



**Escola Universitària Politècnica  
de Vilanova i la Geltrú**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# “PROGRAMACIÓ D’UN PIC AMB MICRO’PIC TRAINER”

**AUTORS:** CRISTINA LOZAR I BARBOSA  
GEMA CAVIEDES I ZAMBRANO  
NÈSTOR GIMÉNEZ I ORÓ

**TITULACIÓ:** ENGINYERIA TÈCNICA EN ELECTRÒNICA INDUSTRIAL

**DIRECTOR:** PERE PONSA ASENSIO

**DEPARTAMENT:** ESAII

**DATA:** 22 de Maig de 2002

## MINIPROJECTE

### RESUM

Aquest projecte consta de tres parts. La primera tracta d'una petita introducció als microcontroladors, especialment el PIC i les seves diferents famílies, i més concretament el PIC16F84.

Una segona part on es pot trobar la descripció del funcionament de les diferents parts que componen l'entrenador de PIC's, el Micro PIC Trainer. I del software necessari per la programació d'un PIC.

La tercera part consisteix en l'explicació de diverses aplicacions efectuades sobre el PIC, amb l'objectiu de governar el comportament d'un PIC-BOT 2.

### Paraules clau

Microcontrolador	PIC	PIC 16F84	Microbòtica
Micro Pic Trainer	PIC-BOT2	Picme-Tr	MPASM

## INDEX

1.	VISIÓ GENERAL DELS MICROCONTROLADORS	4
1.1	INTRODUCCIÓ	4
1.2	ARQUITECTURA INTERNA	4
1.3	PROGRAMACIÓ DE MICROCONTROLADORS	7
2.	EL MICROCONTROLADOR PIC	7
2.1	FAMÍLIES DELS PIC	8
2.2	EL MICROCONTROLADOR PIC16F84	12
2.2.1	ASPECTE EXTERN	13
2.2.1.1	FREQÜÈNCIA DE FUNCIONAMENT	15
2.2.1.2	REINICIACIÓ O RESET	16
2.2.1.3	ELS PORTS D'E/S	17
3.	MICRO'PIC TRAINER	17
3.1	COMPONENTS	17
3.2	DESCRIPCIÓ PER BLOCS DEL MICRO'PIC TRAINER	19
4.	CONNEXIÓ ENTRE PC , MICROPIC-TRAINER I EL PIC	24
5	PROGRAMACIÓ DEL PIC	24
5.1	PASSOS A SEGUIR PER LA PROGRAMACIÓ DEL PIC	25
5.2	TRANSFERÈNCIA DEL PROGRAMA DES DEL MICRO PIC-TRAINER AL PIC	26
5.3	TUTORIAL MPASM	27
5.4	TUTORIAL PICME-TR	28
6	APLICACIONS	31
6.1	RASTREJADOR	35
6.2	LAPA	37
6.3	EXPLORADOR	39
6.4	TRAJECTÒRIA	43

## 1. VISIÓ GENERAL DELS MICROCONTROLADORS

## 1.1 INTRODUCCIÓ

Un microcontrolador és un circuit integrat en el que el seu xip inclou un computador de prestacions limitades, normalment es dedica a resoldre tasques específiques i que pel seu reduït format i pes va incorporat a l'aplicació que controla, pel que s'anomena microcontrolador incrustat. Es pot trobar per exemple als microbots, als telèfons mòbils i al teclat dels ordinadors.

La meitat de la producció dels microcontroadors es dedica al mercat de les comunicacions i productes de consum. La resta va a parar al sector informàtic, l'automoció i a les aplicacions industrials.

Les limitacions tecnològiques impedeixen que les parts del computador tinguin una potència i capacitat elevada..

En el mercat existeixen molts models i hem de seleccionar aquell que s'ajusti més idòniament a l'aplicació que desitgem realitzar, segons la capacitat, la velocitat de funcionament, el tipus de memòria, el nombre de línies d'entrada/sortida, la possibilitat de disposar d'un convertidors analògic digital, etc.

## 1.2 ARQUITECTURA INTERNA

Les principals parts que componen un microcontrolador són:

- 1.- Processador.
- 2.- Memòria no volàtil: ROM, EPROM, OTP, EEPROM, FLASH.
- 3.- Memòria RAM.
- 4.- Línies d'E/S pels controladors de perifèrics:
  - 4.1.- Comunicació paral·lela.
  - 4.2.- Comunicació sèrie.
  - 4.3.- Portes de comunicació (bus I<sup>2</sup>C, USB, etc.)
- 5.- Recursos auxiliars:
  - 5.1.- Relotge.
  - 5.2.- Temporitzadors.

5.3.- Watchdog.

5.4.- Convertidors A/D i D/A.

5.5.- Comparadors analògics.

5.6.- Protecció contra fallades d'alimentació.

5.7.- Estat de repòs o baix consum.

### **1.- Processador:**

Conté un nombre reduït d'instruccions, i gràcies a les arquitectures Harvard i RISC i la tècnica de la segmentació del processador (descomposant en etapes per poder processar una instrucció diferent en cadascuna d'elles i treballar amb varies alhora, pipe-line), aconseguim un rendiment i una velocitat més elevades.

### **2.-Memòria no volàtil:**

-ROM:

La memòria es grava durant el procés de fabricació. No es pot tornar a gravar mai més.

-EPROM:

La memòria es grava utilitzant un dispositiu físic comandat des de un PC, en la càpsula hi ha un cristall, si s'apliquen raigs ultraviolats es pot esborrar la memòria i tornar a utilitzar el xip.

-OTP(Only Time Programable):

L'usuari només pot gravar una vegada aquesta memòria, no es pot tornar a esborrar.

-EEPROM:

Pot ser programada o esborrada tantes vegades com es vulgui.

-FLASH:

Aquesta memòria es pot gravar i esborrar els cops que vulguis

### **3.-Memòria RAM:**

Els programes que utilitzem canvien constantment, per això necessitem una memòria de lectura i escriptura, encara que aquest tipus de memòria desaparegui quan apaguem el PC.

#### **4.-Línies d'E/S pels controladors de perifèrics:**

Els pins d'un microcontrolador serveixen per rebre l'alimentació, el cristall de quars, regular la freqüència de treball, per provocar el reset i per suportar la comunicació amb els perifèrics externs. Les línies d'E/S que s'adapten amb els perifèrics treballen amb informació en paral·lel, s'agrupen en conjunts de vuit que reben el nom de Portes. Hi ha models que suporten la comunicació sèrie, d'altres disposen de conjunts de línies que implementen portes de comunicació per a diversos protocols (I<sup>2</sup>C, USB...)

#### **5.-Recursos auxiliars:**

Cada model de microcontrolador incorpora diferents dispositius segons l'aplicació que hagin de realitzar. Els més comuns són:

##### **5.1. – Relotge:**

Genera impulsos que sincronitzen el funcionament del sistema.

##### **5.2. –Temporitzadors:**

Orientats a controlar els temps.

##### **5.3. –Watchdog:**

Provoca una reinicialització quan el programa queda bloquejat.

##### **5.4. –Convertidors A/D i D/A:**

Per poder rebre i enviar senyals analògics.

##### **5.5. –Comparadors analògics:**

Per a verificar el valor d'una senyal analògica.

##### **5.6. –Protecció contra fallades de l'alimentació.**

##### **5.7. –Estat de repòs o baix consum:**

El nivell d'energia es redueix al mínim.

### **1.3 PROGRAMACIÓ DE MICROCONTROLADORS**

Degut a que la capacitat de la memòria de contenció d'instruccions no és molt gran el llenguatge que s'ha d'emprar ha de ser de nivell baix perquè generi un codi reduït, per tant el llenguatge utilitzat és l'Ensamblador, que és el llenguatge més proper a la màquina. També es poden fer servir altres llenguatges de més alt nivell, com C o BASIC, que permeten l'execució del programa línia a línia.

## 2. EL MICROCONTROLADOR PIC

Els **microcontroladors PIC** pertanyen al tipus de microcontroladors de 8 bits desenvolupat i fabricat per **Microchip**. Aquest tipus de micro és el més emprat en els dissenys degut a la seva flexibilitat i eficiència.

Els microcontroladors **PIC** han estat ven rebuts pels professionals de la Microelectrònica i els de la Microinformàtica degut a que ofereixen un gran ventall de característiques i propietats, com són:

- La facilitat de maneig.
- El seu cost és més econòmic que altres micros existents en el mercat fabricat per empreses de la competència.
- Ofereixen una bona mitjana de paràmetres:
  - Elevada velocitat de funcionament.
  - Baix consum amb ampli rang de tensions d'alimentació.
  - Els programes són compactes.
  - Un nombre reduït d'instruccions.
- Existeixen una gran varietat d'eines hardware que permeten gravar, depurar, esborrar i simular el comportament del PIC les quals són de fàcil utilització i econòmiques.
- La gran quantitat de models del PIC permet seleccionar aquell que més s'adequa a les necessitats de l'aplicació.

Microchip fabrica diferents tipus de microcontroladors de 4, 8, 16 i 32 bits però els més utilitzats en el desenvolupament de projectes és el de 8 bits, és a dir el PIC, degut a les

seves prestacions ja que cobreixen gairebé totes les necessitats que qualsevol altre microcontrolador pot oferir.

## 2.1 FAMÍLIES DELS PIC

**Microchip** posseeix de quatre famílies de 8 bits amb l'objectiu de poder cobrir les necessitats dels clients majoristes ja que és molt important a l'hora de realitzar un projecte escollir el microcontrolador que més s'adapti a les necessitats per tal que aquest realitzi la seva funció correctament amb el mínim pressupost possible.

Les famílies de micro**PIC** es classifiquen en gammes en funció del nombre de pins:

### - **Gamma nana: PIC12C(F)XXX de 8 pins**

Aquests micro es caracteritzen per la seva reduïda dimensió doncs disposa de només 8 pins. S'alimenta a una tensió de corrent continua (DC) compresa entre 2,5 V i 5,5 V i el seu consum és menor de 2 mA treballant a una freqüència de 4 MHz, amb una alimentació de 5 V i el cicle d'execució d'una instrucció és d'un microsegon, excepte la de salt que la seva durada és el doble. El format de les instruccions pot ser de 12 bits amb un nombre de 33 instruccions o un format de 14 bits amb una varietat de 35 instruccions, els més utilitzats són els de 12 bits ja que permeten una major compressió del codi del programa.

Aquests PIC poden destinar 6 dels seus 8 pins com a línies d'E/S per a la connexió de perifèrics degut a la disposició d'un oscil·lador intern R-C.

Dins d'aquesta subfamília es troben diferents models els quals per la seva arquitectura es troben dins d'altres gammes com són els models PIC12C5XX que pertanyen a la gamma baixa on el format de les instruccions és de 12 bits; els PIC12C6XX són de la gamma mitja i tenen 14 bits per instrucció. Els models PIC12F6XX contenen memòria FLASH destinada a guardar el programa i memòria EEPROM per a les dades.

Els PIC nans són emprats en aplicacions de control de personal, en sistemes de seguretat i en dispositius de baix consum que gestionen receptors i transmissors de senyals. Aquest microcontrolador és adequat per a projectes on la reduïda mida sigui una de les principals qualitats.



- **Gamma baixa o bàsica: PIC16C5X amb instruccions de 12 bits**

Aquesta família és una de les que ofereix una millor relació preu-prestacions però, per contra, els seus recursos són limitats. Els diferents models poden estar encapsulats amb 18 i 28 pins i poden ser alimentats a partir d'una tensió de 2,5 V per tant són molt adequats en les aplicacions que sigui necessari piles. El seu format és de 12 bits i el seu llistat d'instruccions conté un total de 33. Les desavantatges d'aquesta sèrie és que no permet cap tipus d'interrupció i la Pila només disposa de dos nivells.

Dins de la gamma baixa trobem el PIC14000 el qual incorpora un convertidor A/D (de 10 a 16 bits), un sensor intern de temperatura, comunicació sèrie I<sup>2</sup>C i molt baix consum. Són utilitzats pel control de carregadors de bateries, monitors de l'estat de piles i bateries, sistemes d'alimentació ininterrompudament (SAI) i sistemes d'adquisició de dades, especialment de temperatura.

El PIC16C5X, representant del de la gamma baixa, sobretot s'utilitza a la indústria de l'automoció en dispositius d'elevada velocitat, en el control de motors, receptors/transmissors i processadors de baix consum encarregats de la gestió de comunicació.

La subfamília PIC16C55X disposa d'una pila de 8 nivells i diverses fonts d'interrupció, recursos propis de la gamma mitja.

- **Gamma mitja: PIC16CXXX amb instruccions de 14 bits**

És la gamma més variada i completa dels PIC. Aquesta disposa de models amb encapsulats des de 18 fins a 68 pins. El nombre d'instruccions que executa és 35 sent cadascuna de 14 bits i compatible amb la gamma baixa. Disposen d'interrupcions i una Pila de 8 nivells la qual cosa permet registrar subrutines.

Aquest grup de microcontroladors es pot classificar en subfamílies de la següent manera:

- Gamma mitja estàndard PIC16C55X
- Gamma mitja amb comparador analògic PIC16C62X/64X/66X
- Gamma mitja amb mòduls de captura (CCP), modulació d'amplada d'impulsos (PWM) i port sèrie PIC16C6X
- Gamma mitja amb CA/D de 8 bits PIC16C7X

- Gamma mitja amb CA/D de pressió PIC14000
- Gamma mitja amb memòria FLASH i EEPROM PIC16X8X
- Gamma mitja amb driver LCD PIC16C92X

Dins d'aquesta gamma es troba els PIC14C000 el qual és apte per al disseny de controladors intel·ligents per a carregadors de bateries, piles petites, SAI, i qualsevol sistema d'adquisició i processament de senyals que requereixin gestió de l'energia d'alimentació.

Els PIC16C62X és molt recomanable en aplicacions de gestió de carregadors de bateries, control de sensors a distància amb baix consum, etc.

Els PIC16C6X incorpora nombrosos recursos: diversos temporitzadors, mòdul de captura/comparació, port sèrie SPI i I<sup>2</sup>C, port paral·lela PSP, etc.

Els PIC16C7X funciona a 20 MHz, amb un cicle d'instrucció de 200 ns, per això són molt apreciats en els sistemes d'alarma i seguretat, i en el control a distància de sensors a l'automoció.

Els PIC16X8X incorporen memòria EEPROM o FLASH per l'emmagatzematge del programa. Les quals permeten ser gravades i esborrades elèctricament, en sèrie, tantes vegades com es vulgui, sent per tant, aquests dispositius adequats per al procés de disseny de l'aplicació. Així doncs són molt emprats en el control de portes de garatge, instrumentació, immobilitzadors de vehicles, targetes codificades, petits sensors, etc. També són recomanats per l'emmagatzematge de dades de calibratge i per a la modificació del programa al variar les condicions de l'entorn ja que la gravació es realitza en el mateix circuit.

Els PIC16C9XX incorporen un controlador programable de LCD el qual gestiona qualsevol pantalla de cristall líquid.

Una altra sèrie de microcontroladors RISC FLASH que s'adapta a la gamma mitja és la PIC16F87X (3,4,5,6 i 7). Aquest models són més potents que el PIC16F84 però manté les mateixes facilitats de gravació i esborrament elèctric.

Aquests PIC tenen una capacitat de memòria de codi de 8K paraules de 14 bits en memòria FLASH, 386 x 8 bytes de memòria RAM i 256 x 8 bytes de memòries de dades EEPROM. Poden ser programats amb +5 VDC i es pot accedir a la lectura i escriptura de la memòria de programa. A més disposen de fins a 3 Timer o comptadors, 2 mòduls de

captura, comparació i PWM, un convertidor A/D de 10 bits, canal de comunicacions síncron SSP amb mode I<sup>2</sup>C i SPI, canal USART/SCI, port paral·lel slave asíncron de 8 bits i detecció de fallada d'alimentació (BOR).

- **Gamma alta: PIC17CXXX amb instruccions de 16 bits**

El nombre d'instruccions que pot realitzar és de 58 de 16 bits cadascuna. Aquests, contenen un sistema de gestió de les interrupcions vectoritzades molt potents, diversos controladors de perifèrics, portes de comunicació sèrie i paral·lel amb elements externs, i un multiplicador hardware de gran velocitat.

La característica més rellevant d'aquesta família és la seva arquitectura oberta, és a dir, la capacitat de poder ampliar el microcontrolador amb la incorporació d'elements externs. Per la qual cosa el microcontrolador treu a l'exterior mitjançant els pins les línies de busos de dades, direccions i control, a les quals es connecten memòries o controladors de perifèrics.

Dins d'aquesta gamma trobem els PIC17C4X amb 58 instruccions de 16 bits que són executades cadascuna en 121 nanosegons, excepte les de salt que es realitzen el doble de temps, quan treballen a una freqüència de 33MHz. Aquests disposen d'una Pila de 16 nivells, un major nombre de fonts d'interrupcions que els de la gamma mitja, un multiplicador hardware de 8x8 bits en un cicle per aplicacions que requereixin un intensiu càlcul matemàtic. Són ideals per aplicacions a la indústria que precisin una elevada velocitat i càlculs complexos, instrumentació i telecomunicacions.

El PIC17C752 i el PIC17C756 són els més representatius de la gamma alta i estan encapsulats amb 64 i 68 pins respectivament. Consten d'una elevada capacitat de memòria

la qual és de 8K x 16 i 16K x 16 per a la contenció de les instruccions i 454 i 902 bytes reservats a les dades, respectivament. A més incorporen 12 canals per a un convertidor A/D de 10 bits, 4 canals per a un mòdul de captura de 16 bits, 2 USART, bus I<sup>2</sup>C, etc.

A la següent taula podem observar les característiques més rellevants dels PIC's i la diferència entre cadascuna de les gammes d'aquests tipus de microcontroladors.

## 2.2 EL MICROCONTROLADOR PIC16F84

El microcontrolador PIC que tractarem en aquest treball és el PIC16F84 que és el que programarem amb la Micro'PIC Trainer i situarem en el PICBOT per a la avaluació del seu

Gamma	Alimentació	Nº Instruccions	Format	Nº Pins	Línia E/S	Interrup?	Característiques Pila
Nana	2,5 a 5,5 V	33	12 bits	8	6		
		35	14 bits				
Baixa	≥ 2,5 V	33	12 bits	18	4-12	No	2 nivells
				28	20		
Mitja	2 a 6 V	35	14 bits	18 a 68	13	Si	8 nivells
Alta		58	16 bits	40 a 68	35/50	Si	16 nivells

comportament davant diferents aplicacions.

El PIC16F84, com ja hem vist pertany a la gamma mitja, està fabricat amb tecnologia CMOS i és un dels més petits, doncs consta de 18 pins i, encara que, disposa de menys recursos que els altres de la mateixa família, compleix amb tots els requisits per tal de poder desenvolupar projectes.

El fet que el fa ser tan utilitzat és el tipus de memòria de programa que posseeix. Aquesta és de tipus FLASH amb una capacitat de 1K paraules de 14 bits cadascuna.

La memòria FLASH com la EEPROM són elèctricament gravables i esborrables, fet que permet gravar i esborrar el microcontrolador, sense treure'l del sòcol de la Micro'PIC Trainer, amb el mateix dispositiu tantes vegades com sigui necessari, sobretot, en el període de disseny del projecte. Aquesta propietat també es veu afavorida pels microcontroladors amb memòria EPROM ja que permeten ser reprogramats elèctricament, però, en canvi, per

ser esborrats s'han de exposar durant un determinat temps a raigs ultraviolats, per tant, s'han de treure del sòcol i posar-los en un esborrador de EPROM.

La memòria FLASH ofereix una major velocitat i un menor consum en front a la EEPROM, però, per contra, la EEPROM pot ser gravada/esborrada 1milió de vegades mentre que la FLASH suporta mil cicles d'escriptura/esborrada.

Les característiques i dades més remarcables del PIC16F84 són les següents:

- **Memòria de programa:** 1K x 14, FLASH
- **Memòria de dades RAM:** 68 bytes
- **Memòria de dades EEPROM:** 64 bytes
- **Pila (Stack):** de 8 nivells
- **Interrupcions:** 4 tipus diferents
- **Jocs de instruccions:** 35
- **Encapsulat:** plàstic DIP de 18 pins
- **Freqüència de treball:** 10 MHz màxim
- **Temporitzadors:** només un, el TMR0. També té gos guardià (WDT)
- **Línies d'E/S digitals:** 13 (5 Port A i 8 Port B)
- **Corrent màxim absorbit:** 80 mA Port A i 150 mA Port B
- **Corrent màxim subministrat:** 50 mA Port A i 100 mA Port B
- **Corrent màxim absorbit per línia:** 25 mA
- **Corrent màxim subministrat per línia:** 20 mA
- **Voltatge d'alimentació ( $V_{DD}$ ):** de 2 a 6 VDC
- **Voltatge de gravació ( $V_{PP}$ ):** de 12 a 14 VDC

### 2.2.1 ASPECTE EXTERN

L'encapsulat del microcontrolador PIC16F84 és el que es pot observar a la següent figura on el funcionament o missió de cadascun dels pins es desenvolupa a continuació:

- **$V_{DD}$ :** pin (14) pel qual s'aplica al microcontrolador la tensió positiva d'alimentació.

- **Vss** : pin (5) connectat a massa o negatiu de l'alimentació.
- **OSC1/CLKIN**: pin (16) a través del qual s'aplica l'entrada del circuit oscil·lador extern que proporciona la freqüència de treball del microcontrolador.
- **OSC2/CLKOUT**: pin (15) auxiliar del oscil·lador.
- **MCLR#**: aquest pin (4) s'activa a nivell lògic baix '0', per tant es simbolitza amb #. L'activació d'aquest pin provoca la reinicialització o el Reset del microcontrolador.

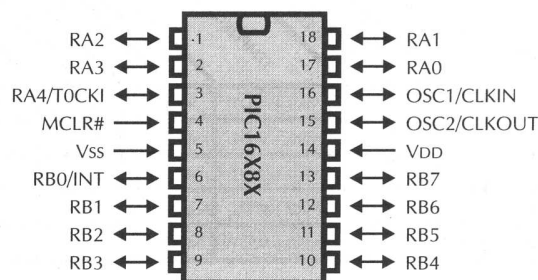
Aquest pin també és emprat durant la gravació de la memòria de programa per introduir per ella la tensió Vpp.

- **RA0-RA4**: pins (17, 18, 1, 2 i 3, respectivament) per on es connecten les 5 línies d'E/S digitals corresponents al Port A.

El pin 3 pot realitzar dues funcions com es pot veure, línia RA4 o T0CKI la qual rep una freqüència externa per alimentar al temporitzador intern TMR0.

- **RB0-RB7**: pins (6, 7, 8, 9, 10, 11, 12 i 13, respectivament) que corresponen a les 8 línies d'E/S digitals del Port B.

El pin 6 multiplexa amb dues funcions, RB0 i INT que serveix com entrada de petició d'interrupció externa.



### 2.2.1.1 FREQUÈNCIA DE FUNCIONAMENT

La freqüència de treball és el paràmetre que determina la velocitat d'execució de les instruccions i del consum d'energia.

Al microcontrolador PIC16F84 les instruccions s'executen en 4 períodes de rellotge, per tant el cicle d'instrucció serà 4 x període. Totes les instruccions es realitzen en un cicle d'instruccions menys les de salt, que triguen el doble.

A través de l'entrada OSC1/CLKIN del PIC els impulsos de rellotges generats externament s'apliquen a aquest on es divideixen en 4 senyals que realitzen les següents operacions, durant un cicle d'instrucció:

1.- Durant aquest primer impuls es genera un increment al comptador de programa.

2-3.- Durant aquests dos impulsos es produeix la descodificació i l'execució de les instruccions.

4.- En aquest impuls es busca el codi de la instrucció a la memòria de programa i es carrega al registre.

El processador utilitza la tècnica de segmentació o "*pipe-line*" per tal de executar cadascuna de les instruccions en un cicle d'instrucció, que consisteix en efectuar en paral·lel les dues fases de que consten cadascuna de les instruccions. Aquestes dues fases d'una mateixa instrucció es realitzen, en realitat, en dos cicles d'instruccions, al primer s'efectua la busca del codi de la instrucció a la memòria del programa i, al segon, es realitza la descodificació i l'execució. Així doncs, aquest sistema de segmentació permet que en un mateix cicle es produeix l'execució d'una instrucció i la busca del codi de la instrucció a la memòria de la següent instrucció.

Quan la instrucció que s'executa és una instrucció de salt no es sap quina és la següent instrucció a executar fins que no s'hagi executat completament, per tant no es podrà efectuar la busca del codi de la següent instrucció deixant aquest cicle en "buit".

### **Tipus d'oscil·ladors**

El microcontrolador PIC admet quatre tipus d'oscil·ladors externs per l'aplicació de la freqüència de treball. La especificació del tipus d'oscil·lador que es vol utilitzar es determina per mitjà de dos bits, FOSC1 i FOSC0, de la paraula de configuració.

Els tipus d'oscil·ladors són:

- **Oscil·lador "RC":**

És un oscil·lador constituït per una simple resistència i un condensador. Aquest oscil·lador de baix cost proporciona una no bona estabilitat de la freqüència el valor de la qual ve determinada pel valor dels components R-C.

- **Oscil·lador “HS”:**

Aquest aconseguix una elevada velocitat compresa entre 4 i 10 MHz i es basa en un cristall de quars o ressonador ceràmic.

- **Oscil·lador “XT”:**

Es tracta d'un oscil·lador de cristall o ressonador per a freqüències estàndards compreses entre els 100 KHz i els 4 MHz.

- **Oscil·lador “LP”:**

Consisteix en un oscil·lador de baix consum amb cristall o ressonador on el seu rang de freqüències de treball és de 35 a 200 KHz.

El cristall de quars és ubicat entre els pins OSC1 i OSC2 del PIC.

### 2.2.1.2 REINICIACIÓ O RESET

El pin encarregat d'aquestes dues operacions, com ja hem vist amb anterioritat, és el MCLR#, així doncs quan apliquem un baix nivell lògic, "0", a aquest, el microcontrolador reinicialitza el seu estat. Tant quan es produeix un Reset o uan reinicialització es produeixen dos fets:

1. El comptador de programa es carrega amb la direcció 0, assenyalant la primera direcció de la memòria de programa on haurà d'estar situada la primera instrucció del programa de l'aplicació.
2. La majoria dels registres d'estats i control del processador adquireixen un valor conegut i determinat.

### 2.2.1.3 ELS PORTS D'E/S

El PIC16F84 només disposa de dos Ports d'E/S. El Port A que consta de 5 línies, RA0-RA4, on una d'aquestes línies suporta dues funcions la RA4/T0CKI, és a dir, pot funcionar com una línia d'E/S o com una entrada d'impulsos que ha de rebre



el Timer ,TMR0. El Port B que conté 8 línies, RB0-RB7, el qual també té una línia multiplexada que pot efectuar les funcions RB0/INT, una línia d'E/S i una línia per on s'envien els impulsos quan es genera una interrupció externa.

Les línies d'E/S es poden configurar com entrades o com sortides de manera independent segons es posi un nivell lògic alt "1" o un nivell lògic baix "0", respectivament, el bit del registre de configuració de cadascun dels Ports, TRISA i TRISB. Els registres que emmagatzemen la informació d'entrada i de sortida dels Ports són els PORTA i PORTB i ocupen les direccions 5 i 6 del banc 0 de la memòria de dades. Els registres de configuració TRISA i TRISB ocupen també les direccions 5 i 6 però del banc 1.

### **3. MICRO'PIC TRAINER**

#### **3.1 COMPONENTS**

El micro'PIC Trainer és una tarja fabricada per Microsystems Enginering que permet la programació del Pic.

La tarja consta de:

Un gravador de PIC

Hardware i software d'adaptació al PC

Programa de comunicacions

Targeta que inclou la majoria del perifèrics típics: interrupcions, LED's, display de set segments, potenciómetres per a senyals analògiques i pantalla LCD

El micro'PIC Trainer és un sistema que permet la realització d'una gran quantitat d'aplicacions i projectes basats en el microcontrolador PIC de gamma mitja (18 a 24 pins) de MICROCHIP. Aquest equip incorpora una sèrie d'E/S que permeten la connexió amb diversos perifèrics per a la comprovació del funcionament PIC.

- L'alimentació de l'entrenador procedeix des d'un transformador de 12VAC o des d'una pila de 9VDC. Inclou un circuit estabilitzador amb el qual s'obté una tensió general de +5V (Vcc) i la de gravació de +13.8V.

- Pot treballar amb qualsevol microcontrolador de 18 o 28 pins de la gamma mitja. La freqüència de treball és de 4MHz per defecte la qual és pot modificar canviant el cristall de quars.

- La generació del RESET pot ser automàticament o manual per mitjà d'un polsador accionat per l'usuari.

- Consta de:

- 5 entrades digitals, les quals són implementades mitjançant 5 interruptors.

- 4 entrades de tensió analògica, les quals poden ser variades per mitjà de potenciòmetres.

- 8 sortides digitals connectades a 8 indicadors lluminosos tipus LED i/o display de set segments. Per mitjà de jumpers es selecciona un o altre.

- Connexió d'un mòdul de visualització tipus LCD de 2 ×16 (dos línies de 16 caràcters cadascuna) i de control del contrast de la LCD per mitjà de potenciòmetres.

- Connector d'expansió PIC-BUS el qual connecta totes les línies d'E/S del microcontrolador per tal de poder ser utilitzades per altres perifèrics dels quals no disposa el Micro'PIC Trainer.

- Els perifèrics d'entrada i de sortida poden ser activat i desactivats a través dels respectius jumpers. Es desactivaran tots aquells que no s'utilitzin en la avaluació d'una determinada aplicació amb la finalitat de reduir la càrrega elèctrica en les línies E/S del PIC.

- La senyal RA4/T0CKI pot ser emprada com a entrada o sortida de senyals digitals o bé com a entrada de polsos de rellotge per al TIMER 0 des de l'exterior.

- La senyal RB0/INT pot ser utilitzada com a d'entrada o sortida digital o bé com a entrada d'interruptió externa.

- Circuit gravador. Amb el software de control necessari permet gravar el PIC que estigui inserit en el sòcol. Aquesta funció és molt útil per als models

16C84 i 16F84 ja que disposen de memòria EEPROM i Flash respectivament, les quals poden ser gravades i regravades sense necessitat de treure el PIC del sòcol.

- Led indicador de presència de la tensió de gravació  $V_{pp}$ .
- Connector DB25 per a la connexió en paral·lel amb qualsevol PC.
- Software per a l'execució de les instruccions de gravació: lectura, gravació, verificació, esborrat, edició, etc.

### **3.2 DESCRIPCIÓ PER BLOCS DEL MICRO'PIC TRAINER**

#### **Font d'alimentació**

La funció de la font és obtenir els valors de tensió necessaris de treball: +5V de tensió  $V_{cc}$  i +13.8V per a la gravació.

La tensió d'alimentació  $V_{cc}$  s'obté a partir de dues piles de 9V o bé d'un transformador de 12VAC connectat al jumper1 (J1) la qual es rectificada a través del pont de diodes D1 i es filtra per mitjà dels condensadors C1 i C2 aconseguint d'aquesta manera una tensió continua i sense estabilitzar.

La tensió continua s'estabilitza a +5V mitjançant el regulador U6, uA7805, i per mitjà del regulador uA7812 i dels tres diodes D2, D3 i D4, 1N4007, s'aconsegueix la estabilització de la tensió a 13.8V.

#### **El microcontrolador**

Com ja sabem el micro'PIC Trainer pot treballar amb els PIC de gamma mitja de 18 o 28 pins per tant l'entrenador consta de dos sòcols.

Entre els dos models existeixen diferències que fan que la forma d treballar amb ells des del punt de mira de micro'PIC Trainer sigui una determinada. Una diferència bàsica a destacar entre els dos models és que el PIC de 18 pins disposa d'un Port A de 5 línies (RA0-RA4) i un Port B de 8 línies (RB0-RB7) en canvi el PIC de 28 pins disposa d'un Port A de 6 línies (RA0-RA5), un Port B de 8 línies (RB0-RB7) i un Port de 8 línies (RC0-RC7).

Al micro PIC Trainer s'empren les línies comuns 5 línies del Port A i 8 línies del Port B. A través del connector PIC-BUS d'expansió l'usuari pot utilitzar les línies de les portes anterior i les del Port C.

La freqüència de treball per a tots dos models ve determinada pel cristall de quars X1 i dels condensadors C6 i C7 la qual és per defecte 4MHz. Aquesta freqüència podrà ser modificada canviant el valor del cristall i dels condensadors segons les característiques tècniques de cadascuns dels models.

### **Entrades digitals**

Aquestes estan formades per 5 interruptors (SW2-SW6) connectats a les línies RA0-RA4 del Port A els quals generen nivells lògics "1" i "0" per aquestes línies que poden ser programades per tal d'efectuar altres funcions. Les línies RA0-RA3 poden actuar com a entrades analògiques i la RA4 com a entrada externa de polsos de rellotge per al TMR0 (T0CK).

Cadascuna de les línies del Port A que s'utilitzen en el Trainer té uns jumpers associats per mitjà dels quals es pot seleccionar el tipus d'entrada que volem obtenir per a la execució de les diferents aplicacions. Així doncs, els jumpers J8(0)-J8(3) seleccionen si per les línies RA0-RA3, respectivament, les entrades seran digitals introduïdes pels interruptors (SW2-SW6), seran analògiques generades per els potenciòmetres P2-P5 o bé si les línies es desactivaran dels interruptors i dels potenciòmetres podent-les emprar com a línies d'E/S per a altres perifèrics.

Per mitjà del J9 es selecciona la línia RA4 com a entrada digital proporcionada pel interruptor SW6, com a entrada de senyal de rellotge per al TMR0 (T0CK) la qual es aplicada des de l'exterior per mitjà del connector J2(1) o bé es desactiva la línia per a la seva utilització des de altres perifèrics.

### **Entrades analògiques**

Aquestes estan constituïdes per 4 resistències variables o potenciòmetres (P2-P5) que s'alimenten a +5V. Amb la variació de l'eix del potenciòmetres s'obté una tensió variable entre 0 i 5V pels terminals centrals d'aquests.

La tensió variable existent en els potenciómetres P2-P5 és transmesa als jumpers J8(0)-J8(3), respectivament. Per tant els jumpers aplicaran aquesta tensió variable a les línies RA0-RA3 en el cas que alguna d'aquestes estigui seleccionada com a entrada analògica.

### **Sortides digitals**

Aquestes estan connectades al Port B i es tracten de dos perifèrics, per una banda, d'una barra de diodes LED's lluminosos D8 que representen l'estat lògic de les línies RB0-RB7, i per altra banda, en un display alfanumèric de set segments.

Aquests dos perifèrics estan connectats entre sí en paral·lel al Port B. I a més, els LED's estan connectat al jumper J7 i el display al J5 els quals permeten desactivar-los en cas de no ser emprats a l'aplicació actual amb l'objectiu de disminuir el consum del port B i/o per poder emprar aquestes línies com a altres configuracions per altres perifèrics a través del connector d'expansió PIC-BUS.

El paquet de resistències RPACK 1 conté 8 resistències de  $330\Omega$  la funció de les quals és la de resistències d'absorció o limitació.

La barra de diodes D8 conté 10 diodes LED. Els 8 primers LED's representen el nivell lògic de les senyals RB0-RB7 quan aquestes estan habilitades pel jumper J7, el 9è diode no s'utilitza i el dècim indica de la presència de la tensió d'alimentació +5V, per tant a partir d'aquest LED podem saber si la barra de diodes està activa o no.

La línia RB0 es pot configurar com entrada d'interrupció externa INT. Aquesta senyal d'entrada és aplicada a la línia a través del connector J2(3) i per mitjà del jumper J10 la línia RB0 es connecta als LED's i al display, a l'entrada de interrupció INT o bé es deixa desactivada.

### **Mòdul LCD**

Consisteix en un mòdul de visualització alfanumèric de cristall líquid amb capacitat de representació de 2 línies de 16 caràcters cadascuna.

Per mitjà del corresponents codis es pot representar diferents efectes de visualització com pampallugues, scroll, activació d'un cursor, etc. També es pot generar nous caràcters definits per l'usuari.

Les 8 línies de dades D0-D7 estan connectades amb les 8 línies del Port B RB0-RB7. El Port B, de vegades, actua com a sortida del PIC i entrada cap el mòdul en el qual es visualitza els resultats de l'aplicació amb la utilització d'un codi de control, i d'altres, com a entrada al PIC i sortida del mòdul ja que la LCD retorna per mitjà de codi l'estat intern, el contingut del buffer en memòria, etc.

El mòdul està connectat a les línies RA0, RA1 i RA2 del Port A del PIC. Aquestes línies són emprats com a sortides i destinades a l'enviament de les senyals de control del mòdul, que són les següents:

- RS: aquesta instrucció permet seleccionar el registre que conté la informació de les instruccions a efectuar o el registre que conté les dades a visualitzar o enviar.

RS = 0                    selecció del registre d'instruccions

RS = 1                    selecció del registre de dades

- R/W: aquesta instrucció és la que permet escriure les dades al mòdul LCD o bé llegir les dades introduïdes al mòdul LCD, segons el valor lògic seleccionem:

R/W = 0 cicle d'escriptura sobre la LCD

R/W = 1 cicles de lectura de la LCD

- E: aquesta és una entrada cap al mòdul LCD el qual permet visualitzar o no a la LCD la informació requerida.

E = 0                    mòdul LCD desactivat

E = 1                    mòdul LCD activat

La senyal E està connectada, a través del jumper J6, amb la línia del Port A, RA2. Aquesta línia controlarà la habilitació o no del mòdul LCD quan el J6 estigui tancat. En cas contrari, la senyal E quedarà connectada a massa (nivell "0") a través de la resistència R16, per tant, la LCD sempre estarà desactivada de manera que les línies del Port A, RA0-RA2, i del Port B, RB0-RB7, podran ser utilitzats per altres perifèrics del mateix MicroPIC Trainer o d'altres connectats al PIC-BUS.

### **Circuit gravador**

Aquest circuit gravador és el que permet gravar tots els models de la gamma mitja de PIC's els quals consten de 18 i 28 pins. Els PIC's es situen al sòcol de MicroPIC Trainer on seran manipulats per varis programes

Aquest circuit és el que permet efectuar la programació dels microcontroladors PIC de la gamma mitja de 18 i 28 pins, directament des del sòcol del MicroPIC Trainer. Aquesta facilitat de programació que permet el Trainer s'afavoreix pels models 16C84 i 16F84 ja que disposen de memòria no volàtil, EEPROM i Flash, respectivament, les quals permeten ser gravades i esborrades gairebé infinites vegades per diversos programes i aplicacions, sent, alhora, executades i comprovades sobre el MicroPIC Trainer.

Per mitjà del connector J3 es connecta el canal paral·lel de qualsevol ordinador personal (PC/XT/AT). Un programa encarregat de gestionar totes les operacions (esborrar, verificar, gravar, llegir el contingut, establir la paraula de configuració, etc.) sobre el PIC és el PICME-TR, en castellà i sota MS-DOS. Un altre programa que permet realitzar la gravació és el MPLAP, en anglès i sota entorn Windows.

### **Connector d'expansió**

És un connector de 26 contactes en que estan disponibles totes les senyals de les diferents portes d'E/S, a més a més de les d'alimentació i MCLR. Aquest permet connectar ampliacions de nous perifèrics (expansions) generats i desenvolupats bé per Microsystems Engineering o bé pel propi usuari.

Amb l'objectiu d'evitar possibles conflictes entre els perifèrics connectats al PIC-BUS i els propis del MicroPIC Trainer és necessari desconnectar aquells que vagin a ser reemplaçats per mitjà del seu respectiu jumper.

La connexió de nous perifèrics que siguin requerits per a la comprovació de l'aplicació pot efectuar-se en la placa prototipus PIC18-ME.

## **4. CONNEXIÓ ENTRE PC , MICROPIC-TRAINER I EL PIC**

La connexió entre el PC i el Micro'PIC Trainer es realitza a través del canal paral·lel. El circuit integrat U3-CONTROL #4.0 és el que s'encarrega de verificar el hardware de Micro'PIC Trainer i la comunicació entre aquest i el PC.

La connexió entre el microPIC-Trainer i el PIC és realitza en sèrie.

## **5. PROGRAMACIÓ DEL PIC**

Una vegada ja finalitzat el disseny i desenvolupament del projecte és hora d'efectuar les pertinents proves per a l'avaluació del projecte. Les determinades avaluacions es poden portar a terme per mitjà de programes simuladors del comportament del PIC o bé gravant directament sobre el PIC el programa del projecte a través de la Micro'PIC Trainer.

Un d'aquest programes de simulació és el SIMUPIC'84, de Microsystems Engineering, que permet la verificació del correcte funcionament del PIC16F84 i de la versió PIC16C84 davant l'aplicació.

### **5.1 PASSOS A SEGUIR PER LA PROGRAMACIÓ DEL PIC**

La programació del PIC16F84 la efectuarem fent ús del hardware Micro'PIC Trainer, desenvolupat per Microsystems Engineering, i del software creat per Microchip.

Per efectuar correctament la programació del PIC s'han d'efectuar un seguit de passos els quals es descriuen a continuació:

- 1) En primer lloc s'ha d'escriure el programa de control de l'aplicació en un editor de textos, com el **EDIT** del **MS-DOS**, emprant les instruccions del llenguatge escollit. D'aquesta manera, s'obtindrà el programa font que s'haurà de traduir a llenguatge màquina per tal de poder-lo carregar a la memòria d'instruccions del microcontrolador i executar.



2) Així doncs, en segon lloc, es procedirà a la compilació o del programa font. El programa compilador o ensamblador que utilitzarem serà el **MPASM** de l'empresa Microchip. S'accedirà a ell pel directori **C:\ARCHIV1\SIMUPIC\MPASM**. Una vegada en la pantalla del compilador s'haurà de seleccionar el fitxer de l'aplicació (*aplicació.ASM*), el tipus de PIC (16F84) i a continuació es compila per verificar que el programa és correcte. Aquí ens crea el fitxer amb extensió **.HEX**, i també ens crea altres fitxers necessaris per l'execució del programa dels quals el més important és el fitxer que explica els errors produïts en la compilació.

3) Una vegada obtingut el codi màquina del programa i comprovat que és correcte ja es pot carregar al microcontrolador. La càrrega del codi del programa de l'aplicació a la memòria del microcontrolador PIC16F84 es realitzarà amb el programa **PICME-TR** al qual s'accedeix amb el directori **C:\PIC\PICME-TR**. Estant a la pantalla del programa el primer que s'efectuarà serà obrir l'arxiu de l'aplicació (*aplicació.HEX*). A continuació, amb l'objectiu d'assegurar-nos que el PIC està bé connectat al sòcol del MicroPIC Trainer es realitzarà la lectura del PIC. Seguidament, s'esborrarà el contingut de la memòria d'instrucció del PIC i posteriorment es verificarà el correcte esborrament. I per últim, es procedirà a la gravació del PIC amb el programa de l'aplicació. Per poder efectuar aquesta operació, primer s'hauran de deshabilitar els perifèrics del MicroPIC Trainer traient els corresponents jumpers J5, J6 i J7 i a continuació, seleccionem la opció gravar tot.

En el moment de la gravació podrem observar que s'il·lumina un LED el qual ens indica que la operació de gravat s'està executant.

## **5.2 TRANSFERÈNCIA DEL PROGRAMA DES DEL MICRO PIC-TRAINER AL PIC**

Un cop ensamblat el programa de l'aplicació es procedeix a la gravació del PIC utilitzant la placa Trainer.

La lectura/gravació del PIC es realitza en sèrie. Els bits de dades es llegeixen o escriuen de manera seqüencial, un rera l'altre, a través de la senyal RB7 i, per mitjà dels amplificadors U4D i U4F, són transferides al canal paral·lel per al seu posterior tractament pel software de control.

La senyal RB6 actua com a entrada de rellotge per a sincronitzar la transferència dels bits de dades durant el procés de lectura/gravació del PIC. Aquesta senyal de rellotge es generada pel software de control i s'aplica pel canal paral·lel a través de l'amplificador U4C.

El software de control, a més, genera una altra senyal la qual es connecta al transistor de comunicació Q1 a través de l'amplificador U4B. Aquest transistor, Q1, al activar-se envia els 13.8V generats des de la font d'alimentació al pin MCLR/VPP del PIC el qual s'està llegint o gravant en aquest moment. Alhora el diode LED D5 s'il·lumina indicant activitat al circuit gravador de MicroPIC Trainer.

Finalment, a través de l'amplificador U4A, el software de control genera la senyal MCLR, encarregada d'efectuar el reset i la reinicialització al microcontrolador. Normalment això succeeix al finalitzar qualsevol operació de lectura i/o gravació, per tant el PIC de manera immediata es posarà a executar el programa emmagatzemat a la seva memòria. Una indicació a destacar és el fet que l'usuari, en qualsevol moment i en qualsevol cas, pot provocar un Reset al microcontrolador de manera manual per mitjà de l'accionament del polsador RESET SW1.

Degut a que les senyals que s'utilitzen per a la lectura/gravació d'un PIC són RB6 i RB7 i aquestes estan connectades a la vegada als LED's, displays i mòdul LCD, és important i necessari desconnectar aquests perifèrics mitjançant els jumpers J7, J5 i J6, respectivament, per tal d'evitar qualsevol càrrega a les senyals anomenades i alteració de la informació llegida o gravada.

### **5.3 TUTORIAL MPASM**

Els programes que ensamblarem tindran extensió .ASM, converteix el text en llenguatge màquina, el programa ensamblador MPASM és l'encarregat de fer això, primer

fa una primera passada en la que comprova la correcta sintaxi del programa : instruccions, nom etiquetes..., en la segona passada converteix totes les instruccions en codi màquina.

Passos a seguir:

**Source File:** Nom del programa que es vol ensamblar seguit d' ENTER.

**Processor Type:** tipus de PIC que utilitzarem, en el nostre cas fem servir el 16F84.

**Error File: Yes:** es crea automàticament, conté un llistat dels possibles errors trobats durant la compilació.

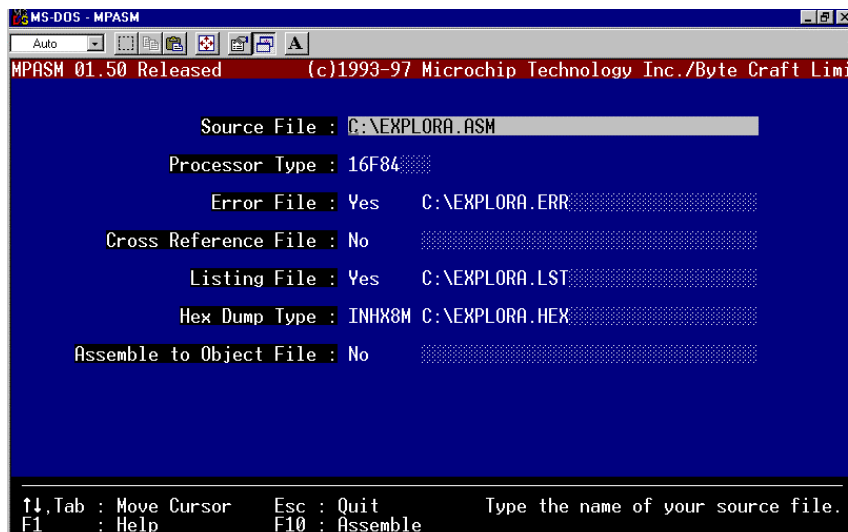
**Cross Reference File: No:** no es crea per defecte, si es volgués crear hauríem de polsar YES sobre el mateix, aquest fitxer conté un llistat de totes les etiquetes trobades.

**Listing File: Yes:** es crea per defecte, conté el llistat complet de la compilació del programa.

**Hex Dump Type: INHX 8M:** fitxer executable que s'obté per defecte, servirà per la gravació de la memòria del microcontrolador.

**Assemble to Object File: No:** opció desactivada per defecte.

Un cop fet això per compilar el programa polsarem F10, al final ens indicarà si hi ha hagut errors o Warnings.



## 5.4 TUTORIAL PICME-TR

El software que permet realitzar correctament la gravació del PIC rep el nom de **PICME-TR**, es tracta d'una versió en castellà i que treballa sota entorn MS-DOS. Té un entorn gràfic amb finestres i botons que l'usuari anirà seleccionant amb el ratolí segons el que vulgui fer. Es necessari que el PC estigui connectat per el port paral·lel amb el microPIC Trainer i que tots els perifèrics connectats a la Port B a través de les línies RB6 i RB7 estiguin deshabilitats.

### Buffer de memòria

Aquesta finestra apareix a l'esquerra de la pantalla del PC, representa la memòria de programa del microcontrolador escollit o bé la memòria EEPROM/FLASH de dades segons la que estigui activa, es visualitza la informació que es gravarà en el xip i que procedeix d'un fitxer executable d'extensió .HEX, en aquest fitxer es troba ensamblat el programa que volem gravar en el PIC.

També podem trobar les dades de la memòria EEPROM/FLASH del PIC en el cas que aquest no estigui protegit contra lectura, per defecte però, apareix la del programa, si no s'ha llegit res apareix plena del contingut 03FF.

### **Selecció del model**

En aquesta finestra podem escollir el model de PIC amb el que volem treballar, et permet escollir qualsevol model de la gamma mitja de 18 o 28 pins.

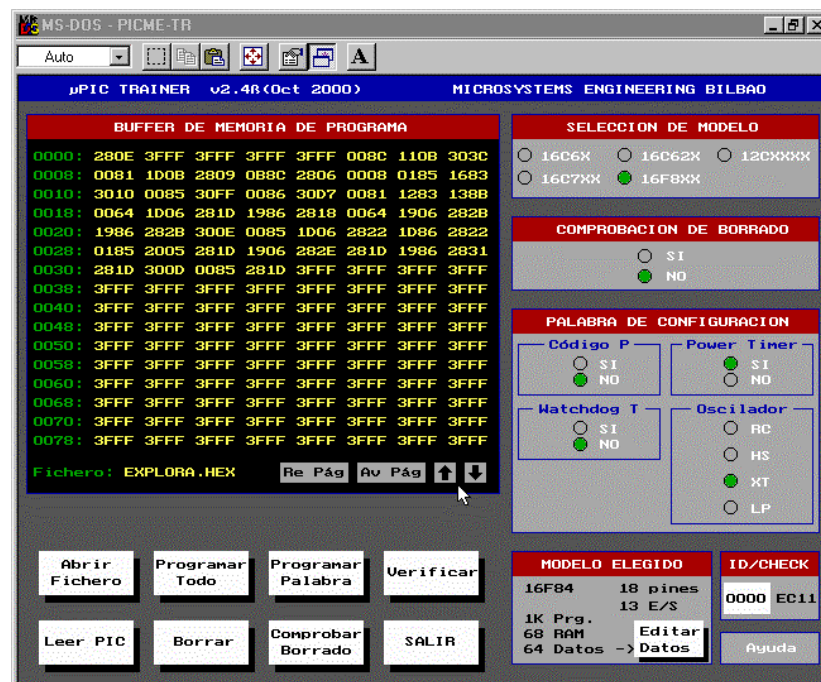
Un cop seleccionat el model en la part inferior de la pantalla apareixerà una finestra amb certa informació del PIC escollit com el nombre de pins, número de línies E/S, dimensió de la memòria de programa, dimensió de la memòria RAM de dades.

A la part inferior esquerra de la pantalla apareixen una sèrie d'opcions entre les que pots escollir:

- Obrir fitxer: Llegeix un fitxer amb extensió .HEX el seu contingut es descarrega a la memòria de dades.
- Llegir PIC: Llegeix el contingut de la EEPROM/FLASH del PIC.
- Programar: Grava el contingut del buffer de memòria de dades sobre les dades de la memòria EEPROM/FLASH del PIC.
- Esborrar: Esborra la memòria EEPROM/FLASH del PIC.
- Verificar: Compara el contingut del buffer de memòria de dades amb el de la EEPROM/FLASH per comprovar que s'hagi gravat correctament.
- Acceptar: la pantalla presenta el mateix aspecte que al començar.

### **Comprovació de l'esborrat**

El programa s'assegura que abans de tornar a programar el PIC, aquest està totalment esborrat.



### Paraula de configuració

Pots seleccionar diferents **configuracions** de treball del PIC:

- Codi P: ( protecció de codi ), seleccionant aquesta opció es fon un fusible intern a l'hora de gravar el xip així s'evita que pugui ser llegit. Si desactivem l'opció el xip pot ser llegit.
- Watchdog : és un temporitzador, s'encarrega de provocar un reset i reinicialitzar el microcontrolador si el software de l'aplicació no s'encarrega de refrescar-lo cada cert temps. Si seleccionem aquesta opció el software del sistema haurà d'anar-se'n refrescant per evitar el reset.

- Power Timer : és un temporitzador que quan s'activa el procés de reinicialitzar el microcontrolador triga uns 70 ms en realitzar-se per donar temps així a que s'estabilitzi la tensió d'alimentació.

- Oscil·lador: permet escollir entre 4 oscil·ladors diferents:

**RC:** econòmic, esta estabilitzat externament mitjançant una xarxa RC, no es recomana per a freqüències superiors a 4MHz.

**HS:** alta velocitat, l'oscil·lador intern s'estabilitza amb un cristall de quars, màxima velocitat de 20MHz però augmentem el consum.

**XT:** velocitat estàndard, l'oscil·lador s'estabilitza amb un cristall de quars de fins 4MHz.

**LP:** baix consum, la velocitat màxima és de 200KHz.

## **6. APLICACIONS**

Un cop programat el PIC el Trainer permet realitzar-ho proves, simulant entrades (amb switch) i sortides (LCD, leds i display de 7 segments).

El PIC ja programat es treu del Trainer i es connecta a al dispositiu que ha de governar.

En primer lloc per familiaritzar-nos amb la programació i amb la Trainer hem realitzat alguns exercicis de prova sobre aquesta.

El primer exercici consistia en mostrar pels led's el valor lògic introduït per les entrades lògiques connectats al port A.

Un segon exercici consistia en mostrar diferents missatges a través del LCD, cada cert interval de temps.

Un tercer exercici consistia en un número entre 1 i 6 aleatòriament i mostrar-lo emprant el display de set segments.

Posteriorment hem passat a programar el PIC perquè el PICBOT realitzi diferents tasques.

Degut a la complexitat del control del PICBOT i ja que moltes línies de programació estan repetides, es fa necessari la utilització de llibreries (extensió **.INC**), la primera d'aquestes llibreries fa la definició dels registres (el seu direccionament):

P16F84.inc

NOLIST

W	EQU	H'0000'
F	EQU	H'0001'
INDF	EQU	H'0000'
TMR0	EQU	H'0001'
PCL	EQU	H'0002'
STATUS	EQU	H'0003'
FSR	EQU	H'0004'
PORTA	EQU	H'0005'
PORTB	EQU	H'0006'
EEDATA	EQU	H'0008'
EEADR	EQU	H'0009'
PCLATH	EQU	H'000A'
INTCON	EQU	H'000B'
OPTION_REG	EQU	H'0081'
TRISA	EQU	H'0085'
TRISB	EQU	H'0086'
EECON1	EQU	H'0088'
EECON2	EQU	H'0089'
GIE	EQU	H'0007'
EEIE	EQU	H'0006'
T0IE	EQU	H'0005'
INTE	EQU	H'0004'
RBIE	EQU	H'0003'
T0IF	EQU	H'0002'



INTF EQU H'0001'  
RBIF EQU H'0000'

NOT\_RBPU EQU H'0007'  
INTEDEG EQU H'0006'  
T0CS EQU H'0005'  
T0SE EQU H'0004'  
PSA EQU H'0003'  
PS2 EQU H'0002'  
PS1 EQU H'0001'  
PS0 EQU H'0000'

IRP EQU H'0007'  
RP1 EQU H'0006'  
RP0 EQU H'0005'  
NOT\_T0 EQU H'0004'  
NOT\_PD EQU H'0003'  
Z EQU H'0002'  
DC EQU H'0001'  
C EQU H'0000'

EEIF EQU H'0004'  
WRERR EQU H'0003'  
WREN EQU H'0002'  
WR EQU H'0001'  
RD EQU H'0000'

LIST

La segona llibreria conté les macros dels moviments que realitzarà el PICBOT, com avançar, retrocedir, parar i girar tant a dreta com esquerra.

MANIO.inc

```
ADELANTE    macro
            movlw    b'00001101'
            movwf    PORTA
            endm
```

```
ATRAS       macro
            movlw    b'00001110'
            movwf    PORTA
            endm
```

```
DERECHA     macro
            movlw    b'00001111'
            movwf    PORTA
            endm
```

```
IZQUIERDA  macro
            movlw    b'00001100'
            movwf    PORTA
            endm
```

```
STOP       macro
            clrf     PORTA
            endm
```

Totes les aplicacions fan servir aquestes dues llibreries, per fer-ho s'han d'introduir les següents línies de codi:

```

INCLUDE "P16F84.INC"
INCLUDE "MANIO.INC"

```

## 6.1 RASTREJADOR

Permet al PICBOT seguir una trajectòria marcada al terra a través d'una cinta negra, mitjançant els sensors òptics de reflexió situats a la part inferior davantera del PICBOT. Si durant el recorregut algun dels bumpers detecta un obstacle el PICBOT es para.

```

LIST P=16F84          ;tipus de processador
LIST C=132           ;nº de caràcters per línia
LIST N=66 ;         ;nº de línies per pàgina
INCLUDE "P16F84.INC"
INCLUDE "MANIO.INC"

#DEFINE J0          PORTB,0          ;sensor de posició 1
#DEFINE J1          PORTB,1          ;sensor de posició 2
#DEFINE J2          PORTB,2          ;sensor d'obstacle dret
#DEFINE J3          PORTB,3          ;sensor d'obstacle esquerra

ORG 0X00
GOTO RESET          ;vector de reset
ORG 0X05

RESET               CLRf          PORTA
                   BSF          STATUS,RP0          ;selecció pàgina 1
                   MOVLW        b'00010000'
                   MOVWF        TRISA          ;RA0-RA3 out, RA4 in
                   MOVLW        b'11111111'
                   MOVWF        TRISB          ;RB0-RB3 in
                   MOVLW        B'11110000'
                   MOVWF        OPTION_REG          ;RA4 clock del TMR0

```

```

        BCF      STATUS,RP0      ;pàgina 0
        BCF      INTCON,GIE      ;interrupció en OFF
;s'espera a que s'accioni qualsevol bumper devanter per iniciar la seqüència

ESPERA      CLRWDT                ;situación inicial de parada
            STOP                  ;activat el sensor dret?
            BTFSS     J2          ;si, inici
            BTFSC     J3          ;activat el sensor esquerra?
            GOTO      ESPERA      ;si, inici

;programa principal

BUCLE      CLRWDT
            BTFSS     J2          ;activat el sensor dret?
            GOTO      PARADA      ;si, parada
            BTFSC     J3          ;activat el sensor esquerra?
            GOTO      NO_OBSTACULO ;no, marxa

PARADA      STOP                  ;parar motors
            GOTO      BUCLE

NO_OBSTACULO BTFSC     J1          ;ha sortit per l'eaquerra?
            GOTO      NO_IZQUIERDA ;no
            DERECHA          ;si, moviment dreta
            GOTO      BUCLE

NO_IZQUIERDA BTFSC     J0          ;ha sortit per la dreta?
            GOTO      NO_DERECHA  ;no
            IZQUIERDA          ;si, moviment esquerra
            GOTO      BUCLE

NO_DERECHA  ADELANTE          ;marxa

```

GOTO BUCLE

END

## 6.2 LAPA

Permet que el PICBOT segueixi a un objecte en moviment. Primer avança fins que topa amb algun objecte (els bumpers frontals s'activen), s'atura fins que un dels bumpers es desactiva, si es desactiva el bumper dret girarà cap a l'esquerra, si ho fa l'esquerra girarà cap a la dreta i si tots dos estan desactivats el PICBOT avança fins tornar a topar amb l'objecte. Per poder realitzar el seguiment correctament és necessari que l'objecte en moviment sigui més lent que el PICBOT.

```
LIST P=16F84           ;tipus de processador
LIST C=132             ;nº de caràcters per línia
LIST N=66              ;nº de línies per pàgina
INCLUDE "P16F84.INC"
INCLUDE "MANIO.INC"
#DEFINE J0 PORTB,0     ;sensor de posició 1
#DEFINE J1 PORTB,1     ;sensor de posició 2
#DEFINE J2 PORTB,2     ;sensor d'obstàcle dre
#DEFINE J3 PORTB,3     ;sensor d'obstàcle esquerra
;programa principal

                ORG     0x00
                GOTO    RESET           ;vector de reset
                ORG     0x05

RESET          CLRF     PORTA
```

```

BSF      STATUS,RP0      ;selecció de pàgina 1
MOVLW   b'00010000'
MOVWF   TRISA            ;RA0-RA3 out, RA4 in
MOVLW   b'11111111'
MOVWF   TRISB            ;RB0-RB3 entrades
MOVLW   b'11010111'     ;prescaler de 256 per el TMR0
MOVWF   OPTION_REG      ;Fosc/4 clock del TMR0
BCF     STATUS,RP0      ;pàgina 0
BCF     INTCON,GIE      ;interrupcions OFF

```

; s'espera a que s'accioni qualsevol bumper devanter per iniciar la seqüència

```

ESPERA   CLRWDT
          STOP            ;situació inicial de parada
          BTFSS   J2      ;sensor dret ON?
          GOTO    BUCLE   ;si, inici
          BTFSC   J3      ;sensor esquerra ON?
          GOTO    ESPERA  ;si, inici

```

;bucle principal

```

          ADELANTE       ;inici marxa endavant
BUCLE    CLRWDT
          BTFSC   J2      ;sensor obstacle dret ON?
          GOTO    NO_OBSTACULO ;no
          BTFSC   J3      ;sensor esquerra ON?
          GOTO    NO_OBSTACULO ;no
          STOP
          GOTO    BUCLE

```

```

NO_OBSTACULO  BTFSC          J2          ;moure a la dreta?
               GOTO          NO_DERECHA ;no
               DERECHA          ;si, moviment dreta
               GOTO          BUCLE

NO_DERECHA    BTFSC          J3          ;moure a l'esquerra?
               GOTO          NO_DERECHA ;no
               IZQUIERDA        ;si, moviment a l'esquerra
               GOTO          BUCLE

NO_IZQUIERDA ADELANTE          ;avanç
               GOTO          BUCLE

               END

```

### 6.3 EXPLORADOR

El PICBOT es desplaça fins a trobar algun obstacle. Si topa lateralment girarà cap al sentit contrari, si topa frontalment realitzarà un retrocés.

```

               LIST          P=16F84      ;tipus de processador
               LIST          C=132        ;nº de caràcters per línia
               LIST          N=66         ;nº de línies per pàgina

               INCLUDE       "P16F84.INC"
               INCLUDE       "MANIO.INC"

#DEFINE       J0            PORTB,0      ;sensor de posició 1
#DEFINE       J1            PORTB,1      ;sensor de posició 2
#DEFINE       J2            PORTB,2      ;sensor d'obstacle dret

```

```

#DEFINE      J3          PORTB,3          ;sensor d'obstacle esquerra

DELAY_1      EQU        0x0C             ;variable temporal
; Inici del programa

                ORG      0x00
                GOTO     RESET           ;vector de reset
                ORG      0X05

```

;DELAY50MS: rutina de temporizació que temporitza 50 ms tantes vegades como indiqui el valor de la variable DELAY\_1. Treballant a una freqüència de 4MHz, el TMR0 s'incrementa cada microsegon. Amb un preescaler de 256, si el TMR0 es carregat amb el valor 195, que en hex equival a C, s'aconsegueix un retras de 50ms.

```

DELAY50MS    MOVWF      DELAY_1

DELAY50MS_2  BCF        INTCON,T0IF      ;borra el flag T0IF
              MOVLW     0x3C
              MOVWF     TMR0             ;complement hex del valor 195

DELAY50MS_1  BTFSS     INTCON,T0IF      ;TMR0 ha rebassat?
              GOTO     DELAY50MS_1      ;encara no
              DECFSZ    DELAY_1,F        ;decrementa la variable Delay_1

              GOTO     DELAY50MS_2      ;encara no és 00
              RETURN                    ;ara si

```

;Programa principal

```

RESET        CLRF      PORTA
              BSF      STATUS,RP0       ;selecció pàgina 1

```



```

MOVLW    b'00010000'
MOVWF    TRISA        ;RA0-RA3 out, RA4 in
MOVLW    b'11111111'
MOVWF    TRISB        ;RB0-RB3 in
MOVLW    b'11010111'
MOVWF    OPTION_REG   ;Fosc/4 clock del TMR0
BCF      STATUS,RP0   ;pàgina 0
BCF      INTCON,GIE   ;interrupcions OFF

```

;s'espera a que s'accioni qualsevol bumper

```

ESPERA    CLRWDT
          STOP        ;situació inicial de parada
          BTFSS      J2        ;sensor dret ON?
          GOTO      BUCLE     ;si,inici
          BTFSC     J3        ;sensor esquerra ON?
          GOTO      ESPERA    ;si,inici

```

;Bucle principal

```

BUCLE    LRWDT
          BTFSC     J2        ;sensor dret ON?
          GOTO      NO_OBSTACULO ;no, parada
          BTFSC     J3        ;sensor esquerra ON?
          GOTO      NO_OBSTACULO ;no, arranc

```

```

RETROCESO ATRAS        ;retrocés
          BTFSS     J2        ;desactivat dret?
          GOTO      RETROCESO ;no
          BTFSS     J3        ;desactivat esquerra?
          GOTO      RETROCESO ;no

```

```

STOP                                     ;si, parar motors
CALL      DELAY50MS                       ;temporiza 0.5 segons
GOTO      BUCLE

NO_OBSTACULO  BTFSC      J2                 ;ha sortit per l'esquerra?
GOTO      NO_IZQUIERDA ;no
IZQUIERDA
GOTO      BUCLE                             ;si, moviment dreta

NO_IZQUIERDA BTFSC      J3                 ;ha sortit per la dreta?
GOTO      NO_DERECHA  ;no
DERECHA
GOTO      BUCLE                             ;si, moviment esquerra

NO_DERECHA      ADELANTE                    ;avanç
GOTO      BUCLE

END

```

## 6.4 TRAJECTÒRIA

El PICBOT es desplaça seguint una trajectòria determinada, sempre que no topi amb cap objecte, i repetirà la seqüència de moviments successivament.

La seqüència és la següent:

- 1.- Avançar cap endavant 0.25 metres.
- 2.- Girar a la dreta i avançar 0.5 metres.

- 3.- Girar a l'esquerra i avançar 0.3 metres.
- 4.- Girar a l'esquerra i avançar 0.3 metres
- 5.- Retrocedir 1 metre.
- 6.- Avançar cap endavant 1 metres.
- 7.- Girar a l'esquerra i avançar 0.5 metres.

```

LIST      P=16F84          ;tipus de processador
LIST      C=132           ;nº de caracters per linia
LIST      N=66            ;nº de linies per pàgina
INCLUDE   "P16F84.INC"    ;definició de registres
INCLUDE   "MANIO.INC"     ;inclusió de macros

#DEFINE   J0              PORTB,0      ;sensor de posició 1
#DEFINE   J1              PORTB,1      ;sensor de posició 2
#DEFINE   J2              PORTB,2      ;sensor d'obstàcle dret
#DEFINE   J3              PORTB,3      ;sensor d'obstàcle esquerra
#DEFINE   J4              PORTB,4      ;encoder

          ORG              0x00
          GOTO             RESET        ;vector de reset
          ORG              0x05

;avanç de tants passos como indiqui W

AVANCE    MOVWF            TMR0         ;carrega el timer
          ADELANTE         ;avança
AVANCE_1  BTFSS           INTCON,T0IF  ;s'han realitzat els passos?
          GOTO             AVANCE_1    ;no, esperar
          BCF              INTCON,T0IF  ;si, desactivar flag
          RETURN

```

;retrocedir tants passos como indiqui W

```
RETROCEDE      MOVWF    TMR0      ;carregar el timer
                ATRÁS      ;retrocés
RETROCEDE_1    BTFSS    INTCON,T0IF  ; realitzat els passos?
                GOTO      RETROCEDE_1 ;no, esperar
                BCF      INTCON,T0IF  ;si, desactivar flag
                RETURN
```

;gir a l'esquerra tantes passes como indiqui W

```
IZDA           MOVWF    TMR0      ;carregar el timer
                IZQUIERDA  ;gir a l'esquerra
IZDA_1         BTFSS    INTCON,T0IF  ;realizats els passos?
                GOTO      IZDA_1     ;no, esperar
                BCF      INTCON,T0IF  ;si, desactivar flag
                RETURN
```

;gir a la dreta tantes passes como indique W

```
DCHA           MOVWF    TMR0      ;carregar el timer
                DERECHA    ;gir a la dreta
DCHA_1         BTFSS    INTCON,T0IF  ; realitzats els passos?
                GOTO      DCHA_1     ;no, esperar
                BCF      INTCON,T0IF  ;si, desactivar flag
                RETURN
```

```
RESET          CLRF      PORTA
                BSF      STATUS,RP0  ;selecció de pàgina 1
```

```

        MOVLW    b'00010000'
        MOVWF    TRISA            ;RA0-RA3 out, Ra4 in
        MOVLW    B'11111111'
        MOVWF    TRISB            ;RB0-RB3 in
        MOVLW    b'11101000'
        MOVWF    OPTION_REG       ;RA4 clock del TMR0
        BCF      STATUS,RP0       ;pàgina 0
        CLRF     INTCON           ;interrupcions OFF

        ADELANTE                   ;avanç de les rodes
NO_CALIBRADO  BTFSC             J4           ;prova sensor de gir
               GOTO          NO_CALIBRADO
NO_CALIBRADO_1 BTFSS          J4           ;prova sensor de gir
               GOTO          NO_CALIBRADO_1
;espera a que s'accioni qualsevol dels bumpers detectors d'obstacles per iniciar la seqüència

ESPERA       CLRWDT
               STOP            ;situació inicial de parada
               BTFSS          J2           ;sensor dret ON?
               GOTO          BUCLE       ;si, inici
               BTFSC          J3           ;sensor esquerra ON?
               GOTO          ESPERA      ;si, inici

;bucle principal

BUCLE       MOVLW    0xF8           ;avanç 250mm (7 pasos)
               CALL     AVANCE
               MOVLW    0xF0           ;gir a la dreta 16 pasos
               CALL     DCHA
               MOVLW    0xF1           ;avanç 500 mm (14 pasos)
               CALL     AVANCE

```

```
MOVLW    0xF0           ;gir a l'esquerra 16 pasos
CALL     IZDA
MOVLW    0xEA           ;retrocedir 750mm (21 pasos)
CALL     RETROCEDE
GOTO     ESPERA

END
```