



Escola Universitària
Politécnica de Mataró

Enginyeria Tècnica Industrial: Especialitat Electrònica Industrial

**ANALITZADOR DE LA XARXA ELÈCTRICA
TRIFÀSICA DE PROPÒSIT GENERAL**

A – Memòria descriptiva

**JOAN MARTÍNEZ NÚÑEZ
JULIÁN HORRILLO TELLO**

PRIMAVERA 2010

RESUM

En els processos productius industrials no hi ha res pitjor que el disjuntor o el magnetotèrmic que es disparen contínuament en el moment més inoportú. O aquella instal·lació que de sobte es queda sense tensió perquè s'han fos els fusibles degut a un curtcircuit, aturant la producció. Aquí sí que hi ha tensió!... però no elèctrica, sinó tensió nerviosa perquè la producció no surt. Hi ha alguna manera d'evitar aquests errors que sovint són fatals? Es podria preveure un possible mal funcionament?

La monitorització i enregistrament de dades en temps real dels consums dels sistemes trifàsics, com per exemple bombes, ventiladors, instal·lacions o màquines i el seu posterior anàlisi poden ajudar als tècnics de manteniment a conèixer el rendiment real de la seva instal·lació i preveure possibles avaries. Estan compensats els consums de les tres fases? Quina és la evolució o tendència del consum del sistema trifàsic? Si té tendència a l'alça, vol dir que hi ha alguna part del sistema que necessita manteniment preventiu?

Aquest projecte pretén donar resposta a aquestes qüestions i tracta sobre el disseny, fabricació i programació d'una tarja amb microcontrolador, capaç de mesurar en temps real una magnitud de la xarxa elèctrica trifàsica (en aquest cas el corrent), amb la finalitat d'estudiar el comportament dinàmic i evolució de les càrregues d'una aplicació real, uns transelevadors d'un magatzem automàtic.

RESUMEN

No hay nada más molesto en un proceso industrial que un disyuntor o un magnetotérmico que se disparan continuamente en el momento más inoportuno. O aquella instalación que de repente se queda sin tensión porque se han fundido los fusibles por un cortocircuito, deteniendo la producción. ¡Aquí sí que hay tensión!... pero no del tipo eléctrico sino debida a los nervios porque no sale la producción. ¿Existe alguna manera de evitar que se produzcan estos errores fatales? ¿Se podría prever un posible mal funcionamiento?

La monitorización y registro de datos en tiempo real del consumo de sistemas trifásicos, como son bombas, ventiladores, instalaciones o máquinas y su posterior análisis pueden ser de gran ayuda para los técnicos de mantenimiento, pues les permite conocer el rendimiento real de su instalación y anticiparse a posibles averías. ¿Están compensadas las cargas de las tres fases? ¿Cuál es la evolución o tendencia de los consumos del sistema trifásico? Si tiene tendencia alcista, ¿quiere decir que hay una parte del sistema que requiere mantenimiento preventivo?

Este proyecto pretende dar respuesta a estas cuestiones y trata sobre el diseño, la fabricación y programación de una tarjeta con microcontrolador capaz de medir en tiempo real una magnitud de la red eléctrica trifásica (la corriente en este caso), con la finalidad de estudiar el comportamiento dinámico y la evolución de las cargas de una aplicación real, unos transelevadores de un almacén automático.

ABSTRACT

There's nothing more annoying in an industrial process than a circuit breaker or a breaker that faults continuously at the most inopportune time. Or that installation that suddenly runs out of tension because the fuses are blown by a short circuit, shutting down the production. Here there is indeed some tension! ... But not the electric kind, but due to nerves because it leaves the production. Is there any way to avoid these fatal errors? Could provide a possible malfunction?

Monitoring and recording data in real time consumption phase systems such as pumps, fans, plant or machinery and its subsequent analysis may be helpful for maintenance technicians, it can help them meet their real performance installation and anticipate possible breakdowns. Are offset the costs of the three phases? What are the evolution or consumption trends of three-phase system? If it has a rising trend, does that mean that there is a part of the system that requires preventive maintenance?

This project seeks to answer these questions and discusses the design, manufacturing and programming a microcontroller card capable of real-time measuring a magnitude of three phase electrical network (the current in this case) with the aim of studying the dynamic behavior and the evolution of a real application loads, some cranes in an automatic warehouse.

ÍNDIX

1. OBJECTE DEL PROJECTE	1
2. ABAST DEL PROJECTE	3
3. ANTECEDENTS DEL PROJECTE I ESTAT DE L'ART	5
4. ESPECIFICACIONS TÈCNIQUES	11
5. SOLUCIÓ TECNOLÒGICA	13
5.1. Introducció.....	13
5.2. Sensor de corrent	14
5.2.1. Tipus de sensors de corrent	14
5.2.2. <i>Shunt</i> de corrent de baixa resistència.....	14
5.2.3. Transformador de corrent	15
5.2.4. Sensor d'efecte Hall.....	15
5.2.5. Criteris de selecció dels sensors de corrent	16
5.2.6. Els transformadors de corrent TP-420.....	17
5.3. Microcontrolador	20
5.3.1. Criteris de selecció del microcontrolador.....	20
5.3.2. Microcontrolador PIC16F877A-I/P de <i>Microchip</i>	22
5.4. Circuit analògic.....	26
5.4.1. Convertidor I/V.....	26
5.4.2. Adaptador de nivells.....	29
5.4.3. Linealitat del circuit adaptador de nivells.....	35
5.5. Mòdul convertidor A/D del microcontrolador PIC16F877A	36
5.6. Mòduls CCP2 i TIMER1: Automatització del convertidor A/D.....	38
5.7. Interrupcions del PIC16F877A.....	42
5.7.1. Vector d'interrupció	44
5.7.2. Salvar el context o estat de l'aplicació	44
5.7.3. Interrupcions externa i de conversió A/D finalitzada.....	45
5.8. Teclat i <i>key-encoder</i>	46
5.9. Visualitzador LCD.....	52
5.10. Connectivitat RS232.....	55

5.11. RTC i bus I ² C™	57
5.11.1. RTC	57
5.11.2. Bus I ² C™	59
6. MUNTATGE DEL PROTOTIP I PROVES DE FUNCIONAMENT.....	63
6.1. Muntatge del prototip	63
6.2. Proves de funcionament	65
7. DESCRIPCIÓ DEL CODI DE PROGRAMA.....	67
7.1. Eines de desenvolupament	67
7.2. <i>Driver</i> del visualitzador LCD.....	70
7.3. <i>Driver</i> del RTC.....	73
7.4. Codi del programa principal.....	75
7.5. Codi de les interrupcions.....	80
7.5.1. Rutina servei interrupció teclat	80
7.5.2. Rutina servei interrupció ADC.....	86
8. CONCLUSIONS I MILLORES.....	91
9. REFERÈNCIES	93
9.1. Bibliografia.....	93
9.2. Webgrafia.....	94

ÍNDIX DE FIGURES

Figura 3.1. Passadís d'emmagatzematge dels transelevadors	6
Figura 3.2. Carros mòbils de presa de corrent dels transelevadors	6
Figura 3.3. Anàlitzador de la xarxa elèctrica de <i>CIRCUTOR</i>	7
Figura 3.4. Anàlitzador de la xarxa elèctrica de <i>FLUKE</i>	8
Figura 3.5. Pantalles <i>FLUKE 434</i>	9
Figura 5.1. Diagrama de blocs de l'anàlitzador de la xarxa elèctrica trifàsica.....	13
Figura 5.2. Sensor de corrent <i>shunt</i>	14
Figura 5.3. Transformador de corrent.....	15
Figura 5.4. Sensor de corrent d'efecte Hall per PCB	16
Figura 5.5. Transformador de corrent de nucli partit obert	17
Figura 5.6. Transformador de corrent <i>TP-420-23</i> de 200A.....	18
Figura 5.7. Relació R_{Lmax} / tensió d'alimentació del llaç de corrent	19
Figura 5.8. Característiques tècniques del TC model <i>TP-420-23-M70216</i>	19
Figura 5.9. Microcontrolador PIC16F877A-I/P de <i>Microchip</i>	24
Figura 5.10. Arquitectura interna del microcontrolador PIC16F877A	25
Figura 5.11. Circuit convertidor corrent – tensió (I/V)	26
Figura 5.12. Esquema convertidor I/V de tres fases.....	28
Figura 5.13. Circuit convertidor I/V de tres fases	29
Figura 5.14. Funció de transferència $V_o = f(V_i)$	30
Figura 5.15. Circuit sumador – restador	31
Figura 5.16. Esquema adaptador de nivells per una fase.....	32
Figura 5.17. Divisor de tensió per la tensió de referència	34
Figura 5.18. Circuit adaptador de nivells de tres fases	35
Figura 5.19. Model de les entrades analògiques.....	36
Figura 5.20. Diagrama de blocs del convertidor A/D.....	37
Figura 5.21. Diagrama de flux de la conversió A/D automàtica	41
Figura 5.22. Lògica de control de les interrupcions	43
Figura 5.23. Teclat matricial 4x4	47
Figura 5.24. Connexió interna dels polsadors del teclat.....	48
Figura 5.25. Diagrama de blocs del <i>key-encoder</i> MM74C922.....	50
Figura 5.26. Gràfiques freqüència d'escaneig i temps anti-rebot.....	51

Figura 5.27. Connexió del <i>key-encoder</i> en el circuit.....	51
Figura 5.28. <i>Key-encoder</i> i condensadors externs.....	52
Figura 5.29. Visualitzador LCD.....	52
Figura 5.30. Diagrama de blocs del visualitzador.....	53
Figura 5.31. Connexió del visualitzador en el circuit.....	54
Figura 5.32. Configuració del ST232CN.....	56
Figura 5.33. Connexió del ST232CN en el circuit.....	57
Figura 5.34. Configuració dels pins del DS1307.....	58
Figura 5.35. Connexió del DS1307 en el circuit.....	59
Figura 5.36. Transferència sencera d'una dada al bus I2C™.....	61
Figura 6.1. Vista de la cara superior del prototip.....	64
Figura 6.2. Vista de la cara inferior del prototip.....	64
Figura 6.3. Muntatge del projecte sencer.....	65
Figura 6.4. Proves de funcionament amb el simulador.....	66
Figura 7.1. Connexió del ICD2 en el circuit.....	69
Figura 7.2. MPLAB® ICD2.....	69
Figura 7.3. Registres amb les adreces i els formats del DS1307.....	73
Figura 7.4. Organització de la memòria del PIC16F877A.....	76
Figura 7.5. Codi del programa principal des del MPLAB® IDE.....	78
Figura 7.6. Ordinograma programa principal.....	79
Figura 7.7. Opcions menú principal.....	81
Figura 7.8. Primera opció menú principal.....	82
Figura 7.9. Ordinograma RSI teclat.....	84
Figura 7.10. Ordinograma acció tecla A polsada.....	85
Figura 7.11. Ordinograma acció tecla 5 polsada.....	85
Figura 7.12. Ordinograma RSI ADC.....	88

ÍNDIX DE TAULES

Taula 3.1. Cost equip analitzador de la xarxa elèctrica de <i>CIRCUTOR</i>	8
Taula 3.2. Especificacions <i>FLUKE 434</i>	10
Taula 5.1. Resultats proves linealitat circuit adaptador de nivells	35
Taula 5.2. Relació del codi segons la tecla polsada	50
Taula 7.1. Exemple d'ús del modificador %nt	72

1. OBJECTE DEL PROJECTE

L'objecte del projecte és el disseny, construcció i programació d'una tarja amb microcontrolador capaç de mesurar i visualitzar en temps real en un *display* LCD, una magnitud de la xarxa elèctrica trifàsica, en aquest cas el corrent, amb la finalitat d'estudiar el comportament dinàmic i l'evolució del consum elèctric dels transelevadors d'un magatzem automàtic.

Tot i què l'equip s'utilitzarà en una aplicació molt concreta, com a monitoritzador de càrregues d'uns transelevadors, per les seves prestacions i versatilitat ha de mantenir el caràcter d'aplicació de propòsit general, per tal de poder analitzar el comportament d'equips i sistemes trifàsics, per exemple: bombes, ventiladors, i d'altres instal·lacions industrials.

L'analitzador ha de ser capaç d'enviar mostres periòdiques del consum de cada fase (R, S, T) al PC a través del port de comunicacions sèrie, utilitzant el protocol RS232. Això permetrà el posterior anàlisi de les dades, fer gràfiques per poder veure la tendència del consum i la possible descompensació entre fases, calcular el consum elèctric real, etc. En definitiva l'analitzador ha de permetre conèixer el rendiment real de la instal·lació.

També ha de reaccionar quan es produeixi un consum excessiu, donant avís al sistema de govern de la instal·lació, com ara un PLC. L'usuari podrà interactuar amb l'equip a través d'un visualitzador LCD i un teclat matricial 4x4.

Es pretén oferir un equip senzill, fàcil d'utilitzar i amb bones prestacions, tot integrat en una sola tarja i a un preu raonable.

Des del punt de vista personal el desenvolupament del present projecte suposa portar a la pràctica els coneixements i les habilitats que s'han anat adquirint al llarg dels estudis d'Enginyeria Tècnica Industrial, des del desenvolupament de sistemes electrònics complexes fins a la programació en llenguatges d'alt nivell, com ara el llenguatge C.

2 *Objecte del projecte*

2. ABAST DEL PROJECTE

L'abast d'aquest projecte es centra en el desenvolupament d'una tarja amb tots els elements necessaris per tal de ser utilitzada com analitzador de la xarxa elèctrica trifàsica de propòsit general i que sigui capaç d'assolir els objectius fixats en l'apartat "Objecte del Projecte".

L'analitzador es centrarà en mesurar una magnitud elèctrica, en aquest cas el corrent. El mòdul principal no inclourà la part del sensor ni el circuit analògic condicionador del senyal, els quals formaran un mòdul independent. El fet de dissenyar un mòdul principal per a l'analitzador i un mòdul secundari per al sensor incrementa les possibilitats de què sigui un equip obert i què amb poques modificacions de *software* i de *hardware* es puguin mesurar altres magnituds.

L'*Analitzador de la Xarxa Elèctrica Trifàsica de Propòsit General* no pretén rivalitzar amb els equips comercials per les seves prestacions però sí per el preu i per la facilitat d'ús. Per utilitzar l'analitzador no és necessari que l'usuari tingui molts coneixements d'electricitat o d'electrònica; més aviat amb aquest equip es pretén que qualsevol operari que tingui unes poques habilitats sobre el PC i els programes d'ofimàtica pugui conèixer el rendiment real de la instal·lació i fer-ne l'anàlisi.

Per fer les proves de funcionament del prototip també es dissenyarà un mòdul per mesurar el corrent elèctric de la xarxa elèctrica trifàsica, incloent-t'hi els sensors, uns transformadors de corrent amb un rang de 0 a 200A i el circuit analògic condicionador del senyal per tal de es pugui connectar a l'entrada del ADC del microcontrolador.

El sistema inclourà un RTC, és a dir, un rellotge de temps real, que permetrà l'equip fer diverses operacions, com per exemple: càlcul del promig del consum diari per fase i enregistrament de l'hora en què s'ha produït l'alarma.

Es pretén mantenir una forta interacció entre l'equip i l'usuari, de manera que el binomi teclat – LCD permeti a l'usuari conèixer en tot moment l'estat de la instal·lació: es podrà programar la data i l'hora i visualitzar-les en qualsevol moment sense perdre l'enregistrament de dades en temps real. També es podrà programar el llindar de consum a partir del qual s'activarà l'alarma i consultar-lo en qualsevol moment. Es podrà consultar els valors de consum màxim per fase i el consum promig del dia anterior. I evidentment es podrà visualitzar en temps real en el LCD el consum de cada fase.

Si en qualsevol moment se supera el llindar d'alarma programat, el microcontrolador ha d'activar un LED vermell d'alarma i un relé, de manera que, convenientment connectat al PLC de govern de la instal·lació, es puguin engegar les accions pertinents per atendre l'alarma, per exemple, donar l'ordre a la màquina per què s'aturi. A partir de l'instant en què el microcontrolador detecta que s'ha produït un excés de consum s'emmagatzemarà en la memòria RAM cinquanta mostres per fase consecutives a l'alarma, juntament amb l'hora en què s'ha produït l'esdeveniment. Això permetrà comprovar si les accions correctores han estat suficients i si el sistema s'ha restablert de la incidència. Quan l'usuari ho desitgi es podrà enviar tota aquesta informació al PC pitjant un botó.

Tot i què l'enregistrament de les dades en suport PC és un dels trets més importants d'aquest projecte, queda fora de l'abast del mateix el tractament de les dades o l'automatització de les gràfiques. Només això justificaria la realització d'un altre projecte. Per fer l'enregistrament de les dades s'utilitzarà el *software* de comunicacions sèrie *HyperTerminal*, que ve inclòs amb el paquet del sistema operatiu *Microsoft Windows*[®] *XP*. Aquest programa permet l'enregistrament permanent de les dades en un arxiu de text sense limitació de capacitat. Per fer les proves sobre el sistema real això serà suficient.

El muntatge dels circuits es farà utilitzant la tècnica de construcció de prototips electrònics anomenada *wire-wrapping* combinada amb la soldadura manual amb estany. Aquesta tècnica permet muntar els circuits amb menys temps que el sistema tradicional de fotolit i insolació sobre placa de circuit imprès; a més facilita enormement la correcció d'errors.

Tot el sistema es muntarà sobre unes plaques metàl·liques per facilitar el trasllat i s'inclouran les fonts d'alimentació que es necessitin per el funcionament dels circuits.

3. ANTECEDENTS DEL PROJECTE I ESTAT DE L'ART

Aquest projecte és la resposta a la petició del Departament de Manteniment del Magatzem Automàtic que l'empresa Henkel Ibèrica, S.A. té a la seva planta de producció de Montornès del Vallès. Es va sol·licitar un equip senzill que fos capaç de mesurar en temps real i enregistrar mostres en un PC del consum per fase dels transelevadors del magatzem.

Els transelevadors son unes màquines mòbils i autònomes, integrades en el procés productiu de la planta, completament automatitzat, que poden emmagatzemar el producte acabat en un magatzem.

Des de fa cert temps es venen produint unes avaries esporàdiques i imprevisibles en el sistema d'alimentació d'aquestes màquines que han arribat a produir curtcircuits, fonent els fusibles de protecció i aturant la producció. Després de moltes comprovacions sobre les característiques de disseny dels dispositius d'alimentació i l'aplicació d'aquests en els transelevadors, no s'ha trobat la causa que provoca els curtcircuits. És per això que el Departament de Manteniment del Magatzem Automàtic va valorar la possibilitat de muntar un equip de mesura que aportés informació sobre l'estat de la xarxa d'alimentació, especialment dels consums. El present projecte, com ja s'ha esmentat abans, pretén donar resposta a aquesta petició.

Es va valorar la possibilitat de muntar un equip comercial, però es va descartar per el seu elevat cost, amb la premissa que no se sabia amb exactitud si el resultat que s'obtindria justificaria la inversió. Els equips comercials són molt potents i ofereixen unes prestacions que en aquesta aplicació son innecessàries. A més per utilitzar adequadament aquests equips es necessita cert entrenament. El *software* d'anàlisi, el mòdul de comunicacions sèrie i les sortides digitals sovint es venen com a mòduls separats, la qual cosa encareix més el producte.

És per això que es pretén dissenyar un equip que compleixi amb les especificacions, fàcil d'utilitzar i relativament econòmic.



Figura 3.1. Passadís d'emmagatzematge dels transelevadors
(Font: Elaboració pròpia)



Figura 3.2. Carros mòbils de presa de corrent dels transelevadors
(Font: Elaboració pròpia)

Com exemple de l'oferta que es pot trobar al mercat s'han seleccionat dos equips: un dels fabricants catalans de *CIRCUTOR*, molt modular i per instal·lar en l'armari on hi ha l'aparellatge elèctric de la instal·lació, i un altre equip portàtil molt potent de la casa americana *FLUKE*.

M.5
Analizadores de redes, CVM


Serie CVM 96									
SERIE CVM 96 - PANEL 96x96 - ALIMENTACIÓN 230 Vc.a.									
	Tipo	Código	Entrada ITF	Salida relés	Corriente neutro	Armónicos	Protocolo	Comunicación	EURO
	CVM 96	[*] M51100	-	-	-	-	-	-	330,02
	CVM 96-ITF	[*] M51200	Si	-	-	-	-	-	385,68
	CVM 96-ITF-RS485-C2	[*] M51211	Si	2	-	-	Modbus/RTU	RS-485	449,71
	CVM 96-F-ITF-RS485-C2-HAR-IN	[*] M51513	Si	2	Si	I (31°)	Modbus/RTU	RS-485	493,77
	CVM 96-ITF-Ethernet-C2-TCP (*)	[*] M51241	Si	2	-	-	Modbus/TCP	Ethernet	690,32
	CVM 96-ITF-Johnson-C2	[*] M51711	Si	2	-	-	Johnson Controls	RS-485	493,77
	(*) Equipos con comunicaciones Ethernet sólo disponible con alimentación 230 Vc.a.								

Figura 3.3. Analitzador de la xarxa elèctrica de *CIRCUTOR*

(Font: Catàleg de *CIRCUTOR*)

A la figura 3.3 es pot veure un dels equips més econòmics que ofereix *CIRCUTOR*, amb les següents prestacions:

- Mesura de volts i ampers del sistema trifàsic.
- Valor eficaç real.
- Màxim consum d'energia.
- Descomposició harmònica.
- Comptador d'energia kW·h, kW·h C, kW·h L.
- Dues sortides digitals.
- *Display* per LEDs.
- Comunicació RS485.

Hi ha la opció d'escollir entre diferents nivells de connectivitat, fins i tot connectivitat Ethernet, però el preu augmenta considerablement des de la versió més econòmica. A més s'ha de comptar apart els transformadors de corrent, tres unitats del model TP-23, amb un rang de 0...200A al primari i 0...5A al secundari (75,77 € per unitat més IVA); un convertidor de RS232/485 per a PC (322,13 € més IVA) i el *software* de visualització i enregistrament dels paràmetres elèctrics en temps real, el *PowerStudio* (165,01 € més IVA). La taula 3.1 reflecteix el cost final d'un analitzador de la xarxa elèctrica trifàsica comercial, de la casa *CIRCUTOR*:

Data: 24/05/2010

Mòdul	unitats	Cost unitari	Cost total (+ IVA)
CVM 96-ITF-RS485-C2	1	449,71 €	521,66 €
Transformadors TP-23 200/5A	3	75,77 €	263,68 €
Convertidor RS232/485 per a PC	1	322,13 €	373,67 €
Software <i>PowerStudio</i>	1	165,01 €	191,41 €
COST TOTAL (IVA inclòs) =			1.350,43 €

Taula 3.1. Cost equip analitzador de la xarxa elèctrica de *CIRCUTOR*

(Font: Elaboració pròpia)

L'equip dels americans de *FLUKE* es mostra a la figura següent:



Fluke 434

Figura 3.4. Analitzador de la xarxa elèctrica de *FLUKE*

(Font: Catàleg de *FLUKE*)

Les prestacions d'aquest equip, el *FLUKE 434*, son espectaculars, i el preu també: 4.404 €, sense IVA. L'equip inclou quatre pinces amperimètriques, també per fer mesures al neutre. És capaç de mesurar tots els paràmetres que es poden mesurar en una xarxa elèctrica trifàsica: tensió, corrent, potència, energia, desequilibri de fases, harmònics i interharmònics, fluctuacions, transitoris, interrupcions i canvis sobtats de tensió.

La figura 3.5 mostra dues de les pantalles que ofereix el *FLUKE 434* per visualitzar les mesures i la tendència; i la taula 3.2 reflecteix les especificacions de l'equip.

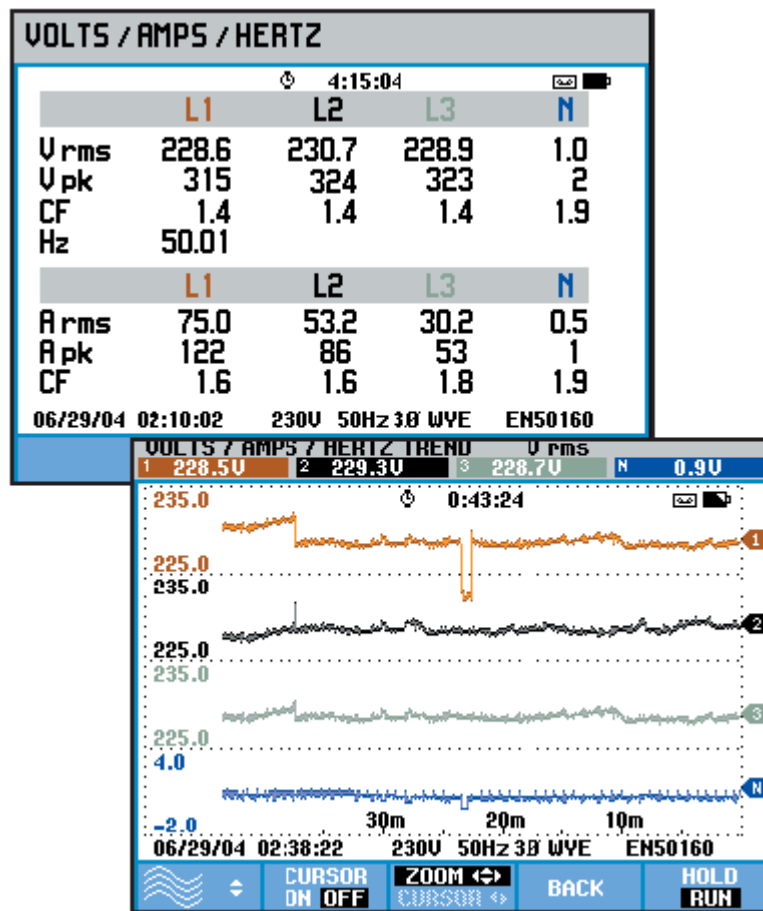


Figura 3.5. Pantalles *FLUKE 434*

(Font: Catàleg de *FLUKE*)

Entradas	Número de entradas	4 de tensión y corriente (3 fases + neutro)		
	Tensión máxima de entrada	1000 Vrms (pico de 6kV)		
	Velocidad máxima de muestreo	200 kS/s en cada canal simultáneamente		
Voltios/Amperios/Hz		Rango de medida	Precisión	
	Vrms (CA+CC)	1...1000 V	± 0,5% de la tensión nominal	
	Vpk	1...1400 V	5% de la tensión nominal	
	Factor de cresta	1,0 ... > 2,8	± 5%	
	Arms (CA+CC)	0...20,000 A	± 1% ± 5 cuentas	
	Amperios de pico	0 - 5500 A	5%	
	Factor de cresta	1 ... 10	± 5%	
	Hz	50Hz nominal	42.50 ... 57.50 Hz	± 0,1% de la frecuencia nominal
	Fluctuaciones	Vrms (CA+CC) ²	0,0% ...100% de la tensión nominal	
		Arms (CA+CC) ²	0 ... 20,000 A ²	
Armónicos	Armónicos (interarmónicos) (n)	DC, 1.50; (desactivado, 1.49) medido respecto a la fundamental o al valor rms total		
	Vrms	0,0 ... 1000 V	± 5% ± 2 cuentas	
	Arms	0,0 ... 4000 mV x escala de la pinza de corriente	± 5% ± 5 cuentas	
	Vatios	Según escala de la pinza de corriente		
			± 5% ± n x 2% o lectura, ± 10 cuentas	
	Tensión de CC	0,0 ... 1000 V	± 5% ± 10 cuentas	
	THD	0,0 ... 100,0 %	± 2,5%	
	Hz	0 ... 3500 Hz	± 1 Hz	
	Ángulo de fase	-360° ... +360°	± n x 1,5°	
	Potencia y energía	Vatios, VA, VAR	1,0 ... 20,00 MVA ¹	± 1,5% ± 10 cuentas
kWh, kVAh, kVARh		00,00 ... 200,0 GVAh ¹	± 1,5% ± 10 cuentas	
Factor de potencia/Cos φ /DPF		0...1	± 0,03	
Flicker (Parpadeo de tensión)	Pst (1minuto), Pst, Plt, PFS	0,00 ... 20,00	±5%	
Desequilibrio	Voltios	0,0 ... 5,0%	± 0,5%	
	Corriente	0,0 ... 20%	± 1%	
Captura de transitorios	Voltios	± 6000 V	± 2,5% de Vrms	
	Duración de detección mínima	5 µs (muestreo de 200 kS/s)		
Modo arranque de motores	Arms (CA + CC)	0,000 ... 20,00 kA ¹	± 1% de medidas ± 5 cuentas	
	Duración del arranque (seleccionable)	7,5 s ... 30 minutos	± 20 ms (frecuencia nominal - 50 Hz)	
Registro AutoTrend	Muestreo	Hasta 100 lecturas por segundo de muestreo continuo por canal		
	Memoria	Hasta 3600 puntos de min, max y promedio		
	Tiempo de registro	Hasta 450 días		
	Zoom	Hasta 12 aumentos de zoom horizontal		
Memoria	Pantallas	50 en Fluke 434; 25 en Fluke 433		
	Datos	10 en Fluke 434; 5 en Fluke 433		
Normas	Procedimientos de medida utilizados	IEC61000-4-30 clase A; EN50160; IEC 61000-4-15; IEC 61000-4-7		

Taula 3.2. Especificacions *FLUKE 434*(Font: Catàleg de *FLUKE*)

L'equip que és objecte d'aquest projecte no pretén assolir els nivells de prestacions dels millors equips comercials... ni tampoc el seu elevat cost. L'acceptació d'aquest repte ofereix la possibilitat de desenvolupar un projecte que tindrà una aplicació real. Això sens dubte provoca una forta motivació, ja que és més interessant la realització d'un projecte real que desenvolupar un treball que només tingui una aplicació teòrica.

Aquest és un projecte d'enginyeria en el sentit més ampli del terme, donat que en alguns llibres de teoria es diu que "l'activitat principal dels enginyers és *resoldre problemes pràctics*" (Monté i Safont, 2004).

4. ESPECIFICACIONS TÈCNIQUES

Les característiques tècniques de l'Analitzador de la Xarxa Elèctrica Trifàsica de Propòsit General són les següents:

- Alimentació del circuit amb $\pm 15V$ i pressa de 0V per connector de tres vies.
- Indicació de presència de tensió de 5V al circuit amb un LED verd.
- *Display* LCD de dues línies i setze caràcters per línia (2 x 16).
- Retroil·luminació del *display* per LEDs, alimentat amb font d'alimentació externa.
- Teclat matricial 4x4.
- Connectivitat RS232 amb connector DB9 femella.
- Connectivitat de programació i depuració en circuit ICD2 de *Microchip* amb connector RJ11 femella de 6 vies.
- RTC incorporat (*Real Time Clock*, o rellotge de temps real).
- LED vermell d'alarma.
- Relé d'alarma amb contacte lliure de potencial N.O. de 200V / 500mA amb connector de dues vies.
- Connector de 10 vies amb alimentació $\pm 15V$, 5V, massa, i tres entrades analògiques amb un rang de 0 a 5V, per connectar-hi el circuit analògic del sensors.
- Potenciòmetre d'ajust del contrast del LCD.
- Circuit de reset del microcontrolador.
- Memòria de les 50 mostres consecutives a l'alarma per fase.
- Circuit analògic anivellador de senyal de 4...20mA a 0...5V.
- Circuit analògic convertidor corrent – tensió de 4...20mA a 1...5V.
- Transformadors de corrent amb un rang de 0...200A al primari i 4...20mA al secundari en valor eficaç.

5. SOLUCIÓ TECNOLÒGICA

5.1. Introducció

En aquest capítol s'expliquen les solucions adoptades per dotar l'equip amb la funcionalitat descrita en capítols anteriors i per complir amb les especificacions tècniques. També s'ha intentat explicar amb la major claredat possible el funcionament de cada una de les parts que componen aquest projecte, tant de *hardware* com de *software*. Al CD que s'adjunta amb aquest document s'inclou informació extra, com els *data sheets* de tots els elements i notes d'aplicació addicionals que s'han utilitzat en la realització del projecte.

La figura següent mostra el diagrama de blocs funcional de l'analyzador de la xarxa elèctrica trifàsica, amb tots els elements necessaris per al seu funcionament:

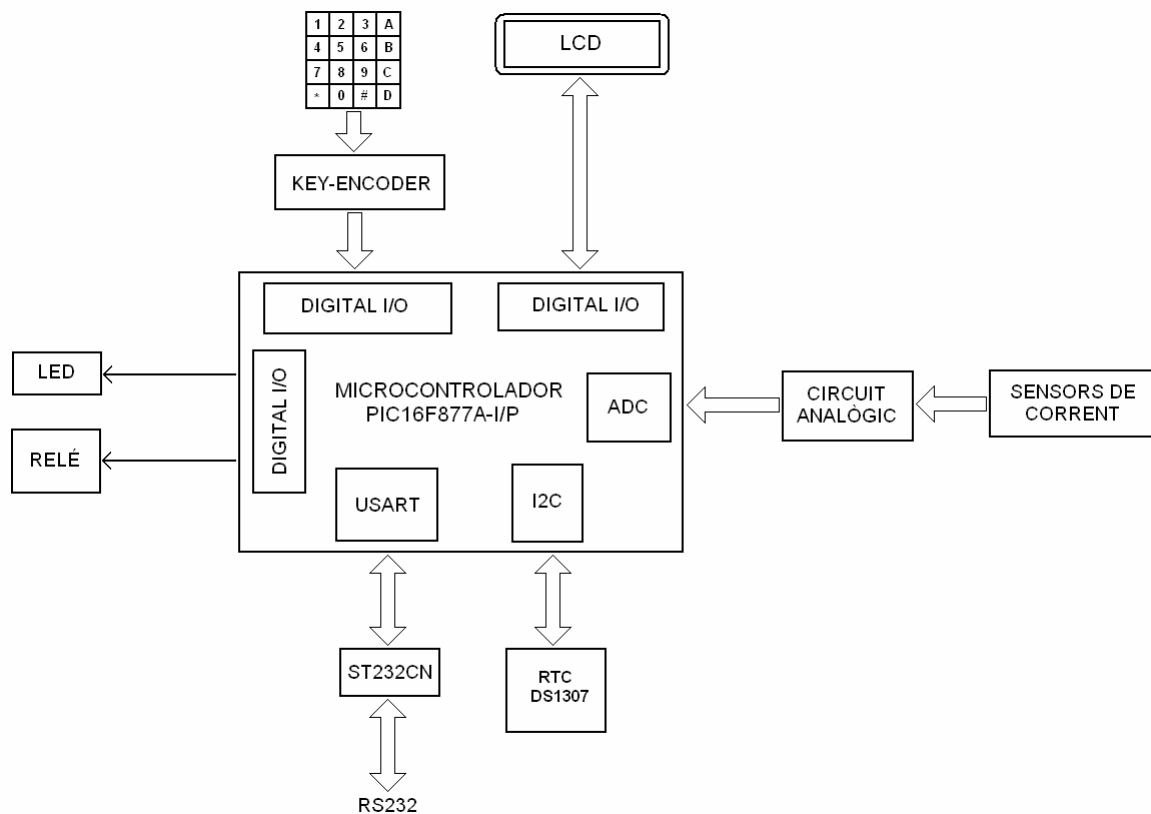


Figura 5.1. Diagrama de blocs de l'analyzador de la xarxa elèctrica trifàsica

(Font: Elaboració pròpia)

5.2. Sensor de corrent

El sensor de corrent és un dels elements clau del circuit. La seva selecció condicionarà la resta dels components del projecte, com per exemple, el microcontrolador, i també condicionarà l'estratègia del programa de control, si el sensor proporciona el valor instantani del corrent mesurat, el valor eficaç o el valor mig.

5.2.1. Tipus de sensors de corrent

Les tres tecnologies més habituals per mesurar el corrent elèctric (Koon, 2001) són:

- *Shunt* de corrent de baixa resistència
- Transformador de corrent (TC)
- Sensor d'efecte Hall

5.2.2. *Shunt* de corrent de baixa resistència

El *shunt* de corrent de baixa resistència és la solució més econòmica actualment disponible al mercat i ofereix una lectura senzilla amb excel·lent precisió. Es pot modelitzar el *shunt* com una resistència de baix valor òhmic que produeix en els seus extrems una caiguda de tensió proporcional al corrent que circula per sí mateix. Té l'inconvenient de que s'ha de tallar el circuit per connectar-hi el *shunt* i l'equip de mesura no pot estar massa lluny, per la caiguda de tensió a la línia de conducció, ja que el *shunt* proporciona com a resposta unes poques desenes de milivolts. A més la pèrdua de potència és elevada. Sovint s'utilitzen en la mesura de corrent DC.



Figura 5.2. Sensor de corrent *shunt*
(Font: Catàleg de *CIRCUTOR*)

5.2.3. Transformador de corrent

El transformador de corrent (TC) es basa en el principi del transformador per a convertir un corrent primari elevat en un corrent secundari més petit. Té l'avantatge que pot mesurar corrents elevats i consumeix poca potència, amb l'inconvenient que presenta la saturació del nucli ferromagnètic i el cicle de histèresi quan s'ha magnetitzat amb pics forts de corrent o amb presència de corrent DC en la línia conductora.



Figura 5.3. Transformador de corrent
(Font: Catàleg de *CIRCUTOR*)

5.2.4. Sensor d'efecte Hall

Aquests sensors es basen en el principi d'efecte Hall, descobert per E. H. Hall el 1879, que consisteix en l'aparició d'una diferència de potencial transversal en un conductor o semiconductor per el que circula corrent, quan hi ha un camp magnètic aplicat en direcció perpendicular al corrent (Pallás, 2003). Per mesurar el corrent, es disposa d'un sensor d'efecte Hall en l'entreferro d'un toroide obert en el que el pas del corrent genera un camp magnètic proporcional. De sensors d'efecte Hall per mesurar el corrent n'hi ha de dos tipus, d'anell obert (*open-loop*) i d'anell tancat (*closed-loop*). El segon ofereix millor precisió i rangs dinàmics més amplis però amb un cost més elevat; la majoria dels sensors d'efecte Hall que es troben en el mesuradors d'energia utilitzen el disseny d'anell obert per obtenir costos més baixos.

El sensors de corrent d'efecte Hall tenen una excel·lent resposta freqüencial i la capacitat de mesurar corrents elevats. En canvi presenten l'inconvenient de desviació de la mesura per l'efecte de la temperatura i el seu elevat cost.



Figura 5.4. Sensor de corrent d'efecte Hall per PCB
(Font: Catàleg de *LEM*)

5.2.5. Criteris de selecció dels sensors de corrent

Els criteris que s'han tingut en compte per la selecció dels sensors de corrent d'aquest projecte han estat els següents:

- **Mesura de corrent elevat:** Els sensors han de ser capaços de mesurar corrents elevats, de fins a 200A.
- **Seguretat:** El sensors no han de tenir contacte directe amb la instal·lació per minimitzar el risc d'electrocució. Aquest és un dels punts més importants pel fet de què s'han de mesurar corrents elevats i això limita molt la selecció del sensor al tipus de transformador de corrent, ja que els altres sensors (*shunt* i efecte Hall) per la seva naturalesa necessàriament han d'estar en contacte directe amb la instal·lació.
- **Facilitat d'instal·lació:** El sensor hauria de ser capaç de ser instal·lat sense haver d'obrir el circuit.

- **Senyal de resposta analògic de baixa potència:** El sensor ha de proporcionar una resposta de valor analògic proporcional al corrent mesurat i d'una magnitud de l'ordre de pocs volts o miliampers.

En els criteris de selecció no es té en compte altres paràmetres, com per exemple, una gran exactitud o el cost. L'objectiu del mesurador no és que sigui molt precís donat que ha de mesurar corrents elevats, i en principi el cost tampoc ha de ser un factor rellevant. Els més importants són els que tot just s'acaben d'explicar.

5.2.6. Els transformadors de corrent TP-420

Segons els criteris de selecció descrits en l'apartat anterior, s'ha trobat que els transformadors de corrent de la sèrie TP-420-23 de *CIRCUTOR* reuneixen les característiques. Aquests transformadors son de nucli partit, és a dir, el transformador té la capacitat de ser instal·lat en el cable sense haver-lo de desconnectar del circuit. A més ofereixen una resposta al secundari de 4...20mA proporcional al valor eficaç del corrent mesurat. I tenen un rang de mesura de corrent al primari entre 2,5A i 4.000A, segons el model.

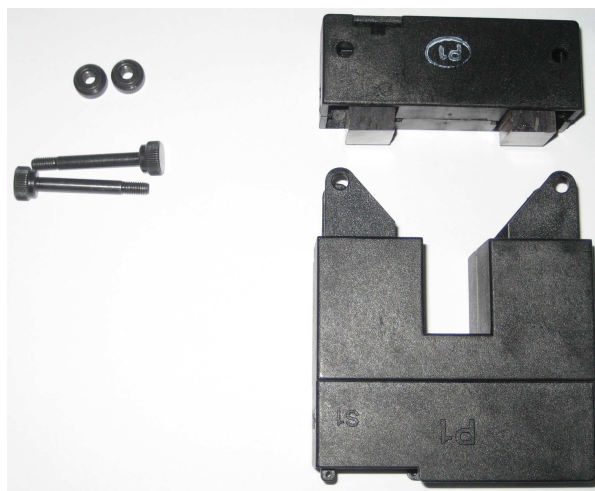


Figura 5.5. Transformador de corrent de nucli partit obert
(Font: Elaboració pròpia)

En concret el model M70216 és el que millor s'ajusta a les necessitats del projecte ja que pot mesurar corrents de fins a 200A. És un transformador rectangular, i la obertura per el cable té unes dimensions de 20 x 30 mm.



Figura 5.6. Transformador de corrent TP-420-23 de 200A

(Font: Elaboració pròpia)

Per alimentar el llaç de corrent del secundari es necessita una font externa. El valor màxim de la resistència de càrrega és funció de la tensió d'alimentació del llaç, tal i com es pot veure a la figura 5.7. En el circuit analògic que s'utilitzarà com a convertidor de corrent a tensió i com adaptador de nivells s'utilitzarà una càrrega de 250Ω . Amb aquest valor es poden utilitzar fonts d'alimentació a partir de 15V. En el muntatge del prototip s'utilitzaran fonts d'alimentació comercials de 24V. A la figura 5.8 es poden observar les característiques tècniques més rellevants del model escollit, el TP-420-23-M70216 de *CIRCUTOR*.

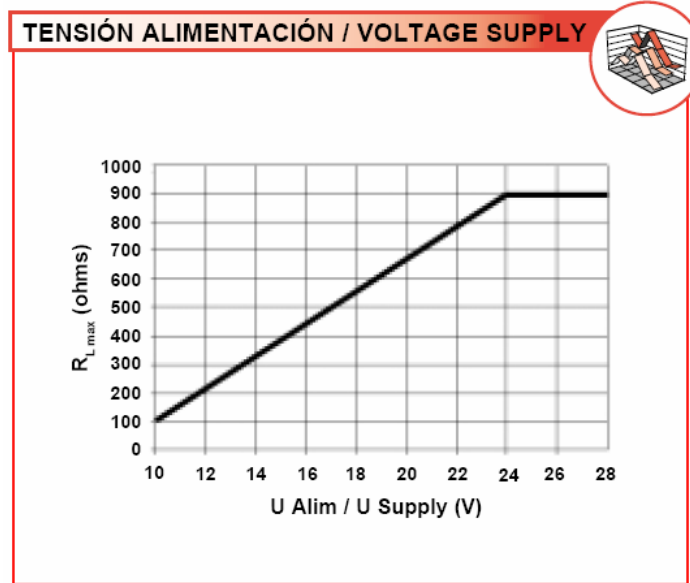


Figura 5.7. Relació R_{Lmax} / tensió d'alimentació del llaç de corrent
(Font: Catàleg de *CIRCUTOR*)


Current transformers. Measuring and protection			
	CARACTERÍSTICAS TÉCNICAS	TI-420, TP-420, TCM-420, TCB-420	TECHNICAL CHARACTERISTICS
	Condiciones de trabajo		Operating conditions
	Intervalo de clase	0 / +50 °C	Class range
	Temperatura de trabajo	-10 / +50 °C	Operating temperature
	Tensión más elevada para el material (U _m)	0,72 kV c.a. / a.c.	Highest voltage for equipment (U _m)
	Circuito de medida		Measurement circuit
	Primario pasante		Passing bar primary
	Corriente primaria	Según tipos / Depending on types	Primary current
	Respuesta en frecuencias lineal	50...60 Hz	Frequency linear response
Corriente secundaria	4...20 mA c.c. / d.c.	Secondary current	
Precisión (entre el 5...110 % I _n)	± 1,5 I_n lectura / of reading	Accuracy (within the 5...110 % I _n)	
Sobrecargas (a temperatura ambiente)	1,5 I_n permanentemente / permanently	Overloads (at operation temperature)	
Normas	IEC 44-1, UNE 21 088-1, IEC 664, VDE 0110, VDE 0414, UL 94, IEC 1010-1, EN 61010-1	Standards	

Figura 5.8. Característiques tècniques del TC model *TP-420-23-M70216*
(Font: Catàleg de *CIRCUTOR*)

5.3. Microcontrolador

L'existència en el mercat de potents microcontroladors amb moltes funcionalitats incorporades justifica la utilització d'aquests per sobre dels microprocessadors tradicionals. Els microcontroladors inclouen blocs interns que implementen moltes funcions *hardware*, com ara funcions de connectivitat (USART, I²C™, CAN, Ethernet), memòria de programa (Flash EEPROM, ROM), memòria de dades (SRAM), entrades i sortides digitals paral·lel, entrades analògiques (ADC), modulació d'ample de polsos (PWM), temporitzadors... tot plegat minimitza la necessitat d'utilitzar molts circuits integrats externs i es simplifica enormement la realització de projectes i prototips. Per aquesta raó en aquest projecte s'utilitzarà un microcontrolador.

5.3.1. Criteris de selecció del microcontrolador

El microcontrolador és l'altre component clau del projecte. Quan se sap quins sensors de corrent són els adequats per l'aplicació, aleshores ja es pot triar el microcontrolador. Aquest és el criteri més important a l'hora de d'escollir-lo, tal i com es justifica a continuació: Si els sensors ofereixen el valor *instantani* del corrent mesurat, reproduint la forma d'ona del corrent del primari en el debarat del secundari, aleshores l'exigència temporal del microcontrolador és molt elevada. Per exemple, si es mesura el corrent de la xarxa elèctrica trifàsica que treballa a 50Hz s'haurien de prendre com a mínim 20 mostres per cicle i per fase, és a dir, el microcontrolador hauria de ser capaç d'obtenir mostres del corrent cada 300µs aprox. (per les tres fases), emmagatzemar-les en memòria, trobar quin és el valor màxim i dividir-lo entre $\sqrt{2}$ per obtenir el valor eficaç. Això sense comptar tota la feina complementària que hauria de realitzar entre mostra i mostra, com la de gestionar el port de comunicacions RS232, el teclat, el *display* LCD i la resta del circuit.

Com que els sensors que s'han seleccionat ofereixen una resposta en valor eficaç proporcional al corrent mesurat, aleshores la restricció temporal no és tan elevada. De totes maneres es mantindrà un fort ritme de vigilància sobre els consums per poder detectar qualsevol anomalia de la instal·lació. Es pensa que mantenint un temps de mostreig de 2ms per fase l'analitzador serà prou ràpid per detectar qualsevol incidència.

Per tant els criteris de selecció que s'han tingut en compte per trobar el microcontrolador adequat han estat els següents:

- **Processament de dades:** S'entén per processament de dades la capacitat del microcontrolador per realitzar les operacions de baix nivell amb la màxima rapidesa i eficiència possible. El requeriment per el tractament de les dades de l'analitzador de la xarxa elèctrica trifàsica no és molt exigent. Per tant s'utilitzarà un microcontrolador de 8 bits. En quant a la velocitat de processament existeix certa restricció temporal, per tant s'utilitzarà un microcontrolador capaç de treballar fins a 20 MHz.
- **Capacitat de Memòria:** S'entén per capacitat de memòria la quantitat de dades que el sistema pot emmagatzemar. El microcontrolador ha de tenir la suficient capacitat per poder emmagatzemar les dades temporals, per exemple, les 50 mostres per fase en cas de què es produís una incidència (memòria volàtil) i també la capacitat per emmagatzemar el programa de l'aplicació (memòria no volàtil).
- **Tipus de memòria:** Es necessita disposar d'una memòria de programa que es pugui programar i esborrar varies vegades per poder realitzar proves, per tant, s'utilitzaran memòries del tipus EEPROM o FLASH que es poden esborrar elèctricament. A més, tenen la capacitat de ser programades "en circuit", es a dir, que no cal extreure el circuit integrat de la placa per tal de realitzar modificacions. Per a emmagatzemar les dades temporals serà necessari que el microcontrolador inclogui uns quants *bytes* de memòria tipus RAM.
- **Entrades i Sortides Analògiques i Digitals:** Es consideren els requeriments d'entrada/sortida analògica i digital com la capacitat per interactuar amb els diferents elements que formen part de l'analitzador de la xarxa elèctrica, com ara el visualitzador LCD, díodes LED o un *key-encoder*. Les E/S digitals del microcontrolador han de ser prou potents com per interactuar amb aquests elements sense la necessitat d'utilitzar de circuits integrats externs, que augmentarien el cost i la grandària del prototip. Un altre aspecte a tenir en compte és el *nombre de ports* dedicats a l'entrada/sortida, per tal de implementar totes les funcionalitats previstes

amb un únic microcontrolador. A més és imprescindible que el microcontrolador incorpori varis canals d'entrades analògiques, com a mínim tres, i un ADC, per tal de poder llegir la informació que proporcionaran els sensors de corrent.

- **Comunicacions:** Es consideren els requeriments de comunicacions com la capacitat per interactuar amb els diferents elements del circuit i/o amb altres sistemes, com pot ser un PC. És necessari que el microcontrolador disposi de comunicació I²C™ per el RTC i un mòdul USART per implementar la connectivitat RS232.
- **Facilitat de desenvolupament:** Al voltant del microcontrolador han d'existir eines que facilitin la seva programació amb llenguatges de programació més fàcils d'aprendre que l'assemblador, com ara els llenguatges d'alt nivell. S'utilitzarà el llenguatge C de CCS. D'aquesta manera el desenvolupament del programa es farà en amb menys temps i per tant amb menor cost.

5.3.2. Microcontrolador PIC16F877A-I/P de *Microchip*

Per aquest projecte s'ha triat el PIC16F877A-I/P del fabricant *Microchip*, que reuneix les condicions descrites anteriorment. El fabricant *Microchip* s'ha consolidat com un dels líders mundials de vendes de microcontroladors de 8 bits (DS39630F *Microchip*, 2009). Amb més de 400 models diferents de microcontroladors en cartera (250 models de microcontroladors PIC® de 8 bits), *Microchip* ofereix moltes prestacions de *hardware* integrades en un sol xip, eines de desenvolupament de projectes fàcils d'utilitzar, una molt bona i extensa documentació dels dispositius (disponibles de forma gratuïta), i aplicacions per als dissenyadors d'arreu del món recolzats per la extensa xarxa de distribució.

Es diu que els microcontroladors PIC® de *Microchip* han esdevingut un estàndard per a molts usuaris, ja que des del 1990 fins el 2009 s'han venut més de 6.000.000.000 d'unitats, gairebé un xip per cada habitant del planeta Terra.

El microcontrolador PIC16F877A-I/P és un dispositiu de 40 pins en encapsulat PDIP (de l'anglès *Plastic Dual In-Line Package*), i pertany a la família de gamma mitja, la PIC16F87XA, amb les característiques funcionals següents:

- Són dispositius amb arquitectura HARDVARD. A diferència de l'arquitectura VON NEUMANN, l'arquitectura HARDVARD té dos bussos separats internament: un per a la memòria de dades (8 bits) i un altre per a la memòria de programa (14 bits en la gamma mitja), millorant notablement el rendiment.
- Incorporen un microprocessador RISC d'altas prestacions, amb un joc d'instruccions reduït de 35 instruccions.
- L'accés a memòria de programa suporta una estructura *pipe-line*, és a dir, durant l'execució d'una instrucció ja s'està accedint a la memòria de programa per obtenir la següent instrucció que s'ha d'executar a continuació.
- Totes les instruccions només ocupen una posició de memòria (instruccions de mida de 14 bits).
- Ortogonalitat de les instruccions: són vàlids tots els modes de d'adreçament per a totes les instruccions (indirecte, directe i relatiu) amb tots els registres disponibles i amb la memòria. Això augmenta l'eficiència del codi degut a la gran compactació dels programes.
- Execució del codi a molt alta velocitat: 5 milions d'instruccions per segon a una freqüència de rellotge de 20 MHz. Totes les instruccions s'executen en un sol cicle d'instrucció (4 cicles de rellotge), excepte les instruccions de salt, que necessiten dos cicles.
- Eines de desenvolupament de *software* i *hardware* molt abundants i de baix cost. L'entorn de desenvolupament integrat MPLAB[®] IDE és completament gratuït i es pot aconseguir fàcilment des de la pàgina web de *Microchip*. Permet programar, gravar el programa en el xip, fer la simulació del codi i la depuració *en circuit*.



Figura 5.9. Microcontrolador PIC16F877A-I/P de Microchip
(Font: Elaboració pròpia)

A més de totes les funcionalitats que s'han descrit anteriorment, aquest microcontrolador inclou les següents característiques tècniques:

- Memòria de programa CMOS FLASH de fins a 8K x 14 paraules.
- Memòria de dades RAM de fins a 368 x 8 bytes.
- Memòria de dades CMOS EEPROM de fins a 256 x 8 bytes.
- Fins a 15 fonts d'interrupció (1 interrupció externa).
- Pila d'*stack* de 8 nivells.
- 33 línies d'E/S digitals (fins a 25mA de corrent Sink/Source per línia).
- Freqüència de rellotge de fins a 20MHz.
- 1 convertidor A/D de 10-bits i 8 canals d'entrades analògiques.
- 2 mòduls CCP (Captura – Comparador – PWM).
- 1 mòdul de comunicacions I²C™.
- 1 mòdul de comunicacions USART.
- 1 temporitzador de 8 bits i 2 temporitzadors de 16 bits.

La designació I/P fa referència a que és de la sèrie industrial, per tant pot treballar a la màxima freqüència de rellotge (fins a 20MHz) i pot suportar temperatures des de -40°C fins a +85°C.

L'arquitectura interna del microcontrolador a nivell de diagrama de blocs, on es descriuen totes les funcionalitats *hardware* incorporades, es mostra a la figura següent:

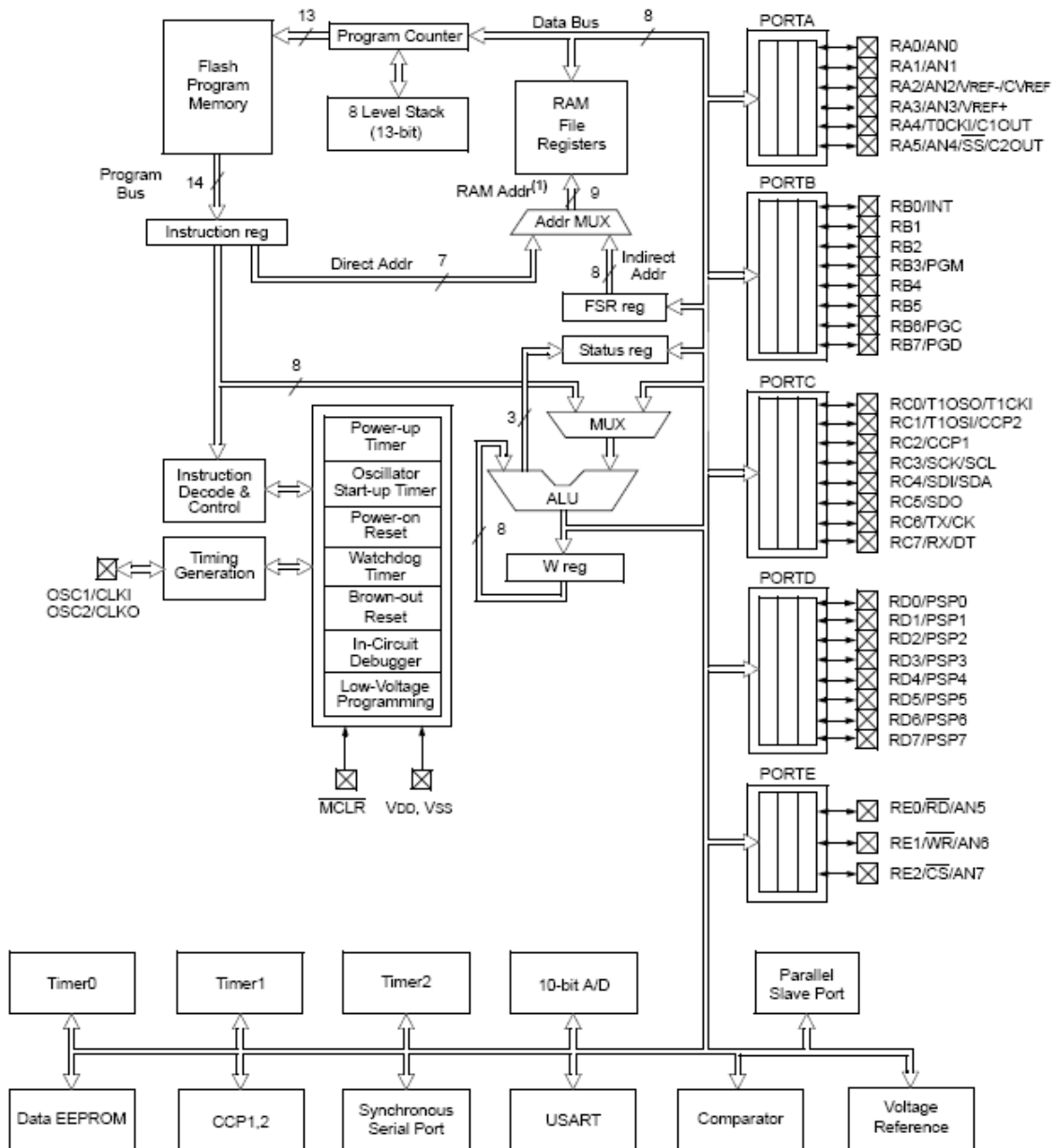


Figura 5.10. Arquitectura interna del microcontrolador PIC16F877A

(Font: Data Sheet PIC16F87XA)

5.4. Circuit analògic

5.4.1. Convertidor I/V

El circuit analògic és el nexa d'unió entre el sensor de corrent i el microcontrolador. Les entrades analògiques només poden llegir valors de tensió entre 0 i 5V (màxim 5,5V). Si el senyal aplicat excedís aquests límits es destruiria el microcontrolador. No es poden aplicar en cap cas tensions negatives. Tenint present aquestes limitacions i que el senyal que proporcionen els sensors de corrent és de 4...20mA es van dissenyar dos circuits: un circuit més senzill com convertidor I/V (de corrent a tensió) i un altre més complex, que implementa el circuit anterior més un adaptador de nivells. El circuit de la figura següent compleix amb el primer requisit: és un convertidor I/V.

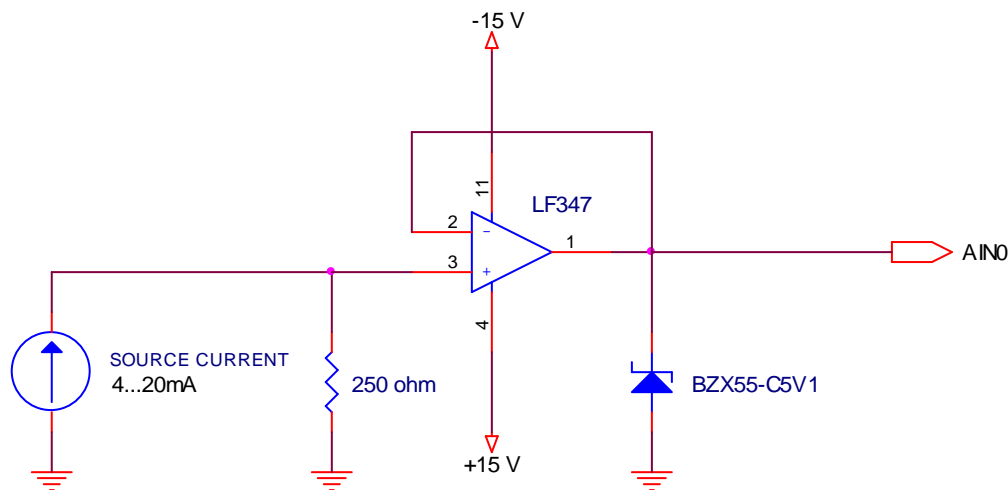


Figura 5.11. Circuit convertidor corrent – tensió (I/V)

(Font: Elaboració pròpia)

La font de corrent (*source current*) representa el model del senyal que proporcionen els transformadors de corrent. El circuit operacional s'utilitza com a adaptador d'impedàncies en un muntatge de seguidor de tensió i el díode zener protegeix les entrades analògiques del microcontrolador, limitant el senyal aplicat a 5,1V en el llindar de tensió màxim superior.

Quan el corrent del llaç circula per la resistència de 250Ω es produeix una caiguda de tensió en els seus borns, segons la Llei d'Ohm:

$$V_{R_{\min}} = I_{\min} \cdot R = 0,004 \cdot 250 = 1V \quad (1)$$

$$V_{R_{\max}} = I_{\max} \cdot R = 0,020 \cdot 250 = 5V \quad (2)$$

Segons les equacions (1) i (2) la caiguda de tensió a la resistència per els corrents mínim i màxim del llaç són valors que el microcontrolador podria llegir perquè es troben dins els límits admissibles, però té l'inconvenient que així no es pot aprofitar tot el rang del ADC, entre 0 i 5V. En canvi aquest circuit té l'avantatge de què, programat adequadament, el microcontrolador podria detectar la ruptura del fil o el mal funcionament d'un dels sensors.

En les figures 5.12 i 5.13 es poden veure l'esquema electrònic i el circuit convertidor de corrent – tensió muntat per les tres fases, respectivament.

El connector de deu vies permet la connectivitat amb el circuit principal. Inclou totes les tensions necessàries per el funcionament dels OP AMP, una alimentació simètrica de $\pm 15V$. La tensió de 5V no s'utilitza en aquest circuit, però s'ha dissenyat d'aquesta manera per afavorir la màxima compatibilitat amb l'altre circuit, l'adaptador de nivells. En els tres connectors de dues vies es connecten els llaços de corrent que proporcionen els sensors.

Perquè el circuit convertidor corrent - tensió funcioni amb exactitud les resistències han de ser de molt bona qualitat, de precisió. Les resistències del circuit de la figura 5.13 són de pel·lícula metàl·lica, amb una tolerància de $\pm 1\%$ i $1/4$ de watt de potència. Per obtenir l'exactitud màxima possible es van comprar 20 unitats de resistències de 250Ω i es van mesurar amb un òhmmetre. Les resistències amb el millor valor òhmic (les que més s'ajustaven a 250Ω) són les que es van utilitzar en la fabricació del circuit.

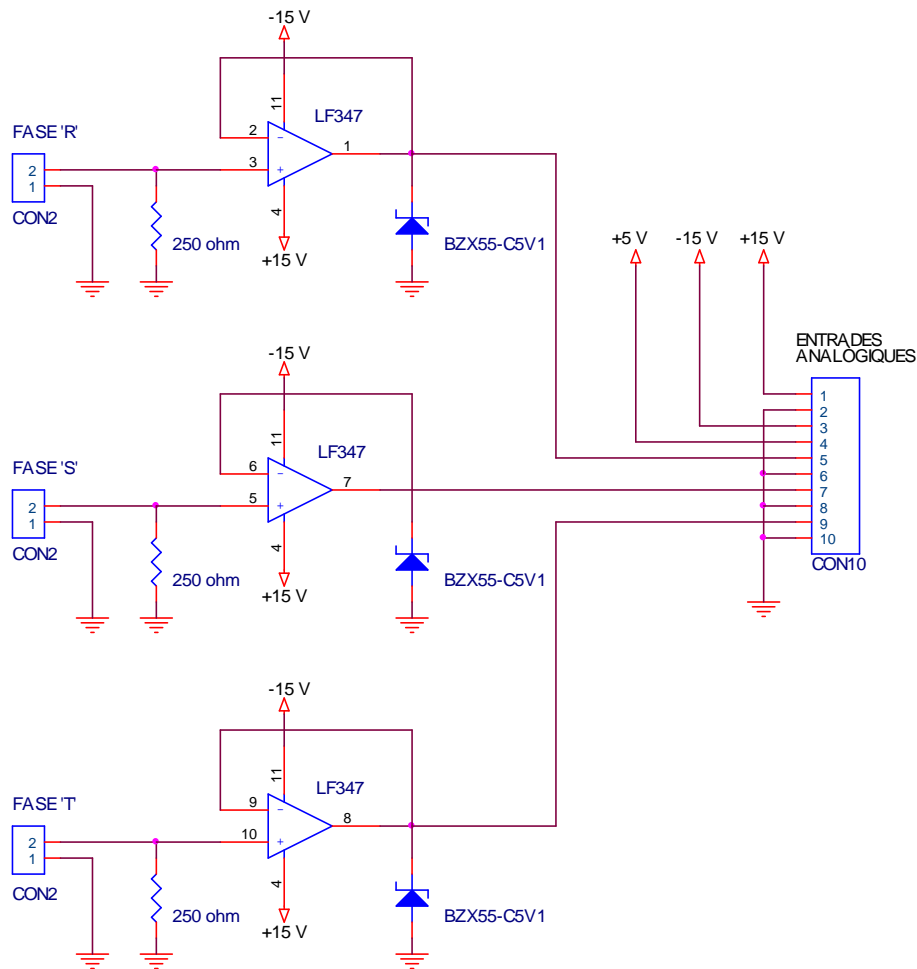


Figura 5.12. Esquema convertidor I/V de tres fases
(Font: Elaboració pròpia)

S'ha utilitzat un OP AMP de reconeguda qualitat i fiabilitat, el LF347, que inclou quatre OP AMP construïts amb transistors JFET, cosa que assegura una excel·lent impedància d'entrada, molt elevada, de l'ordre de $10^{12}\Omega$ segons el *data sheet* del fabricant. El seu gran ample de banda (de 3 MHz) i el ràpid *slew rate* d'aquest OP AMP (de 13 V/ μ s) el fan ideal per aquesta aplicació, on es poden produir canvis sobtats en el senyal aplicat.

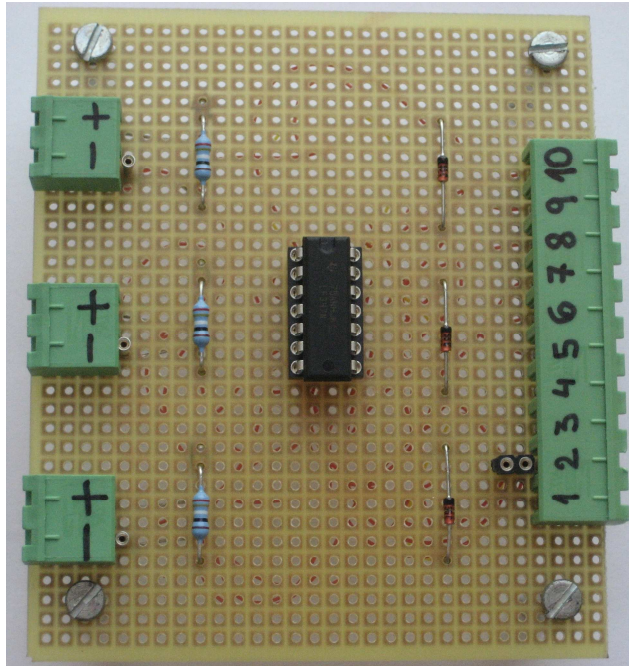


Figura 5.13. Circuit convertidor I/V de tres fases
(Font: Elaboració pròpia)

La fotografia de la figura anterior permet visualitzar l'estat final del circuit muntat. A l'esquerra de la figura destaquen els connectors per connectar-hi el llaç de corrent de 4...20mA dels transformadors de corrent i les resistències de 250Ω de precisió. Al centre es pot veure el OP AMP, el LF347 muntat sobre un sòcol de *wire-wrapping*. A continuació els díodes zener, per limitar la tensió màxima aplicada a les entrades analògiques i finalment el connector de deu vies per connectar-hi el circuit principal. Val a dir que tots els connectors ofereixen una compatibilitat total amb l'altre circuit, l'adaptador de nivells.

5.4.2. Adaptador de nivells

Per aprofitar tot el rang de les entrades analògiques s'ha dissenyat un circuit adaptador de nivells, que juntament amb el convertidor corrent - tensió formen el circuit analògic principal d'enllaç entre els sensors de corrent i el microcontrolador.

A la figura següent es pot veure quina és la funció de transferència $V_o = f(V_i)$ que ha de complir el circuit adaptador de nivells:

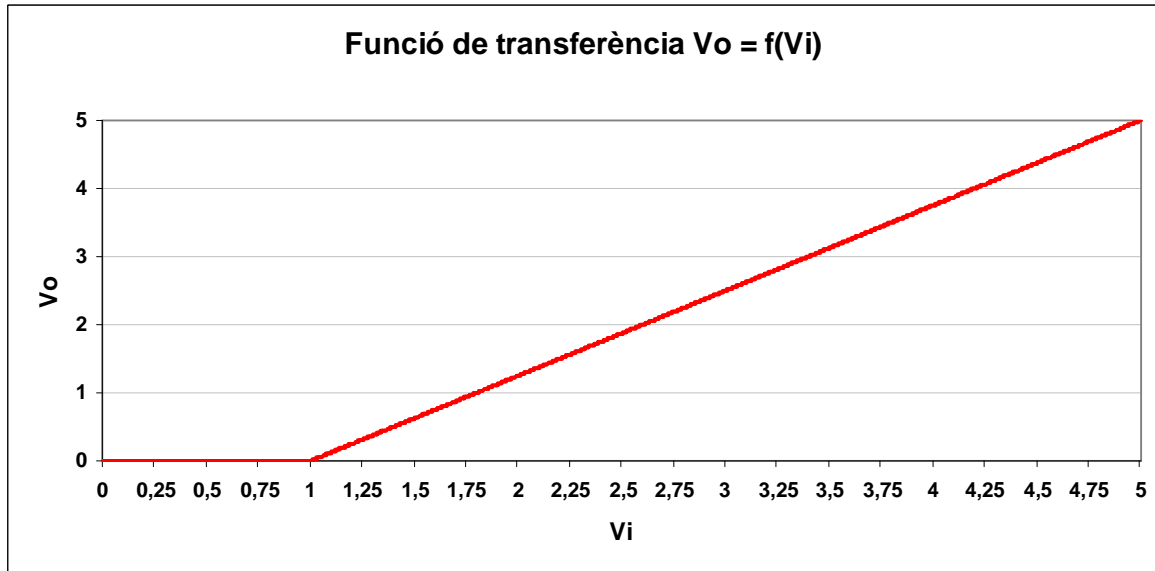


Figura 5.14. Funció de transferència $V_o = f(V_i)$

(Font: Elaboració pròpia)

Per trobar el circuit s'ha de resoldre una equació de primer ordre, del tipus:

$$V_o = a \cdot V_i + b, \quad (3)$$

on a és el pendent de la recta i b és el punt on la funció talla l'eix de les ordenades.

Per trobar aquests punts i resoldre les incògnites aplicarem els valors coneguts de a i de b en un sistema de dues equacions i dues incògnites:

$$\left. \begin{array}{l} V_i = 1V \rightarrow V_o = 0V \\ V_i = 5V \rightarrow V_o = 5V \end{array} \right\} \begin{array}{l} 0 = a + b \\ 5 = a \cdot 5 + b \end{array} \quad (4)$$

Si es resten les dues equacions s'obté el següent resultat:

$$-5 = -4 \cdot a$$

A partir d'aquí conèixer el valor dels termes és trivial:

$$a = \frac{-5}{-4} = 1,25$$

$$b = -a = -1,25$$

Finalment, la equació que resol el circuit és la següent:

$$V_o = 1,25 \cdot V_i - 1,25 \cdot 1 \tag{5}$$

Si s'analitza amb més detall els termes de la equació es veu que el circuit que resol la equació és un circuit sumador – restador amb OP AMP. El terme *unitat* que multiplica el valor de -1,25 aplicat al circuit restador representa el valor de la tensió de referència. L'esquema del circuit electrònic sumador - restador es pot veure a la figura següent:

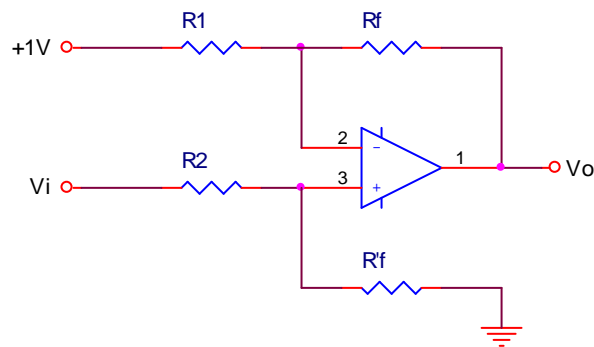


Figura 5.15. Circuit sumador – restador
(Font: Elaboració pròpia)

La equació de transferència del circuit de la figura anterior és la següent:

$$V_o = -\underbrace{\frac{R_f}{R'_f}}_{1,25} \cdot 1 + \underbrace{\frac{R_f}{R'_f}}_{1,25} \cdot V_i \quad (6)$$

Si s'estableix que $R_f = R'_f = 100\text{K}\Omega$, aleshores $R_1 = R_2$:

$$\frac{R_f}{R_1} = \frac{R'_f}{R_2} = 1,25$$

$$R_1 = R_2 = \frac{100\text{K}\Omega}{1,25} = 80\text{K}\Omega$$

A la figura següent es pot veure l'esquema del circuit adaptador de nivells, només per una fase. El circuit sencer es pot veure en el document dedicat als esquemes electrònics, inclòs en aquest llibre.

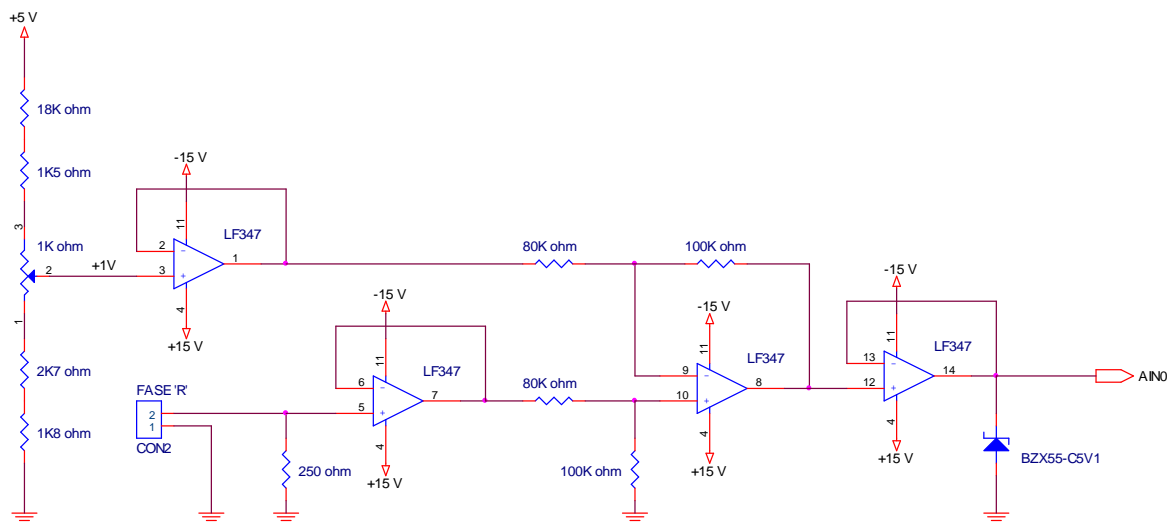


Figura 5.16. Esquema adaptador de nivells per una fase

(Font: Elaboració pròpia)

Finalment s'ha dissenyat el circuit amb quatre OP AMP. El xip LF347 inclou els quatre OP AMP amb l'avantatge d'utilitzar només dues línies per alimentar tots quatre. Un OP AMP s'utilitza en un muntatge de seguidor de tensió per el convertidor corrent - tensió, un altre per implementar el circuit sumador - restador i dos més per aïllar elèctricament les diferents parts del circuit: la tensió de referència d'1V de la resta del circuit i la sortida del circuit sumador - restador de les entrades analògiques del microcontrolador.

Per dissenyar el circuit que proporciona la tensió de referència d'1V es va pensar en un divisor de tensió on es pogués ajustar la tensió de sortida, tal i com es pot veure a la figura 5.17. Es va imposar un valor de corrent per el divisor de tensió de 0,2mA perquè les resistències dissipessin la mínima potència possible.

La connexió al circuit adaptador de nivells es faria a través d'un seguidor de tensió amb OP AMP, com s'ha explicat anteriorment. S'imposen a més les següents condicions:

$$R_{TOTAL} = \frac{5V}{0,2mA} = 25K\Omega \quad \begin{cases} V_{ref_{min}} = 0,9V \\ V_{ref_{max}} = 1,1V \end{cases}$$

Quan es produeix el cas que $V_{R3} = V_{ref_{min}} = 0,9V$ aleshores el valor de R3 és:

$$V_{ref_{min}} = 0,9 = 0,1 \cdot R3 \quad R3 = \frac{0,9V}{0,2mA} = 4K5\Omega$$

I quan es produeix el cas que $V_{R2} = V_{ref_{max}} = 1,1V$ aleshores el valor de R2 és:

$$V_{ref_{max}} = 1,1 = 0,2 \cdot (R2 + R3) \quad R2 + R3 = \frac{1,1V}{0,2mA}$$

$$R2 = \frac{1,1V}{0,2mA} - R3 = 5,5 - 4,5 = 1K\Omega$$

Finalment es calcula el valor de R1:

$$R1 = R_{TOTAL} - R2 - R3 = 25K - 1K - 4,5K = 19,5K\Omega$$

Per aconseguir un valor de tensió de referència d'1V exactes s'han utilitzat varies resistències de valors normalitzats per aconseguir els valors de R1 i R3, juntament amb un potenciòmetre de precisió de deu voltes d'1K Ω per la R2. A la figura 5.18 es pot veure el muntatge final del circuit adaptador de nivells per les tres fases.

L'acció sobre el potenciòmetre R2 permet ajustar la tensió de referència amb molta exactitud, de l'ordre de milivolts. Aquest és l'únic ajust que s'ha de fer en el circuit.

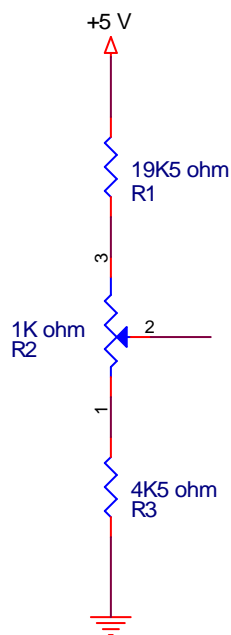


Figura 5.17. Divisor de tensió per la tensió de referència
(Font: Elaboració pròpia)

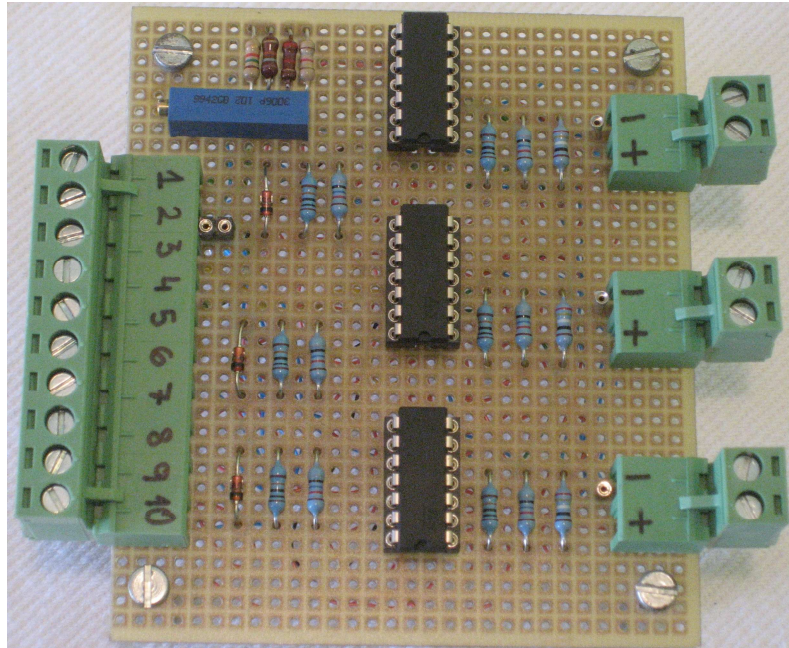


Figura 5.18. Circuit adaptador de nivells de tres fases
(Font: Elaboració pròpia)

5.4.3. Linealitat del circuit adaptador de nivells

Es va provar la linealitat del circuit analògic adaptador de nivells injectant a cada fase un senyal de corrent de 4...20mA amb un generador de corrent. La taula 5.1 resumeix el resultat de la prova:

A_i (mA)	V_o teòric (V)	V_o 'R' (V)	Error	V_o 'S' (V)	Error	V_o 'T' (V)	Error
4	0,000	0,000	0,00%	0,000	0,00%	0,000	0,00%
5	0,313	0,310	0,96%	0,309	1,28%	0,313	0,00%
6	0,625	0,619	0,96%	0,619	0,96%	0,621	0,64%
7	0,938	0,927	1,17%	0,928	1,07%	0,930	0,85%
8	1,250	1,235	1,20%	1,237	1,04%	1,239	0,88%
9	1,563	1,545	1,15%	1,546	1,09%	1,548	0,96%
10	1,875	1,854	1,12%	1,855	1,07%	1,857	0,96%
11	2,188	2,162	1,19%	2,164	1,10%	2,165	1,05%
12	2,500	2,470	1,20%	2,473	1,08%	2,474	1,04%
13	2,813	2,780	1,17%	2,782	1,10%	2,783	1,07%
14	3,125	3,089	1,15%	3,091	1,09%	3,092	1,06%
15	3,438	3,397	1,19%	3,400	1,11%	3,401	1,08%
16	3,750	3,705	1,20%	3,704	1,23%	3,709	1,09%
17	4,063	4,015	1,18%	4,018	1,11%	4,018	1,11%
18	4,375	4,324	1,17%	4,327	1,10%	4,327	1,10%
19	4,688	4,633	1,17%	4,636	1,11%	4,636	1,11%
20	5,000	4,940	1,20%	4,945	1,10%	4,945	1,10%

Taula 5.1. Resultats proves linealitat circuit adaptador de nivells
(Font: Elaboració pròpia)

5.5. Mòdul convertidor A/D del microcontrolador PIC16F877A

El microcontrolador PIC16F877A inclou un mòdul de conversió de senyal analògic a senyal digital. Els mòduls ADC que utilitza *Microchip* fan un mostreig i retenció (*sample & hold*) amb un condensador i després utilitza el mòdul de conversió. El mòdul de conversió A/D és del tipus d'aproximacions successives. Aquest tipus de convertidors s'utilitzen en aplicacions on es necessiten velocitats de conversió elevades.

El mòdul de conversió es caracteritza per els següents paràmetres:

- Rang d'entrada: **0 a 5V**.
- Nombre de Bits: **10 bits**.
- Resolució: **4,8mV** si $V_{REF+} = V_{DD}$ i $V_{REF-} = V_{SS}$.
- Temps de conversió: **22µs** aproximadament.

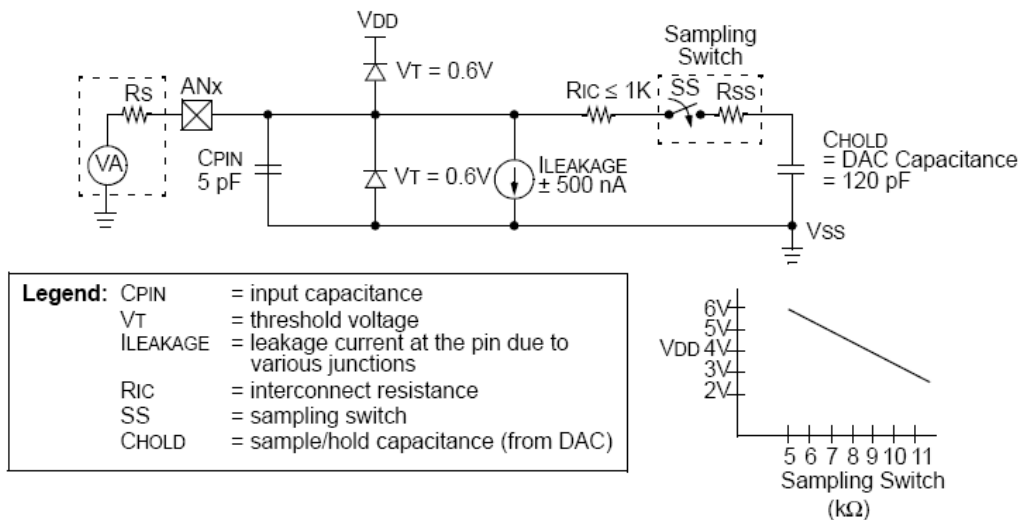


Figura 5.19. Model de les entrades analògiques

(Font: Data Sheet PIC16F87XA)

El paràmetre de *temps de conversió* en realitat és la suma del temps d'adquisició pròpiament dit (T_{ACQ}) més el temps de conversió (T_{AD}).

El T_{ACQ} és el temps necessari perquè es carregui el condensador de retenció (C_{HOLD}) amb la tensió d'entrada. Aquest temps és més o menys constant, de l'ordre de $20\mu s$. Cada cop que es selecciona un nou canal d'entrada s'ha d'esperar aquest temps.

El T_{AD} és el temps necessari per obtenir el valor digital de la tensió analògica d'entrada i depèn de la freqüència del rellotge de la CPU. Per una freqüència de rellotge de 20MHz T_{AD} té un valor de $1,6\mu s$.

La figura 5.19 mostra el model de les entrades analògiques dels microcontroladors de la família PIC16F87XA i la figura 5.20 el diagrama de blocs del convertidor A/D.

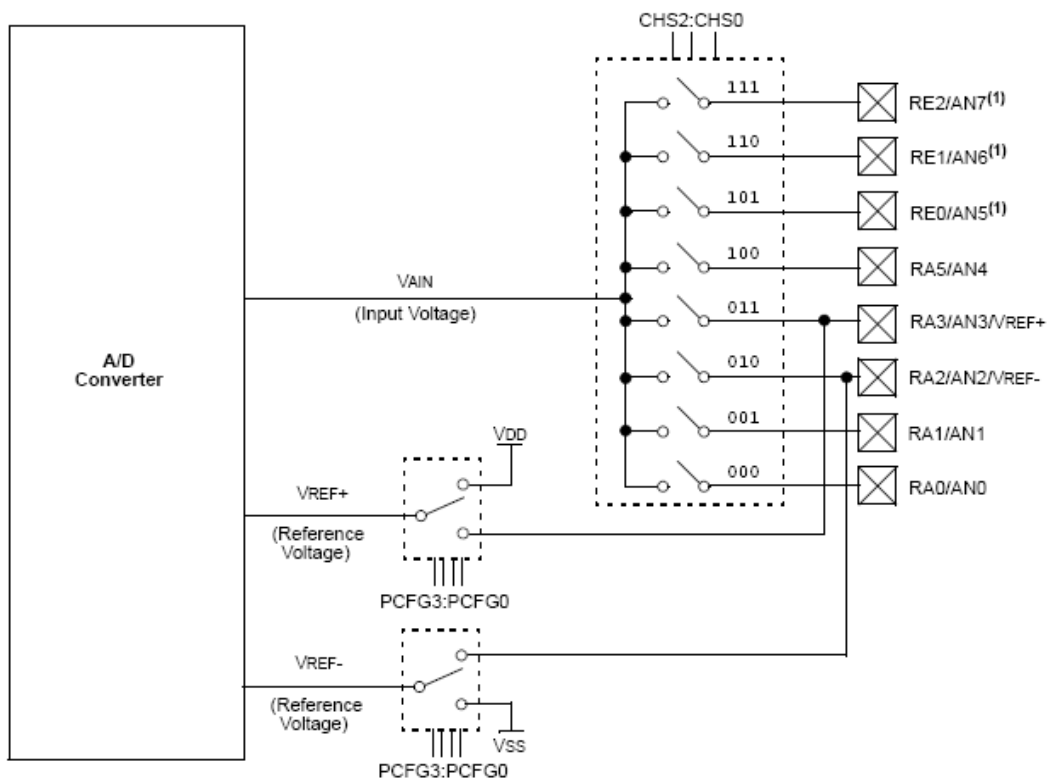


Figura 5.20. Diagrama de blocs del convertidor A/D
(Font: Data Sheet PIC16F87XA)

L'acció sobre els registres del mòdul convertidor A/D permeten la selecció del canal d'entrada, la configuració de les entrades del PORT A (com entrades analògiques o com E/S digitals), l'origen de les tensions de referència, la font del rellotge del convertidor, el temps d'adquisició i la representació del format del resultat.

Per realitzar aquestes operacions s'han de configurar adequadament els registres ADCON0 i ADCON1. El resultat de la conversió es diposita en els registres ADRESH (*byte* alt) i ADRESL (*byte* baix).

5.6. Mòduls CCP2 i TIMER1: Automatització del convertidor A/D

El microcontrolador PIC16F877A permet engegar automàticament les conversions A/D amb l'ajut dels mòduls interns CCP2 i TIMER1. Configurats adequadament aquests mòduls permeten obtenir valors digitals de senyals analògics cada cert temps, l'anomena't temps de mostreig (T_m).

Els mòduls CCP1 i CCP2 (acrònim de *Comparador–Captura–PWM*) permeten realitzar diverses funcions utilitzant els temporitzadors:

- Comparador: Compara el valor del temporitzador amb el valor d'un registre i es provoca una reacció en el PIC[®].
- Captura: Obté el valor del temporitzador en un moment donat, fixat per l'acció d'un terminal del PIC[®].
- PWM: Es genera un senyal modulad en amplitud de pols (PWM, de l'anglès *Pulse Width Modulation* o modulació de l'amplada del pols).

Cada mòdul CCP té associat un registre de 16 bits que es pot utilitzar de tres maneres diferents:

- Registre de 16 bits per a capturar el valor del temporitzador quan es produeix un esdeveniment (mode CAPTURA).
- Registre de 16 bits per comparar el seu valor amb el valor del temporitzador TIMER1, produint un esdeveniment quan s'assoleix el valor contingut en aquest registre (mode COMPARADOR).
- Registre de 10 bits per el cicle de treball d'un senyal PWM.

El microcontrolador PIC16F877A també disposa de tres temporitzadors o *timers* integrats en mòduls amb diverses funcionalitats *hardware*: TIMER0 (8 bits), TIMER1 i TIMER2 (ambdós de 16 bits). Aquests mòduls permeten realitzar comptes tant d'esdeveniments interns com externs. Quan la compta és interna es parla de *temporitzador* i quan la compta és externa es parla de *comptador*.

El mòdul TIMER1 té les següents característiques:

- Treballa amb 16 bits (amb dos registres de 8 bits: TMR1H i TMR1L).
- Ambdós registres es poden llegir i escriure.
- Interrupció per desbordament de 0xFFFF a 0x0000.
- Reset del TIMER1 per comparació amb els mòduls CCPx.

Per automatitzar les conversions A/D cada cert temps s'ha de configurar el mòdul CCP2 en mode COMPARACIÓ. Configurats en aquest mode, tots dos mòduls CCPx fan un *reset* del TIMER1, però a més el mòdul CCP2 engega alhora una conversió A/D, si aquesta opció es troba habilitada.

Si el mòdul CCP2 es configura d'aquesta manera i el parell de registres CCPR2H-CCPR2L es carreguen amb el temps de mostreig desitjat, cada vegada que el TIMER1 assolixi aquest valor s'iniciarà una conversió A/D. Així s'automatitzen les conversions A/D en el microcontrolador PIC16F877A.

En aquest projecte s'utilitza aquest mètode per automatitzar les conversions A/D. Si es programa el microcontrolador en el llenguatge C de CCS el codi d'aquesta part del programa és molt senzill:

```
// Configuració del mòdul CCP2 per un Tm del ADC de 2 ms (500 Hz):
setup_ccp2(CCP_COMPARE_RESET_TIMER);
CCP_2_LOW = 0x05;
CCP_2_HIGH = 0x0D;

// Configuració del TIMER1:
setup_timer_1(T1_INTERNAL | T1_DIV_BY_1);
set_timer1(0);
```

Quan la conversió A/D ha finalitzat es produeix una interrupció per EOC o fi de la conversió. La rutina de servei de la interrupció (anomenada per l'acrònim RSI) llegeix el valor digital, el diposita en una variable i prepara la lectura del següent canal de les entrades analògiques. El codi del programa s'explica amb més detall en l'Annex II.

La figura 5.21 mostra en un diagrama de flux com funcionen les conversions A/D automàtiques. Cal recordar que el funcionament dels dispositius són automàtics, per *hardware*, el programador només ha d'escriure el codi de la configuració dels mòduls ADC, CCP2 i TIMER1.

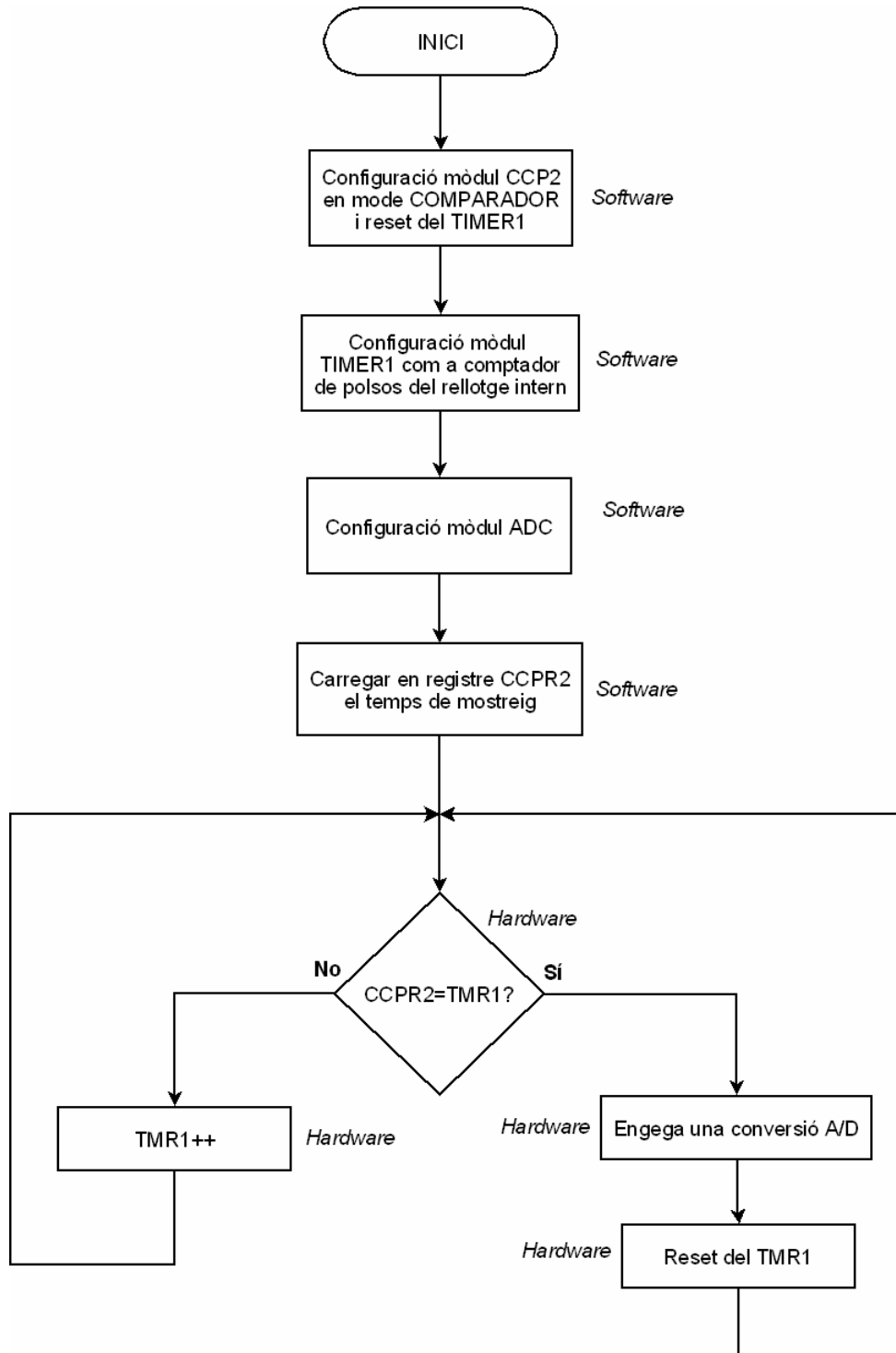


Figura 5.21. Diagrama de flux de la conversió A/D automàtica
(Font: Elaboració pròpia)

5.7. Interrupcions del PIC16F877A

Una interrupció es pot definir com “un indicador de la ocurrència d’un esdeveniment en el món real” (Horrillo, 2009). Sense les interrupcions seria impossible implementar sistemes de control en temps real. Les interrupcions tenen com a objectiu descarregar el processador de treball.

En el present projecte s'utilitzaran les interrupcions per el control de dos dispositius: el teclat i el convertidor A/D. Donat que l'analitzador de la xarxa elèctrica trifàsica és un sistema de temps real on la vigilància de l'excés de consum de la instal·lació i l'enviament de dades al PC són tasques primordials, l'ús adequat de les interrupcions del microcontrolador serà vital per assolir els objectius del projecte. Fins i tot es pot assegurar que la utilització correcta de les interrupcions és un dels aspectes clau d'aquesta aplicació. Més endavant, en el capítol dedicat a l'explicació del codi del programa, s'explica amb més detall l'ús adequat de les interrupcions.

El microcontrolador PIC16F877A disposa de 15 fonts o nivells d'interrupcions: 1 font d'interrupció externa i 14 fonts d'interrupció dels mòduls *hardware* interns. Quan es produeix una interrupció s'activa el bit o *flag* corresponent. Totes les fonts d'interrupció es poden emmascarar individualment o de forma global, deshabilitant totes les interrupcions. El dispositiu reconeix 4 fonts d'interrupció principals:

- Desbordament del TIMER0 (utilitzat també com a *wath-dog*)
- Interrupció externa (terminal RB0)
- Interrupció per canvi del PORT B (bits del 4 al 7)
- Interrupció dels perifèrics (dels 12 mòduls *hardware* interns)

La figura 5.22 mostra la lògica de les interrupcions. El bit d'habilitació global de les interrupcions, anomenat GIE (de l'anglès *Global Interrupt Enable*), permet l'atenció o no a totes les interrupcions que no estiguin emmascarades. Cada font d'interrupció es pot emmascarar individualment, amb el seu corresponent bit. A més del bit GIE, els perifèrics disposen de dues màscares més: el bit de màscara associat a cada perifèric i un bit de màscara general dels perifèrics anomenat PEIE (de l'anglès *PEripheral Interrupt Enable*).

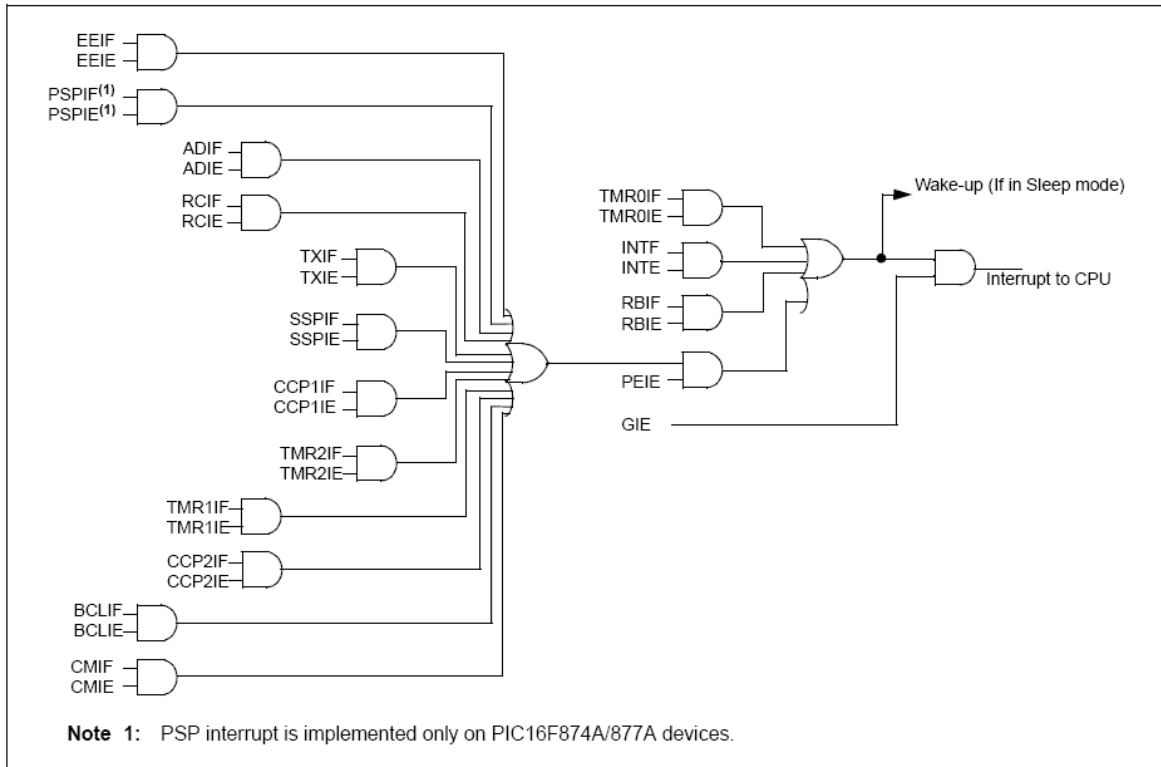


Figura 5.22. Lògica de control de les interrupcions
(Font: Data Sheet PIC16F87XA)

Dels perifèrics es disposa de les següents interrupcions:

- Escriptura finalitzada de la EEPROM interna.
- *Data in* al port esclau paral·lel (associat al PORT D).
- Conversió A/D finalitzada.
- S'ha omplert el *buffer* de recepció del port RS232.
- S'ha buidat el *buffer* de transmissió del port RS232.
- Hi ha activitat de comunicacions I²C™ o SPI.
- Captura o Comparació del mòdul CCP1.
- Desbordament del TIMER2.
- Desbordament del TIMER1.
- Captura o Comparació del mòdul CCP2.
- Hi ha col·lisió en el bus sèrie.
- S'ha produït un esdeveniment en el mòdul Comparador.

5.7.1. Vector d'interrupció

La família de microcontroladors PIC16F87XA només té un únic vector d'interrupció, associat a la posició 0x0004 de la memòria de programa, per tant no hi ha la possibilitat d'establir cap prioritat de les interrupcions.

Quan es produeix una interrupció el bit GIE es deshabilita per *hardware* per no permetre més interrupcions, la direcció de retorn al programa principal es guarda en la pila d'*stack* i el comptador de programa (anomenat PC, de l'anglès *Program Counter*) es carrega amb la direcció del vector d'interrupció, 0x0004. Aleshores la font que ha provocat la interrupció es determina per *polling* (és a dir, per sondeig), analitzant un per un quin bit o *flag* dels registres d'interrupció s'ha activat i a quin mòdul pertany. En primer lloc es fa el sondeig sobre el registre INTCON, per saber si la causa de la interrupció és produïda per la interrupció externa, el desbordament del TIMER0, si s'ha produït un canvi en els bits 4 al 7 del PORT B o si es tracta d'una interrupció produïda per algun perifèric. Si la causa de la interrupció és deguda a aquest últim, aleshores s'ha de tornar a fer *polling* en els registres PIE1 i PIE2 per conèixer exactament quin perifèric ha produït la interrupció.

5.7.2. Salvar el context o estat de l'aplicació

Quan es produeix una interrupció automàticament es salva la direcció de retorn al programa principal en l'*stack* i es deshabilita el bit GIE per evitar reentrades. Aquestes són les dues úniques accions que el microcontrolador executa per *hardware*. Per salvar l'estat de l'aplicació, per exemple registres temporals, el registre de treball *W* (equivalent al registre *acumulador* dels processadors *Intel 80x86*), el registre *Status* i variables del programa, s'ha de fer per *software*.

Quan se surt de la rutina de servei de la interrupció amb la instrucció RETFIE (instrucció en llenguatge ensamblador) automàticament també es restableix el bit GIE, però el *flag* de l'esdeveniment que ha provocat la interrupció s'ha d'esborrar per *software* en la rutina de servei de la interrupció abans de tornar a permetre les interrupcions, per evitar interrupcions recursives.

Tot el que s'ha explicat fins ara és vàlid sempre i quan es programi en llenguatge ensamblador. Si es programa en llenguatge C de CCS aleshores no cal tenir tantes precaucions perquè el compilador té present totes les regles del joc en comptes del programador. Si s'utilitzen les directives d'interrupció el compilador genera el codi necessari per executar la funció C que segueix a la directiva (la rutina de servei de la interrupció): el codi necessari per fer el *polling*, detectar la font de la interrupció i salvar el context al principi de la rutina i restituir-lo al final. També esborrarà el *flag* actiu per la interrupció. Sens dubte això suposa un avantatge considerable perquè per programar correctament en ensamblador s'ha de conèixer molt bé el microcontrolador, una feina llarga i feixuga.

En canvi aprendre a programar en C és més fàcil i ràpid, amb uns mínims coneixements del microcontrolador. El desavantatge evident és la manca de transparència entre el codi que gestiona les interrupcions i el microcontrolador i una sensible pèrdua del control d'una part del programa des del punt de vista del programador.

Les mancances del sistema interruptiu del microcontrolador PIC16F877A són, per una banda, la inexistència d'un mètode per establir prioritats en les interrupcions i per l'altre, el fet de què només tingui un únic vector d'interrupció (que obliga a fer *polling* per trobar la font de la interrupció). Els microcontroladors de les famílies superiors com ara els PIC18 resolen les mancances dels seus germans petits.

5.7.3. Interrupcions externa i de conversió A/D finalitzada

En el present projecte es generarà codi per atendre dues fonts d'interrupció: la interrupció externa i la de conversió A/D finalitzada.

El microcontrolador PIC16F877A només té un terminal per connectar-hi una font de interrupció externa: el bit RB0 del PORT B de E/S. Aquest terminal és electrònicament *edge triggered*, és a dir, té la capacitat de ser activat per flanc, ja sigui positiu o negatiu. En el codi del programa principal s'ha d'incloure una línia que especifiqui quin tipus de flanc provocarà l'activació d'aquesta interrupció. En aquest terminal es connectarà el senyal *data available* del *key-encoder* del teclat.

Quan el *key-encoder* detecta que s'ha polsat una tecla aquest senyal assoleix un '1' lògic. Així que la interrupció externa s'ha de configurar perquè s'activi quan detecti un flanc positiu, quan passi del '0' lògic a '1' lògic.

Per habilitar la interrupció de conversió A/D finalitzada només s'ha d'incloure una línia de codi en el programa principal quan es programa en llenguatge C de CCS i generar el codi de la rutina de servei de la interrupció. A continuació es pot veure que posar en marxa el sistema interruptiu de l'analitzador de la xarxa elèctrica trifàsica és molt senzill si es fa amb aquest llenguatge de programació:

```
// Habilitació de la interrupció externa RB0 per flanc de pujada
ext_int_edge(L_TO_H);
enable_interrupts(INT_EXT);

// Habilitació de la interrupció conversió A/D finalitzada
enable_interrupts(INT_AD);

// Habilitació general de les interrupcions
enable_interrupts(GLOBAL);
```

Al capítol dedicat al codi del programa s'explica amb més detall les rutines de servei a les dues interrupcions. El codi generat per atendre les interrupcions serà clau per poder respectar les restriccions temporals del sistema de vigilància del consum de corrent elèctric del sistema trifàsic que s'hagi de mesurar.

5.8. Teclat i *key-encoder*

En una infinitat de projectes es necessita transmetre certa informació al microcontrolador. Sovint s'utilitzen polsadors o selectors per què l'usuari pugui interactuar amb l'equip quan la informació que s'ha de transmetre és molt simple, però quan la informació requereix més contingut aleshores s'utilitzen teclats de major o menor complexitat.

En el projecte de l'analitzador de la xarxa elèctrica trifàsica s'utilitza un teclat matricial de 4x4, tal i com es pot veure a la figura 5.23.



Figura 5.23. Teclat matricial 4x4
(Font: Elaboració pròpia)

El teclat matricial està constituït per una matriu de polsadors disposats en files i columnes amb l'objectiu de reduir el nombre de línies. La figura 5.24 mostra la connexió interna dels polsadors del teclat.

Si el teclat es connectés directament al microcontrolador es necessitarien 8 línies de E/S i uns quants components externs, com díodes de protecció i resistències de *pull-up*. El PIC16F877A i molts altres microcontroladors de *Microchip* tenen la opció d'habilitar unes resistències internes del bits 4 al 7 del PORT B per connectar-hi un teclat, però no s'utilitzarà aquesta opció en aquest projecte. Per la connexió del teclat s'utilitzarà un dispositiu anomenat *key-encoder* o codificador de tecla.

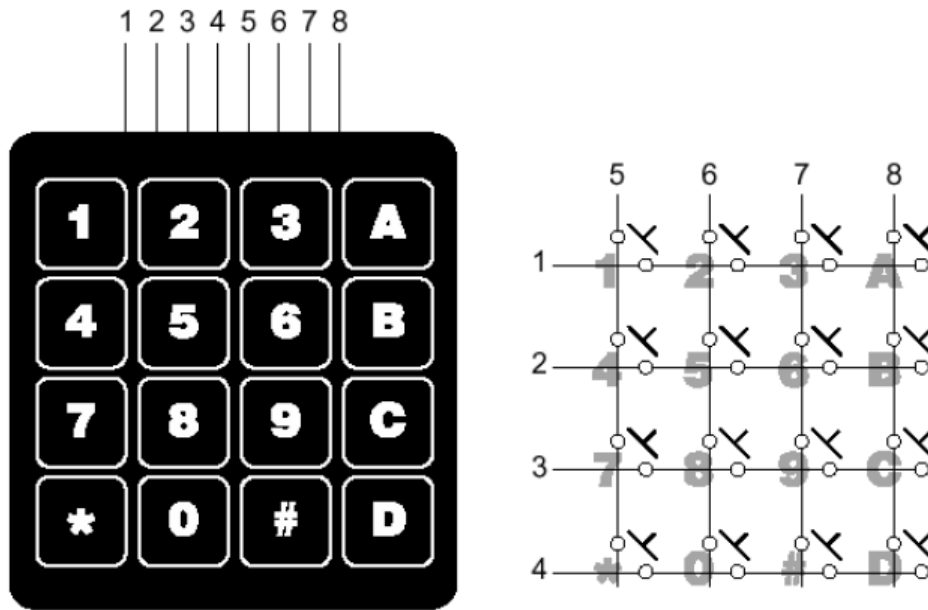


Figura 5.24. Connexió interna dels pulsadors del teclat

(Font: Data Sheet teclat)

Hi ha molts *key-encoder* al mercat, però s'ha escollit el MM74C922 dels fabricants *Fairchild Semiconductor* perquè poden treballar amb 5V, només necessiten un parell de condensadors externs per funcionar i amb només 5 línies es connecta al microcontrolador: 4 línies per els bits del codi de la tecla pulsada i una línia per el senyal d'interrupció. Les característiques més rellevants d'aquests dispositiu són:

- Opció de rellotge intern amb un sol condensador extern.
- Resistències de *pull-up* integrades en el xip.
- Pot detectar quina tecla es troba pulsada encara que s'hagin pulsat dues alhora.
- Eliminació de rebots amb un sol condensador extern.
- Memòria en registre LATCH de la darrera tecla pulsada connectat a les sortides.
- Sortida 3-STATE de tecnologia CMOS compatible amb TTL de baixa potència.
- Ampli rang de tensió d'alimentació: de 3V a 15V.
- Baix consum: 0,55 a 1,1mA.

Teoria de funcionament del *key-encoder* MM74C922

Els *key-encoders* MM74C922 incorporen tota la lògica necessària per poder ser utilitzats com a interfícies entre els teclats matricials de 16 tecles i els sistemes digitals. Poden convertir una tecla polsada en un *nibble* de 4 bits. El dissenyador pot escollir tant la freqüència d'escaneig del teclat com el temps anti-rebot de les tecles escollint el valor de dos condensadors externs; així el *key-encoder* MM74C922 es pot adaptar a molts tipus de teclats.

Quan no es polsa cap tecla les files d'entrada assoleixen un estat lògic alt a través de les resistències de *pull-up* internes i les columnes de sortida treuen un '0' lògic de manera seqüencial. Aquestes sortides són de drenador obert i es troben en estat baix el 25% del temps. La resta del temps es "desconnecten" (estat d'alta impedància). La freqüència d'escaneig de les columnes es controla per un oscil·lador *Schmitt trigger*, un comptador de 2 bits i un descodificador de 4 bits.

Quan es polsa una tecla, per exemple la tecla 0, no succeeix res quan el terminal X1 es troba desconnectat i el terminal Y1 manté l'estat alt per l'acció de les resistències de *pull-up*. Quan s'escaneja la columna X1 aleshores assoleix l'estat baix i Y1 també. Això deshabilita el comptador i es manté X1 en estat baix. Aleshores es posa en marxa el circuit anti-rebots i es bloquegen les altres entrades de les files Y. El codi que representa la tecla polsada és la combinació de la fila i la columna on es troba la tecla, com es pot veure a la taula 5.2. A les sortides del *key-encoder* (numerades amb les lletres A, B, C, D) s'obté el codi expressat en binari. Quan el temps de l'anti-rebot s'esgota aleshores el registre *lacth* memoritza el codi de la tecla polsada i el terminal de sortida *Data Available* assoleix l'estat lògic alt, activant la interrupció del microcontrolador.

Finalment les sortides del registre *lacth* que conté el codi de la darrera tecla polsada s'activen quan el senyal d'entrada *Output Enable* assoleix l'estat lògic baix. En aquell moment el microcontrolador rep per les quatre línies d'entrada el codi de la tecla polsada. La figura 5.25 mostra el diagrama de blocs intern del *key-encoder* MM74C922.

A	B	C	D	Tecla pulsada	Codi Hex
0	0	0	0	'1'	0x00
0	0	0	1	'2'	0x01
0	0	1	0	'3'	0x02
0	0	1	1	'A'	0x03
0	1	0	0	'4'	0x04
0	1	0	1	'5'	0x05
0	1	1	0	'6'	0x06
0	1	1	1	'B'	0x07
1	0	0	0	'7'	0x08
1	0	0	1	'8'	0x09
1	0	1	0	'9'	0x0A
1	0	1	1	'C'	0x0B
1	1	0	0	'*'	0x0C
1	1	0	1	'0'	0x0D
1	1	1	0	'#'	0x0E
1	1	1	1	'D'	0x0F

Taula 5.2. Relació del codi segons la tecla pulsada
(Font: Elaboració pròpia)

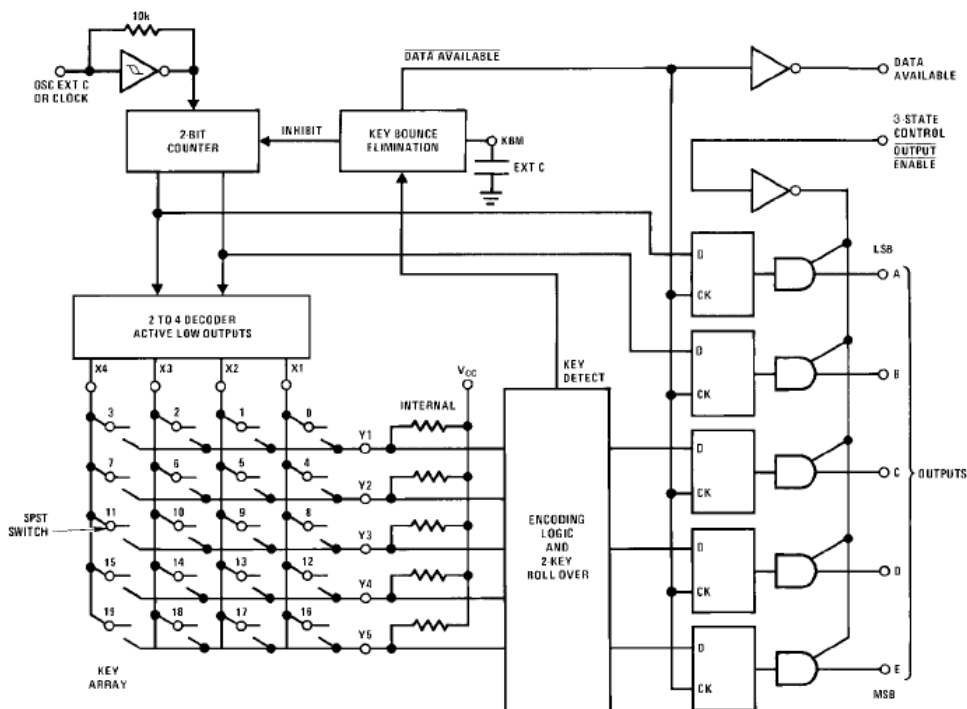


Figura 5.25. Diagrama de blocs del key-encoder MM74C922
(Font: Data Sheet MM74C922)

Les gràfiques representades a la figura següent permeten escollir tant el temps d'escaneig de les columnes (i per tant la velocitat de resposta del teclat) com el temps anti-rebot en funció de dos condensadors externs convenientment connectats al dispositiu.

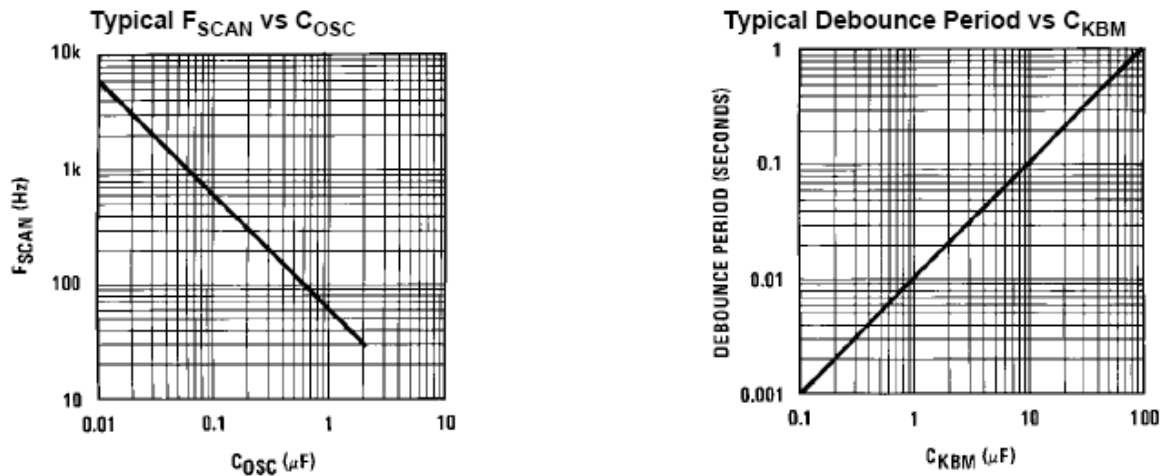


Figura 5.26. Gràfiques freqüència d'escaneig i temps anti-rebot

(Font: Data Sheet MM74C922)

Per aquest projecte s'ha escollit un condensador de $0,1\mu\text{F}$ per obtenir una freqüència d'escaneig del teclat de 600Hz i un condensador de $1\mu\text{F}$ per un temps anti-rebot de 10ms . Aquests condensadors es van muntar sobre un sòcol (figura 5.28) per si de cas la freqüència d'escaneig o el temps anti-rebot no fossin els adequats.

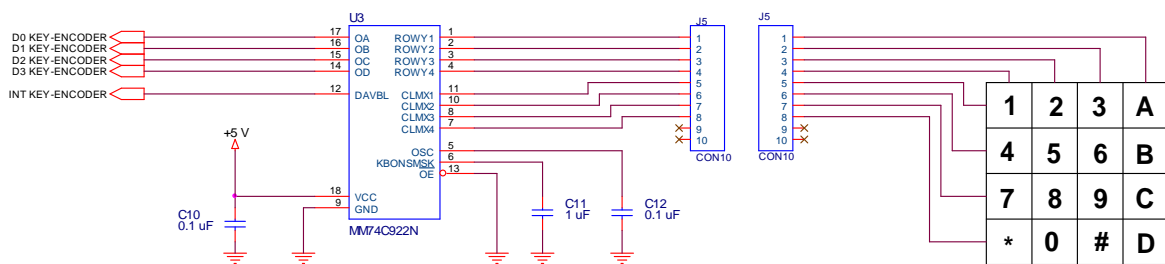


Figura 5.27. Connexió del key-encoder en el circuit

(Font: Elaboració pròpia)

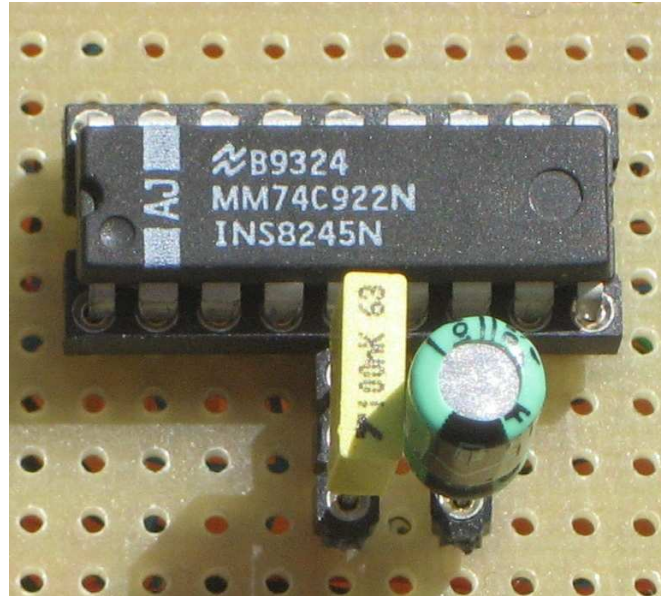


Figura 5.28. Key-encoder i condensadors externs
(Font: Elaboració pròpia)

5.9. Visualitzador LCD

L'altre dispositiu que permetrà la interacció entre l'usuari i l'analitzador de la xarxa elèctrica trifàsica és un visualitzador o *display* LCD (de l'anglès *Liquid Crystal Display*) de dues línies i setze caràcters per línia. La inclusió d'un *display* en qualsevol projecte sempre li dona cert valor afegit, perquè el fa més agradable i interessant.



Figura 5.29. Visualitzador LCD
(Font: Elaboració pròpia)

Les pantalles de cristall líquid per missatges tenen la capacitat de mostrar qualsevol caràcter alfanumèric i permeten representar la informació que genera qualsevol equip electrònic d'una forma fàcil i econòmica. El model utilitzat en aquest projecte consta d'una matriu de setze caràcters de 5x7 punts distribuïts en dues línies. El procés de visualització és governat per un microcontrolador incorporat a la pantalla, sent el *Hitachi 44780* el model més utilitzat.

El mercat ofereix diversos models de visualitzadors amb diferents prestacions. El model escollit per aquest projecte és el PC1602LRS-LWA del fabricant *Powertip*. Les característiques generals d'aquest dispositiu són les següents:

- Pantalla de caràcters ASCII, caràcters japonesos Kanji, caràcters grecs i símbols matemàtics.
- Desplaçament dels caràcters cap a l'esquerra o la dreta.
- Memòria de 40 caràcters per línia de pantalla, tot i què només es poden visualitzar setze caràcters alhora.
- Moviment del cursor i canvi del seu aspecte.
- Permet que l'usuari pugui programar vuit caràcters nous.
- Es pot governar de dues maneres principals:
 - Connexió amb bus de 4 bits.
 - Connexió amb bus de 8 bits.

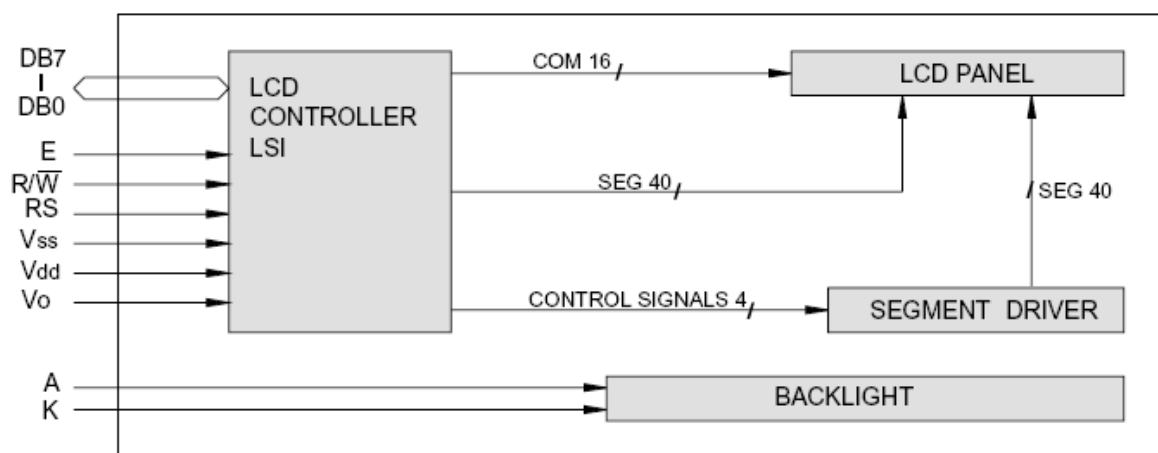


Figura 5.30. Diagrama de blocs del visualitzador

(Font: Data Sheet PC1602LRS-LWA)

Els terminals d'aquests visualitzadors es poden connectar directament als terminals d'E/S dels microcontroladors sense l'addició de cap dispositiu extern ni cap *buffer*. L'alimentació és de +5V. La regulació del contrast es realitza mitjançant la tensió que s'obté en dividir els 5V amb una resistència ajustable de 10k Ω i aplicar-la al terminal Vo. En alguns projectes es connecta aquest terminal Vo a massa, aconseguint així el màxim contrast.

Aquest visualitzador necessita una font d'alimentació extra per donar tensió a la retroil·luminació per díodes LED. A la figura següent es pot veure el detall la connexió del visualitzador al circuit de l'anàlitzador de la xarxa elèctrica trifàsica:

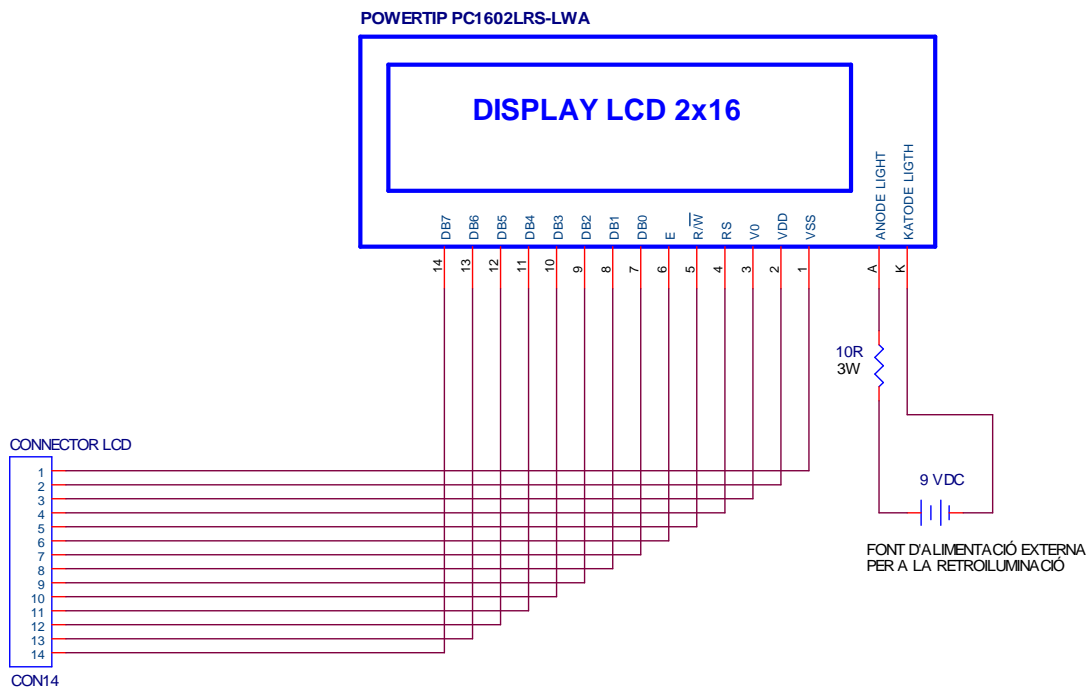


Figura 5.31. Connexió del visualitzador en el circuit
(Font: Elaboració pròpia)

El visualitzador té diferents modes de funcionament: *mode comandament* per rebre ordres com “esborrar el *display*” o “moure el cursor”; *mode caràcter* o *dada* per rebre per el bus un caràcter ASCII per visualitzar-lo i *mode de lectura del Busy Flag* o *LCD ocupat*, per saber quan es pot accedir de nou al bus per transmetre un nou caràcter. El compilador C de CCS inclou un *driver* que simplifica molt programar el visualitzador. Els detalls de la programació s'expliquen en un capítol posterior.

5.10. Connectivitat RS232

Per realitzar la comunicació amb el PC s'utilitzarà l'estàndard RS232. Aquest estàndard, elaborat per la EIA (de l'anglès *Electronics Industries Alliance*), defineix la interfície mecànica, els terminals, els senyals i els protocols que ha de complir aquesta comunicació sèrie.

En la versió més simple només s'utilitzen dos terminals per a la comunicació sèrie, el senyal de transmissió (TX) i el de recepció (RX). La norma RS232 requereix els següents nivells de tensió:

- Un '1' lògic és un voltatge comprés entre -5 i -15 V en el transmissor i entre -3 i -25 V en el receptor.
- Un '0' lògic és un voltatge comprés entre +5 i +15 V en el transmissor i entre +3 i +25 V en el receptor.

El mòdul USART del PIC16F877A (de l'anglès *Universal Synchronous Asynchronous Receiver Transmitter*) treballa amb nivells de tensió digital TTL, de manera que s'han d'adaptar els de l'estàndard RS232. Per fer la conversió dels nivells de tensió del mòdul USART als de la norma RS232 s'utilitza el circuit integrat ST232CN.

Aquest circuit integrat soluciona els problemes dels nivells de tensió quan s'han d'enviar senyals digitals sobre un línia RS232. Es tracta d'una solució adequada per aplicacions on no es disposa de fonts de $\pm 12V$. El ST232CN només necessita per al seu funcionament una única font de +5V i cinc condensadors de 0.1 μ F. Internament disposa d'un elevador de tensió que converteix la tensió de +5V a una de doble polaritat de $\pm 10V$. En el mercat es poden trobar una gran varietat de circuits integrats que compleixen amb la norma RS232 i que són similars al ST232CN.

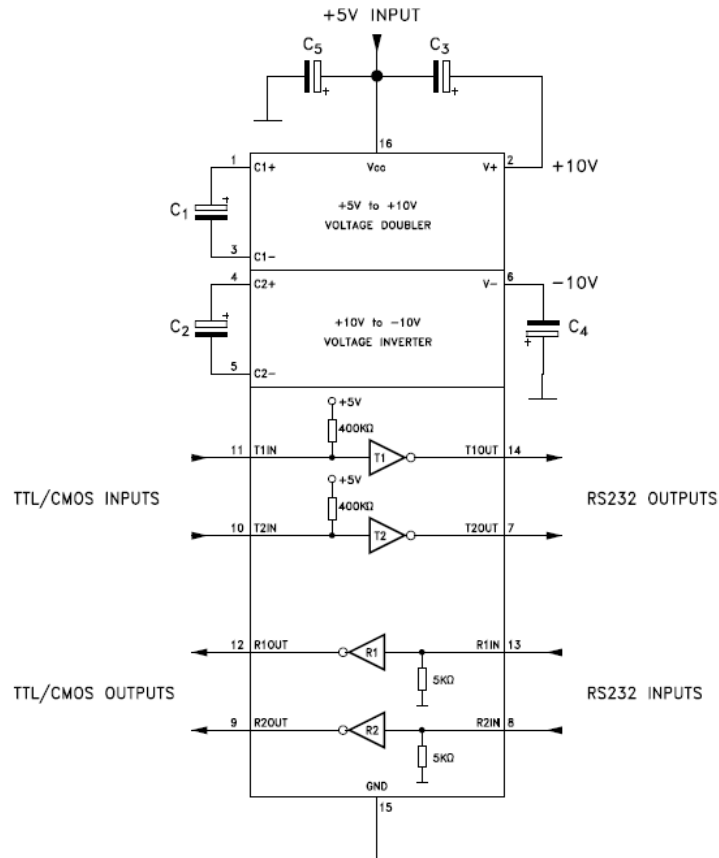


Figura 5.32. Configuració del ST232CN
(Font: Data Sheet ST232CN)

En la figura anterior es poden observar els cinc condensadors necessaris per al correcte funcionament d'aquest circuit i la configuració interna del ST232CN. Disposa de dos *drivers* i dos *receivers*, tot i que per la implementació del port RS232 en el circuit de l'analitzador de la xarxa elèctrica trifàsica només s'utilitzarà un *driver* i un *receiver*, així com la configuració més simple de la norma RS232, només amb els senyals de transmissió i recepció.

Si es programa el microcontrolador en el llenguatge C de CCS s'ha d'incloure el codi següent en les directives del preprocessador per habilitar el mòdul USART. Es defineix la velocitat en bauds, els terminals de connexió Tx i Rx, la paritat i el nombre de bits:

```
// Configuració del mòdul USART per la comunicació RS232
#include <delay.h> // Freqüència CPU 20 MHz
#include <rs232.h> // Configuració del mòdul USART
```

A la figura 5.33 es pot observar la connexió del ST232CN en el prototip, on TXRS232 és la sortida Tx i RXRS232 és l'entrada Rx del mòdul USART del PIC16F877A. La sortida del *driver* T1 es connecta al terminal 2 del connector DB9 i l'entrada del *receiver* R1 es connecta al terminal 3 del connector DB9, seguint amb la norma RS232. Els condensadors de 0,1µF es connecten seguint les recomanacions del fabricant.

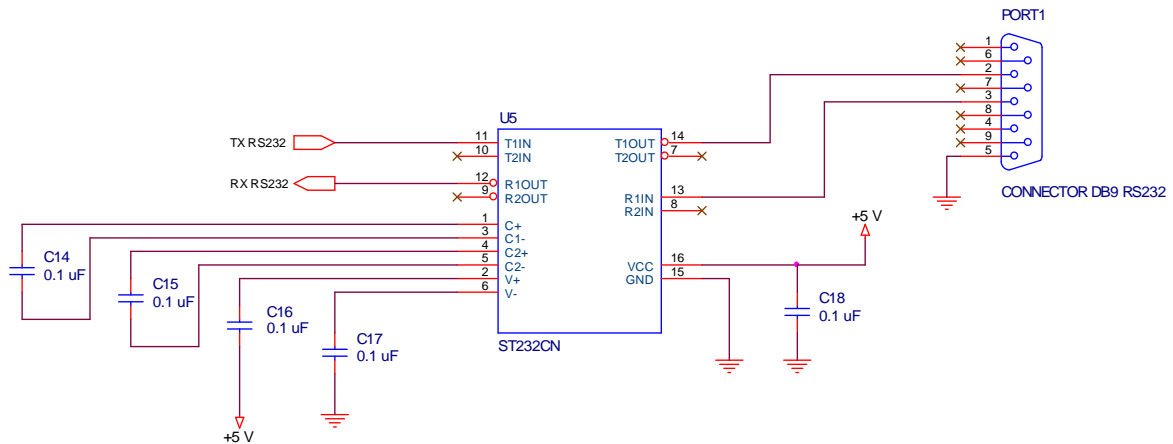


Figura 5.33. Connexió del ST232CN en el circuit
(Font: Elaboració pròpia)

5.11. RTC i bus I²C™

5.11.1. RTC

L'analitzador de la xarxa elèctrica trifàsica inclou un RTC (de l'anglès *Real Time Clock*), és a dir, un rellotge de temps real, per enregistrar l'hora en què s'ha produït una incidència, per calcular el consum promig per fase de les darreres 24 hores i altres funcions més.

De tots els RTC que hi ha al mercat s'ha escollit el DS1307 fabricat per *Dallas Semiconductors*. Aquest dispositiu electrònic és un potent rellotge i calendari de temps real que compleix perfectament amb moltes de les necessitats en l'adquisició i registre del temps. A més és molt fàcil d'incloure en qualsevol projecte i molt senzill de programar.

Les seves característiques més destacades són:

- Encapsulat de 8 terminals.
- Connectivitat I²C™ només amb dos fils.
- Relloige i calendari en temps real que compta segons, minuts, hores, dia de la setmana (dilluns, dimarts, etc.), data, mes i any fins el 2100.
- Emmagatzema les dades en format BCD.
- Memòria RAM de 56 bytes, no volàtil amb bateria externa.
- Sortida d'ona quadrada programable per el terminal SQW/OUT.
- Circuit intern de recolzament de l'alimentació amb bateria externa de 3V en cas de fallida de l'alimentació principal. És capaç de mantenir la data i l'hora actualitzats fins i tot quan tot el sistema es troba desconnectat.
- Tensió d'alimentació entre 4,5V i 5,5V.
- Baix consum, menys de 500nA en el mode de recolzament amb la bateria.
- Utilitza un cristall de quars extern de 32.768Hz per obtenir temps exactes i no dependre del microcontrolador.
- El darrer dia del mes s'ajusta automàticament i té en compte els anys de traspàs.
- Pot treballar en format de 24 hores o de 12 hores amb indicador de AM/PM.

Les figures següents mostren els terminals del DS1307 en encapsulat DIP i la connexió en el circuit de l'analitzador de la xarxa elèctrica trifàsica:

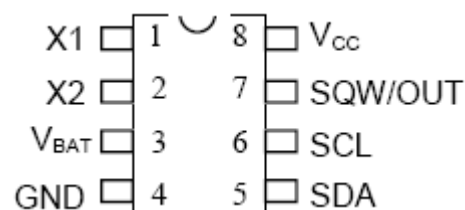


Figura 5.34. Configuració dels pins del DS1307

(Font: Data Sheet DS1307)

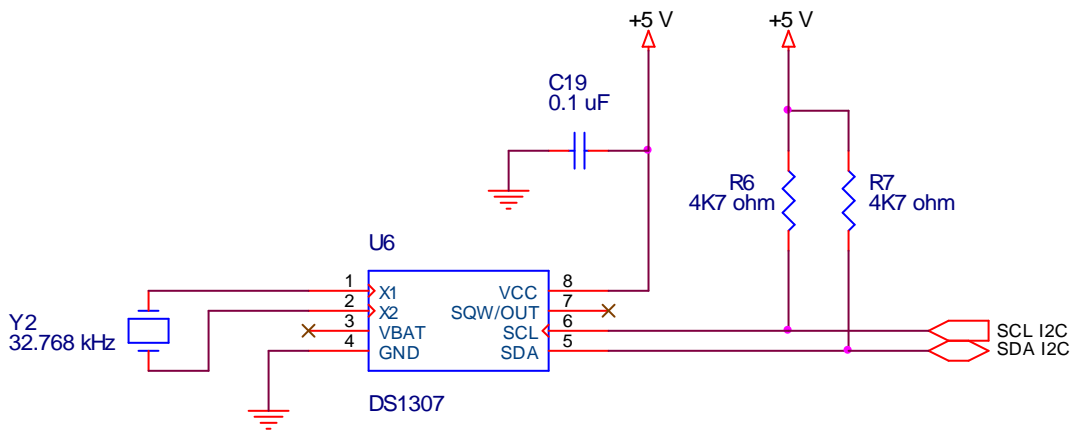


Figura 5.35. Connexió del DS1307 en el circuit

(Font: Elaboració pròpia)

5.11.2. Bus I²C™

La connexió del RTC amb el microcontrolador es realitza per mitjà del bus I²C™. Aquest és un bus sèrie síncron i *half-duplex*, desenvolupat per l'empresa *Philips Semiconductors*, qui es va adonar que nombrosos projectes amb microcontroladors de 8 bits tenien els mateixos criteris de disseny:

- El sistema consta de almenys un microcontrolador i varis dispositius perifèrics com memòries, convertidors A/D, RTC, visualitzadors, etc.
- La connexió entre els diferents dispositius que formen part del sistema ha de ser fàcil de realitzar i el seu cost mínim.
- Aquests sistemes acostumen a realitzar funcions que no requereixen un tràfic molt alt de transferència de dades, generalment no superior als 100 kbits per segon.
- El sistema no ha de dependre dels dispositius que estiguin connectats. Ha de ser possible realitzar modificacions i/o millores.

Tots aquests criteris son la base i els fonaments de les especificacions del bus sèrie per a la interconnexió de circuits integrats o bus I²C™ (de l'anglès *IIC*, *Inter Integrated Circuit Bus*).

De fet el bus I²C™ ha esdevingut un estàndard mundial, implementat en més de 1000 diferents circuits integrats i amb més de 50 fabricants de dispositius que utilitzen aquest sistema de comunicació. És un bus que ha anat evolucionant al llarg dels anys, incrementant notablement la velocitat de transferència, des dels 100 Kbps de la versió estàndard fins als 3,4 Mbps de la versió d'alta velocitat o *High-Speed*. La versió del bus I²C™ que porta el PIC16F877A és la versió ràpida o *Fast*, que permet assolir velocitats de transferència de dades de fins a 400 Kbps, 4 vegades per ràpid que el bus convencional. Com que en el bus I²C™ del circuit de l'analitzador de la xarxa elèctrica trifàsica només es connectarà un dispositiu, el RTC DS1307, aleshores aquesta velocitat és suficient.

Teoria de funcionament del bus I²C™

El bus I²C™ es basa en la comunicació a través de dos fils: un fil per les dades i un altre per el senyal de rellotge. Cada dispositiu connectat al bus té una direcció. Es pot configurar amb un mestre i varis esclaus o amb una configuració *Multi-mestre*. En totes dues és el mestre qui gestiona la transferència: pren la iniciativa, decideix amb qui es realitza, el sentit de la comunicació i quan es finalitza. Quan el mestre inicia una comunicació primer transmet la direcció del dispositiu amb el qual es vol comunicar i a continuació els esclaus comproven si es tracta de la seva direcció.

La transmissió pot ser de lectura o d'escriptura: el darrer bit de la transmissió ho indica; així el mestre actuarà com a transmissor i l'esclau com a receptor i a l'inrevés. En qualsevol cas el senyal de rellotge sempre el genera el mestre.

Els dos fils del bus I²C™ són dues línies de col·lector obert: el senyal de rellotge *SCL* i la línia de dades *SDA*. És per aquesta raó que s'han d'utilitzar dues resistències de *pull-up* per assegurar un nivell alt quan no hi ha cap dispositiu connectat al bus. A la figura 5.35 es pot veure amb claredat aquestes resistències *R6* i *R7*, d'un valor de 4K7Ω, que són les que recomana el fabricant.

El nombre de dispositius connectats i la longitud del bus estan limitats per la capacitat d'adreçament (de 7 a 10 bits) i per la càrrega màxima del bus (400pF).

La transmissió comença amb un bit d'inici o *START* i finalitza amb el bit d'*STOP*. La condició d'*START* s'estableix amb una transició de nivell alt a nivell baix en la línia *SDA* (que normalment es troba a nivell alt, per les resistències de *pull-up*), quan la línia *SCL* es troba a nivell alt. L'*STOP* s'estableix quan es produeix una transició de nivell baix a nivell alt en la línia *SDA* quan *SCL* es troba a nivell alt; fent-ho d'aquesta manera les dades que es transmeten per la línia *SDA* només canvien quan la línia *SCL* es troba en estat baix.

Quan comença la transmissió el mestre envia l'adreça de l'esclau amb el que desitja establir la comunicació. L'adreça pot ser de 7 o 10 bits amb format de byte (un o dos bytes respectivament). Darrera de l'adreça s'adjunta un bit de lectura/escriptura.

Una vegada el mestre ja ha enviat l'adreça (o les dades), l'esclau genera un bit de reconeixement (*ACK*). Si el mestre no rep aquest bit la comunicació queda interrompuda, generant el senyal *STOP*. El mestre també pot rebre dades, en aquest cas és ell qui genera el bit de reconeixement per a cada byte rebut menys per el darrer; en aquest cas l'esclau allibera la línia *SDA* i el mestre genera un *STOP*.

La figura 5.36 mostra en un diagrama temporal la seqüència sencera de transmissió d'una dada utilitzant el bus *I2C™*. Es pot observar com es sincronitzen els senyal *SDA* i *SCL* quan s'estableix la comunicació.

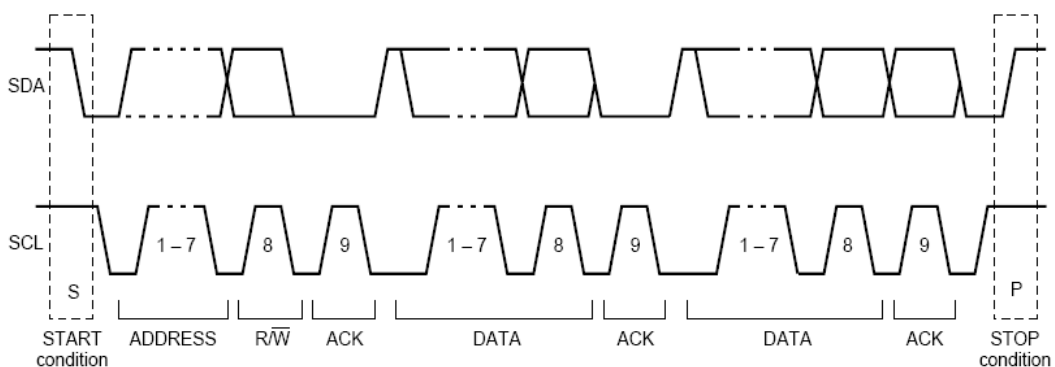


Figura 5.36. Transferència sencera d'una dada al bus *I2C™*
(Font: Data Sheet bus *I2C™* de *Philips Semiconductors*)

El microcontrolador PIC16F877A inclou un mòdul que permet la comunicació sèrie amb el bus I2C™. El mòdul s'anomena MSSP (de l'anglès *Master Synchronous Serial Port*). Aquest mòdul permet detectar les condicions d'*START* i d'*STOP* per interrupció. Admet tres modes de funcionament:

- Mestre
- Esclau amb direcció de 7 bits
- Esclau amb direcció de 10 bits

En el present projecte s'utilitzarà el mode *mestre* donada la poca complexitat del circuit.

El compilador C de CCS facilita molt l'ús d'aquest bus. En una línia de codi es configura es bus, i amb sentències del tipus I2C_START(), I2C_READ(), I2C_WRITE() s'estableix la comunicació, es llegeixen i s'escriuen les dades.

```
// Configuració del bus I2C™  
#use i2c(master,fast,sda=PIN_C4,scl=PIN_C3,force_hw)
```

Amb aquesta directiva del preprocessador (que no es troba a dins del *void_main()*) es configura el bus I2C™ del microcontrolador en mode *master* (mestre), amb velocitat ràpida (*fast*) de 400 Kbps i s'estableixen quins són els terminals que s'utilitzaran per implementar els senyals *SDA* i *SCL*. El paràmetre *force_hw* estableix que s'utilitzaran les funcions I2C™ del mòdul *hardware*.

En el capítol dedicat a l'explicació del codi de programa es donen més detalls de com s'utilitzen les funcions que proporciona el compilador C de CCS per comunicar-se amb el RTC amb el bus I2C™.

6. MUNTATGE DEL PROTOTIP I PROVES DE FUNCIONAMENT

6.1. Muntatge del prototip

El muntatge del prototip, incloent-t'hi tots els components que s'han descrit al capítol anterior, s'ha realitzat sobre un placa perforada de prototips utilitzant la tècnica *wire-wrapping*, combinada amb la soldadura manual amb estany.

S'ha escollit aquest tipus de muntatge per la facilitat amb la que es poden detectar errors, realitzar modificacions i millores, tasques molt importants quan es munten prototips.

A més, el fet de realitzar el muntatge de manera modular, dividint-lo en diferents parts, ha minimitzat l'aparició d'errors, degut a la gran quantitat de connexions que s'han hagut de realitzar.

El muntatge s'ha dividit en les següents parts: alimentació, entrades i sortides digitals, entrades i sortides analògiques, circuit de *reset*, connectivitat RS232, relé i LED, i finalment la connectivitat ICD2 per a la programació del microcontrolador. En finalitzar cada una de les diferents parts del muntatge s'han efectuat proves de funcionament *hardware*, per verificar el muntatge i el correcte funcionament dels diferents elements que formen part de l'analitzador de la xarxa elèctrica trifàsica.

El muntatge sencer del prototip s'ha realitzat sobre dues plaques metàl·liques per poder fer la demostració: en una placa es troben: la placa del prototip principal, la font d'alimentació de $\pm 15V$, el visualitzador LCD i el teclat.

En l'altre placa es troben els sensors de corrent (els transformadors), les tres fonts d'alimentació de 24V per els llaços de corrent i les dues plaques dels circuits analògics.

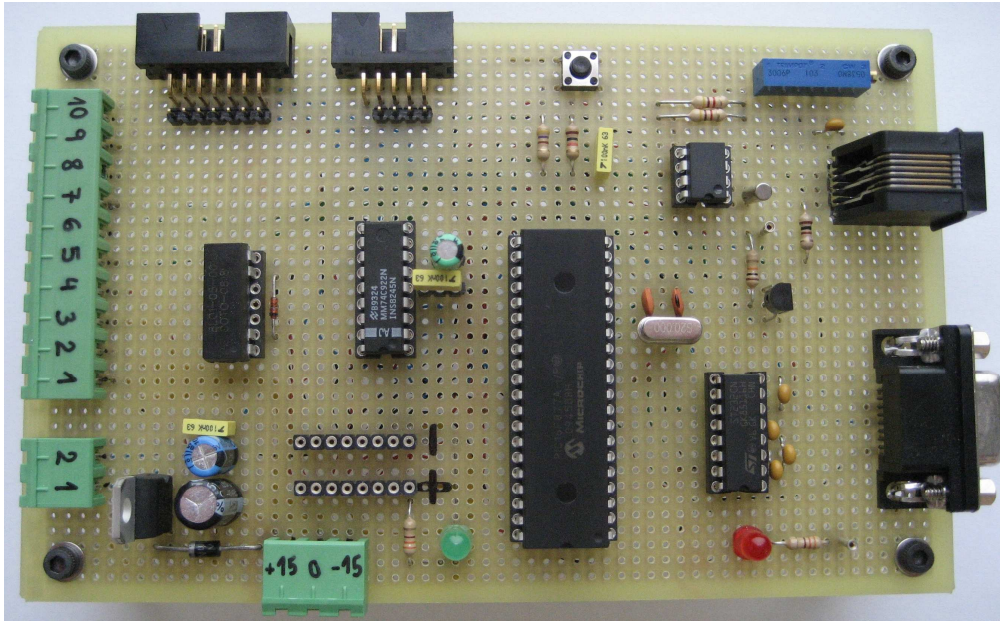


Figura 6.1. Vista de la cara superior del prototip
(Font: Elaboració pròpia)

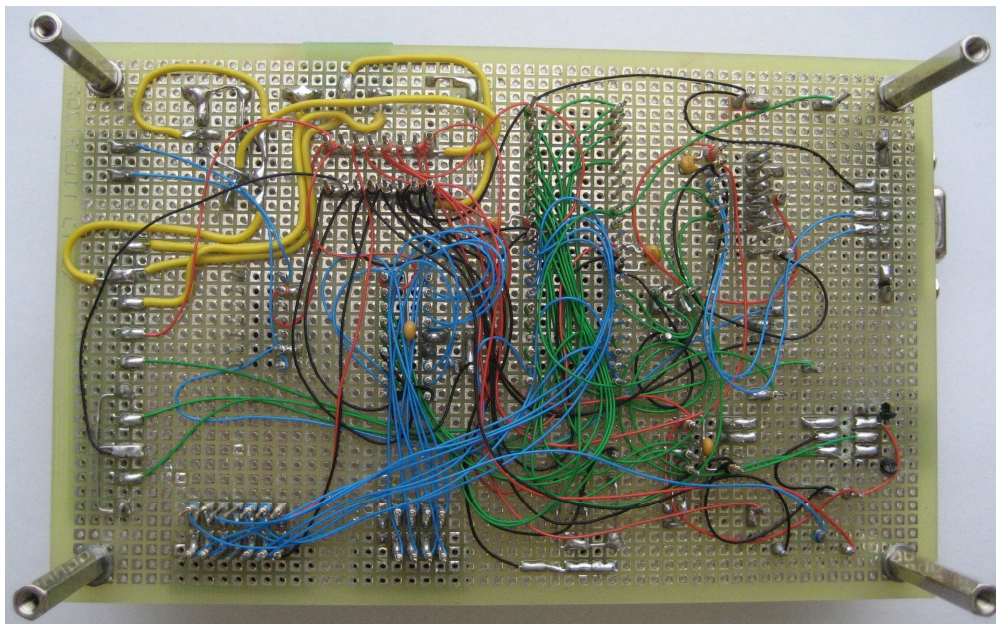


Figura 6.2. Vista de la cara inferior del prototip
(Font: Elaboració pròpia)

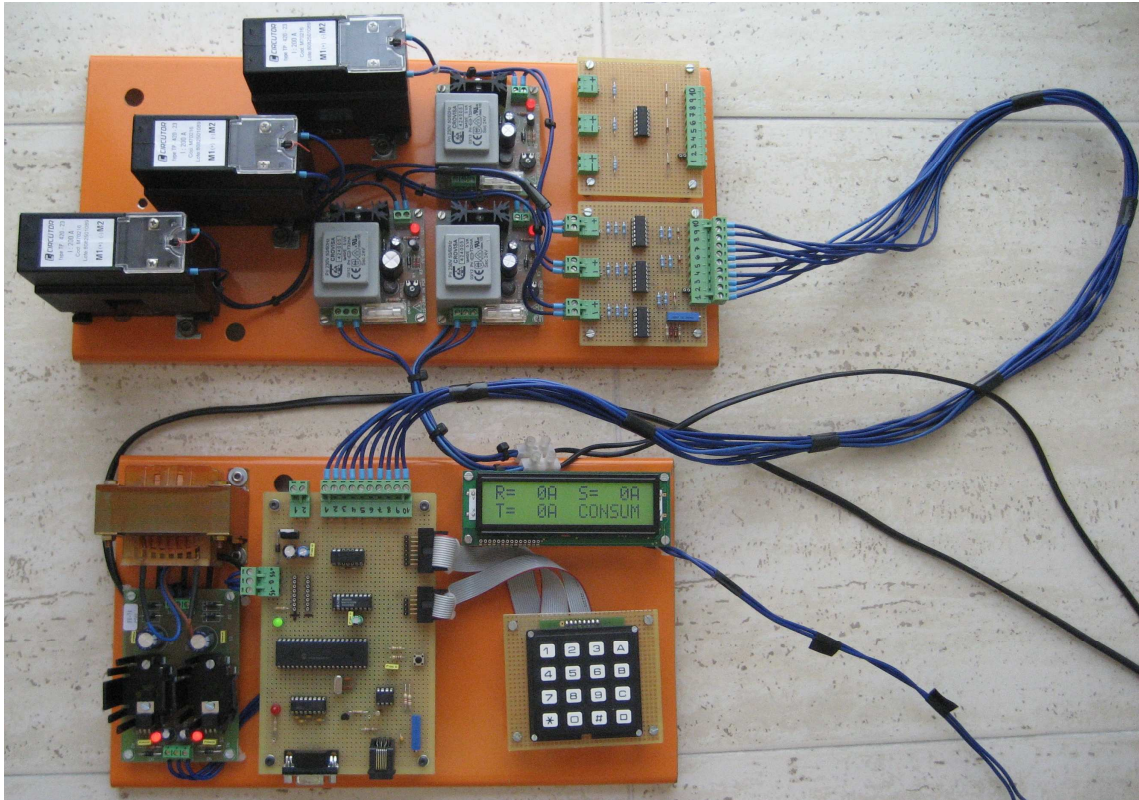


Figura 6.3. Muntatge del projecte sencer

(Font: Elaboració pròpia)

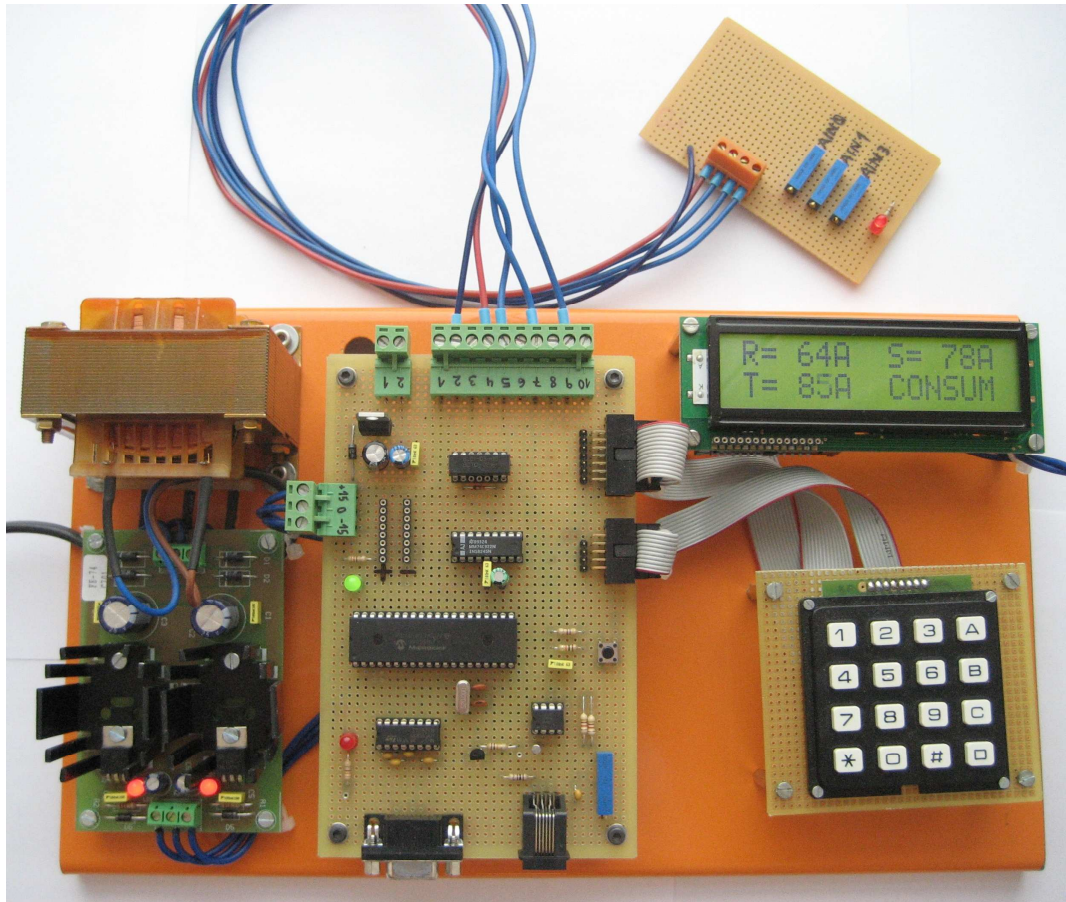
6.2. Proves de funcionament

S'han realitzat dos tipus de proves de funcionament: una prova amb potència (amb uns redòstats de 1.000W) i una altra prova amb senyal analògic de baixa potència, amb un senzill simulador. Per fer les proves de funcionament amb senyal analògic de baixa potència s'ha construït un simulador dels sensors de corrent, un dispositiu molt senzill capaç de proporcionar un valor de tensió de 0 a 5V per a les tres fases amb potenciòmetres multivolta.

En totes dues proves el resultat ha estat satisfactori. En la prova amb potència s'ha comprovat que la lectura que proporcionaven els sensors no assolien la exactitud desitjada; però aquest fet és del tot normal ja que el fabricant dels transformadors de corrent assegura una exactitud de 1,5% sempre i quan el corrent mesurat es trobi per sobre del 5% del valor màxim que pot mesurar el transformador.

Si el transformador és de 200A, el 5% són 10A, per tant no es pot arribar a assolir aquest valors de corrent amb els mitjans disponibles. Per raons de seguretat no s'ha provat amb més potència que els 1.000W dels reòstats del Laboratori de Potència de l'EUPMT.

La figura 6.4 ofereix una vista de la prova amb el simulador. El fet d'utilitzar el simulador ha permès provar totes les funcions que ofereix l'analitzador de la xarxa elèctrica trifàsica de propòsit general sense assumir cap risc.



**Figura 6.4. Proves de funcionament amb el simulador
(Font: Elaboració pròpia)**

7. DESCRIPCIÓ DEL CODI DE PROGRAMA

En aquest capítol s'explica el funcionament del codi de programa, el *software*, que és allò que dona vida al sistema *hardware* i el fa funcionar.

7.1. Eines de desenvolupament

La política oberta de desenvolupament de *Microchip* ha afavorit l'existència al mercat de compiladors de diferents llenguatges i de diferents fabricants per els famosos microcontroladors PIC[®]. Així, a més del compilador en ensamblador MPASM propietari de *Microchip* existeixen altres compiladors d'altres llenguatges, per exemple, de llenguatges d'alt nivell com C, BASIC i d'altres.

Compilador CCS C v4.057

Entre els compiladors aliens a *Microchip* destaca el compilador de C de l'empresa CCS que és el que s'ha triat per aquest projecte. Aquest compilador permet desenvolupar programes en C, facilitant molt el desenvolupament d'aplicacions i projectes. Ha estat desenvolupat específicament per els microcontroladors PIC[®], obtenint la màxima optimització del compilador amb aquests dispositius. Disposa d'una àmplia llibreria de funcions predefinides, comandaments de *preprocessat* i exemples. A més subministra els *drivers* per a diversos dispositius, com els visualitzadors, convertidors A/D, rellotges en temps real sèrie, EEPROM sèrie, etc. El codi font que genera aquest compilador es pot gravar en el microcontrolador a través de dispositius de gravació i depuració en circuit com el MPLAB[®] ICD2 de *Microchip*.

La versió que s'ha utilitzat és una versió de prova de 30 dies de durada que es pot descarregar directament de la pàgina de CCS, <http://www.ccsinfo.com>. Des d'aquí també es pot descarregar de forma gratuïta els manuals d'ús del compilador i molts exemples.

El llenguatge C de CCS és C estàndard (ANSI C) i a més de les directives estàndard (*#include*, etc.), subministra directives específiques per als microcontroladors PIC[®] (*#device*, etc.); a més inclou altres funcions específiques, com l'acció sobre les E/S a nivell de bit. Es pot integrar en l'entorn de desenvolupament de *Microchip*, el MPLAB[®] IDE.

MPLAB[®] IDE v8.46

MPLAB[®] IDE és un editor de codi gratuït destinat als productes de la casa *Microchip* que s'executa sobre sistemes operatius *Windows*[®]. És una eina que ens permet desenvolupar aplicacions per microcontroladors i controladors de senyal digital. Inclou un eficient compilador de codi ensamblador, el MPASM, l'editor de codi, el MPLAB *Editor* i un potent simulador, el MPLAB SIM *Simulator*. La darrera versió es pot descarregar des de la seva pàgina web, <http://www.microchip.com>. A més dels editors i compiladors, en aquest enllaç es pot trobar una extensa biblioteca de manuals i *data sheets* de tots els seus microcontroladors i altres dispositius.

S'anomena Entorn de Desenvolupament Integrat, o IDE (de l'anglès *Integrated Development Environment*), ja que proporciona un únic entorn integrat per desenvolupar codi per als microcontroladors. MPLAB[®] IDE ens permet editar, compilar, simular, depurar i programar (aquestes dues últimes funcions amb el hardware adequat).

MPLAB[®] ICD2

El MPLAB[®] ICD2 es un dispositiu depurador *en circuit* que serveix d'interfície entre el microcontrolador i el PC. Està dissenyat per treballar amb el sistema operatiu *Windows*[®] i el software de *Microchip* MPLAB[®] IDE. El MPLAB[®] ICD2 és compatible amb la majoria de microcontroladors i controladors de senyal digital amb memòria *flash* de la casa *Microchip*. El depurador suporta en temps real la visió de variables i registres, i es poden afegir *breakpoints* o punt de ruptura del depurador en el codi de programa. El MPLAB[®] ICD2 també s'utilitza per programar els microcontroladors. Aquesta eina es connecta al PC mitjançant el port USB, i no necessita cap alimentació externa. Per connectar el ICD2 amb el microcontrolador s'ha d'utilitzar un connector RJ11.

Les figures que venen a continuació mostren l'esquema electrònic de connexió del ICD2 i una fotografia que permet veure quin és el seu aspecte físic:

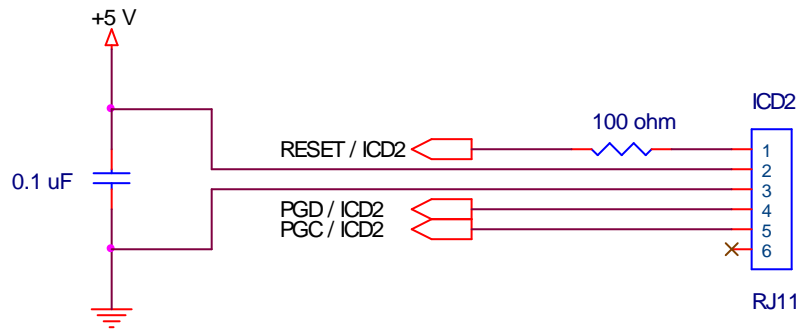


Figura 7.1. Connexió del ICD2 en el circuit
(Font: Elaboració pròpia)



Figura 7.2. MPLAB® ICD2
(Font: Microchip)

7.2. Driver del visualitzador LCD

El compilador C de *CCS* inclou un *driver* per configurar i utilitzar el visualitzador anomenat **lcd.c**. Les funcions que ofereix aquest *driver* es descriuen a continuació:

lcd_init(): Aquesta és la primera funció que s'ha d'executar abans d'utilitzar les altres tres funcions perquè és la de inicialització del visualitzador. Un cop s'ha executat, el visualitzador es configura amb les següents característiques:

- S'esborra el visualitzador.
- Bus de dades de format de 4 bits.
- Dues línies.
- Caràcters de 5x8 punts.
- Mode encès.
- Cursor apagat i sense parpelleig.
- Autoincrement del punter de direccions.
- Sense desplaçament del missatge en el visualitzador.

lcd_gotoxy(byte x, byte y): S'utilitza per posicionar el cursor en el visualitzador. Per exemple, (1,1) fa referència a la primera posició de la primera línia i (1,2) indica la primera posició de la segona línia.

lcd_getc(byte x, byte y): Llegeix el caràcter de la posició (x,y) del visualitzador.

lcd_putc(char s): *S* és una variable de tipus *char* o caràcter. Aquesta funció escriu la variable en la posició del visualitzador a on es troba el cursor. Si a més s'indica:

\f es neteja el LCD

\n el cursor es posiciona a la posició (1,2)

\b el cursor retrocedeix una posició.

El compilador de C ofereix una funció més versàtil per treballar amb el visualitzador:

printf(cstring, values): *String* és una cadena o un *array* de caràcters i *values* és una llista de variables separades per comes. El format de representació de les variables s'especifica amb els modificadors **%nt**, tal i com es fa amb l'ANSI C. Amb aquest format **n** és opcional i pot ser:

1-9: per especificar quants caràcters s'han d'especificar.

01-09: per indicar la quantitat de zeros a l'esquerra.

1.1-9.9: per variables representades en punt flotant.

El modificador **%t** pot indicar:

c	Caràcter
s	Cadena o caràcter
u	Sencer sense signe
d	Sencer amb signe
Lu	Sencer llarg sense signe
Ld	Sencer llarg amb signe
x	Sencer hexadecimal (minúscules)
X	Sencer hexadecimal (majúscules)
Lx	Sencer llarg hexadecimal (minúscules)
LX	Sencer llarg hexadecimal (majúscules)
f	Flotant amb truncat
g	Flotant amb arrodoniment
e	Flotant en format exponencial
w	Sencer sense signe amb decimals inserits

L'ús del modificador `%nt` està molt lligat al tipus de variable. La taula 7.1 mostra amb un exemple la funció d'aquest modificador:

Format	Valor = 0x12	Valor = 0xFE
<code>%03u</code>	018	254
<code>%u</code>	18	254
<code>%2u</code>	18	*
<code>%d</code>	18	-2
<code>%x</code>	12	fe
<code>%X</code>	12	FE
<code>%4X</code>	0012	00FE
<code>%3.1w</code>	1.8	25.4

Taula 7.1. Exemple d'ús del modificador `%nt`
(Font: Elaboració pròpia)

El *driver* `lcd.c` està pensat per treballar amb el PORT D o el PORT B. Per defecte utilitza el PORT D si no s'indica el contrari. Per utilitzar el *driver* s'ha d'utilitzar la directiva de preprocessat `#include`. En el programa de l'analitzador de la xarxa elèctrica trifàsica s'inclouen les següents línies de codi:

```
//----- DIRECTIVES DE CONFIGURACIÓ DEL LCD -----
#define LCD_DATA_PORT D          // S'utilitza el port D
#define LCD_TYPE 2              // LCD de dues línies
#define LCD_TRIS_LOCATION 0x0F  // Direcció registre port D
#include <lcd.c>                 // Driver del visualitzador
```

Les directives `#define` estableixen quin port del microcontrolador s'utilitzarà per el visualitzador i algunes opcions de configuració perquè tot funcioni correctament. Finalment amb la directiva `#include` se li diu al compilador que inclogui el codi del *driver* amb el codi del programa com si es tractés d'una funció més.

El codi del *driver* del visualitzador es descriu a l'Annex II.

7.3. Driver del RTC

Per establir la correcta comunicació amb el DS1307 s’han de conèixer les seves característiques internes, dades que proporciona el fabricant *Dallas Semiconductors* en el *data sheet*. Posseeix una sèrie de registres que permeten configurar el dispositiu i obtenir les dades, les quals es subministren en format BCD. A la figura de la dreta s’especifica el format de representació:

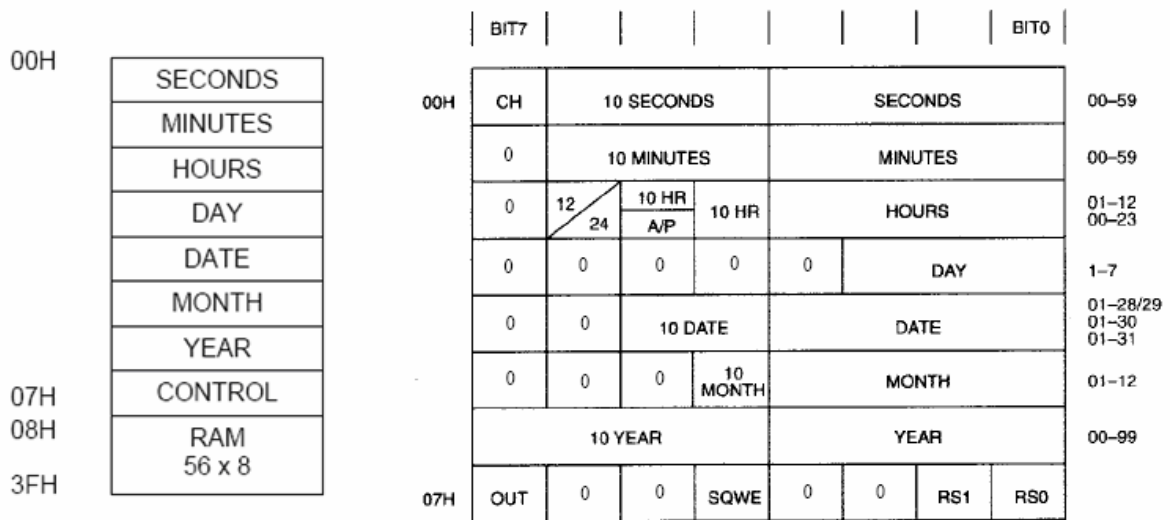


Figura 7.3. Registres amb les adreces i els formats del DS1307
(Font: Data Sheet DS1307)

Segons el fabricant, el cicle d’escriptura ha de començar amb la paraula 0xD0 i el de lectura amb 0xD1. En el cicle d’escriptura, el segon byte es un punter que ha de indicar l’adreça d’inici (en aquest cas 0). Com es pot veure en el codi del *driver*, les funcions de lectura i escriptura segueixen aquestes recomanacions.

El compilador C de *CCS* inclou el *driver* per configurar i utilitzar el rellotge de temps real anomenat **ds1307.c**.

Les funcions que ofereix aquest *driver* es descriuen a continuació i permeten treballar amb aquest dispositiu d'una manera molt senzilla:

ds1307_init(): Aquesta és la primera funció que s'ha d'executar abans d'utilitzar les altres tres funcions perquè és la de inicialització del rellotge de temps real. Un cop s'ha executat el DS1307 es configura de la manera següent:

- Es posa en marxa l'oscil·lador i si el microcontrolador es queda sense tensió aleshores la bateria externa de 3V manté el valor del registre dels segons.
- Es deshabilita la sortida d'ona quadrada.

ds1307_set_date_time(day,mth,year,dow,hour,min,sec): Aquesta funció permet la programació del DS1307 amb la data i l'hora real, per aquest ordre: data, mes, any, dia de la setmana (dow), hora, minut i segon.

ds1307_get_date(day,mth,year,dow): Aquesta funció permet obtenir del DS1307 el valor que tingui en aquell moment de: la data, el mes, l'any i el dia de la setmana (dow). Val a dir que les variables on s'emmagatzemaran aquestes variables han de ser de tipus sencer.

ds1307_get_time(hr,min,sec): Aquesta funció permet obtenir del DS1307 el valor que tingui en aquell moment de: l'hora, el minut i els segons.

Amb aquestes quatre funcions es resol tota la comunicació amb el RTC a través del bus I²C™. El codi del *driver* del RTC es descriu amb detall a l'Annex II.

En el programa de l'analitzador de la xarxa elèctrica trifàsica s'ha d'incloure la següent línia de codi, perquè el compilador que inclogui el codi del *driver* amb el codi del programa com si es tractés d'una funció més:

```
// Driver de funcionament del DS1307
#include <ds1307.c>
```


7.4. Codi del programa principal

El programa principal ha de gestionar l'analitzador de la xarxa elèctrica trifàsica de manera que constantment es realitzin les tasques següents:

- Transmissió al PC dels consums de les tres fases a través del port RS232.
- Visualització dels consums per el *display* LCD cada 200ms.
- Transmissió al PC del nombre de mostres per segon a través del port RS232.
- Actualització dels valors màxims dels consums de les tres fases.
- Actualització dels acumulats cada 5 minuts per el càlcul del consum promig.
- Càlcul del consum promig de les tres fases si és mitjanit.
- Visualització de la data i l'hora si aquesta opció es troba seleccionada.
- Posar en marxa les accions d'alarma si la opció de reconeixement de l'alarma es troba activa.

Després de què s'ha polsat el botó de *reset* o després de donar tensió, el programa configurarà els elements que intervenen en el funcionament de l'analitzador. A continuació el programa realitzarà les tasques que s'han descrit anteriorment de manera cíclica, en un bucle infinit, sempre que no s'activi cap interrupció. En cas contrari, el microcontrolador executarà la rutina de servei de interrupció corresponent. El codi de les RSI ha de ser prou ràpid i eficient com per optimitzar l'ús del microcontrolador envers les tasques principals.

A l'hora de programar el codi van sorgir algunes de les limitacions del microcontrolador i del llenguatge de programació utilitzat. Hi ha certes funcions que es podrien haver programat utilitzant altres fonts d'interrupció (per exemple, el refresc del *display* LCD es podria haver programat amb la interrupció del TIMER0), però es va trobar l'inconvenient de la pila d'*stack* del microcontrolador PIC16F877A, que només té vuit nivells de profunditat.

La figura 7.4 següent mostra com s'organitza la memòria del microcontrolador, incloent-hi la pila d'*stack*, el vector d'interrupció i la paginació de la memòria de programa.

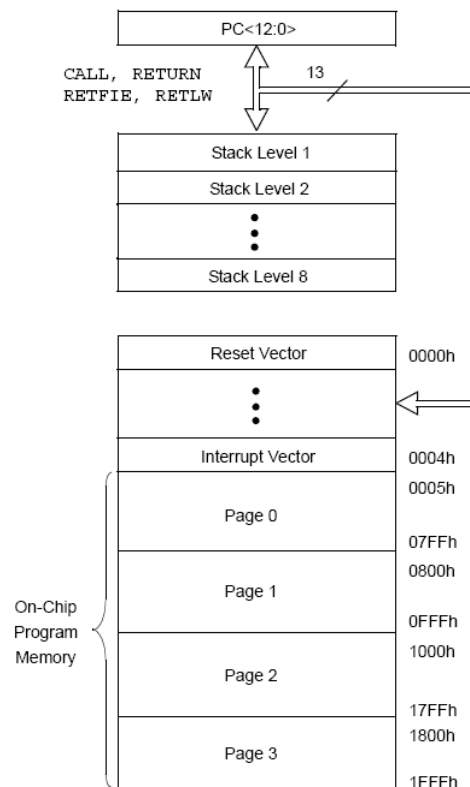


Figura 7.4. Organització de la memòria del PIC16F877A
(Font: Data Sheet PIC16F877A)

La pila d'*stack* del PIC16F877A

La pila d'*stack* de la família de microcontroladors PIC16F87XA només té una profunditat de vuit nivells. És una pila gestionada totalment per *hardware*, i no es pot llegir ni escriure cap dels registres. A més la pila d'*stack* treballa com un *buffer* circular. És a dir, quan s'ha omplert el nivell número vuit, el novè accés a la pila automàticament sobreescriu el valor que es troba emmagatzemat en el primer nivell. El desè accés sobreescriu el segon nivell, i així successivament.

Això és realment un problema greu perquè el microcontrolador no dona cap avís al programador (a través d'un *flag*, per exemple) de què la pila s'ha corromput.

Què succeeix quan el PC perd l'adreça de retorn a certa part del programa? Que el programa es penja i l'aplicació entra en un funcionament erroni.

A més, si es programa en C el programador perd transparència amb la màquina i també perd el control de la gestió de l'*stack*. (Aquesta qüestió es va explicar amb detall en el capítol 5, en l'apartat 5.7 dedicat a les interrupcions). En una aplicació on es criden funcions dins de funcions, o es criden funcions dins de les rutines d'atenció a les interrupcions, es pot arribar a superar el nivell de profunditat de l'*stack* i produir-se un fatal i imprevisible desenllaç.

De fet això és el que va succeir durant les primeres proves de programació. La pila del microcontrolador es va saturar a tal punt que el sistema no era capaç de prendre les mesures dels canals analògics ni enviava mostres al PC per el port RS232. És per aquesta raó que certes accions s'han de realitzar per *polling* durant l'execució del programa principal, amb sentències de tipus condicional *if*. També es va haver de treure de la rutina de servei de la interrupció del teclat la part del reconeixement de l'alarma i la visualització de la data i l'hora.

Si el microcontrolador tingués dues fonts d'interrupció externes i una pila d'*stack* de més profunditat, aleshores es podria haver connectat el senyal d'ona quadrada del RTC al nivell d'interrupció de menys prioritat.

Adequadament programat, aquest senyal podria establir la freqüència d'emmagatzematge dels acumulats dels consums de cada fase, o els càlculs del consum promig o la freqüència de refresc del visualitzador si es troba en el mode de representació de la data i l'hora. Sens dubte, quan el *hardware* augmenta la seva potència el *software* se simplifica notablement.

Aleshores, com es pot justificar la selecció d'aquest llenguatge de programació si afavoreix l'aparició d'errors imprevisibles i la manca de control sobre certes parts del procés? No hagués estat millor utilitzar un microcontrolador més potent?

La resposta a les dues preguntes és immediata: per una banda el microcontrolador ja ofereix prestacions més que suficients per a la realització d'aquest projecte, aleshores, per què gastar-se més diners en un altre microcontrolador més potent, si aquest ja cobreix les necessitats de l'aplicació?

En quant al llenguatge de programació, sí que s'ha de reconèixer que la programació en el llenguatge assemblador de *Microchip* és més eficient i permet més control de certes parts de l'aplicació, no obstant, la facilitat d'aprendre i utilitzar un llenguatge d'alt nivell (amb el poc temps que hi ha per desenvolupar un PFC en un quadrimestre) compensa o justifica aquests inconvenients.

Per tant, la clau del codi de programa rau en una adequada *estratègia de programació*. A la figura 7.6 es pot veure l'ordinograma que representa el mètode per resoldre el codi de programa de l'anàlitzador de la xarxa elèctrica trifàsica. A l'Annex II es descriu amb detall el codi del programa principal.

L'ús de les variables globals també augmenta l'eficiència del codi però té l'inconvenient que es poden produir errors inesperats si no se sap en cada moment quin és el valor exacte de la variable en qüestió. La figura següent mostra part del codi del programa des de l'editor del MPLAB[®] IDE:

```

133 //----- DECLARACIÓ DE FUNCIONS -----
134
135 #SEPARATE void actualitza_promig(void);
136 #SEPARATE void calcul_promig(void);
137 void AD_isr(void); // RSI interrupció teclat
138 void EXT_isr(void); // RSI interrupció EOC del ADC
139 short verifica_entrada_teclat(int minia,int maxia,long numero);
140 void validar_numero(int key);
141 void display_misatge(void);
142
143
144 //***** PROGRAMA PRINCIPAL *****
145 //*****
146
147 void main()
148 {
149     disable_interrupts(GLOBAL); // Es desconnecten les interrupcions
150
151     TRISB = 0xFF; // 8 bits del PORT B com a entrades
152
153     output_low(PIN_E0); // Apaga el LED d'alarma
154     output_low(PIN_B1); // Desactiva el relé d'alarma
155
156     lcd_init(); // LCD en format de bus de dades de 4 bits,
157                // dues línies, caracters de 5x8 punts,
158                // mode encès, cursor apagat i sense pampallugues,
159                // amb auto-increment del punter de direcció
160                // i sense desplaçament del display real
161
162     lcd_gotoxy(1,1); // Es posiciona el cursor del LCD
163     printf(lcd_putc, "\f Analitzador"); // Missatge inicial
164     printf(lcd_putc, "\n de la xarxa"); // de l'aplicació
165     delay_ms(4000); // Retard de 5 segons
166
167     lcd_gotoxy(1,1); // Es posiciona el cursor del LCD
168     printf(lcd_putc, "\f electrica"); // Segona part del missatge inicial
169     printf(lcd_putc, "\n trifasica");
170     delay_ms(4000); // Retard de 5 segons
171     lcd_putc('\f'); // S'esborra el LCD
172
173     // Es programa el rellotge i el calendari del RTC amb els valors per defecte
174     ds1307_init();
175     delay_ms(100);
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Figura 7.5. Codi del programa principal des del MPLAB[®] IDE

(Font: Elaboració pròpia)

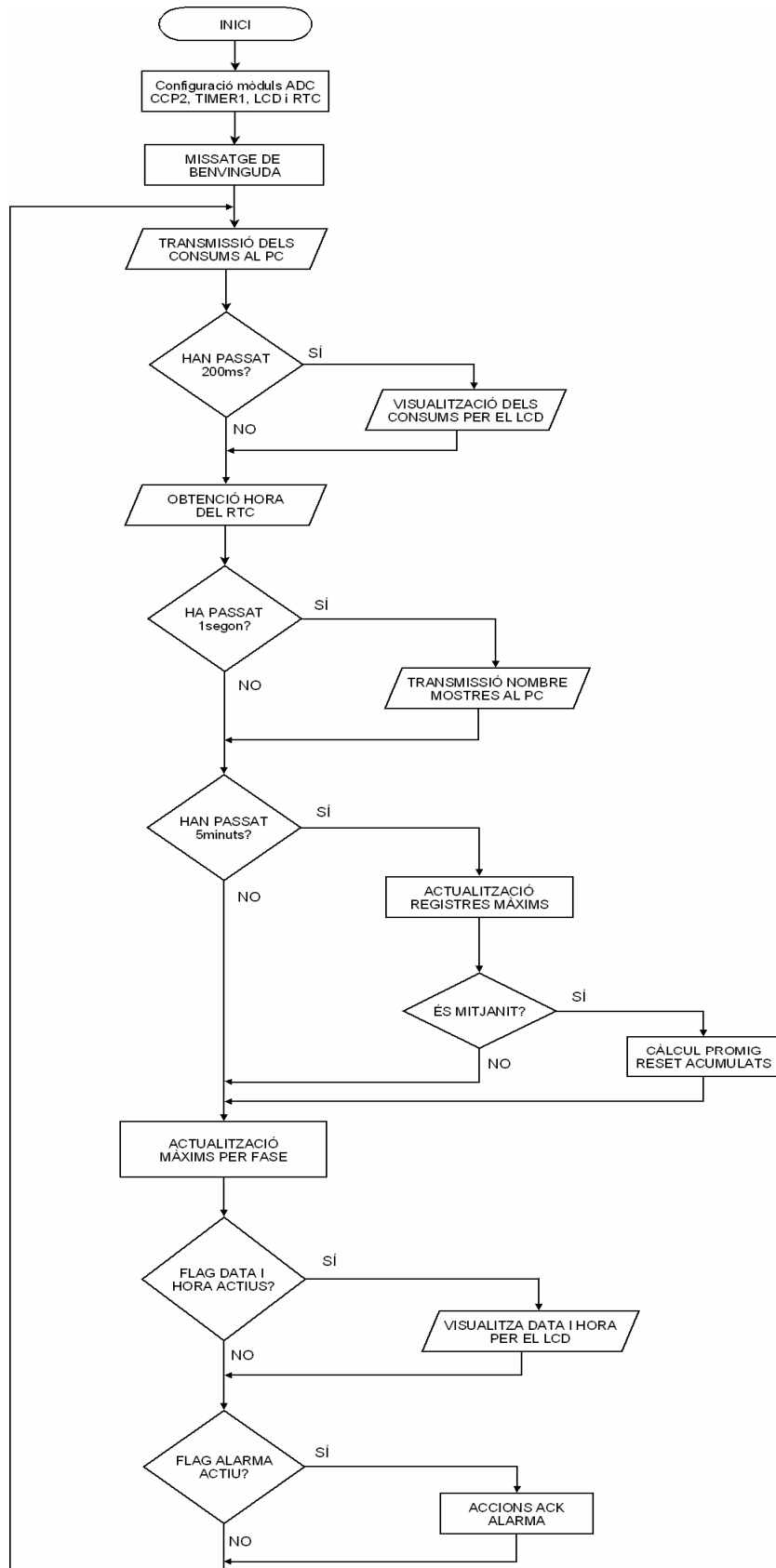


Figura 7.6. Ordinograma programa principal

(Font: Data Sheet PIC16F877A)

7.5. Codi de les interrupcions

En capítols anteriors ja s'ha comentat que el codi les rutines d'atenció a les interrupcions és una de les parts més importants del projecte. La seva eficiència serà determinant per assolir els objectius d'aquest projecte.

7.5.1. Rutina servei interrupció teclat

Aquesta ha estat la part del programa més difícil de desenvolupar. La gestió dels menús des de la rutina de servei a la interrupció del teclat sense que afecti al rendiment del sistema no ha estat una tasca senzilla.

Les diferents tecles del teclat estan assignades per realitzar diferents funcions, tal i com es descriu a continuació:

- A** → Accés al menú principal
- B** → Visualització de la data i l'hora
- C** → Visualització del llindar d'alarma programat
- D** → Reconeixement de l'alarma (*ACKNOWLEDGEMENT ALARM*)
- *** → ESCAPE
- #** → ENTER

Els números de l'1 al 6 serveixen per seleccionar els menús secundaris disponibles des del menú principal o com a números sencers en els altres casos, amb el pes o valor assignat per el símbol de la tecla. Els números del 7 al 9 sempre s'utilitzen com a números sencers. La gestió del codi de servei al teclat i dels menús s'ha realitzat de la següent manera:

- Cada tecla pulsada només executa una acció alhora:
 - Visualització de variables per el LCD.
 - Operacions matemàtiques senzilles, com una suma.
 - Assignació d'un valor a una variable.

- La transició entre menús es fa a través de la tecla ENTER i per sortir de qualsevol menú en qualsevol nivell es fa amb la tecla ESCAPE.
- Es pot accedir als diferents menús amb la tecla **A**. Fora del menú les tecles numèriques no realitzen cap acció.
- En el mode de funcionament normal (el de visualització dels consum en temps real) només es troben actives les tecles **A** (accés al menú principal), **B** (visualització de la data i l'hora), **C** (visualització del llindar d'alarma programat) i **D** (accions de reconeixement de l'alarma només quan s'ha produït una alarma).
- Quan se selecciona la opció de visualitzar la data i l'hora en realitat només s'activa un *flag*, i la gestió real d'aquesta opció es fa des del programa principal. En l'apartat anterior es justifica aquesta solució. Succeeix el mateix per la tecla **D** de reconeixement de l'alarma.

La clau de la rutina es troba en què només es produeix una acció cada vegada que es prem una tecla. Una variable emmagatzema en quin punt es troba el menú actiu i a través d'estructures d'alternativa múltiple (sentència *switch...case*) només s'executa l'acció que toqui en aquell moment. Així el microcontrolador pot dedicar la major part del temps a l'execució de les tasques principals. El menú principal (que s'accedeix polsant la tecla **A**) ofereixen les següents opcions:

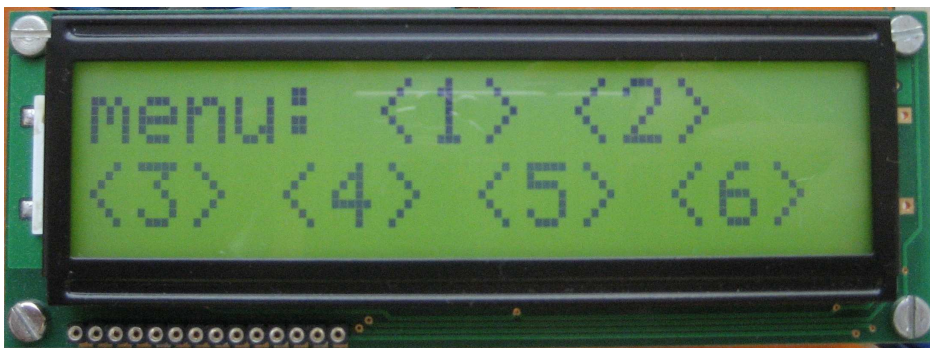


Figura 7.7. Opcions menú principal
(Font: Elaboració pròpia)

<1> Programació de la data: Aquest menú permet introduir per el teclat la data, el mes, l'any i el dia de la setmana actuals. Amb la tecla ENTER i les tecles numèriques es poden programar tots els paràmetres relacionats amb la data.



Figura 7.8. Primera opció menú principal
(Font: Elaboració pròpia)

<2> Programació del rellotge: Aquest menú permet programar el rellotge: l'hora, els minuts i els segons. En els menús <1>, <2> i <3> es verifica que la dada introduïda sigui coherent (no es pot introduir el mes 14 ni el minut 70, per exemple). A més sempre ofereix una ajuda sobre el format, dient si la dada esperada és de dues o tres xifres.

<3> Programació del llindar d'alarma: en aquest menú es pot programar el llindar de consum elèctric a partir del qual s'engegaran les accions d'alarma. El programa verifica que el número introduït per el teclat estigui dins de marges, entre 0 i 200A. S'han d'introduir tres xifres, polsant la tecla ENTER per validar-les.

<4> Visualització del consum promig de cada fase: en aquest menú es visualitza en el *display* LCD el consum promig del dia anterior per fase. El càlcul del promig es fa a mitjanit amb les dades que s'han obtingut durant el dia. Aquest registre té una durada de un dia. Si l'endemà no es revisen aquests registres es perden amb el promig del dia següent.

<5> Visualització dels consums màxims: en aquest menú es visualitza en el *display* LCD el consum màxim que cada fase ha assolit fins aquell moment.

<6> **Reset del consums màxims:** Quan se selecciona aquesta opció s'esborren els registres dels consums màxims per fase. A partir d'aquest instant el microcontrolador enregistrarà els nous màxims.

La fotografia de la figura 7.8 il·lustra com els missatges dels diferents menús animen a l'usuari d'una manera agradable a realitzar les accions pertinents.

La figura 7.9 mostra l'ordinograma que representa el mètode per resoldre el codi de la rutina d'atenció a la interrupció del teclat, amb un primer nivell de detall. En canvi les figures 7.10 i 7.11 mostren els ordinogrames de diferents accions amb més nivell de detall, per exemple el de les accions associades amb la tecla **A** i la tecla **5** respectivament.

Com es pot veure en el codi de la rutina, el primer que es fa és llegir del PORT B el codi de la tecla pulsada. La línia de codi **key = PORTB & 0x1E** assigna a la variable *key* només el valor dels bits <RB0:RB4> del PORT B. La resta de bits no es tenen en compte per l'acció de la funció AND. El codi de la RSI del teclat es descriu a l'Annex II.

També es pot veure que s'han utilitzat tres funcions addicionals per automatitzar aquelles accions repetitives, com ara:

- Verificar que el número introduït es troba dins de marges. Els paràmetres de pas a la funció són el número que s'ha d'analitzar, a més dels límits mínim i màxim.
- Validar el número que s'ha introduït per el teclat. Aquesta funció verifica que només es visualitzin els números si el sistema es troba dins d'un menú que requereix l'entrada d'un número. Aquest funció impedeix que apareguin en el *display* LCD els números de les tecles pulsades en un moment que no li correspon.
- Visualització del missatge "*Aquest número no és correcte*" si el número introduït per el teclat es troba fora dels marges de la variable a on anava destinat.

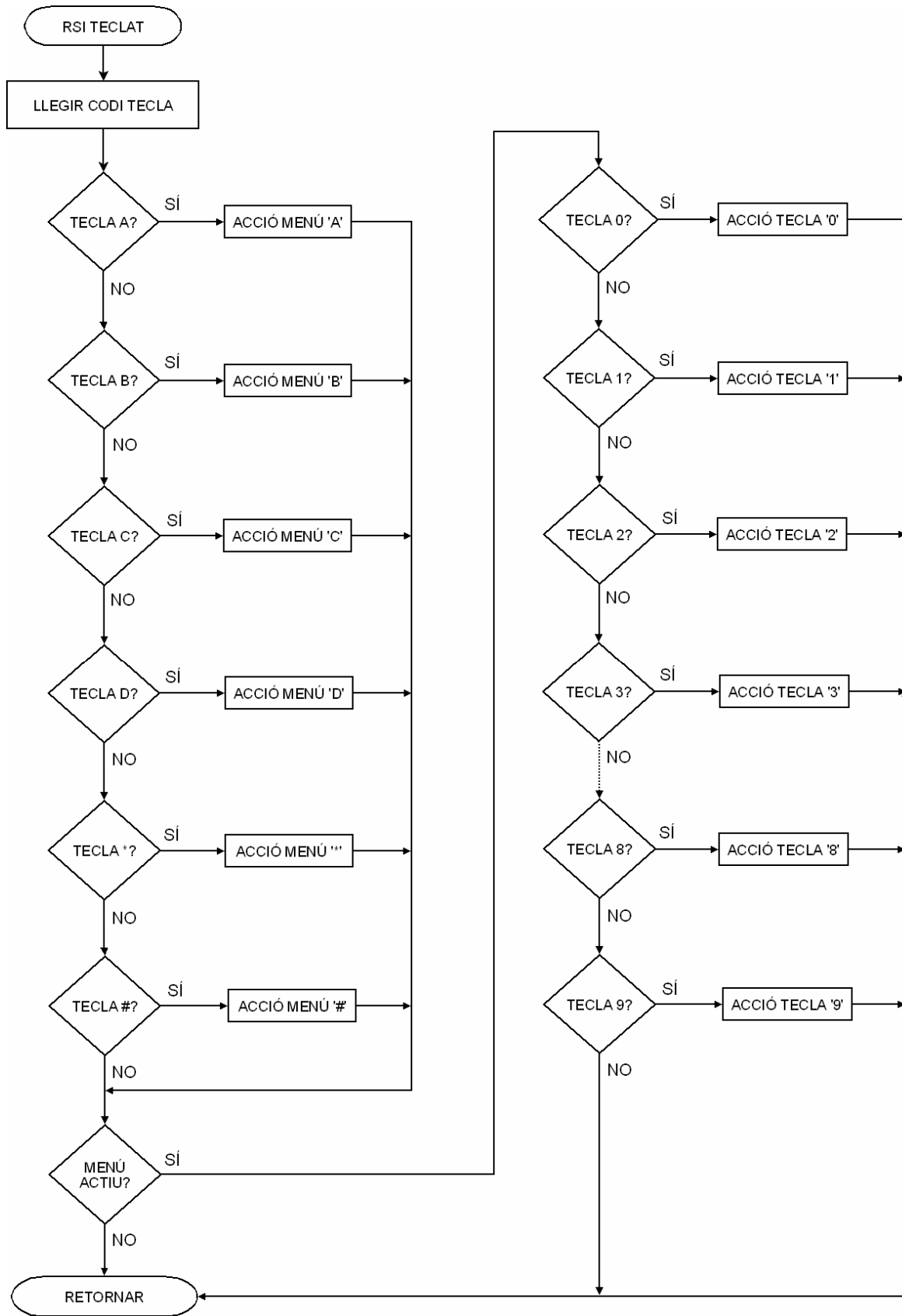


Figura 7.9. Ordinograma RSI teclat

(Font: Elaboració pròpia)

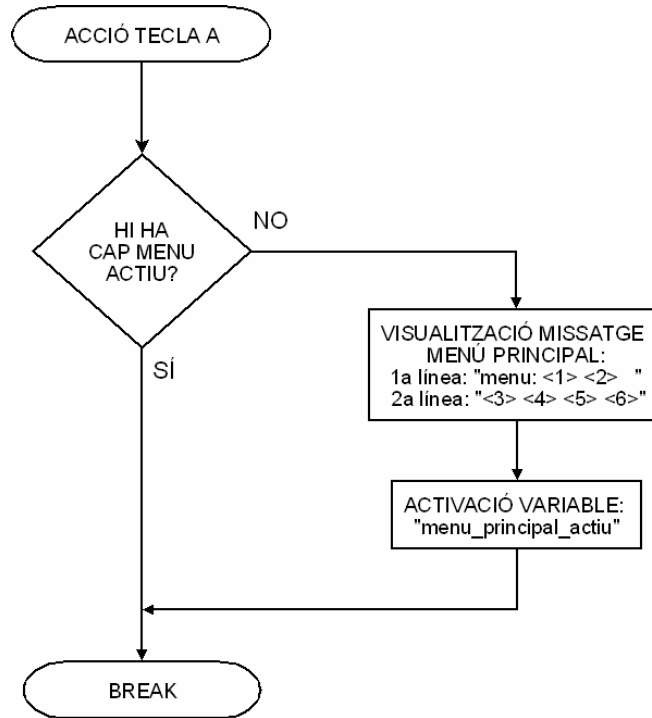


Figura 7.10. Ordinograma acció tecla A polsada
(Font: Elaboració pròpia)

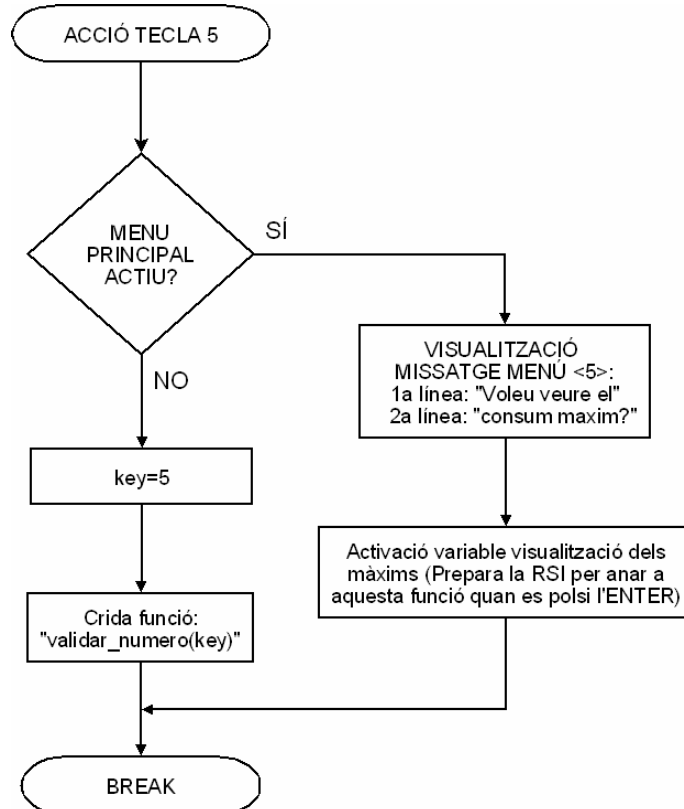


Figura 7.11. Ordinograma acció tecla 5 polsada
(Font: Elaboració pròpia)

7.5.2. Rutina servei interrupció ADC

La rutina de servei de la interrupció del ADC es genera automàticament quan el convertidor té a punt una dada d'un canal analògic. Ja s'ha vist en capítols anteriors com s'automatitzava el convertidor A/D. En aquest apartat s'estudiarà el codi d'aquesta rutina, una de les més importants: és qui vigila l'estat dels consums i qui ha d'engegar les accions d'alarma, preparant el sistema per respondre a aquest esdeveniment.

El temps de mostreig del convertidor A/D està programat per defecte a 2ms per fase, això vol dir que cada 666µs aproximadament aquesta rutina s'executarà.

Aquesta rutina ha d'atendre les següents tasques cada vegada que s'executi:

- Llegir el valor de la conversió del canal analògic actiu.
- Analitzar si el valor mesurat supera el llindar programat.
- Si es produeix una alarma s'han de d'inhibir tots els menús perquè res no pugui interrompre l'emmagatzematge de les 50 mostres consecutives a l'alarma.
- També s'han d'executar les accions d'alarma: encendre el LED, activar el relé i el *flag* d'alarma.
- Emmagatzemar de les 50 mostres consecutives a l'alarma per fase.
- Preparar el següent canal analògic que s'ha de llegir (la fase següent).
- Actualitzar la variable que conté el nombre de mostres per fase.
- Actualitzar la variable de refresc del visualitzador LCD.

La figura 7.12 mostra l'ordinograma que representa el mètode per resoldre el codi de la rutina d'atenció a la interrupció del ADC amb un nivell de detall de programació (és a dir, que fins i tot es podria transcriure a codi C directament).

El compilador C de CCS ofereix una sèrie de funcions que faciliten la tasca de programar la lectura dels canals analògics:

setup_adc(mode): Serveix per configurar la font de rellotge del mòdul convertidor A/D i es correspon amb els bits <7:6> del registre ADCON0.

setup_adc_ports(valor): Es defineixen quin bits s'utilitzaran com a entrades analògiques i es correspon amb els bits <7:6> del registre ADCON0.

set_adc_channel(canal): Selecció del canal analògic i es correspon amb els bits <5:3> del registre ADCON0.

valor=read_adc(mode): Llegeix el resultat del canal analògic que ha estat seleccionat anteriorment. *Valors* ha de ser una variable de 16 bits de tipus long, tot i què només seran útils 10 bits. Aquesta funció admet tres modes de funcionament

- **ADC_START_AND_READ:** Si no s'especifica res és la opció per defecte. Permet iniciar i llegir el convertidor A/D.
- **ADC_START_ONLY:** Només inicia la conversió.
- **ADC_READ_ONLY:** Només llegeix els registres del convertidor A/D

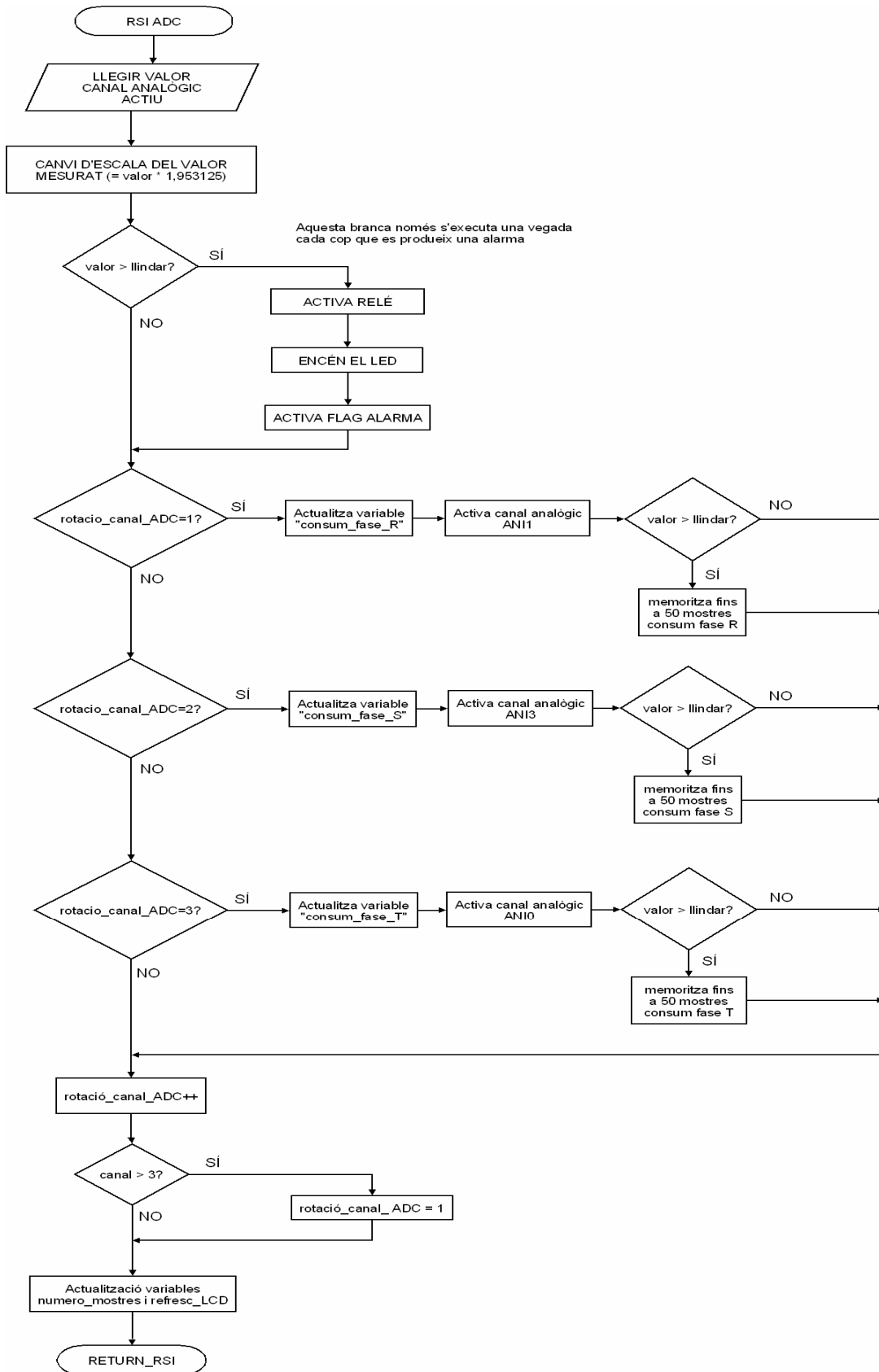


Figura 7.12. Ordinograma RSI ADC

(Font: Elaboració pròpia)

El convertidor A/D proporciona un valor decimal (entre 0 i 1024 perquè és una paraula de 10 bits) proporcional al senyal analògic d'entrada (0 ÷ 5V). Per canviar les unitats de representació s'ha de multiplicar el valor obtingut per una constant, tal i com es pot veure en el codi de la RSI del convertidor A/D (veure Annex II):

```
valor_ADC = read_adc(ADC_READ_ONLY) * 0.1953125;
```

De on surt aquesta constant? Si s'entén que encara que el convertidor A/D mesura volts en realitat estem mesurant ampers, en un rang de 0 – 200A, aleshores surt de dividir el fons d'escala de la mesura real (els ampers) entre el fons d'escala dels punts del convertidor:

$$CONSTANT = \frac{\text{Valor_fons_escala_mesura}}{\text{Valor_fons_escala_convertidorA/D}} = \frac{200}{1024} = 0,1953125$$

Una altre part del programa que s'ha d'explicar és l'ús de la variable *refresc_LCD*. Per què serveix aquesta variable? El seu nom és força descriptiu. S'utilitza per calcular el temps de refresc del *display* LCD.

En les primeres versions del programa aquest refresc era d'un segon, aprofitant que cada segon s'envia al PC el número de mostres. No obstant es va veure que era massa temps: es perdia certa resposta del sistema, encara que només fos des del punt de vista visual.

Per altra banda, si no existia cap temps de refresc i el microcontrolador treia per el visualitzador constantment els consums es produïa un ball de números intel·ligible. Per tant era obligat incloure d'alguna manera un temps de refresc del *display* LCD, ni que fos tant petit com 200 ó 300ms.

Tal i com es va explicar en capítols anteriors el microcontrolador té certa limitació que impedeix utilitzar una altre interrupció, per exemple la del TIMER0, per automatitzar el refresc del *display* LCD.

Aleshores el que es fa és el següent: si el microcontrolador executa la RSI del convertidor A/D cada $666\mu\text{s}$, quantes vegades s'ha d'incrementar en una unitat una variable perquè el visualitzador tingui un refresc de 200ms? La resposta és: unes 300 vegades, més o menys.

$$\text{refresc_LCD} = \frac{200\text{ms}}{666\mu\text{s}} \approx 300$$

I això és el que es fa. En el programa principal, quan s'arriba al codi de refresc del *display* s'analitza si la variable *refresc_LCD* supera les 300 unitats. Si la resposta és certa vol dir que han transcorregut 200ms aproximadament.

Una de les millores que es pensa introduir és incloure un altre menú on es pugui seleccionar el temps de refresc del *display* LCD entre dues o tres opcions.

Per canviar el temps de refresc només s'ha de modificar aquest valor numèric, el que es troba ressaltat:

```
// Visualització per LCD dels consums cada 200ms
if ((refresc_lcd>300) && (menu_principal_actiu==FALSE) &&
(menu_data_i_hora_actiu==FALSE) && (menu_llindar_actiu==FALSE))
{
    // Posiciona el cursor a la primera posició del LCD
    lcd_gotoxy(1,1);

    // Missatge per a la primera línia del LCD
    printf(lcd_putc," R=%3uA S=%3uA ",consum_fase_R,consum_fase_S);

    // Missatge per a la segona línia del LCD
    printf(lcd_putc,"\n T=%3uA CONSUM ",consum_fase_T);

    // Actualització de la variable refresc
    refresc_lcd = 0; ← Important aquesta línia per tornar a entrar
}
```


8. CONCLUSIONS I MILLORES

Aquest projecte ha estat, sens dubte, un bon punt i final a sis anys de treball i esforç per aconseguir un dels reptes personals que més s'han allargat en el temps: la consecució d'un títol universitari.

El fet de desenvolupar un projecte per una aplicació real ha fet més atractiu el repte de realitzar aquest treball de final de carrera: l'*Analitzador de la Xarxa Elèctrica Trifàsica de Propòsit General* ha complert amb escreix amb les expectatives i els requisits tècnics que s'havien marcat. Allò que fa uns mesos només era una idea avui ja és una realitat.

No ha estat un projecte fàcil. S'ha començat des de zero gairebé amb tot: la selecció dels sensors de corrent, el microcontrolador, el llenguatge de programació, etc. La part més difícil ha estat l'aprenentatge de l'entorn de programació, el MPLAB[®] IDE de *Microchip* i el llenguatge de programació en C de *CCS*. Només per aconseguir comunicar-se amb el PIC16F877A i que s'encengués el LED es va necessitar una setmana! No obstant, una vegada es va conèixer una mica més l'editor i el llenguatge de programació es va veure que oferien unes eines que facilitaven moltíssim el desenvolupament del projecte.

La selecció del microcontrolador i del llenguatge de programació (el C de *CCS*) han esdevingut clau en l'èxit del projecte. Per una banda, les potents prestacions del microcontrolador han simplificat molt el *hardware*; i per l'altre, el compilador incorpora *drivers* i funcions que faciliten la programació. Si s'hagués de programar en ensamblador segurament avui l'analitzador encara no funcionaria del tot. La casa *Microchip* proveeix un excel·lent entorn de desenvolupament totalment gratuït. A més ofereix una quantitat ingent de documentació a la seva pàgina web que han estat de gran ajuda en el projecte. I molts, molts programadors d'arreu del món utilitzen aquest microcontrolador, el PIC16F877A. La xarxa d'Internet és a vessar d'exemples i de les seves experiències amb aquest xip.

Els resultats del projecte han estat molt bons. La interfície visual amb el *display* LCD ha estat un encert. Juntament amb el teclat, el *display* LCD dona cert valor afegit al projecte ja que fa més agradable i atractiu l'ús de l'analitzador de la xarxa elèctrica trifàsica.

També ha estat un encert el muntatge del prototip amb el mètode de fabricació de circuits electrònics *wire-wrapping*. Aquest mètode consisteix en enrotllar en els terminals dels components un cable molt primet amb una eina especial. El cablejat és molt ràpid, així com la correcció d'errors, amb un bon acabat final.

No han sorgit problemes especials en les proves, llevat d'alguns petits errors en el cablejat. El circuit de reset, l'oscil·lador, el rellotge de temps real, el *key-encoder* i la resta de components, tots han funcionat a la primera. El visualitzador LCD ha donat mica més de guerra perquè el *data sheet* no aportava massa informació i algunes de les funcions s'han descobert per el mètode d'assaig i error.

El present projecte ha estat un exercici molt útil des del punt de vista didàctic perquè ha permès dur a la pràctica molts dels coneixements que s'han anat adquirint al llarg dels estudis d'enginyeria. Un dels conceptes més interessants és el de les interrupcions dels processadors. En aquest projecte era fonamental entendre bé què és una interrupció i com funcionen. És una de les parts que més s'ha hagut de treballar. Es va haver d'estudiar amb molt deteniment el funcionament particular de les interrupcions del PIC16F877A i la gestió que fa aquest microcontrolador del vector d'interrupció, les màscares de les interrupcions i també la gestió de la pila d'*stack*.

Una de les millores que es poden proposar és integrar en un sol circuit imprès el circuit analògic i el digital, i simplificar l'ús de les fonts d'alimentació. En aquesta línia es podria estudiar la possibilitat d'incloure OP AMP d'alimentació unipolar.

Finalment el *display* LCD es substituiria per un model similar, amb les mateixes característiques però més petit i de menys consum. Així l'alimentació de la retroil·luminació es podria fer a través d'una sortida de PWM del microcontrolador i no amb una font externa.

També es podria incloure un menú des de on es pogués escollir el temps de refresc del *display* LCD entre dues o tres opcions.

9. REFERÈNCIES

9.1. Bibliografia

ANGULO, J. M^a. (2003). *Microcontroladores PIC. Diseño práctico de aplicaciones*. Primera parte. Tercera edición. Mc Graw-Hill. Madrid

ANGULO, J. M^a.; ROMERO, S.; ANGULO, I. (2004). *Diseño práctico con microcontroladores*. Ed Thomson. Madrid

CIRCUTOR. (2005). *Transformadores de medida y shunts*. Viladecavalls

Custom Computer Services, Inc. (1994-2010). *C Compiler Reference Manual*. Brookfield

FRAILE, J. (2003). *Máquinas eléctricas*. Quinta edición. Mc Graw-Hill. Madrid

GARCÍA, E. (2009). *Compilador C CCS y Simulador PROTEUS para Microcontroladores PIC*. Segunda edición. Ed. Marcombo. Barcelona

HORRILLO, J. (1995-2009). *Informática Industrial: Estructura de computadores*. EUPMT. Mataró

HORRILLO, J. (2009). *Informática Industrial: Introducció a la programació en C*. EUPMT. Mataró

KOON, W. (2001). *Nuevos desarrollos en sensores de corriente en medidores de estado sólido*. Analog Devices, Inc. Norwood, EE.UU

Microchip Technology Inc. (2003). *PIC16F87XA Data Sheet*. DS39582B

Microchip Technology Inc. (2009). *8-bit PIC[®] Microcontrollers Solutions*. DS39630F

Microchip Technology Inc. (2009). *MPLAB IDE User's Guide with MPLAB Editor and MPLAB SIM Simulator*. DS51519C

MONTÉ, A.; SAFONT, R. (2004). *Electricitat i electrònica*. EUPMT. Mataró

PALACIOS, E.; REMIRO, F.; LÓPEZ, L. J. (2005). *Microcontrolador PIC16F84. Desarrollo de proyectos*. Segunda edición. Ed. Ra-ma. Madrid

PALLÁS, R. (2003). *Sensores y acondicionadores de señal*. Cuarta edición. Ed. Marcombo. Barcelona

PITARCH, J. (2009). *Mòdul microcontrolador universal*. PFC-EUPMT. Mataró

SAFONT, R. (2006). *Electrònica Analògica (Apunts)*. EUPMT. Mataró

9.2. Webgrafia

<http://www.ccs.com>, Desenvolupadors de software i aplicacions, especialitzats en els microcontroladors PIC[®]

<http://www.circutor.es>, Fabricant de sensors, equips de mesura i transformadors

<http://es.rs-online.com/web>, RS-Amidata, distribuïdor de productes electrònics a nivell mundial

<http://es.farnell.com>, Farnell, distribuïdor de productes electrònics a nivell mundial

<http://www.microchip.com>, Fabricant dels microcontroladors PIC[®]



Escola Universitària
Politécnica de Mataró

Enginyeria Tècnica Industrial: Especialitat Electrònica Industrial

**ANALITZADOR DE LA XARXA ELÈCTRICA
TRIFÀSICA DE PROPÒSIT GENERAL**

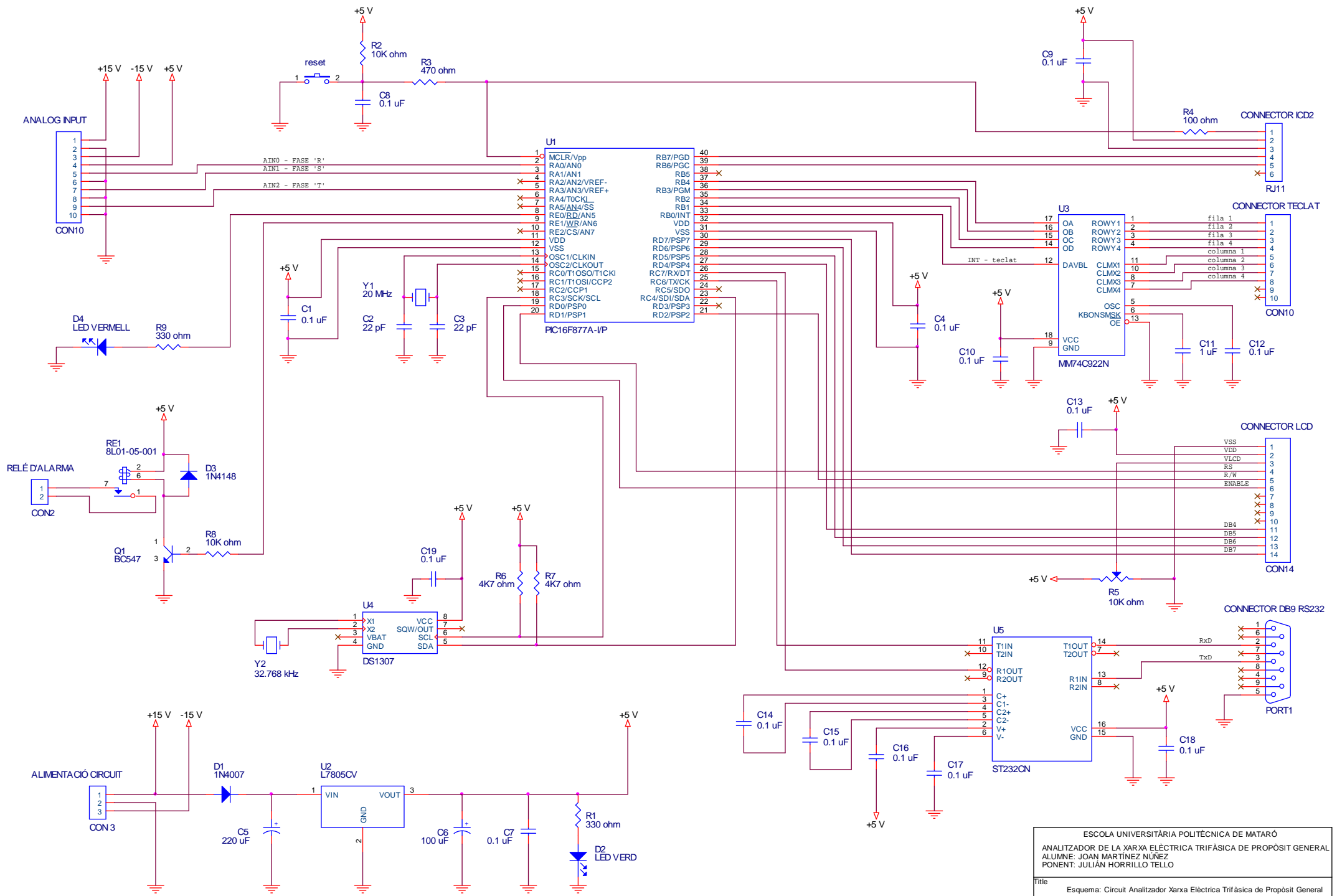
B – Plànols

**JOAN MARTÍNEZ NÚÑEZ
JULIÁN HORRILLO TELLO**

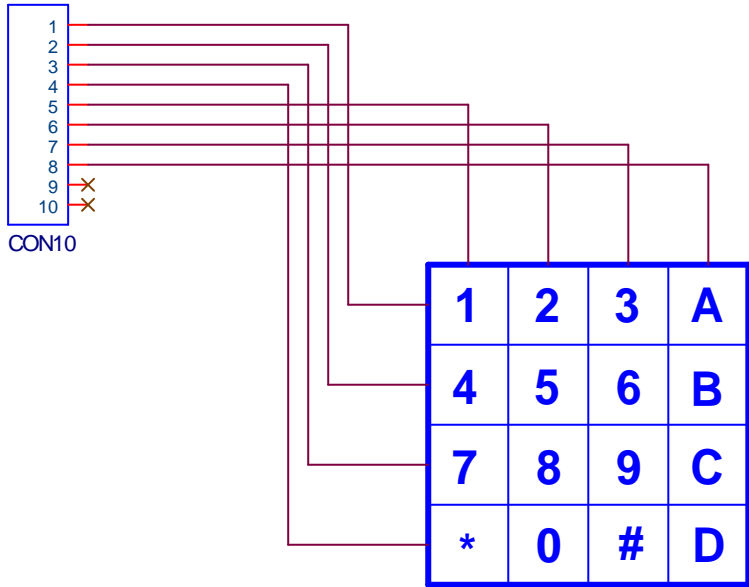
PRIMAVERA 2010

ÍNDEX

1. CIRCUIT PRINCIPAL MICROCONTROLADOR	1
2. CONNEXIÓ LCD I TECLAT	2
3. CIRCUIT ANALÒGIC ADAPTADOR DE NIVELLS.....	3
4. CIRCUIT ANALÒGIC CONVERTIDOR CORRENT - TENSIÓ	4
5. CONNEXIÓ TRANSFORMADORS DE CORRENT	5



CONNECTOR TECLAT



Teclat matricial 4x4

POWERTIP PC1602LRS-LWA

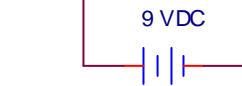


CONNECTOR LCD



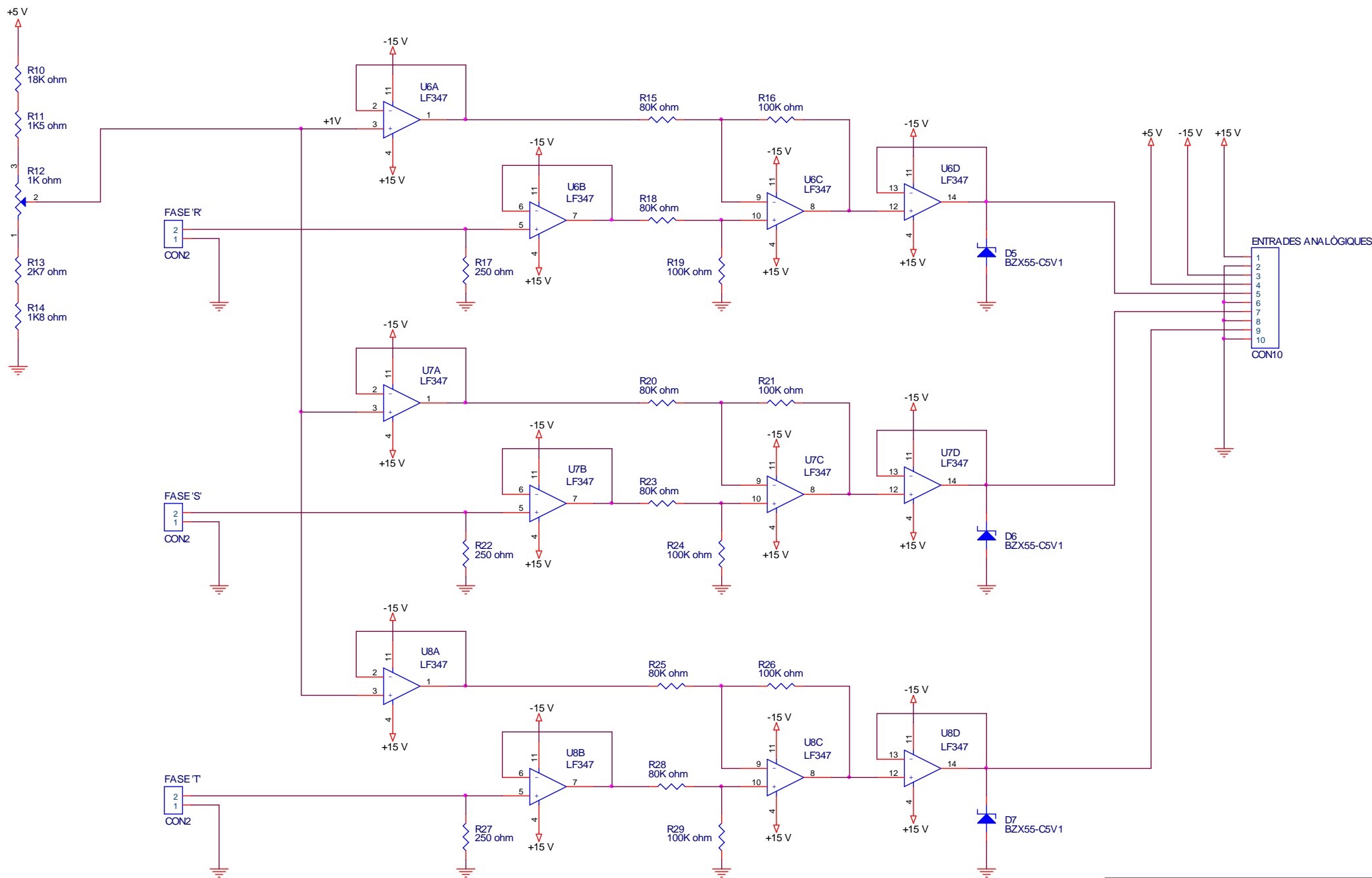
CON14

10R
3W

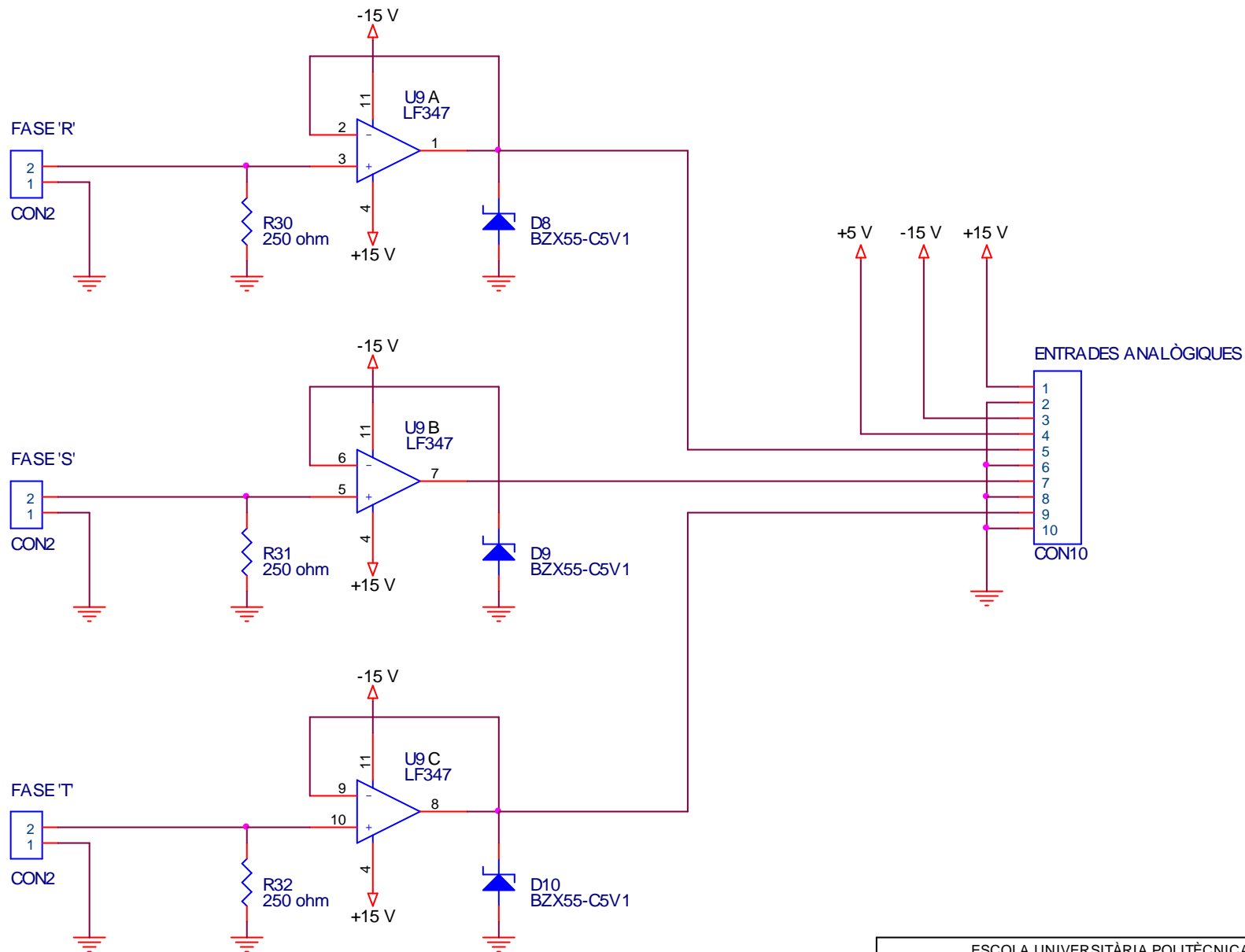


FONT D'ALIMENTACIÓ EXTERNA
PER A LA RETROILUMINACIÓ

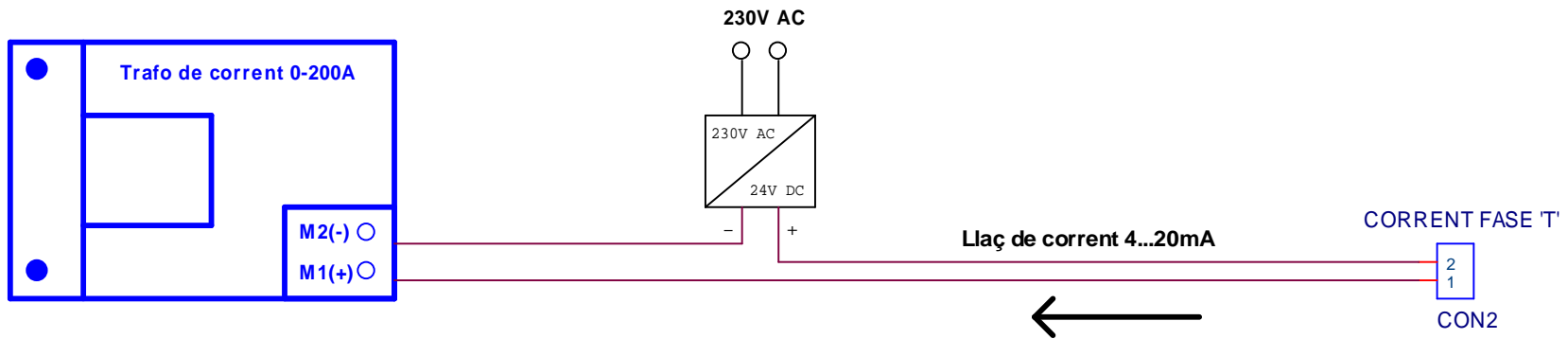
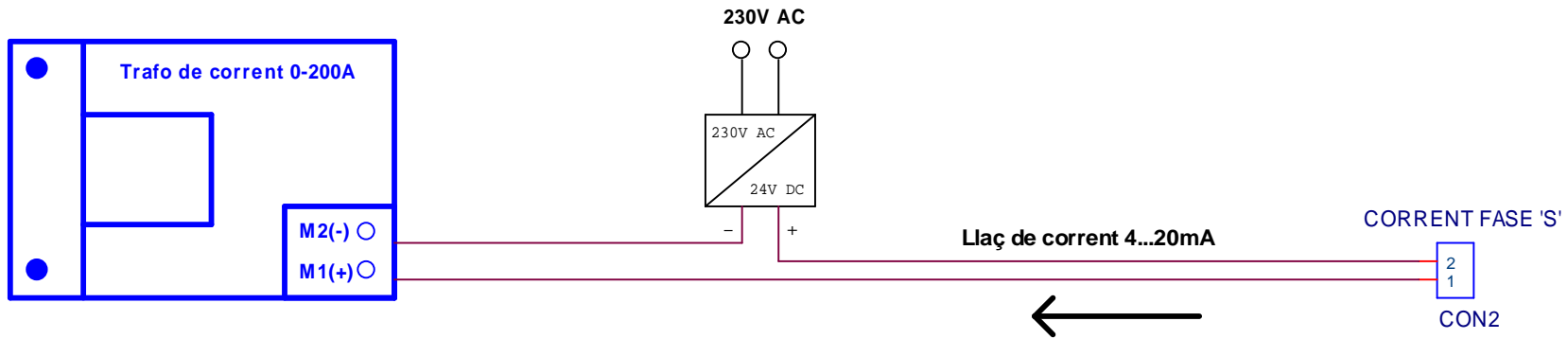
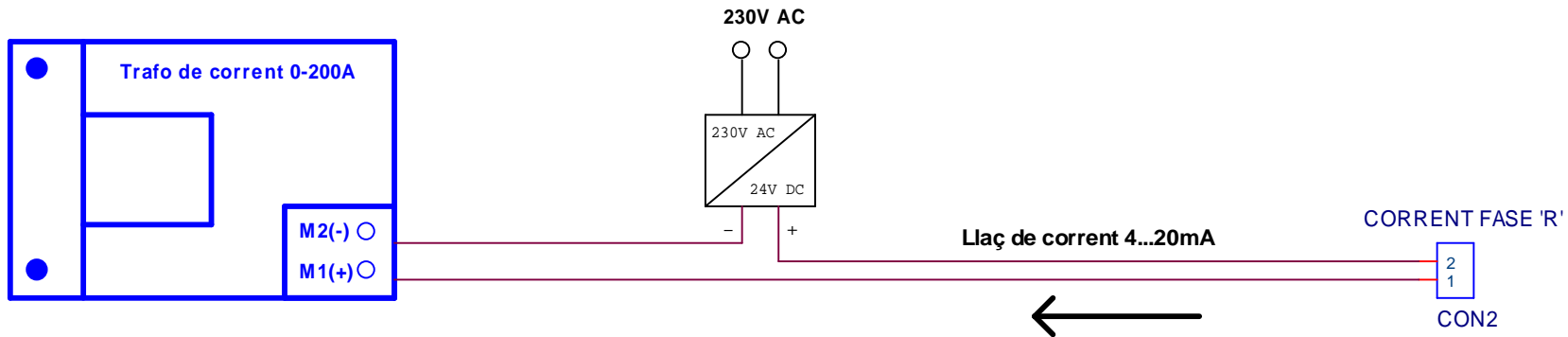
ESCOLA UNIVERSITÀRIA POLITÈCNICA DE MATARÓ		
ANALITZADOR DE LA XARXA ELÈCTRICA TRIFÀSICA DE PROPÒSIT GENERAL		
ALUMNE: JOAN MARTÍNEZ NÚÑEZ		
PONENT: JULIÁN HERRILLO TELLO		
Title		
Esquema: Connexió LCD i teclat		
Size	Document Number	Rev
A4	B-2	1
Date:	Saturday, June 19, 2010	Sheet 2 of 5



ESCOLA UNIVERSITÀRIA POLITÈCNICA DE MATARÓ ANALITZADOR DE LA XARXA ELÈCTRICA TRIFÀSICA DE PROPÒSIT GENERAL ALUMNE: JOAN MARTINEZ NÚÑEZ PONENT: JULIÀN HERRILLO TELLO		
Title		
Esquema: Circuit Analògic Adaptador de Nivells de 4...20mA a 0...5V		
Size	Document Number	Rev
A3	B-3	1
Date:	Saturday, June 19, 2010	Sheet 3 of 5



ESCOLA UNIVERSITÀRIA POLITÈCNICA DE MATARÓ ANALITZADOR DE LA XARXA ELÈCTRICA TRIFÀSICA DE PROPÒSIT GENERAL ALUMNE: JOAN MARTÍNEZ NÚÑEZ PONENT: JULIÁN HERRILLO TELLO		
Title		
Esquema: Circuit Convertidor Corrent-Tensió		
Size	Document Number	Rev
A4	B-4	1
Date:	Saturday, June 19, 2010	Sheet 4 of 5



ESCOLA UNIVERSITÀRIA POLITÈCNICA DE MATARÓ ANALITZADOR DE LA XARXA ELÈCTRICA TRIFÀSICA DE PROPÒSIT GENERAL ALUMNE: JOAN MARTÍNEZ NÚÑEZ PONENT: JULIÁN HERRILLO TELLO		
Title		
Esquema: Transformadors de corrent. Mesurador de corrent trifàsic		
Size	Document Number	Rev
A4	B-5	1
Date:	Saturday, June 19, 2010	Sheet 5 of 5



Escola Universitària
Politécnica de Mataró

Enginyeria Tècnica Industrial: Especialitat Electrònica Industrial

**ANALITZADOR DE LA XARXA ELÈCTRICA
TRIFÀSICA DE PROPÒSIT GENERAL**

C – Pressupost

**JOAN MARTÍNEZ NÚÑEZ
JULIÁN HERRILLO TELLO**

PRIMAVERA 2010

ÍNDIX

1. PRESSUPOST.....	1
1.1. Costos directes.....	1
1.1.1. Costos d'enginyeria.....	1
1.1.2. Cost del material.....	1
1.1.3. Amortització instrumental.....	4
1.1.4. Càlcul dels costos directes.....	4
1.2. Costos indirectes.....	5
1.3. Costos totals de desenvolupament del projecte.....	5
2. ESTUDI ECONÒMIC.....	7
2.1. Estudi econòmic per una previsió de vendes de 1.000 unitats.....	8
2.2. Estudi econòmic per una previsió de vendes de 100 unitats.....	8

1. PRESSUPOST

1.1. Costos directes

1.1.1. Costos d'enginyeria

Concepte	Hores	Preu/hora	Total
Estudi de mercat	10	60 €	600 €
Estudi i documentació	45	60 €	2.700 €
Disseny	50	60 €	3.000 €
Muntatge i proves de funcionament	180	30 €	5.400 €
Redacció memòria	58	30 €	1.740 €
Total			13.440 €

1.1.2. Cost del material

Descripció	Quantitat	Preu	Total
Circuits integrats			
Micro PIC16F877A	1	5,87 €	5,87 €
Regulador LM7805	1	0,75 €	0,75 €
Key-encoder MM74C922N	1	8,50 €	8,50 €
RTC DS1307	1	3,91 €	3,91 €
ST232CN	1	0,50 €	0,50 €
OP AMP LF347	4	0,19 €	0,76 €
Transistors			
BC547B	1	0,02 €	0,02 €
Relés			
8L01-05-001	1	1,53 €	1,53 €
Díodes i LEDs			
Díode 1N4007	1	0,02 €	0,02 €
Díode 1N4148	1	0,02 €	0,02 €
Díode LED verd	1	0,16 €	0,16 €
Díode LED vermell	1	0,16 €	0,16 €
Díode Zener BZX55-C5V1	6	0,12 €	0,70 €

Descripció	Quantitat	Preu	Total
Cristalls de quars			
Cristall 20MHz	1	1,20 €	1,20 €
Cristall 32.768kHz	1	0,62 €	0,62 €
Resistències			
Resistència 330Ω 1/4W	2	0,04 €	0,08 €
Resistència 10KΩ 1/4W	1	0,04 €	0,04 €
Resist. potenciòmetre 10KΩ	2	2,00 €	4,00 €
Resistència 470Ω 1/4W	1	0,04 €	0,04 €
Resistència 100Ω 1/4W	1	0,04 €	0,04 €
Resistència 4K7Ω 1/4W	2	0,04 €	0,08 €
Resistència 18KΩ 1/4W	1	0,04 €	0,04 €
Resistència 1K5Ω 1/4W	1	0,04 €	0,04 €
Resistència 1KΩ 1/4W	1	0,04 €	0,04 €
Resistència 2K7Ω 1/4W	1	0,04 €	0,04 €
Resistència 1K8Ω 1/4W	1	0,04 €	0,04 €
Resistència 80KΩ 1/4W ±1%	6	0,05 €	0,30 €
Resistència 100KΩ 1/4W ±1%	6	0,05 €	0,30 €
Resistència 250Ω 1/4W ±1%	6	0,05 €	0,30 €
Resistència 10Ω 5W	1	2,29 €	2,29 €
Condensadors			
Condensador 0.1μF	15	0,02 €	0,30 €
Condensador 22pF	2	0,02 €	0,04 €
Condensador 220μF	1	0,46 €	0,46 €
Condensador 100μF	1	0,12 €	0,12 €
Condensador 1μF	1	0,10 €	0,10 €
Connectors			
Connector DB9 femella	1	1,22 €	1,22 €
Connector RJ11 6p	1	0,56 €	0,56 €
Con. colzat 2p femella	7	0,25 €	1,75 €
Con. colzat 3p femella	1	0,50 €	0,50 €
Con. colzat 10p femella	3	1,54 €	4,62 €
Con. colzat 2p mascle	7	0,20 €	1,40 €
Con. colzat 3p mascle	1	0,30 €	0,30 €
Con. colzat 10p mascle	3	0,81 €	2,44 €
Con. cinta plana 14p femella	2	0,48 €	0,95 €
Con. cinta plana 10p femella	2	0,40 €	0,80 €
Con. cinta plana 14p mascle	1	0,74 €	0,74 €
Con. cinta plana 14p mascle	1	0,49 €	0,49 €

Descripció	Quantitat	Preu	Total
PCBs			
Placa WR fibra vidre 100x160	1	7,56 €	7,56 €
Placa WR fibra vidre 78x90	2	2,18 €	4,35 €
Polsadors			
Polsador miniatura CI	1	0,09 €	0,09 €
Sòcols			
Sòcol WR DIP 8	1	0,33 €	0,33 €
Sòcol WR DIP 14	5	0,58 €	2,92 €
Sòcol WR DIP 16	1	1,05 €	1,05 €
Sòcol WR DIP 18	1	1,23 €	1,23 €
Sòcol WR DIP 40	1	1,58 €	1,58 €
F.A.			
Font alimentació 24V 175mA	3	19,70 €	59,10 €
Font alimentació ±15V 1A	1	38,50 €	38,50 €
Trafos de corrent			
TC 200A / 4...20mA	3	156,23 €	468,69 €
Varis			
Cable WR	10	0,32 €	3,20 €
Connexions CI WR	45	0,07 €	3,15 €
Teclat 4x4	1	13,15 €	13,15 €
Display LCD 2x16	1	26,90 €	26,90 €
Ferreteria	20	0,67 €	13,40 €
Cable pla 10p	1	0,4445	0,44 €
Cable pla 14p	1	0,94 €	0,94 €
Cable 0,75 mm ² Ø	10	0,30 €	3,00 €

Cost del material	698,73 €
--------------------------	-----------------

1.1.3. Amortització instrumental

Instrumentació informàtica:

Equip utilitzat	Hores	Preu/hora	Total
Ordinador	250	0,50 €	125 €
Software MPLAB IDE v8.46	160	0,00 €	0 €
Software ORCAD	25	2,00 €	50 €
Microsoft Office	58	1,00 €	58 €
Total			233 €

Instrumentació electrònica:

Equip utilitzat	Hores	Preu/hora	Total
Multímetre	5	1,00 €	5 €
Generador corrent	1	2,00 €	2 €
Font d'alimentació	5	0,50 €	3 €
Equip de soldadura	10	10,00 €	100 €
Eines de mà	20	0,30 €	6 €
Total			116 €

1.1.4. Càlcul dels costos directes

Pertanyen a aquest grup tots els costos d'enginyeria, materials, eines i altres elements que intervenen en la fabricació directa del prototip:

Costos d'enginyeria	15.540 €
Costos de material	699 €
Costos d'amortització	349 €
Total	16.587 €

1.2. Costos indirectes

Per valorar amb exactitud els costos que s'imputen en un projecte s'han de comptar els costos indirectes. En una empresa real es podrien calcular amb més o menys exactitud i normalment els costos indirectes acostumen a representar entre el 10% i 25% dels costos directes. Però en un projecte com aquest és molt difícil de valorar, per tant s'assigna un 10% dels costos directes per als costos indirectes:

Costos indirectes	1.658 €
--------------------------	----------------

1.3. Costos totals de desenvolupament del projecte

Costos directes	16.587 €
Costos indirectes	1.658 €
Total	18.245 €

2. ESTUDI ECONÒMIC

L'estudi econòmic pretén calcular un preu de venda en el mercat de l'analyzador de la xarxa elèctrica trifàsica. Per el càlcul del preu de venda es valoren dos escenaris: un escenari positiu amb una previsió de vendes de 1.000 unitats i un escenari negatiu o realista, amb una previsió de vendes de 100 unitats.

Abans de realitzar l'estudi d'ambdós escenaris però, cal fer algunes consideracions:

- Els transformadors de corrent són els elements més costosos del projecte. El fet de que els transformadors puguin mesurar fins a 200A és específic d'aquest projecte. Normalment amb un rang de fins a 50A es cobreix un ampli ventall d'aplicacions. És amb aquests transformadors que es farà l'estudi econòmic.

Descripció	Quantitat	Preu	Total
Trafos de corrent			
TC 50A / 4...20mA	3	119,71 €	359,13 €

- Els costos de fabricació són els corresponents al muntatge en sèrie de l'analyzador de la xarxa elèctrica trifàsica. Si es contractés una empresa especialitzada en el muntatge de circuits electrònics es podrien aconseguir preus de 18€ per la fabricació de 100 unitats i 10€ per la fabricació de 1.000 unitats, segons diverses fonts consultades. El preu inclou la fabricació de les plaques de circuit imprès i el muntatge dels components.

Costos de fabricació 100 unitats	18 €
Costos de fabricació 1.000 unitats	10 €

- El visualitzador LCD del prototip és massa car i té un elevat consum de corrent. Per tant s'utilitzaria una versió més econòmica. El fabricant de *kits* d'electrònica CEBEK ven un display LCD amb les mateixes característiques funcionals i més econòmic, per uns 8,50€, el model CEBEK C-2602. De fet aquest display LCD

manté una compatibilitat total amb l'utilitza't en aquest projecte i es podria connectar directament sense haver de fer cap modificació al circuit.

- Per l'estudi econòmic del primer escenari (amb una previsió de vendes de 1.000 unitats) es consideren uns descomptes de material de fins al 40% per el fet de comprar en grans quantitats. Els descomptes de material per al segon escenari (amb una previsió de vendes de 100 unitats) seria del 25%, tot i què en segons quins components el descompte podria ser encara major.

2.1. Estudi econòmic per una previsió de vendes de 1.000 unitats

Concepte	Valor unitari	Total
Costos del material amb descompte del 40%	342 €	342.462 €
Costos fabricació 1000 unitats	10 €	10.000 €
Cost del prototip	---	18.245 €

Cost per unitat	371 €
------------------------	--------------

Preu amb un marge comercial del 30%	482 €
--	--------------

2.2. Estudi econòmic per una previsió de vendes de 100 unitats

Concepte	Valor unitari	Total
Costos del material amb descompte del 25%	428 €	42.808 €
Costos fabricació 100 unitats	18 €	1.800 €
Cost del prototip	---	18.245 €

Cost per unitat	629 €
------------------------	--------------

Preu amb un marge comercial del 30%	817 €
--	--------------



Escola Universitària
Politécnica de Mataró

Enginyeria Tècnica Industrial: Especialitat Electrònica Industrial

**ANALITZADOR DE LA XARXA ELÈCTRICA
TRIFÀSICA DE PROPÒSIT GENERAL**

D – Annexes

**JOAN MARTÍNEZ NÚÑEZ
JULIÁN HORRILLO TELLO**

PRIMAVERA 2010

ANNEX I
CONTINGUT DEL CD-ROM

ANNEX I: CONTINGUT DEL CD-ROM

Amb el present projecte s'han inclòs dos CD-ROM. El contingut d'ambdós CD-ROM és idèntic i es mostra a continuació:

- Documentació del projecte
- Article resum de la memòria
- *Data sheets* dels components i del material utilitzat en el projecte
- Programari i manuals de l'editor MPLAB[®] IDE i del compilador CCS C
- Notes d'aplicació de *Microchip*
- Codi font en C i executables

ANNEX II
CODI DEL PROGRAMA

ANNEX II: CODI DEL PROGRAMA

En aquest segon annex s'inclou el codi del programa, des del codi del programa principal fins les rutines d'atenció a les interrupcions i els *drivers* del visualitzador LCD i el rellotge de temps real.

Codi del programa principal:

```
// PROGRAMA ANALITZADOR DE LA XARXA ELÈCTRICA TRIFÀSICA

/*----- DESCRIPCIÓ DEL PROGRAMA -----

Programa:      Aquest programa forma part del projecte de final de carrera anomenat
                "Anàlitzador de la Xarxa Elèctrica Trifàsica", un sistema electrònic
                capaç de mesurar en temps real una magnitud de la xarxa elèctrica
                trifàsica, en aquest cas el corrent.

Assignatura:   Projecte de Final de Carrera

Ponent:        Julián Horrillo Tello

Programador:   Joan Martínez Núñez

Data:          19 de maig de 2010

Descripció:    Es visualitza en un LCD en temps real el consum del sistema trifàsic
                i s'envien mostres al PC. A través d'interrupcions es prenen mostres
                del ADC consecutivament de cada una de les tres entrades
                analògiques. Des del teclat es pot accedir als diferents menús, que
                ens permetran programar la data i l'hora del RTC, programar el
                llindar d'alarma, veure els consums màxims i el promig, la data i
                l'hora en temps real i reaccionar quan se supera el llindar
                d'alarma.

Fitxers:       programa analitzador xarxa electrica.c
                16F877A.h
                lcd.c
                ds1307.c

*/

//----- DIRECTIVES DEL PROCESSADOR -----

#include <16F877A.h>      // Arxiu de capçalera (Header) del micro PIC16F877A
#define * =16           // Permet la paginació dels 4 bancs de la RAM
#define adc=10         // La dada que s'obté del ADC és de 10 bits

//----- PROGRAMACIÓ DE LA PARAULA DE CONFIGURACIÓ -----

#define NOWDT, HS, PUT, NOPROTECT, NODEBUG, NOBROWNOUT, NOLVP, NOCPD, NOWRT

// NOWDT: No hi ha WatchDog
// HS: Rellotge d'alta velocitat
// PUT: Temporització de 72ms del RESET quan se li dona tensió
// NOPROTECT: No hi ha protecció de la EEPROM
// NODEBUG: No es permet la depuració en el circuit
// NOBROWNOUT: No hi ha BrownOut
// NOLVP: No hi ha baixa tensió de programació
```

2 Annex II

```
// NOCPD: No hi ha protecció del codi
// NOWRT: No hi ha protecció d'escriptura a la Flash EEPROM

//---- CONFIGURACIÓ DEL RELLOTGE, DE LA USART I DEL BUS I2C -----
#use delay(clock=2000000) // Freq. CPU 20MHz
#use rs232(baud=19200,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8) // Config. USART
#use i2c(master,fast,sda=PIN_C4,scl=PIN_C3,force_hw) // Config. bus I2C

//---- DIRECTIVES DE CONFIGURACIÓ DEL LCD -----
#define LCD_DATA_PORT D // S'utilitza el port D per el LCD
#define LCD_TYPE 2 // LCD de dues línies
#define LCD_TRIS_LOCATION 0x0F // Direcció registre configuració port D
#include <lcd.c> // Driver amb les funcions per programar el LCD

//---- DIRECTIVES DE CONFIGURACIÓ DEL RTC -----
#include <ds1307.c> // Driver amb les funcions per programar el RTC

//---- CONFIGURACIÓ DELS PORTS I/O -----
#use standard_io(E) // Directiva per utilitzar les funcions del port E
#byte TRISB = 0x86 // Adreçament del registre de configuració del PORT B
#byte PORTB = 0x06 // Adreçament del registre de control del PORT B

//---- DECLARACIÓ I INICIALITZACIÓ DE LES VARIABLES GLOBALES -----
short menu_data_i_hora_actiu = FALSE; // Flag menú actiu: data i hora
short menu_llindar_actiu = FALSE; // Flag menú actiu: llindar
short alarma_exces_llindar = FALSE; // Flag menú actiu: alarma
short reconeixement_alarma_actiu = FALSE; // Flag menú actiu: reconeixement alarma
short numero_actiu = FALSE; // Flag número actiu o no
int menu_principal_actiu = FALSE; // Variable menú i submenús actius
int key = 0; // Variable que conté el codi de la tecla polsada
int i = 0; // Índex pel vector d'emmagatzemar les mostres R
int j = 0; // Índex pel vector d'emmagatzemar les mostres S
int k = 0; // Índex pel vector d'emmagatzemar les mostres T
int hr = 0; // Variable intermitja per programar l'hora
int min = 0; // Variable intermitja per programar els minuts
int sec = 0; // Variable intermitja per programar els segons
int hr_alarm = 0; // Variable que conté l'hora de l'alarma
int min_alarm = 0; // Variable que conté els minuts de l'alarma
int sec_alarm = 0; // Variable que conté els segons de l'alarma
int hora = 0; // Variable que conté l'hora
int minuts = 0; // Variable que conté els minuts
int segons = 0; // Variable que conté els segons
int dia_setmana = 1; // Variable que conté el dia de la setmana
int data = 1; // Variable que conté la data
int mes = 1; // Variable que conté el mes
int any = 0; // Variable que conté l'any
int segons_anterior = 0; // Per detectar el canvi d'un segon
int minuts_anterior = 0; // Per detectar el canvi d'un minut
int rotacio_canal_ADC = 1; // Rotació dels canals del ADC
int promig_consum_R = 0; // Per emmagatzemar el consum promig de la fase R
int promig_consum_S = 0; // Per emmagatzemar el consum promig de la fase S
int promig_consum_T = 0; // Per emmagatzemar el consum promig de la fase T
int divisor_promig = 0; // S'utilitza com a divisor per calcular el promig
int consum_fase_R = 0; // Conté el consum instantani real de la fase R
int consum_fase_S = 0; // Conté el consum instantani de la fase S
int consum_fase_T = 0; // Conté el consum instantani de la fase T
int maxim_consum_R = 0; // Conté el consum màxim de la fase R
```



```

int maxim_consum_S = 0; // Conté el consum màxim de la fase S
int maxim_consum_T = 0; // Conté el consum màxim de la fase T
int taula_mostres_R[45]; // Vector amb 45 mostres consecutives a l'alarma fase R
int taula_mostres_S[45]; // Vector amb 45 mostres consecutives a l'alarma fase S
int taula_mostres_T[45]; // Vector amb 45 mostres consecutives a l'alarma fase T
long valor_ADC = 0; // Conté el valor del ADC en decimal (0 - 1024)
long tecla_polsada = 0; // Variable que conté el número de la tecla polsada
long llindar_alarma = 100; // Valor llindar d'alarma
long llindar_alarma_prog = 100; // Variable intermitja per programar el llindar
long acumulat_consum_R = 0; // Variable per emmagatzemar l'acumula't fase R
long acumulat_consum_S = 0; // Variable per emmagatzemar l'acumula't fase R
long acumulat_consum_T = 0; // Variable per emmagatzemar l'acumula't fase R
long numero_de_mostres = 0; // Número de mostres que ha realitzat el ADC
long refresc_lcd = 0; // Temps de refresc del LCD: 300 unitats aprox.= 200ms

```

//---- DECLARACIÓ DE FUNCIONS -----

```

#SEPARATE void actualitza_promig(void);
#SEPARATE void calcul_promig(void);
void AD_isr(void); // RSI interrupció teclat
void EXT_isr(void); // RSI interrupció EOC del ADC
short verifica_entrada_teclat(int minim,int maxim,long numero);
void validar_numero(int key);
void display_missatge(void);

```

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

```

void main()
{
    disable_interrupts(GLOBAL); // Es desconnecten les interrupcions

    TRISB = 0xFF; // 8 bits del PORT B com a entrades
    output_low(PIN_E0); // Apaga el LED d'alarma
    output_low(PIN_E1); // Desactiva el relé d'alarma

    lcd_init(); // LCD en format de bus de dades de 4 bits,
    delay_ms(100); // dues línies, caràcters de 5x8 punts,
    // mode encès, cursor apagat i sense pampallugues,
    // amb auto-increment del punter de direcció
    // i sense desplaçament del display real

    lcd_gotoxy(1,1); // Es posiciona el cursor del LCD
    printf(lcd_putc, "\f Analitzador"); // Missatge inicial
    printf(lcd_putc, "\n de la xarxa"); // de l'aplicació
    delay_ms(4000); // Retard de 4 segons
    lcd_gotoxy(1,1); // Es posiciona el cursor del LCD
    printf(lcd_putc, "\f electrica"); // Segona part del missatge inicial
    printf(lcd_putc, "\n trifasica");
    delay_ms(4000); // Retard de 4 segons
    lcd_putc('\f'); // S'esborra el LCD

    // Es programa el rellotge i el calendari del RTC per defecte
    ds1307_init();
    delay_ms(100); // Retard de 100ms
    ds1307_set_date_time(data,mes,any,dia_setmana,hora,minuts,segons);

    setup_adc_ports(AN0_AN1_AN3); // RA0, RA1 i RA3 son entrades analògiques
    setup_adc(ADC_CLOCK_DIV_32); // Es configura el rellotge del ADC
    set_adc_channel(0); // Es selecciona el canal del ADC

    setup_ccp2(CCP_COMPARE_RESET_TIMER); // Config. mòdul CCP2 i es carrega
    CCP_2_LOW = 0x05; // per un Tm del ADC de 2 ms (freq. de 500 Hz)
    CCP_2_HIGH = 0x0D; // quan la freqüència de rellotge és de 20 MHz

```

4 Annex II

```
setup_timer_1(T1_INTERNAL | T1_DIV_BY_1);    // Es configura el TIMER1
set_timer1(0);                               // Es posa a 0 el TIMER1
ext_int_edge(L_TO_H);                        // Habilitació de la INT del teclat RB0...
enable_interrupts(INT_EXT);                 // per flanc de pujada
enable_interrupts(INT_AD);                  // Habilitació de la INT EOC del ADC
enable_interrupts(GLOBAL);                 // Habilitació general de les INT

do
{
    // Transmissió al PC del consum de cada fase
    printf("%3u %3u %3u\r", consum_fase_R, consum_fase_S, consum_fase_T);

    // Visualització per LCD dels consums cada 200ms
    if ((refresc_lcd > 300) && (menu_principal_actiu == FALSE) &&
        (menu_data_i_hora_actiu == FALSE) && (menu_llindar_actiu == FALSE))
    {
        // Posiciona el cursor a la primera posició del LCD
        lcd_gotoxy(1,1);

        // Missatge per a la primera línia del LCD
        printf	lcd_putc, " R=%3uA S=%3uA ", consum_fase_R, consum_fase_S);

        // Missatge per a la segona línia del LCD
        printf	lcd_putc, "\n T=%3uA CONSUM ", consum_fase_T);

        // Actualització de la variable refresc
        refresc_lcd = 0;
    }

    // Obtenció de l'hora del RTC, el DS1307
    ds1307_get_time(hora, minuts, segons);

    // Si ha transcorregut 1 segon...
    if (segons != segons_anterior)
    {
        // Transmissió al PC del número de mostres per segon
        printf("mostres = %5.0w\r", numero_de_mostres);

        // Reset de la variable amb el número de mostres
        numero_de_mostres = 0;

        // Actualització de la variable segons_anterior
        segons_anterior = segons;
    }

    // Si ha transcorregut 1 minut, aleshores
    // s'actualitzen els acumulats i es calcula el promig
    if (minuts != minuts_anterior)
    {
        // Cada 5 minuts s'actualitza l'acumulat
        if (minuts % 5 == 0)
        {
            // Crida a la funció per actualitzar el promig
            actualitza_promig();
        }

        // Càlcul del promig si és mitjanit
        if ((hora == 23) && (minuts == 59))
        {
            // Crida a la funció per calcular el promig
            calcul_promig();
        }

        // Actualització de la variable minuts_anterior
        minuts_anterior = minuts;
    }
}
```

```

// Actualització del valor màxim fase R
if (consum_fase_R > maxim_consum_R)
{
    maxim_consum_R = consum_fase_R;
}

// Actualització del valor màxim fase S
if (consum_fase_S > maxim_consum_S)
{
    maxim_consum_S = consum_fase_S;
}

// Actualització del valor màxim fase T
if (consum_fase_T > maxim_consum_T)
{
    maxim_consum_T = consum_fase_T;
}

// Display data i hora si el flag és actiu
if (menu_data_i_hora_actiu == TRUE)
{
    // Obtenció de la data, perquè l'hora ja s'ha actualitzat abans
    ds1307_get_date(data,mes,any,dia_setmana);

    // Posiciona el cursor a la primera posició del LCD
    lcd_gotoxy(1,1);

    // Visualitza la data per la primera línia del LCD
    printf(lcd_putc,"data %02u/%02u/20%02u",data,mes,any);

    // Visualitza l'hora per la segona línia del LCD
    printf(lcd_putc,"\nhora %02u:%02u:%02u",hora,minuts,segons);
}

// Reconeixement de l'alarma amb la tecla 'D' si el flag d'alarma és actiu
if (reconeixement_alarma_actiu == TRUE)
{
    // Desconnecta les interrupcions
    disable_interrupts(GLOBAL);

    // Missatge per LCD amb l'acció que es realitzarà
    printf(lcd_putc,"\fEnviant mostres");
    printf(lcd_putc,"\ni dades al PC...");

    // Envia missatge al PC pel port RS232 amb l'acció que es realitzarà
    printf("\r Enviant l'hora en què s'ha produït l'alarma...\r\r");

    // Envia l'hora de l'alarma al PC pel port RS232
    printf(" %02u:%02u:%02u\r\r",hr_alarma,min_alarma,sec_alarma);

    // Envia missatge al PC pel port RS232 amb l'acció que es realitzarà
    printf("\r Enviant les mostres de l'alarma al PC:\r\r");

    // Sentència FOR que recorre els vectors amb les mostres
    // i les envia cap al PC a través del port RS232
    for (i=0;i<45;i++)
    {
        printf("%3u %3u %3u\r", taula_mostres_R[i], taula_mostres_S[i],
            taula_mostres_T[i]);
    }

    // Reset dels vectors, preparats per una altre alarma
    i = 0; j = 0; k = 0;

    // Missatge al PC fi de la transmissió
    printf("\r ...fi de la transmissió.\r\r");
}

```

```

// Apaga el LED d'alarma
output_low(PIN_E0);

// Desactiva el relé d'alarma
output_low(PIN_E1);

// Deshabilita tots els menús actius
menu_principal_actiu = FALSE;
menu_llindar_actiu = FALSE;

// Desactiva els flags que s'activen
// quan es produeix una alarma
reconeixement_alarma_actiu = FALSE;
alarma_exces_llindar = FALSE;

// Retard de 150 ms
delay_ms(150);

// Reconnecta les interrupcions
enable_interrupts(GLOBAL);

} // end if

} // end do-while

while (TRUE); // Bucle infinit

} // end main

//----- FUNCIO PER ACTUALITZAR EL PROMIG -----
#SEPARATE void actualitza_promig(void)
{
    acumulat_consum_R += consum_fase_R;
    acumulat_consum_S += consum_fase_S;
    acumulat_consum_T += consum_fase_T;
    divisor_promig++;
} // end funció actualitza_promig()

//----- FUNCIO PER CALCULAR EL PROMIG -----
#SEPARATE void calcul_promig(void)
{
    promig_consum_R = acumulat_consum_R / divisor_promig;
    promig_consum_S = acumulat_consum_S / divisor_promig;
    promig_consum_T = acumulat_consum_T / divisor_promig;
    acumulat_consum_R = 0;
    acumulat_consum_S = 0;
    acumulat_consum_T = 0;
    divisor_promig = 0;
} // end funció calcul_promig()

```

Codi de la rutina d'atenció a la interrupció del teclat:

```
//----- RUTINA SERVEI INTERRUPCIÓ DEL TECLAT (INT-RB0) -----
#int_EXT void EXT_isr(void)
{
    key = PORTB & 0x1E; // "key" conté el valor dels bits RB4-RB3-RB2-RB1

    switch (key)
    {
        case 0x06: // Tecla 'A' menú principal
        {
            if ((menu_principal_actiu==FALSE)&&(menu_data_i_hora_actiu==FALSE)&&
                (menu_llindar_actiu==FALSE)&&(alarma_exces_llindar==FALSE))
            {
                printf(lcd_putc, "\fmenu: <1> <2>");
                printf(lcd_putc, "\n<3> <4> <5> <6>");
                menu_principal_actiu = 10;
            }
        }
        break;

        case 0x0E: // Tecla 'B' visualització data i hora
        {
            if ((menu_principal_actiu==FALSE)&&(menu_data_i_hora_actiu==FALSE)&&
                (menu_llindar_actiu==FALSE)&&(alarma_exces_llindar==FALSE))
            {
                menu_data_i_hora_actiu = TRUE;
            }
        }
        break;

        case 0x16: // Tecla 'C' visualització llindar d'alarma
        {
            if ((menu_principal_actiu==FALSE)&&(menu_data_i_hora_actiu==FALSE)&&
                (menu_llindar_actiu==FALSE)&&(alarma_exces_llindar==FALSE))
            {
                printf(lcd_putc, "\fLlindar alarma:");
                printf(lcd_putc, "\n >%3.0wA", llindar_alarma);
                menu_llindar_actiu = TRUE;
            }
        }
        break;

        case 0x18: // Tecla '*' (ESCAPE)
        {
            menu_principal_actiu = FALSE;
            menu_data_i_hora_actiu = FALSE;
            menu_llindar_actiu = FALSE;
        }
        break;

        case 0x1C: // Tecla '#' (ENTER)
        {
            switch (menu_principal_actiu)
            {
                case 11:
                {
                    printf(lcd_putc, "\fdata (1-31):_");
                    printf(lcd_putc, "\n(dos digits)");
                    lcd_gotoxy(13,1);
                    menu_principal_actiu = 111;
                }
                break;
            }
        }
    }
}
```

```

case 12:
{
    printf(lcd_putc, "\fhora (0-24):_");
    printf(lcd_putc, "\n(dos digits)");
    lcd_gotoxy(13,1);
    menu_principal_actiu = 121;
}
break;

case 13:
{
    printf(lcd_putc, "\fValor maxim:_");
    printf(lcd_putc, "\n(tres digits)");
    lcd_gotoxy(13,1);
    menu_principal_actiu = 131;
}
break;

case 14:
{
    printf(lcd_putc, "\f R=%3uA S=%3uA ",
    promig_consum_R, promig_consum_S);
    printf(lcd_putc, "\n T=%3uA PROMIG ",
    promig_consum_T);
}
break;

case 15:
{
    printf(lcd_putc, "\f R=%3uAS=%3uA", maxim_consum_R,
    maxim_consum_S);
    printf(lcd_putc, "\n T=%3uA MAXIMS", maxim_consum_T);
}
break;

case 20:
{
    maxim_consum_R = 0; // Menú de reset dels màxims
    maxim_consum_S = 0;
    maxim_consum_T = 0;
    menu_principal_actiu = FALSE;
}
break;

case 111:
{
    lcd_putc('_');
    lcd_putc('\b');
    data = tecla_polsada * 10;
    menu_principal_actiu = 112;
}
break;

case 112:
{
    data = data + tecla_polsada;

    if (verifica_entrada_teclat(1,31,data))
    {
        printf(lcd_putc, "\fmes (1-12):_");
        printf(lcd_putc, "\n(dos digits)");
        lcd_gotoxy(12,1);
        menu_principal_actiu = 113;
    }
}

```

```

        else
        {
            display_missatge();
        }
    }
break;

case 113:
{
    lcd_putc('_');
    lcd_putc('\b');
    mes = tecla_polsada * 10;
    menu_principal_actiu = 114;
}
break;

case 114:
{
    mes = mes + tecla_polsada;

    if (verifica_entrada_teclat(1,12,mes))
    {
        printf(lcd_putc, "\fany (00-99):_");
        printf(lcd_putc, "\n(dos digits)");
        lcd_gotoxy(13,1);
        menu_principal_actiu = 115;
    }
    else
    {
        display_missatge();
    }
}
break;

case 115:
{
    lcd_putc('_');
    lcd_putc('\b');
    any = tecla_polsada * 10;
    menu_principal_actiu = 116;
}
break;

case 116:
{
    any = any + tecla_polsada;

    if (verifica_entrada_teclat(0,99,any))
    {
        printf(lcd_putc, "\fdia de la");
        printf(lcd_putc, "\nsetmana (1-7):_\b");
        menu_principal_actiu = 117;
    }
    else
    {
        display_missatge();
    }
}
break;

case 117:
{
    dia_setmana = tecla_polsada;

```

```

    if (verifica_entrada_teclat(1,7,dia_setmana))
    {
        ds1307_set_date_time(data,mes,any,
        dia_setmana,hora,minuts,segons);
        menu_principal_actiu = FALSE;
    }
    else
    {
        display_missatge();
    }
}
break;

case 121:
{
    lcd_putc('_');
    lcd_putc('\b');
    hr = tecla_polsada * 10;
    menu_principal_actiu = 122;
}
break;

case 122:
{
    hr = hr + tecla_polsada;

    if (verifica_entrada_teclat(0,23,hr))
    {
        printf(lcd_putc, "\fminut (0-59):_");
        printf(lcd_putc, "\n(dos digits)");
        lcd_gotoxy(14,1);
        menu_principal_actiu = 123;
    }
    else
    {
        display_missatge();
    }
}
break;

case 123:
{
    lcd_putc('_');
    lcd_putc('\b');
    min = tecla_polsada * 10;
    menu_principal_actiu = 124;
}
break;

case 124:
{
    min = min + tecla_polsada;

    if (verifica_entrada_teclat(0,59,min))
    {
        printf(lcd_putc, "\fsegon (0-59):_");
        printf(lcd_putc, "\n(dos digits)");
        lcd_gotoxy(14,1);
        menu_principal_actiu = 125;
    }
    else
    {
        display_missatge();
    }
}
break;

```



```

case 125:
{
    lcd_putc('_');
    lcd_putc('\b');
    sec = tecla_polsada * 10;
    menu_principal_actiu = 126;
}
break;

case 126:
{
    sec = sec + tecla_polsada;

    if (verifica_entrada_teclat(0,59,sec))
    {
        ds1307_set_date_time(data,mes,any,
        dia_setmana,hr,min,sec);
        menu_principal_actiu = FALSE;
    }
    else
    {
        display_missatge();
    }
}
break;

case 131:
{
    llindar_alarma_prog = tecla_polsada * 100;
    lcd_putc('_');
    lcd_putc('\b');
    menu_principal_actiu = 132;
}
break;

case 132:
{
    llindar_alarma_prog=llindar_alarma_prog+(tecla_polsada*10);
    lcd_putc('_');
    lcd_putc('\b');
    menu_principal_actiu = 133;
}
break;

case 133:
{
    llindar_alarma_prog=llindar_alarma_prog+tecla_polsada;

    if (verifica_entrada_teclat(0,200,llindar_alarma_prog))
    {
        llindar_alarma = llindar_alarma_prog;
        menu_principal_actiu = FALSE;
    }
    else
    {
        display_missatge();
    }
}
break;

case 199:menu_principal_actiu = FALSE;

default: break;

} // end switch tecla (#) ENTER

numero_actiu = TRUE;

```

```

    } // end case 0x1C
    break;

    case 0x1E: // Tecla 'D'
    {
        if (alarma_exces_llindar == TRUE)
        {
            reconeixement_alarma_actiu = TRUE;
        }
    }
    break;

    default: break;

} // end switch key

if (menu_principal_actiu)
{
    switch (key)
    {
        case 0x1A: // Menu de recuperació del LCD
        {
            if (menu_principal_actiu == 10)
            {
                lcd_init();
                menu_principal_actiu = FALSE;
            }
            else
            {
                key = 0;
                validar_numero(key);
            }
        }
        break;

        case 0x00: // Menú de programació de la data
        {
            if (menu_principal_actiu == 10)
            {
                printf(lcd_putc, "\fVoleu programar");
                printf(lcd_putc, "\nla data d'avui?");
                menu_principal_actiu = 11;
            }
            else
            {
                key = 1;
                validar_numero(key);
            }
        }
        break;

        case 0x02: // Menú de programació de l'hora
        {
            if (menu_principal_actiu == 10)
            {
                printf(lcd_putc, "\fVoleu programar");
                printf(lcd_putc, "\nel rellotge?");
                menu_principal_actiu = 12;
            }
            else
            {
                key = 2;
                validar_numero(key);
            }
        }
        break;
    }
}

```

```

case 0x04: // Menú de programació del llindar d'alarma
{
    if (menu_principal_actiu == 10)
    {
        printf(lcd_putc, "\fVoleu programar");
        printf(lcd_putc, "\nel llindar?");
        menu_principal_actiu = 13;
    }
    else
    {
        key = 3;
        validar_numero(key);
    }
}
break;

case 0x08: // Visualització del promig
{
    if (menu_principal_actiu == 10)
    {
        printf(lcd_putc, "\fVoleu veure el");
        printf(lcd_putc, "\nconsum promig?");
        menu_principal_actiu = 14;
    }
    else
    {
        key = 4;
        validar_numero(key);
    }
}
break;

case 0x0A: // Visualització dels màxims
{
    if (menu_principal_actiu == 10)
    {
        printf(lcd_putc, "\fVoleu veure el");
        printf(lcd_putc, "\nconsum maxims?");
        menu_principal_actiu = 15;
    }
    else
    {
        key = 5;
        validar_numero(key);
    }
}
break;

case 0x0C: // Reset dels màxims
{
    if (menu_principal_actiu == 10)
    {
        printf(lcd_putc, "\fVoleu fer reset");
        printf(lcd_putc, "\ndels maxims?");
        menu_principal_actiu = 20;
    }
    else
    {
        key = 6;
        validar_numero(key);
    }
}
break;

```

```

        case 0x10:
        {
            key = 7;
            validar_numero(key);
        }
        break;

        case 0x12:
        {
            key = 8;
            validar_numero(key);
        }
        break;

        case 0x14:
        {
            key = 9;
            validar_numero(key);
        }
        break;

        default: break;

    } // end switch
} // end if
} // end RSI teclat

//----- FUNCIO PER VERIFICAR LES DADES INTRODUIDES DES DEL TECLAT -----
short verifica_entrada_teclat(int minim, int maxim, long numero)
{
    short resultat = FALSE;

    if ((numero >= minim) && (numero <= maxim))
    {
        resultat = TRUE;
    }

    return(resultat);
} // end funcio verifica_entrada_teclat()

//----- FUNCIO PER VALIDAR ELS NUMEROS INTRODUITS DES DEL TECLAT -----
void validar_numero(int key)
{
    if
    ((menu_principal_actiu > 100) && (menu_principal_actiu < 150) && (numero_actiu == TRUE))
    {
        tecla_polsada = key;
        lcd_putc(tecla_polsada + 48); // Per convertir un número sencer en un ASCII
        numero_actiu = FALSE;
    }
} // end funcio validar_numero()

//----- FUNCIO PER VISUALITZAR PER EL LCD QUE EL NUMERO NO ÉS CORRECTE -----
void display_missatge(void)
{
    printf(lcd_putc, "\fAquest numero no");
    printf(lcd_putc, "\nes correcte!");
    menu_principal_actiu = 199;
} // end funcio display_missatge()

```

Codi de la rutina d'atenció a la interrupció del convertidor A/D:

```
//----- RUTINA SERVEI INTERRUPTIÓ CONVERSIÓ A/D FINALITZADA -----
#int_AD void AD_isr(void)
{
    // Es llegeix el valor del convertidor A/D del canal actiu.
    // Quan es multiplica per la constant el número es converteix
    // per un fons d'escala de 200A.
    valor_ADC = read_adc(ADC_READ_ONLY) * 0.1953125;

    // Si el valor mesurat supera el llindar d'alarma s'engegen les accions
    // corresponents. Aquí només s'entra una sola vegada.
    if ((valor_ADC > llindar_alarma) && (alarma_exces_llindar == FALSE))
    {
        // S'encén el LED
        output_high(PIN_E0);

        // Activa el relé
        output_high(PIN_E1);

        // Es memoritza l'hora en què s'ha produït l'alarma
        hr_alarm = hora;
        min_alarm = minuts;
        sec_alarm = segons;

        // Activa el flag d'alarma. Ara es troba operatiu el botó D
        alarma_exces_llindar = TRUE;
    }

    // S'analitza quin canal ADC es troba actiu...
    switch (rotacio_canal_ADC)
    {
        // Canal 1 actiu (AIN0)
        case 1:
        {
            // S'actualitza la variable amb el consum de la fase R
            consum_fase_R = valor_ADC;

            // Es prepara el canal ADC següent
            set_adc_channel(1);

            // Si s'ha produït una alarma s'emmagatzemen 45 mostres immediatament
            // posteriors a l'alarma
            if ((alarma_exces_llindar == TRUE) && (i < 45))
            {
                taula_mostres_R[i] = consum_fase_R;
                i++;
            }
        }
        break;

        // Canal 2 actiu (AIN1)
        case 2:
        {
            // S'actualitza la variable amb el consum de la fase S
            consum_fase_S = valor_ADC;

            // Es prepara el canal ADC següent
            set_adc_channel(3);
        }
    }
}
```

```

    // Si s'ha produït una alarma s'emmagatzemen 45 mostres
    if ((alarma_exces_llindar == TRUE) && (j < 45))
    {
        taula_mostres_S[j] = consum_fase_S;
        j++;
    }
}
break;

// Canal 3 actiu (AIN3)
case 3:
{
    // S'actualitza la variable amb el consum de la fase T
    consum_fase_T = valor_ADC;

    // Es prepara el canal ADC següent
    set_adc_channel(0);

    // Si s'ha produït una alarma s'emmagatzemen 45 mostres immediatament
    // posteriors a l'alarma
    if ((alarma_exces_llindar == TRUE) && (k < 45))
    {
        taula_mostres_T[k] = consum_fase_T;
        k++;
    }
}
break;

default: break;

} // end switch

// La variable de rotació dels canals evoluciona
rotacio_canal_ADC++;

if (rotacio_canal_ADC > 3)
{
    // La variable evoluciona entre 1 i 3
    rotacio_canal_ADC = 1;
}

// S'actualitza la variable del nombre de mostres per segon
numero_de_mostres++;

// S'actualitza la variable per el refresc del display
refresc_lcd++;

} // end RSI ADC

```

Codi del driver del visualitzador LCD:

```

////////////////////////////////////
////                               LCD.C                               ////
////          Driver for common LCD modules                          ////
////                                                                 ////
//// lcd_init() Must be called before any other function.          ////
////                                                                 ////
//// lcd_putc(c) Will display c on the next position                ////
////                   of the LCD.                                  ////
////                                                                 ////
//// The following have special meaning:                            ////
////          \f  Clear display                                     ////
////          \n  Go to start of second line                       ////
////          \b  Move back one position                           ////
////                                                                 ////
//// lcd_gotoxy(x,y) Set write position on LCD                      ////
////                   (upper left is 1,1)                         ////
////                                                                 ////
//// lcd_getc(x,y) Returns character at position x,y               ////
////                   on LCD                                       ////
////                                                                 ////
////////////////////////////////////
//  As defined in the following structure the pin
//  connection is as follows:
//
//      D0  enable
//      D1  rs
//      D2  rw
//      D4  D4
//      D5  D5
//      D6  D6
//      D7  D7
//
//      LCD pins D0-D3 are not used and PIC D3 is not used.

struct lcd_pin_map // This structure is overlaid
{
    BOOLEAN enable; // on to an I/O port to gain
    BOOLEAN rs;     // access to the LCD pins.
    BOOLEAN rw;     // The bits are allocated from
    BOOLEAN unused; // low order up.  ENABLE will
    Int data : 4;   // be pin B0.
} lcd;

```

```

#if defined use_portb_lcd
    #ifndef __pch__
        #locate lcd = 0xf81
    #else
        #locate lcd = 6
    #endif
    #define set_tris_lcd(x) set_tris_b(x)

#else
    #ifndef __pch__
        #locate lcd = 0xf83
    #else
        #locate lcd = 8
    #endif
    #define set_tris_lcd(x) set_tris_d(x)
#endif

#ifndef lcd_type
    #define lcd_type 2 // 0 = 5x7, 1 = 5x10, 2 = 2 lines
#endif

#define lcd_line_two 0x40 // LCD RAM address for the second line

// These bytes need to be sent to the LCD to start it up.
BYTE const LCD_INIT_STRING[4] = {0x20|(lcd_type << 2), 0xc, 1, 6};

// The following are used for setting the I/O port register:
struct lcd_pin_map const LCD_WRITE = {0,0,0,0,0};

struct lcd_pin_map const LCD_READ = {0,0,0,0,15};

BYTE lcd_read_byte()
{
    BYTE low,high;

    set_tris_lcd(LCD_READ);
    lcd.rw = 1;
    delay_cycles(1);
    lcd.enable = 1;
    delay_cycles(1);
    high = lcd.data;
    lcd.enable = 0;
    delay_cycles(1);
    lcd.enable = 1;
    delay_us(1);
    low = lcd.data;
    lcd.enable = 0;
    set_tris_lcd(LCD_WRITE);
    return((high<<4) | low);
}

```



```

void lcd_send_nibble(BYTE n)
{
    lcd.data = n;
    delay_cycles(1);
    lcd.enable = 1;
    delay_us(2);
    lcd.enable = 0;
}

void lcd_send_byte(BYTE address, BYTE n)
{
    lcd.rs = 0;
    while ( bit_test(lcd_read_byte(),7) ) ;
    lcd.rs = address;
    delay_cycles(1);
    lcd.rw = 0;
    delay_cycles(1);
    lcd.enable = 0;
    lcd_send_nibble(n >> 4);
    lcd_send_nibble(n & 0xf);
}

void lcd_init()
{
    BYTE i;

    set_tris_lcd(LCD_WRITE);
    lcd.rs = 0;
    lcd.rw = 0;
    lcd.enable = 0;
    delay_ms(15);
    for(i=1;i<=3;++i)
    {
        lcd_send_nibble(3);
        delay_ms(5);
    }

    lcd_send_nibble(2);
    for(i=0;i<=3;++i)
        lcd_send_byte(0,LCD_INIT_STRING[i]);
}

```

```
void lcd_gotoxy(BYTE x, BYTE y)
{
    BYTE address;

    if(y != 1)
        address = lcd_line_two;
    else
        address = 0;

    address += x-1;
    lcd_send_byte(0,0x80 | address);
}

void lcd_putc(char c)
{
    switch (c)
    {
        case '\f':  lcd_send_byte(0,1);
                    delay_ms(2);
                    break;

        case '\n':  lcd_gotoxy(1,2);
                    break;

        case '\b':  lcd_send_byte(0,0x10);
                    break;

        default    :  lcd_send_byte(1,c);
                    break;
    }
}

char lcd_getc(BYTE x, BYTE y)
{
    char value;

    lcd_gotoxy(x,y);
    while (bit_test(lcd_read_byte(),7));
    lcd.rs = 1;
    value = lcd_read_byte();
    lcd.rs = 0;
    return(value);
}
```

Codi del driver del rellotge de temps real DS1307:

```

////////////////////////////////////
////                               DS1307.C                               ////
////                               ////
////                               Driver for Real Time Clock           ////
////                               ////
//// ds1307_init() - Enable oscillator without clearing                 ////
////                               the seconds. Register used when PIC   ////
////                               loses power and DS1307 run from       ////
////                               3V BAT. Disable squarewave output      ////
////                               ////
//// ds1307_set_date_time(day,mth,year,dow,hour,min,sec)              ////
////                               ////
////                               - Set the date/time                    ////
////                               ////
//// ds1307_get_date(day,mth,year,dow)                                ////
////                               ////
////                               - Get the date                          ////
////                               ////
//// ds1307_get_time(hr,min,sec)                                       ////
////                               ////
////                               - Get the time                          ////
////                               ////
////                               ////
////////////////////////////////////

BYTE bin2bcd(BYTE binary_value);
BYTE bcd2bin(BYTE bcd_value);

void ds1307_init(void)
{
    BYTE seconds = 0;

    i2c_start();
    i2c_write(0xD0);           // WR to RTC
    i2c_write(0x00);           // REG 0
    i2c_start();
    i2c_write(0xD1);           // RD from RTC
    seconds = bcd2bin(i2c_read(0)); // Read seconds in DS1307
    i2c_stop();
    seconds &= 0x7F;
    delay_us(3);
    i2c_start();
    i2c_write(0xD0);           // WR to RTC
    i2c_write(0x00);           // REG 0
    i2c_write(bin2bcd(seconds));
    i2c_start();
    i2c_write(0xD0);           // WR to RTC
    i2c_write(0x07);           // Control Register
    i2c_write(0x80);           // Disable squarewave output pin
    i2c_stop();
}

```

```

void ds1307_set_date_time(BYTE day, BYTE mth, BYTE year, BYTE dow,
BYTE hr, BYTE min, BYTE sec)
{
    sec &= 0x7F;
    hr &= 0x3F;

    i2c_start();
    i2c_write(0xD0);           // I2C write address
    i2c_write(0x00);           // Start at REG 0 - Seconds
    i2c_write(bin2bcd(sec));    // REG 0 seconds (sec)
    i2c_write(bin2bcd(min));    // REG 1 minutes (min)
    i2c_write(bin2bcd(hr));     // REG 2 hours (hr)
    i2c_write(bin2bcd(dow));    // REG 3 day of week (dow)
    i2c_write(bin2bcd(day));    // REG 4 day (day)
    i2c_write(bin2bcd(mth));    // REG 5 month (mth)
    i2c_write(bin2bcd(year));   // REG 6 year (year)
    i2c_write(0x80);           // REG 7 Disable squarewave output
    i2c_stop();
}

```

```

void ds1307_get_date(BYTE &day, BYTE &mth, BYTE &year, BYTE &dow)
{
    i2c_start();
    i2c_write(0xD0);
    i2c_write(0x03);           // Start at REG 3 - dow
    i2c_start();
    i2c_write(0xD1);
    dow = bcd2bin(i2c_read() & 0x7f); // REG 3 day of week
    day = bcd2bin(i2c_read() & 0x3f); // REG 4 day
    mth = bcd2bin(i2c_read() & 0x1f); // REG 5 month
    year = bcd2bin(i2c_read(0));      // REG 6 year
    i2c_stop();
}

```

```

void ds1307_get_time(BYTE &hr, BYTE &min, BYTE &sec)
{
    i2c_start();
    i2c_write(0xD0);
    i2c_write(0x00);           // Start at REG 0 - Seconds
    i2c_start();
    i2c_write(0xD1);
    sec = bcd2bin(i2c_read() & 0x7f);
    min = bcd2bin(i2c_read() & 0x7f);
    hr = bcd2bin(i2c_read(0) & 0x3f);
    i2c_stop();
}

```

```
BYTE bin2bcd(BYTE binary_value)
{
    BYTE temp;
    BYTE retval;

    temp = binary_value;
    retval = 0;

    while(1)
    {
        if(temp >= 10)
        {
            temp -= 10;
            retval += 0x10;
        }
        else
        {
            retval += temp;
            break;
        }
    }
    return(retval);
}
```

```
BYTE bcd2bin(BYTE bcd_value)
{
    BYTE temp;

    temp = bcd_value;
    temp >>= 1;
    temp &= 0x78;
    return(temp + (temp >> 2) + (bcd_value & 0x0f));
}
```


ANNEX III
CARACTERÍSTIQUES TÈCNIQUES DELS
TRANSFORMADORS DE CORRENT

TRANSFORMADOR DE CORRIENTE CON CONVERTIDOR INCORPORADO

4...20 mA c.c.
(alimentación externa o interna)

En muchas instalaciones y procesos de control es necesario disponer de una señal en corriente entre 4 y 20 mA. Actualmente se consigue mediante un transformador de corriente .../5 A, más un convertidor 5 A / 4...20 mA.

CIRCUTOR ofrece una amplia gama de equipos que incorporan transformador-convertidor en un sólo módulo, sustituyendo los sistemas convencionales.

Estos convertidores incluyen un transformador de corriente constituido por el conductor activo que abarca el transformador como primario y un arrollamiento secundario bobinado en un núcleo ferromagnético. Todos los tipos disponen de una salida de 4...20 mA con una respuesta en frecuencia lineal entre 50 y 60 Hz.

La principal aplicación de los convertidores es su conexión a Autómatas programables. También pueden usarse en cualquier proceso industrial o sistema de control que necesite una señal lineal en corriente de 4...20 mA.

Los equipos pueden alimentarse en corriente continua (10...28 V c.c.), o bien en corriente alterna (230 V c.a.) según la serie seleccionada.

CURRENT TRANSFORMER WITH BUILT-IN CONVERTER

4...20 mA d.c.
(external or internal supply)

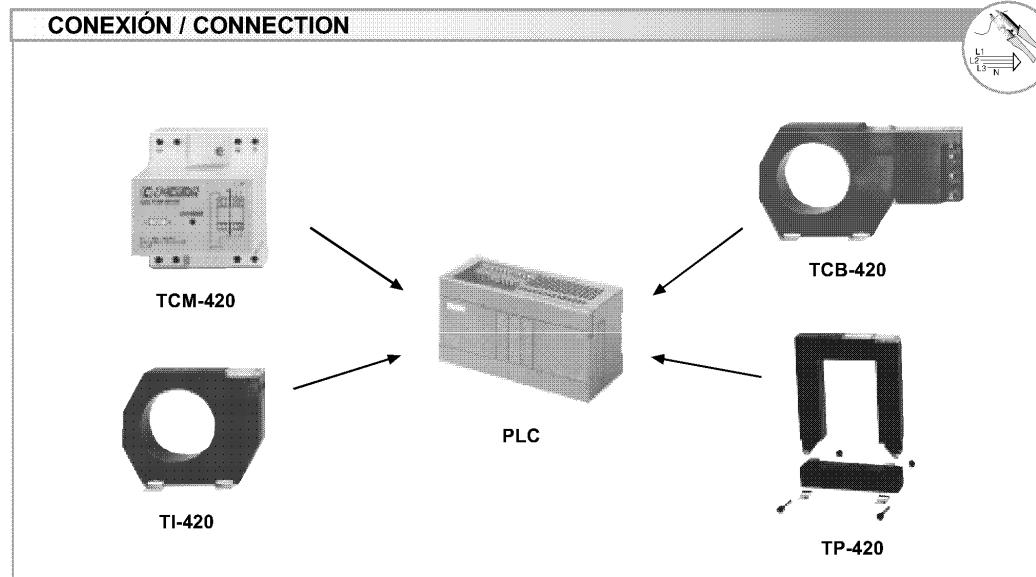
In many installations and control processes it is necessary to have a current signal between 4 and 20 mA. This signal is presently obtained by means of a .../5 A current transformer together with a 5 A / 4...20 mA converter.

CIRCUTOR offers a wide range of equipments incorporating the transformer-converter single module, which that substitute the classic systems.

These converters have a current transformer consisting of an active conductor as primary winding, and a secondary winding coiled around a ferromagnetic core. All of the types have a 4...20 mA output with a frequency response that is linear between 50 and 60 Hz.

The main application of these converters is their connection to a P.L.C. They can also be used in any industrial process or control system requiring a 4...20 mA lineal current signal.

The equipments can be supplied either with direct current (10...28 V d.c.), or with alternating current (230 V a.c.) according to the selected series.





CIRCUITOR

TRANSFORMADORES DE CORRIENTE CON CONVERTIDOR INCORPORADO A 4...20 mA c.c.

(alimentación externa de la salida 4...20 mA)

Serie TP-420

DE NÚCLEO PARTIDO

Los transformadores-convertidores TP-420 han sido especialmente diseñados para facilitar su colocación, tanto en instalaciones nuevas como en las ya existentes. Gracias a su núcleo partido permiten su instalación sin necesidad de interrumpir ningún cable o pletina.

El equipo no incluye la fuente de alimentación, por tanto debe alimentarse entre 10 y 28 V c.c.

Las características más destacadas son:

- Pequeño tamaño y facilidad de instalación
- Amplias dimensiones de la ventana interior, permitiendo abrazar cables gruesos o grandes pletinas
- Amplia gama de tamaños que permiten adaptarlos a cualquier instalación

Circuito de medida:

- Resistencia de carga: depende de la tensión de alimentación (ver fig. 1)

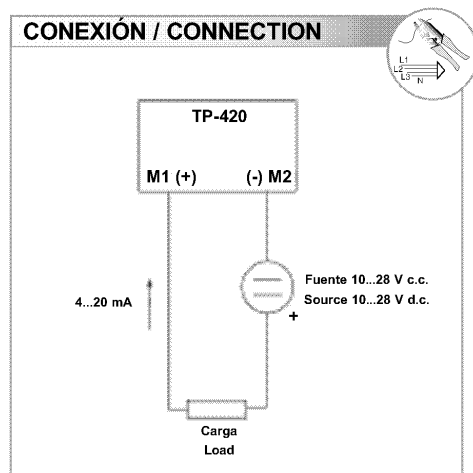
Circuito de alimentación:

- Tipo: alimentación externa de la salida 4...20 mA
- Tensión de alimentación: 10...28 V c.c.

Características constructivas:

- Equipo con núcleo practicable, que permite su apertura
- Envoltorio autoextinguible, según UL 94 VO
- Soportes metálicos para su fijación
- Bornes niquelados protegidos con tapa de plástico transparente
- Caja de bornes secundarios precintables

Dimensiones página 36



CURRENT TRANSFORMER WITH BUILT-IN CONVERTER AT 4...20 mA d.c.

(external supply of the 4...20 mA output)

SPLIT CORE

TP-420 Series



The transformer-converters of TP-420 Series have been especially designed for make their installation easy, both in new and already existing networks. Thanks to their split core they allow for their installation without opening any cable or bus bar.

The equipment doesn't include the supply source, therefore it has to be supplied between 10 and 28 V d.c.

The most important characteristics are:

- Small size and easy installation
- Wide inner window, allowing for the clamping of big cables or bus bars
- Wide range of sizes to accommodate all the existing installations

Measurement circuit:

- Load resistor: depending on the voltage supply (see fig. 1)

Supply circuit:

- Type: external supply of the 4...20 mA output
- Supply voltage: 10...28 V d.c.

Constructive characteristics:

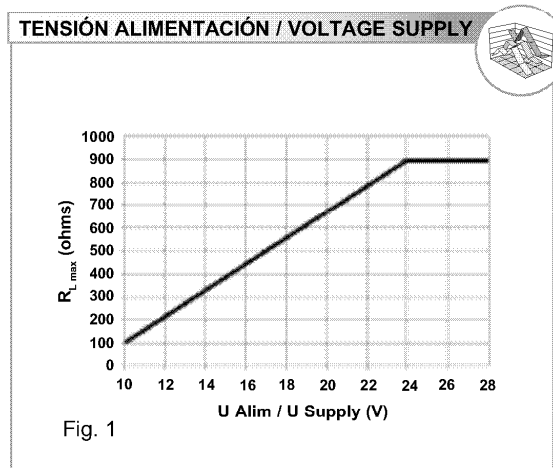
- Split core equipment
- Self-extinguishing coating, according to UL 94 VO
- Metallic holders for easy-fitting
- Nickel plated connection terminals protected with a transparent cover
- Sealable connection terminal box at the secondary side

Dimensiones page 36

Código y tipo Code and type	Sección útil Useful window ϕ mm	Primario Primary A	Código y tipo Code and type	Sección útil Useful window ϕ mm	Primario Primary A
2 23 301 TP-420-23-5	20 x 30	5	2 23 331 TP-420-812-250	80 x 120	250
2 23 302 TP-420-23-10	20 x 30	10	2 23 332 TP-420-812-500	80 x 120	500
2 23 303 TP-420-23-20	20 x 30	20	2 23 333 TP-420-812-750	80 x 120	750
2 23 304 TP-420-23-50	20 x 30	50	2 23 334 TP-420-812-1000	80 x 120	1000
2 23 305 TP-420-23-100	20 x 30	100	2 23 335 TP-420-812-1500	80 x 120	1500
2 23 306 TP-420-23-200	20 x 30	200			
2 23 307 TP-420-23-250	20 x 30	250	2 23 341 TP-420-816-500	80 x 160	500
2 23 308 TP-420-23-500	20 x 30	500	2 23 342 TP-420-816-750	80 x 160	750
			2 23 344 TP-420-816-1000	80 x 160	1000
2 23 311 TP-420-58-100	50 x 80	100	2 23 343 TP-420-816-1500	80 x 160	1500
2 23 312 TP-420-58-250	50 x 80	250	2 23 345 TP-420-816-2000	80 x 160	2000
2 23 313 TP-420-58-500	50 x 80	500	2 23 346 TP-420-816-3000	80 x 160	3000
2 23 314 TP-420-58-750	50 x 80	750	2 23 347 TP-420-816-4000	80 x 160	4000
2 23 321 TP-420-88-100	80 x 80	100			
2 23 322 TP-420-88-250	80 x 80	250			
2 23 323 TP-420-88-500	80 x 80	500			
2 23 324 TP-420-88-750	80 x 80	750			
2 23 325 TP-420-88-1000	80 x 80	1000			
2 23 326 TP-420-88-1500	80 x 80	1500			

La resistencia de carga R_L depende de la tensión de alimentación (ver fig. 1).

The load resistor R_L depends on the supply voltage (see fig. 1).





CARACTERÍSTICAS TÉCNICAS	TI-420, TP-420, TCM-420, TCB-420	TECHNICAL CHARACTERISTICS
Condiciones de trabajo		Operating conditions
Intervalo de clase	0 / +50 °C	Class range
Temperatura de trabajo	-10 / +50 °C	Operating temperature
Tensión más elevada para el material (U _m)	0,72 kV c.a. / a.c.	Highest voltage for equipment (U _m)
Circuito de medida		Measurement circuit
Primario pasante		Passing bar primary
Corriente primaria	Según tipos / Depending on types	Primary current
Respuesta en frecuencias lineal	50...60 Hz	Frequency linear response
Corriente secundaria	4...20 mA c.c. / d.c.	Secondary current
Precisión (entre el 5...110 % I _n)	± 1,5 I _n lectura / of reading	Accuracy (within the 5...110 % I _n)
Sobrecargas (a temperatura ambiente)	1,5 I _n permanentemente / permanently	Overloads (at operation temperature)
Normas	IEC 44-1, UNE 21 088-1, IEC 664, VDE 0110, VDE 0414, UL 94, IEC 1010-1, EN 61010-1	Standards