixen el mapa de memòria de l'equip, definint la informació que hi ha en cada posició, les seves unitats, l'ordre de bytes i les funcions per accedir-hi.

7.1. Anàlisi del sistema

Aquí es presenta un anàlisi inicial que s'ha fet del sistema per tal d'entendre quins són els requeriments del mapa de memòria ModBus.

Bàsicament, el màster de ModBus farà dos tipus de peticions al microprocessador. Una d'elles és passar-li un valor de tensió que vulgui fixar a una de les sortides analògiques, indicant-li quina és la sortida per la qual vol escriure. L'altra acció que s'ha de contemplar és quan el màster ModBus demani al microprocessador el valor d'una de les entrades analògiques, corrent o tensió; també indicant-li des de quina d'elles vol llegir.

Per tant, en la memòria ModBus necessitarem quatre registres per a entrades, i dos per a sortides. Aquests sis registres tots contenen valors digitals.

Com ja s'ha esmentat en apartats anteriors, aquest expansor d'entrades i sortides analògiques incorpora RS-485 com a interfície física de ModBus.

Aquest equip tindrà les següents funcions ModBus:

Funció	Codi (hex)	Descripció
Read Holding Registers	0x03	Llegeix el contingut de blocs d'informació de registres de retenció consecutius.
Read Input Registers	0x04	Llegeix de 1 a 125 registres d'entrada consecutius. Els canals d'entrades analògiques aniran del 1 al 4.
Write Single Register	0x06	Escriu un únic registre de retenció a un equip remot.
Diagnostics	0x08	Realitza diagnòstics ModBus entre el màster i l'esclau.
Write Multiple Registers	0x10	Escriu un bloc de registres consecutius (de 1 a 123) a un equip remot. Els canals de sortida seran 1 i 2.

Taula 7 – Funcions de ModBus de l'equip

La taula següent resumeix la correlació entre el nostre equip i l'espai de referència ModBus.

Equip	Referència ModBus	Adreces	Funcions suportades
Entrades analògiques	Registres d'entrada (3xxxx)	De 0000 a 270E	-0x04: llegir registres d'entrada 3x
Paràmetres i codis de control	Registres de retenció (4xxxx)	De 0000 a 270E	-0x03: llegir registres 4x -0x06: escriure un únic registre 4x

			-0x10: escriure registres 4x

Taula 8 – Espai de referència ModBus

Els registres de lectura d'entrada 3x contenen un nombre de 16 bits rebut d'una font externa, com pugui ser un senyal analògic. En canvi, els registres de lectura i escriptura de sortida, o registres de retenció, 4x, són utilitzats per emmagatzemar dades numèriques de 16 bits o per enviar-les des d'una CPU a un canal de sortida.

7.2. El mapa de memòria ModBus

Amb ModBus cada codi de funció demana accés a unes determinades referències ModBus. El primer dígit no s'inclou en el camp de direcció d'un missatge ModBus. Totes les adreces de dades en els missatges ModBus estan referenciades a zeros, amb les primeres dades adreçades com a zero. Per exemple, el registre 40001 s'adreçarà com a 0000 en el camp d'adreça d'un missatge. El codi de la funció ja especifica a quin grup de registres pertany. En aquest cas, el codi de funció ja indicaria que pertany al grup 4xxxx.

A continuació es descriu la correlació de cada sèrie de referències ModBus:

- **Correlació 3x – Entrades ModBus**: el convertidor correlaciona la informació següent amb les direccions ModBus 3x, els registres d'entrada ModBus, que és qualsevol entrada analògica. En aquest cas, tenim quatre entrades analògiques, dues de tensió i dues de corrent. La taula següent resumeix aquestes entrades:

Referència	Adreça	Nom	Comentaris
30001	0000	EA1	Comunica el nivell de l'entrada analògica que conté el voltatge de l'entrada 1
30002	0001	EA2	Comunica el nivell de l'entrada analògica que conté el voltatge de l'entrada 2
30003	0002	EA3	Comunica el nivell de l'entrada analògica que conté el corrent de l'entrada 1
30004	0003	EA4	Comunica el nivell de l'entrada analògica que conté el corrent de l'entrada 2

Taula 9 – Entrades ModBus

Aquest equip suporta els següents codis de funcions ModBus per aquests registres 3x:

Codi de funció	Descripció
0x04	Llegir el valor de l'entrada 3x

Taula 10 – Funcions ModBus dels registres 3x

- **Correlació 4x – Registres de retenció ModBus**: el convertidor correlaciona els seus paràmetres i altres dades amb els registres de retenció 4x com segueix:

Referència	Adreça	Nom	Comentaris
40001	0000	SA1	Comunica el nivell de sortida analògica que conté el voltatge de la sortida 1
40002	0001	SA2	Comunica el nivell de sortida analògica que conté el voltatge de la sortida 2
40003	0002	A	Permet modificar l'adreça del dispositiu de la que s'assigna per defecte
40004	0003	V	Permet modificar la velocitat de transmissió de la que s'assigna per defecte
40005	0004	P	Permet modificar la paritat de la que s'assigna per defecte

Taula 11 – Sortides ModBus

Aquest equip suporta els següents codis de funcions ModBus per aquests registres 4x:

Codi de funció	Descripció
0x03	Llegir els registres de retenció 4x
0x06	Escriu un únic registre 4x de retenció a un equip remot
0x10	Escriu un bloc de registres 4x consecutius a un equip remot

Taula 12 – Funcions ModBus dels registres 4x

A la vegada, també té els següents codis d'excepció per controlar diferents situacions en què es pugui produir algun problema ja sigui en la trama transmesa o la transmissió en sí:

Codi excepció	Nom	Significat
01	ILLEGAL FUNCTION	La comanda no és vàlida o autoritzada.
02	ILLEGAL DATA ADDRESS	La direcció de dades rebuda a la consulta no és permesa o el paràmetre no està definit.
03	ILLEGAL DATA VALUE	Algun valor del camp de dades de consulta no és permès per l'equip. Pot ser degut a: - No es troba dins els límits mínims o màxims - El paràmetre és només de lectura - El missatge és massa llarg
04	SLAVE DEVICE FAILURE	L'esclau ha tingut un error no recuperable quan feia l'operació demanada.
05	ACKNOWLE DGE	L'esclau ha acceptat la petició i la processa, però vol indicar que tardarà molt. Així s'evita un error de *timeout*.

06	SLAVE DEVICE BUSY	L'esclau està processant una comanda molt llarga, i vol que el màster retransmeti la petició més tard.
08	MEMORY PARITY ERROR	L'esclau ha intentat llegir de la memòria, però ha detectat un error de paritat.

Taula 13 – Codis d'excepció ModBus

I finalment passem a detallar el mapa de memòria d'aquest equip. El mapa de registres és important per monitoritzar, configurar i controlar el mòdul d'entrades i sortides.

Paràmetre	**Offset registre**	**Adreça registre**	**Funcions suportades**	**Unitats**	**Mínim escala**	**Màxim escala**
Voltatge EA1	0	0000	0x04	V	0	Vmax
Voltatge EA2	1	0001		V	0	Vmax
Corrent EA3	2	0002		mA	Imin	Imax
Corrent EA4	3	0003		mA	Imin	Imax
Voltatge SA1	0	0000	0x03 0x06 0x10	V	0	Vmax
Voltatge SA2	1	0001		V	0	Vmax
Adreça A	2	0002	0x06	-	1	247
Velocitat V	3	0003	0x06	bps	1200	19200
Paritat P	4	0004	0x06	-	-	-

Taula 14 – Mapa de memòria ModBus

On:

- V_{max}= 1023, que equival a 10 V, i cal dividir-lo per 1023 externament
- I_{min}= 1023, que equival a la sortida de 4 mA, però cal dividir-lo per 1023 externament
- I_{max} = 0, que equival a 20 mA, però cal dividir-lo per 1023 externament
- La paritat podrà ser parella, senar o no haver-hi bit de paritat.

També falta assignar una adreça al nostre equip, que serà sempre la mateixa. Assumim que la seva adreça per defecte és la 1. Aquesta es pot canviar si es desitja, a través del registre 40003, sempre i quan sigui entre 1 i 247.

Cal dir també que es considera que aquest dispositiu guarda i transmet el byte *high* primer, i després el *low* byte.

A més, s'assumeix que la paritat per defecte és parella, però a través del registre 40005 es pot canviar. La velocitat de transmissió també es pot modificar de la que s'ha assignat per defecte utilitzant el registre 40004. La velocitat de transmissió assignada per defecte és de 9600 bps.

8. El microprocessador

En aquest apartat s'explica el procés de selecció del microprocessador més adequat per a aquesta aplicació. A la vegada, es dissenya l'algorisme de control del microprocessador, tant pel control intern del processador com per la gestió del protocol de comunicacions ModBus.

8.1. Selecció del microprocessador

La selecció del microcontrolador PIC és un aspecte important del projecte ja que representa el centre d'operacions de l'equip. El microcontrolador té diferents entrades, les quals aportaran una sortida determinada segons un algorisme que s'haurà creat prèviament. Aquest algorisme ha de valorar el tipus d'entrada i la transformació a aplicar per tal d'obtenir la sortida esperada.

Per aquest equip necessitem que el microprocessador permeti quatre entrades analògiques que tinguin la capacitat de poder convertir el senyal a digital amb un mínim de 10 bits de precisió, és a dir, que tingui com a mínim quatre canals conversors d'analògic a digital (ADC – *Analogic to Digital Converter*).

Per altra banda, necessitarem un mínim de dues sortides analògiques que incorporin conversors de senyal digital a analògic (DAC – *Digital to Analogic Converter*) també amb un mínim de 10 bits de precisió.

I també és necessari que tingui un UART compatible amb RS-485, amb selecció de paritat, selecció de velocitat de transmissió i identificador d'adreça de dispositiu.

També n'és important el seu rellotge, que marca el ritme de funcionament i sincronitza els diferents elements.

I finalment, s'ha buscat un microcontrolador que consumeixi el mínim possible.

Tenint en compte totes aquestes consideracions, el microcontrolador que s'ha escollit és el dsPIC33FJ64PG802[9] de Microchip.

Les principals característiques d'aquest controlador es mostren a continuació:

Device	Pins	Program Flash Memory (Kbyte)	RAM (Kbyte)(1)	Remappable Peripheral									RTCC	I²C™	CRC Generator	10-bit/12-bit ADC (Channels)	16-bit Audio DAC (Pins)	Analog Comparator (2 Channels/Voltage Regulator)	8-bit Parallel Master Port (Address Lines)	I/O Pins	Packages
				Remappable Pins	16-bit Timer(2)	Input Capture	Output Compare Standard PWM	Data Converter Interface	UART	SPI	ECAN™	External Interrupts(3)									
dsPIC33FJ64GP804	44	64	16	26	5	4	4	1	2	2	1	3	1	1	1	13	6	1/1	11	35	QFN TQFP

Note 1: RAM size is inclusive of 2 Kbytes of DMA RAM for all devices except dsPIC33FJ32GP302/304, which include 1 Kbyte of DMA RAM.
2: Only four out of five timers are remappable.
3: Only two out of three interrupts are remappable.

Figura 22 – Característiques principals del microcontrolador seleccionat

El microprocessador té memòria de programa, FLASH, on s'hi emmagatzema el codi; i memòria de dades RAM, on s'hi emmagatzemen les dades a enviar. Bàsicament, els

[9] http://ww1.microchip.com/downloads/en/DeviceDoc/70292G.pdf

microcontroladors tenen una memòria interna que emmagatzema dos tipus de dades, les intruccions, que corresponen al programa que s'executa; i els registres, que són les dades que l'usuari manipula, i també registres de control per a les diferents funcions del microprocessador.

L'arquitectura d'aquest controlador es mostra a la figura següent.

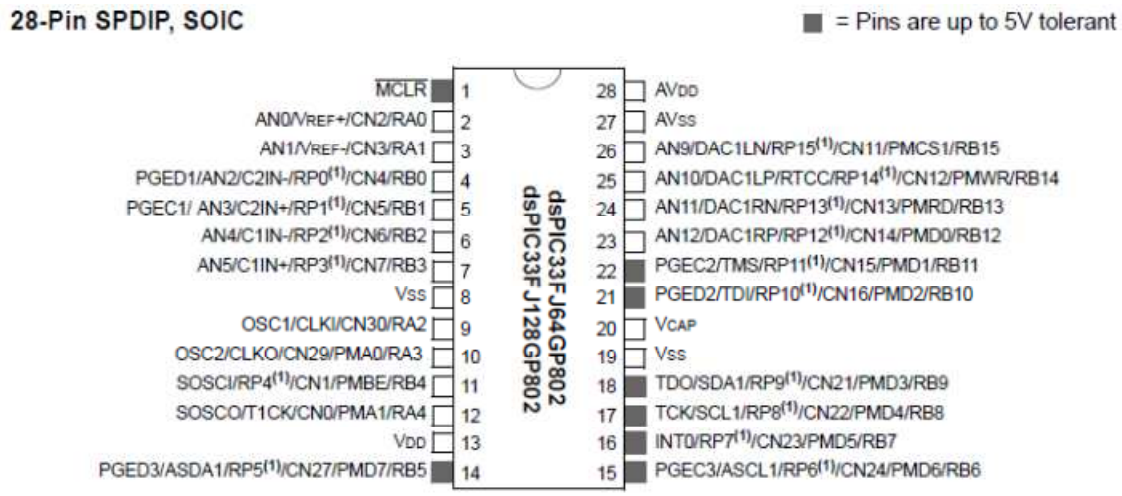


Figura 23 – Diagrama de pins del microcontrolador

En el nostre cas, s'han connectat les entrades i sortides en els pins com s'indica a la taula següent, a part de l'alimentació i massa, que van connectats òbviament als pins corresponents i també al pin MCLR, com s'indica al seu *datasheet*.

Paràmetre	**Offset registre**	**Adreça registre**	**Adreça ModBus**	**Pin (número)**	**Pin (nom)**
Voltatge EA1	0	0000	30001	2	AN0
Voltatge EA2	1	0001	30002	3	AN1
Corrent EA3	2	0002	30003	4	AN2
Corrent EA4	3	0003	30004	5	AN3
Voltatge SA1	0	0000	40001	23	DAC1RP
Voltatge SA2	1	0001	40002	25	DAC1LP
UART	-	-	-	15, 16 i 17	RP6, RP7 i RP8

Taula 15 – Relació entre adreces i pins del microprocessador

Presentem a continuació el mapa de memòria FLASH d'aquest microcontrolador:

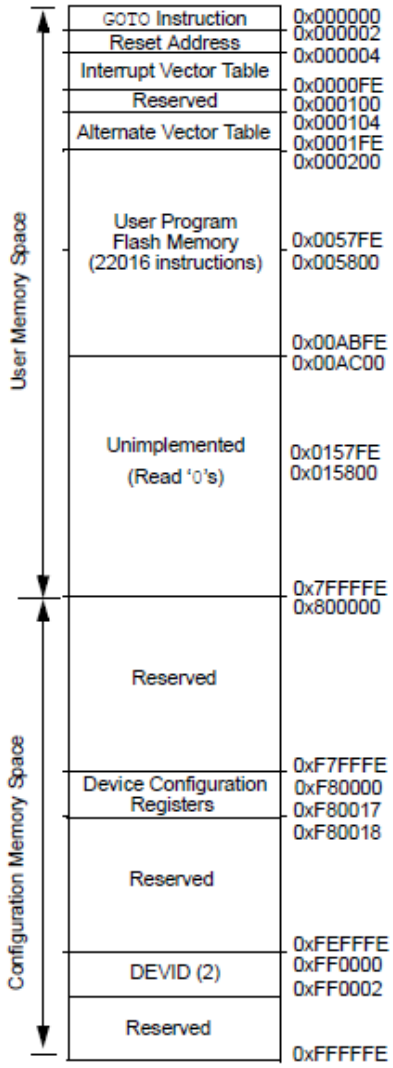


Figura 24 – Mapa de memòria FLASH del microcontrolador

8.2. El transductor RS-485

Un cop seleccionat el microprocessador, s'ha vist que aquest no incorpora una sortida RS-485, i per tant, cal un adaptador que passi la sortida UART del microprocessador a un format compatible amb RS-485. Hem seleccionat el 3483E[10] de Maxim pel seu baix consum i també perquè ja té els pins d'entrada i sortida aïllats i protegits fins a +/-15 kV. També redueix les reflexions causades pels cables que no estan acabats adequadament. Pot arribar a transmetre fins a 250 kbps, cosa que ja és suficient per a la nostra aplicació.

Veiem a la figura següent que cada encapsulat conté un emissor i un receptor. A la vegada, també cal fixar-nos amb la resistència R_t, la resistència de finalització, que és necessària per assegurar que la informació es transmeti sense rebots. El seu valor és

[10] http://datasheets.maximintegrated.com/en/ds/MAX3483E-MAX3491E.pdf

normalment de 120 Ohms, que es pot veure més endavant que s'ha inclòs a l'esquemàtic del circuit.

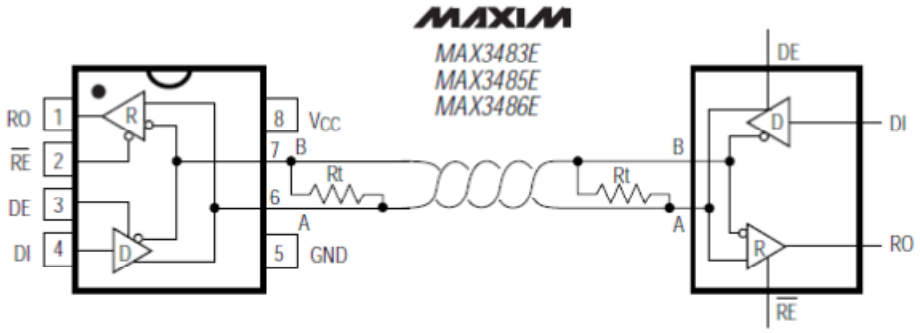


Figura 25 – Comunicació transductor Maxim 3483E

Com que treballem amb mode *half duplex*, els pins *enable* d'aquest xip estan units per tal d'aconseguir que o bé funcioni el receptor i es puguin rebre trames, o bé funcioni l'emissor i es puguin enviar trames. Ho podem veure a la figura següent.

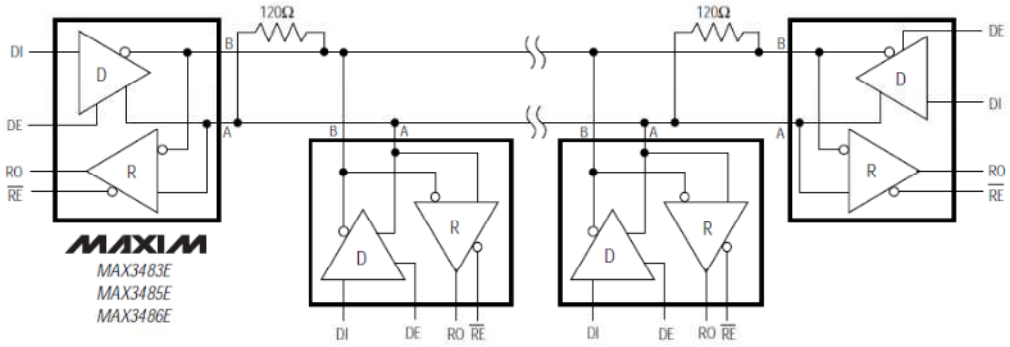


Figura 26 – Xarxa *half duplex* transductor Maxim 3483E

8.3. Programació del microcontrolador

Per a la programació del microcontrolador s'ha utilitzat el llenguatge C i el programa de simulació MPLAB del fabricant del microprocessador, Microchip, amb el compilador adequat pel microchip escollit, XC16. Tots dos es poden descarregar gratuïtament des de la web de Microchip.

Per a facilitar la programació s'utilitzen algunes llibreries que ja venen amb el compilador i incorporen funcions per a la programació dels mòduls UART, ADC i DAC; són *uart.h*, *adc.h* i *dac.h*. No adjuntarem aquestes llibreries en aquesta memòria ja que són de domini públic. No s'ha trobat cap llibreria que incorpori les funcions ModBus adequades per a aquest compilador. Per tant, s'ha hagut de programar des de zero. El programa consisteix en l'arxiu main.c, i les capçaleres que acabem d'esmentar, a més d'algunes altres capçaleres generals, les *stdio.h* i *stdlib.h*.

A continuació s'explica en què consisteix la filosofia del programa. Cal dir que no s'ha incorporat la funció 0x03 en el programa. Primer de tot, cal fer la configuració dels mòduls UART, ADC i DAC, tenint en compte el mode de funcionament que es necessita per a la nostra aplicació, amb un OpenUART1, OpenADC1 i OpenDAC. Els detalls de la configuració de cada un d'ells s'expliquen en el programa main.c, que

s'adjunta amb la memòria, i també en mostrem algunes de les parts més rellevants a continuació. Fem la configuració del UART com segueix:

```
baud = 5; //Valor per escriure al registre UxBRG per establir el 'baud rate'

U1MODEvalue = UART_EN & UART_IDLE_CON & // Habilitem (Enable) el UART &
//habilitem el mode d'operació Idle
        UART_IrDA_DISABLE & UART_MODE_SIMPLEX & //Desahibilitem el IrDA & el
//mode del pin UxRTS com a Simplex
        UART_UEN_00 & //Posem el mode dels pins UxCTS i UxRTS habilitant UxTx UxRx
        UART_EN_WAKE & //Permet que es desperti quan està en mode Sleep
        UART_DIS_LOOPBACK & UART_EN_ABAUD & //Desahibilitem el mode loopback
//& habilitem el mòdul de captura del 'baud rate' automàticament
        UART_UXRX_IDLE_ONE & UART_BRGH_FOUR & //Posem UxRX Idle state a '1'
//& posem 'High Speed mode'
        UART_EVEN_PAR_8BIT & UART_1STOPBIT; //Transmissió de dades de 8 bits
//amb paritat parella i un bit de stop

U1STAvalue = UART_SYNC_BREAK_ENABLED & //Habilitem que envii un 'Sync Break'
//a la següent transmissió
        UART_TX_ENABLE &  //Habilitem la transmissió pel pin TX
        UART_TX_BUF_FUL & //Li diem que envii una interrupció quan el buffer de
//transmissió està ple
        UART_ADR_DETECT_DIS & UART_RX_OVERRUN_CLEAR; //Deshabilitem el mode
//de detecció d'adreces & habilitem el bit de l'estat de overrun clear

//Obrim el mòdul UART

OpenUART1 (U1MODEvalue, U1STAvalue, baud);
```

I configurem el mòdul ADC com segueix:

```
config1 = ADC_MODULE_ON & ADC_IDLE_STOP & //Engeguem el mòdul ADC & parem el mòdul quan
//entra el mode idle
    ADC_AD12B_10BIT & //Posem el mode d'operació a 10 bits i 4 canals
    ADC_FORMAT_INTG & ADC_CLK_AUTO & //Posem el format de sortida com a unsigned int &
//el rellotge automàtic
    ADC_MULTIPLE & ADC_AUTO_SAMPLING_ON ; //Habilitem el mode de mostreig múltiple en
//seqüencia & Habilitem el mostreig automàtic

config2 = ADC_VREF_AVDD_AVSS & ADC_SCAN_ON & //Posem la referència de voltatge AVDD i
//AVSS & Habilitem l'escaneig pel CH0
    ADC_SELECT_CHAN_0123 & //Es permet utilitzar els canals 0, 1, 2 i 3
    ADC_BUF_FILL_0x8_0xF & ADC_ALT_BUF_OFF ;//ADC omple el buffer 0x8-0xF & Configurem
//el buffer en 2 de 8 bits

config3 =  ADC_CONV_CLK_SYSTEM & //S'utilitza el rellotge del sistema
    ADC_CONV_CLK_2Tcy; //Selecció de bits del ADC ADCS<7:0>

config4 = ADC_DMA_BUF_LOC_32; //Selecció dels paràmetres de configuració de registre
//ADxCON4, localització buffer DMA

configport_h = ENABLE_ALL_ANA_16_31;//Definició entrades analògiques registre ADPCFG

configport_l = ENABLE_ALL_ANA_0_15; //Definició entrades analògiques registre ADPCFG

configscan_h = SCAN_ALL_16_31;//Escanagem totes les entrades analògiques

configscan_l = SCAN_ALL_0_15;//Escanagem totes les entrades analògiques

//Obrim el mòdul ADC

OpenADC1(config1,config2,config3,config4,configport_l,configport_h,
configscan_h,configscan_l);
```

I per últim, la configuració del DAC s'ha fet com es mostra a continuació:

```
config1a = (DAC_MODULE_ON|DAC_CLOCK_DIV(1)| //Engegem el mòdul DAC & la selecció de
//divisors d'entrada del rellotge '1'
        DAC_FORMAT_UNSIGN_INT|DAC_ANALOG_AMP_OFF); //Posem el format de dades a 'Signed
//Int' & en mode 'Stop-in Idle'
config2a = (DAC_LEFT_ENABLE|DAC_RIGHT_ENABLE); //Habilitem el canal esquerra & habilitem
```

```
//el canal dret
config3a = (AUX_CLOCK_SOURCE_FVCO|AUX_OSC_MODE_DISABLED); //Seleccionem l'oscil.lador
//auxiliar, però el deshabilitem

//Obrim el mòdul DAC
    OpenDAC1(config1a,config2a,config3a);
```

A continuació, es segueix amb la recepció de dades des de UART pel mètode de *polling*, quan n'hi ha de disponibles, ja que és més senzill de programar i en aquest cas les interrupcions no són necessàries. Bàsicament aquesta serà la part principal del programa, on tenim un bucle que ens va mirant si hem rebut missatges ModBus, com presentem a continuació:

```
while (1){

    // Operacions UART

    while (BusyUART1());

    while(!DataRdyUART1())
           ;

    comptador=0;
    while(DataRdyUART1() && comptador<8 ){
            ( *(Receiveddata)++) = ReadUART1();
            comptador=comptador+1;
          }

    //mirar si la trama és correcta
    tramac=comprovatrama(Receiveddata);
    if (tramac==1){//si la trama és correcta - fem el processament
        processtrama(Receiveddata);
    }
    else
        //retorna error Modbus - segons l'error que es troba a la funció comprovatrama()
        funcioerror(cerror);
}
```

Quan es reben dades des de UART, cal comprovar que la trama sigui correcta, que ho fem cridant la funció *comprovatrama()*, com es veu a sobre. Aquesta funció comprovarà primer de tot que l'adreça de l'esclau sigui correcta, és a dir, que sigui 1, que tal i com ja s'ha comentat anteriorment s'ha assignat aquesta adreça al nostre dispositiu esclau. A continuació cal comprovar que la funció ModBus existeixi en el nostre dispositiu. Com que s'han programat les funcions ModBus 0x04, 0x06 i 0x10, qualsevol altra funció diferent haurà de retornar un error.

Es fa un tractament diferent quan la funció és la 0x10, és a dir, quan es volen escriure múltiples registres, ja que els missatges seran de més de 8 caràcters. Per tant, per aquesta funció, es mirarà quants registres vol escriure. De fet, com que només tenim dues sortides analògiques només podrà escriure una o dues sortides, i dependrà també de si comença a escriure a la sortida 1 o no. Cal comprovar que el nombre de bytes que ens indica coincideixi amb el nombre de registres multiplicat per dos, ja que cada registre són dos bytes. Si no s'ha detectat cap error, es llegeixen la resta de caràcters que se'ns passen des de UART, que depèn del nombre de registres que vol escriure. En canvi, per a les altres dues funcions, ens limitem a comprovar que el missatge sigui de 8 registres.

Finalment, cal comprovar que el valor CRC que porten les dades que ens arriben des de UART és correcte, cridant la funció que també s'ha programat en el main.c, anomenada CRC16().

La funció *comprovatrama()* retorna un 0 si hi ha algun error, indicant quin és el codi d'error; o bé un 1 si la trama és correcta, i es procedeix amb el processament de

dades, cridant la funció *processtrama()*. Mostrem a continuació el codi d'aquesta funció:

```
//Funció comprovar si la trama és correcta
int comprovatrama(char *Receiveddata){
    int tramacorrecta;

    //mirarem a sota si la llargada del missatge és correcta
    //suposem d'entrada que la trama és correcta
    tramacorrecta = 1;

    // mirem l'adreça on es vol adreçar, convertim els dos bytes en un únic nombre
    address = (Receiveddata[2] * 256) + Receiveddata[3];

    //cal comprovar que l'adreça sigui vàlida més endavant, dins de cada funció
    // mirar si l'adreca de l'esclau és la correcta
    if (Receiveddata[0]!=1){
        //torna l'error ModBus 0x03
        cerror=3;
        tramacorrecta=0;
    }
    //mirar si la funció que UART ens passa amb el missatge Modbus existeix
    else if (Receiveddata[1]!=4||Receiveddata[1]!=6||Receiveddata[1]!=16){
    //funció error 01 - la funció no és suportada pel dispositiu
        cerror=1;
        tramacorrecta=0;
    }
    //si la funció és la 16 (llegir múltiples registres) cal mirar quants en vol
    //escriure i continuar llegint
    else if (Receiveddata[1]==16){
        quantity = (Receiveddata[4]*256)+Receiveddata[5]; //mirem quants registres vol
//escriure
        if (quantity == 1 || quantity == 2){
            if (Receiveddata[6]==(2*quantity)){//comprovem que el número de bytes sigui
//el doble que el número de registres
                max = 2 * quantity + comptador + 1;//afegim les dades extres (2 per
//registre + 1 de byte count)
                while(DataRdyUART1() && comptador<max ){//continuem llegint pel nombre
//d'entrades
                        ( *(Receiveddata)++) = ReadUART1();
                        comptador=comptador+1;
                    }
            }
            else
                cerror=3;
                tramacorrecta=0;
        }
        else
            cerror=3;//retorna error 0x03
            tramacorrecta=0;
    }
    else if (Receiveddata[1]==3||Receiveddata[1]==4){
    //mirar si tenim un missatge de 8 caràcters
        if (comptador<8||comptador>8){
            //torna l'error Modbus 0x03
            cerror=3;
            tramacorrecta=0;
        }
    }
    //fer comprovació CRC cridant la funció CRC16()
    CRCresult = CRC16(Receiveddata,comptador-2);
    //posem el CRC en una única variable
    CRCllegit = (Receiveddata[comptador-2] * 256) + Receiveddata[comptador-1];

    //comprovem CRC sigui correcte
    if (tramacorrecta == 1){
        if (CRCresult != CRCllegit){
            //error modbus 0x02
            cerror=2;
            tramacorrecta=0;
        }
    }
    return tramacorrecta;
}
```

Pel tractament d'errors s'ha programat una funció anomenada *funcioerror(),* que es crida quan hi ha un error passant-li el codi d'error pertinent, i aquesta retorna una resposta ModBus al UART. La seva programació s'ha fet com segueix:

```
//Funció error pel processament d'errors
void funcioerror(unsigned int codierror){
        error[0]=Receiveddata[0];
        error[1]=Receiveddata[1]+0x80;
        //torna error ModBus amb el codierror que se li ha passat
        error[2]=codierror;
        CRCresult=CRC16(error,3);
        error[3]=CRCresult/256;
        error[4]=CRCresult%256;
        putsUART1((unsigned int *)error);
}
```

Hem comentat que tenim tres funcions diferents, depenent de si la instrucció és de lectura (0x04), on es voldrà llegir alguna de les entrades analògiques; o bé si és d'escriptura (0x06), on es voldrà escriure un cert valor en alguna de les sortides analògiques; o si és d'escriptura de múltiples registres (0x10), on es voldrà escriure per més d'una de les sortides analògiques.

Si el què es vol és llegir d'una entrada analògica, s'haurà d'identificar la quantitat de valors que vol llegir i per quina adreça vol començar la lectura, on cada una de les quatre entrades analògiques té una adreça, tal i com s'ha descrit al capítol de ModBus. Cal comprovar doncs, depenent de la quantitat de valors que vol llegir i l'adreça on vol començar a llegir, si és possible. Bàsicament, si es comença a llegir des de la primera entrada es podran llegir com a molt quatre valors, un per cada entrada. Ara bé, si es comença a llegir per la quarta entrada, només es podrà llegir el valor d'aquesta. Un cop fet això, es trien els canals del ADC amb la funció *SetChanADC1()*, i es comença la conversió i lectura de les dades amb les funcions *ConvertADC1()* i *ReadADC1()*. A la vegada, cal tenir clar quins valors es llegiran, depenent de si llegim de les entrades EA1, EA2 o bé EA3, EA4, és a dir, les entrades de tensió o corrent, com segueix:

- Entrades de tensió – les entrades de tensió analògiques van de 0 a 10 V, i un cop llegit el valor que entra al microprocessador i passat per l'ADC corresponent, caldrà dividir-lo per 1023 (2^{10}-1), ja que tenim una precisió de 10 bits. Com ja s'ha vist, es suposa que l'operació de dividir-lo per 1023 es farà externament.
- Entrades de corrent – aquestes entrades van de 4 a 20 mA. En aquest cas també caldrà dividir per 1023, i també deixem aquesta operació perquè es faci de forma externa.

Un cop obtinguts els valor d'intensitat o corrent que es volen llegir, llavors es procedeix a passar la resposta altra vegada amb format ModBus cap al UART utilitzant la funció *putsUART1()*. Per posar el missatge a punt cal calcular-ne el CRC per tal d'afegir-lo al final del missatge de resposta.

Passem ara a descriure el programa en la situació en què ModBus passa un missatge amb una funció d'escriptura 0x06. En aquest cas, el missatge ens passa l'adreça on vol escriure de les dues sortides possibles que tenim a l'equip, i el valor que hi vol escriure. El valor de les sortides analògiques pot ser entre 0 V a 10 V, i per tant, aquí s'haurà fer una conversió a l'escala de 0 V a 3.3 V, que després de passar pel DAC, per passar-la a analògica, s'adaptarà segons el circuit d'adaptació de sortides que s'ha explicat en apartats anteriors. Per tant, en aquest cas s'haurà de dividir el valor que ens passa ModBus per 3, per tal de tenir uns valors de voltatge compatibles amb el microprocessador. A la vegada, s'haurà de multiplicar per 1023, per la mateixa raó

explicada a sobre. Llavors s'ha de tornar una resposta a través de UART també amb format ModBus, que en aquest cas serà exactament el mateix missatge que ens ha passat el UART.

Per a la funció d'escriptura de registres múltiples, cal primer de tot comprovar que l'adreça de les sortides analògiques sigui correcta, és a dir, que sigui 0 o 1. També cal mirar que el nombre de sortides que vol escriure és possible, depenent de des de quina adreça vol començar a escriure, de manera semblant a com s'ha fet per la lectura d'entrades analògiques. Si no hi ha cap error, llavors es fa la conversió de digital a analògic. I finalment es retorna una resposta ModBus indicant quants registres s'han escrit.

El tractament de lectures i escriptures es fa en una funció que s'ha anomenat *processtrama()*, que retornarà un error, o escriurà per UART les respostes de les lectures o escriptures. Les multiplicacions i divisions es fan de manera externa. El codi d'aquesta funció, que és la funció principal i més complexa, es presenta de manera resumida a continuació:

```
//Funció de processament de la trama
void processtrama(char *Receiveddata){
    //mirar quina funció porta si no hi ha trobat cap dels errors anteriors
    //porta funció lectura 0x04
    if (Receiveddata[1]==4){
        //mirem quants registres vol llegir
        quantity=(Receiveddata[4] * 256) + Receiveddata[5];
        //mirem la quantitat de registres sigui dins el rang de 1 a 4 que permet la
        //configuració del nostre dispositiu, ja que només tenim 4 entrades analògiques
        if (quantity>4||quantity<1){
            //exception code 03
            funcioerror(3);
        }
        //comprovem que l'adreça sigui correcta
        else if (address!=0 && address!=1 && address!=2 && address!=3){
            //torna error ModBus 0x03
            funcioerror(3);
        }
        else if (address==1 && quantity>3){
            //error 02 - si vol començar a llegir per l'adreça 1, només podrà llegir 3
//entrades
            funcioerror(2);
        }
        else if (address==2 && quantity>2){
            //error 02 - si vol començar a llegir per l'adreça 2, només podrà llegir 2
//entrades
            funcioerror(2);
        }
        else if (address==3 && quantity >1){
            //error 02 - si vol començar a llegir per l'adreça 3, només podrà llegir una
//entrada
            funcioerror(2);
        }
        else //si no ha trobat abans cap error fem la lectura de les entrades del ADC
//corresponent
        ii=0;
        for (i=0;i<quantity;i++){
            switch(address){
                case 0: //primera entrada analògica de voltatge
                    while(1){
                          ConvertADC1(); //Captura l'entrada analògica i la passa a
                          //digital
                          while(BusyADC1() )//Esperem que s'acabi la conversió
                                ;
                        SetChanADC1(ADC_CH123_NEG_SAMPLEA_VREFN &
ADC_CH123_POS_SAMPLEA_0_1_2, ADC_CH0_NEG_SAMPLEA_VREFN &ADC_CH0_POS_SAMPLEA_AN0 );
                        ventrada = ReadADC1(0);//Llegim el valor del ADC
                    }
                case 1://segona entrada analògica de voltatge
                    //fem el mateix que pel cas 0
                case 2://tercera entrada analògica - corrent
                    //fem el mateix que pel cas 0
```

```
                case 3://quarta entrada analògica - corrent
                    //fem el mateix que pel cas 0
             break;
            }
            ii=i*2;
            txdata[ii+3]=ventrada/256; //passem el valor a dos caràcters diferents
            txdata[ii+4]=ventrada%256;
            address=address+1;
        }
        txdata[0]=Receiveddata[0];//posem el missatge per tal de respondre a UART
        txdata[1]=Receiveddata[1];
        txdata[2]=ii;
        CRCresult = CRC16(txdata,ii+4);
        txdata[ii+5]=CRCresult/256;
        txdata[ii+6]=CRCresult%256;

        //passem dades a UART
        putsUART1((unsigned int *)txdata);
    }
    //porta funció escriptura 0x06
    else if (Receiveddata[1]==6){
        if (address!=0 && address!=1){ // mirem que l'adreça sigui vàlida
            //torna error Modbus 0x03
            funcioerror(3);
        }
        else //si no ha trobat cap error anteriorment
        switch(address){
            case 0://primera sortida de voltage pel DAC
                    while(1){
                        while(BusyDAC1R)//Esperem que acabi la conversió
                        ;
                        valueregister=(Receiveddata[4] * 256) + Receiveddata[5];
                        WriteDAC1R(valueregister);
                    }
            case 1://segona sortida de voltage pel DAC
                    //fem el mateix que pel cas 0
                break;
        }
        //passem dades al UART
        putsUART1((unsigned int *)Receiveddata); //passem la resposta a UART - en aquest
//cas és la mateixa que la pregunta
    }
    //porta funció escriptura registres múltiples 0x10
    if (Receiveddata[1]==16){
        //hem mirat la quantitat de registres sigui dins el rang de 1 a 2 que permet la
        //configuració del nostre dispositiu, ja que només tenim 2 sortides analògiques
        //a la funció comprovatrama()
        //comprovem que l'adreça sigui correcta
        if (address!=0 && address!=1 ){
            //torna error ModBus 0x03
            funcioerror(3);
        }
        else if (address==1 && quantity>1){
           //error 02 - si comença a llegir per l'adreca 1, només podrà llegir 1 entrada
            funcioerror(2);
        }
        else //si no ha trobat abans cap error fem l'escriptura de les sortides del DAC
//corresponent
        ii=0;
        for (i=0;i<quantity;i++){
            switch(address){
                case 0://primera sortida de voltage pel DAC
                        while(1){
                            while(BusyDAC1R) //Esperem que acabi la conversió
                            ;
                           valueregister=(Receiveddata[7+ii] * 256) + Receiveddata[8+ii];
                           WriteDAC1R(valueregister);
                        }
                case 1://segona sortida de voltage pel DAC
                      //fem el mateix que pel cas 0                        }
                    break;
                }
            ii=i*2;
            address=address+1;
        }
```

```
        txdata[0]=Receiveddata[0];//posem el missatge de resposta en format ModBus
        txdata[1]=Receiveddata[1];
        txdata[2]=Receiveddata[2];
        txdata[3]=Receiveddata[3];
        txdata[4]=Receiveddata[4];
        txdata[5]=Receiveddata[5];
        CRCresult=CRC16(txdata,6);
        txdata[7]=CRCresult/256;
        txdata[8]=CRCresult%256;
        //passem dades a UART
        putsUART1((unsigned int *)txdata);
    }
}
```

I falta afegir que per tots els casos d'error o excepcions que s'han esmentat, amb un *putsUART1()* es passa un missatge amb el codi d'excepció corresponent i format ModBus cap al màster, com es veu en el codi de més amunt.

I finalment, quan s'ha llegit un missatge que arriba des de UART, s'ha processat i se n'ha tornat la resposta, el dispositiu s'ha de quedar esperant preparat per si arriba un altre missatge des de l'UART, i ja pot tornar a començar el cercle de nou.

Només s'han programat les funcions de lectura de registres, 0x04, la d'escriptura d'un sol registre, 0x06, i la d'escriptura de múltiples registres, 0x10. No s'han inclòs funcions per a canviar de paritat, adreça de dispositiu o velocitat del dispositiu.

S'adjunten els fitxers del programa amb l'entrega de la memòria.

9. Disseny de la PCB

En aquest apartat s'explica el procés de disseny de la PCB (*printed circuit board*). El programa que s'ha utilitzat és l'*Eagle*, versió *Light*. Aquest programa no permet importar els esquemàtics dels diferents circuits que ja hem dissenyat en apartats anteriors, i per tant se n'ha de tornar a fer el seu esquemàtic des de zero. Com podem veure a l'esquemàtic de la Figura 27, la placa està formada pels següents circuits:

- Entrada – alimentacions d'entrada V_{dd} entre 12 V i 48 V, el terra *GND* i les quatre entrades analògiques.
- Sortida - Port d'interfície sèrie RS-485 entre la PCB i el PC, i les dues sortides analògiques.
- Components:
    - Circuits d'adaptació d'entrades i sortides
    - Font d'alimentació
    - Transductor RS-485
    - Microprocessador dsPIC

Cal esmentar també que alguns dels components no s'han trobat a cap de les llibreries disponibles a la web del programa *Eagle*, i s'han hagut de dissenyar des de zero.

Un cop es té l'esquemàtic dibuixat, ja es pot procedir amb el disseny del *layout*. El *layout* seria una aproximació al model real, és a dir, una fotografia de tal i com hauria de quedar la PCB un cop fabricada. El *layout* consisteix en la distribució física dels components electrònics a la placa amb les connexions definides a l'esquemàtic i en traçar les pistes entre ells. Els components es col·loquen de manera que minimitzin les interferències i el soroll, i també cal tenir en compte que els recorreguts haurien de ser el més curts i eficients possibles.

El mateix programa disposa d'un *Autorouter* que dibuixa de manera automàtica les pistes que connecten els diferents components dels circuits. Per circuits complexos l'enrutat no acostuma a ser òptim i cal fer modificacions a mà, com ha estat el cas en aquest projecte. S'ha tingut en compte que no és recomanable tenir angles de noranta graus en les pistes per tal de facilitar la circulació del corrent i evitar problemes per pèrdua de gruix de les pistes. A la vegada, també s'ha acabat el *layout* manualment ja que la funció d'enrutament automàtic no aconseguia acabar-lo de tot. Bàsicament, s'han introduït algunes vies addicionals per tal de passar algunes pistes per la cara inferior de la PCB.

En aquest cas, no tenim restriccions d'espai, i per tant escollim la mida de PCB que ens convé i que ens és més còmode per poder traçar les pistes sense problemes. Ara bé, cal tenir en compte que en un dispositiu real segurament tindríem unes mides màximes de la PCB on hi hauríem de col·locar tots els components i pistes. Les mides finals de la PCB han estat les màximes que ens permet la versió de prova d´*Eagle*, que són de 100 mm x 80 mm.

Actualment, la majoria de components estan disponibles al mercat en format *through hole* (TH), que significa a través del forat, o com a *surface mounting devices* (SMD), de muntatge superficial. El de tipus TH són els més antics i els més econòmics. Els SMD normalment estan orientats a entorns de gran consum. En el nostre cas utilitzaren components SMD sempre que puguem ja que acostumen a ser més petits i per tant permeten que el disseny de la PCB pugui ser més reduït. Un altre avantatge del tipus

SMD és que els components poden estar més junts i per tant es gasta menys pista; cosa que disminueix les interferències que emet o rep la placa, ja que hi ha menys zones de coure a la superfície de la PCB. A la vegada, també facilita l'enrutat de la PCB ja que és molt més net i no està plena de vies i forats per a les potes dels components. I per últim, cal dir que si la PCB i els seus components són més petits també se'n redueix el seu pes, cosa que podria ser molt important per aplicacions aerospacials, per exemple. Ara bé, els components SMD també tenen alguns inconvenients, com que pot ser difícil soldar-los a mà i també pot costar refredar-los ja que tenen menys superfície on dissipar la temperatura.

El disseny de la PCB s'ha fet en dues capes per tal de simplificar l'espai i complexitat. En la Figura 28 i la Figura 29 podem veure en verd els components, en vermell la capa superior de la PCB i en blau la seva capa inferior. També normalment s'afegeix el text a la cara superior.

Una altra consideració que s'ha tingut en compte és que s'han fet les pistes que porten més corrent una mica més gruixudes que les altres. Aquestes són, com hem vist a l'apartat del disseny de la font d'alimentació, les que entren i surten del circuit que transforma el voltatge d'entrada de 12 V a 48 V fins als 15 V, que alimentarà els amplificadors operacionals dels circuits adaptadors d'entrades i sortides analògiques i també inclourà les entrades de corrent cap als dos altres circuits de la font d'alimentació.

També s'ha afegit una capa de massa a la cara superior de la PCB, que consisteix en crear una àrea de coure pel senyal de massa en les àrees no utilitzades de la PCB, que és útil per reduir el soroll d'alta freqüència, i a la vegada ajuda a disminuir el nombre de pistes de la PCB.

Adjuntem a continuació l'esquemàtic i el *layout* que s'han dissenyat tenint en compte totes aquestes consideracions. Presentem també el *layout* de la cara superior de la PCB separadament, on es veu el pla de massa. Pel *layout* de la Figura 28, que inclou les dues cares, no es mostra el pla de massa, ja que sinó no es pot veure la resta. Cal dir però que en aquest *layout* les vies de massa no estan enrutades, ja que formaran part d'aquest pla de massa que en aquesta imatge no s'ensenya.

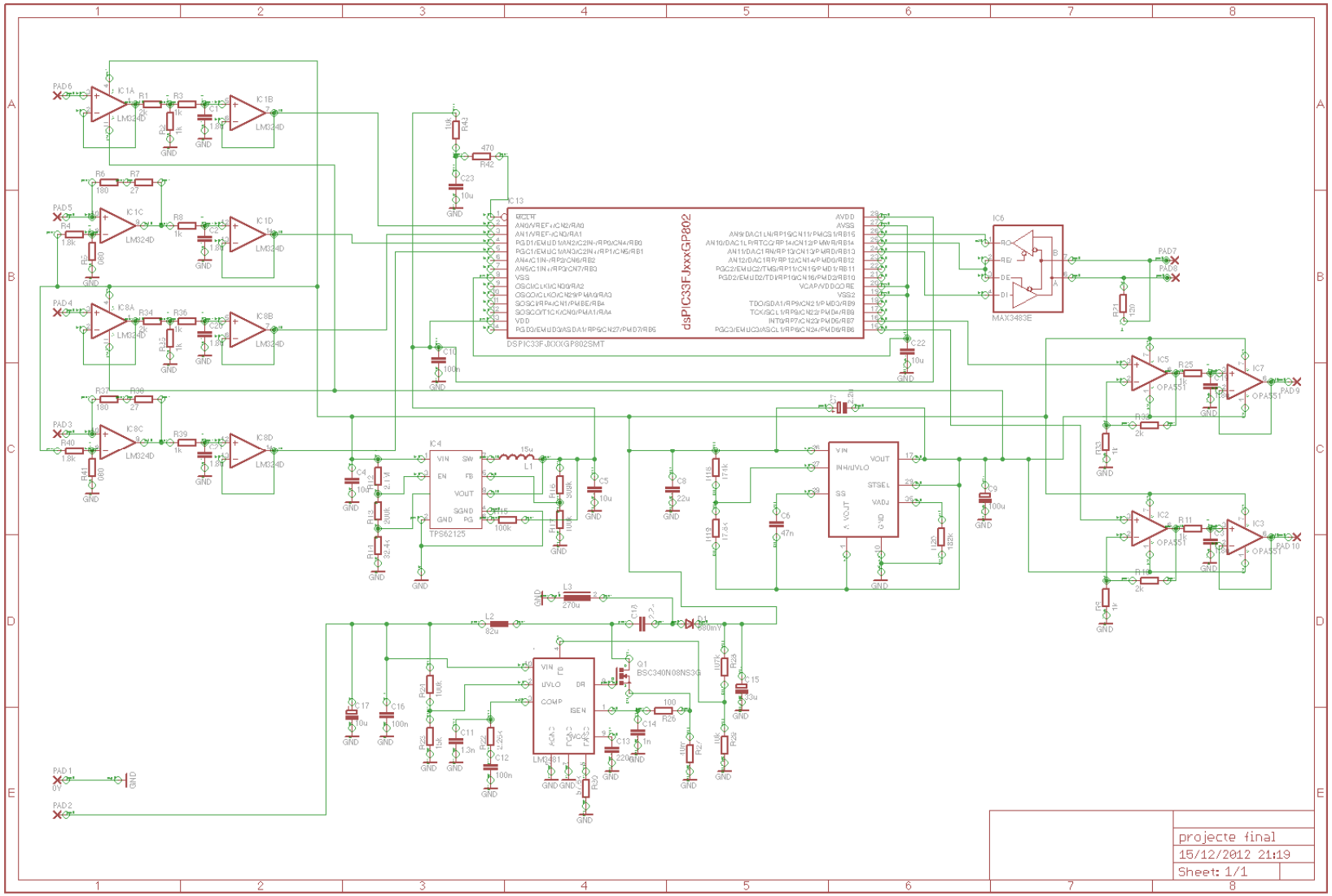


Figura 27 – Esquemàtic del circuit

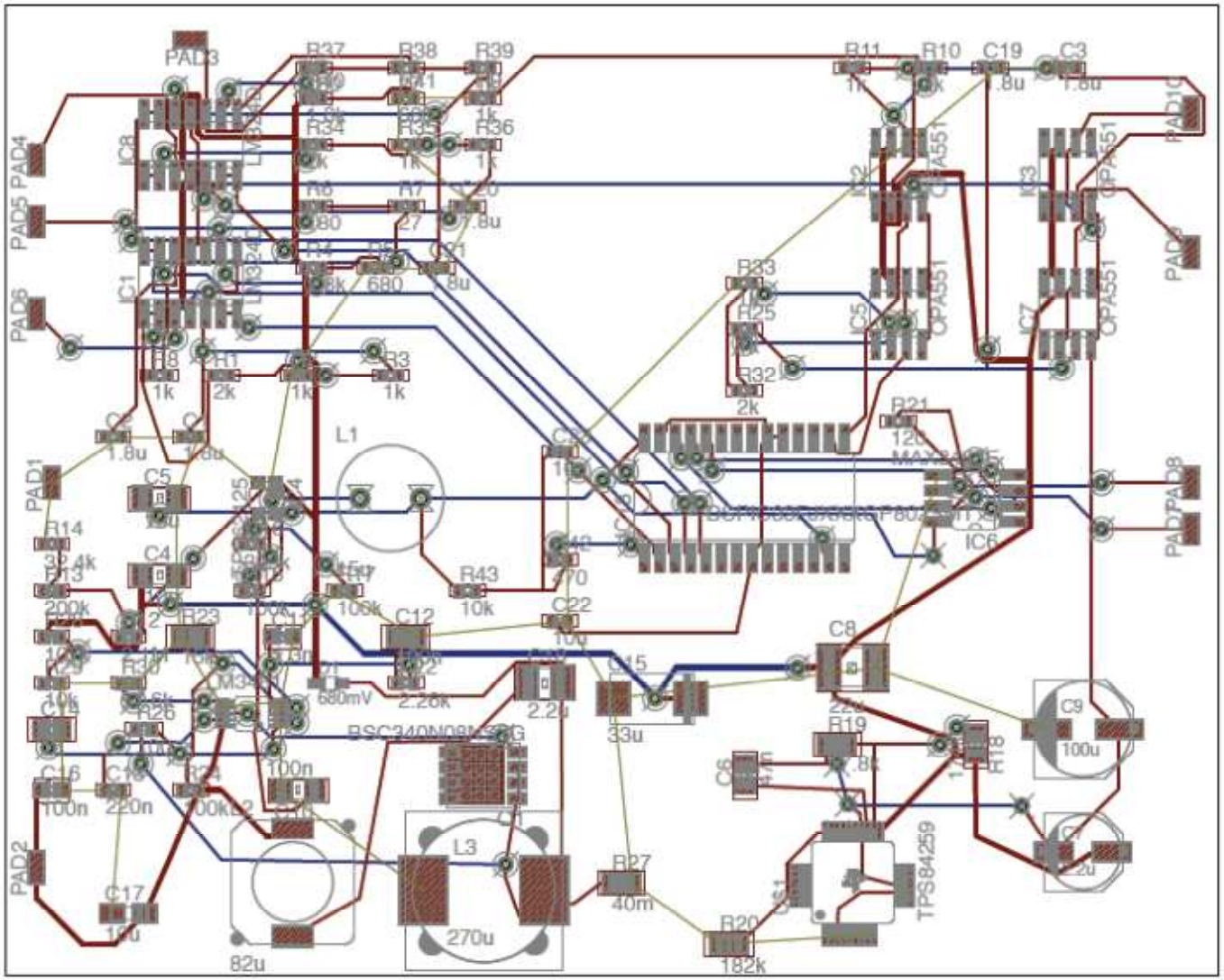


Figura 28 – *Layout* de la PCB

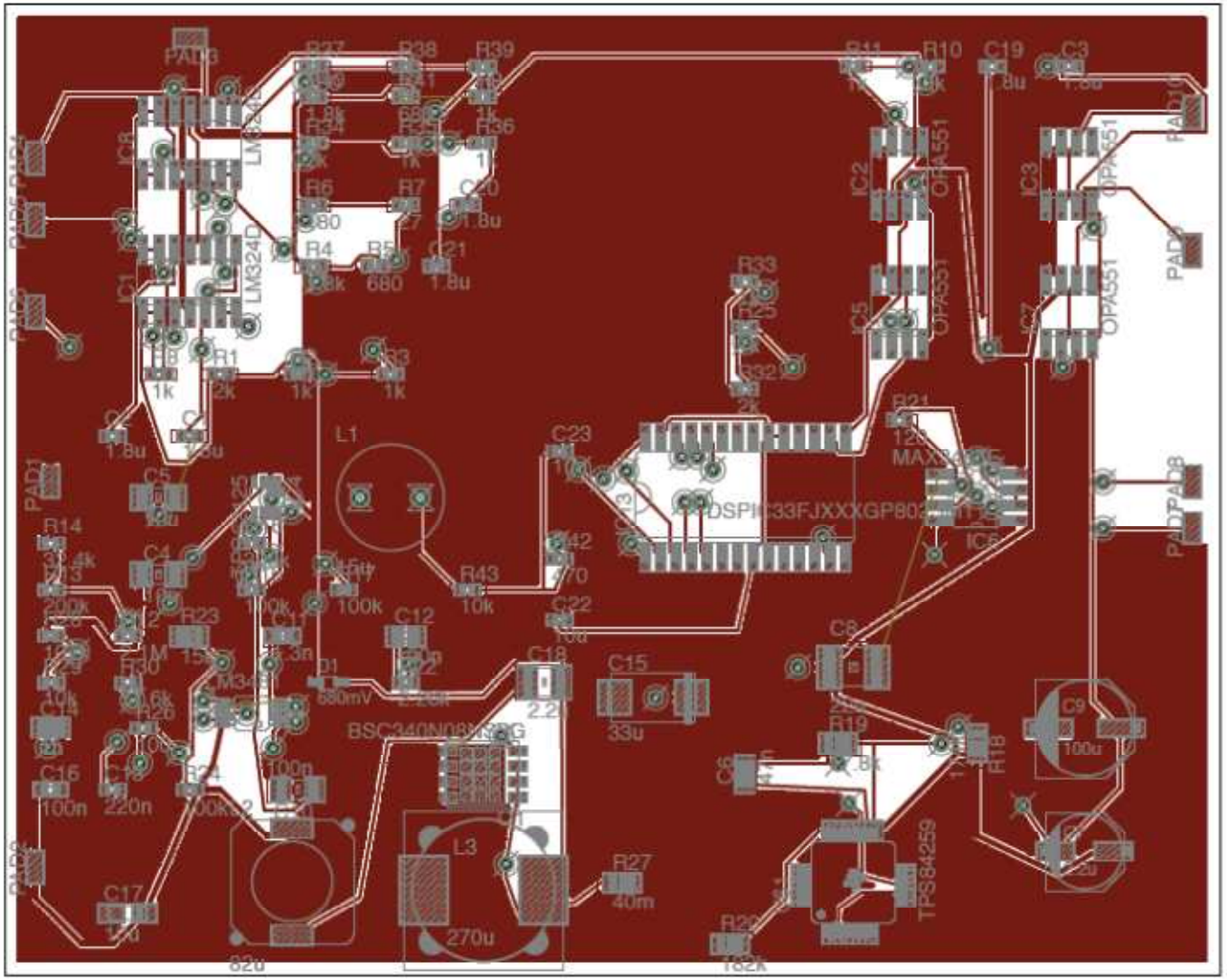


Figura 29 – *Layout* de la cara superior de la PCB amb el pla de massa

10. Conclusions

En aquest capítol fem un resum dels objectius assolits durant el desenvolupament d'aquest treball i també esmentarem les lliçons apreses.

Per començar, podem dir que en termes generals hem assolit els objectius finals d'aquest projecte, que serien comprendre i posar en pràctica els continguts teòrics que s'han anat adquirint durant la carrera d'Enginyeria Tècnica de Telecomunicacions. En alguns casos, com per exemple la programació del microprocessador, fins i tot s'han excedit aquests objectius, ja que és un tema que mai s'havia vist anteriorment, i per tant ha estat una part del treball més de recerca i ampliació que no pas de consolidació de coneixements. A més, també cal dir que ha estat molt útil el fet de desenvolupar el projecte tenint en compte la seva gestió, és a dir, passant per la seva planificació, seguiment, gestió de riscos i de costos, valoració de les lliçons apreses, etc. ja que és així com es gestiona un projecte en una empresa, i per tant és útil en qualsevol projecte que ens puguem trobar en el món laboral o quotidià.

Ja valorant ara els objectius tècnics, podem dir que hem assolit els majors objectius proposats inicialment; s'ha dissenyat tot un equip que permet adquirir valors de diferents entrades analògiques i treure diferents sortides analògiques, utilitzant el protocol ModBus a través de la interfície RS-485. S'ha dissenyat i simulat tota la part del hardware de manera acurada, tant pel què fa la font d'alimentació com els circuits d'adaptació d'entrades i sortides analògiques. A la vegada, s'ha entès el protocol ModBus i se n'ha generat el mapa de memòria. I també s'ha fet un bon disseny del *layout* de la PCB. Finalment, s'ha aconseguit entendre i aprofundir amb el tema dels microprocessadors, que era un tema totalment desconegut. Se n'ha entès el seu funcionament i el seu disseny tant des d'un punt de vista *hardware* com *software*, i se n'ha desenvolupat el programa que el controla, amb l'excepció que no hem aconseguit fer-ne la simulació degut a què el microprocessador que es va triar és massa complicat.

Per tant, una de les lliçons apreses és que si no es té experiència amb els microprocessadors, cal escollir-ne un de ben simple, com més senzill millor per tal de simplificar-ne la seva programació i simulació. Un altre punt important és escollir un dispositiu que sigui àmpliament utilitzat, ja que és molt més fàcil trobar-ne exemples i funcions o capçaleres ja programades. A més, sembla ser que potser MPLAB no és el programari més senzill o més comú, cosa que tampoc sabíem d'entrada, i també ens ha dificultat la seva programació. En el nostre cas, ha estat molt complicat trobar exemples, i a més, les funcions ModBus s'han hagut de programar des de zero, ja que no s'ha aconseguit trobar-ne cap que es pogués utilitzar per aquest microprocessador i el compilar XC16 de MPLAB.

Com s'ha explicat a l'apartat 3, a l'inici del projecte es va fer una planificació inicial, i se n'ha fet un seguiment al llarg del desenvolupament del treball. Si en fem una valoració, diria que la identificació de les tasques es va aconseguir fer de manera prou acurada tenint en compte el desconeixement de les dificultats tècniques que ens trobaríem. A la vegada, la planificació també es considera que ha estat acurada i s'han pogut assolir els objectius i lliuraments com es va planificar a l'inici del projecte; amb l'excepció de la programació del microcontrolador, que es va allargar molt més del planificat, però es va poder compensar amb la definició del mapa de memòria ModBus, que ens va portar menys temps del que estava programat. Ara bé, cal dir que en general, el projecte ens ha portat moltes més hores de les que s'havien estimat, sobretot la part de programació del microprocessador, on les hores que s'hi han dedicat s'han com a mínim multiplicat per quatre. Cal tenir en compte que això podria ser un greu problema

si estiguéssim dissenyant un producte per a comercialitzar, ja que significaria que els costos de disseny se’ns haurien multiplicat per quatre. En el nostre cas, com que ja era un dels riscos identificats, s’ha solucionat dedicant-hi més hores.

Així doncs, una altra de les lliçons apreses és que cal tenir un mínim coneixement tècnic del producte per tal de fer una bona planificació i estimació de les hores i costos del projecte abans de començar, ja que sinó això podria significar per un costat que no fos possible fer el projecte en els període amb el que ens hem compromès, o per un altre costat, amb els costos que s’han estimat. I per tant, una altra conclusió que n’hem tret, és que com més desconeixença es té del projecte, més contingència cal afegir en l’estimació d’hores i cost del producte. Ara bé, això pot no ser una situació ideal, ja que pot posar-nos en una posició poc competitiva en el mercat.

En resum, diria que s’ha après molt durant aquest projecte, tant pel què fa a la gestió d’un projecte, com per un altre costat per consolidar i aprofundir en els conceptes i coneixements estudiats al llarg de la carrera, com el disseny de *hardware*, i també de *software*; a més d’adquirir altres coneixements nous i que eren totalment desconeguts per nosaltres abans de començar aquest treball.

11. Futures ampliacions

Anem a explicar aquí algunes ampliacions que creiem que es podrien fer partint de la base d'aquest projecte.

Una de les ampliacions més evidents seria acabar de fer el programa del microprocessador perquè incorpori totes les funcions que s'han llistat a l'apartat de ModBus d'aquesta memòria, i degut a les restriccions de temps i d'experiència no s'han incorporat en l'abast d'aquest treball.

Per altra banda, també es podria canviar el microprocessador per un altre que fos potser més econòmic i adequat per a aquesta aplicació. Considerem que el microprocessador que s'ha escollit té moltes funcions que no són necessàries per a l'abast d'aquest projecte, que molts dels altres més senzills no incorporen, tot i que té l'avantatge d'incorporar els DACs. Escollir-ne un altre de més senzill simplificaria la programació del dispositiu, a més de fer el disseny del producte total més econòmic.

Pel què fa al disseny de la PCB, en aquest cas no és massa important ja que no es condueix molt corrent; però un aspecte que potser estaria bé mirar seria la dissipació, si més no seria un aspecte important pel disseny de PCBs al qual no hem estat exposats. Seria una extensió ja que no entra dins els objectius d'aquest treball, però seria interessant poder aprofundir més en aquest tema ja que és un aspecte que normalment cal tenir en compte.

Un altre aspecte que seria interessant és fer una avaluació dels costos totals de tot l'equip expansor d'entrades i sortides analògiques dissenyat. Cal dir que durant el desenvolupament del projecte ja s'han tingut en compte els costos per a l'elecció de components, amb l'excepció del microprocessador, que no s'ha fet així per falta d'experiència. Ara bé, el què no s'ha tingut en compte és la seva fabricació, ja que ens hauria calgut tenir els *gerbers* de la PCB i enviar-los a un fabricant perquè ens en fes un pressupost. A la vegada, per tenir el preu total, també caldria considerar les hores que hem passat fent el disseny. Cal dir però, que en aquest cas no hem cregut adequat fer aquest anàlisi de costos, ja que degut a la inexperiència ens sortiria un dispositiu molt car; però si que seria un aspecte important a tenir en compte a la vida real.

També crec que hauria estat molt interessant i gratificant poder fabricar els circuits, i programar un microprocessador real, i poder-los testejar al laboratori. Crec que aquest és un aspecte fonamental que en tot projecte d'electrònica tindria molt de sentit poder fer.

12. Bibliografia

Es llisten a continuació els materials que s'han consultat al llarg del treball:

[1] **Pérez M. A.; Álvarez J. C.; Campo J.C.; Ferrero F.J.; Grillo G.J., "Instrumentación electrónica" 2004**, Editorial Thomson.

[2] **Doebelin. E.O., "Measurement systems. Application and design." 2003**, Editorial McGraw-Hill.

[3] **Recasens Bellver, M.A.; González Calabuig, J., "Diseño de circuitos impresos con ORCAD CAPTURE y LAYOUT v 9.2" 2004**, Paraninfo.

[4] **Roselló Canal, M., "Aplicacions electromagnètiques i electròniques" 2006,** PID_00163582, Materials Docents de la UOC.

[5] **Pérez, A.; Bara, M.; Martínez, J.A., Roselló, M., "Fonaments tecnològics II" 2010**, PID_00153323, Materials Docents de la UOC.

[6] **Pérez, A.; Martínez, J.A.; Muñoz, O., "Sistemes electrònics digitals" 2010**, 19.010, Materials Docents de la UOC.

[7] www.analog.com

[8] www.ti.com

[9] www.microchip.com

[10] www.linear.com

[11] www.modbus.org

[12] www.wikipedia.com

[13] http://www.lammertbies.nl/comm/info/RS-485.html

[14] www.termcat.cat

[15] http://www.electronicafacil.net/

[16] http://www.electronics-tutorials.ws/filter/filter_5.html

[17] http://www.national.com/AU/design/courses/117/print.htm#00sol08

[18] http://metis.umh.es/jacarrasco/docencia/ep/Tema3/DCDC.pdf

[19] http://www.dte.us.es/ing_inf/ins_elec/temario/Tema%201.%20Amplificadores%20Operacionales.pdf

[20] www.simplymodbus.ca

[21] www.acromag.com

[22] www.exemys.com

[23] www.selco.com

[24] www.tyco.com

[25] www.abb.com

[26] www.cadsoftusa.com

[27] www.tutoelectro.com

[28] http://developer.berlios.de/project/showfiles.php?group_id=6120

[29] http://www.0x00f.com/files/ebooks/Programming%20dsPIC%20_Digital%20Signal%20Controllers_%20in%20C.pdf

[30] http://www.microbuilder.eu/Tutorials/Fundamentals/SMDPackages.aspx