

Treball Final de Carrera

Regulador de llum controlat per ordinador

David Martín Humanes

Enginyeria Tècnica Industrial. Esp. Electrònica Industrial

Director: Antoni Suriñach i Albareda

Vic, Juny de 2010

Els més sincers agraïments a l'Antoni Suriñach per accedir a ser el director del treball, a en Domènec Iborra per fer tot el possible per cedir-me els mòduls de radiofreqüència, a l'Agustí Bartrina per l'aportació de diversos components i molt especialment a en Josep Serra, professor de l'Escola del treball del Ripollès que sense la seva ajuda no hagués estat possible la construcció física d'aquest regulador de llum.

ÍNDEX

RESUM	4
1. OBJECTE DEL PROJECTE.....	6
2. OBJECTIUS.....	7
3. ESPECIFICACIONS.....	8
4. UTILITAT I SITUACIÓ.....	9
5. ANTECEDENTS	10
5.1. REGULADORS DE LLUM SILENCIOSOS.....	10
5.2. REGULADORS DE LLUM INDUCTIUS GIRATÒRIS	10
5.3. REGULADORS INTERNS	11
5.4. REGULADORS EXTERNS.....	11
5.5. REGULADORS APLICABLES A ENDOLLS.....	12
5.6. REGULADORS INDUSTRIALS MONOFÀSICS	12
5.7. REGULADORS AMB AUTOTRANSFORMADORS.....	13
5.8. REGULADOR DE LLUM AMB COMANDAMENT.....	13
5.9. REGULADOR CONTROLAT PER INFRAROJOS	14
5.10. REGULADOR DE LLUM CONTROLAT PER PWM.....	14
5.11. REGULADORS CONTROLATS PER PC	14
5.12. CONTROL DE LLUMS D'ESCENARI	15
6. DIAGRAMA DE BLOCS GENERAL.....	16
7. DISSENY	17
7.1. POSSIBLES SOLUCIONS I SOLUCIÓ ADOPTADA	18
7.1.1. PROGRAMA INFORMÀTIC A UTILITZAR.....	18
7.1.2. PORT A UTILITZAR.....	18
7.1.3. COMUNICACIÓ DEL PC AMB LA PLACA ELECTRÒNICA.....	19
7.1.4. CIRCUIT DE CONTROL.....	21
7.1.5. CIRCUIT DE POTÈNCIA.....	23
7.1.6. SOLUCIÓ ADOPTADA.....	24
7.2. DESCRIPCIÓ GENERAL	25
7.2.1. APLICACIÓ VISUAL BASIC	25
7.2.2. ALIMENTACIÓ DEL CIRCUIT	34
7.2.3. CIRCUIT DE PAS PER 0	35
7.2.4. CIRCUIT DE CONTROL.....	37
7.2.5. CIRCUIT DE POTÈNCIA.....	51
7.2.6. PROCÉS DE DISSENY I CONSTRUCCIÓ DEL CIRCUIT IMPRÈS	56
7.3. RESOLUCIÓ D'IMPREVISTOS	72

7.3.1.	IMPREVISTOS DE RECERCA	72
7.3.2.	IMPREVISTOS DE CONSTRUCCIÓ DEL PROTOTIP.....	73
7.3.3.	IMPREVISTOS DE DISSENY DEL PCB	73
7.3.4.	IMPREVISTOS DE FABRICACIÓ DEL CIRCUIT IMPRÈS	74
7.3.5.	IMPREVISTOS DEL PROTOTIP	74
7.3.6.	IMPREVISTOS NO SOLUCIONATS	75
8.	INSTRUCCIONS D'ÚS	76
8.1.	REGULADOR DE LLUM AMB CABLE USB.....	76
8.2.	REGULADOR DE LLUM RADIOFREQUÈNCIA.....	76
8.3.	CANVI DE DISPOSITIU AMB EL PROGRAMA EN MARXA	76
8.4.	PROGRAMA “REGULADOR DE LLUM PC”	77
8.5.	IMPREVISTOS.....	79
9.	PROVES I RESULTATS.....	80
9.1.	PROVES.....	80
9.1.1.	SISTEMES OPERATIUS	80
9.1.2.	REGULACIÓ DE VELOCITAT DE MOTORS	80
9.1.3.	CONSUM.....	80
9.2.	RESULTATS	83
10.	COST DEL PROJECTE	84
10.1.	PLACA DE CIRCUIT IMPRÈS	84
10.2.	ENCAPSULAT	86
10.3.	CONSUMIBLES.....	86
10.4.	COST TOTAL.....	87
11.	CONCLUSIONS	88
12.	BIBLIOGRAFIA	89
A.	ANNEX.....	90
A.1.	DVD ADJUNT.....	90
A.2.	PLÀNOLS	90
A.3.	DIAGRAMA DE BLOCS CODI PIC.....	95
A.4.	CODI MICROCONTROLADOR PIC16F876	96
A.5.	CODI VISUAL BASIC.....	110
A.6.	FOTOLITS	164



Resum del projecte

Enginyeria Tècnica Industrial. Esp, Electrònica Industrial

Títol: Regulador de llum controlat per ordinador

Paraule clau: control, ordinador, intensitat lumínica, focus, disseny electrònic, circuit.

Autor: David Martín Humanes

Direcció: Antoni Suriñach i Albareda

Data: Juny de 2010

Resum

El llum elèctric és un tipus d'energia amb la que s'il·lumina tot el món i s'utilitza tant per a il·luminar la nit com per a disposar de llum addicional durant el dia. L'energia es pren directament de la xarxa de subministrament elèctric i permet encendre tot tipus de focus i bombetes.

Actualment la necessitat de controlar la intensitat lumínica de focus és de gran utilitat i es poden veure exemples en escenaris de teatres, concerts musicals, domòtica bàsica a vivendes, botigues, restaurants, etc. on s'incorporen aparells òptims per aquest control. Aspectes com la programació d'encesa, apagat i intensitat desitjada de focus a una hora convinguda facilita el fet de fer-ho manualment i disposar de més temps propi.

L'objectiu principal d'aquest treball és dissenyar i construir un regulador de llum controlat per ordinador capaç de regular la intensitat lumínica de 8 focus independentment l'un de l'altre. El control de regulació s'efectua mitjançant un programa informàtic compatible amb ordinadors que incorporin el sistema operatiu Windows i és programable en el temps permetent seleccionar la intensitat desitjada a diferents hores del dia seleccionat.

Com a conclusions es pot destacar un estalvi energètic al regular la intensitat dels focus evitant així la permanent connexió a una tensió màxima de 230 VAC i la oportunitat de construir un regulador de llum amb els documents subministrats.



Project Summary

Industrial Technical Engineering. Esp. Industrial Electronics

Title: Computer controlled dimmer

Key words: control, PC, light intensity, focus, electronic design, circuit.

Author: David Martín Humanes

Direction: Antoni Suriñach i Albareda

Date: June 2010

Summary

The electric light is a kind of energy that illuminates the whole world, being used to illuminate the night and to provide additional light during the day. This light is powdered directly from the electricity grid and allows the activation of all types of light bulbs and focus.

Currently the need to control the light intensity of focus is nowadays useful and it can be seen on stage theatres, concerts, home automation, shops, restaurants, etc. where optimal devices are incorporated for this control. Aspects like the on, off and the required intensity focus programming at the desired hour, makes it easier than do it manually and you can have more own time.

The main objective of this work is to design and build a computer controlled dimmer to regulate the light intensity of 8 focuses independently of each other. The regulation control is performed by a computer program compatible with computers that include Windows operating system and it is programmable in time allowing to select the light intensity to be regulated at the wished hour of the selected day.

The most out-standing conclusions are the energy saving when the light intensity is regulated avoiding the permanent connection to a maximum voltage of 230 VAC and the opportunity of build a computer controlled dimmer with the documents supplied.

1. OBJECTE DEL PROJECTE

A partir dels coneixements informàtics i electrònics adquirits, es vol dissenyar i construir un regulador de llum controlat per ordinador. Es pretén que aquest regulador sigui capaç de controlar la intensitat lumínica de varis focus de manera independent i conjunta mitjançant equalitzadors mòbils i botons d'opció d'encesa, apagat i efectes. Aquests controls es troben situats en un programa informàtic que transmet les dades necessàries per a la gestió d'intensitat lumínica a través del port USB de l'ordinador el qual comunica amb un circuit electrònic.

2. OBJECTIUS

Per a poder controlar el procés de regulació desitjat s'utilitzen cinc elements clau:

- Programa informàtic
- Comunicació amb cable USB
- Sincronització de xarxa elèctrica
- Electrònica de control
- Electrònica de potència

Partint de la base d'un programa informàtic que crea una interfície entre l'usuari i l'ordinador, es pretén transmetre dades pel port USB en el qual es troba connectat un cable que comunica amb un circuit electrònic sincronitzat amb la xarxa elèctrica. Les dades rebudes són introduïdes en un microprocessador encarregat de gestionar la regulació dels focus mitjançant l'electrònica de potència.

A l'hora de dur a terme el disseny i muntatge del regulador, es vol afegir l'opció d'escollir entre una comunicació per cable entre l'ordinador i el circuit electrònic, o bé una comunicació mitjançant dispositius de radiofreqüència.

Les principals característiques desitjades pel producte final són:

- Flexibilitat en la instal·lació aconseguint que la manipulació dels components no esdevingui un problema.
- Econòmic en quant a la seva fabricació.
- Fàcil manipulació. Això implica, per una banda, que el dispositiu es pugui connectar amb qualsevol ordinador que disposi de sistema operatiu Windows i port USB sense cap mena de complicació. D'altra banda, el resultat ha de ser un dispositiu compacte, que es pugui transportar sense problemes.
- Facilitat d'ús. El programa a dissenyar ha de tenir una interfície gràfica adequada per aconseguir que l'usuari el pugui entendre i utilitzar sense dificultat.

La realització d'aquest projecte suposa una gran motivació ja que el dispositiu a dissenyar té una clara i gran utilitat i, a més, permet assentar i assimilar els coneixements adquirits durant els estudis referents a electrònica i informàtica. D'altra banda, també es volen assolir nous coneixements, com poden ser la programació d'una interfície gràfica entre l'usuari i l'ordinador, el tractament de dades procedents del PC, el tractament d'un senyal procedent de la xarxa elèctrica i el control d'un circuit de potència.

3. ESPECIFICACIONS

Per a dur a terme el disseny i la construcció d'un regulador de llum controlat per ordinador s'han previst unes especificacions a seguir, que se citen a continuació:

- Control mitjançant un ordinador amb sistema operatiu Windows
- Interfície gràfica de control entre usuari i ordinador
- Comunicació per cable mitjançant el port USB
- Gestió del regulador mitjançant un microcontrolador
- Regulació d'intensitat lumínica de 0 a 230 VAC
- Regulació de velocitat de motors AC
- Regulació de 8 canals simultàniament
- Potència màxima de 100W per canal
- Intensitat màxima de 3,5 A
- Intensitat lumínica programable
- Efectes lumínics
- Habilitació d'un sistema per a una possible comunicació per radiofreqüència

4. UTILITAT I SITUACIÓ

El regulador de llum consta bàsicament d'un ordinador des del qual l'usuari dona les ordres d'il·luminació i que es comunica amb un circuit electrònic en el que hi ha connectats els focus. La principal utilitat del regulador de llum és la de regular la intensitat lumínica d'aquests focus de forma independent utilitzant diferents canals.

La utilització d'aquest aparell pot fer referència a varis camps entre els quals es destaquen els següents:

- **Il·luminació controlada d'escenaris**

Utilització del regulador de llum per al control lumínic d'escenaris de teatre, escenaris musicals, sales de conferència, etc. Apte tant per a la regulació de llum com per a la creació d'efectes lumínics.

- **Domòtica bàsica**

Utilització del regulador de llum per al control lumínic d'una vivenda, jardí, pati, botiga, restaurant, etc... des d'un punt central on anirà situat l'ordinador.

La instal·lació d'aquest aparell haurà de ser simple, realitzant les mínimes connexions possibles i, segons el tipus de comunicació que s'utilitzi, l'aparell podrà anar situat en un lloc o en un altre. Si es realitza una comunicació sense cables, l'aparell es pot situar a prop dels focus; d'altra banda, si és amb cable, el regulador haurà de situar-se prop de l'ordinador i lluny de la instal·lació dels focus.

5. ANTECEDENTS

Actualment al mercat existeix una àmplia gamma de reguladors de llum. Hi ha reguladors de diferents característiques i diferents aspectes per adaptar-se a cada aplicació en concret.

En els següents apartats es fa una breu explicació sobre les característiques i funcionalitat dels reguladors més destacables.

5.1. REGULADORS DE LLUM SILENCIOSOS

Utilitzant la més avançada tecnologia de semiconductors de potència, IGBT, s'ha aconseguit introduir en una mida mínima tot un sistema de regulació commutat a baixa freqüència, amb el que s'aconsegueix un nivell de soroll 0, i un nivell de radiacions interferents totalment nul. Són reguladors especialment indicats per a llums de capçalera, hospitals, avionica, i ambients silenciosos.

Permet regular tensions de 0 a 230 VAC i de 0 a 110 VAC.



Fig. 5.1. Regulador de llum silencios

5.2. REGULADORS DE LLUM INDUCTIUS GIRATÒRIS

Reguladors dissenyats especialment per a la regulació de tensió/fase, sobre càrregues inductives. Especialment indicat per a la regulació d'instal·lacions d'il·luminació amb transformadors adaptats, motors, ventiladors, extractors, etc.

Permeten regular tensions de 0 a 230 VAC.



Fig. 5.2. Reguladors de llum inductius giratoris

5.3. REGULADORS INTERNS

Reguladors de llum resistius especialment ideats per a ocupar el mínim espai a l'interior de qualsevol làmpada. Sotmesos a un exclusiu procés d'impermeabilització per resina, se'ls permet treballar en ambients humits, vibrants o agressius. Reguladors versàtils i adaptats a tots els tipus de subjecció.

Permet regular tensions de 0 a 230 VAC.

5.4. REGULADORS EXTERNS

Reguladors externs accionats manualment o amb el peu. Es poden aplicar a qualsevol làmpada sense necessitat de modificacions en la mateixa. El seu perfecte disseny admet la regulació de làmpades halògenes i de petits dispositius a motor, com per exemple ventiladors i extractors. Amb indicació per pilot led.

Permet regular tensions de 0 a 230 VAC.



Fig. 5.3. Regulador de llum extern

5.5. REGULADORS APLICABLES A ENDOLLS

Es connecten directament a l'endoll i regulen la intensitat de llum mitjançant un potenciòmetre existent en el seu mateix encapsulat.

Permet regular tensions de 0 a 230 VAC.



Fig. 5.4. Regulador aplicable a endolls

5.6. REGULADORS INDUSTRIALS MONOFÀSICS

Reguladors d'intensitat lumínica o de velocitat de motors que permeten un ajust intern de la velocitat mínima i de la velocitat màxima. La seva gamma inclou versions d'1,5 a 10 A. Es poden subministrar en dues versions: amb una regulació de mínim a màxim, i amb una regulació de màxim a mínim.

Permeten regular tensions de 0 a 230 VAC.



Fig. 5.5. Regulador industrial monofàsic

Dins d'aquesta sèrie es poden trobar els reguladors ERV que tenen una entrada que permet controlar el regulador mitjançant un voltatge extern de 0 a 10 VDC o un corrent de 4 a 20 mA.

5.7. REGULADORS AMB AUTOTRANSFORMADORS

Basats en el control del voltatge a través d'autotransformadors, es poden utilitzar per a controlar el voltatge de motors i lluminàries. Incorporen un selector que permet seleccionar 5 nivells de voltatge diferent.

Aquests reguladors es poden trobar en format monofàsic i trifàsic. En els autotransformadors monofàsics es poden trobar models d'1,5 fins a 10 A i poden seleccionar tensions fixes de 0 a 230 VAC. En els trifàsics es poden trobar models de 5 a 12 A amb una regulació de 0 a 380 VAC.



Fig. 5.6. Regulador amb autotransformador

5.8. REGULADOR DE LLUM AMB COMANDAMENT

Conjunt compost per un comandament múltiple emissor de control i un regulador de llum endollable a una presa de corrent de 230V. Dissenyat per a encendre, apagar i regular la intensitat lumínica a distància de làmpades que disposin d'endoll.

És apte per a làmpades incandescentes i halògenes de 230 VAC sense transformador i sense regulador de llum ja incorporat. Té un abast de 50 metres en camp obert.



Fig. 5.7. Comandament i regulador de llum

5.9. REGULADOR CONTROLAT PER INFRAROJOS

Regulador de llum que conté un emissor i un receptor que es comuniquen per infrarojos. Gràcies al seu disseny compacte, aquests circuits poden ser allotjats a les bases d'encastar estàndard.

La intensitat màxima que poden suportar és de 2 A, i de 4 A amb dissipador. Abasta una distància màxima de 7 metres.

Permet regular tensions de 0 a 230 VAC.

5.10. REGULADOR DE LLUM CONTROLAT PER PWM.

Els reguladors de llum PWM (modulació per amplada de polsos) ofereixen una eficàcia energètica i un complet silenci. Aquesta tècnica és utilitzada per a controlar l'amplitud de l'ona que alimenta la làmpada. En termes bàsics, l'alimentació d'entrada es fa mitjançant un mostreig a altes freqüències (40 kHz) i equips IGBT controlats durant cada període de mostreig mitjançant la variació de la ràtio de temps On/Off.

El període d'encesa és proporcional a la potència requerida per a coincidir amb la forma sinusoidal en cada punt del cicle.

Els processadors PWM afegeixen menys d'un 1% de distorsió a la alimentació principal, i això fa que aquests reguladors siguin completament silenciosos i amb una completa facilitat a l'hora de regular qualsevol tipus de càrrega.

5.11. REGULADORS CONTROLATS PER PC

Avui dia existeixen reguladors de llum controlats per un PC i per microcontrolador que realitzen funcions bastant complexes. Es controlen des del port sèrie o USB del PC i, a part de regular la intensitat de la llum, poden fer diverses funcions com la de crear un programador horari, és a dir, poden activar o desactivar la tensió en unes hores determinades i amb la potència programada. També poden reproduir una seqüència gravada a la memòria EEPROM del microcontrolador des del PC. Una altra característica (comuna amb el projecte que s'està realitzant) és la de controlar la intensitat del focus mitjançant un equalitzador.

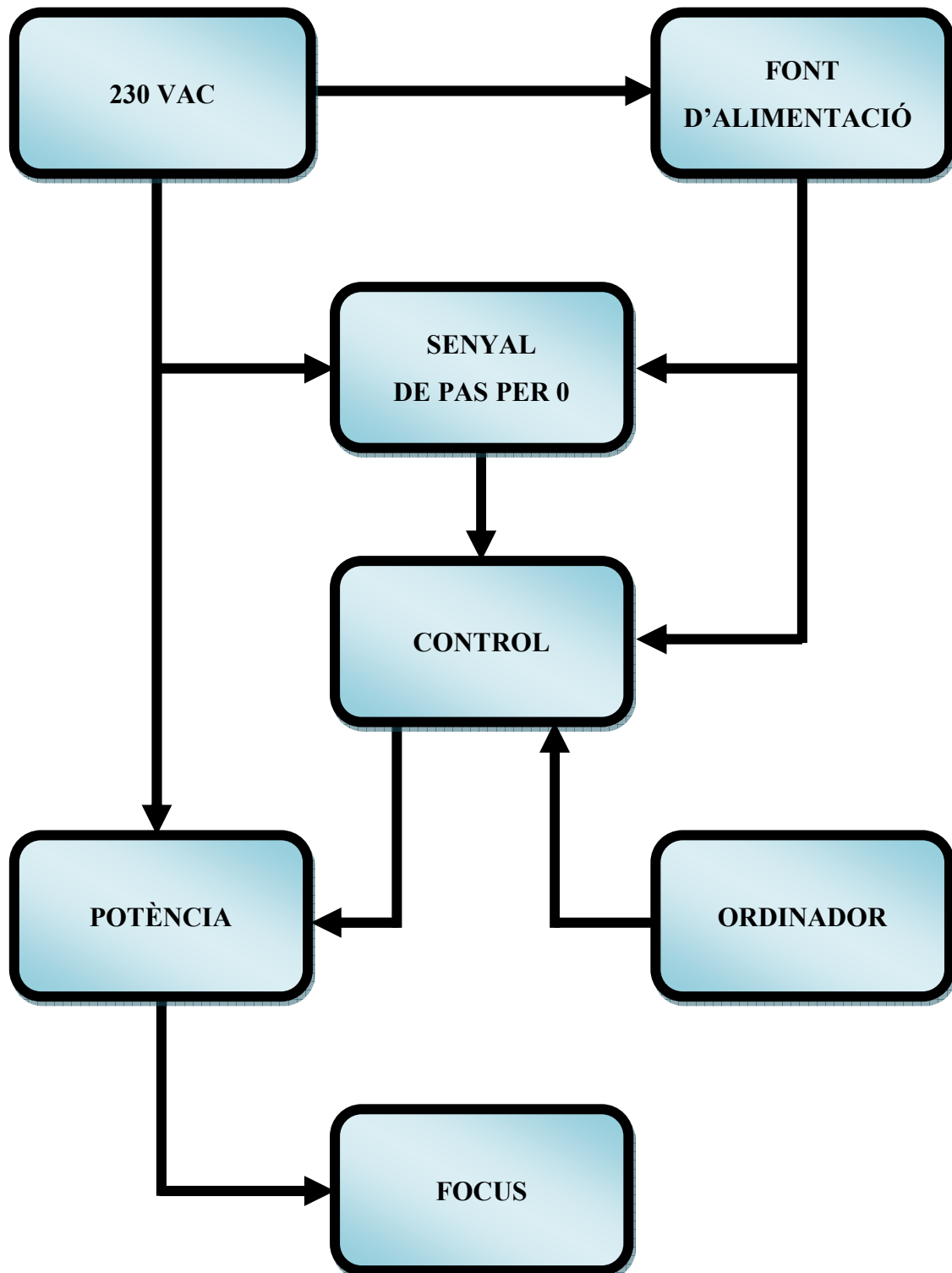
5.12. CONTROL DE LLUMS D'ESCENARI

Aquest control es presenta en taules de varis canals. Permet regular la intensitat de llum dels focus d'un escenari, moure els focus i fixar efectes. Porten incorporats una memòria flash d'un funcionament similar a la d'un disc dur. Gràcies a aquesta memòria es poden guardar el nivell de regulació, els efectes i el moviment desitjats així com implementar efectes que la memòria porta per defecte.



Fig. 5.8. Control de llum d'escenari

6. DIAGRAMA DE BLOCS GENERAL



7. DISSENY

Per dur a terme un projecte com aquest és necessari tenir clar què es vol construir i de quins mitjans es disposa, per a seleccionar els que més s'adapten a la nostra necessitat ja sigui en qüestions de disseny, com de preu.

El primer pas ha estat realitzar una recerca d'informació general, amb l'objectiu d'iniciar-se en el món dels reguladors de llum. Una vegada s'ha analitzat aquesta informació es passa a una recerca més concreta fins a trobar informació sobre com i amb quines eines i components electrònics es pot construir un regulador de llum controlat per ordinador tant a nivell de hardware com de software. Quan s'han seleccionat aquestes eines i s'han fet les proves corresponents, és llavors quan s'ha construït el circuit electrònic i s'han dut a terme els programes per a gestionar-lo. Són varies les solucions que es poden implementar a l'hora de dur a terme el projecte, però només una és la que s'adapta a les nostres necessitats.

Pel que fa a la normativa a seguir en el projecte, s'ha complert per una banda la normativa del Reglament Electrotècnic de Baixa Tensió (REBT) i la normativa RoHs (Restriction of Hazardous Substances).

Del REBT es poden destacar les següents normes:

- **ITC-BT-24:** PROTECCIÓ CONTRA CONTACTES DIRECTES I INDIRECTES
- **ITC-BT-36:** INSTAL·LACIONS A MOLT BAIXA TENSIÓ
- **ITC-BT-44:** INSTAL·LACIÓ DE RECEPTORS (ENLLUMENAT)
- **UNE 20460.5:** INSTAL·LACIONS ELÈCTRIQUES EN EDIFICIS
- **UNE 21031:** CABLES DE TENSIÓ INFERIOR O IGUAL A 450/750V

La normativa RoHs restringeix l'ús dels següents materials perillosos en la fabricació de circuits elèctrics i electrònics:

- Pb: Plom
- Hg: Mercuri
- Cd: Cadmi
- Cr IV: Crom IV
- PBB: substància retardant de flames utilitzades en alguns plàstics
- PBDE: substància retardant de flames utilitzades en alguns plàstics

Tot seguit es comenten amb deteniment les pautes que s'han seguit en la realització d'aquest projecte.

7.1. POSSIBLES SOLUCIONS I SOLUCIÓ ADOPTADA

A l'hora de valorar les possibles solucions s'han tingut en compte diversos aspectes:

- **Programa informàtic a utilitzar**
- **Port de l'ordinador a utilitzar**
- **Comunicació del PC amb la placa electrònica**
- **Circuit de control**
- **Circuit de potència**

7.1.1. PROGRAMA INFORMÀTIC A UTILITZAR

Com que el regulador de llum es controla mitjançant un ordinador personal, s'ha d'implementar un programa que realitzi les funcions desitjades. Per fer aquest programa s'han estudiat dues eines:

- **Visual Basic 2008**
- **Labview**

El Visual Basic 2008 constitueix un IDE (entorn de desenvolupament integrat) que ha estat empaquetat com un programa d'aplicació, es a dir, consisteix en un editor de codi (programa amb el que s'escriu el codi font), un depurador (programa que corregeix errors), un compilador (programa que tradueix el codi font a llenguatge màquina), i un constructor d'interfície gràfica (programació de forma visual). D'aquesta manera, permet crear programes i aplicacions.

El LabView és una eina gràfica per a probes, control i disseny mitjançant la programació amb un llenguatge gràfic. És útil per a realitzar programes que controlin aplicacions electròniques.

7.1.2. PORT A UTILITZAR

La transmissió de dades entre l'ordinador i la targeta electrònica s'ha de dur a terme mitjançant un port de l'ordinador. Els diferents ports que es poden utilitzar són:

- **Port sèrie**
- **Port paral·lel**
- **Port USB d'alta velocitat (2.0)**

El port sèrie és una interfície de comunicacions de dades digitals en la que la informació es transmet bit a bit.

El port paral·lel és una interfície de comunicacions en la que la informació es transmet enviant un byte cada vegada.

Tots dos comparteixen els mateixos avantatges i inconvenients.

Avantatges:

- Fàcil programació del port

Inconvenients:

- Pràcticament ja no s'utilitza

El port USB permet connectar perifèrics a un ordinador i alimentar-los sense necessitat d'una font exterior. Una de les seves principals característiques és la seva capacitat plug & play, concepte que es refereix a que només amb connectar el dispositiu al servidor central, aquest sigui capaç d'interpretar la informació del dispositiu i donar-lo d'alta immediatament.

Avantatges:

- Actualment és el port més utilitzat
- Alta velocitat de transmissió i recepció
- Plug & Play

Inconvenients:

- Dificil programació

7.1.3. COMUNICACIÓ DEL PC AMB LA PLACA ELECTRÒNICA

Per tal de comunicar l'ordinador amb la placa electrònica s'han plantejat tres solucions:

- **Comunicació per cable**
- **Comunicació inalàmbrica**
- **Comunicació mixta**

La comunicació per cable consisteix en connectar un cable USB A mascle / B mascle entre el port USB de l'ordinador i la targeta electrònica.



Fig.7.1. Cable USB A mascle/B mascle

Avantatges:

- Comunicació amb un preu econòmic
- Fàcil instal·lació

Inconvenients:

- Dependència de la placa electrònica amb l'ordinador
- Increment de la llargada del cable per a connectar els focus

El cable limita la distància entre la placa electrònica i el PC que hauran d'estar relativament a prop l'un de l'altre. D'aquesta manera la situació dels focus del regulador de llum es veurà obligada a utilitzar una instal·lació de cables amb una major llargada.

La comunicació inalàmbrica es fa mitjançant mòduls de radiofreqüència i permet connectar un emissor al port USB de l'ordinador que emet a un receptor situat a la targeta electrònica.

Avantatges:

- Comunicació sense cables
- Reducció de la llargada dels cables per a connectar els focus

Inconvenients:

- Increment del preu del producte

Fent ús d'una comunicació sense cables es pot controlar el regulador de llum utilitzant un ordinador situat a una distància considerable de la placa electrònica. La targeta electrònica es pot situar molt a prop dels focus i reduir així la distància del cablejat d'aquests. D'aquesta manera hom es pot moure amb total llibertat dins del rang d'abast dels mòduls de radiofreqüència regulant així la intensitat dels focus.

La comunicació mixta es basa en utilitzar les dues solucions anteriors i així poder construir un regulador que per una banda permeti la selecció d'una comunicació inalàmbrica i, per l'altra, una comunicació amb cable.

Avantatges:

- Els usuaris poden triar quina comunicació els és més adient
- En cas de fallida d'una de les dues, es pot utilitzar l'altra

7.1.4. CIRCUIT DE CONTROL

El circuit de control és l'encarregat de gestionar les dades procedents de l'ordinador per tal de regular la intensitat dels focus.

Per a dissenyar aquest circuit de control es parteix de la base de quatre possibles solucions:

- **PIC 18F2550**
- **Expand I/O USB**
- **USB - 232 i PIC 16F876**
- **Mòduls de radiofreqüència i PIC 16F876**
- **Control mixt**

El PIC 18F2550 disposa d'un mòdul per a comunicacions USB que permet establir una comunicació entre el microcontrolador i el port. Existeixen diverses maneres de fer-ho, de les quals se'n destaquen dues:

- Programació visual bàsic
- EasyHID

EasyHID és un software que genera dos arxius, un en visual bàsic per a comunicar el port USB amb el PIC i l'altre en Pic Basic Pro per a gravar-lo al microcontrolador.

Avantatges:

- Utilització d'1 sol microprocessador
- Reducció de la mida del circuit imprès

Inconvenients:

- Díficil programació
- Útil només en un port USB

L'Expand I/O USB és un firmware implementat en un PIC18F2450 subministrat per la casa anglesa HexWax que permet la comunicació entre l'ordinador i la targeta electrònica.

És un dispositiu programable mitjançant únicament comandes de 4 bytes procedents de l'ordinador.

Avantatges:

- *Plug & Play*
- No necessita *drivers* d'instal·lació
- Preu econòmic
- Utilització d'1 sol integrat
- Reducció de la mida del circuit imprès

Inconvenients:

- Dificil programació

És un dispositiu que es pot connectar i funcionar a l'instant en qualsevol port USB.

Una altra solució és utilitzar l' USB - 232 que també és un firmware implementat en un PIC 18F2450 proporcionat per HexWax. Aquest dispositiu transforma dades USB a dades sèries asíncrones. D'aquesta manera es poden introduir en un PIC 16F876 que les interpreta.

Avantatges:

- *Plug & play*
- No necessita *drivers* d'instal·lació
- No necessita programació
- Preu econòmic

Inconvenients:

- Utilització de dos integrats
- Increment de la mida del circuit imprès

Utilitzant l'USB-232, únicament s'ha d'introduir el codi necessari al microcontrolador PIC 16F876 per a que el circuit de control funcioni.

Una altra opció és utilitzar mòduls de radiofreqüència i un PIC 16F876, d'aquesta manera es pot establir una comunicació inalàmbrica. El mòdul de radiofreqüència

interpreta les dades USB i les transforma a dades sèrie que arriben directament al microprocessador.

Avantatges:

- No necessita programació
- Comunicació sense cables
- Utilització d'un sol microprocessador

Inconvenients:

- Preu elevat

Finalment, l'última opció és la de realitzar un control mixt implementant un circuit electrònic el qual permeti triar entre utilitzar l'USB-232 o bé els mòduls de radiofreqüència. Totes dues opcions van complementades amb un PIC 16F876.

7.1.5. CIRCUIT DE POTÈNCIA

El circuit de potència s'encarrega de la regulació dels focus en funció del paràmetre que hagi rebut del circuit de control. Les solucions possibles per a aquest circuit es troben fent ús dels següents elements:

- Tiristor
- Triac
- IGBT

El tiristor és un semiconductor de quatre capes i tres terminals. El seu funcionament és similar al d'un díode però amb la particularitat que el pas de tall a conducció es realitza de forma controlada mitjançant un terminal anomenat porta, essent el sentit del corrent des de ànode a càtode quedant bloquejat el sentit invers de circulació.

Avantatges:

- Reducció de soroll al filament del focus

Inconvenients:

- Preu elevat
- Control d'1 sol semicicle de la senyal sinusoïdal

Un triac és un semiconductor basat en la connexió de dos tiristors en antiparal·lel i compartint la mateixa porta.

Avantatges:

- Tecnologia simple i econòmica
- Control de dos semicicles de la senyal sinusoidal

Inconvenients:

- Aparició de soroll al filament dels focus

El transistor bipolar de porta aïllada IGBT és un dispositiu semiconductor que s'aplica com a interruptor controlat en circuits d'electrònica de potència.

Aquest dispositiu posseeix la característica dels senyals de porta dels transistors d'efecte de camp amb la capacitat de l'elevat corrent i voltatge del transistor bipolar.

Avantatges:

- Soroll pràcticament nul al filament del focus

Inconvenients:

- Preu elevat

7.1.6. SOLUCIÓ ADOPTADA

Prèvia avaluació dels avantatges i inconvenients de totes les solucions possibles s'ha arribat a la conclusió que la solució adoptada per a dur a terme aquest projecte és la següent:

- **Programa informàtic a utilitzar:** Visual Basic
- **Port de l'ordinador a utilitzar:** Port USB
- **Comunicació del PC amb la placa electrònica:** Comunicació mixta
- **Circuit de control:** Control mixt
- **Circuit de potència:** Triac

Cal destacar que la comunicació entre el PC i el circuit electrònic serà una comunicació a través de cable. A més a més i com a segona opció, s'implementarà un sistema tant a nivell informàtic com a nivell de circuit electrònic apte per a establir una comunicació sense cables mitjançant mòduls de radiofreqüència.

7.2. DESCRIPCIÓ GENERAL

Es presenta una descripció de la composició i el funcionament d'aquest projecte, estructurada i segmentada en les diverses parts que el formen:

- **Aplicació Visual Basic**
- **Alimentació del circuit**
- **Circuit de pas per 0**
- **Circuit de control**
- **Circuit de potència**

7.2.1. APLICACIÓ VISUAL BASIC

Per poder controlar la intensitat dels focus mitjançant un ordinador s'ha adaptat una aplicació dissenyada i programada en un entorn de programació anomenat Visual Basic. L'aplicació rep el nom de “*generic hid*” i és un software lliure que es pot trobar a Internet.

El propòsit d'aquest software és comunicar el port USB de l'ordinador amb un dispositiu genèric d'interfície humana (HID) utilitzant el sistema operatiu Windows.

Funciona amb les versions Windows 98 i posteriors.

El software troba el dispositiu connectat mitjançant el Vendor ID (codi del fabricant) i el Product ID (codi del producte) de què disposen els dispositius USB.

L'aplicació “*generic hid*” conté els següents mòduls:

- **GenericHid.cs:** accés a l'aplicació.
- **FrmMain.cs:** rutines específiques al formulari.
- **Hid.cs:** rutines específiques per a les comunicacions.
- **DeviceManagement.cs:** rutines per a la obtenció de dispositius.
- **Debugging.cs:** conté una rutina per a mostrar els errors d' API's.
- **HidDeclarations.cs:** declaracions per funcions API utilitzades per Hid.cs.
- **FileIODeclarations.cs:** declaracions per l'arxiu relacionat a les funcions Device- Management.cs.
- **Declarations.cs:** declaracions per a les funcions API utilitzades per Device- Management.cs.
- **DebuggingDeclarations.cs:** declaracions per a les funcions API utilitzades per Debugging.cs.

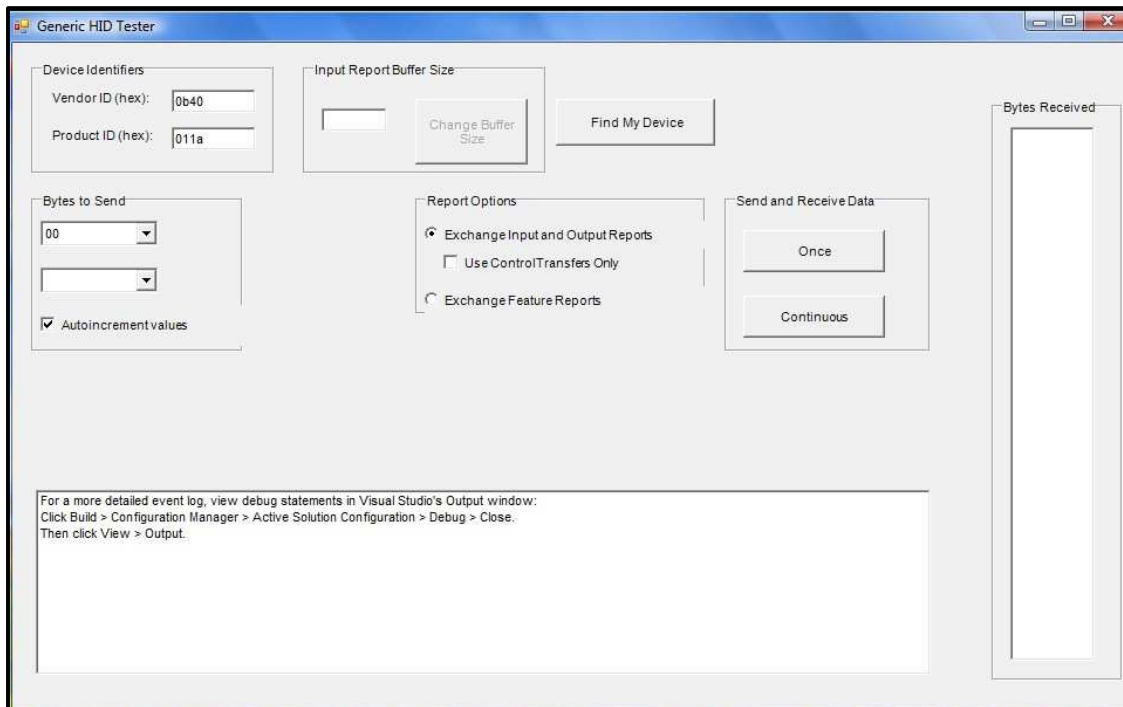


Fig. 7.2. Interfície "generic hid"

Per tal d'adaptar el programa "generic hid" a les necessitats d'un regulador de llum s'ha modificat el codi i l'entorn visual.

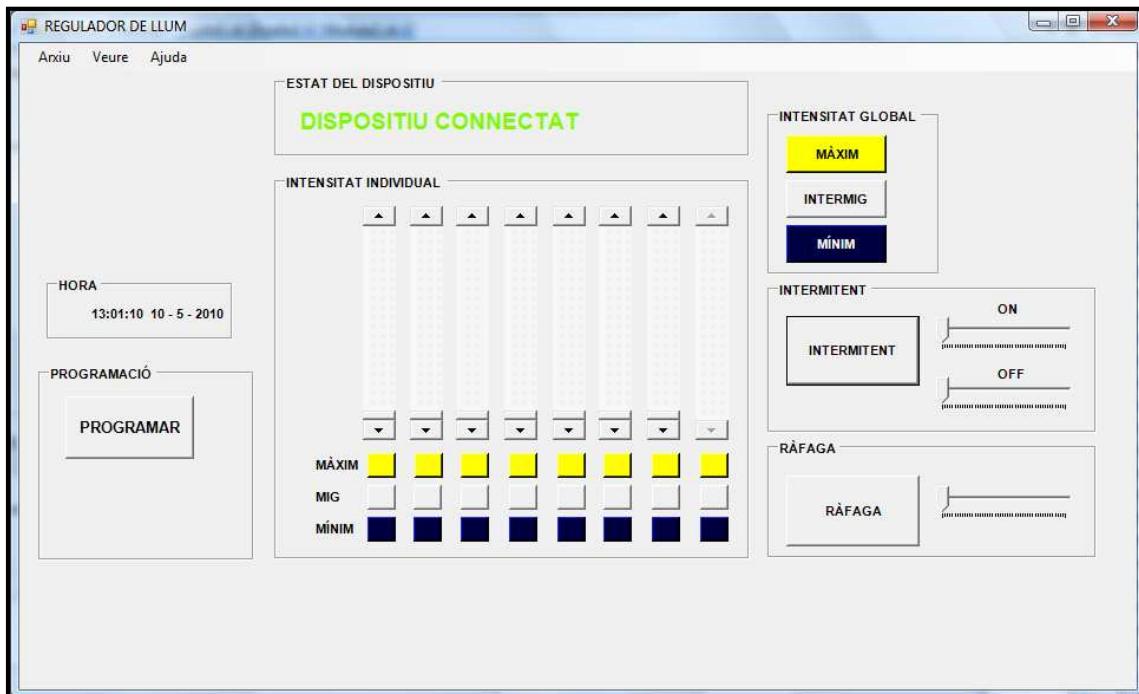


Fig. 7.3. Pantalla principal programa Regulador de llum

La funció principal d'aquest programa és enviar i rebre dades al circuit de control pel port USB. Aquestes dades es codifiquen en binari i permeten obtenir la regulació desitjada per a cada focus prèvia interpretació del circuit de control.

Al formulari de l'aplicació apareixen 8 barres equalitzadores, una per cada focus. El valor que s'envia és el valor numèric de la posició de cada una d'elles.

Quan l'equalitzador està a la posició mínima, el focus associat romandrà apagat, mentre que si es mou l'equalitzador fins al valor més elevat, el focus s'encendrà amb una intensitat màxima.

Cada una de les barres te els següents rangs de valors associats a enviar pel port USB:

Barra 1:	valor mínim: 1	valor màxim: 25
Barra 2:	valor mínim: 33	valor màxim: 57
Barra 3:	valor mínim: 64	valor màxim: 89
Barra 4:	valor mínim: 96	valor màxim: 120
Barra 5:	valor mínim: 128	valor màxim: 151
Barra 6:	valor mínim: 160	valor màxim: 183
Barra 7:	valor mínim: 192	valor màxim: 215
Barra 8:	valor mínim: 224	valor màxim: 242

El valor mínim correspon a la intensitat màxima dels focus mentre que un valor màxim correspondrà als focus apagats.

Les barres porten associats uns quadres de text en els que es pot observar el valor que assoleixen cada una d'elles. Cada vegada que es mou la barra, el valor del quadre de text variarà en funció del valor d'aquesta i s'envia pel port USB una trama amb el nou valor codificat en binari. Com que els rangs associats a cada focus no se superposen, no cal que la trama d'informació indiqui explícitament a quin focus es refereix, sinó que serà el programa del microprocessador qui ho decideixi. Per exemple, si s'envia el valor 172, aquest valor correspon a deixar el focus 6 a mitja intensitat.

Amb la finalitat de detectar i controlar possibles errors de transmissió, el PIC s'ha programat de manera que cada vegada que rep un nou valor reenvia cap al PC la mateixa trama que ha rebut (trama d'eco). D'aquesta manera, l'aplicació del PC pot comprovar que el valor enviat i el valor rebut coincideixen. Els valors enviats i rebuts sempre han de ser iguals ja que, si no es així, vol dir que hi ha hagut un problema de comunicació i que, per tant, no hi haurà un correcte accionament de les càrregues.

El programa “*REGULADOR DE LLUM*” permet:

- **Detecció automàtica de dispositius**
- **Establir connexió amb diferents dispositius**
- **Control d'intensitat individual de cada focus**
- **Control d'intensitat global de focus**
- **Efecte d'intermitència**
- **Efecte de ràfega**
- **Programació**
- **Comprovacions**
- **Visualitzacions horàries**

Detecció automàtica de dispositius:

El software programat, permet detectar automàticament la connexió de dispositius a qualsevol port USB de l'ordinador. Quan s'engega el programa, si el dispositiu està connectat el detecta automàticament i estableix connexió; per contra si aquest no està connectat, el programa espera a que es connecti mitjançant un *timer* que sondeja cada 3 segons si s'ha connectat un dispositiu. Una vegada detecta el dispositiu, es para el *timer* i el programa es manté en espera.

Quan el programa està en marxa i es desconnecta el dispositiu, es para automàticament la connexió i quan es torna a connectar, s'estableix altre vegada la connexió.

Quan s'ha establert connexió amb el dispositiu s'indicarà amb el text “*DISPOSITIU CONNECTAT*” en color verd; per contra, si es desconnecta, s'indica amb el text “*DISPOSITIU DESCONNECTAT*” en color vermell.

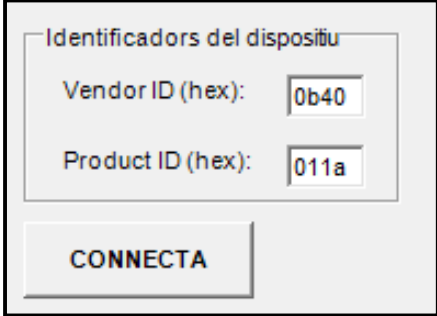


Fig. 7.4. Estat del dispositiu

Quan el dispositiu està desconnectat s'inhabiliten totes les funcions del programa fins que no es connecti un dispositiu.

Establir connexió amb diferents dispositius:

Cada dispositiu a connectar al port USB té el seu "Product ID" i el seu "Vendor ID". El programa permet canviar aquests valors de manera que es podrà seleccionar el dispositiu a connectar. Aquesta opció es pot trobar al menú "Veure-Identificadors dels dispositius". Al seleccionar-la, apareix la següent opció a la pantalla principal del programa:



The image shows a software dialog box titled "Identificadors del dispositiu". It contains two text input fields. The first field is labeled "Vendor ID (hex):" and contains the value "0b40". The second field is labeled "Product ID (hex):" and contains the value "011a". Below these fields is a button labeled "CONNECTA".

Fig. 7.5. Identificadors del dispositiu

Introduint els valors als quadres de text i polsant el botó "CONNECTA", es podrà establir comunicació amb el dispositiu connectat.

Control d'intensitat individual de cada focus:

Com s'ha esmentat anteriorment, es pot controlar la intensitat individual de cada focus mitjançant les barres de la figura, però també s'ha programat un sistema que permet establir la intensitat màxima, mínima i mitjana de cada focus individualment al polsar els diferents botons que es troben a sota de cada barra. Aquests botons envien pel port USB el valor corresponent per a que el focus romangui a l'estat seleccionat.

Polsant els botons màxims d'intensitat s'envia pel port USB el valor mínim de la barra seleccionada mentre que si es polsen els botons mínims, s'envia el valor màxim. Pel que fa als botons mitjos, al polsar-los s'envia el valor de la intensitat mitja de cada focus.

Al polsar els botons l'estat de la barra canvia visualment a la posició seleccionada.

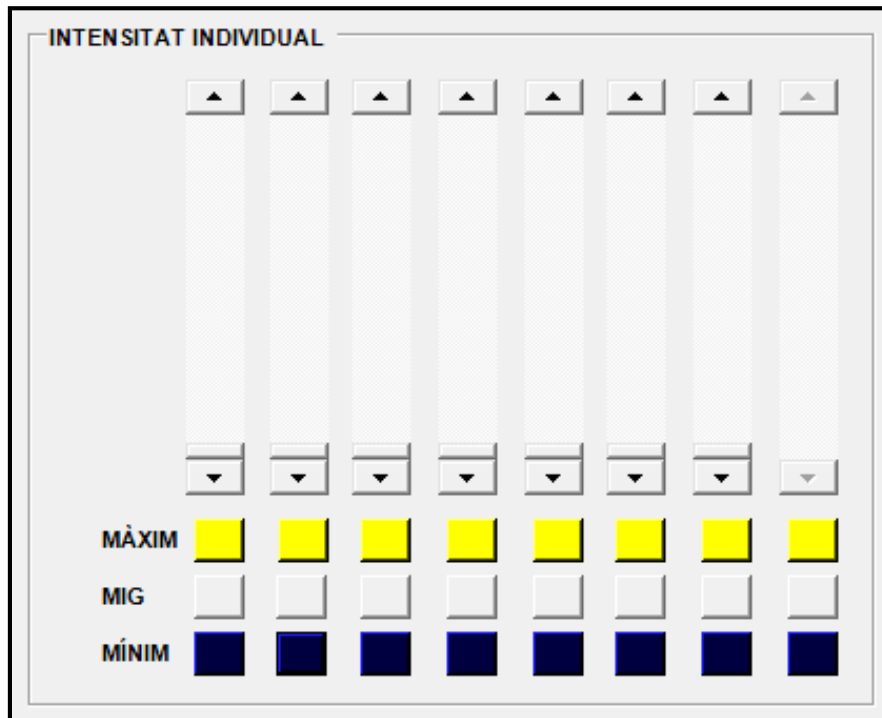


Fig. 7.6. Intensitat individual

Control d'intensitat global de focus:

Aquesta funció permet establir la intensitat màxima, mínima i mitjana dels focus com a conjunt. Al pulsar els botons, tots els focus passaran a l'estat desitjat a la vegada. Quan es polsen aquests botons s'envien 8 comandes pel port USB, una per a cada focus.

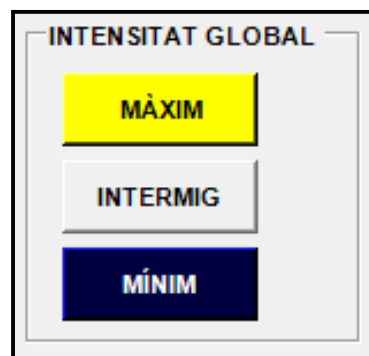


Fig. 7.7. Intensitat global

Efecte d'intermitència:

Permet realitzar intermitències al nivell d'intensitat desitjat, prèvia selecció a les barres, als botons d'intensitat individual o als botons d'intensitat global. També permet establir el temps d'intermitència mitjançant les dues barres horitzontals associades. La

barra “ON” permet seleccionar el temps que els focus estaran encesos, mentre que la barra “OFF” ho farà amb el temps d’apagat.

Quan es polsa el botó “INTERMITENT”, s’inicia la intermitència i el text canvia a “PARAR INTERMITENT”. Al tornar-lo a polsar es para la intermitència, i el text torna a canviar. Amb un sol botó es pot activar i desactivar la intermitència dels focus.

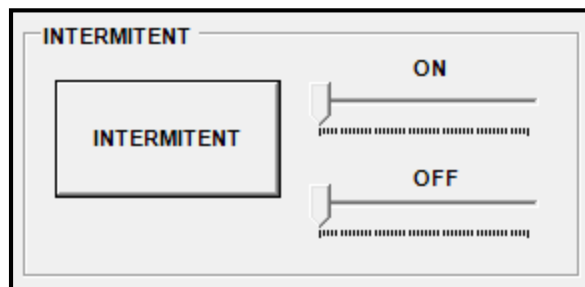


Fig. 7.8. Intermitent

Efecte de ràfega:

Al polsar el botó efectua un efecte de ràfega als focus a la intensitat desitjada seleccionada prèviament amb els botons d’intensitat individual o d’intensitat global. La ràfega encén el primer focus, l’apaga, encén el segon, l’apaga, i així successivament fins a tornar a començar per el primer. Al igual que amb la intermitència, es pot seleccionar la velocitat de la ràfega mitjançant la barra horitzontal i s’activa i es para amb un sol botó.

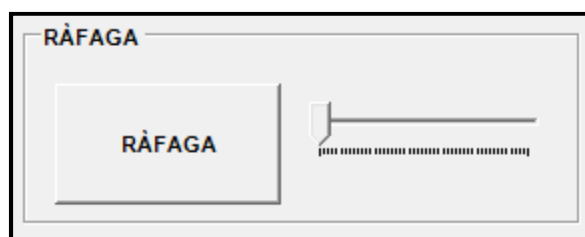


Fig. 7.9. Ràfega

Programació:

Un altre factor important a tenir en compte a l’hora de realitzar el projecte ha estat incloure un sistema de programació, es a dir, fer que el regulador de llum pugui programar els temps d’activació/desactivació de cada focus.

A la pantalla principal del programa s’ha creat un botó de comandament “PROGRAMAR” que condueix a una segona pantalla.

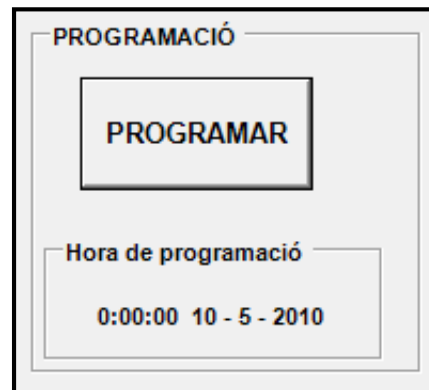


Fig. 7.10. Programació

En aquesta segona pantalla es pot trobar un calendari que permet escollir el dia desitjat de programació, uns desplegable per a escollir l'hora, el minut i el segon en els que es vol regular la intensitat lumínica escollida prèviament en els 8 equalitzadors o bé en els botons d'intensitat individual d'aquesta pantalla.

En definitiva, l'usuari del producte pot programar el regulador de llum per a encendre, apagar i regular els focus, el dia, l'hora, el minut i el segon que desitgi sense haver de fer-ho manualment. Per dur a terme aquesta programació es sincronitza un rellotge de programació amb el rellotge de Windows i la data escollida amb la data actual de Windows. Quan les dates coincideixen, llavors s'envia pel port USB la comanda de programació.

Per a comunicar les dues pantalles del programa, es fa servir un mòdul de programació en el qual s'han definit les variables a utilitzar pels dos formularis.

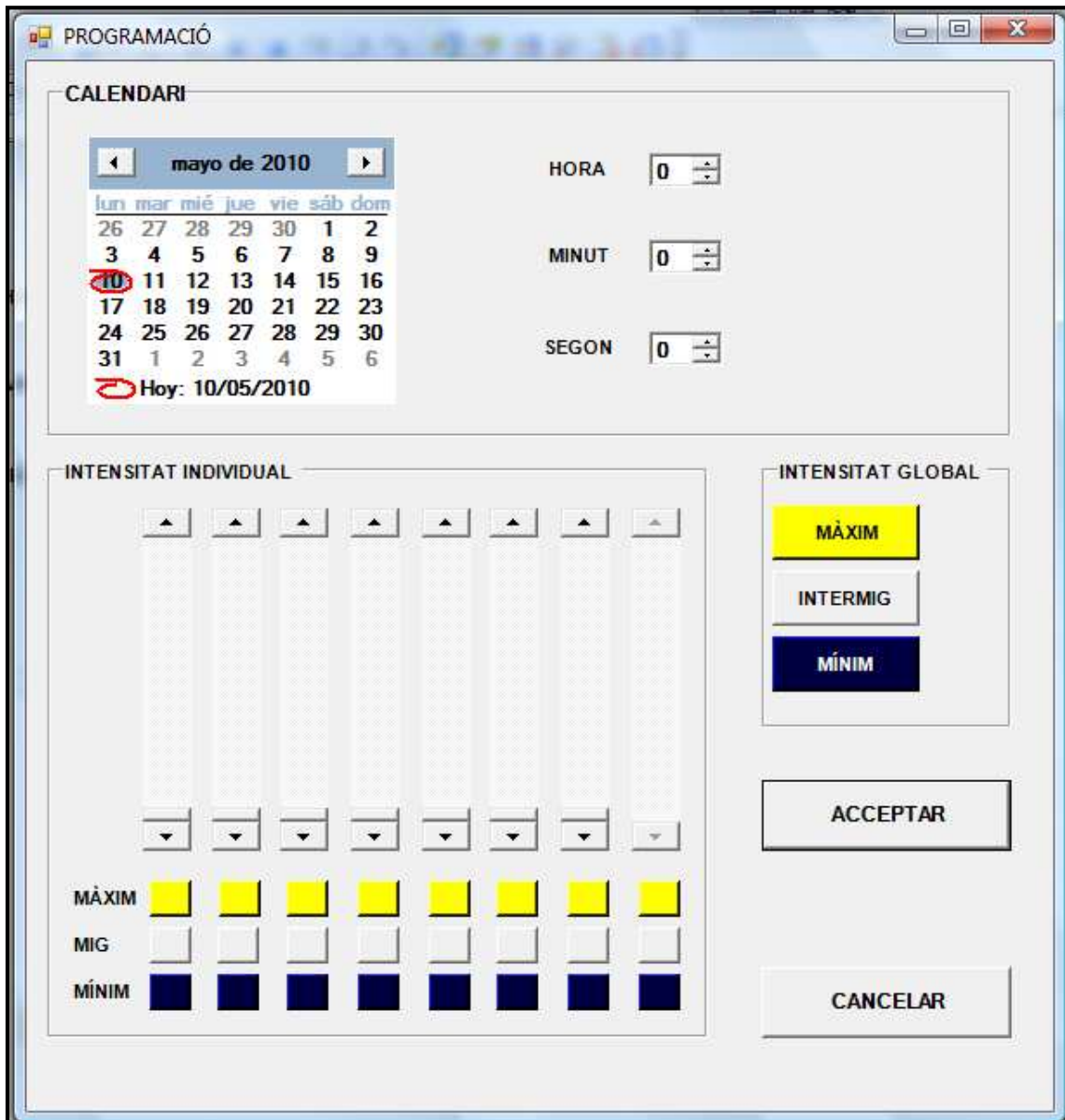


Fig. 7.11. Pantalla de programació

Comprovacions:

Al menú “Veure-Comprovacions” hi ha l’opció de veure els valors numèrics de les barres i l’opció de veure una comprovació de resultats que permet veure el Product ID, el Vendor ID del dispositiu connectat i el seu estat.

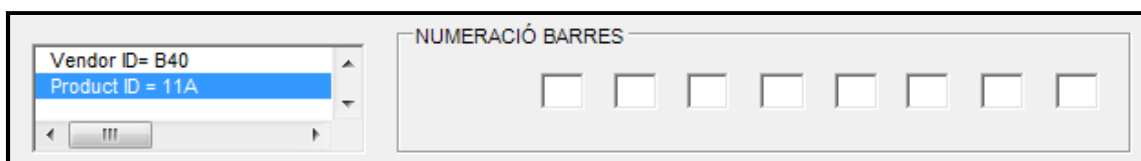


Fig. 7.12. Comprovacions

Visualitzacions horàries:

Al menú “*Veure-Hora*” es permet escollir l’opció de veure l’hora actual del sistema i veure l’hora de programació escollida.

7.2.2. ALIMENTACIÓ DEL CIRCUIT

Per a aportar alimentació al circuit, es parteix de la tensió provinent de la xarxa elèctrica. Les tensions necessàries per alimentar la part de control del circuit són de 5 V_{CC} i 3.3 V_{CC}, per tant ha estat necessari utilitzar una font d’alimentació que transformi el senyal altern de la xarxa en un senyal de contínua adequat per al circuit.

Aquesta font d’alimentació està constituïda per diversos blocs. A continuació es mostra el seu esquema:

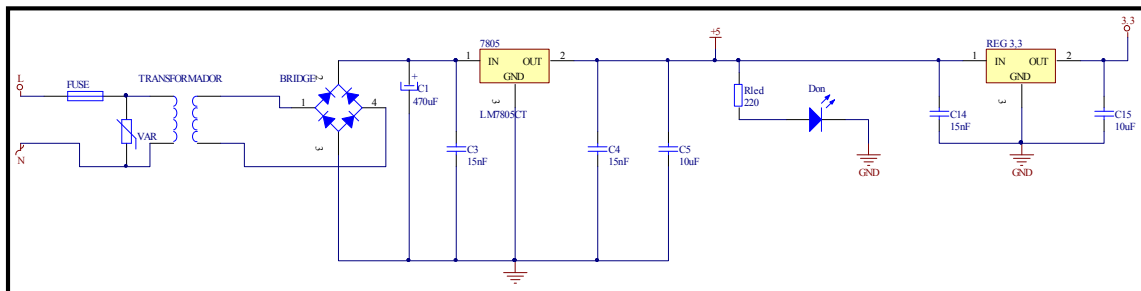


Fig. 7.13. Esquema alimentació del circuit

La protecció elèctrica en aquesta part del circuit és essencial, i s'utilitza un fusible de 3,15 A. Aquest element té com a finalitat resguardar la integritat de la resta de components en cas d'una sobrecàrrega.

Per a protegir el circuit de variacions i pics bruscos de tensió, es col·loca en paral·lel al circuit un varistor de 230 VAC que absorbeix els pics de tensió majors a la seva tensió nominal disminuint la resistència òhmica quan la tensió elèctrica augmenta.

L'últim bloc de la font d'alimentació el componen els reguladors de tensió, que s'encarreguen d'estabilitzar la tensió de sortida de la font davant de canvis a la tensió alterna o a la càrrega. La tensió necessària per alimentar la part de control del circuit és de 5 V_{CC} i de 3,3 V_{CC}. Per aquest motiu s'han col·locat dos reguladors de voltatge lineal en sèrie, un *MC78M05* que dona una tensió de 5 V_{CC} amb un corrent de sortida de 1.5 A i un *LM2937* que a la seva sortida ofereix 3,3 V_{CC}.



Fig. 7.14. MC78M05

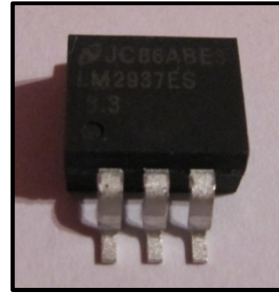


Fig. 7.15. LM2937

A la sortida del *MC78M05* s'ha connectat un díode led que indica si l'aparell està o no en funcionament.

7.2.3. CIRCUIT DE PAS PER 0

En tot sistema de regulació de corrent altern es fa necessari conèixer el moment en el que el senyal altern a regular passa pel valor de 0 VAC. És per això que es precisa d'un circuit de "detecció de pas per 0" que generi un senyal pilot cada vegada que es detecti aquest fenomen.

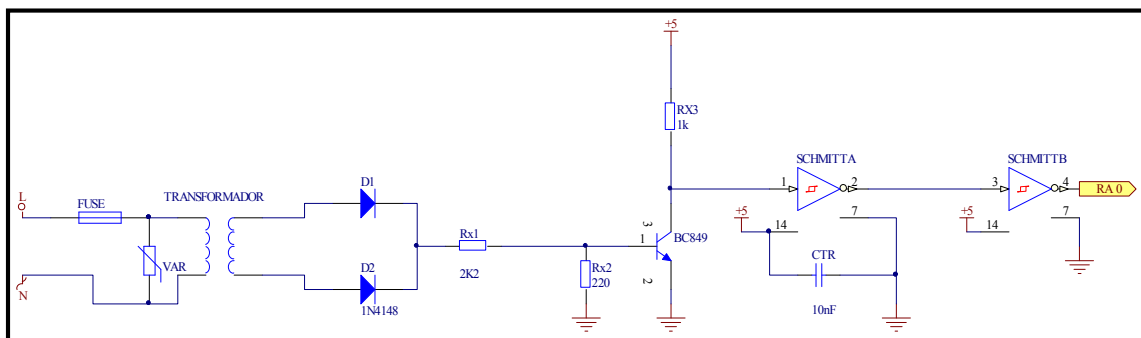


Fig. 7.16. Esquema circuit de detecció de pas per 0

Un sistema rectificador format per dos díodes rep la tensió alterna sinusoidal provinent del secundari del transformador. Els càtodes dels díodes estan connectats entre si i es en aquest punt on s'obté un senyal de polsos rectificada de 100 Hz (per a una freqüència de xarxa de 50 Hz). Aquest senyal s'aplica a la base d'un transistor, en aquest cas un *BC849C*. Cada vegada que el senyal sinusoidal val 0 V el transistor entra en tall i la seva tensió de col·lector, en una ràpida transició, puja a V_{cc} generant així un pols. Quan el senyal de polsos d'entrada té un valor diferent de 0 V, el transistor es manté en saturació i la tensió de col·lector és 0.

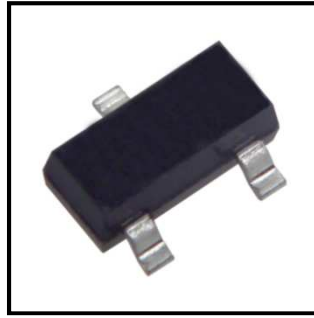


Fig. 7.17. BC849

Al col·lector del transistor se li ha connectat un *trigger d'Schmitt CD74AC14M*, la funció del qual es evitar que la tensió de soroll pugui arribar a provocar una senyal de pas per 0. El *trigger* inverteix el senyal de pas per 0, així doncs, cal utilitzar-ne dos per a invertir dues vegades el senyal i obtenir el senyal de polsos correcte.

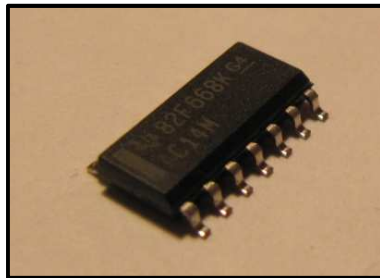


Fig. 7.18. CD74AC14M

En resum, mitjançant un circuit de detecció de pas per 0 s'obté un pols lògic positiu cada vegada que el senyal altern té el valor de 0 V, tal i com queda reflectit a l'oscil·lograma.

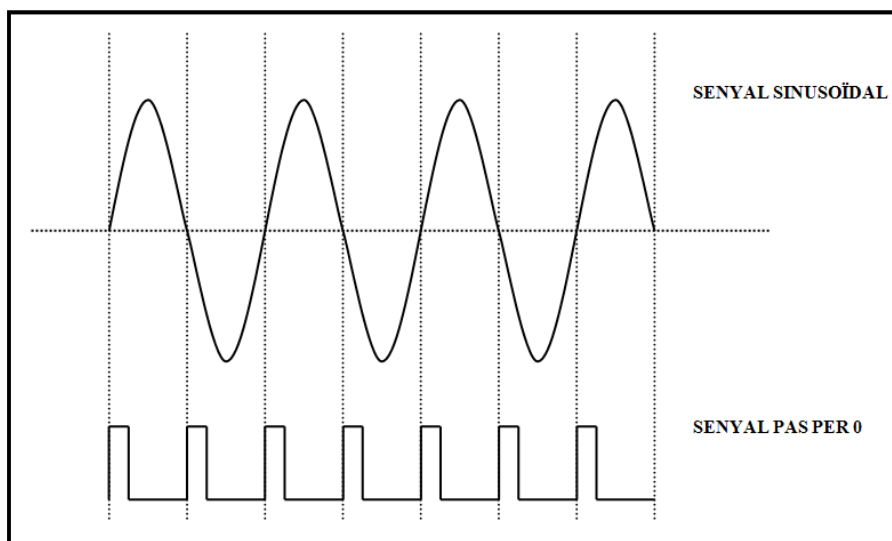


Fig. 7.19. Oscil·lograma detecció pas per 0

7.2.4. CIRCUIT DE CONTROL

Per gestionar els canvis de la intensitat dels focus que s'ordenen des del PC, s'ha implementat el següent circuit de control.

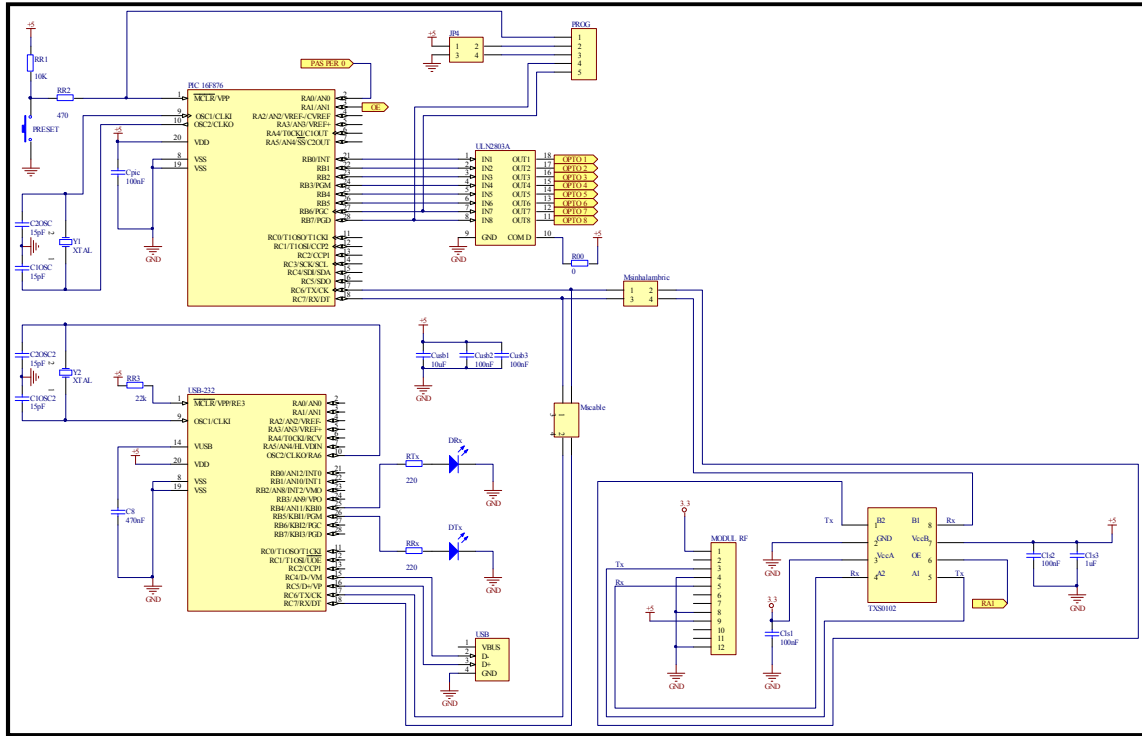


Fig. 7.20. Esquema circuit de control

Aquest circuit està dissenyat principalment per a establir una comunicació per cable, però també dona l'opció a connectar mòduls de radiofreqüència.

Per tal de seleccionar una comunicació o una altra, s'utilitzen dos microswitch que es troben al circuit electrònic.

COMUNICACIÓ	MS CABLE	MS RADIOFREQUÈNCIA
AMB CABLE	ON	OFF
RADIOFREQUÈNCIA	OFF	ON

Taula 7.1. Selecció microswitch per a comunicació

Quan s'utilitza la comunicació per cable es troben els següents elements:

- **Connector USB**
- **USB-232**
- **PIC 16F876**
- **ULN 2803**
- **Circuit de potència**

El connector USB de tipus B és un connector estàndard que permet connectar un cable procedent del Port USB 2.0 de l'ordinador amb el circuit electrònic.



Fig. 7.21. Connector USB tipus B

El port USB 2.0 consta de 4 línies:

PIN	NOM	DESCRIPCIÓ	COLOR DEL CABLE
1	VCC	+ 5VDC	VERMELL
2	D-	DADA -	BLANC
3	D+	DADA +	VERD
4	GND	MASSA	NEGRE

Taula 7.2. Connexió USB

Una vegada han arribat les dades al circuit, s'utilitza un convertidor de dades USB, concretament l'*USB-232* de la casa Hexwax, un circuit integrat que transforma dades sèrie USB a dades sèrie UART (*Universal Asynchronous Receiver-Transmitter*) asíncrones. Aquest xip simplifica la connexió de dispositius electrònics a l'ordinador ja que utilitza l'HID (Dispositiu d'Interfície Humana). És un dispositiu "plug & play" cosa que permet connectar-lo a qualsevol port USB fent un ús immediat sense necessitat d'instalar *drivers* i sense crear un port virtual.

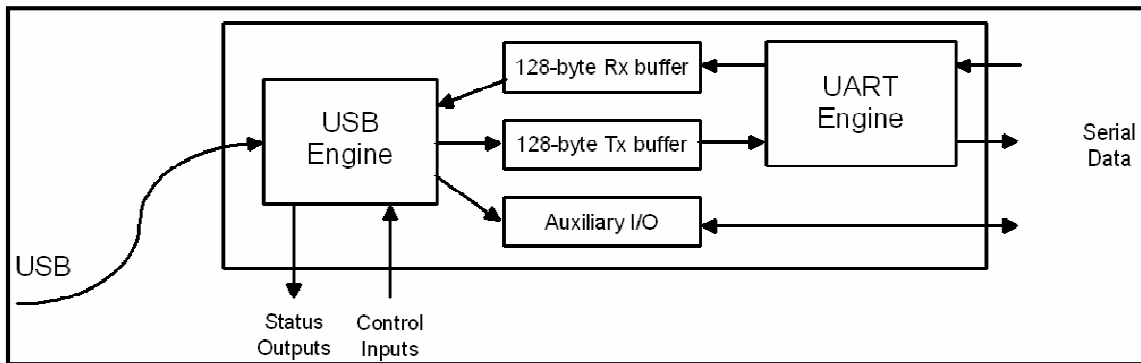


Fig. 7.22. Composició USB-232

L'USB-232 és un *firmware* aplicat al PIC 18LF2450 i al PIC 18F14K50 disponible en encapsulat DIL de 28 potes i SSOP de 20 potes respectivament.

Reset#	1	28	PGD	Vdd	1	20	Vss
n.c.	2	27	PGC	OSC1	2	19	D+/PGD
n.c.	3	26	RxInd	OSC2	3	18	D-/PGC
n.c.	4	25	TxInd	Reset#	4	17	Vusb
DCD#	5	24	Suspend#	DCD#	5	16	Send
DSR#	6	23	DTR#	DSR#	6	15	CTS#
RI#	7	22	CTS#	RI#	7	14	BufferEmpty
Vss	8	21	Send	DTR#	8	13	AllSysGo#
OSC1	9	20	Vdd	Suspend#	9	12	RxD
OSC2	10	19	Vss	TxD	10	11	RTS#
RTS#	11	18	RxD				
AllSysGo#	12	17	TxD				
BufferEmpty	13	16	D+				
Vusb	14	15	D-				

USB-232 pinout
Pins marked n.c. should be grounded

Fig. 7.23. Diagrama de pins USB-232

En aquest projecte s'utilitza l'integrat de 28 pins que va connectat directament al port USB pels pins d'entrada de comunicació D+ (pin 16) i D- (pin 15). Les dades sèrie UART asíncrones es poden trobar als pins de sortida TxD (pin 17) i RxD (pin 18).

Per a establir una correcta comunicació, s'ha configurat el dispositiu de la següent manera, utilitzant el software que proporciona el fabricant:

- Velocitat de transmissió: 9600 bps
- Format de la trama: 8 bits de dades, 1 bit de stop i sense ús de paritat.
- Control de flux: no s'utilitza control de flux.

El software realitza un sondeig per a buscar el dispositiu connectat al port USB de l'ordinador. Quan l'usuari l'obre s'habiliten totes les opcions que incorpora (canviar el nom del producte, el codi de fabricant, el codi de producte, la intensitat màxima del bus i les funcions de cada pin...) També dona l'opció d'enviar caràcters pel port USB en format hexadecimal, binari, ASCII, etc.

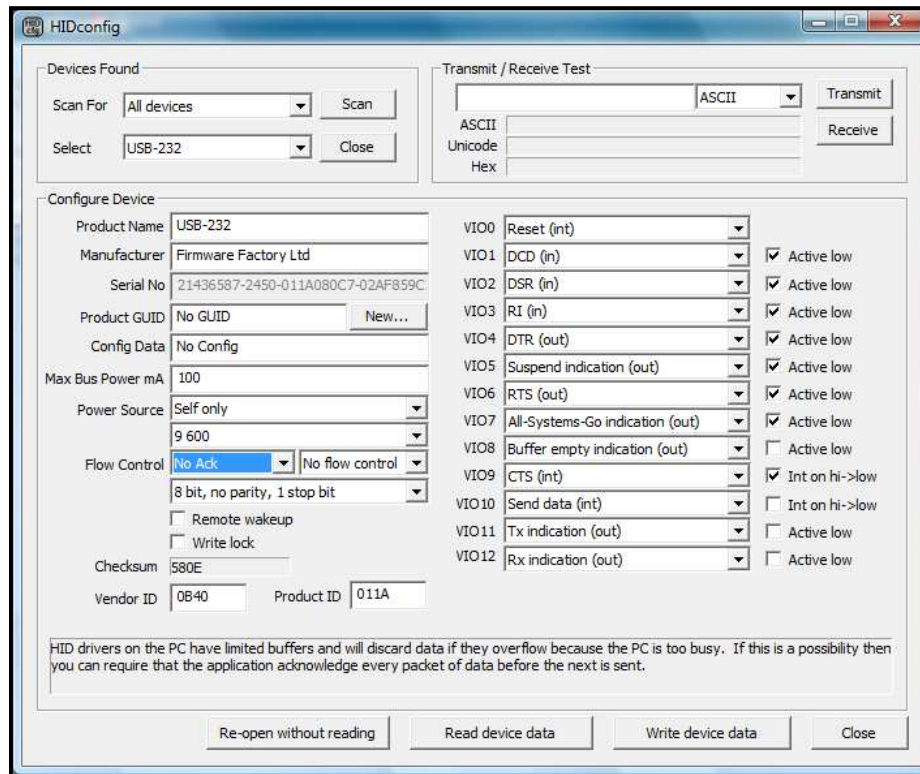


Fig. 7.24. Software de configuració USB-232

L'USB-232 s'encarrega de rebre les dades procedents de l'ordinador, transformar-les i enviar-les al PIC perquè aquest les pugui interpretar. De la mateixa manera també rep dades del PIC, les transforma i les envia a l'ordinador per a comprovar que la dada enviada per l'ordinador ha estat correcta. Les dades que rep són en format byte enviades des del programa realitzat amb Visual Basic i corresponen al valor de les barres equalitzadores.

Amb la finalitat que aquest circuit interpreti correctament les dades rebudes, prèviament ha de rebre un identificador de la quantitat de dades que se li enviaran.

IDENTIFICADOR	DADES REBUDES	SENTÈNCIA A ENVIAR
0x01	1	0x01+dada1
0x02	2	0x02+dada1+dada2
0x03	3	0x03+dada1+dada2+dada3
0x04	4	0x04+dada1+dada2+dada3+dada4
0x05	5	0x05+dada1+dada2+dada3+dada4+dada5
0x0n	N	0x0n+dada1+dada2+dada3+dada4+dada5+...+dadan

Taula 7.3. Sentències a enviar al USB-232

Cada vegada que es mou una barra equalitzadora el programa envia un byte, així doncs aquest integrat rep la sentència $0x01+dada1$.

Per a observar la transmissió i la recepció de dades s'han connectat dos leds, un al pin RxInd (pin 26) i l'altre al pin TxInd (pin 25). D'aquesta manera quan s'envii una dada des de l'ordinador al USB-232 s'ilumina el led connectat a RxInd i quan l'USB-232 envia una dada al PC s'ilumina el led connectat a TxInd.

Aquest circuit s'alimenta a 5 VDC i té una freqüència de rellotge de 12MHz. El condensador C8 de 470 nF és un filtre per a un regulador intern. Els condensadors Cusb1 i Cusb2, han d'estar a prop del connector USB i Cusb3 ha d'estar a prop dels pins +5 i GND de l'integrat.



Fig. 7.25. USB-232-DIL

Per a utilitzar la comunicació sense cable es troben els següents elements:

- **Receptor de radiofreqüència**
- **TXS0102**
- **PIC 16F876**
- **ULN 2803**
- **Circuit de potència**

Els mòduls de radiofreqüència que es pretenen connectar són els mòduls **Iknos[®]** que transmeten informació de forma bidireccional sense cablejat i tenen les següents característiques:

- Potència de transmissió: 5dB
- Banda de freqüència: 2,4 GHz
- Sensibilitat: -95dBm
- Guany antenes: 2,7dBi
- Abast: 800m en espai obert

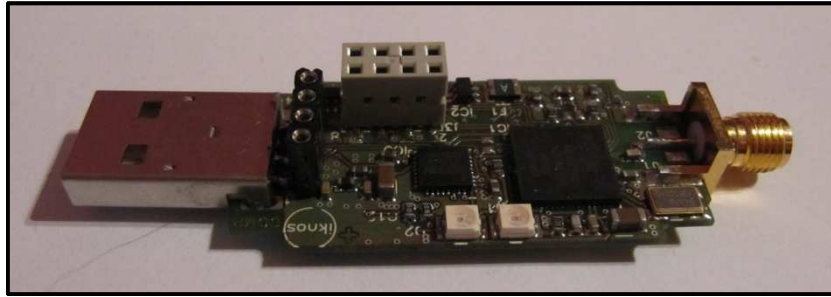


Fig. 7.26. Receptor de radiofreqüència IKNOS

El *TXS0102* és un traductor de voltatge bidireccional de dos bits que s'utilitza per a establir la compatibilitat entre dos sistemes de comunicació digital mixta de tensió. L'integrat utilitza dues fonts de tensió per separat que fixen els voltatges desitjats.

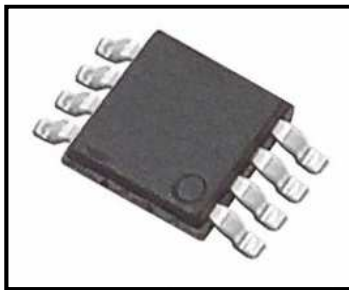


Fig.7.27. *TXS0102DCUR*

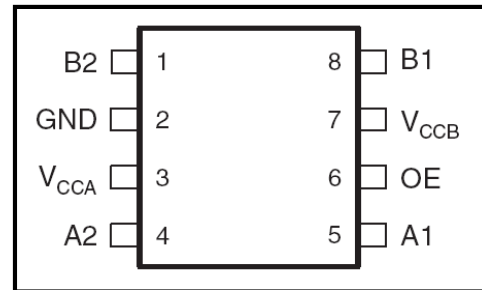


Fig. 7.28. Diagrama de pins *TXS0102DCUR*

El mòdul de radiofreqüència i el PIC no treballen a la mateixa tensió a nivell de dades, el mòdul treballa a una tensió de 3.3 VDC i el *PIC* a una de 5 VDC, es per això que cal convertir aquests valors de tensió perquè els dos dispositius puguin treballar correctament.

Els voltatges fixats al *TXS0102* són 3.3 VDC per al canal A introduïts a l'integrat pel pin *V_{CCA}* i 5 VDC per al canal B introduïts pel pin *V_{CCB}*.

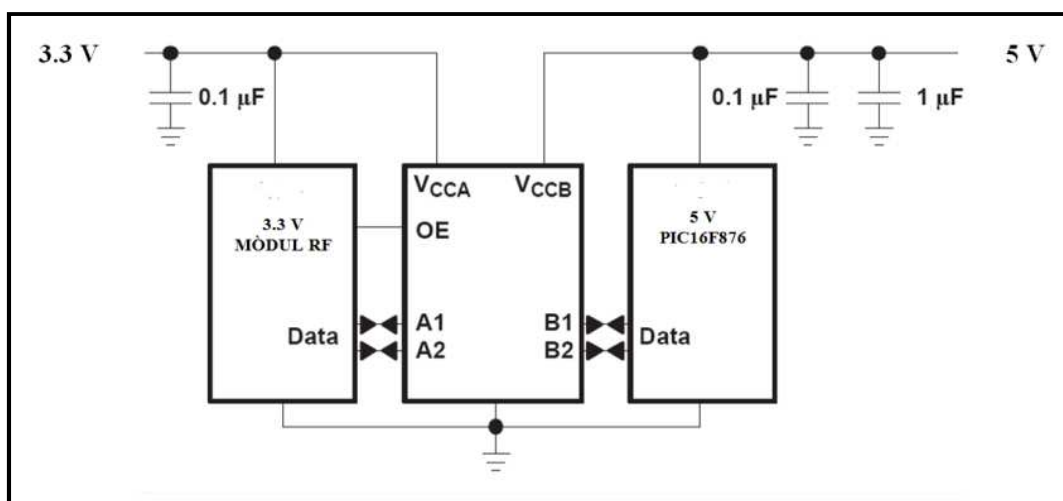


Fig. 7.29. Diagrama de connexió *TXS0102DCUR*

Les dades procedents del mòdul de radiofreqüència de 3.3 VDC entren per la pota A1 del *TXS0102*, a la sortida B1 d'aquest integrat s'obté la mateixa dada però amb una tensió de 5VDC apte per a introduir al PIC. D'altra banda les dades que s'envien del PIC al *TXS0102* entren per la pota B2 fixada a una tensió de 5 VDC i surten per la pota A2 a una tensió de 3.3 VDC.

L'encarregat de gestionar les dades rebudes i enviades de la part de control d'aquest projecte és el *PIC 16F876*. Aquest PIC és un microcontrolador de gamma mitja comprès a la subfamília dels PIC 16F87X, que es caracteritza per tenir una memòria de programa de tipus FLASH i una sèrie de recursos semblants als models més potents, com poden ser els PIC 16C73/4, tenint aquests últims l'inconvenient de que la seva memòria és de tipus EPROM.

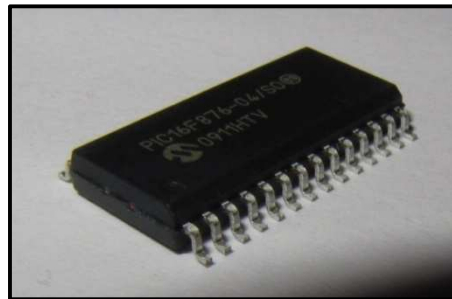


Fig. 7.30. PIC16F876-04/SO

El *PIC 16F876* es caracteritza per la independència entre la memòria de codi i la de dades. Així, tant la capacitat com la mida dels busos de cada memòria s'adapten estrictament a les necessitats de disseny, facilitant el treball en paral·lel de les dues memòries, cosa que permet obtenir altes cotes de rendiment.

Utilitza una arquitectura RISC, cosa que es fa patent en el reduït número d'instruccions que formen el seu repertori. Només consta de 35 instruccions, que s'executen en un cicle d'instrucció, equivalent a quatre períodes de rellotge, excepte les de salt, que necessiten dos cicles.

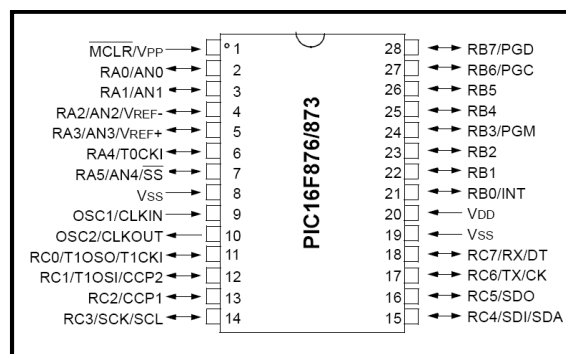


Fig. 7.31. Diagrama de pins PIC16F876-04/SO

Els PIC 16F876 tenen un mòdul MSSP amb dues portes per a la comunicació sèrie síncrona, o sigui, amb senyal de rellotge. A més a més, també disposen d'un mòdul USART capaç de suportar la comunicació sèrie síncrona i asíncrona.

L'USART, anomenat SCI (Serial Communications Interface), pot funcionar com un sistema de comunicació *full duplex* o bidireccional asíncron, adaptant-se a la multitud de perifèrics i dispositius que transfereixen informació d'aquesta forma, tals com el PC. També pot treballar en mode síncron unidireccional o *half duplex* per a suportar perifèrics com memòries, convertidors, etc...

En resum, l'USART pot treballar de tres maneres:

- **ASÍNCRONA** (full duplex, bidireccional)
- **SÍNCRONA-MESTRE** (half duplex-unidireccional)
- **SÍNCRONA-ESCLAU** (half duplex-unidireccional)

A la següent figura es mostra un esquema del comportament de l'USART en mode asíncron, que és el mode de comunicació que es fa servir en aquest projecte.

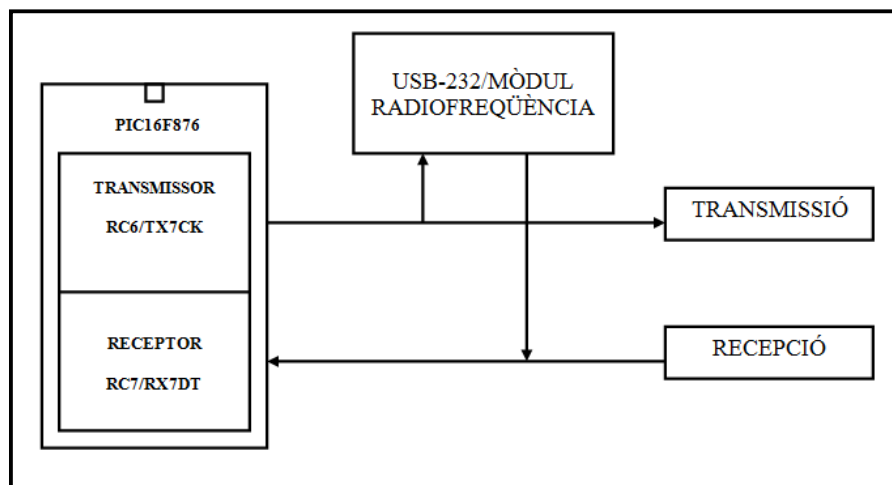


Fig. 7.32. Comunicació de l'USART en mode síncron

Les transferències d'informació es realitzen sobre dues línies *TX* (transmissió) i *RX* (recepció). La freqüència de transferència, controlada internament per l'USART, és de 9600 bps.

Les línies de comunicació són les dues de més pes del PORTC: *RC6/TX/CK* i *RC7/RX/DT*.

En aquesta forma de comunicació sèrie s'usa el protocol RS-232. Les trames consten de 8 ó 9 bits (en funció de si s'utilitza o no paritat) i van precedides per un bit d'START (inici) i al darrere de la trama a transmetre es col·loca un bit d'STOP (parada), d'acord amb les normes de format estàndard NRZ (NonReturn-to-Zero). Els bits es transfereixen a una velocitat fixa i normalitzada.

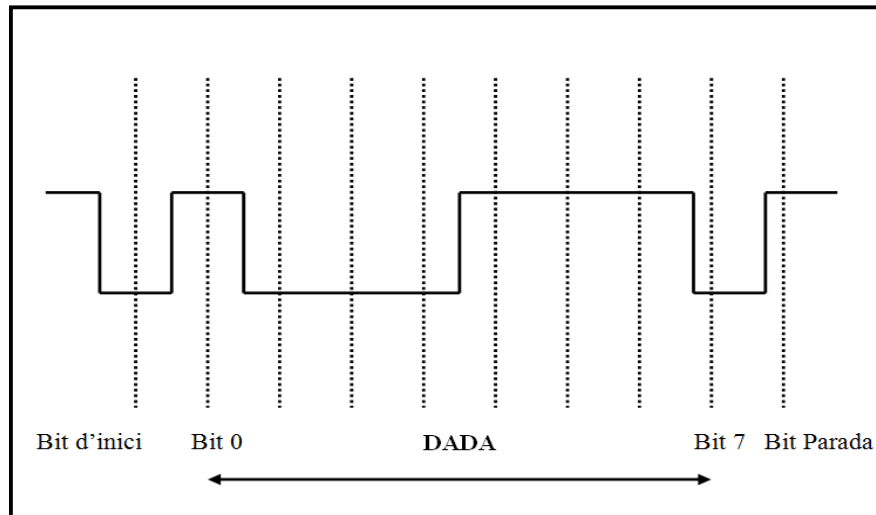


Fig. 7.33. Trama protocol RS-232

Els quatre blocs que configuren l'arquitectura de l'USART en mode asíncron son:

- **Circuit de mostreig**
- **Generador de baud-rate**
- **Receptor asíncron**
- **Transmissor asíncron**

Tot seguit es comenten aquests quatre blocs per tal de conèixer aquest tipus de comunicació amb més detall.

El circuit de mostreig actua sobre el pin *RC7/RX/DT*, que és per on es rep el bit d'informació o control, i s'encarrega de mostrejar tres vegades el seu valor, per a decidir aquest per majoria.

En el protocol asíncron RS-232, la velocitat de transferència en bauds i en bps (bits per segon) a la que es realitza la transferència és la mateixa, i s'ha d'efectuar a un valor normalitzat: 330, 660, 1200, 2400, 4800, 9600, 38400, etc. Per a generar aquesta freqüència, l'USART disposa d'un Generador de Baud Rate, *BRG*, el valor del qual es controla pel contingut gravat al registre *SPBRG*.

A més del valor X carregat al registre $SPBRG$, el valor del baud rate del generador depèn també del bit $BRGH$ del registre $TXSTA$ <2>. En el cas de que $BRGH$ sigui 0 es treballa a baixa velocitat i si $BRGH=1$ es treballa a alta velocitat. Segons aquest bit s'obté el valor d'una constant K necessària en la determinació de la freqüència de funcionament.

$$\text{Freqüència (baud)} = \frac{F_{osc}}{K * (X + 1)}$$

A on X és el valor carregat al registre $SPBRG$

- Si $BRG = 0$, baixa velocitat i $K= 64$
- Si $BRG = 1$, alta velocitat i $K= 16$

Les dades es reben en sèrie, bit a bit pel pin $RC7/RX/DT$ i es van introduint seqüencialment al registre de desplaçament RSR , que funciona a una freqüència 16 vegades més ràpida que la de treball. Quan $CREN = 1$ al registre $RCSTA$ <4>, s'habilita la recepció.

Els passos a seguir en el mode de recepció són els següents:

- Es carrega amb el valor X el registre $SPBRG$ per a treballar amb la freqüència desitjada, controlant el valor $BRGH$.
- S'habilita l'USART en mode asíncron, amb $SPEN=1$ i $SYNC=0$.
- Si es desitja que es generi una interrupció amb l'arribada del bit de parada, es posa el bit $RCIE=1$, a més d'habilitar les interrupcions en general.
- Per a detectar l'adreça, cal posar el bit $ADDEN=1$.
- Si es volen habilitar les interrupcions, cal posar el bit $CREN = 1$. Al completar la recepció $RCIF$ es posarà a 1 i es produirà una interrupció si s'havia permès.
- Es llegeix el registre $RCSTA$, on es pot consultar si s'ha produït algun error. Si és així, cal posar $CREN=0$.
- Llegir els 8 bits del registre $RCREG$ per a determinar si el dispositiu és el direccionat. Si el dispositiu ha estat direccionat, posar $ADDEN=0$ per permetre la recepció de la informació.

La dada que es vol transmetre per l' USART transmissor es diposita al registre $TXREG$ i a continuació es traspasa al registre de desplaçament TSR que va traient els bits seqüencialment pel pin $RC6/TX/CK$ a la velocitat establerta. A més a més, abans dels bits de dades d'informació s'inclou un bit d'inici i després de treure totes les dades s'afegeix un bit de parada. La transferència entre els dos registres es realitza en un cicle

i aleshores el bit de senyal *TXIF* es posa a 1, per a advertir que el registre de transmissió s'ha buidat. També en aquest moment es pot produir una interrupció a nivell de programació si s'ha possibilitat el bit *TXIE=1* al registre *PIE<4>*. Quan s'escriu una altra dada sobre *TXREG*, el bit de senyal *TXIF* es posa a 0. El bit *TRMT* serveix per a indicar l'estat del registre *TSR* i val 1 quan està buit.

La seqüència d' instruccions a seguir per a implementar una transferència a l'USART es la següent:

- Configurar les línies *RC6/TX/CK* com a sortida i *RC7/RX/DT* com a entrada.
- Posar *SYNC=0* i *SPEN=1* per a activar l' USART en mode asíncron.
- Si es desitja treballar amb interrupció, posar *TXIE =1*, a més a més d'habilitar les interrupcions en general.
- Si la dada consta de 9 bits, enlloc dels 8 típics, posar el bit *TX9= 1*. El novè bit es col·locarà en *TX9D (TXSTA)*.
- Es carrega el valor *X* adequat al registre *SPBRG*, per a produir la freqüència de treball desitjada. Controlar el bit *BRGH*.
- Activar la transmissió amb *TXEN=1*. El bit *TXIF* tindrà valor 1 ja que *TXREG* es troba buit.
- Carregar a *TXREG* la dada a transmetre i començar la transmissió.

El procés de funcionament d'aquesta part de control és el següent:

Al microcontrolador se li ha introduït un programa en codi ensamblador. El programa que s'ha realitzat fa que el PIC es comuniqui via sèrie amb un programa terminal que es trobarà en un PC. El PIC espera a rebre un caràcter del PC, fa les operacions corresponents segons el caràcter enviat i transmet aquest caràcter a l'aplicació del PC per verificar que el caràcter ha estat correctament rebut pel PIC.

La recepció del caràcter al PIC es controla mitjançant una interrupció, és a dir, cada vegada que des de l'aplicació de Visual Basic s'emeti un caràcter, aquest arriba al PIC i es genera una interrupció. El codi encarregat d'atendre la interrupció és el que realitza les instruccions corresponents segons la dada rebuda.

La dada procedent del PC és un byte que entra al PIC pel pin receptor (*RC7*). Aquest byte correspon al número escollit a la barra equalitzadora del programa en Visual Basic. Si per exemple el número escollit a la barra és el 45, al PIC li arriba el byte '00101101' que es correspon amb el valor 45 codificat en binari. Quan arriba aquest caràcter el PIC genera una interrupció i la RSI comprova quin ha estat el caràcter que ha arribat. Una vegada se sap quin valor ha generat la interrupció, el PIC tria el

focus a encendre segons el valor enviat pel PC, és a dir, si el valor 45 correspon a una certa intensitat de la barra número dos, doncs es dispararà el triac corresponent a la barra dos i s'encendrà el focus número dos amb la corresponent intensitat.

Per a generar els polsos de disparada del triac el PIC fa un tractament de retard a la senyal de pas per 0 acord amb la dada rebuda de l'ordinador, generant així els polsos necessaris per a encendre els diferents focus a la intensitat desitjada.

El pols de disparada del triac surt del PIC pels terminals del PORTB de RB0 a RB7 i exciten un opto-triac que després donarà la mateixa excitació al triac per així encendre els focus.

Degut a la petita intensitat que subministren els microcontroladors, s'han amplificat les seves sortides utilitzant l'integrat ULN2803. Aquest integrat és un conjunt de transistors darlington amb els que es pot controlar càrregues de fins a 500 mA i porta díodes de protecció contra les sobretensions produïdes per càrregues inductives. L'ULN2803 té les sortides invertides per tant hauran d'anar connectades al càtode del díode emissor.

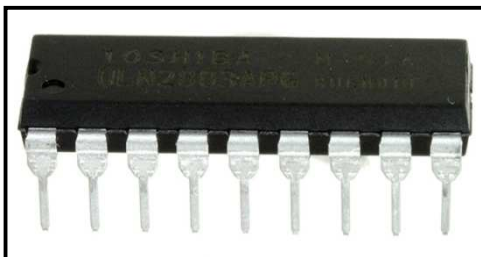


Fig. 7.34. ULN2803A

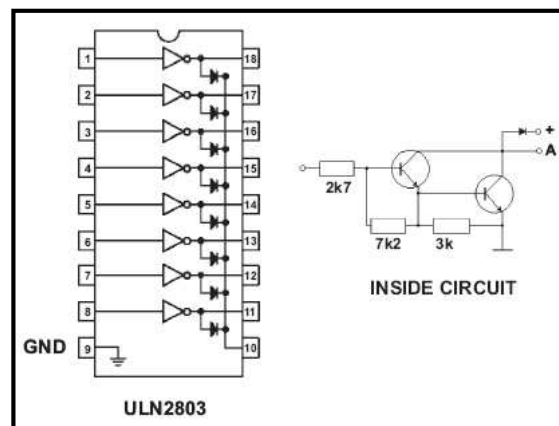


Fig. 7.35. Diagrama ULN2803A

Per tal d'ajustar la intensitat del focus, s'utilitza un senyal que prové de la xarxa prèviament tractat i configurat per tal de que realitzi un pols quadrat cada 100Hz.

Aquest senyal s'introdueix al PIC pel terminal RA0, configurat prèviament com a terminal d'entrada. El rang de la intensitat a graduar de cada focus està comprès entre els dos polsos de 100Hz i a mesura que es va incrementant o decrementant l'equalitzador, el pols de disparada del triac es desplaça dins d'aquests marges.

A la següent figura es pot observar de color negre els polsos que s'efectuen cada 100Hz i de color vermell el pols de disparada del triac que pot recórrer el marge de disparada marcat a la mateixa figura.

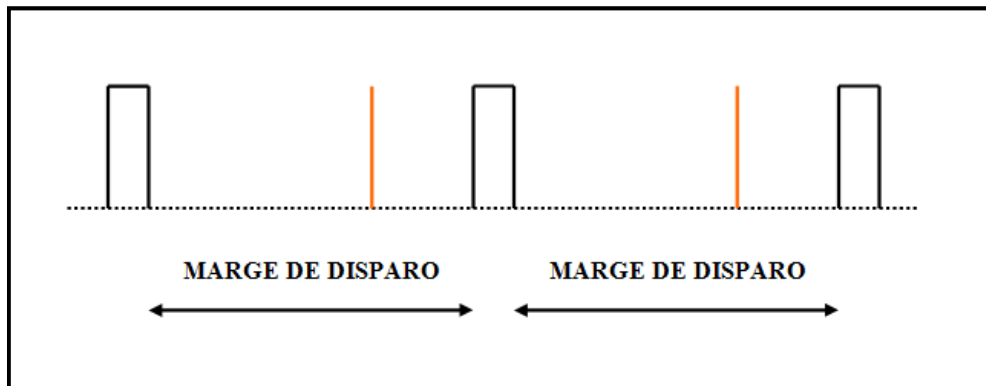


Fig. 7.36. Polsos i marge de disparo triac

L'alimentació d'aquest xip està compresa entre 2 i 5,5V, en aquest cas i aprofitant la tensió regulada a 5 volts, s'alimentarà a aquest voltatge.

El microcontrolador funciona amb un cristall de quars de 4MHz.

Per a programar el microcontrolador s'utilitza el compilador *MPLAB IDE* i el programador *PICKIT2* de la casa *Microchip*. El compilador tradueix les sentències de programació en llenguatge ensamblador al llenguatge màquina utilitzat pel PIC i el programador efectua les gestions necessàries per introduir les instruccions al PIC.



Fig. 7.37. Programador PICKIT 2

A l'hora de realitzar el projecte, s'ha dotat el circuit amb un complement bastant útil pel que fa a la programació del microcontrolador. Es tracta de poder programar el microcontrolador sense haver de retirar-lo del circuit de control. Per a aconseguir

aquesta finalitat, s'ha inserit una tira de pins al circuit que permet connectar el programador amb els pins de programació *PGD* (pin 28), *PGC* (pin 27) i \overline{MCLR} (pin 1).

Per a programar-lo, el *PIC* ha d'estar alimentat a 5VDC. Aquest voltatge es pot aconseguir de la mateixa targeta electrònica connectada a la xarxa elèctrica, o bé del mateix programador sense necessitat de tenir connectat el regulador de llum. Mitjançant el microswitch JP4 i el software de configuració del *PICKIT 2* es pot escollir una modalitat o l'altra.

Amb aquest programador inserit a la placa la manipulació manual del PIC és nul·la i innecessària, és a dir, no s'ha de treure el microcontrolador del circuit cada vegada que aquest es vol programar, sinó que es pot programar directament des del circuit imprès.

L'última característica a esmentar és el polsador de Reset que inclou aquesta part de control. Es tracta d'un polsador connectat al pin1 del *PIC* (\overline{MCLR}/VPP) mitjançant una resistència de pull-up. Cada vegada que es polsi aquest el programa gravat al *PIC* retornarà a la seva posició inicial.

Dissenyant, construint i programant aquesta part de control, s'obté una comprensió bastant exacta en termes electrònics del circuit. Aquest muntatge és l'essència bàsica per a controlar juntament amb el PC el projecte pròpiament dit.

7.2.5. CIRCUIT DE POTÈNCIA

La part de potència d'aquest projecte consta d'un circuit que mitjançant uns polsos amb el seu corresponent retard procedents del PIC i aplicats a uns opto-triacs, aconseguixen variar la il·luminació de 8 bombetes.

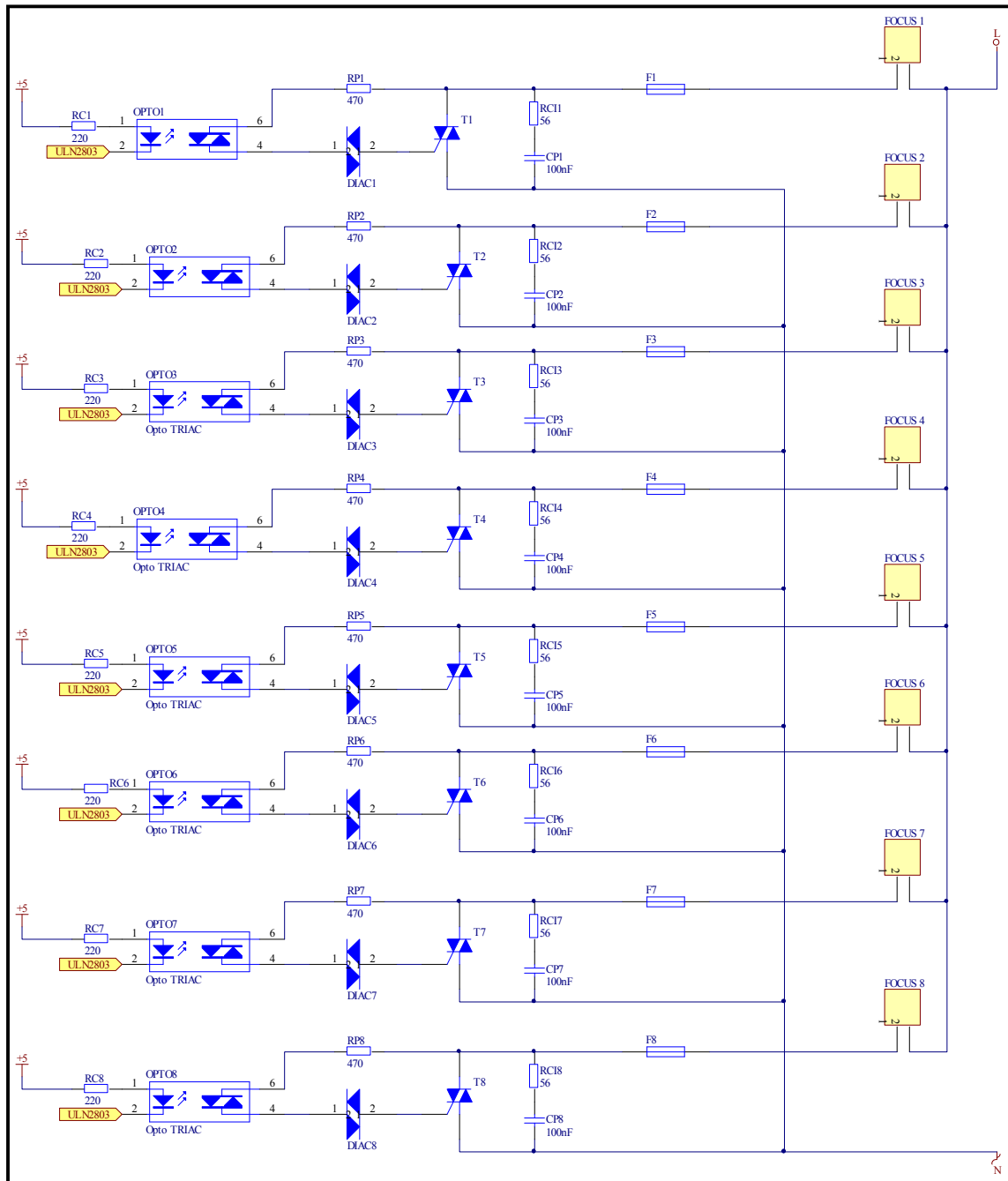


Fig. 7.38. Esquema circuit de potència

El primer element que es troba és un opto-triac, concretament el *MOC3021*. Els triacs acoblats òpticament combinen un díode emissor de llum (LED) amb un triac foto

detector (foto-triac o opto-triac) dins d'un mateix encapsulat tal i com es mostra a la figura.



Fig. 7.39. MOC3021

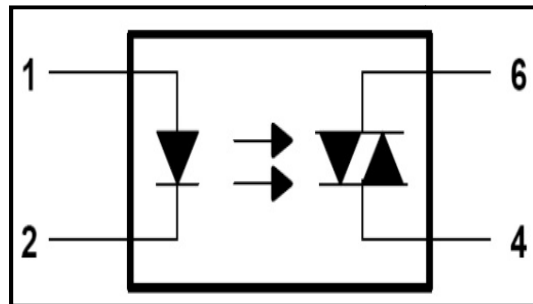


Fig. 7.40. diagrama de pins MOC3021

Al no existir connexió elèctrica entre l'entrada (díode emissor) i la sortida (foto-triac), l'acoblament és unidireccional (LED a foto-TRIAC) i permet un aïllament elèctric entre ambdós dispositius de fins a 7500 V.

Per a disparar l'opto-triac MOC 3021 únicament es necessita un corrent pel LED de 10mA. Quan el led està apagat, el foto-triac està bloquejat conduint un petit corrent de fuga denominat I_{DRM} (peak blocking current). Quan el díode condueix dispara el foto-triac podent circular entre 100mA i 1A. Al no ser un dispositiu que suporti grans nivells de potència, el propi opto-triac actua sobre el control d'un triac de major potència.

Per tal de protegir el díode emissor de llum, es col·loca una resistència de 220 Ω entre +5 VDC i el pin 1 l'opto-triac.

S'utilitza un opto-triac per tal de no atacar directament al triac i així separar la part de control de la part de potència ja que si hi hagués un problema a la part de potència, aquest no afectes a la part de control i a l'inrevés.

El triac té dos estats de funcionament, bloqueig i conducció. Condueix el corrent entre els seus terminals principals en un sentit o en l'invers per tant es pot dir que és un element bidireccional.

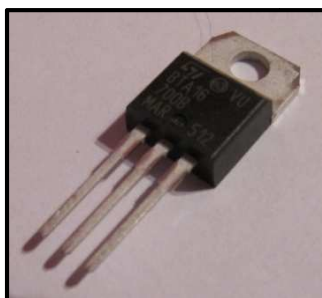


Fig. 7.41. BTA16

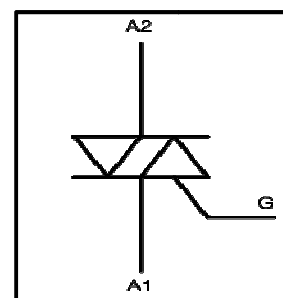


Fig. 7.42. Diagrama de pins triac

El triac utilitzat és el *BTA16* que condueix entre els seus ànodes (A1 i A2) quan se li aplica un senyal a la porta (G) i deixa de conduir quan el senyal de la càrrega passa per 0 tal i com es mostra a la figura.

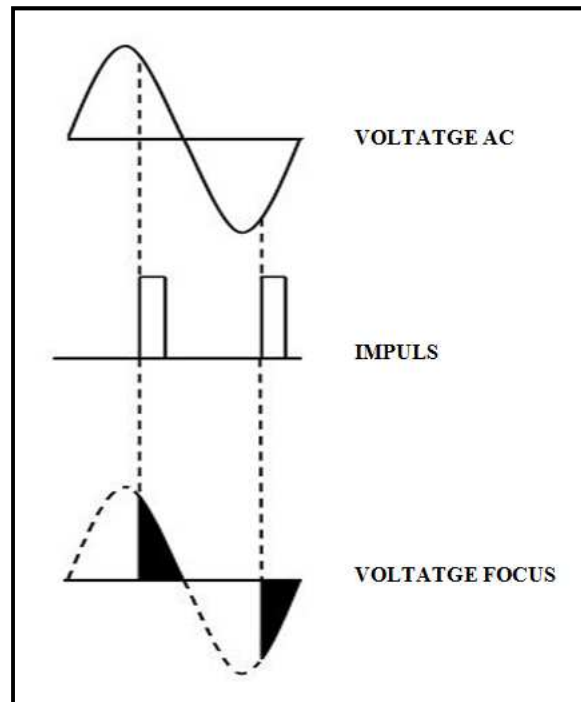


Fig. 7.43. Diagrama de conducció triac

Un altre element utilitzat en aquesta part de potència es el diac, concretament un *TMMDB3* en format smd. El diac és un dispositiu semiconductor de dos connexions. És un díode bidireccional que condueix el corrent només quan s'ha superat la seva tensió de disparada. És un element que actua com un interruptor obert i quan assoleix una tensió de ruptura de 30 volts als seus extrems es tanca.



Fig. 7.44. TMMDB3

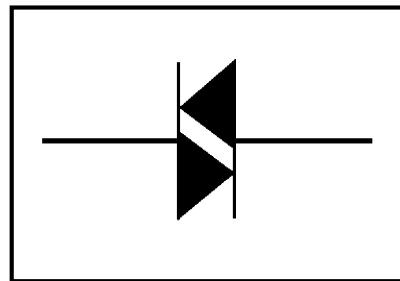


Fig. 7.45. Diagrama de pins diac

El funcionament d'aquesta part de potència és el següent:

Un senyal lògic a nivell "0" procedent del PIC i aplicat a la pota negativa de l'opto-triac, activa el seu díode emissor. Un ànode està connectat directament a una resistència de 470Ω per tal de limitar la intensitat i l'altre ànode es connecta directament a un diac per assegurar així la disparada del triac i que aquest arribi a entrar en conducció i apliqui tensió alterna a la càrrega, en aquest cas el focus.

El control de la potència total aplicada es basa en determinar en quin moment es produeix el senyal de disparada que s'aplica a l'entrada del triac. Per a determinar aquest moment s'utilitza el senyal de referència de pas per 0. Aquest senyal de pas per 0 es produeix cada 10 ms, cada vegada que l'ona d'alterna a controlar passa pel valor de 0 volts. Si justament es provoca la disparada de l'opto-triac en el moment de rebre el senyal de pas per 0 es diu que l'angle de disparada és de 0° . No hi ha retard entre el pas per 0 i el cebat del triac. La potència que s'aplica a la càrrega correspon a la totalitat de qualsevol dels dos semicicles d'alterna de manera que el focus s'encendrà al màxim.

Si un cop rebut el senyal de pas per 0 es realitza un retard de 5ms a la disparada del triac, es diu que hi ha un retard de 90° en el cebat del mateix. A la càrrega només se li aplica la potència corresponent a la meitat de qualsevol semicicle, és a dir, el focus s'encendrà a la meitat de la potència màxima.

Si finalment, rebut el senyal de pas per 0, es realitza un retard entorn als 10 ms en la disparada del triac, es diu que hi ha un retard de 180° en el seu cebat. La càrrega no rep potència atès que el cebat es produeix just quan l'alterna torna a passar per 0 en el següent semicicle. Els focus romanen apagats.

RETARD (ms)	ANGLE ($^\circ$)	VOLTATGE FOCUS (VAC)
0	0	≈ 227
5	90	≈ 113
10	180	0

Taula 7.4. Voltatge dels focus en relació a l'angle de dispar i el retard aplicat

Dit en altres paraules, controlant el retard en que es produeix el senyal de disparada del triac, prenent com a punt de partida el senyal de referència de pas per 0, es controla la potència total que s'aplica a la càrrega. El retard es pot variar entre 0 i 10 ms des del senyal de pas per 0 i això es correspon amb angles de retard d'entre 0° i 180° .

Un factor important a tenir en compte ha estat el de posar una protecció al circuit per si s'han de connectar carregues inductives com bé pot ser un motor. Un motor pot produir una variació brusca de tensió i aquesta podria fer que el triac entrés en conducció de manera indesitjada. Aquesta protecció s'anomena xarxa snubber i consta d'una resistència de 47Ω en sèrie amb un condensador de 100nF connectats entre els ànodes del triac. El condensador limita les variacions brusques de tensió, i quan el triac entra en conducció es descarrega a través d'aquest provocant un corrent elevat capaç de destruir el triac, es llavors quan la resistència limita el corrent per tal que això no passi.

7.2.6. PROCÉS DE DISSENY I CONSTRUCCIÓ DEL CIRCUIT IMPRÈS

Abans de construir el circuit electrònic del regulador de llum, s'ha realitzat un muntatge amb components THD utilitzant una placa de proves i una placa de potència.

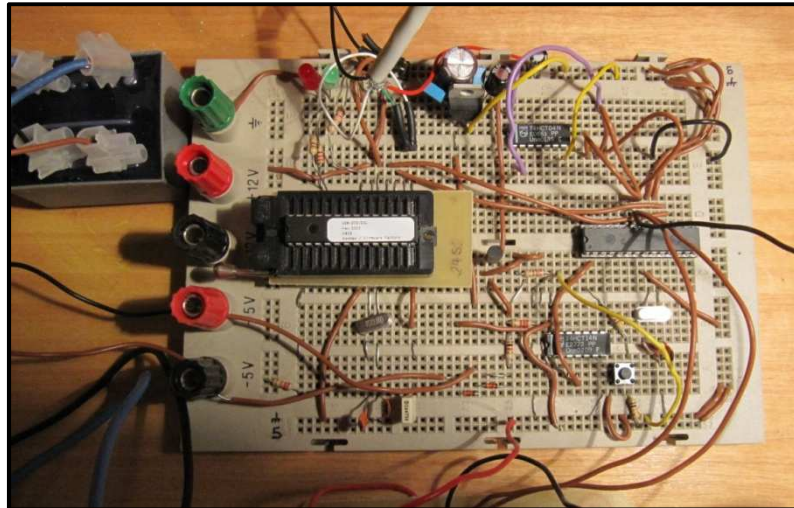


Fig. 7.46. Muntatge en placa de proves

Aquest circuit es capaç de regular la intensitat de 3 focus independentment l'un de l'altre utilitzant un programa provisional implementat en Visual Basic.

Aquest prototip únicament permet establir una comunicació amb l'ordinador utilitzant un cable USB.

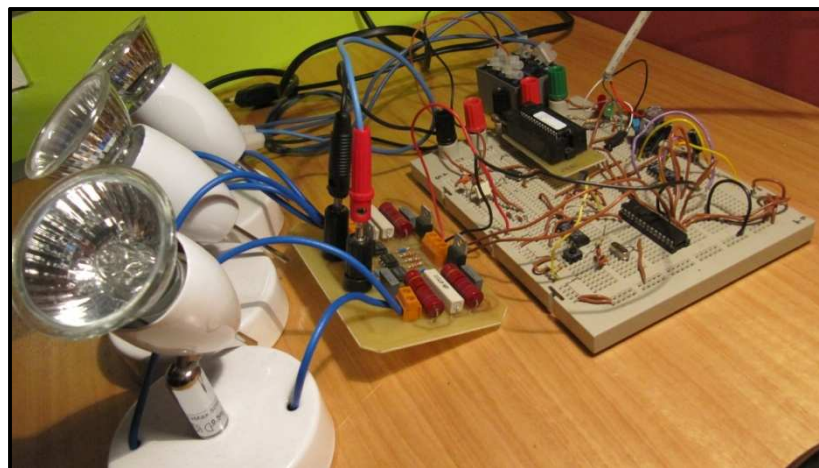


Fig. 7.47 Muntatge en placa de proves

Quan el prototip ha funcionat correctament, s'ha dissenyat i muntat el prototip electrònic del projecte. És en aquest punt on s'ha de tornar a fer una recerca per a veure quins són els components que més s'adapten a les característiques de la placa, tant en

funcionament com en dimensions, aconseguint un bon disseny i una placa més fiable i de dimensions reduïdes.

Per tal de dissenyar i construir un projecte d'aquestes característiques, s'han hagut de seguir uns passos essencials, seguidament comentats, fent una divisió entre la part de disseny del circuit, la part de construcció d'aquest i finalment el seu encapsulat.

Per a dissenyar el regulador de llum, s'ha optat per fer ús d'un software anomenat ALTIUM DESIGNER, un programa líder en el disseny de circuits impresos.



Fig. 7.48. Altium Designer 2009

El programa consta d'una graella on s'efectua el disseny del circuit esquemàtic, allà es col·loquen els components a utilitzar en el projecte amb el seu corresponent patillatge. Els components electrònics es poden trobar a les llibreries internes del programa, o bé es poden crear amb el seu patillatge corresponent.

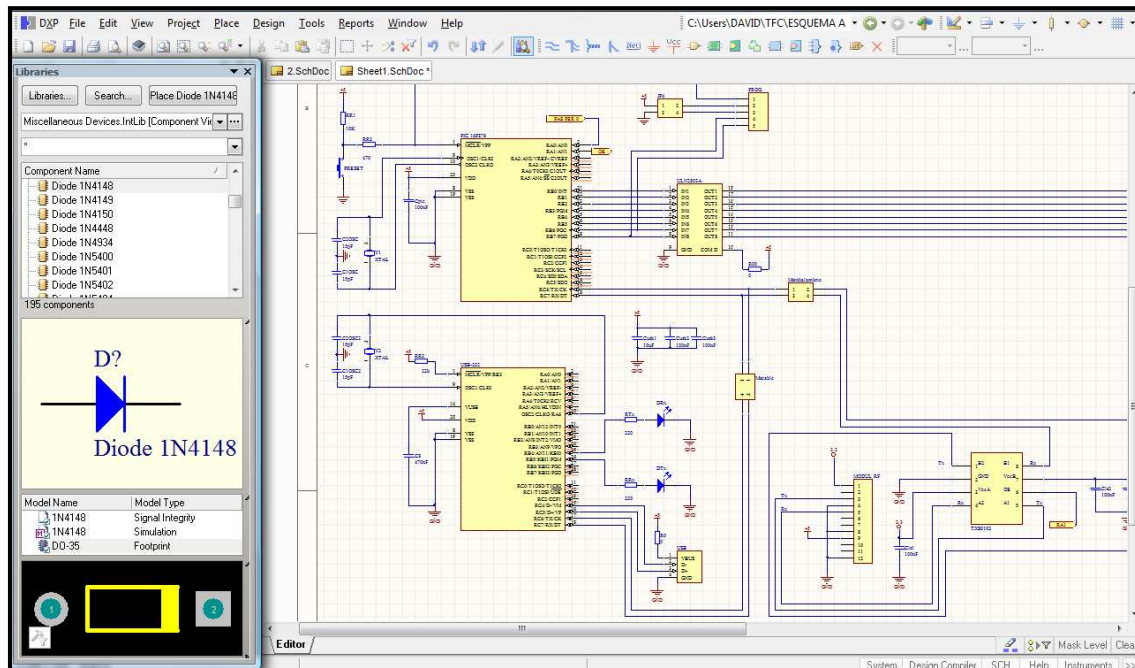


Fig. 7.49. Entorn de disseny de circuits esquemàtics Altium Designer

Una vegada s'ha efectuat la col·locació i connexió de components es crea un netlist que descriu la connectivitat del circuit electrònic per tal de realitzar el disseny de la placa electrònica en un circuit imprès o PCB (Printed Circuit Board). El software Altium Designer incorpora una segona graella per a col·locar els components electrònics i així poder realitzar el corresponent traçat de pistes del què serà la placa final.

La pantalla de disseny de PCB està composta de diferents capes de les que es poden destacar les següents, representades amb el seu corresponent color:

CAPA	DESCRIPCIÓ
TOP LAYER	Capa superior de la placa electrònica a on es col·loquen components i s'efectua el traçat de pistes.
BOTTOM LAYER	Capa inferior de la placa electrònica a on s'efectua el traçat de pistes.
TOP OVERLAY	Capa per a visualitzar els components i la seva descripció.
KEEP OUT LAYER	Capa per a definir el contorn de la placa electrònica.
MULTI-LAYER	Capa a on es col·loquen els pads i les vies dels components.

Taula 7.5. Capes utilitzades en el disseny de PCB

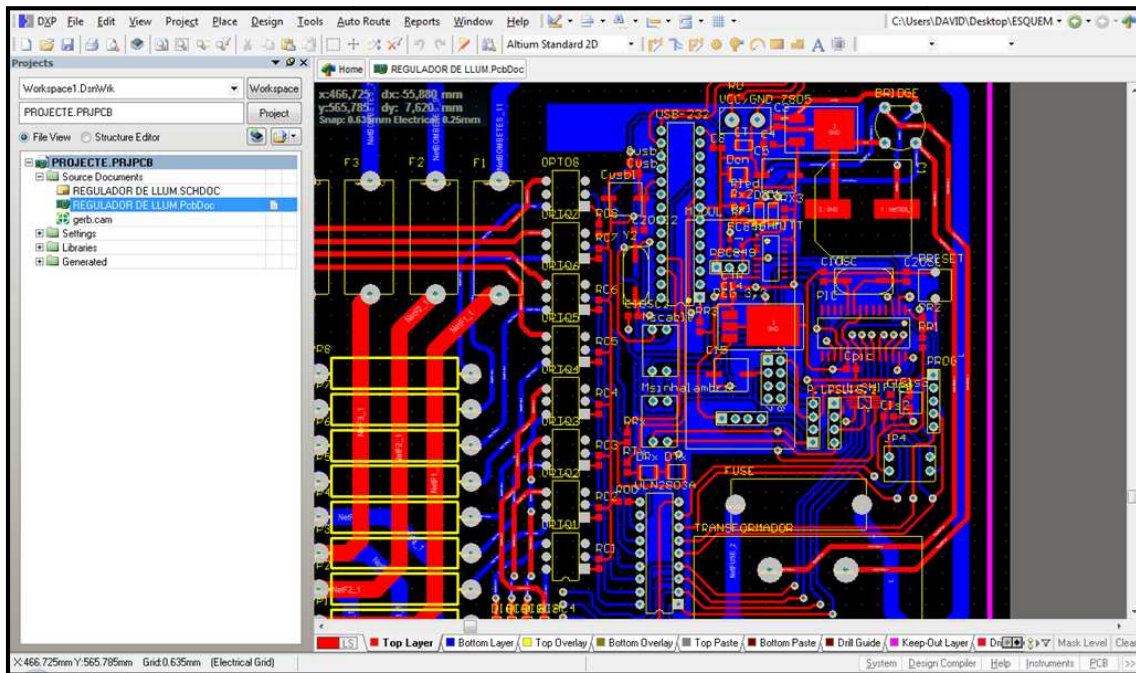


Fig. 7.50. Disseny de PCB Altium Designer

El circuit imprès del projecte està format per components de forat passant THD (*Trough-Hole Device*) i per components de muntatge superficial SMD (*Surface Mount Device*) col·locats a la cara superior en una placa electrònica de doble cara i amb traçat de pistes per la cara superior i la cara inferior. Tal i com es pot veure al PCB, els components s'han situat de manera que la part de potència i la part de control quedin separades mitjançant opto-triacs.

Per a dissenyar el PCB s'han seguit unes normes de disseny fixades pel dissenyador i configurables a l'entorn Altium:

- **Amplada de pistes**
- **Distància entre pistes**
- **Forats dels pads**
- **Traçat de cantonades**
- **Mida de les vies**
- **Reducció de soroll**
- **Desacoblament**

Amplada de pistes:

Dóna la opció de fixar l'amplada de les pistes a traçar de la cara superior i inferior del PCB sota el criteri observat a la figura.

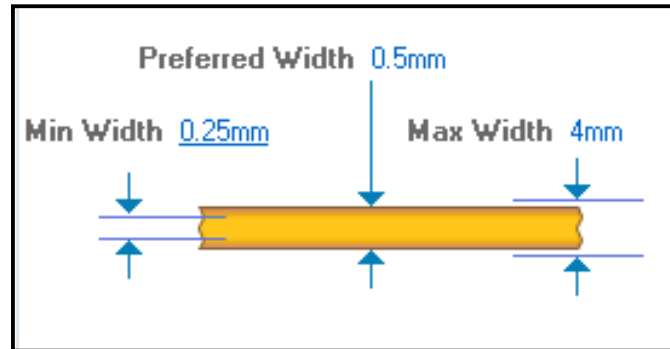


Fig. 7.51. Amplada de pistes

Les pistes de control tenen una amplada de 0.5mm i les de potència 3,2mm.

Distància entre pistes:

S'ha fixat una distància mínima entre pistes diferents de 0.5mm



Fig. 7.52. Distància entre pistes

Forats del pads:

Per tal de foradar la placa per inserir els components, s'ha fixat una distància mínima de forat passant de 0.3mm i una distància màxima de 4mm.

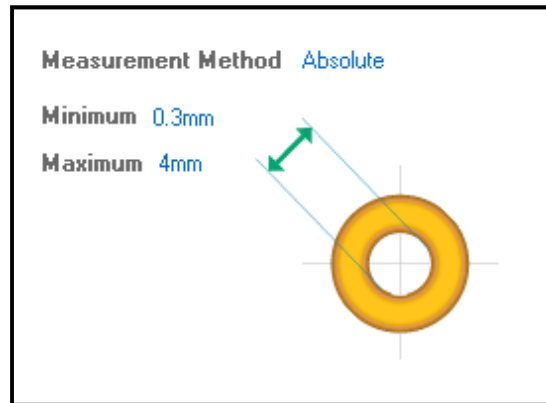


Fig. 7.53. Distàncies de forat passant

Traçat de cantonades:

Per a reduir les reflexions (variacions brusques de la impedància d'una línia de transmissió) al circuit s'ha estipulat que les pistes que fan cantonada tinguin una inclinació de 45° amb una amplada de 2.54mm.

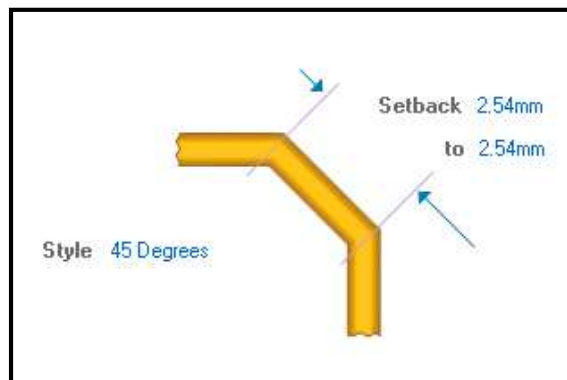


Fig. 7.54. Inclinació cantonades

Mida de les vies:

Les vies comuniquen ambdues cares del circuit, els diàmetres i la mida dels forats a utilitzar en aquest disseny s'aprecien a la figura.

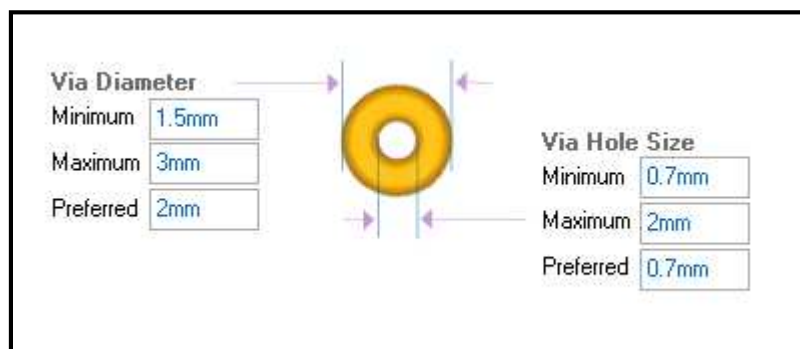


Fig. 7.55. Mida de les vies

Reducció de soroll i dissipació

Per a reduir el soroll del circuit i per a generar una major àrea de dissipació de potència procedent dels reguladors de voltatge, s'ha implementat un pla de massa a la cara posterior del circuit.

Desacoblament:

Un factor important és la col·locació de condensadors de desacoblament en els diferents integrats digitals que componen el regulador de llum. S'han col·locat condensador de 100nF entre la línia +5VDC i GND el més a prop possible de l'integrat per evitar que les interferències produïdes per aquest entrin al circuit. Quan s'activen varies sortides del PIC, el consum augmenta i es produeix una caiguda de la tensió d'alimentació del PIC i llavors la resposta és més lenta. Col·locant el condensador de desacoblament el més proper al PIC, s'assegura que les respostes d'aquest siguin sempre igual de ràpides.

La placa on s'ha construït el circuit és de baquelita i conté una capa de coure a banda i banda amb un espessor de 35 μm , una vegada metal·litzada (procés que es veu més endavant), i assoleix un espessor de 60 μm .

El disseny del circuit està pensat per a controlar càrregues de fins a 100W per canal, per tant es pot deduir l'amplada de la pista de la següent manera:

Tenint en compte el voltatge màxim amb el que es pot regular les càrregues $V \approx 230\text{VAC}$, la potència d'aquestes $P=100\text{W}$ per canal i tenint 8 canals, de la següent expressió s'obté:

$$\begin{aligned}P &= I * V \\I &= \frac{800}{230} \\I &\approx 3,5A\end{aligned}$$

On I és la intensitat màxima que circula per les pistes de potència de la càrrega. Per una intensitat com aquesta i un espessor de 60 μm , s'aconsella una amplada de pista de 3,2mm.

Pel que fa a les pistes de control, s'ha optat per a fer-les de 0.5mm perquè sigui més senzill construir el circuit imprès ja que si es fan més primes dificulta el procés. Amb aquesta amplada, les pistes poden suportar fins a 600mA.

No s'han tingut en compte condicionants especials per fer el traçat de pistes de freqüència elevada ja que aquest factor en aquest projecte no té una influència important.

Una vegada finalitzat el disseny s'obté un PCB de 207,6 mm de llargada i 196,2 mm d'amplada amb 128 components, 435 pads i 59 vies.

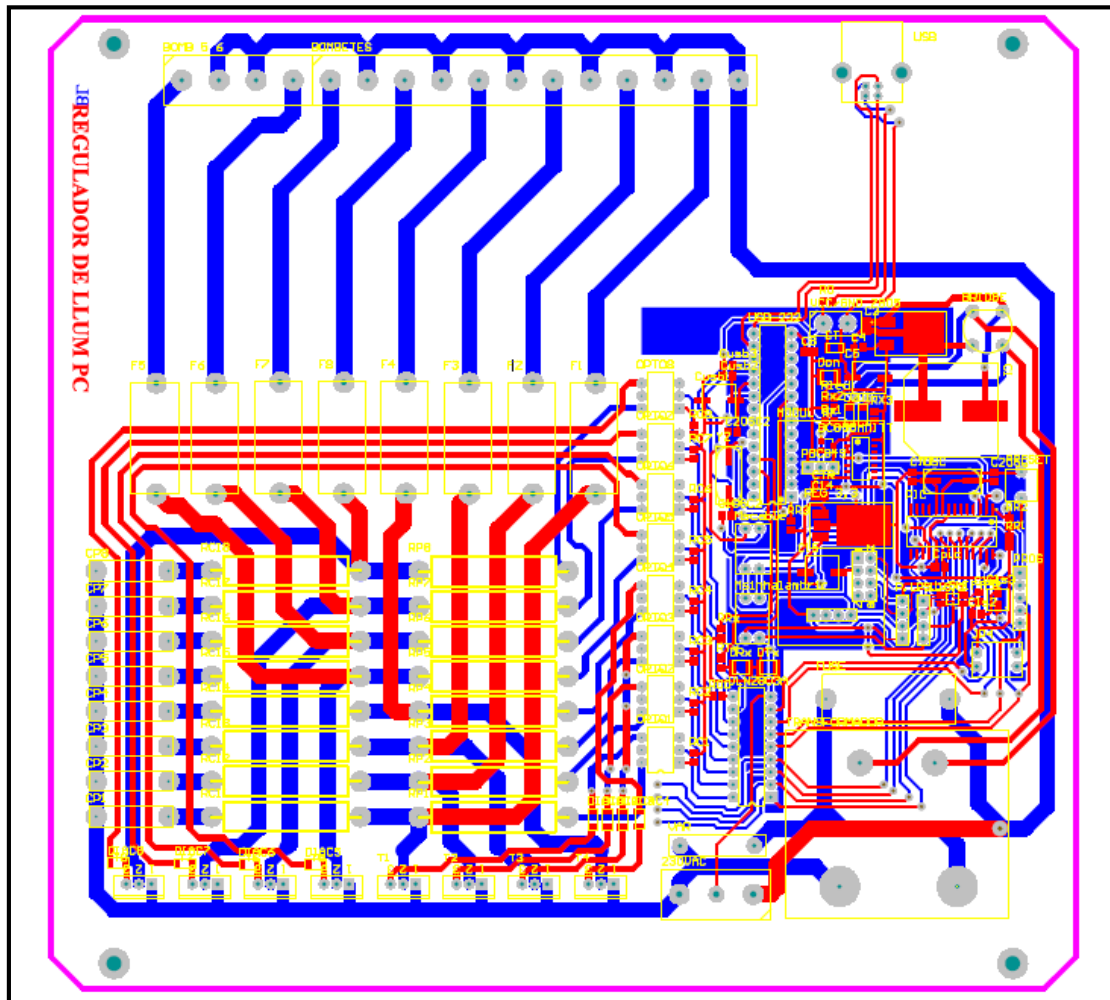


Fig. 7.56. Disseny de PCB

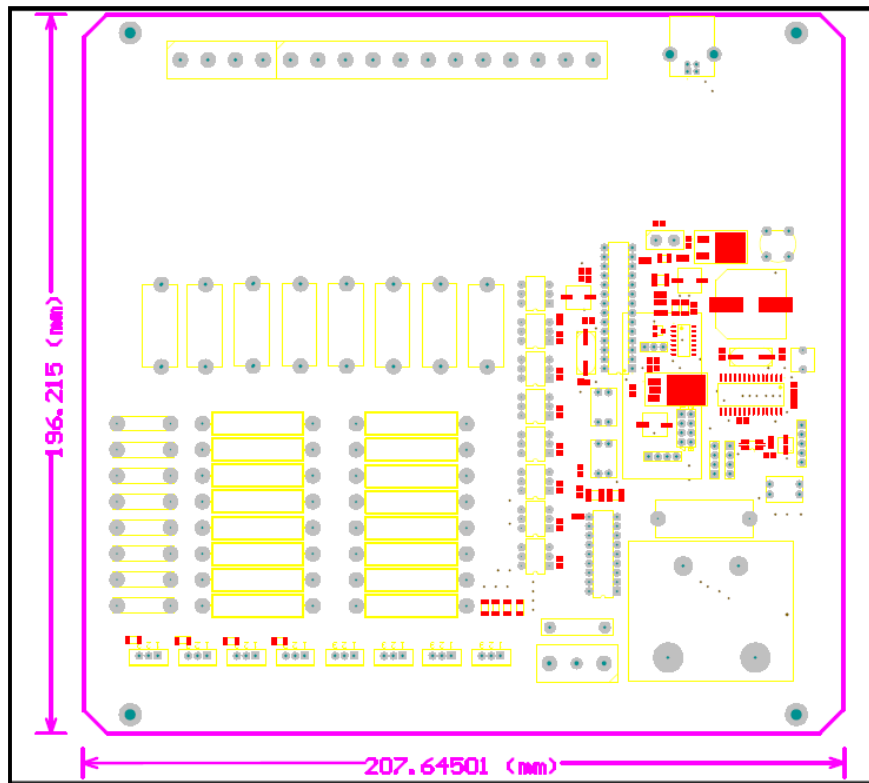


Fig. 7.57. Mida i situació de components al PCB

El software Altium permet una visualització en 3 dimensions del disseny final.

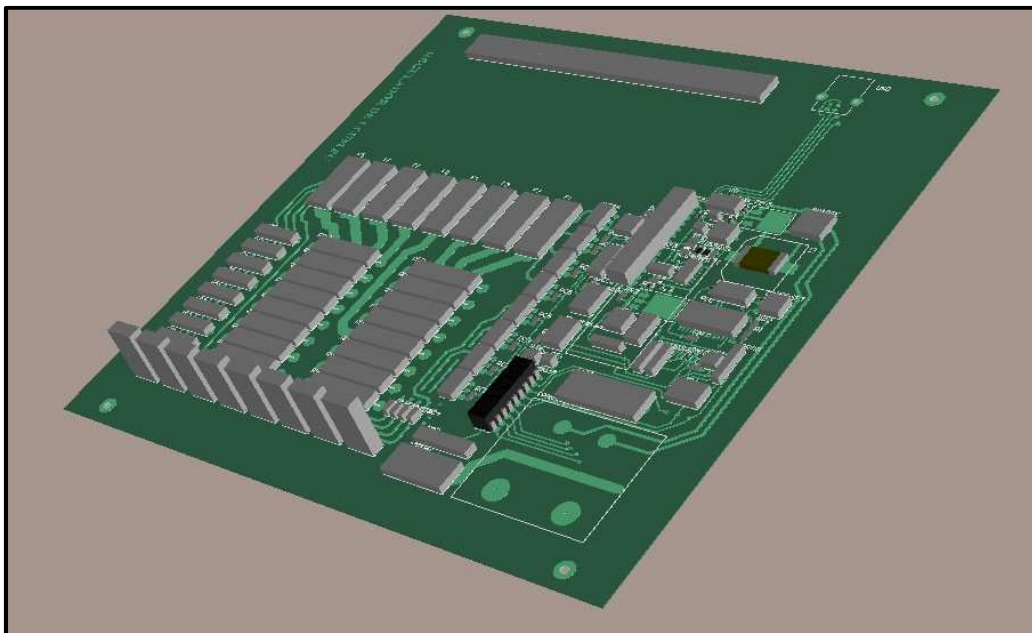


Fig. 7.58. Simulació en 3D del PCB

Una vegada finalitzat el disseny del PCB s'ha procedit a construir una placa a doble cara. La construcció s'ha dut a terme al laboratori d'electrònica de l'Escola del Treball del Ripollès.

Primerament s'han exportat els gerbers del PCB. Un gerber és un format d'arxiu utilitzat per les màquines de construcció de plaques de circuit imprès per a dissenyar les connexions elèctriques com poden ser les vies, les pistes i els pads del PCB. El gerber també conté informació dels forats a realitzar i del fresatge de la placa. Els gerbers a exportar han estat el de la capa superior de la placa (*GTL*), el de la capa inferior (*GBL*) i el de coordenades de forats (*GDR*) en un format estàndard *RS-274-X*.

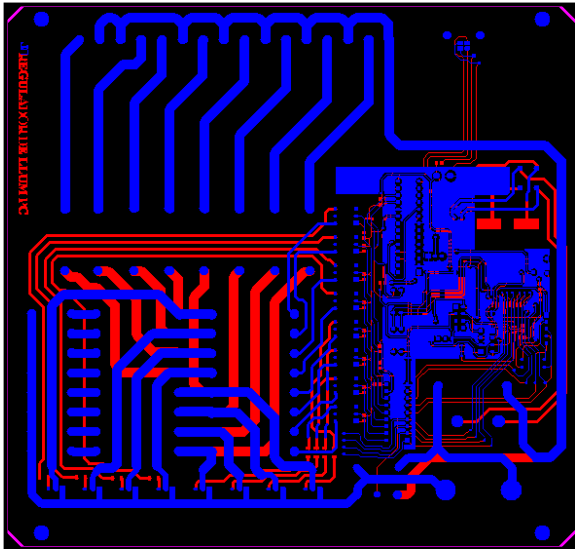


Fig. 7.59. Gerbers cara superior i cara inferior

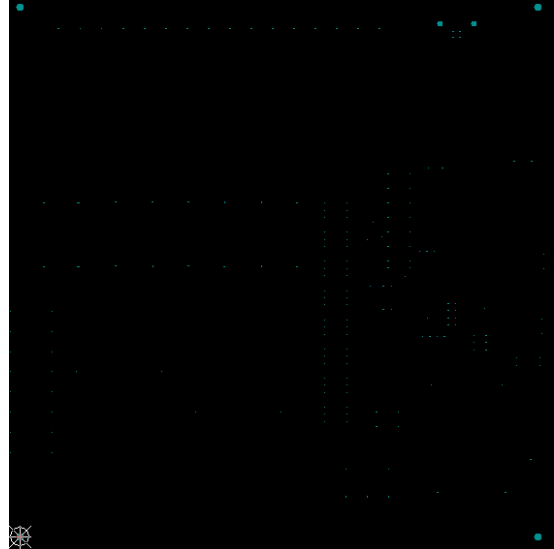


Fig. 7.60. Gerber coordenades forats

El següent pas ha estat foradar la placa mitjançant un trepant amb control numèric connectat a un ordinador i que utilitza el gerber de foradat de la placa.



Fig. 7.61. Trepant controlat per ordinador

Ara la placa ja està foradada, però falta comunicar elèctricament la cara superior amb la cara posterior metal·litzant les vies i els pads.

Per fer aquest procés s'ha utilitzat la màquina *PL903S* que efectua un metal·litzat electrolític dels forats de la placa de circuit imprès dipositant 25 µm de coure.



Fig. 7.62. PL903S

El temps de durada del procés és de 60 minuts aproximadament i es divideix en els sub-processos següents:

1. Preparació preliminar:

Escalfar les solucions a utilitzar fins a la temperatura de treball.
Netejar la placa.

2. Condicionament/neteja:

Es submergeix la placa en el líquid *ABC189* que neteja la superfície de la placa i elimina els residus. Aquesta solució origina també un potencial elèctric a les parets dels forats i en les superfícies de coure, atraient les partícules del catalitzador i creant una fina capa conductora.

- Temps: 7 minuts.
- Temperatura: 65°C.

3. Rentat:

Rentat de la placa per a eliminar els restes del condicionador i evitar la contaminació del següent procés.

- Temps: 60 segons.

4. Cuba de pre-dip:

Submergir la placa a la solució *ABC 889A*.

- Temps: 30 segons.
- Temperatura: 20-25°C.

5. Catalitzat:

Submergir la placa a la solució *ABC889* per a recobrir les parets dels forats d'una fina i uniforme capa conductora de catalitzador. Durant el procés la superfície de coure també es recobreix de catalitzador.

- Temps: 7 minuts.
- Temperatura: 42°C.

6. Rentat:

Idem. punt 3.

- Temps: 60 segons.

7. Eliminat de sals:

S'utilitza el líquid *ABC581* i s'elimina el catalitzador dipositat a les superfícies de coure, sense fer-ho dels forats.

- Temps: 4 minuts.
- Temperatura: 45°C.

8. Rentat:

Idem. punt 3.

- Temps: 2minuts

9. Metal·litzat de coure:

Diposita electrolíticament una capa de coure de 25µm d'espessor a les parets dels forats i a la superfície de la placa.

- Temps: 36 minuts.
- Temperatura: 25-30°C.

10. Rentat:

Idem. punt 3.

- Temps: 15 segons

11. Assecat:

Assecar la placa amb paper i aire calent per a evitar la oxidació.



Fig. 7.63. Procés de metal·litzat PL903S

El resultat del procés de metal·litzat es pot apreciar a la següent imatge:

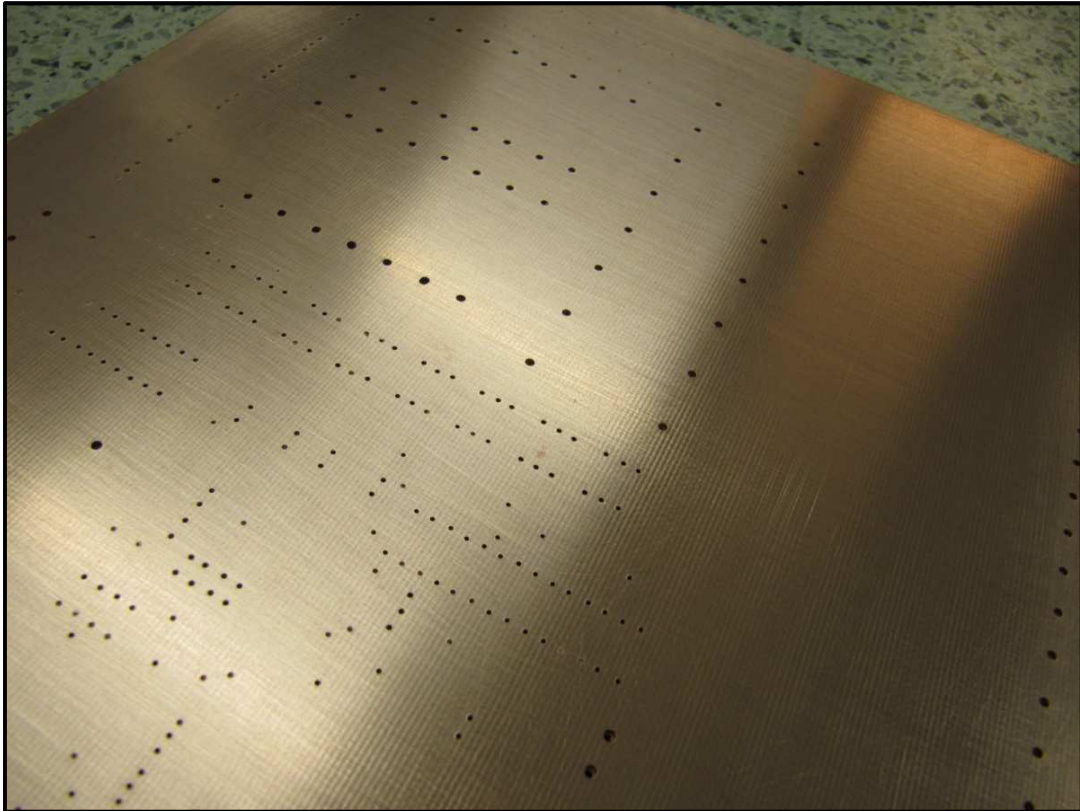


Fig. 7.64. Placa de coure metal·litzada

Quan ja es té la placa metal·litzada s'ha de continuar el procés en una sala fosca degut a que es treballa amb material fotosensible. Es ara quan s'ha de sensibilitzar la placa, aquest procés s'ha efectuat utilitzant una laminadora de film negatiu de doble cara *GX12*.

Efectuant un altre procés s'han obtingut els fotolits negatius per a insolar la placa de coure utilitzant el *Photoplotter 'FP-800'* que és un equip que dibuixa imatges en una pel·lícula fotogràfica mitjançant un díode làser. Aquest aparell es controla amb el programa *Run_Photo_USB* que llegeix fitxers bitmap d'alta resolució (.fpf) instal·lat en un ordinador fora de la sala fosca. L'arxiu *fpf* s'obté a partir de convertir els gerbers mitjançant el programa *Gerb2Bitmap*.

Per a insolar la placa es necessiten dos fotolits negatius, el de la cara superior i el de la cara inferior. Una vegada s'obtenen els fotolits s'han de revelar, fixar i netejar. El temps necessari per a realitzar el procés es de 33 minuts per fotolit separats en:

- Photoplotter: 27 minuts.
- Revelat: 1 minut.
- Fixat: 1 minut.
- Neteja i assecat: 4 minuts.



Fig. 7.65. Photoplotter FP-800

El pas següent és insolar durant 25 segons els fotolits damunt la placa de coure amb els forats ja fets i revelar-la utilitzant la màquina 'PA310' durant 4 minuts a 35°C per agilitzar el procés.

Per a treure el coure sobrant de la placa s'introdueix a la unitat d'atacat per spray 'CP300DS' que li aplica una solució de clorur fèrric. Finalment s'elimina la pel·lícula fotosensible que ha quedat damunt les pistes i es neteja.

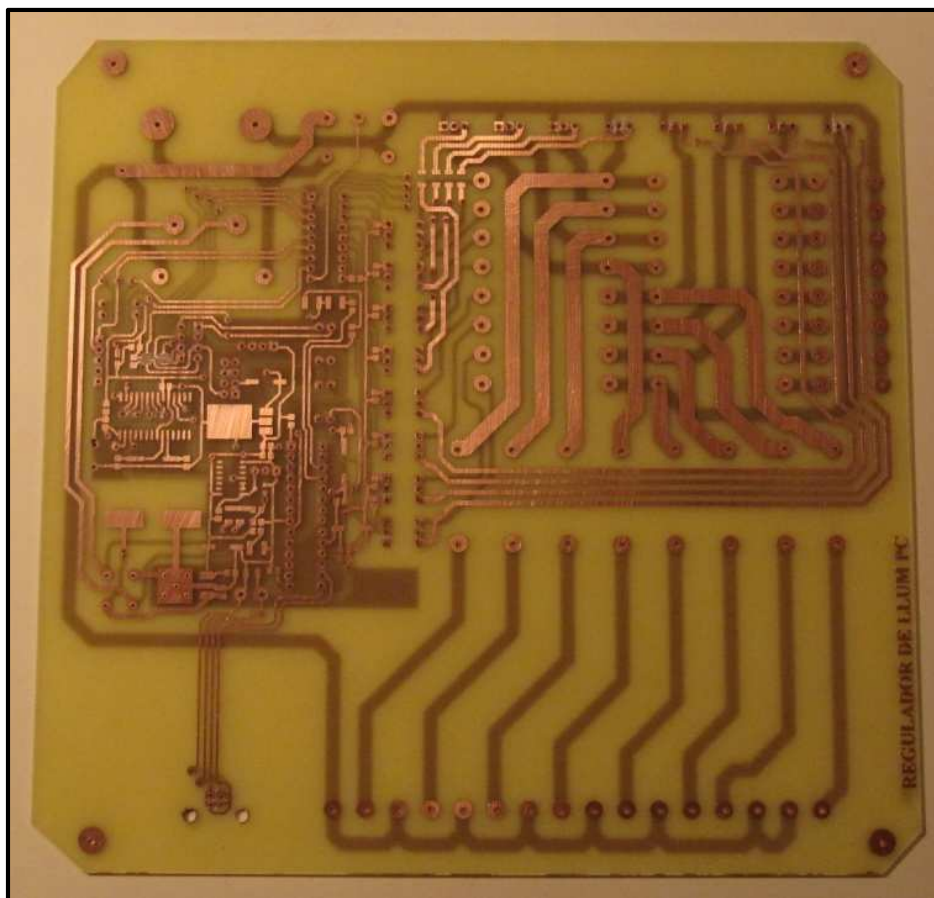


Fig. 7.66. Placa de circuit imprès

Una vegada s'obté la targeta de circuit imprès es comprova la bona conductivitat entre les dues cares del circuit i acte seguit se solden tots els components que formen la placa amb un soldador de components smd i amb un soldador convencional.

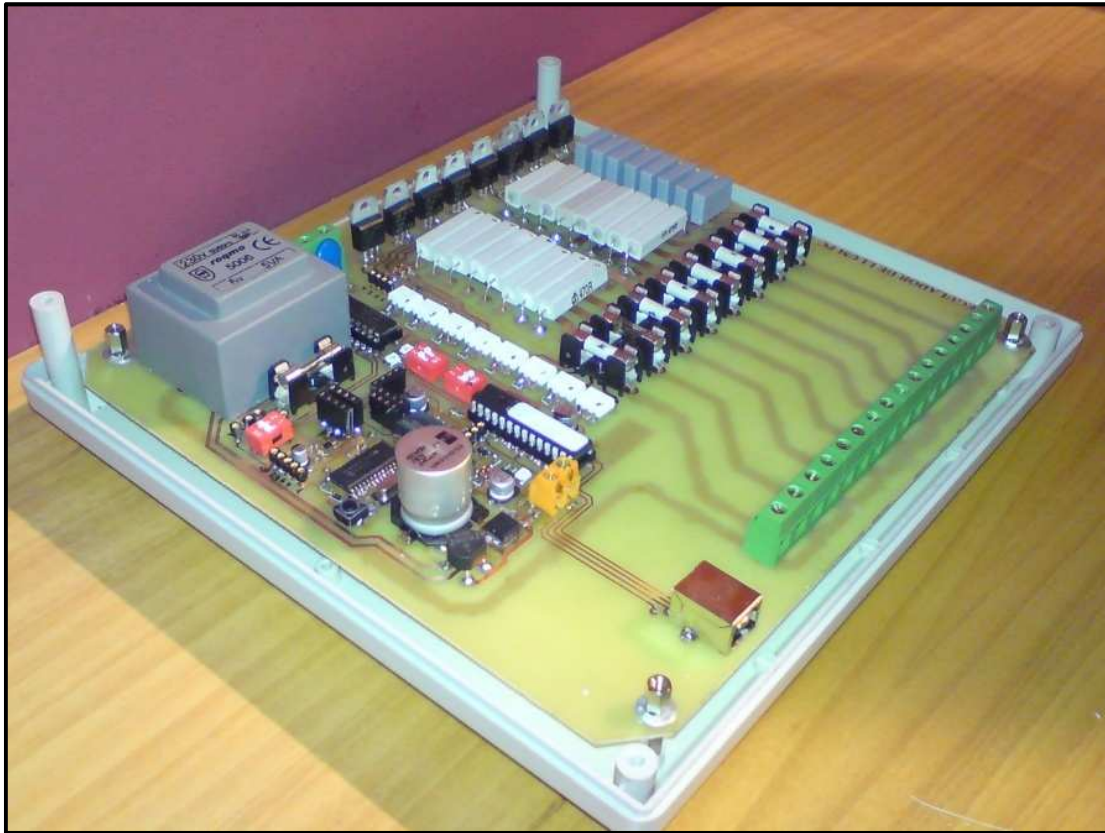


Fig. 7.67. Regulador de llum controlat per ordinador

Per a obtenir unes bones prestacions i una bona seguretat, s'ha decidit encapsular la placa en una caixa per a muntatges electrònics amb tapa d'alumini fabricada en plàstic ABS d'alt impacte. Les dimensions de la caixa són de 228x216x75/50mm.

El circuit electrònic es subjecta mitjançant cargols de mètric 3mm col·locats a cada una de les cantonades de la placa.

L'encapsulat protegeix el circuit de possibles manipulacions externes i també protegeix als usuaris de possibles descàrregues elèctriques, aspecte a tenir en compte ja que hi han pistes per on hi circula una tensió de 230 VAC amb una intensitat de 2 A.

A l'encapsulat es poden trobar connexions per a connectar els 8 focus i el cable USB procedent de l'ordinador. També s'ha posat un interruptor per a efectuar l'encesa i l'apagada de l'aparell.



Fig. 7.68. Encapsulat regulador de llum controlat per ordinador

7.3. RESOLUCIÓ D'IMPREVISTOS

A l'hora de realitzar aquest treball han sorgit varies complicacions i imprevistos que s'han anat resolent a mesura que el temps ho ha permès.

Els imprevistos s'estructuren de la següent manera:

- **Imprevistos de recerca**
- **Imprevistos de construcció del prototip**
- **Imprevistos de disseny del PCB**
- **Imprevistos de fabricació del circuit imprès**
- **Imprevistos del prototip**
- **Imprevistos no solucionats**

7.3.1. IMPREVISTOS DE RECERCA

Abans de decidir quin seria el disseny final del circuit, es van provar altres solucions citades a la memòria descriptiva.

Utilitzant el component EXPANDIO/USB i el PIC18F2550 es va comprovar que la programació resultava complicada.

7.3.2. IMPREVISTOS DE CONSTRUCCIÓ DEL PROTOTIP

Mentre s'esperava l'arribada del component USB-232 es van fer proves de comunicació entre l'ordinador i un circuit electrònic utilitzant el port sèrie. Fent aquestes proves es van trobar varis problemes:

- El cable sèrie a utilitzar ha de ser null-modem i s'estava provant amb un RS232 estàndard.
- La pota de recepció de dades del microcontrolador estava internament creuada a massa.
- El port sèrie de l'ordinador estava malmès.

Per a solucionar aquests problemes, es va comprar un cable null-modem, un nou microcontrolador i es va provar el circuit amb un altre ordinador.

Al construir la font d'alimentació del circuit es va trobar un pont de díodes malmès i es va canviar per un de nou.

Una soldadura malmesa a la pista de massa de la placa de potència impedia el funcionament del circuit.

A nivell de programació en Visual Basic, no es podia regular la intensitat lumínica dels focus mentre que amb el software subministrat per Hexwax, la regulació era perfecta. Es va establir contacte amb la casa Hexwax i es va descobrir que s'havia d'enviar una comanda d'identificació dels bits a enviar.

7.3.3. IMPREVISTOS DE DISSENY DEL PCB

Degut als anys sense utilitzar el programa de disseny '*ALTIUM DESIGNER*' i a la constant evolució d'aquest, es va tornar a "aprendre" el seu funcionament i les noves funcions que incorpora.

Quan ja es tenia dissenyat tot el PCB, es van afegir components que van prolongar i dificultar el disseny.

Hi va haver una mala exportació del format gerber que va impedir tirar endavant a l'hora de fabricar el circuit.

7.3.4. IMPREVISTOS DE FABRICACIÓ DEL CIRCUIT IMPRÈS

Al utilitzar la maquinària per a la fabricació de circuits electrònics de l'Escola del Treball del Ripollès de la qual se'n desconeixia el seu funcionament, es van tenir diversos imprevistos.

Quan la placa estava metal·litzada, aquesta, en contacte amb l'aire es va oxidar i es va haver de treure aquest òxid utilitzant paper de vidre i unes gomes especials.

Per a obtenir els fotolits negatius amb el *'photoplotter'* es va trobar que aquest no establia connexió amb l'ordinador. Com que el *'photoplotter'* es troba en una sala fosca i l'ordinador en una altra sala, el cable USB es bastant llarg i travessa la paret, cosa que dificulta el seu canvi. Es van desinstal·lar i tornar a instal·lar els *drivers* inclús en diversos ordinadors, però seguia sense funcionar. La solució va estar en canviar el cable USB.

El procés d'obtenció de fotolits es va fer tres vegades sense èxit. La primera vegada va sortir el fotolit tort, la segona el revelador no va fer la seva funció i la tercera el fixador no va fixar la impressió al paper fotogràfic. Es van comprovar les proporcions i al observar que eren correctes es va procedir a realitzar una trucada telefònica al fabricant del *'photoplotter'* que va assegurar que els líquids devien estar en mal estat i va suggerir que es fes la proporció amb líquids nous. Utilitzant un revelador i un fixador nous, es va obtenir el fotolit.

Al ser la placa electrònica massa gran, a l'hora de revelar-la a la màquina de revelat, el líquid no va penetrar a tota la superfície de la placa impeding així un correcte atacat de clorur fèrric. Es va atacar la placa diverses vegades per a eliminar sense èxit el coure de la placa on no havia tocat el revelador. Aquest procés va malmetre el metal·litzat de les vies i d'algun pad. Es va optar per realitzar una placa nova.

Al fabricar la segona placa també es van tenir problemes amb el revelat i es va haver de fer una tercera placa definitiva.

7.3.5. IMPREVISTOS DEL PROTOTIP

Alguna de les vies no comunicava i es va utilitzar un cable per a comunicar ambdues cares del circuit.

Es van trobar dos fusibles nous malmesos.

L'*USB-232* va deixar de funcionar per causes desconegudes. Es va establir contacte amb el fabricant exposant el problema i no s'ha rebut cap solució. Es va canviar per un de nou.

Un canal del regulador no funcionava; primerament es va canviar l'integrat *ULN2803A* pensant que aquest estava malmès i no enviava el senyal adequat a l'optotriac. Al veure que seguia sense funcionar, es va deduir que el problema podia venir del *PIC*. La pota 3 del *PORTB* del *PIC* està malmesa internament i no emet cap tipus de senyal. Canviar el pic resulta difícil degut al ser de format smd i s'ha optat per no canviar-lo. Degut a aquest fet, el regulador de llum únicament pot regular la intensitat de 7 focus.

Problemes de soldadura amb l'integrat *TXS0102*. Al ser tant petit, el procés d'atacat per àcid no va definir les pistes. Aquestes es van fer a mà amb un bisturí.

Al connectar 7 focus al regulador de llum, es va veure que quan aquests es trobaven a la seva intensitat màxima i al baixar la intensitat fins a 0 d'un d'ells, aquest no s'apagava. Per solucionar-ho es van donar més punts de regulació tant al programa de Visual Basic com al programa del microcontrolador. Hi ha un petit inconvenient, i es que quan tots els focus estan apagats, hi ha varis punts de regulació en què el focus roman apagat.

7.3.6. IMPREVISTOS NO SOLUCIONATS

Un dels imprevistos no solucionats és el control de regulació de velocitat de motors. No s'ha pogut fer un correcte control de velocitat d'aquests.

Un altre imprevist es pot veure en la regulació d'intensitat, quan es regula un focus, la intensitat lumínica dels altres disminueix mínimament.

Degut a imprevistos d'última hora, no s'han rebut els mòduls de radiofreqüència. De totes maneres tant el programa informàtic que gestiona la intensitat dels focus com el circuit electrònic estan preparats per a aquest tipus de comunicació segons les característiques fixades pel fabricant. S'espera tenir els mòduls de radiofreqüència per a la presentació d'aquest projecte.

8. INSTRUCCIONS D'ÚS

A continuació es descriuen els passos a seguir per tal d'obtenir un correcte funcionament del regulador de llum.

8.1. REGULADOR DE LLUM AMB CABLE USB

1. Connectar focus d'una potència màxima de 100W a cada canal del regulador.
2. Desactivar microswitch comunicació radiofreqüència.
3. Activar microswitch comunicació amb cable.
4. Connectar el Regulador de llum a una tensió de 230 VAC.
5. Encendre l'interruptor.
6. Connectar el cable al Port USB de l'ordinador.
7. Obrir el programa "*REGULADOR DE LLUM PC*" adjunt al DVD.
8. Regular la intensitat dels focus.

8.2. REGULADOR DE LLUM RADIOFREQUÈNCIA

1. Connectar focus d'una potència màxima de 100W a cada canal del regulador.
2. Desactivar microswitch comunicació amb cable.
3. Activar microswitch comunicació radiofreqüència.
4. Connectar el Regulador de llum a una tensió de 230 VAC.
5. Encendre l'interruptor.
6. Connectar el mòdul de radiofreqüència IKNOS al port USB.
7. Obrir el programa "*REGULADOR DE LLUM PC*" adjunt al DVD.
8. Obrir la pestanya "Veure" del menú del programa "*REGULADOR DE LLUM PC*".
9. Fer click sobre la opció "Identificadors del dispositiu".
 - Introduir a Vendor ID el següent codi: 0403
 - Introduir a Product ID el següent codi: 6001
10. Premer el botó "*CONNECTA*".
11. Regular la intensitat dels focus.

8.3. CANVI DE DISPOSITIU AMB EL PROGRAMA EN MARXA

1. Extreure el dispositiu connectat al Port USB.
2. Desactivar microwitch del dispositiu actual.
3. Activar microswitch corresponent al nou dispositiu.
4. Connectar el nou dispositiu.

5. Obrir la pestanya “Veure” del menú del programa “*REGULADOR DE LLUM PC*”.
6. Fer click sobre la opció “*Identificadors del dispositiu*”.
7. Per a utilitzar el regulador amb cable:
 - Introduir a Vendor ID (hex) el següent codi: 0b40
 - Introduir a Product ID (hex) el següent codi: 011a
8. Per a utilitzar el regulador sense cable:
 - Introduir a Vendor ID (hex) el següent codi: 0403
 - Introduir a Product ID (hex) el següent codi: 6001
9. Prémer el botó “*CONNECTA*”.

8.4. PROGRAMA “REGULADOR DE LLUM PC”

“ESTAT DEL DISPOSITIU”:

Permet veure si el dispositiu està o no connectat.

“INTENSITAT INDIVIDUAL”:

- Barres: Regulació d'intensitat lumínica de 7 focus independentment l'un de l'altre.
- Botons “*MÀXIM*”, “*MIG*” i “*MÍNIM*”: Encesa màxima, mitjana i mínima individual dels focus.

“INTENSITAT GLOBAL”:

- Botó “*MÀXIM*”: Encesa màxima de tots els focus.
- Botó “*INTERMIG*”: Encesa mitja de tots els focus.
- Botó “*MÍNIM*”: Focus apagats.

“INTERMITENT”:

- Botó “*INTERMITENT*”: Intermitència dels focus prèvia selecció de la intensitat lumínica desitjada. Al polsar-lo el botó canvia la seva funció i permet parar la intermitència.
- Barra “*ON*”: Permet escollir el temps que els focus romandran encesos.
- Barra “*OFF*”: Permet escollir el temps que els focus romandran apagats.

“RÀFAGA”:

- Botó “*RÀFAGA*”: Ràfega de focus prèvia selecció de la intensitat desitjada. Al pulsar-lo el botó canvia la seva funció i permet parar la ràfega.
- Barra “*RÀFAGA*”: permet escollir la velocitat de la ràfega.

“PROGRAMACIÓ”:

- Botó “*PROGRAMAR*”: Obre una finestra de programació.
 - “*CALENDAR*”: Permet seleccionar la data o dates (de 1 a 365 dies), la hora, el minut i el segon en què es desitja que s’activi o es desactivi el regulador de llum.
 - “*INTENSITAT INDIVIDUAL*”: Intensitat individual desitjada per a la programació.
 - “*INTENSITAT GLOBAL*”: Intensitat global desitjada per a la programació.

Per acceptar la programació, fer click al botó “*ACCEPTAR*” o pulsar la tecla “*ENTER*”.

Per a cancel·lar la programació, fer click al botó “*CANCEL·LAR*” o pulsar la tecla “*ESC*”.

“MENÚ”:

- “*Arxiu*”--“*Sortir*”: Permet sortir del programa.
- “*Veure*”--“*Hora*”: Permet mostrar o ocultar la hora actual de l’ordinador i la hora de programació.
- “*Veure*”--“*Identificadors del dispositiu*”: Permet mostrar i ocultar el Product (ID) i el Vendor (ID) dels dispositius a connectar.
- “*Veure*”--“*Comprovacions*”--“*Numeració de barres*”: Permet mostrar i ocultar el valor numèric associat a cada una de les barres.
- “*Veure*”--“*Comprovacions*”--“*Resultats*”: Permet mostrar i ocultar una pantalla de resultat útil per al programador.

8.5. IMPREVISTOS

- En cas que el sistema operatiu deixi de funcionar, els focus romandran encesos amb la última intensitat seleccionada.
- En cas de falta de subministrament elèctric durant el funcionament del regulador, s'haurà de reiniciar el procés.
- En cas de salt del diferencial principal de l'edifici on està situat, s'haurà de reiniciar el procés.

9. PROVES I RESULTATS

Per a obtenir les especificacions finals del producte, s'han realitzat un seguit de proves per tal d'obtenir les especificacions finals del regulador de llum.

9.1. PROVES

- **Sistemes operatius**
- **Regulació de velocitat de motors**
- **Consum**

9.1.1. SISTEMES OPERATIUS

S'ha provat el regulador de llum amb els següents sistemes operatius:

- **Windows 98**
- **Windows XP**
- **Windows vista**
- **Windows 7**

El resultat ha estat satisfactori, i es pot dir que el regulador funciona sense cap mena de problema amb aquests sistemes operatius.

9.1.2. REGULACIÓ DE VELOCITAT DE MOTORS

S'han fet proves connectant dos motors de ventiladors i un motor monofàsic de 50W. Amb els motors dels ventiladors, la regulació comença quan el voltatge de sortida de la placa electrònica cap a aquest estava per sobre dels 100VAC, obtenint així un control sobre el ventilador de 4 posicions de velocitat.

Pel que fa al motor monofàsic de 50W, cal dir que el motor funciona correctament a voltatges d'entre 0 i 150 aproximadament, per sobre d'aquest voltatge el funcionament no és correcte.

9.1.3. CONSUM

Per a observar el consum del regulador de llum, s'han efectuat unes mesures connectant diversos focus de potències diferents als canals del regulador durant un temps determinat. Les taules següents indiquen l'amperatge que gasten els focus per a

una intensitat màxima, mitjana i mínima. També s'ha mesurat el consum total del regulador tenint en compte el consum de tots els elements que el componen.

Aquestes mesures s'han realitzat amb 7 focus i no amb 8, degut al no funcionament d'una pota del PIC.

- **Mesura1:** temps 20 minuts

CANAL	POTÈNCIA (W)	I.MAXIMA (A)	I.MIG (A)	I.MÍNIM (A)
1	50	0,19	0,09	0,00
2	50	0,18	0,09	0,00
3	50	0,19	0,1	0,00
4	100	0,41	0,14	0,00
5	100	0,42	0,15	0,00
6	40	0,16	0,07	0,00
7	35	0,13	0,05	0,00
8	-	-	-	-

Taula 9.1. Mesura 1 d'intensitat per canal

I.TOTAL MÀXIMA (A)	I. TOTAL MIG (A)	I.TOTAL MÍNIMA (A)
1,8	0,77	0,00

Taula 9.2. Mesura 1 d'intensitat total

D'aquesta mesura es pot dir que el regulador de llum funciona sense cap mena de problemes al no rebassar la intensitat total màxima permesa establerta pel fusible de línia de 3,15 A.

- **Mesura2:** temps 20 minuts.

CANAL	POTÈNCIA (W)	I.MAXIMA (A)	I.MIG (A)	I.MÍNIM (A)
1	100	0,42	0,15	0,00
2	100	0,41	0,15	0,00
3	100	0,41	0,14	0,00
4	100	0,41	0,14	0,00
5	100	0,42	0,15	0,00
6	100	0,42	0,15	0,00
7	100	0,42	0,15	0,00
8	-	-	-	-

Taula 9.3. Mesura 2 d'intensitat per canal

I.TOTAL MÀXIMA (A)	I. TOTAL MIG (A)	I.TOTAL MÍNIMA (A)
3,03	1,14	0,00

Taula 9.4. Mesura 2 d'intensitat total

Connectant focus de 100W per canal, es pot dir que el regulador treballa sense cap problema degut a no superar el corrent màxim de línia de 3,15 A. No es poden connectar càrregues de més de 100W per canal amb 7 canals útils, el fusible de línia no ho permetria i es fondria evitant fer malbé les pistes del circuit.

Si es volgués canviar el PIC i que el regulador utilitzés els 8 canals, s'hauria d'augmentar el fusible de línia a 3,5 A tal i com s'estableix als càlculs de l'apartat "*procés de disseny i construcció del circuit imprès*".

D'ambdues mesures es pot dir que regulant la intensitat de llum es pot estalviar energia ja que al baixar aquesta i al no deixar sempre els focus connectats a 230 VAC la intensitat total de la línia i el neutre es redueixen.

9.2. RESULTATS

S'ha aconseguit dissenyar un circuit electrònic, dissenyar una placa de circuit imprès i construir aquesta placa amb el seu corresponent encapsulat amb les següents especificacions:

- Control amb ordinadors que incorporin el sistema operatiu Windows 98 o posteriors.
- Control amb una interfície gràfica intuïtiva i senzilla entre l'usuari i l'ordinador de manera que es pugui regular la intensitat lumínica, els diferents efectes i la programació dels focus.
- Comunicació pel port USB mitjançant cable.
- Gestió de dades rebudes de l'ordinador amb un microcontrolador PIC.
- Regulació d'intensitat lumínica de 0 a 230 VAC amb varis punts de regulació.
- Regulació 7 focus simultàniament independentment l'un de l'altre.
- Regulació de bombetes incandescentes i focus halògens.
- No apte per a bombetes de baix consum
- Potència màxima de 100W per canal.
- Intensitat màxima de 0,42 Amperes per canal.
- Programar en el temps la intensitat lumínica per tal que el regulador s'encengui el dia, l'hora, el minut i el segon que es desitgi.
- Efectes de ràfega i intermitència.
- Habilitació d'un sistema tant a nivell de programa de gestió com de circuit electrònic, per a connectar-hi mòduls de radiofreqüència IKNOS.
- No apte per a la regulació de motors.

10. COST DEL PROJECTE

A continuació es detalla el cost del prototip del projecte segmentat en:

- **Placa de circuit imprès**
- **Encapsulat**
- **Cables**
- **Consumibles**

10.1. PLACA DE CIRCUIT IMPRÈS

COMPONENT	UNITATS	PREU UNITARI (€)	COST (€)
Varistor 230 VAC	1	0,43	0,43
Transformador 230/6VAC 5VA	1	4,18	4,18
Pont de díodes WO6G	1	0,33	0,33
MC78M05CDT	1	0,87	0,87
Díode led verd	1	0,12	0,12
Díode led vermell	2	0,12	0,12
LM2937	1	4,59	4,59
PMLL4448	2	0,05	0,10
BC849C	1	0,06	0,06
CD74AC14M	1	0,47	0,47
SW-098	1	0,30	0,30
MICROINTERRUPTOR DIP2	3	0,84	2,52
Connector USB tipus B	1	0,47	0,47
Cristall de quars 4MHz	1	0,47	0,47
Cristall de quars 12 MHz	1	0,68	0,68
USB-232-DIL	1	7,19	7,19
PIC 16F876 04/SO	1	7,30	7,30
ULN2803A	1	0,83	0,83
TXS0102DCUR	1	0,69	0,69

MODUL RF	1	-	-
Condensador 10uF/50V	3	0,19	0,57
Condensador 470uF/63V	1	0,35	0,35
Condensador 1uF/50V	1	0,13	0,13
Condensador 2.2uF/16V	1	0,13	0,13
Condensador 15nF/50V	3	0,02	0,06
Condensador 100nF/50V	5	0,02	0,10
Condensador 470nF/50V	1	0,02	0,02
Condensador 100nF/400V	8	0,18	1,44
Condensador 15pF/50V	4	0,02	0,08
Resistència 220Ω/¼ W	12	0,02	0,24
Resistència 2.2KΩ/¼ W	1	0,02	0,02
Resistència 22KΩ/¼ W	1	0,02	0,02
Resistència 10kΩ/¼ W	1	0,02	0,02
Resistència 1KΩ/¼ W	1	0,02	0,02
Resistència 470Ω/¼ W	1	0,02	0,02
Resistència 470Ω/4 W	8	0,40	3,20
Resistència 47Ω/4 W	8	0,40	3,20
MOC3021-M	8	0,53	4,24
Diac TMMDB3	8	0,65	5,20
BTA16 700B	8	1,39	11,12
Fusible 3,15 A	7	0,47	3,76
Portafusible	9	0,87	7,83
Bloc terminal 3	1	0,90	0,90
Bloc terminal 2	8	0,54	4,32

Taula 10.1. Cost placa de circuit imprès

10.2. ENCAPSULAT

COMPONENT	UNITATS	PREU UNITARI (€)	COST (€)
Caixa plàstic 228x216x75/50mm	1	15,15	15,15
Cargols mètric 3mm	4	0,07	0,28
Interruptor	1	0,89	0,89

*Taula 10.2. Cost encapsulat***10.3. CONSUMIBLES**

COMPONENT	UNITATS	PREU UNITARI (€)	COST (€)
Líquids metal·litzat (recipient)	1	-	-
Aigüa destilada (recipient)	1	1,89	1,89
Revelador (recipient)	1	6,90	6,90
Fixador (recipient)	1	8,20	8,20
Paper fotogràfic	3	1,20	3,60
Film sensibilitzador negatiu	3	1,08	3,24
Alcohol (recipient)	1	0,83	0,83
Estany (recipient)	1	3,50	3,50
Placa de coure doble cara 200x300	2	9,62	19,24

Taula 10.3. Cost consumibles

10.4. COST TOTAL

APARTATS	COST (€)
PLACA DE CIRCUIT IMPRÈS	86,59
ENCAPSULAT	16,32
CABLES	12,10
CONSUMIBLES	47,40
BASE IMPONIBLE	162,41
IVA (16%)	25,98
TOTAL	188,21

Taula 10.4. Cost total

11. CONCLUSIONS

Amb la realització d'aquest treball final de carrera s'ha aprofundit en coneixements electrònics i informàtics creant un producte apte per al control d'intensitat lumínica amb un funcionament immediat i un cost assequible.

Aquesta memòria, juntament amb els programes subministrats i els fotolits que es troben a l'Annex, permet crear físicament un regulador de llum amb les mateixes característiques o bé amb característiques similars al construït donant opció a modificar els codis dels programes acord amb les necessitats que hom tingui.

El regulador de llum permet un estalvi energètic al poder regular la intensitat dels focus, evitant així tenir-los connectats a una intensitat màxima de 230 VAC.

És un treball que queda obert a possibles modificacions com poden ser la regulació de càrregues de més de 100 W per canal, la connexió de més de 8 focus utilitzant les sortides disponibles del microcontrolador, una programació horària d'intensitat lumínica més exhaustiva, la creació de nous efectes, l'ampliació de les funcions a realitzar pel programa informàtic, evitar que quan es regula un focus la intensitat dels altres baixi mínimament, el correcte control de velocitat de motors i possiblement un llarg etc.

12. BIBLIOGRAFIA

- ANGULO USASTEGUI, José Maria *et al.* *Microcontroladores PIC Diseño práctico de aplicaciones*. Madrid: Mc Graw-Hill, 2000.
- REMIRO, DOMÍNGUEZ, Fernando *et al.* *Lógica digital y microprogramable*. Madrid: Mc Graw-Hill, 1999.
- CHARTE OJEDA, Francisco. *Visual Basic 6*. Madrid: ANAYA, 1999.

Adreces d'Internet:

- Catàleg de Datasheet:
<http://www.alldatasheet.com/> (Consultat periòdicament)
- Visual Basic:
<http://www.lvr.com//> (Consultat periòdicament)
- Funcionament de components electrònics
www.todopic.com.ar (Consultat periòdicament)
www.yoreparo.com (Consultat periòdicament)
www.forosdeelectronica.com (Consultat periòdicament)
- Comanda components electrònics:
www.es.farnell.com (Consultat periòdicament)
www.mouser.com (Consultat periòdicament)

A. ANNEX

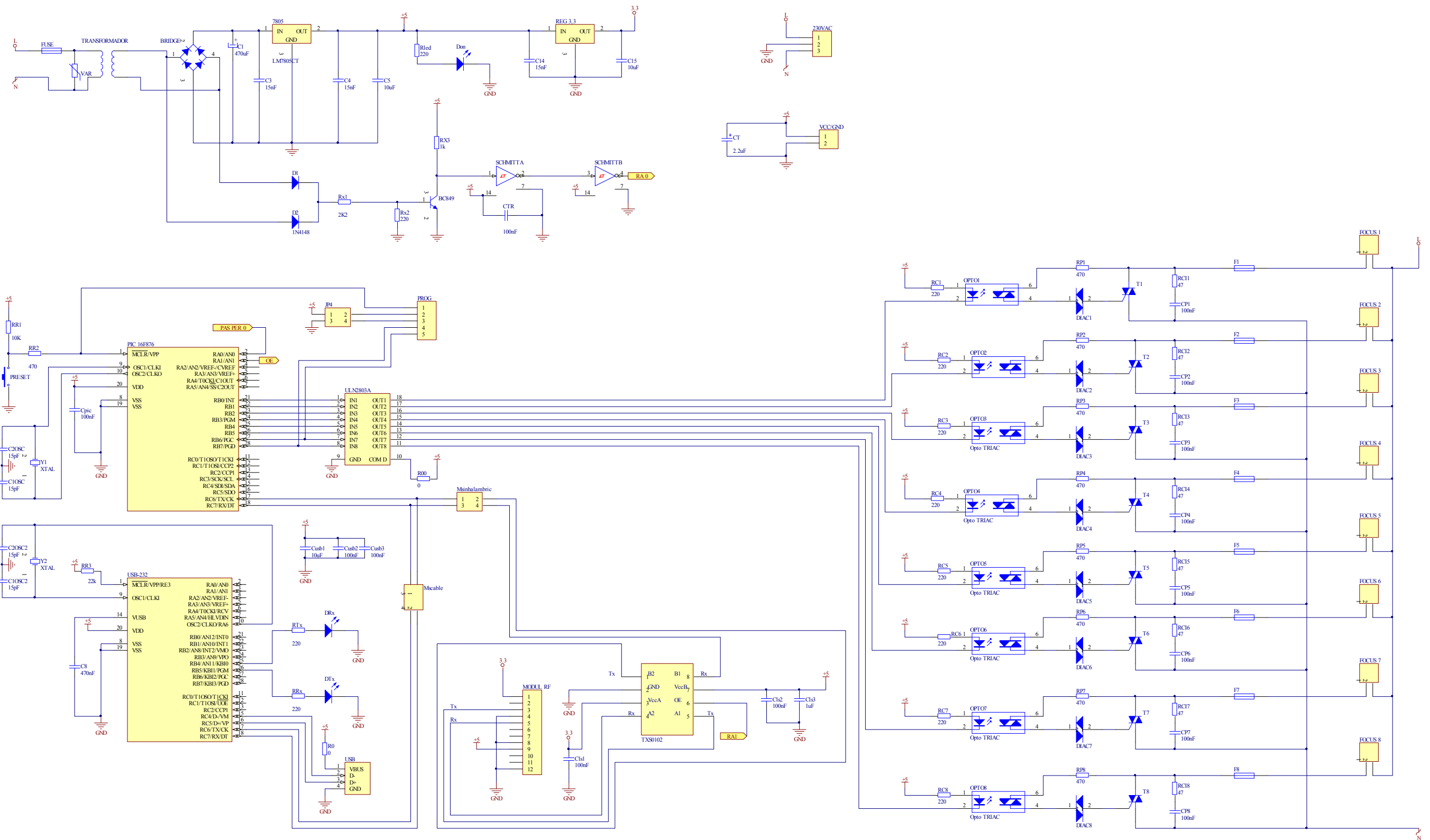
A.1. DVD ADJUNT

Juntament amb la memòria del Treball Final de Carrera es fa l'entrega d'un DVD on s'hi pot trobar el següent:

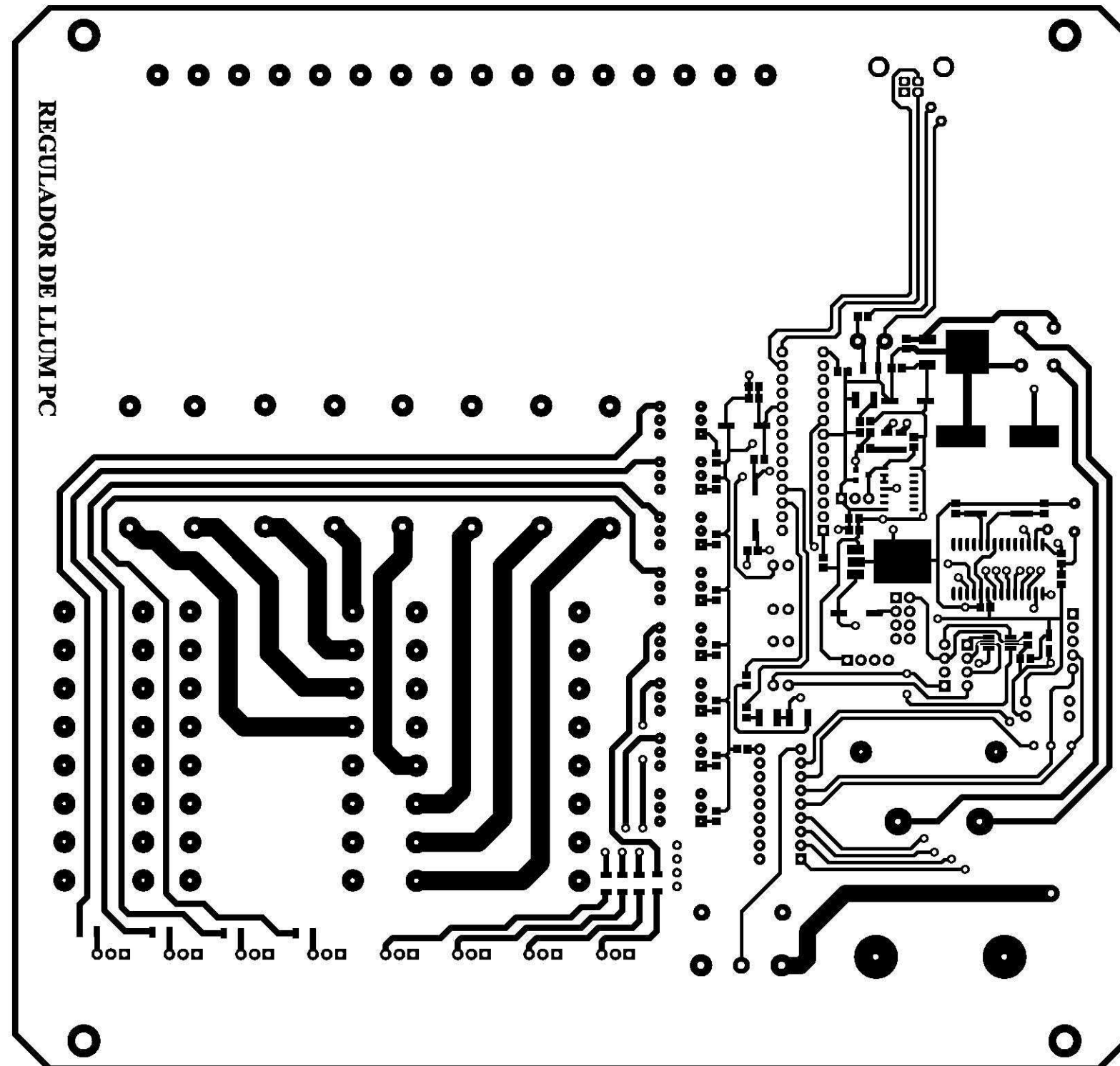
- **DVD REGULADOR DE LLUM CONTROLAT PER ORDINADOR**
 - **TFC REGULADOR DE LLUM CONTROLAT PER ORDINADOR**
 - **PROGRAMA “REGULADOR DE LLUM PC”**
 - **ANNEX:**
 - Codi:
 - Codi Visual Basic
 - Codi microcontrolador PIC16F876
 - Disseny i construcció de PCB:
 - Regulador de llum Altium Designer
 - NC Drill
 - Gerbers
 - Datasheets

A.2. PLÀNOLS

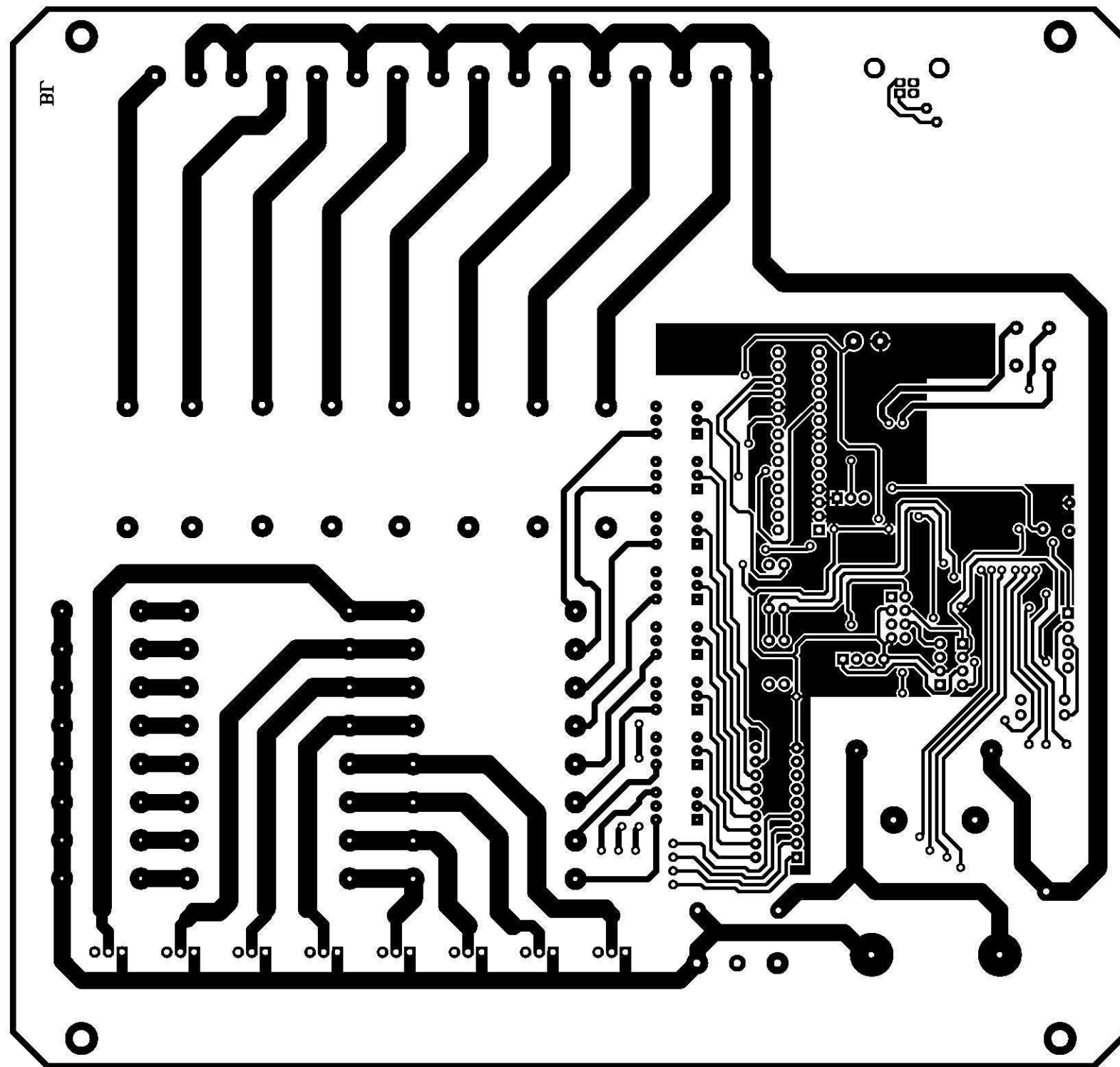
- **Esquema unifilar**
- **Disseny PCB cara superior**
- **Disseny PCB cara inferior**
- **Situació de components**



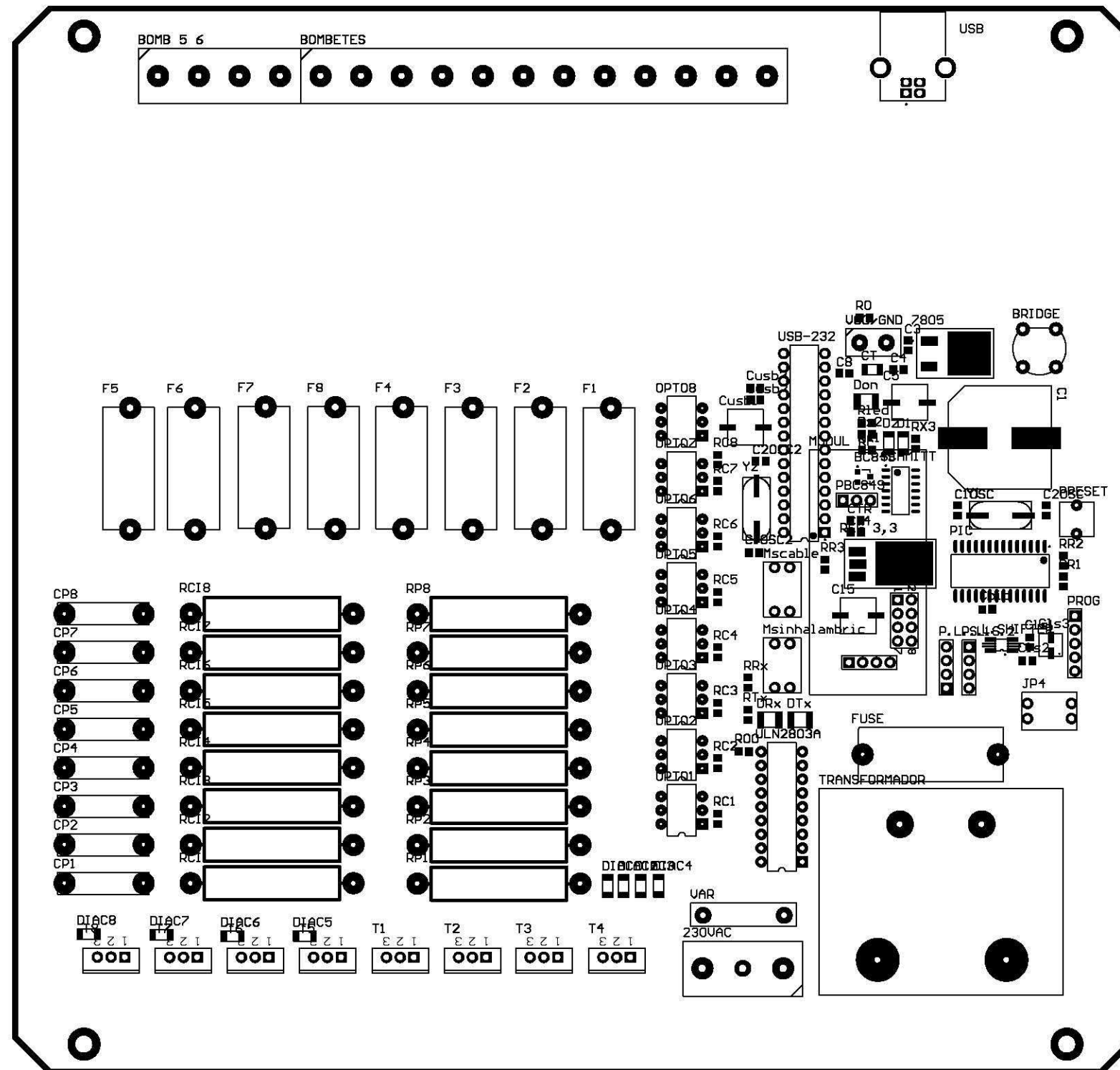
	Dibuixat:	D. Martin	Data:	03/07/2010	Plànol n°1
	Comprovat:	A. Surinyach	Data:	03/07/2010	
Escala: 1:1	ESQUEMA UNIFILAR				UVIC UNIVERSITAT DE VIC



	Dibuixat:	D. Martin	Data:	03/07/2010	Plànol n°2
	Comprovat:	A. Surinyach	Data:	03/07/2010	
Escala: 1:1	DISSENY PCB CARA SUPERIOR				UVIC UNIVERSITAT DE VIC

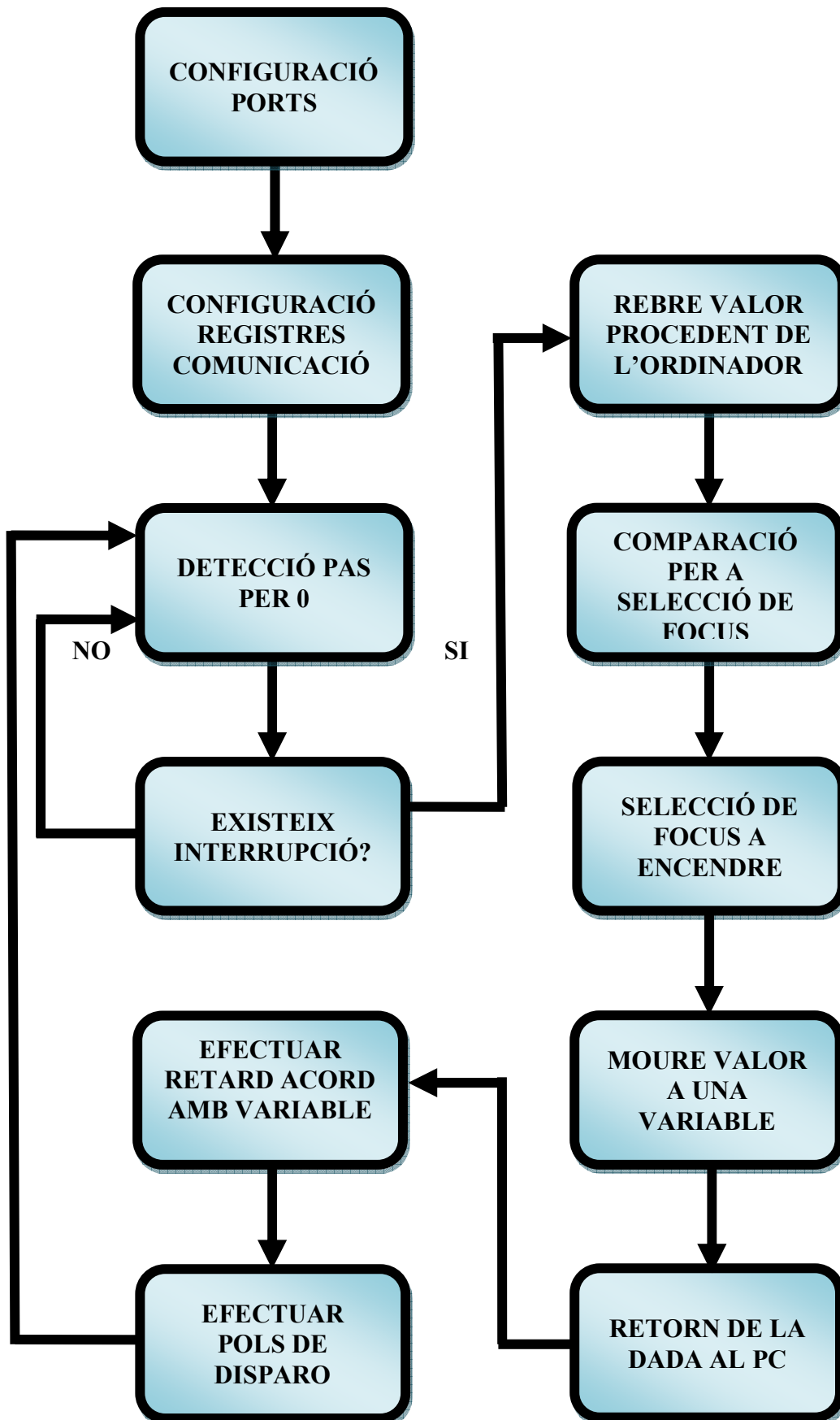


	Dibuixat:	D. Martin	Data:	03/07/2010	Plànol n°3
	Comprovat:	A. Surinyach	Data:	03/07/2010	
Escala: 1:1	DISSENY PCB CARA INFERIOR				UVIC UNIVERSITAT DE VIC



	Dibuixat:	D. Martin	Data:	03/07/2010	Plànol nº4
	Comprovat:	A. Surinyach	Data:	03/07/2010	
Escala: 1:1	SITUACIÓ DE COMPONENTS				

A.3. DIAGRAMA DE BLOCS CODI PIC



A.4. CODI MICROCONTROLADOR PIC16F876

```
;***** INICIALITZACIONS *****  
  
;TIPUS DE MICROCONTROLADOR  
  
list p=16F876           ;Tipus de processador.  
include "P16F876.INC"  ;Definicions de registres interns.  
  
;DEFINICIONS  
;*****  
;#DEFINE  
  
;VARIABLES I CONSTANTS  
  
registre equ 0x20  
registre1 equ 0x21  
dadapc equ 0x22  
  
valorpolsf1 equ 0x23  
valorretardf1 equ 0x24  
retardf1 equ 0x25  
polsf1 equ 0x26  
  
valorpolsf2 equ 0x27  
valorretardf2 equ 0x28  
retardf2 equ 0x29  
polsf2 equ 0x30  
  
valorpolsf3 equ 0x31  
valorretardf3 equ 0x32  
retardf3 equ 0x33  
polsf3 equ 0x34  
  
valorpolsf4 equ 0x35  
valorretardf4 equ 0x36  
retardf4 equ 0x37  
polsf4 equ 0x38
```

```

valorpolsf5 equ 0x39
valorretardf5 equ 0x40
retardf5 equ 0x41
polsf5 equ 0x42

```

```

valorpolsf6 equ 0x43
valorretardf6 equ 0x44
retardf6 equ 0x45
polsf6 equ 0x46

```

```

valorpolsf7 equ 0x47
valorretardf7 equ 0x48
retardf7 equ 0x49
polsf7 equ 0x50

```

```
registre2 equ 0x51
```

```

valorpolsf8 equ 0x52
valorretardf8 equ 0x53
retardf8 equ 0x54
polsf8 equ 0x55

```

```
;***** PROGRAMA *****
```

```

org 0x00
goto inici

```

```

org 0x04
goto inter

```

```
;***** RETORN DE LA DADA AL PC *****
```

```

TX_DATO      bcf    PIR1,TXIF
              movwf  TXREG
              bsf    STATUS,RP0
              bcf    STATUS,RP1

```

```

TX_DAT_W     btfs   TXSTA,TRMT
              goto  TX_DAT_W
              bcf    STATUS,RP0
              return

```

;***** INTERRUPTIÓ *****

```
inter          btfss  PIR1,RCIF
               goto   VOLVER
               bcf    PIR1,RCIF
               movf   RCREG,W
               bcf    STATUS,0
               movwf  dadapc
```

;***** FOCUS *****

```
comp          movlw  .26
               subwf  dadapc,0
               btfss  STATUS,C
               goto   focus
               movlw  .58
               subwf  dadapc,0
               btfss  STATUS,C
               goto   focus2
```

```
               movlw  .90
               subwf  dadapc,0
               btfss  STATUS,C
               goto   focus3
```

```
               movlw  .121
               subwf  dadapc,0
               btfss  STATUS,C
               goto   focus4
```

```
               movlw  .153
               subwf  dadapc,0
               btfss  STATUS,C
               goto   focus5
```

```
               movlw  .185
               subwf  dadapc,0
               btfss  STATUS,C
```

```
goto focus6

movlw .216
subwf dadapc,0
btfss STATUS,C
goto focus7

;movlw .248
;subwf dadapc,0
;btfss STATUS,C
;goto focus8

VOLVER      retfie

focus1     movf dadapc,0
            movwf valorretardf1
            rlf valorretardf1,1
            rlf valorretardf1,1
            rlf valorretardf1,1
            call TX_DATO
            retfie

focus2     movf dadapc,0
            movwf valorretardf2
            rlf valorretardf2,1
            rlf valorretardf2,1
            rlf valorretardf2,1
            call TX_DATO
            retfie

focus3     movf dadapc,0
            movwf valorretardf3
            rlf valorretardf3,1
            rlf valorretardf3,1
            rlf valorretardf3,1
            call TX_DATO
            retfie
```

```
focus4      movf dadapc,0
             movwf valorretardf4
             rlf valorretardf4,1
             rlf valorretardf4,1
             rlf valorretardf4,1
             call TX_DATO
             retfie

focus5      movf dadapc,0
             movwf valorretardf5
             rlf valorretardf5,1
             rlf valorretardf5,1
             rlf valorretardf5,1
             call TX_DATO
             retfie

focus6      movf dadapc,0
             movwf valorretardf6
             rlf valorretardf6,1
             rlf valorretardf6,1
             rlf valorretardf6,1
             call TX_DATO
             retfie

focus7      movf dadapc,0
             movwf valorretardf7
             rlf valorretardf7,1
             rlf valorretardf7,1
             rlf valorretardf7,1
             call TX_DATO
             retfie

;focus8     movf dadapc,0
             ;movwf valorretardf8
             ;rlf valorretardf8,1
             ;rlf valorretardf8,1
             ;rlf valorretardf8,1
             ;call TX_DATO
             ;retfie
```

***** PROGRAMA PRINCIPAL *****

***** PORT A COM A ENTRADA *****

```
inici          bcf STATUS,RP0
               bcf STATUS,RP1
               clrf PORTA
               bsf STATUS,RP0
               movlw 0x06
               movwf ADCON1
               movlw b'11111111'
               movwf TRISA
               bcf STATUS,RP0
```

***** PORTB I PORTC *****

```
               clrwdt
               clrf  PORTB
               clrf  PORTC
               bsf  STATUS,RP0
               bcf  STATUS,RP1
               clrf  TRISB
               movlw b'10111111'
               movwf TRISC
```

*****CONFIGURACIÓ DE REGISTRES PER A LA COMUNICACIÓ SÈRIE***

```
               movlw      b'11101111'
               movwf      OPTION_REG
               movlw b'00100100'
               movwf      TXSTA
               movlw .25
               movwf      SPBRG
               bsf PIE1,RCIE ;habilita la interrupció per recepció
               bcf  STATUS,RP0
               movlw      b'10010000'
               movwf      RCSTA
               movlw b'11000000'
               movwf      INTCON
```

;***** INICIALITZACIÓ DE POLS I RETARD *****

```
movlw .5
movwf valorpolsf1
movwf valorpolsf2
movwf valorpolsf3
movwf valorpolsf4
movwf valorpolsf5
movwf valorpolsf6
movwf valorpolsf7
;movwf valorpolsf8
```

```
movlw .20
```

```
movwf valorretardf1
movwf valorretardf2
movwf valorretardf3
movwf valorretardf4
movwf valorretardf5
movwf valorretardf6
movwf valorretardf7
;movwf valorretardf8
```

;***** INICI PROGRAMA *****

```
comprovar          btfsc registre,0
                   call decrementpols1

                   btfsc registre,1
                   call decrementretard1

                   btfsc registre,3
                   call decrementpols2

                   btfsc registre,4
                   call decrementretard2

                   btfsc registre,5
                   call decrementpols3
```

bfsc registre,6
call decrementretard3

bfsc registre1,0
call decrementpols4

bfsc registre1,1
call decrementretard4

bfsc registre1,3
call decrementpols5

bfsc registre1,4
call decrementretard5

bfsc registre1,5
call decrementpols6

bfsc registre1,6
call decrementretard6

bfsc registre2,0
call decrementpols7

bfsc registre2,1
call decrementretard7

;bfsc registre2,3
;call decrementpols8

;bfsc registre2,4
;call decrementretard8

***** PAS PER 0 *****

bfsc PORTA,0
call flancup
btfss PORTA,0
call flandown

goto comprovar

;***** FLANC DE PUJADA O FLANC DE BAIXADA *****

lancup btfsc registre,2
 return
 bsf registre,2
 call canviflanc
 return

flancdown btfss registre,2
 return
 bcf registre,2
 call canviflanc
 return

canviflanc bsf registre,1
 movf valorretardf1,0
 movwf retardf1

 bsf registre,4
 movf valorretardf2,0
 movwf retardf2

 bsf registre,6
 movf valorretardf3,0
 movwf retardf3

 bsf registre1,1
 movf valorretardf4,0
 movwf retardf4

 bsf registre1,4
 movf valorretardf5,0
 movwf retardf5

 bsf registre1,6
 movf valorretardf6,0
 movwf retardf6

```
bsf registre2,1
movf valorretardf7,0
movwf retardf7
```

```
;bsf registre2,4
;movf valorretardf8,0
;movwf retardf8
```

```
return
```

```
;***** TEMPS DE POLS *****
```

```
decrementpols1
```

```
bsf PORTB,7
decfsz polsf1,1
return
bcf PORTB,7
movf valorpolsf1,0
movwf polsf1
bcf registre,0
return
```

```
;***** RETARD *****
```

```
decrementretard1
```

```
decfsz retardf1,1
return
movf valorpolsf1,0
movwf polsf1
bcf registre,1
bsf registre,0
return
```

```
;***** TEMPS DE POLS *****
```

```
decrementpols2
```

```
bsf PORTB,6
decfsz polsf2,1
return
bcf PORTB,6
movf valorpolsf2,0
movwf polsf2
bcf registre,3
```

```
return
```

```
;*****  
; RETARD *****
```

```
decrementretard2          decfsz retardf2,1  
                           return  
                           movf valorpolsf2,0  
                           movwf polsf2  
                           bcf registre,4  
                           bsf registre,3  
                           return
```

```
;*****  
; TEMPS DE POLS *****
```

```
decrementpols3           bsf PORTB,5  
                           decfsz polsf3,1  
                           return  
                           bcf PORTB,5  
                           movf valorpolsf3,0  
                           movwf polsf3  
                           bcf registre,5  
                           return
```

```
;*****  
; RETARD *****
```

```
decrementretard3          decfsz retardf3,1  
                           return  
                           movf valorpolsf3,0  
                           movwf polsf3  
                           bcf registre,6  
                           bsf registre,5  
                           return
```

```
;*****  
; TEMPS DE POLS *****
```

```
decrementpols4           bsf PORTB,4  
                           decfsz polsf4,1
```

```
return
bcf PORTB,4
movf valorpolsf4,0
movwf polsf4
bcf registre1,0
return
```

```
;***** RETARD *****
```

```
decrementretard4      decfsz retardf4,1
return
movf valorpolsf4,0
movwf polsf4
bcf registre1,1
bsf registre1,0
return
```

```
;***** TEMPS DE POLS *****
```

```
decrementpols5      bsf PORTB,0
decfsz polsf5,1
return
bcf PORTB,0
movf valorpolsf5,0
movwf polsf5
bcf registre1,3
return
```

;***** RETARD *****

```
decrementretard5          decfsz retardf5,1
                           return
                           movf valorpolsf5,0
                           movwf polsf5
                           bcf registre1,4
                           bsf registre1,3
                           return
```

;***** TEMPS DE POLS *****

```
decrementpols6           bsf PORTB,1
                          decfsz polsf6,1
                          return
                          bcf PORTB,1
                          movf valorpolsf6,0
                          movwf polsf6
                          bcf registre1,5
                          return
```

;***** RETARD *****

```
decrementretard6        decfsz retardf6,1
                          return
                          movf valorpolsf6,0
                          movwf polsf6
                          bcf registre1,6
                          bsf registre1,5
                          return
```

;***** TEMPS DE POLS *****

```
decrementpols7          bsf PORTB,2
                          decfsz polsf7,1
                          return
                          bcf PORTB,2
```

```

                                movf valorpolsf7,0
                                movwf polsf7
                                bcf registre2,0
                                return

;***** RETARD *****
decrementretard7                decfsz retardf7,1
                                return
                                movf valorpolsf7,0
                                movwf polsf7
                                bcf registre2,1
                                bsf registre2,0
                                return

;***** TEMPS DE POLS *****
;decrementpols8                bsf PORTB,0
;                                decfsz polsf8,1
;                                return
;                                bcf PORTB,0
;                                movf valorpolsf8,0
;                                movwf polsf8
;                                bcf registre2,3
;                                return

;***** RETARD *****
;decrementretard8              decfsz retardf8,1
;                                return
;                                movf valorpolsf8,0
;                                movwf polsf8
;                                bcf registre2,4
;                                bsf registre2,3
;                                return
;
end

```

A.5. CODI VISUAL BASIC

```
Option Strict On
```

```
Option Explicit On
```

```
Imports GenericHid.DeviceManagement
```

```
Imports GenericHid.FileIo
```

```
Imports GenericHid.Hid
```

```
Imports Microsoft.Win32.SafeHandles
```

```
Imports System.Runtime.InteropServices
```

```
Friend Class FrmMain
```

```
Inherits System.Windows.Forms.Form
```

```
Private deviceNotificationHandle As IntPtr
```

```
Private exclusiveAccess As Boolean
```

```
Private hidHandle As SafeFileHandle
```

```
Private hidUsage As String
```

```
Private myDeviceDetected As Boolean
```

```
Private myDevicePathName As String
```

```
Private readHandle As SafeFileHandle
```

```
Private writeHandle As SafeFileHandle
```

```
Private MyDebugging As New Debugging() ' For viewing results of
API calls via Debug.Write.
```

```
Private MyDeviceManagement As New DeviceManagement()
```

```
Private MyHid As New Hid()
```

```
'variables publiques per tal que les puguin utilitzar totes les
funcions''
```

```
Public outputReportBuffer() As Byte
```

```
Public inputReportBuffer() As Byte
```

```
Public byteValue As String
```

```
Public count As Integer
```

```
Public success As Boolean
```

```
Public barra As Integer
```

```
Public Dadaenviar As Byte
```

```
.....
```

```
'variables per saber en quina funció del timer estic
```

```
Public a As Boolean
```

```
Public b As Boolean
```

```
Public c As Boolean
```

```
Public d As Boolean
```

```
Public f As Boolean
```

```
Public g As Boolean
```

```
Public h As Boolean
```

```

Public i As Boolean

Public T1 As Boolean
Public T2 As Boolean
Public T3 As Boolean
Public T4 As Boolean
Public T5 As Boolean
Public T6 As Boolean
Public T7 As Boolean

Friend FrmMy As FrmMain

''' <summary>
''' Define a class of delegates that point to the
Hid.ReportIn.Read function.
''' The delegate has the same parameters as Hid.ReportIn.Read.
''' Used for asynchronous reads from the device.
''' </summary>

Private Delegate Sub ReadInputReportDelegate _
    (ByVal hidHandle As SafeFileHandle, _
    ByVal readHandle As SafeFileHandle, _
    ByVal writeHandle As SafeFileHandle, _
    ByRef myDeviceDetected As Boolean, _
    ByRef readBuffer() As Byte, _
    ByRef success As Boolean)

' This delegate has the same parameters as AccessForm.
' Used in accessing the application's form from a different
thread.

Private Delegate Sub MarshalToForm _
    (ByVal action As String, _
    ByVal textToAdd As String)

''' <summary>
''' Called when a WM_DEVICECHANGE message has arrived,
''' indicating that a device has been attached or removed.
''' </summary>
'''
''' <param name="m"> a message with information about the device
</param>

Friend Sub OnDeviceChange(ByVal m As Message)

    Debug.WriteLine("WM_DEVICECHANGE")

    Try
        If (m.WParam.ToInt32 = DBT_DEVICEARRIVAL) Then

```



```

' If WParam contains DBT_DEVICEARRIVAL, a device has been attached.

        Debug.WriteLine("A device has been attached.")

        ' Find out if it's the device we're communicating
with.

        If MyDeviceManagement.DeviceNameMatch(m,
myDevicePathName) Then
            lstResults.Items.Add("My device attached.")
        End If
        FindTheHid()

    ElseIf (m.WParam.ToInt32 = DBT_DEVICEREMOVECOMPLETE) Then

        ' If WParam contains DBT_DEVICEREMOVAL, a device has
been removed.

        Debug.WriteLine("A device has been removed.")

        ' Find out if it's the device we're communicating
with.

        If MyDeviceManagement.DeviceNameMatch(m,
myDevicePathName) Then

            lstResults.Items.Add("Dispositiu desconnectat.")

            parartot()

            ' Set MyDeviceDetected False so on the next data-`
`transfer attempt,
            ' FindTheHid() will be called to look for the
`device
            ' and get a new handle.

            FrmMy.myDeviceDetected = False

            'DESHABILITAR ELS CONTROLS AL DESCONNECTAR EL DISPOSITIU

        End If
    End If

    ScrollToBottomOfListBox()

Catch ex As Exception
    DisplayException(Me.Name, ex)
Throw

```

```

    End Try
End Sub

''' <summary>
''' Uses a series of API calls to locate a HID-class device
''' by its Vendor ID and Product ID.
''' </summary>
'''
''' <returns>
''' True if the device is detected, False if not detected.
''' </returns>

Private Function FindTheHid() As Boolean

    'Funció que connecta amb el dispositiu USB-232

    Dim deviceFound As Boolean
    Dim devicePathName(127) As String
    Dim hidGuid As System.Guid
    Dim memberIndex As Int32
    Dim myProductID As Int16
    Dim myVendorID As Int16
    Dim success As Boolean

    Try
        myDeviceDetected = False

        ' Crida a la següent funció i agafa el Vendor ID i el
        'Product ID dels quadres de text
        '(myVendorID, myProductID), escrivint això li diem que ens
passi aquests dos valors

        GetVendorAndProductIDsFromTextBoxes(myVendorID,
myProductID)

        ' ***
        ' API function: 'HidD_GetHidGuid

        ' Purpose: Retrieves the interface class GUID for the HID
class.

        ' Accepts: 'A System.Guid object for storing the GUID.
        ' ***

        HidD_GetHidGuid(hidGuid)

        Debug.WriteLine(MyDebugging.ResultOfAPICall("GetHidGuid"))
        Debug.WriteLine("  GUID for system HIDs: " & hidGuid.ToString)
    
```

```
' Fill an array with the device path names of all attached
HIDs.
```

```
deviceFound = MyDeviceManagement.FindDeviceFromGuid _
    (hidGuid, _
    devicePathName)
```

```
' If there is at least one HID, attempt to read the Vendor
`ID and Product ID
' of each device until there is a match or all devices
`have been examined.
```

```
If deviceFound Then
```

```
    memberIndex = 0
```

```
    Do
```

```
        ' ***
        ' API function:
        ' CreateFile

        ' Purpose:
        ' Retrieves a handle to a device.

        ' Accepts:
        ' A device path name returned by
SetupDiGetDeviceInterfaceDetail
        ' The type of access requested (read/write).
        ' FILE_SHARE attributes to allow other processes
to access the device while this handle is open.
        ' A Security structure or IntPtr.Zero.
        ' A creation disposition value. Use OPEN_EXISTING
for devices.

        ' Flags and attributes for files. Not used for
devices.

        ' Handle to a template file. Not used.

        ' Returns: a handle without read or write access.
        ' This enables obtaining information about all
HIDs, even system

        ' keyboards and mice.
        ' Separate handles are used for reading and
writing.

        ' ***
```

```
    hidHandle = CreateFile _
        (devicePathName(memberIndex), _
        0, _
        FILE_SHARE_READ Or FILE_SHARE_WRITE, _
```

```

        IntPtr.Zero, _
        OPEN_EXISTING, _
        0, _
        0)

Debug.WriteLine(MyDebugging.ResultOfAPICall("CreateFile"))
        Debug.WriteLine("  Returned handle: " &
hidHandle.ToString)

        If Not (hidHandle.IsInvalid) Then

            ' The returned handle is valid,
            ' so find out if this is the device we're
looking for.

            ' Set the Size property of DeviceAttributes to
the number of bytes in the structure.

            MyHid.DeviceAttributes.Size =
Marshal.SizeOf(MyHid.DeviceAttributes)

            ' ***
            ' API function:
            ' HidD_GetAttributes

            ' Purpose:
            ' Retrieves a HIDD_ATTRIBUTES structure
containing the Vendor ID,
            ' Product ID, and Product Version Number for a
device.

            ' Accepts:
            ' A handle returned by CreateFile.
            ' A pointer to receive a HIDD_ATTRIBUTES
structure.

            ' Returns:
            ' True on success, False on failure.
            ' ***

            success = HidD_GetAttributes(hidHandle,
MyHid.DeviceAttributes)

            If success Then

                Debug.WriteLine("  HIDD_ATTRIBUTES
structure filled without error.")
                Debug.WriteLine("  Structure size: " &
MyHid.DeviceAttributes.Size)

```

```

        Debug.WriteLine(" Vendor ID: " &
Hex(MyHid.DeviceAttributes.VendorID))
        Debug.WriteLine(" Product ID: " &
Hex(MyHid.DeviceAttributes.ProductID))
        Debug.WriteLine(" Version Number: " &
Hex(MyHid.DeviceAttributes.VersionNumber))

' Find out if the device matches the one
we're looking for.

If (MyHid.DeviceAttributes.VendorID =
myVendorID) And _
(MyHid.DeviceAttributes.ProductID =
myProductID) Then

    Debug.WriteLine(" Dispositiu
connectat ")

    ' Display the information in form's
list box.

    lstResults.Items.Add("Dispositiu
connectat:")

    lstResults.Items.Add(" Vendor ID= " &
Hex(MyHid.DeviceAttributes.VendorID))
    lstResults.Items.Add(" Product ID = "
& Hex(MyHid.DeviceAttributes.ProductID))

    Label6.Visible = True
    Label6.ForeColor = Color.LawnGreen
    Label6.Text = ("DISPOSITIU CONNECTAT")
    ScrollToBottomOfListBox()

    myDeviceDetected = True

'HABILITAR ELS CONTROLS AL CONNECTAR EL
DISPOSITIU

    GroupBox4.Enabled = True
    GroupBox3.Enabled = True
    GroupBox6.Enabled = True
    GroupBox5.Enabled = True
    programar.Enabled = True

' Save the DevicePathName for
OnDeviceChange().

myDevicePathName = devicePathName(memberIndex)

```

```
Else

    ' It's not a match, so close the
    handle.

    myDeviceDetected = False

    hidHandle.Close()

End If

Else

    ' There was a problem in retrieving the
    information.

    Debug.WriteLine(" Error in filling
    HIDD_ATTRIBUTES structure.")
    myDeviceDetected = False
    hidHandle.Close()
End If

End If

' Keep looking until we find the device or there
are no devices left to examine.

memberIndex = memberIndex + 1

Loop Until (myDeviceDetected Or (memberIndex =
devicePathName.Length))

End If

If myDeviceDetected Then

    ' The device was detected.
    ' Register to receive notifications if the device is
    ttached.

    success =
MyDeviceManagement.RegisterForDeviceNotifications _
    (myDevicePathName, _
    FrmMy.Handle, _
    hidGuid, _
    deviceNotificationHandle)

    Debug.WriteLine("RegisterForDeviceNotifications = " &
success)
```

```
' Learn the capabilities of the device.

    MyHid.Capabilities =
MyHid.GetDeviceCapabilities(hidHandle)

    If success Then

        ' Find out if the device is a system mouse or
keyboard.

        hidUsage = MyHid.GetHidUsage(MyHid.Capabilities)

        ' Get the Input report buffer size.

        GetInputReportBufferSize()
        cmdInputReportBufferSize.Enabled = True

        ' Get handles to use in requesting Input and
Output reports.

        readHandle = CreateFile _
            (myDevicePathName, _
            GENERIC_READ, _
            FILE_SHARE_READ Or FILE_SHARE_WRITE, _
            IntPtr.Zero, _
            OPEN_EXISTING, _
            FILE_FLAG_OVERLAPPED, _
            0)

        Debug.WriteLine(MyDebugging.ResultOfAPICall("CreateFile, ReadHandle"))
        Debug.WriteLine(" Returned handle: " &
readHandle.ToString)

        If readHandle.IsInvalid Then

            exclusiveAccess = True
            lstResults.Items.Add("The device is a system "
+ hidUsage + ".")

            lstResults.Items.Add("Windows 2000 and Windows
XP obtain exclusive access to Input and Output reports for this
devices.")

            lstResults.Items.Add("Applications can access
Feature reports only.")
            ScrollToBottomOfListBox()

        Else

            writeHandle = CreateFile _
                (myDevicePathName, _
```

```

        GENERIC_WRITE, _
        FILE_SHARE_READ Or FILE_SHARE_WRITE, _
        IntPtr.Zero, _
        OPEN_EXISTING, _
        0, _
        0)

Debug.WriteLine(MyDebugging.ResultOfAPICall("CreateFile,
WriteHandle"))
        Debug.WriteLine("  Returned handle: " &
writeHandle.ToString)

        ' Flush any waiting reports in the input
buffer. (optional)

        MyHid.FlushQueue(readHandle)

        End If
    End If
Else
    ' The device wasn't detected.

    lstResults.Items.Add("No s'ha trobat cap dispositiu.")
    cmdInputReportBufferSize.Enabled = False
    'cmdOnce.Enabled = True

    Debug.WriteLine(" Device not found.")

    ScrollToBottomOfListBox()
End If

Return myDeviceDetected

Catch ex As Exception
    DisplayException(Me.Name, ex)
    Throw
End Try
End Function

''' <summary>
''' In asynchronous ReadFiles, the callback function
GetInputReportData
''' uses this routine to access the application's form, which runs
in
''' a different thread.
''' The routine performs various application-specific functions
that
''' involve accessing the application's form.

```



```
''' </summary>
'''
''' <param name="action"> a string that names the action to
perform on the form</param>
''' <param name="formText"> text that the form displays or the
code uses for
''' another purpose. Actions that don't use text ignore this
parameter. </param>

Private Sub AccessForm(ByVal action As String, ByVal formText As
String)

Try
    ' Select an action to perform on the form:

    Select Case action

        Case "AddItemToListBox"

            'lstResults.Items.Add(formText)

        Case "AddItemToTextBox"

            'txtBytesReceived.SelectedText = formText & vbCrLf

        Case "EnableCmdOnce"

            ' If it's a single transfer, re-enable the command
button.

        Case "ScrollToBottomOfListBox"

            'lstResults.SelectedIndex = lstResults.Items.Count

- 1

        Case "TextBoxSelectionStart"

            'txtBytesReceived.SelectionStart = Len(formText)

        Case Else

    End Select

Catch ex As Exception
    DisplayException(Me.Name, ex)
    Throw
End Try
```

```
End Sub

''' <summary>
''' Start or stop a series of periodic transfers.
''' </summary>

Private Sub cmdInputReportBufferSize_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmdInputReportBufferSize.Click

    Try
        SetInputReportBufferSize()

    Catch ex As Exception
        DisplayException(Me.Name, ex)
        Throw
    End Try
End Sub

''' <summary>
''' Search for a specific device.
''' </summary>

Private Sub cmdFindDevice_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cmdFindDevice.Click

    Try
        FindTheHid()

    Catch ex As Exception
        DisplayException(Me.Name, ex)
        Throw
    End Try

End Sub

''' <summary>
''' Attempt to write a report and read a report.
''' </summary>
'''

Private Sub DeviceHasChanged()

    Try
        ' If a device was previously detected, stop receiving
        notifications about it.
```

```
        If myDeviceDetected Then

MyDeviceManagement.StopReceivingDeviceNotifications(deviceNotification
Handle)

        End If

        ' Search for the device the next time FindTheHid is
called.

        myDeviceDetected = False

    Catch ex As Exception
        DisplayException(Me.Name, ex)
        Throw
    End Try
End Sub

''' <summary>
''' Sends a Feature report, then retrieves one.
''' Assumes report ID = 0 for both reports.
''' </summary>

Private Sub frmMain_Closed(ByVal eventSender As System.Object,
ByVal eventArgs As System.EventArgs) Handles MyBase.Closed

    Try
        Shutdown()

    Catch ex As Exception
        DisplayException(Me.Name, ex)
        Throw
    End Try
End Sub

''' <summary>
''' Perform startup operations.
''' </summary>

Private Sub frmMain_Load(ByVal eventSender As System.Object, ByVal
eventArgs As System.EventArgs) Handles MyBase.Load

    Try
        FrmMy = Me

        Startup()

        FindTheHid()
        If myDeviceDetected = False Then
            Label6.Text = "DISPOSITIU DESCONNECTAT"
```

```
        parartot()
    End If

    'Posem les barres a baix de tot (mínima intensitat)
    barra1.Value = barra1.Maximum
    barra2.Value = barra2.Maximum
    barra3.Value = barra3.Maximum
    barra4.Value = barra4.Maximum
    barra5.Value = barra5.Maximum
    barra6.Value = barra6.Maximum
    barra7.Value = barra7.Maximum
    barra8.Value = barra8.Maximum

    'els timers s'habiliten i van contant amb l'interval triat
    'fins que no el desabilites.

    hora.Enabled = True 'habilita que es vegi l'hora al labell

    'Si el dispositiu està desconnectat s'inicia un timer cada
4 segons
    'que quan es desborda mira si s'ha connectat el dispositiu
i si es que si,
    'es para el timer.

    If myDeviceDetected = False Then

        Timer6.Interval = 4000
        Timer6.Enabled = True

    End If

    'Dadaenviar = 25 'per enviar una dada pel port USB s'ha de
convertir a byte
    'enviarirebre() 'es crida a la funció enviar i rebre
    'Dadaenviar = 57 'per enviar una ada pel port USB s'ha de
convertir a byte
    'enviarirebre() 'es crida a la funció enviar i rebre
    'Dadaenviar = 89 'per enviar una dada pel port USB s'ha de
convertir a byte
    'enviarirebre() 'es crida a la funció enviar i rebre
    'Dadaenviar = 120 'per enviar una dada pel port USB s'ha
de convertir a byte
    'enviarirebre() 'es crida a la funció enviar i rebre
```

```
        'Dadaenviar = 151 'per enviar una dada pel port USB s'ha
de convertir a byte
        'enviarirebre() 'es crida a la funció enviar i rebre
        'Dadaenviar = 183 'per enviar una dada pel port USB s'ha
de convertir a byte
        'enviarirebre() 'es crida a la funció enviar i rebre
        'Dadaenviar = 215 'per enviar una dada pel port USB s'ha
de convertir a byte
        'enviarirebre() 'es crida a la funció enviar i rebre
        'Dadaenviar = 216 'per enviar una dada pel port USB s'ha
de convertir a byte
        'enviarirebre() 'es crida a la funció enviar i rebre
```

```
    Catch ex As Exception
        DisplayException(Me.Name, ex)
        Throw
    End Try
```

```
End Sub
```

```
''' <summary>
''' Finds and displays the number of Input buffers
''' (the number of Input reports the host will store).
''' </summary>
```

```
Private Sub GetInputReportBufferSize()
```

```
    Dim numberOfInputBuffers As Int32
```

```
    Try
```

```
        ' Get the number of input buffers.
```

```
        MyHid.GetNumberOfInputBuffers _
            (hidHandle, _
            numberOfInputBuffers)
```

```
        ' Display the result in the text box.
```

```
        txtInputReportBufferSize.Text = CStr(numberOfInputBuffers)
```

```
    Catch ex As Exception
        DisplayException(Me.Name, ex)
        Throw
    End Try
```

```

End Sub

''' <summary>
''' Retrieves Input report data and status information.
''' This routine is called automatically when myInputReport.Read
''' returns. Calls several marshaling routines to access the main
form.
''' </summary>
'''
''' <param name="ar"> an object containing status information
about
''' the asynchronous operation. </param>

Private Sub GetInputReportData(ByVal ar As IAsyncResult)

    Dim byteValue As String
    Dim count As Int32
    Dim inputReportBuffer As Byte() = Nothing
    Dim success As Boolean

    Try
        'Define a delegate using the IAsyncResult object.

        Dim deleg As ReadInputReportDelegate = _
            DirectCast(ar.AsyncState, ReadInputReportDelegate)

        ' Get the IAsyncResult object and the values of other
paramaters that the
        ' BeginInvoke method passed ByRef.

        deleg.EndInvoke(myDeviceDetected, inputReportBuffer,
success, ar)

        ' Display the received report data in the form's list box.

        If (ar.IsCompleted And success) Then

            MyMarshalToForm("AddItemToListBox", "An Input report
has been read.")

            MyMarshalToForm("AddItemToListBox", " Input Report ID:
" & String.Format("{0:X2} ", inputReportBuffer(0)))

            MyMarshalToForm("AddItemToListBox", " Input Report
Data:")

            For count = 1 To UBound(inputReportBuffer)

                ' Display bytes as 2-character Hex strings.

```

```
            'Es aquesta la que diu que escriguin
al cuadro de recepcio i al cuadro d'envio
            'també canvia el numero de bits a mostrar del
cuadre de recepcio

            byteValue = String.Format("{0:X2} ",
inputReportBuffer(count))

            MyMarshalToForm("AddItemToListBox", " " &
byteValue)

            MyMarshalToForm("TextBoxSelectionStart",
txtBytesReceived.Text)
            MyMarshalToForm("AddItemToTextBox", byteValue)

        Next count

    Else
        MyMarshalToForm("AddItemToListBox", "The attempt to
read an Input report has failed.")
        Debug.Write("The attempt to read an Input report has
failed")
    End If

    MyMarshalToForm("ScrollToBottomOfListBox", "")

    ' Enable requesting another transfer.

    MyMarshalToForm("EnableCmdOnce", "")

Catch ex As Exception
    DisplayException(Me.Name, ex)
    Throw
End Try

End Sub

''' <summary>
''' Envia el Vendor Id i el Product Id en hexadecimal
''' dels quadres de text i converteix el text en Int 16
''' </summary>
'''
''' <param name="myVendorID"> the Vendor ID as a Int16.</param>
''' <param name="myProductID"> the Product ID as a Int16. </param>

Private Sub GetVendorAndProductIDsFromTextBoxes _
    (ByRef myVendorID As Int16, _
    ByRef myProductID As Int16)

    Try
```

```
        myVendorID = Convert.ToInt16(Val("&h" & txtVendorID.Text))
        myProductID = Convert.ToInt16(Val("&h" &
txtProductID.Text))

    Catch ex As Exception
        DisplayException(Me.Name, ex)
        Throw
    End Try

End Sub

''' <summary>
''' Initialize the elements on the form.
''' </summary>
'''
Private Sub InitializeDisplay()

    Dim count As Integer
    Dim byteValue As String

    Try
        ' Create a dropdown list box for each byte to send in a
report.
        ' Display the values as 2-character hex strings.

        For count = 0 To 255

            byteValue = String.Format("{0:X2} ", count)

        Next count

        ' Select a default value for each box

        ' Check the autoincrement box to increment the values each
time a report is sent.

        ' Don't allow the user to select an input report buffer
size until there is
        ' a handle to a HID.

        cmdInputReportBufferSize.Focus()
        cmdInputReportBufferSize.Enabled = False

        lstResults.Items.Add("For a more detailed event log, view
debug statements in Visual Studio's Output window:")
    
```



```

        lstResults.Items.Add("Click Build > Configuration Manager
> Active Solution Configuration > Debug > Close.")
        lstResults.Items.Add("Then click View > Output.")
        lstResults.Items.Add("")

    Catch ex As Exception
        DisplayException(Me.Name, ex)
        Throw
    End Try

End Sub

''' <summary>
''' Enables accessing a form's controls from another thread
''' </summary>
'''
''' <param name="action"> a string that names the action to
perform on the form </param>
''' <param name="textToDisplay"> text that the form displays or
the code uses for
''' another purpose. Actions that don't use text ignore this
parameter. </param>

Private Sub MyMarshalToForm _
    (ByVal action As String, _
    ByVal textToDisplay As String)

    Dim args() As Object = {action, textToDisplay}
    Dim MarshalToFormDelegate As MarshalToForm

    ' The AccessForm routine contains the code that accesses the
form.

    MarshalToFormDelegate = New MarshalToForm(AddressOf
AccessForm)

    ' Execute AccessForm, passing the parameters in args.
MyBase.Invoke(MarshalToFormDelegate, args)

End Sub

''' <summary>
''' Initiates exchanging reports.
''' The application sends a report and requests to read a report.
''' </summary>

Private Sub ScrollToBottomOfListBox()

```

```
Try
    Dim count As Int32

    lstResults.SelectedIndex = lstResults.Items.Count - 1

    ' If the list box is getting too large, trim its contents
    by removing the earliest data.

    If lstResults.Items.Count > 1000 Then

        For count = 1 To 500
            lstResults.Items.RemoveAt(4)
        Next count

    End If

Catch ex As Exception
    DisplayException(Me.Name, ex)
    Throw
End Try
End Sub

''' <summary>
''' Set the number of Input buffers (the number of Input reports
''' the host will store) from the value in the text box.
''' </summary>

Private Sub SetInputReportBufferSize()

    Dim numberOfInputBuffers As Int32

    Try
        ' Get the number of buffers from the text box.

        numberOfInputBuffers =
Convert.ToInt32(Val(txtInputReportBufferSize.Text))
        ' Set the number of buffers.

        MyHid.SetNumberOfInputBuffers _
            (hidHandle, _
            numberOfInputBuffers)

        ' Verify and display the result.

        GetInputReportBufferSize()

    Catch ex As Exception
        DisplayException(Me.Name, ex)
        Throw
```

```
End Try
End Sub

''' <summary>
''' Perform actions that must execute when the program ends.
''' </summary>

Private Sub Shutdown()

    Try
        ' Close open handles to the device.

        If Not hidHandle Is Nothing Then
            If Not hidHandle.IsInvalid Then
                hidHandle.Close()
            End If
        End If

        If Not readHandle Is Nothing Then
            If Not readHandle.IsInvalid Then
                readHandle.Close()
            End If
        End If

        If Not writeHandle Is Nothing Then
            If Not writeHandle.IsInvalid Then
                writeHandle.Close()
            End If
        End If

        ' Stop receiving notifications.

MyDeviceManagement.StopReceivingDeviceNotifications(deviceNotification
Handle)

        Catch ex As Exception
            DisplayException(Me.Name, ex)
            Throw
        End Try

    End Sub

''' <summary>
''' Perform actions that must execute when the program starts.
''' </summary>

Private Sub Startup()
```

```
Try
    MyHid = New Hid()
    InitializeDisplay()

    ' Default USB Vendor ID and Product ID:

    txtVendorID.Text = "0b40"
    txtProductID.Text = "011a"

Catch ex As Exception
    DisplayException(Me.Name, ex)
    Throw
End Try

End Sub

''' <summary>
''' The Product ID has changed in the text box. Call a routine to
handle it.
''' </summary>

Private Sub txtProductID_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
txtProductID.TextChanged

    Try
        DeviceHasChanged()

    Catch ex As Exception
        DisplayException(Me.Name, ex)
        Throw
    End Try
End Sub

''' <summary>
''' The Vendor ID has changed in the text box. Call a routine to
handle it.
''' </summary>

Private Sub txtVendorID_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles txtVendorID.TextChanged

    Try
        DeviceHasChanged()

    Catch ex As Exception
```

```
        DisplayException(Me.Name, ex)
        Throw
    End Try

End Sub

''' <summary>
''' Finalize method.
''' </summary>

Protected Overrides Sub Finalize()
    MyBase.Finalize()
End Sub

''' <summary>
''' Overrides WndProc to enable checking for and handling
WM_DEVICECHANGE messages.
''' </summary>
'''
''' <param name="m"> a Windows Message </param>

Protected Overrides Sub WndProc(ByRef m As Message)

    Try
        ' The OnDeviceChange routine processes WM_DEVICECHANGE
messages.

        If m.Msg = WM_DEVICECHANGE Then
            OnDeviceChange(m)
        End If

        ' Let the base form process the message.

        MyBase.WndProc(m)

    Catch ex As Exception
        DisplayException(Me.Name, ex)
        Throw
    End Try

End Sub

''' <summary>
''' Provides a central mechanism for exception handling.
''' Displays a message box that describes the exception.
''' </summary>
'''
''' <param name="moduleName"> the module where the exception
occurred. </param>
```

```
''' <param name="e"> the exception </param>

Shared Sub DisplayException(ByVal moduleName As String, ByVal e As
Exception)

    Dim message As String
    Dim caption As String

    ' Create an error message.

    message = "Exception: " & e.Message & ControlChars.CrLf & _
"Module: " & moduleName & ControlChars.CrLf & _
"Method: " & e.TargetSite.Name

    caption = "Unexpected Exception"

    MessageBox.Show(message, caption, MessageBoxButtons.OK)
    Debug.Write(message)

End Sub
Private Sub enviarirebre()

    Try
        'success = False

        ' Don't attempt to exchange reports if valid handles
aren't available
        ' (as for a mouse or keyboard under Windows 2000/XP.)

        If (Not (readHandle.IsInvalid) And (Not
writeHandle.IsInvalid)) Then

            ' Don't attempt to send an Output report if the HID
has no Output report.

            If (MyHid.Capabilities.OutputReportByteLength > 0)
Then

                ' Estableix el límit superior de la memòria
intermèdia de l'informe de sortida
                ' Restar 1 a la sortida de l'informe byte de
longitud (OutputReportByteLength) perquè la matriu comença en l'índex
0.

                ReDim
outputReportBuffer(MyHid.Capabilities.OutputReportByteLength - 1)
```

```

' Guardar l'informe de identificació (report ID) al primer byte del
buffer

        outputReportBuffer(0) = 0

' Guarda l'informe de la dada a enviar seguida del
(report ID)

        'En el cas d'utilitzar USB-232 de la casa hexwax,
s'envia un primer byte
        'el qual ens dirà quants bytes s'enviaran
posteriorment.

        'Exemple: 01 02 ' s'envia un byte amb el valor 02
        'Exemple: 03 01 02 03 ' s'envien 3 bytes que seran
01 02 03

        'Exemple: 01 02 03 04 ' s'envia un byte amb el
valor 02

        'El primer byte enviat ens indica el numero de
bytes que s'enviaran

        outputReportBuffer(1) = 1

        'If UBound(outputReportBuffer) > 1 Then
outputReportBuffer(2) = Dadaenviar
        'End If

        ' Write a report.

Dim myOutputReport As New
Hid.OutputReportViaInterruptTransfer
        success = myOutputReport.Write(outputReportBuffer,
writeHandle)

End If

If success Then
        'lstResults.Items.Add("S'ha escrit un informe de
sortida.")

        ' Display the report data in the form's list box.

        'lstResults.Items.Add(" Identificació de l'informe
de sortida: " & String.Format("{0:X2} ", outputReportBuffer(0)))
        'lstResults.Items.Add(" Dades de l'informe de
sortida:")

```

```

        txtBytesReceived.Text = ""
        For count = 1 To UBound(outputReportBuffer)

            ' Ensenya els bytes com a 2 caracters
hexadecimals

            byteValue = String.Format("{0:X2} ",
outputReportBuffer(count))
            ' lstResults.Items.Add(" " & byteValue)

        Next count

    Else
        lstResults.Items.Add("L'intent d'escriure un
informe de sortida ha fallat.")
    End If

    Else
        lstResults.Items.Add("L'HID no té un informe de
sortida.")
    End If

    ' Llegir un informe d'entrada.

    success = False

    ' Don't attempt to send an Input report if the HID has no
Input report.
    ' (The HID spec requires all HIDs to have an interrupt IN
endpoint,
    ' which suggests that all HIDs must support Input
reports.)

    If (MyHid.Capabilities.InputReportByteLength > 0) Then

        ' Estableix el tamany del buffer d'entrada d'informes
        ' Restar 1 al valor de la estructura Cap perquè la
matriu comença en l'índex 0.

        ReDim
inputReportBuffer(MyHid.Capabilities.InputReportByteLength - 1)

        'Else
        'lstResults.Items.Add("The attempt to read an Input
report has failed.")

        'End If

```



```

        'ScrollToBottomOfListBox()

        ' Enable requesting another transfer.

        'AccessForm("EnableCmdOnce", "")

        'Else
        ' Read a report using interrupt transfers.
        ' To enable reading a report without blocking the main
thread, this
        ' application uses an asynchronous delegate.

        'Dim ar As IAsyncResult
        'Dim myInputReport As New
Hid.InputReportViaInterruptTransfer

        ' Define a delegate for the Read method of
myInputReport.

        ' Dim MyReadInputReportDelegate As _
        ' New ReadInputReportDelegate(AddressOf
myInputReport.Read)

        ' The BeginInvoke method calls myInputReport.Read to
attempt to read a report.
        ' The method has the same parameters as the Read
function,
        ' plus two additional parameters:
        ' GetInputReportData is the callback procedure that
executes when the Read function returns.
        ' MyReadInputReportDelegate is the asynchronous
delegate object.
        ' The last parameter can optionally be an object
passed to the callback.

        ' ar = MyReadInputReportDelegate.BeginInvoke _
        ' (hidHandle, _
        ' readHandle, _
        ' writeHandle, _
        'myDeviceDetected, _
        ' inputReportBuffer, _
        'success, _
        'New AsyncCallback(AddressOf GetInputReportData), _
        'MyReadInputReportDelegate)
        'End If

        'Else
        'lstResults.Items.Add("No attempt to read an Input
report was made.")

```

```
        'lstResults.Items.Add("The HID doesn't have an Input
report.")
    End If
    'Else
        'lstResults.Items.Add("Invalid handle. The device is
probably a system mouse or keyboard.")
        'lstResults.Items.Add("No attempt to write an Output
report or read an Input report was made.")

        'ScrollToBottomOfListBox()

    Catch ex As Exception
        DisplayException(Me.Name, ex)
        Throw
    End Try

End Sub

Private Sub programar_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles programar.Click
    paraintermitent()
    pararafaga()
    'Accedeix a la finestra programar

    'Me.Visible = False      'fa desaparèixer el formulari
regulador de llum
    Form1.Show()
    GroupBox2.Visible = True

End Sub

Private Sub hora_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles hora.Tick

    'funció que mostra la hora actual del pc al label1
    'agafa els valors del form programació i els mostra en un
label
    'compara els dos labels i si son iguals encen les bombetes amb
el valor programat

    'Mostrar hora i data actuals
    'Previament(s) 'ha d'habilitar que es vegi l'hora. Això és fa
al Frmload
```

```
Label1.Text = CStr(TimeOfDay) & " " & CStr(Today.Day & " - " & Today.Month & " - " & Today.Year)
```

```
'Mostrar hores minuts i segons (dia) i la data procedents del formulari programació
```

```
Label2.Text = CStr(dia) & " " & CStr(dial & " - " & mes & " - " & any)
```

```
'Si coincideixen les hores i les dates dels dos labels es mouen les barres de la finestra principal i treuen els seus valors pel port USB
```

```
If Label1.Text = Label2.Text Then
```

```
barra1.Value = barraprg1  
barra2.Value = barraprg2  
barra3.Value = barraprg3  
barra4.Value = barraprg4  
barra5.Value = barraprg5  
barra6.Value = barraprg6  
barra7.Value = barraprg7  
'barra8.Value = barraprg8
```

```
Dadaenviar = Convert.ToByte(barra1.Value)  
enviarirebre()
```

```
Dadaenviar = Convert.ToByte(barra2.Value)  
enviarirebre()
```

```
Dadaenviar = Convert.ToByte(barra3.Value)  
enviarirebre()
```

```
Dadaenviar = Convert.ToByte(barra4.Value)  
enviarirebre()
```

```
Dadaenviar = Convert.ToByte(barra5.Value)  
enviarirebre()
```

```
Dadaenviar = Convert.ToByte(barra6.Value)  
enviarirebre()
```

```
Dadaenviar = Convert.ToByte(barra7.Value)  
enviarirebre()
```

```
'Dadaenviar = Convert.ToByte(barra8.Value)  
'enviarirebre()
```

```
GroupBox2.Visible = False
```

```
End If
```

```
End Sub

Private Sub barra1_Scroll(ByVal sender As Object, ByVal e As
System.Windows.Forms.ScrollEventArgs) Handles barra1.Scroll

    pararafaga()
    paraintermitent()
    txtBytesReceived.Text = ""

    Text1.Text = CStr(barra1.Value) 'mostra el valor de la barra a
un quadre de text
    barra = 1
    Try

        Dadaenviar = Convert.ToByte(barra1.Value) 'per enviar una
dada pel port USB s'ha de convertir a byte
        enviarirebre() 'es crida a la funció enviar i rebre

    Catch ex As Exception
        DisplayException(Me.Name, ex)
        Throw
    End Try

End Sub

Private Sub barra2_Scroll(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.ScrollEventArgs) Handles barra2.Scroll

    pararafaga()
    paraintermitent()
    txtBytesReceived.Text = ""

    Text2.Text = CStr(barra2.Value)
    barra = 2
    Try

        Dadaenviar = Convert.ToByte(barra2.Value)
        enviarirebre()

    Catch ex As Exception
        DisplayException(Me.Name, ex)
        Throw
    End Try

End Sub

Private Sub barra3_Scroll(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.ScrollEventArgs) Handles barra3.Scroll
```

```
pararafaga()  
paraintermitent()  
txtBytesReceived.Text = ""  
  
Text3.Text = CStr(barra3.Value)  
barra = 3  
Try  
  
    Dadaenviar = Convert.ToByte(barra3.Value)  
    enviarirebre()  
  
Catch ex As Exception  
    DisplayException(Me.Name, ex)  
    Throw  
  
End Try  
  
End Sub  
  
Private Sub barra4_Scroll(ByVal sender As System.Object, ByVal e  
As System.Windows.Forms.ScrollEventArgs) Handles barra4.Scroll  
  
    pararafaga()  
    paraintermitent()  
    txtBytesReceived.Text = ""  
  
    Text4.Text = CStr(barra4.Value) 'mostra el valor de la barra a  
un quadre de text  
    barra = 4  
    Try  
  
        Dadaenviar = Convert.ToByte(barra4.Value) 'per enviar una  
dada pel port USB s'ha de convertir a byte  
        enviarirebre() 'es crida a la funció enviar i rebre  
  
    Catch ex As Exception  
        DisplayException(Me.Name, ex)  
        Throw  
    End Try  
  
End Sub  
  
Private Sub barra5_Scroll(ByVal sender As System.Object, ByVal e  
As System.Windows.Forms.ScrollEventArgs) Handles barra5.Scroll  
  
    pararafaga()  
    paraintermitent()  
    txtBytesReceived.Text = ""
```

```
Text5.Text = CStr(barra5.Value) 'mostra el valor de la barra a
un quadre de text
barra = 5
Try

    Dadaenviar = Convert.ToByte(barra5.Value) 'per enviar una
dada pel port USB s'ha de convertir a byte
    enviarirebre() 'es crida a la funció enviar i rebre

Catch ex As Exception
    DisplayException(Me.Name, ex)
    Throw
End Try

End Sub

Private Sub barra6_Scroll(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.ScrollEventArgs) Handles barra6.Scroll

    pararafaga()
    paraintermitent()
    txtBytesReceived.Text = ""

    Text6.Text = CStr(barra6.Value) 'mostra el valor de la barra a
un quadre de text
    barra = 6
    Try

        Dadaenviar = Convert.ToByte(barra6.Value) 'per enviar una
dada pel port USB s'ha de convertir a byte
        enviarirebre() 'es crida a la funció enviar i rebre

    Catch ex As Exception
        DisplayException(Me.Name, ex)
        Throw
    End Try

End Sub

Private Sub barra7_Scroll(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.ScrollEventArgs) Handles barra7.Scroll

    pararafaga()
    paraintermitent()
    txtBytesReceived.Text = ""
```

```
Text7.Text = CStr(barra7.Value) 'mostra el valor de la barra a
un quadre de text
barra = 7
Try

    Dadaenviar = Convert.ToByte(barra7.Value) 'per enviar una
dada pel port USB s'ha de convertir a byte
    enviarirebre() 'es crida a la funció enviar i rebre

Catch ex As Exception
    DisplayException(Me.Name, ex)
    Throw
End Try
End Sub

Private Sub barra8_Scroll(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.ScrollEventArgs) Handles barra8.Scroll

    pararafaga()
    paraintermitent()
    txtBytesReceived.Text = ""

    Text8.Text = CStr(barra8.Value) 'mostra el valor de la barra a
un quadre de text
    barra = 8
    Try

        Dadaenviar = Convert.ToByte(barra8.Value) 'per enviar una
dada pel port USB s'ha de convertir a byte
        enviarirebre() 'es crida a la funció enviar i rebre

    Catch ex As Exception
        DisplayException(Me.Name, ex)
        Throw
    End Try

End Sub

Private Sub ProgramacióToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ProgramacióToolStripMenuItem.Click

    'Permet veure i ocultar el Groupox2

    If GroupBox2.Visible = True Then
        GroupBox2.Visible = False
    Else
        GroupBox2.Visible = True
    End If
End Sub
```

```
End If

End Sub

Private Sub ActualToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ActualToolStripMenuItem.Click

    'Permet veure i ocultar el Groupox1

    If GroupBox1.Visible = True Then
        GroupBox1.Visible = False
    Else
        GroupBox1.Visible = True
    End If
End Sub

Private Sub
IdentificadorsDelDispositiuToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
IdentificadorsDelDispositiuToolStripMenuItem.Click

    'Permet veure i ocultar Vendor i Product ID's

    If fraDeviceIdentifiers.Visible = True Then
        fraDeviceIdentifiers.Visible = False
    Else
        fraDeviceIdentifiers.Visible = True
    End If

    If cmdFindDevice.Visible = True Then
        cmdFindDevice.Visible = False
    Else
        cmdFindDevice.Visible = True
    End If
End Sub

Private Sub maxim_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles maxim.Click

    'Posem les barres a dalt de tot (màxima intensitat)
    barral.Value = barral.Minimum
```



```
barra2.Value = barra2.Minimum  
barra3.Value = barra3.Minimum  
barra4.Value = barra4.Minimum  
barra5.Value = barra5.Minimum  
barra6.Value = barra6.Minimum  
barra7.Value = barra7.Minimum  
barra8.Value = barra8.Minimum
```

```
Dadaenviar = 1  
enviarirebre()  
Dadaenviar = 33  
enviarirebre()  
Dadaenviar = 64  
enviarirebre()  
Dadaenviar = 96  
enviarirebre()  
Dadaenviar = 128  
enviarirebre()  
Dadaenviar = 160  
enviarirebre()  
Dadaenviar = 192  
enviarirebre()
```

```
'Dadaenviar = 224  
'enviarirebre()
```

End Sub

```
Private Sub mínim_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles mínim.Click
```

```
'Posem les barres a baix de tot (mínima intensitat)  
barra1.Value = barra1.Maximum  
barra2.Value = barra2.Maximum  
barra3.Value = barra3.Maximum  
barra4.Value = barra4.Maximum  
barra5.Value = barra5.Maximum  
barra6.Value = barra6.Maximum  
barra7.Value = barra7.Maximum  
barra8.Value = barra8.Maximum
```

```
Dadaenviar = 25  
enviarirebre()  
Dadaenviar = 57
```

```
enviarirebre()  
Dadaenviar = 89  
enviarirebre()  
Dadaenviar = 120  
enviarirebre()  
Dadaenviar = 151  
enviarirebre()  
Dadaenviar = 183  
enviarirebre()  
Dadaenviar = 215  
enviarirebre()  
  
'Dadaenviar = 216  
'enviarirebre()
```

```
End Sub
```

```
Private Sub intermitent_Click(ByVal sender As System.Object, ByVal  
e As System.EventArgs) Handles intermitent.Click
```

```
    pararafaga()  
  
    c = True  
  
    ini() 'inicialitza els temps als trackbars  
  
    If intermitent.Text = "INTERMITENT" Then  
        ' fes intermitencia  
  
        intermitente()  
  
    Else  
        ' para intermitencia  
        c = False  
  
    End If  
  
    If c = True Then  
        intermitent.Text = "PARA INTERMITENCIA"  
  
    Else  
        paraintermitent()  
    End If
```

```
End Sub

Private Sub intermitente()

    Dadaenviar = Convert.ToByte(barra1.Value)
    enviarirebre()
    Dadaenviar = Convert.ToByte(barra2.Value)
    enviarirebre()
    Dadaenviar = Convert.ToByte(barra3.Value)
    enviarirebre()
    Dadaenviar = Convert.ToByte(barra4.Value)
    enviarirebre()
    Dadaenviar = Convert.ToByte(barra5.Value)
    enviarirebre()
    Dadaenviar = Convert.ToByte(barra6.Value)
    enviarirebre()
    Dadaenviar = Convert.ToByte(barra7.Value)
    enviarirebre()
    'Dadaenviar = 204
    'enviarirebre()
    Timer2.Start()

End Sub

Public Sub paraintermitent()
    intermitent.Text = "INTERMITENT"
    c = False
End Sub

Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick

    'intermitent

    If c = True Then

        Timer2.Stop()

        Dadaenviar = 25
        enviarirebre()
        Dadaenviar = 57
        enviarirebre()
        Dadaenviar = 89
        enviarirebre()
        Dadaenviar = 120
        enviarirebre()
        Dadaenviar = 151
```

```
        enviarirebre()  
        Dadaenviar = 183  
        enviarirebre()  
        Dadaenviar = 215  
        enviarirebre()  
  
        'Dadaenviar = 216  
        'enviarirebre()  
  
        Timer3.Start()  
  
    End If  
  
End Sub  
  
Private Sub Timer3_Tick(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Timer3.Tick  
  
    If c = True Then  
        Timer3.Stop()  
        intermitente()  
    End If  
  
End Sub  
  
Private Sub track1_Scroll(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles track1.Scroll  
  
    Timer2.Interval = Convert.ToInt32((track1.Value) * 125)  
  
End Sub  
  
Private Sub ini()  
  
    If c = True Then  
        Timer2.Interval = Convert.ToInt32((track1.Value) * 125)  
        Timer3.Interval = Convert.ToInt32((track2.Value) * 125)  
    End If  
  
End Sub
```

```
Private Sub Track2_Scroll(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles track2.Scroll

    Timer3.Interval = Convert.ToInt32((track2.Value) * 125)

End Sub

Private Sub rafaga_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles rafaga.Click

    paraintermitent()

    i = True

    If rafaga.Text = "RÀFAGA" Then

        rraffaagga()

    Else
        ' para rafaga
        i = False
    End If

    If i = True Then
        rafaga.Text = "PARAR RÀFAGA"
    Else
        i = False
        pararafaga()
    End If

End Sub

Private Sub rraffaagga()

    T1 = True
    Dadaenviar = Convert.ToByte(barral.Value)
    enviarirebre()
    Timer1.Start()

End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick

    Timer1.Stop()

    If T1 = True Then
        Dadaenviar = 25
        enviarirebre()
    End If

End Sub
```

```
Dadaenviar = Convert.ToByte(barra2.Value)
enviarirebre()
Timer4.Start()
T1 = False
T2 = True
End If

If T3 = True Then
    Dadaenviar = 89
    enviarirebre()
    Dadaenviar = Convert.ToByte(barra4.Value)
    enviarirebre()
    Timer4.Start()
    T3 = False
    T4 = True
End If

If T5 = True Then
    Dadaenviar = 151
    enviarirebre()
    Dadaenviar = Convert.ToByte(barra6.Value)
    enviarirebre()
    Timer4.Start()
    T5 = False
    T6 = True
End If

If T7 = True Then
    Dadaenviar = 215
    enviarirebre()
    T7 = False
    Timer5.Start()
    'rraffaagga()
End If

End Sub

Private Sub Timer4_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer4.Tick

    Timer4.Stop()

    If T2 = True Then
        Dadaenviar = 57
        enviarirebre()
        Dadaenviar = Convert.ToByte(barra3.Value)
```

```
        enviarirebre()
        Timer1.Start()
        T2 = False
        T3 = True
    End If

    If T4 = True Then
        Dadaenviar = 120
        enviarirebre()
        Dadaenviar = Convert.ToByte(barra5.Value)
        enviarirebre()
        Timer1.Start()
        T4 = False
        T5 = True
    End If

    If T6 = True Then
        Dadaenviar = 183
        enviarirebre()
        Dadaenviar = Convert.ToByte(barra7.Value)
        enviarirebre()
        Timer1.Start()
        T6 = False
        T7 = True
    End If
End Sub

Private Sub Timer5_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer5.Tick
    If i = True Then
        Timer5.Stop()
        rraffaagga()

    End If
End Sub

Private Sub TrackRRR1_Scroll(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles TrackRRR1.Scroll
    Timer1.Interval = Convert.ToInt32((TrackRRR1.Value) * 125)
    Timer4.Interval = Convert.ToInt32((TrackRRR1.Value) * 125)
End Sub
Private Sub pararafaga()

    rafaga.Text = "RÀFAGA"
    Dadaenviar = 183
    enviarirebre()
    i = False
```

```
End Sub
```

```
Private Sub INTERMIG_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles INTERMIG.Click
```

```
    'Posem les barres a dalt de tot (màxima intensitat)
    barra1.Value = 12
    barra2.Value = 44
    barra3.Value = 75
    barra4.Value = 107
    barra5.Value = 138
    barra6.Value = 170
    barra7.Value = 203
    barra8.Value = 232
```

```
    Dadaenviar = 12
    enviarirebre()
    Dadaenviar = 44
    enviarirebre()
    Dadaenviar = 75
    enviarirebre()
    Dadaenviar = 109
    enviarirebre()
    Dadaenviar = 141
    enviarirebre()
    Dadaenviar = 172
    enviarirebre()
    Dadaenviar = 204
    enviarirebre()
```

```
    'Dadaenviar = 224
    'enviarirebre()
```

```
End Sub
```

```
Private Sub max1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles max1.Click
```

```
    barra1.Value = barra1.Minimum
    Dadaenviar = 1
    enviarirebre()
```

```
End Sub
```

```
Private Sub max2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles max2.Click
```



```
barra2.Value = barra2.Minimum
Dadaenviar = 33
enviarirebre()
```

```
End Sub
```

```
Private Sub max3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles max3.Click
```

```
barra3.Value = barra3.Minimum
Dadaenviar = 64
enviarirebre()
```

```
End Sub
```

```
Private Sub max4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles max4.Click
```

```
barra4.Value = barra4.Minimum
Dadaenviar = 96
enviarirebre()
```

```
End Sub
```

```
Private Sub max5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles max5.Click
```

```
barra5.Value = barra5.Minimum
Dadaenviar = 128
enviarirebre()
```

```
End Sub
```

```
Private Sub max6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles max6.Click
```

```
barra6.Value = barra6.Minimum
Dadaenviar = 160
enviarirebre()
```

```
End Sub
```

```
Private Sub max7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles max7.Click
```

```
barra7.Value = barra7.Minimum
Dadaenviar = 192
enviarirebre()
```

End Sub

```
Private Sub min1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles min1.Click
```

```
    barra1.Value = barra1.Maximum  
    Dadaenviar = 25  
    enviarirebre()
```

End Sub

```
Private Sub min2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles min2.Click
```

```
    barra2.Value = barra2.Maximum  
    Dadaenviar = 57  
    enviarirebre()
```

End Sub

```
Private Sub min3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles min3.Click
```

```
    barra3.Value = barra3.Maximum  
    Dadaenviar = 89  
    enviarirebre()
```

End Sub

```
Private Sub min4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles min4.Click
```

```
    barra4.Value = barra4.Maximum  
    Dadaenviar = 120  
    enviarirebre()
```

End Sub

```
Private Sub min5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles min5.Click
```

```
    barra5.Value = barra5.Maximum  
    Dadaenviar = 151  
    enviarirebre()
```

End Sub

```
Private Sub min6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles min6.Click
```

```
barra6.Value = barra6.Maximum  
Dadaenviar = 183  
enviarirebre()
```

End Sub

```
Private Sub min7_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles min7.Click
```

```
barra7.Value = barra7.Maximum  
Dadaenviar = 215  
enviarirebre()
```

End Sub

```
Private Sub mig1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles mig1.Click
```

```
barra1.Value = 12  
Dadaenviar = 12  
enviarirebre()
```

End Sub

```
Private Sub mig2_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles mig2.Click
```

```
barra2.Value = 44  
Dadaenviar = 44  
enviarirebre()
```

End Sub

```
Private Sub mig3_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles mig3.Click
```

```
barra3.Value = 75  
Dadaenviar = 75  
enviarirebre()
```

End Sub

```
Private Sub mij4_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles mij4.Click
```

```
barra4.Value = 107  
Dadaenviar = 107  
enviarirebre()
```

```
End Sub

Private Sub mig5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mig5.Click

    barra5.Value = 138
    Dadaenviar = 140
    enviarirebre()

End Sub

Private Sub mig6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mig6.Click

    barra6.Value = 170
    Dadaenviar = 172
    enviarirebre()

End Sub

Private Sub mig7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mig7.Click

    barra7.Value = 202
    Dadaenviar = 204
    enviarirebre()

End Sub

Private Sub GuardarProgramacióToolStripMenuItem_Click(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
GuardarProgramacióToolStripMenuItem.Click

    Me.Close()
End Sub

Private Sub GroupBox5_Enter(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles GroupBox5.Enter

    paraintermitent()
    pararafaga()

End Sub

Private Sub GroupBox4_Enter(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles GroupBox4.Enter

    paraintermitent()
    pararafaga()

End Sub
```

```
Private Sub GroupBox6_Enter(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles GroupBox6.Enter
    paraintermitent()
    pararafaga()
End Sub
```

```
Private Sub NumeracióBarresToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NumeracióBarresToolStripMenuItem.Click
```

```
    If GroupBox7.Visible = True Then
        GroupBox7.Visible = False
    Else
        GroupBox7.Visible = True
    End If
End Sub
```

```
Private Sub ResultatsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ResultatsToolStripMenuItem.Click
```

```
    If lstResults.Visible = True Then
        lstResults.Visible = False
    Else
        lstResults.Visible = True
    End If
End Sub
```

```
Private Sub Timer6_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer6.Tick
```

```
    'quan el timer es desborda mira si ha trobat el dispositiu, si no l'ha trobat crida
```

```
    'a FindTheHid. Si l'ha trobat desabilita el timer.
```

```
    If myDeviceDetected = False Then
        FindTheHid()
    Else
        Timer6.Enabled = False
    End If
```

```
End Sub
```

```
Private Sub parartot()
    Label6.ForeColor = Color.Red
```

```
Label6.Text = ("DISPOSITIU DESCONNECTAT")
GroupBox4.Enabled = False
GroupBox3.Enabled = False
GroupBox6.Enabled = False
GroupBox5.Enabled = False
programar.Enabled = False
End Sub

End Class

Option Explicit On

Public Class Form1

    Dim data As Date

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        'Posem les barres a baix de tot (mínima intensitat)

        b1.Value = b1.Maximum
        b2.Value = b2.Maximum
        b3.Value = b3.Maximum
        b4.Value = b4.Maximum
        b5.Value = b5.Maximum
        b6.Value = b6.Maximum
        b7.Value = b7.Maximum
        b8.Value = b8.Maximum

        'Quan es carrega el formulari, posa la data actual a les
següents variables i ho mostra pels quadres de text que són
informatius

        dia1 = mcDateSelection.TodayDate.Day
        mes = mcDateSelection.TodayDate.Month
        any = mcDateSelection.TodayDate.Year

    End Sub

    Private Sub cancelar_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cancelar.Click

        'Sortim de la finestra programació i anem a la finestra
principal del programa
```

```
'programacio = ""
FrmMain.Visible = True
Me.Close()

End Sub

Private Sub aceptar_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles aceptar.Click

    'Fixa la data d'avui al calendari
    mcDateSelection.TodayDate = Now

    'Posa els valors de les hores, minuts i segons a la variable
dia

    dia = Nhora.Value & ":" & Nminut.Value & ":" & Nsegon.Value

    'Posa el valor de les barres a unes variables públiques que
comparteixen els dos formularis

    barraprg1 = b1.Value
    barraprg2 = b2.Value
    barraprg3 = b3.Value
    barraprg4 = b4.Value
    barraprg5 = b5.Value
    barraprg6 = b6.Value
    barraprg7 = b7.Value
    'barraprg8 = b8.Value

    'Tanca la finestra programar i mostra la finestra principal

    FrmMain.Visible = True
    Me.Close()

End Sub

Private Sub mcDateSelection_DateChanged(ByVal sender As
System.Object, ByVal e As System.Windows.Forms.DateRangeEventArgs)
Handles mcDateSelection.DateChanged

    'Guardem les dates(dia, mes i any)que hem triat del calendari
a les variables que comparteixen els dos formularis

    dial = mcDateSelection.SelectionStart.Day
    mes = mcDateSelection.SelectionStart.Month
    any = mcDateSelection.SelectionStart.Year
```

'es mostra en quadres de text el valor del dia mes i any
triats del calendari (informatiu)

End Sub

```
Private Sub maxx1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles maxx1.Click
```

```
b1.Value = b1.Minimum
```

End Sub

```
Private Sub maxx2_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles maxx2.Click
```

```
b2.Value = b2.Minimum
```

End Sub

```
Private Sub maxx3_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles maxx3.Click
```

```
b3.Value = b3.Minimum
```

End Sub

```
Private Sub maxx4_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles maxx4.Click
```

```
b4.Value = b4.Minimum
```

End Sub

```
Private Sub maxx5_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles maxx5.Click
```

```
b5.Value = b5.Minimum
```

End Sub

```
Private Sub maxx6_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles maxx6.Click
```



```
b6.Value = b6.Minimum
```

```
End Sub
```

```
Private Sub maxx7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles maxx7.Click
```

```
b7.Value = b7.Minimum
```

```
End Sub
```

```
Private Sub minn1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles minn1.Click
```

```
b1.Value = b1.Maximum
```

```
End Sub
```

```
Private Sub minn2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles minn2.Click
```

```
b2.Value = b2.Maximum
```

```
End Sub
```

```
Private Sub minn3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles minn3.Click
```

```
b3.Value = b3.Maximum
```

```
End Sub
```

```
Private Sub minn4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles minn4.Click
```

```
b4.Value = b4.Maximum
```

```
End Sub
```

```
Private Sub minn5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles minn5.Click
```

```
b5.Value = b5.Maximum
```

End Sub

```
Private Sub minn6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles minn6.Click
```

```
    b6.Value = b6.Maximum
```

End Sub

```
Private Sub minn7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles minn7.Click
```

```
    b7.Value = b7.Maximum
```

End Sub

```
Private Sub migg1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles migg1.Click
```

```
    b1.Value = 12
```

End Sub

```
Private Sub migg2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles migg2.Click
```

```
    b2.Value = 44
```

End Sub

```
Private Sub migg3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles migg3.Click
```

```
    b3.Value = 75
```

End Sub

```
Private Sub mijj4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles mijj4.Click
```

```
    b4.Value = 107
```

End Sub

```
Private Sub migg5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles migg5.Click
```

```
b5.Value = 138
```

```
End Sub
```

```
Private Sub migg6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles migg6.Click
```

```
b6.Value = 170
```

```
End Sub
```

```
Private Sub migg7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles migg7.Click
```

```
b7.Value = 202
```

```
End Sub
```

```
Private Sub maxim_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles maxim.Click
```

```
b1.Value = b1.Minimum  
b2.Value = b2.Minimum  
b3.Value = b3.Minimum  
b4.Value = b4.Minimum  
b5.Value = b5.Minimum  
b6.Value = b6.Minimum  
b7.Value = b7.Minimum
```

```
End Sub
```

```
Private Sub mínim_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles mínim.Click
```

```
b1.Value = b1.Maximum  
b2.Value = b2.Maximum  
b3.Value = b3.Maximum  
b4.Value = b4.Maximum  
b5.Value = b5.Maximum  
b6.Value = b6.Maximum  
b7.Value = b7.Maximum
```

```
End Sub
```

```
Private Sub INTERMIGG_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles INTERMIGG.Click
```

```
    b1.Value = 12
    b2.Value = 44
    b3.Value = 75
    b4.Value = 107
    b5.Value = 138
    b6.Value = 170
    b7.Value = 202
```

```
End Sub
End Class
```

```
Option Strict Off
Option Explicit On
Module Module2
```

```
    'UPGRADE_WARNING: El límite inferior de la matriz barras ha
cambiado de 1 a 0. Haga clic aquí para obtener más información: 'ms-
help://MS.VSCC.v90/dv_commoner/local/redirect.htm?keyword="0F1C9BE1-
AF9D-476E-83B1-17D43BECFF20"'
```

```
    'Public barras(7) As Byte
Public programacio As Object
Public dia As Date
Public dia1 As Short 'dia del calendari
Public mes As Short 'mes del calendari
Public any As Short 'any del calendari
Public hora As Short
Public minut As Short
Public barraprg1 As Short
Public barraprg2 As Short
Public barraprg3 As Short
Public barraprg4 As Short
Public barraprg5 As Short
Public barraprg6 As Short
Public barraprg7 As Short
Public barraprg8 As Short
```

```
End Module
```

A.6. FOTOLITS

