



VNIVERSITAT ID VALÈNCIA

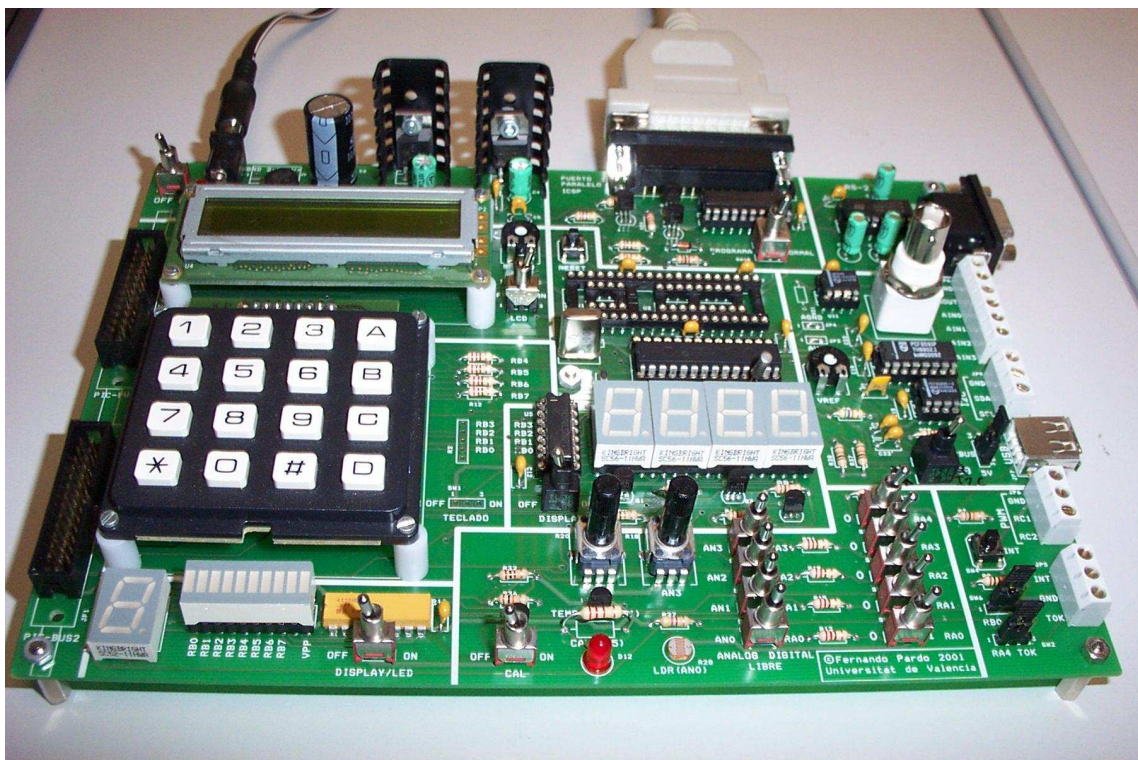
Universitat de València  
Fernando Pardo Carpio  
<http://edumic.uv.es>  
Febrero de 2007

# EduMic

VERSIÓN 2004

Tarjeta de desarrollo para sistemas basados en  
microcontroladores PIC®

## Manual del usuario





# Contenidos

Contenidos .....	iii
1 Introducción .....	1
1.1 Características básicas .....	1
2 Componentes del sistema .....	2
2.1 Fuente de alimentación .....	2
2.2 Módulo de grabación .....	2
2.3 Módulo microcontrolador .....	2
2.4 Módulo de entrada analógico/digital .....	3
2.4.1 Sensor de luminosidad .....	3
2.4.2 Sensor de temperatura .....	3
2.4.3 Calentador .....	3
2.5 Barra de leds y display de 7 segmentos .....	4
2.6 Display múltiple de 4 dígitos .....	4
2.7 Teclado Matricial .....	4
2.8 Pantalla LCD .....	5
2.9 Comunicación serie RS232 .....	5
2.10 Periféricos I2C .....	5
2.10.1 Conversor Analógico/Digital y Digital/Analógico I2C .....	6
2.10.2 Memoria EEPROM I2C .....	6
2.11 Interfaz USB .....	6
2.12 Conexiones externas .....	7
3 Montaje de la tarjeta EduMic .....	8
3.1 Material .....	8
3.1.1 Circuitos integrados y reloj .....	8
3.1.2 Diodos, transistores y pantallas .....	8
3.1.3 Resistencias y potenciómetros .....	8
3.1.4 Condensadores .....	9
3.1.5 Conectores, puentes y jumpers .....	9
3.1.6 Interruptores, botones y teclas .....	9
3.1.7 Otro material adicional .....	9
3.2 Pasos para la realización del montaje .....	10
3.3 Ajustes finales .....	12
4 El programa EduMic .....	14
4.1 Instalación .....	14
4.2 La ventana del programa .....	15
4.3 Funciones del programa .....	15
4.3.1 Menú Archivo .....	15
4.3.2 Menú Dispositivo .....	15
4.3.3 Menú Acción .....	15
4.4 El fichero de configuración .....	17
4.5 Consideraciones particulares .....	18
4.5.1 Problemas con la palabra de configuración .....	18
4.5.2 Programación a bajo voltaje (LVP) .....	18
4.6 Otros programas de descarga .....	18
5 Programas de Ejemplo .....	19
5.1 Descripción de los programas de ejemplo .....	19
5.1.1 Salidas digitales al display simple y barra de leds (DigOut) .....	19
5.1.2 Entradas digitales (DigInp) .....	19
5.1.3 Entradas analógicas (Analnp) .....	19
5.1.4 Display múltiple 4 dígitos (Display4) .....	19
5.1.5 LCD de 2x16 caracteres (LCD) .....	20
5.1.6 Teclado matricial 4x4 (TecladoLCD) .....	20
5.1.7 Comunicación RS232 (RS232) .....	20
5.1.8 Bus I2C: conversor digital/analógico, PCF8591 (I2Canaout) .....	20



## 1 Introducción

La tarjeta EduMic (antes conocida con el nombre de EduPIC) y el programa que la acompaña forman la herramienta ideal para el desarrollo de aplicaciones basadas en los microcontroladores PIC®. El objetivo de la tarjeta es el de dar cabida al mayor número de componentes periféricos en la misma placa, dando la posibilidad de elegir los tipos de periféricos que se desean. Naturalmente la tarjeta cuenta con las conexiones y buses necesarios para conectar otros periféricos de forma sencilla y rápida, sin que unos componentes interfieran con otros.

La tarjeta se complementa con un programa de descarga de uso sencillo y compatible con casi todos los sistemas operativos existentes en la actualidad. Su sencilla interfaz gráfica lo hace muy manejable. Además dispone de funciones adicionales que no se encuentran en otros programas de grabación de micros PIC, como es la posibilidad de reiniciar el microcontrolador para ser reprogramado, o modificar la palabra de configuración según el dispositivo de que se trate.

La tarjeta está preparada para albergar la mayoría de micros PIC con encapsulado DIP. Admite microcontroladores con encapsulados de 8, 18, 28 (ancho y estrecho) y 40 pines sin necesidad de utilizar expansores adicionales. La tarjeta está preparada para trabajar con la mayoría de los dispositivos de las familias 12, 16 y 18.

La programación del microcontrolador PIC se realiza en la propia placa y no hay necesidad de quitar el chip, ni desconectar periféricos, ni apagar el sistema. Basta con mover un simple conmutador y descargar el programa desde un PC. La programación en la propia placa está disponible para los dispositivos de la familia 16 y 18 que admitan ICSP (casi todos). Es posible programar la familia 12, pero de momento no se dispone del software adecuado (consultar en la página web de la tarjeta <http://edumic.uv.es> las actualizaciones del software).

### 1.1 Características básicas

La tarjeta cuenta con numerosos periféricos de entrada y salida, de comunicaciones, analógicos/digitales, etc. A continuación se muestran las características resumidas:

- Admite las familias 12, 16 y 18 (encapsulados DIP de 8, 18, 28 y 40 pines)
- Programación en la misma tarjeta de los microcontroladores PIC de las familias 16 y

18 que soporten ICSP (casi todos). La programación se hace a través del puerto paralelo. El software soporta W95, W98, WNT, W2000, WMe y Linux.

- Conmutador de programación para no interferir con el resto del circuito.
- LCD alfanumérico estándar de 2x16 caracteres con selección de las líneas de control.
- Teclado matricial de 4x4 teclas.
- Fuentes analógicas integradas:
  - 2 potenciómetros (0-5 V)
  - 1 sensor de temperatura
  - 1 sensor de luminosidad
- Resistencia para calentar el sensor de temperatura.
- Barra de LEDs
- Display único 7 segmentos
- Conjunto de 4 displays de 7 segmentos con driver 74LS48
- 5 entradas digitales ajustables a 0 y 1 por interruptor
- Selectores de entradas analógicas, digitales o desconexión.
- Periféricos I2C:
  - PCF8591: conversor analógico/digital y digital/analógico, con 5 entradas y una salida analógicas. Se ha incluido un buffer a la salida analógica así como un conector BNC.
  - PCF8582: Memoria E2PROM, 256 bytes.
- Interfaz serie RS232.
- Interfaz USB (sólo para micros que soportan USB).
- Botón para generar una INT externa.
- Interruptores para habilitar o no cada uno de los dispositivos del sistema.
- Conectores individuales para el bus I2C, entradas y salidas analógicas del I2C, PWM (RC1 y RC2), interrupción externa (INT) y entrada reloj externa timer0 (T0K)
- Dos conectores IDC de 26 pines cada uno para la conexión de tarjetas de expansión. El primer conector contiene todas las señales de los microcontroladores PIC de hasta 28 pines y es compatible con otras tarjetas. El segundo conector contiene el resto de pines presentes en los microcontroladores PIC más grandes.
- Fuente de alimentación 220 V. Posibilidad de alimentación con dos baterías de 9 voltios, o cualquier tensión entre 16 y 20 voltios en continua o 13 y 18 en alterna.

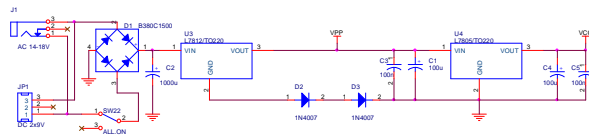
*PIC y MPLAB son marcas registradas de Microchip Technology Inc. en Estados Unidos, Unión Europea y otros países.*

## 2 Componentes del sistema

### 2.1 Fuente de alimentación

La tarjeta cuenta con un módulo de alimentación que se encarga de generar las tensiones continuas estabilizadas que necesitan los microcontroladores PIC para ser reprogramados y el resto de los circuitos para que funcionen. De esta manera sólo se necesita una tensión a la entrada, continua o alterna.

Si se utiliza una fuente externa de tensión continua, por ejemplo unas pilas, la tensión debe ser mayor que 16 voltios y menor que 25. En el caso de alimentación continua esto se consigue fácilmente mediante dos pilas de 9 voltios sin importar la polaridad. Si la fuente externa es alterna, entonces la tensión de alimentación puede ser algo menor, ya que la tarjeta puede funcionar a partir de 12 voltios en alterna y hasta 20. Si se conectan periféricos con un alto consumo de corriente se aconseja aumentar la tensión de entrada hasta al menos 16 voltios en alterna.



### 2.2 Módulo de grabación

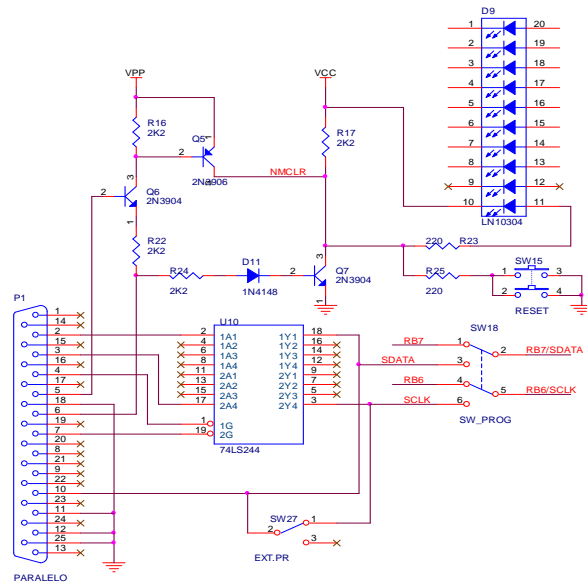
Este módulo se encarga de generar las señales eléctricas adecuadas para la programación en la misma placa de los microcontroladores PIC de las familias 16 y 18 que admitan programación ICSP (*In Circuit Serial Programming*) y que son la mayoría. Este módulo también soporta la familia 12, pero el software aun no ha sido actualizado.

Con la utilización de micros PIC basados en EEPROM o FLASH es posible grabar una y otra vez el micro sin necesidad de extraerlo de la tarjeta. El software EduMic permite la descarga de los programas, la palabra de configuración, identificador y la memoria de datos EEPROM.

La grabación hace uso de los puertos RB6 y RB7 para introducir y sincronizar los datos hacia el microcontrolador PIC. Es importante no interferir con estas líneas durante la grabación para que ésta se realice correctamente. Para evitar cualquier interferencia de los periféricos presentes en la placa, o cualquier otro que añada libremente el usuario, se dispone de un conmutador que aísla estas dos líneas de programación con un simple clic. El usuario puede por tanto incorporar sus propios circuitos periféricos sin preocuparse de las líneas de programación.

La programación se realiza a través del puerto paralelo del PC. Junto con la placa se suministra el software y el cable para esta tarea.

En la nueva versión de la tarjeta EduMic se dispone de un conector externo por si se quieren generar las señales de programación externamente, o aprovechar las de la propia tarjeta.



### 2.3 Módulo microcontrolador

El módulo microcontrolador contiene todos los zócalos necesarios para albergar los micros de las familias 12, 16 y 18 con encapsulados de 8, 18, 28 y 40 pines. Se ha estudiado la disposición de todos estos zócalos (un total de 5) para ocupar el mínimo espacio posible en el circuito impreso.

El módulo microcontrolador incluye también un cristal de cuarzo de 6 MHz, así como un pulsador de Reset para inicializar el microcontrolador de forma manual.

Dependiendo del tamaño del microcontrolador PIC se utilizan más o menos líneas para la conexión de los periféricos:

- **Micros de 8 pines:** GP0 y GP1 están conectados a RB7 y RB6 para poderlos programar. GP4 y GP5 son RB0 y RB1 que están conectadas, entre otros, a dos leds luminosos como salida.
- **Micros de 18 pines:** RA0-RA4 están conectados a las 4 entradas analógicas o 5 digitales, así como a las líneas de control del LCD. RB0-RB7 están conectados a varios dispositivos de salida como el bus de datos del LCD, barra de leds, display 7 segmentos, display 7 segmentos múltiple, y al teclado matricial como dispositivo de entrada. RB0 y RA4 se pueden también configurar para ser



usados como interrupción externa (pulsador) y reloj externo del temporizador 1.

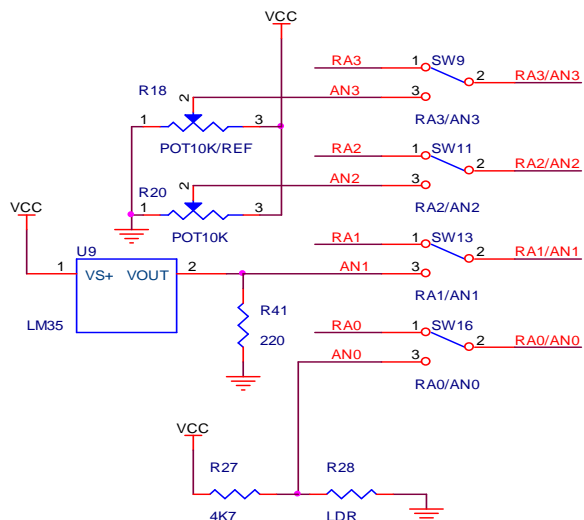
- **Micros de 28 y 40 pines:** Además de las conexiones de los micros de 18 pines, se utilizan las señales RC3, RC4 y RC5 para el interfaz I2C y USB, y las señales RC6 y RC7 para la comunicación serie RS232. Además se han realizado conexiones específicas para las señales RC1 y RC2 para ser usadas como PWM.

Todas las líneas de los micros PIC de 8, 18 y 28 pines son accesibles externamente mediante un conector IDC de 26 pines compatible con otros módulos de otros fabricantes (Microsystems Engineering). Para los de 40 pines se cuenta con otro conector adicional en el que se incorporan además las señales de reloj, reset, etc., que no estaban presentes en el primer bus.

## 2.4 Módulo de entrada analógico/digital

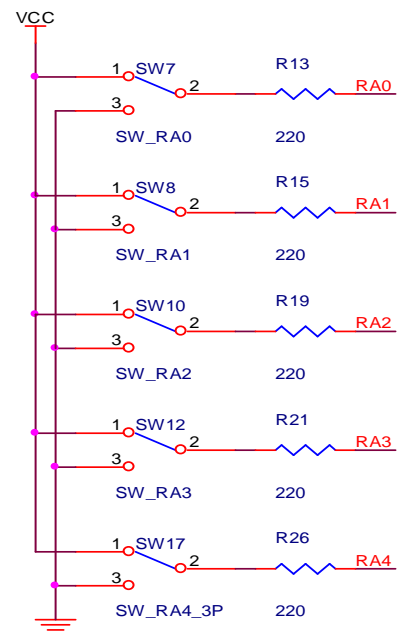
Los puertos del microcontrolador del RA0/AN0 al RA3/AN3 se pueden utilizar indistintamente como entradas analógicas o como puertos digitales normales de entrada/salida.

En la placa EduMic se dispone de 4 entradas analógicas formadas por un sensor de luz LDR (RA0/AN0), un sensor de temperatura LM35 (RA1/AN1) y dos potenciómetros ajustables entre 0 y 5 V (RA2/AN2 y RA2/AN3).



En el mismo módulo digital se cuenta con 5 entradas digitales (RA0-RA4) formadas por conmutadores que conectan estas líneas a 0 o 5 V a través de unas resistencias de protección de 220 Ohmios. Para seleccionar entre las entradas digitales y las analógicas de forma individual, se cuenta con cuatro conmutadores de tres posiciones. A la izquierda se seleccionan las entradas analógicas, a la derecha las digitales, y en la posición central se desconectan estas

entradas del circuito, de manera que no interfieran con otros periféricos que se podrían incorporar sobre estas líneas (como por ejemplo el LCD).



Los conmutadores utilizados para poner los unos y ceros y para conmutar entre analógico/digital son miniaturizados para ocupar poco espacio y tener un diseño compacto. Sin embargo, la alta calidad de estos conmutadores los hace también caros, por lo que existe la opción de utilizar jumpers normales y corrientes en lugar de los conmutadores. El problema es que entonces todas las etiquetas explicativas están al revés, ya que los contactos en los conmutadores están invertidos.

### 2.4.1 Sensor de luminosidad

Este sensor está formado por una resistencia sensible a la luz (LDR) en serie con una resistencia de 4700 Ohmios. El LDR junto a la resistencia 4K7 forman un divisor de tensión que se conecta directamente a la entrada AN0/RA0 del microcontrolador PIC.

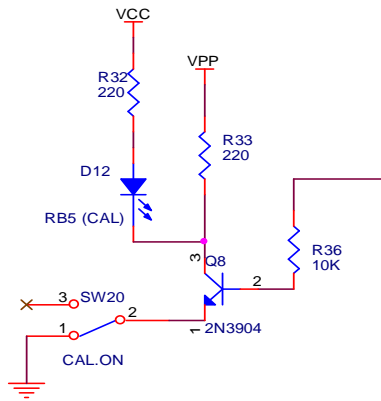
### 2.4.2 Sensor de temperatura

Como sensor de temperatura se utiliza el circuito integrado LM35 que tiene la característica de que cada 10 mV de la tensión de salida equivale a 1 grado centígrado, siendo 0 V equivalente a 0° C. La salida del LM35 está conectada directamente a la entrada AN1/RA1 del microcontrolador PIC.

### 2.4.3 Calentador

Se trata de una resistencia que se enciende poniendo a uno el puerto RB5. Está colocada justo al lado del sensor de temperatura y sirve para cambiar artificialmente la temperatura del

sensor. El interruptor SW20 (CAL) sirve para deshabilitar esta función.

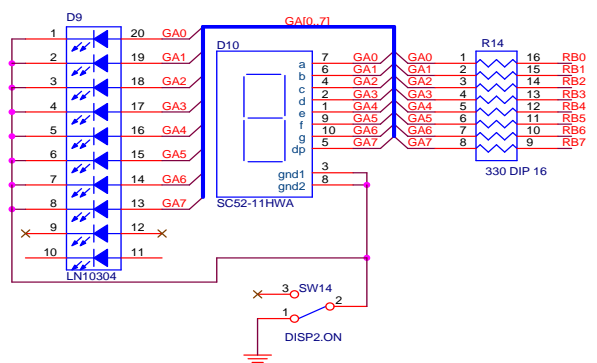


## 2.5 Barra de leds y display de 7 segmentos

La salida digital básica está formada por una barra de 8 leds y un display de 7 segmentos típico. El puerto RB0 está conectado al led de más a la izquierda de la barra y el RB7 es el antepenúltimo empezando por la izquierda; el penúltimo no está conectado y el último indica que se está programando el microcontrolador PIC (Vpp a 13 voltios).

El display de 7 segmentos está conectado en paralelo con la barra de leds. RB0 está conectado al segmento a, RB1 al b, RB2 al c, RB3 al d, RB4 al e, RB5 al f, RB6 al g, y RB7 al dp (punto decimal).

Como la barra y el display comparten el mismo conjunto de resistencias, la barra toma la mayor parte de la corriente por lo que el display se ve algo menos que la barra de leds.



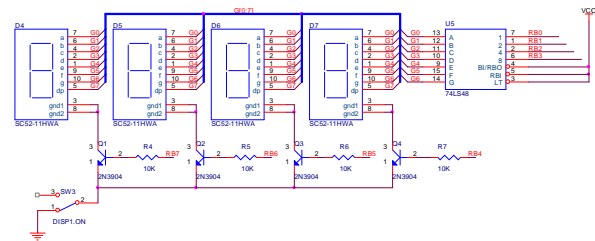
## 2.6 Display múltiple de 4 dígitos

Este display permite la visualización de números de hasta 4 cifras. Se trata de 4 displays de 7 segmentos conectados a un convertidor de BCD a 7 segmentos (74LS48). El convertidor BCD toma su entrada de los 4 bits bajos del puerto B (RB3-RB0). El resto de bits de B se utilizan para

seleccionar cada uno de los displays de 7 segmentos (RB7 selecciona la cifra más significativa y RB4 la menos significativa).

La visualización de un número de 4 cifras se realiza mediante un barrido continuo: primero se coloca en la parte baja de B la primera cifra del número en BCD y en la parte alta '0001' para activar el display menos significativo. A continuación se pone en la parte baja la siguiente cifra y en la alta '0010', y así sucesivamente hasta completar las cuatro cifras, volviendo otra vez a la cifra menos significativa. Es conveniente esperar unos pocos milisegundos entre cambios de cifras para que el display se pueda encender.

Dado que cada display sólo está encendido la cuarta parte del tiempo (en el caso de usar los 4 displays), la luminosidad que se obtiene no es muy alta. Una forma de aumentar la luz en los displays es colocando varios 74LS48 en paralelo.

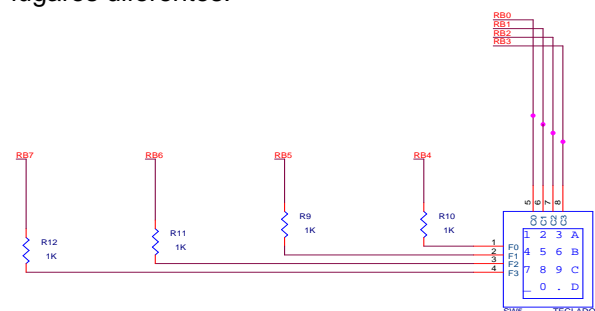


## 2.7 Teclado Matricial

Uno de los elementos de entrada más interesante y útil es el teclado. En este caso se ha previsto a la tarjeta de un teclado matricial de 16 teclas (4x4). El teclado matricial es interesante porque permite tener un mayor número de teclas con un menor número de cables.

Un teclado matricial 4x4 tiene 4 columnas y 4 filas de hilos, de manera que cuando se pulsa una tecla se pone en contacto una columna con una fila; sabiendo qué fila y qué columna se han unido es posible saber cuál de las 16 teclas ha sido pulsada.

Las columnas del teclado que habitualmente se distribuye con la placa, se encuentran conectadas a la parte alta del puerto B y las filas a las bajas. Si se utiliza un teclado propio, las filas y las columnas pueden encontrarse en lugares diferentes.





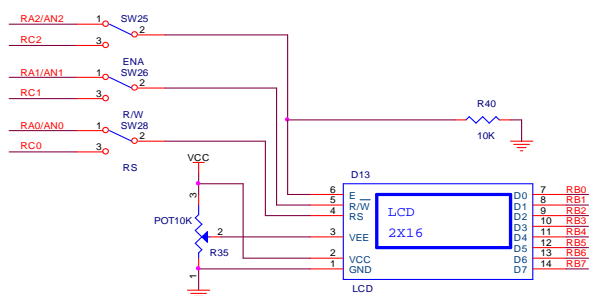
La forma habitual de leer una tecla es en dos fases. En la primera fase se programan los 4 bits menos significativos de B como salida y los más significativos como entrada. Antes de sacar nada por el puerto B hay que habilitar los *pull-ups* internos. A continuación se saca un cero por B (las líneas RB3-RB0 se pondrán a nivel bajo) y seguido se lee B para ver las entradas más significativas (RB7-RB4). Si no se ha pulsado nada, en la parte alta habrá sólo unos (gracias a los *pull-ups* internos), mientras que si se ha pulsado algo, el bit correspondiente a la columna que se hubiera pulsado se pondrá a cero (gracias a que se conecta ese bit con alguno de los de salida que están a nivel bajo).

## 2.8 Pantalla LCD

Los displays de 7 segmentos, incluso los de varias cifras, son bastante limitados cuando se quiere sacar mucha información, implementar menús, etc. Lo ideal es tener una pantalla alfanumérica de un tamaño suficiente para poder hacer frente a la mayoría de aplicaciones empotradas que se quieran probar.

La placa EduMic tiene integrada una pantalla LCD alfanumérica de 2 líneas y 16 caracteres por línea. Se trata de una pantalla estándar que utiliza un patillaje estandarizado y un protocolo también estándar, esto quiere decir que el control de este LCD es común a todos los LCD de 2x16, pudiéndose utilizar los mismos programas incluso en otros LCD que tengan diferente número de líneas o caracteres.

El LCD se conecta al microcontrolador a través de 3 líneas de control y 4 u 8 de datos (configurable). En el caso de la tarjeta EduMic se han utilizado las señales RA0 o RC0, RA1 o RC1 y RA2 o RC1 como control y RB7-RB0 como datos. Se ha dado la posibilidad de asignarle a cada señal de control una línea de RA o de RC por si se quiere utilizar alguno de estos pines para otra cosa. Por defecto se utilizan los de RA y todos los ejemplos que se incluyen utilizan estas líneas. RA0/RC0 corresponde a la señal de control RS (Selección de Registro), RA1/RC1 está conectada a R/W (Lectura/Escritura) y RA2/RC2 es E (Habilitación).



Si no se usa el LCD hay que deshabilitarlo. Para ello hay que quitar el jumper SW25 (ENA).

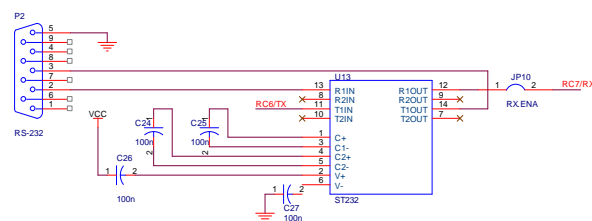
El funcionamiento del LCD es muy simple y se aconseja mirar los diferentes ejemplos y documentación que se adjunta para comprender mejor su funcionamiento.

## 2.9 Comunicación serie RS232

Una de las partes más importantes de los sistemas basados en microcontrolador es la comunicación con el mundo exterior. Tanto es así que la mayoría de los microcontroladores PIC modernos incluyen internamente una UART para la comunicación serie. Esta UART implementa de forma sencilla el protocolo de comunicación serie que utiliza la conexión RS232 por ejemplo.

Pero para poder conectar el microcontrolador PIC mediante una conexión RS232 no sólo basta la UART (que resuelve el problema lógico), sino que es también necesario adaptar las señales eléctricas al estándar RS232. Esto lo realiza el chip ST232 junto con algún condensador adicional.

Se han elegido los puertos RC6 (TX) y RC7 (RX) para conectarlos directamente al MAX232 y realizar la comunicación serie mediante dos hilos. Esta elección se ha hecho así porque los microcontroladores PIC que tienen integrada la UART utilizan estas dos señales como transmisión y recepción.



Si el microcontrolador no posee una UART integrada todavía es posible utilizar la conexión serie implementando el protocolo de comunicación por software. Hay numerosos ejemplos para hacer esto e incluso algunos compiladores lo implementan de forma automática sin más que especificar las líneas utilizadas como transmisión y recepción.

El jumper del interfaz serie permite aislar el puerto RC7 del resto del circuito.

## 2.10 Periféricos I2C

El protocolo de comunicaciones I2C permite la comunicación sencilla entre chips que implementen dicho protocolo mediante dos únicas líneas. Muchos de los dispositivos modernos (incluidos los microcontroladores PIC) poseen este protocolo para poderse comunicar entre sí o con otros dispositivos periféricos.

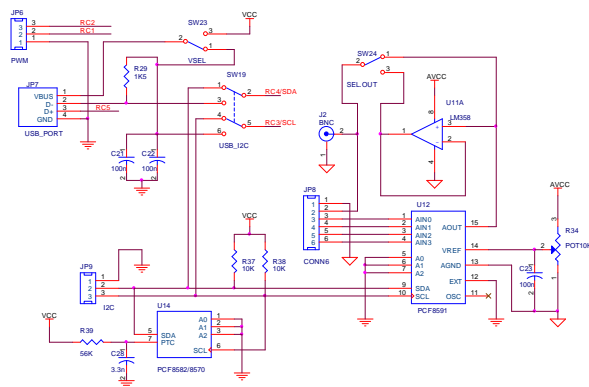
Al igual que ocurría con la comunicación serie anterior, muchos microcontroladores PIC

incluyen internamente la implementación del protocolo del I2C, pero para aquellos microcontroladores PIC que no la incluyan es posible implementarla por software.

Los microcontroladores PIC que integran el protocolo I2C utilizan las líneas RC3 (SCL) y RC4 (SDA) para la comunicación serie I2C. SCL hace de señal de sincronía y SDA de datos. Los micros que no tengan el protocolo I2C deberán utilizar RC3 y RC4 para poderse comunicar con los periféricos incluidos en la placa.

Hay determinados microcontroladores PIC que utilizan precisamente las señales RC3 y RC4 para implementar la comunicación por el puerto USB. Para poder utilizar I2C o USB (nunca ambos al mismo tiempo) se han dispuesto dos jumpers (SW19) para elegir una cosa o la otra. Si los jumpers se dejan al aire entonces RC3 y RC4 quedan libres para cualquier otra aplicación.

Las dos señales del bus I2C están disponibles mediante un conector extra (además del PIC-BUS donde están todas las señales) para poder poner dispositivos I2C externos. Para poder hacer pruebas con el bus I2C se han incluido en la propia tarjeta dos dispositivos típicos como son un convertor A/D y D/A (que complementa al A/D interno del micro), y una memoria EEPROM.



### 2.10.1 Convertor Analógico/Digital y Digital/Analógico I2C

Además de ejemplo perfecto de periférico I2C, el chip PCF8591 posee lo que le venía faltando a la parte analógica de placa de desarrollo y es un convertor de Digital a Analógico. El chip no sólo cuenta con este convertor Digital/Analógico sino que también incorpora un convertor Analógico/Digital con 4 entradas analógicas. El chip tiene por tanto 4 entradas analógicas y una salida analógica.

Las 4 entradas analógicas se encuentran disponibles en un conector para poder conectar señales analógicas externas.

A la salida analógica del chip se ha colocado un amplificador operacional que hace de buffer de

corriente. El jumper SW24 permite elegir si para la salida analógica de la tarjeta (BNC) se toma la del convertor D/A, o bien se toma la del amplificador, cuya misión es la de suministrar corriente y aislar el convertor de posibles sobrecargas.

Las especificaciones de estos chips, así como el de todos los demás, se encuentran disponibles con los ficheros de la distribución y en la página web de la tarjeta EduMic. También se incluyen ejemplos para su uso.

### 2.10.2 Memoria EEPROM I2C

La memoria EEPROM de datos de los microcontroladores PIC puede resultar insuficiente para algunas aplicaciones, de hecho, varios microcontroladores PIC ni siquiera tienen memoria de datos interna. El chip PCF8582 tiene 256 bytes de memoria EEPROM que puede ser escrita y leída por el micro a través del bus I2C.

Las lecturas y escrituras se pueden hacer de forma individual, especificando la dirección y luego el dato, o bien, se pueden hacer en modo ráfaga leyendo o escribiendo bloques enteros. En el mismo zócalo se pueden colocar otros tipos de memorias EEPROM de más capacidad y también basadas en el bus I2C.

## 2.11 Interfaz USB

Algunos microcontroladores PIC incorporan la posibilidad de comunicación a través del puerto USB. Esta comunicación precisa de unas especificaciones eléctricas muy precisas y concretas por lo que la implementación software del protocolo no es suficiente para la conexión del micro a un puerto USB. La interconexión que se ha realizado al conector USB es de tipo A.

La señal de alimentación del puerto USB está especificada a 5 V, sin embargo algunos dispositivos utilizan 3.3 V, por lo que se ha puesto un jumper para poder seleccionar esta tensión entre 5 y 3.3 V. Además este jumper sirve sobre todo para desconectar la tensión que viene por el puerto USB, que de hecho es lo habitual en dispositivos esclavo. Sólo se aconseja la conexión de esta señal a 5 V en el caso de requerir alimentación por parte del maestro (cosa rara en esta placa por tener alimentación propia) o en el caso de que se actúe como maestro, si bien no todos los microcontroladores PIC actuales lo permiten. En los demás casos (casi siempre) es mejor dejar este jumper desconectado.

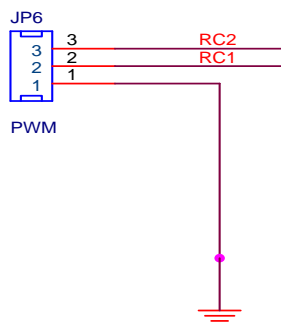
Tal y como se comentó en la sección dedicada al I2C, ambos protocolos comparten las mismas líneas, por lo que no se pueden utilizar los dos al mismo tiempo. En caso de querer utilizar el

puerto USB hay que configurar el par de jumpers SW19 en modo USB.

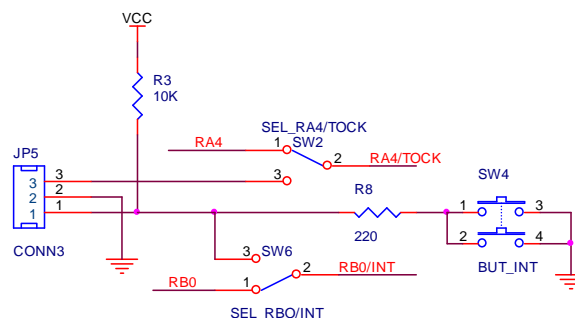
## 2.12 Conexiones externas

A continuación se enumeran las conexiones externas disponibles en la placa, empezando por la esquina superior izquierda y siguiendo según las agujas del reloj:

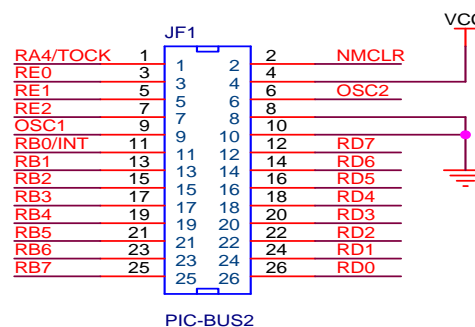
- **Jack de alimentación (J1):** Se trata de un jack simple para acoplar en la mayoría de transformadores del mercado.
- **Pines de alimentación (JP1):** Se han dispuesto dos pines para poder soldar una entrada distinta de alimentación, por ejemplo, dos pilas de nueve voltios.
- **Conector de programación (P1):** Se trata de un conector estándar DB25 macho para cable paralelo. Con un cable paralelo conectado al PC se puede programar el microcontrolador PIC presente en la tarjeta.
- **RS232 (P2):** Se trata de un conector estándar DB9 hembra para la transmisión/recepción serie de datos utilizando comunicación RS232.
- **Conector del conversor analógico (JP8):** Se trata de 6 conexiones de apriete donde se encuentran disponibles las 4 entradas y una salida analógicas del conversor PCF8591 del bus I2C, además de la tierra analógica utilizada en el circuito.
- **Conector I2C (JP9):** Tres conexiones de apriete donde se encuentran las señales del bus I2C (SDA y SCL) además de GND digital.
- **Conector USB (JP7):** Se trata de un conector estándar tipo A para conexión directa mediante un cable USB. (Si se conecta a un ordenador el cable debe ser del tipo A-A.)
- **Conector PWM (JP6):** Tres conexiones de apriete donde están disponibles las salidas RC1, RC2 y GND digital. Esta conexión es útil en aplicaciones donde se precise modulación por anchura de pulsos.



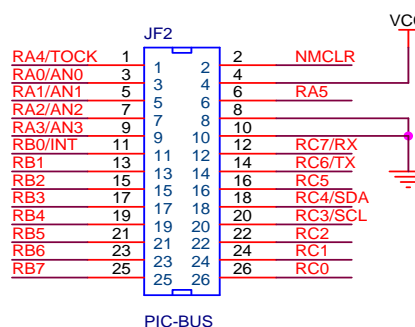
- **Conector T0K e INT (JP5):** Tres conexiones de apriete que incluyen la entrada externa del reloj del timer0 (RA4/T0K) y la interrupción externa (RB0/INT), además del GND digital. En ambos casos se dispone de un jumper para seleccionar RA4/T0K (SW2) y otro para seleccionar RB0/INT (SW6), en este último caso se dispone de un botón para generar a mano la interrupción externa.



- **PIC-BUS2 (JF1):** Contiene los puertos de los micros de 40 pines y algunas señales (como el reloj y reset) que no estaban en el PIC-BUS. Las señales de este puerto son: RA4/T0K, RE0, RE1, RE2, OSC1, OSC2, RB7-RB0, RD7-RD0, NMCLR, VCC y GND.



- **PIC-BUS (JF2):** Contiene los puertos y señales de los micros de 8, 18 y 28 pines. Es compatible con el PIC-BUS de la tarjeta Trainer de Microsystems Engineering, por lo que se pueden acoplar todos los módulos extra de este fabricante. Las señales presentes en este puerto son: RA5-RA0, RB7-RB0, RC7-RC0, NMCLR, VCC y GND.



## 3 Montaje de la tarjeta EduMic

Una de las posibilidades de distribución de la tarjeta es en kit, esto significa que se entregan los componentes junto con el circuito impreso por separado y el usuario debe soldarlos correctamente en la placa. En este apartado se explican los componentes electrónicos de los que está formada la tarjeta EduMic y se dan los pasos a seguir para ensamblar correctamente todos los componentes.

El circuito impreso es una placa de alta calidad fabricada en fibra (FR4 1.6 18/18) con taladros metalizados, doble cara, máscara de protección de soldadura, serigrafía y ha sido testada electrónicamente. El tamaño es de 229x163 mm, que resulta relativamente pequeño dada la cantidad de componentes y funcionalidad de la tarjeta.

Un simple soldador tipo lápiz de baja potencia y estaño son las únicas herramientas que se necesitan para soldar todos los componentes. El estaño puede tener un grosor de hasta 1 mm, pero para soldar los transistores y similares, y sólo para ellos, se aconseja utilizar un cable de estaño algo más fino.

### 3.1 Material

En las explicaciones que siguen se especificará cada componente mediante su referencia que se encuentra visible en la serigrafía de la tarjeta. Para facilitar la asociación de referencias con componentes a continuación se dan unas tablas con toda la lista de materiales:

#### 3.1.1 Circuitos integrados y reloj

Referencia	Descripción
U1	Zócalo micro PIC 8 pines.
U2	Zócalo micro PIC 18 pines.
U3	L7812. Estabilizador de tensión de 12 voltios.
U4	L7805. Estabilizador de tensión de 5 voltios.
U5	74LS48. Decodificador BCD a 7 segmentos.
U6	Zócalo micro PIC 28 pines estrecho.
U7	Zócalo micro PIC 28 pines ancho.
U8	Zócalo micro PIC 40 pines.
U9	LM35. Sensor de temperatura.
U10	74LS244. Buffer para el programador por ISCP.
U11	LM358. Amplificador Operacional doble para la salida del convertor digital/analógico.

U12	PCF8591. Periférico I2C: convertor analógico/digital y digital/analógico. (4 entradas y una salida analógicas.)
U13	ST232CN. Interfaz para comunicación RS232.
U14	PCF8582/8570. Periférico I2C: memoria EEPROM de 256 bytes.
Y1	Cristal de cuarzo de 6 MHz.

#### 3.1.2 Diodos, transistores y pantallas

Referencia	Descripción
D1	B380C1500. Puente rectificador.
D2, D3	1N4007. Diodos para tener una caída de tensión de 1.3 voltios.
D4-D7	SC52-11HWA. Displays de 7 segmentos para el display múltiple.
D8	LED para indicar que la placa tiene tensión (POWER).
D9	LN10304. Barra con 10 leds.
D10	SC52-11HWA. Display simple de 7 segmentos.
D11	1N4148. Diodo bloqueador de corriente inversa para el programador ICSP.
D12	LED. Conectado a la resistencia de calentamiento para indicar que se está calentando.
D13	LCD. Pantalla LCD
Q1-Q4	2N3904. Transistores de selección del display múltiple.
Q5	2N3906. Transistor en el programador ICSP.
Q6-Q7	2N3904. Transistores en el programador ICSP.
Q8	2N3904. Selección de la resistencia de calentamiento.
	Total 2N3904: 7

#### 3.1.3 Resistencias y potenciómetros

Referencia	Descripción
R3-R7, R36-R38, R40	Resistencias de 10K (marrón-negro-naranja) (9 resistencias)
R1, R8-R13, R15, R19, R21, R23, R25, R26, R32, R33, R41	Resistencias de 220 (rojo-rojo-marrón) (16 resistencias)
R14	4116R. Pack de 8 resistencias de 330, DIP16.
R16, R17, R22, R24	Resistencias de 2K2 (rojo-rojo-rojo) (4 resistencias)
R18, R20	Potenciómetros de 10K con varilla. Entradas analógicas AN3 y AN2.
R27	Resistencia 4K7 (amarillo-violeta-rojo).
R28	LDR. Resistencia sensible a la luz.

R29	Resistencia 1K5. (marrón-verde-rojo). Interfaz USB.
R34	Potenciómetro 10K. Tensión de referencia del conversor digital/análogo
R35	Potenciómetro 10K. Regulador del contraste del LCD.
R39	Resistencia 56K (verde-azul-naranja). Para PCF8582 (Opcional)

### 3.1.4 Condensadores

Referencia	Descripción
C1, C4	Electrolítico 100uF. Alimentación.
C2	Electrolítico 1000uF. Alimentación.
C3, C5	Multicapa 100nF. Alimentación.
C6-C18	Multicapa 100nF. Desacoplo. (13 condensadores)
C21, C22	Multicapa 100nF. Conexión USB.
C23	Multicapa 100nF. Estabilizador tensión de referencia PCF8591.
C24-C27	Multicapa 100nF. Interfaz RS232.
	Total de 100nF: 22
C19, C20	Cerámicos 27pF. Estabilizador Cristal 6 MHz.
C28	Multicapa 3.3nF. Estabilizador PCF8582. (Opcional)

### 3.1.5 Conectores, puentes y jumpers

Referencia	Descripción
J1	Jack de alimentación.
J2	BNC, salida analógica.
JF1	IDC 26 pines. PIC-BUS2
JF2	IDC 26 pines. PIC-BUS
JP1	Apriete 3 pines. Entrada opcional alimentación.
JP3	Unión 5 voltios digital y analógico.
JP4	Unión tierra digital y analógica.
JP5	Apriete 3 pines. Interrupción externa (INT), reloj externo timer0 (T0K) y tierra digital (GND).
JP6	Apriete 3 pines. PWM. GND, RC1 y RC2
JP7	USB tipo A.
JP8	Apriete 6 pines. Entradas y salida analógicas del PCF8591 y tierra analógica (AGND).
JP9	Apriete 3 pines. I2C. GND, SDA y SCL.
JP10	Conector externo programación
P1	DB9 hembra. RS232.
P2	DB25 macho. Cable paralelo programador.
SW2	Jumper. Selecciona entre RA4 y T0K
SW6	Jumper. Selecciona entre RB0 e INT.
SW19	2 Jumpers para seleccionar USB, I2C o desconexión de ambos.

### 3.1.6 Interruptores, botones y teclas

Referencia	Descripción
SW3	2pos3pin. Habilita display múltiple de 7 segmentos.
SW4	Botón. Genera interrupción externa (INT)
SW5	Teclado matricial 4x4 teclas.
SW7, SW8, SW10, SW12, SW17	2pos3pin. Pone a cero o a uno las entradas digitales RA0-RA4.
SW9, SW11, SW13, SW16	3pos3pin. Selecciona entre entradas digitales, entradas analógicas o desconexión (libre) de las señales RA0/AN0-RA3/AN3.
SW14	2pos3pin. Habilita display 7 segmentos y barra de leds.
SW15	Botón de RESET.
SW18	2pos2x3pin. Conmutador de programación ICSP.
SW20	2pos3pin. Habilita la resistencia de calentamiento.
SW22	2pos3pin Interruptor general
SW23	Selección tensión USB
SW24	Selección salida analógica (con o sin buffer)
SW25	RA2 o RC2 para LCD (E)
SW26	RA1 o RC1 para LCD (R/W)
SW27	Programación externa
SW28	RA0 o RC0 para LCD (RS)
	Total 2pos3pin: 9
	Total 3pos3pin: 4
	Total 2pos2x3pin: 1

### 3.1.7 Otro material adicional

Cantidad	Descripción
1	TCA-20 Fuente de alimentación (opcional).
1	Soporte metacrilato con 4 pies, 4 tornillos m3, 1 m2 y 1 separador plástico (opcional).
1	Cable paralelo puerto impresora 25 hilos.
1	Placa de circuito impreso (PCB)
2	Radiadores TO220
2	Tornillos y tuercas m3 (radiadores)
4	Separadores hexagonales de metal y tuercas m3
8	Separador plástico cilíndrico 12 mm.
8	Tornillos y tuercas m2
10	Puentes jumper con lengüeta
2	Regleta macho 40 pines (jumpers y LCD)
1	Regleta hembra 40 pines (LCD y teclado)
2	Tiras 32 pines torneados (Zócalo micro PIC dip 28 ancho, display 7 segmentos)

2	Zócalo 8 pines torneado (micro PIC y PCF8582)
3	Zócalo 16 pines torneado (74LS48, PCF8591, ST232)
1	Zócalo 18 pines torneado (micro PIC dip18)
2	Zócalo 20 pines torneado (barra de leds, 74LS244)
1	Zócalo 28 pines torneado estrecho (micro PIC dip28)
2	Zócalo 40 pines torneado (4 displays múltiples, micro PIC dip40)

### 3.2 Pasos para la realización del montaje

Antes de empezar a soldar componentes conviene establecer una estrategia en el orden de soldadura dada la gran cantidad de componentes a soldar. Se aconseja la utilización de algún tipo de esponja o almohadilla de gran tamaño para poder darle la vuelta a los componentes sin que estos se caigan.

1. En primer lugar resulta muy conveniente agrupar todos los componentes por referencias y tipos de manera que sean fáciles de localizar.
2. Una vez hecho esto hay que empezar a soldar aquellos puentes que haya en la placa antes de que el resto de componentes nos molesten. Para estos puentes se utilizará hilo fino o el cable de las patas que nos sobre de las resistencias o condensadores.

Los puentes que hay que realizar son el JP3 (alimentación digital/analógica) y el JP4 (masa digital/analógica). Estos últimos se encuentran en la zona del bus I2C. En el caso de querer aislar bien la zona digital de la analógica se pueden sustituir estos dos puentes por hilos enrollados en ferrita.

3. A continuación hay que soldar aquellos componentes que presentan el perfil más bajo como son las resistencias y diodos. Se soldarán todas las resistencias que aparecen en la tabla de resistencias, exceptuando los potenciómetros.

Aunque el sentido en que se coloquen las resistencias no importa, resulta conveniente colocarlas de una manera que su código sea fácilmente legible. Por esta razón es aconsejable colocar todas las resistencias horizontales con el lado plateado a la derecha y el código a la izquierda para que se puedan leer de izquierda a derecha. En el caso de las verticales las pondremos para que se puedan leer de arriba a abajo o de abajo hacia arriba (según gustos) pero todas igual.

La resistencia R33 (CAL-RB5) puede ser de  $\frac{1}{2} W$  o  $\frac{1}{4} W$  y sirve para calentar el sensor de temperatura LM35, por lo que la colocaremos un poco elevada con respecto al PCB, con 4-5 mm es suficiente.

La resistencia R14 es un pack de 8 resistencias en encapsulado DIP. Se trata de un chip de color naranja con nombre 4116R. En principio se podría utilizar un zócalo, pero dada la disponibilidad y bajo coste de este elemento no resulta necesario.

Los diodos a colocar en este momento serán el D1 (puente rectificador de alimentación) el D2, D3 (1N4007, zona alimentación) y el D11 (1N4148, zona programador ICSP). El puente es un chip redondo negro con 4 pines. Es importantísimo respetar la polaridad de este elemento al igual que cualquier otro diodo o semiconductor. D2 y D3 son diodos rectificadores, son de color negro y una banda blanca indica el cátodo, hay que respetar la polaridad tal como viene en la serigrafía del PCB. El D4 es un diodo de señal, es de cristal y de un color naranja; la banda negra nos indica la polaridad correcta.

Una vez colocados los componentes, se sueldan y se cortan las patillas sobrantes.

4. Los siguientes elementos en altura son los zócalos, chips y tiras de pines. Todos los circuitos integrados tienen su zócalo salvo el pack de resistencias (que ya debería estar colocado) y el integrado LM358. Este último chip se puede montar sin zócalo

Los zócalos se colocan en sus huecos tal y como indica la serigrafía. Es importante el respeto de la orientación para no colocar mal ningún chip. A fin de ganar espacio en el PCB se han agrupado los zócalos de los microcontroladores PIC de 8, 18 y 40 pines. En este caso primero se coloca el U1 (8 pines), luego el U2 (18 pines), y por último el U8 (40 pines). Es posible que este último no se pueda poner porque molesta un larguero interior, de ser así hay que quitarlo (normalmente este zócalo ya se entrega sin este larguero por lo que no debería haber problemas). También puede ocurrir que hayamos confundido el zócalo del display múltiple (que tiene todos sus largueros) con éste; hay que comprobar por tanto que hay dos zócalos de 40 pines y que uno tiene un larguero menos.

El zócalo para el U7 (micro de 28 pines ancho) se realiza mediante unas tiras de pines torneados. Hay que cortar 14 por cada lado y soldarlos en su sitio. Hay que tener mucho cuidado de que la tira quede vertical, de lo contrario el microcontrolador PIC no



entrará en su sitio. Para garantizar que quede bien vertical se puede insertar uno de los displays de 7 segmentos.

El display múltiple (D4, D5, D6 y D7) se realiza mediante un simple zócalo de 40 pines como el U8 pero completo, es decir, al U8 no se le tiene que quitar ningún larguero para que quepan el resto de zócalos en su interior.

También el display D10 tiene su zócalo, pero éste se debe construir con tiras de pines al igual que el U7. Hay que tener especial cuidado para que los pines queden verticales, para ello se aconseja colocar el display sobre los pines mientras se suelda.

Sólo cuando la placa esté completamente terminada y montada colocaremos los chips correspondientes en su zócalo, de esta manera se evita que se puedan dañar.

5. Los siguientes elementos serán los potenciómetros de ajuste (sin mando) (R34 y R35), los condensadores no electrolíticos, los botones y los diodos led. Los potenciómetros a colocar son el R35 (LCD) y el R34 (I2C), no hay que confundirlos con el R18 y el R20 puesto que estos últimos tienen un mando para que sean fácilmente ajustables.

Los condensadores no electrolíticos no tienen polaridad por lo que no importa su orientación en la placa. Los de desacoplo (100 nF) (C3, C5-C18, C21-C27) son pequeños, de color naranja y hay muchos iguales. El C28 (3.3 nF opcional) es también naranja pero es cuadrado, sólo hay uno y viene escrito 3n3. El C19 y C20 son los del cristal de cuarzo (27 pF), sólo hay dos, suelen ser de color verde y negro, y son los más pequeños.

Botones sólo hay dos, el de reset (SW15) y el de interrupción externa (SW4). Se insertan y se sueldan.

Hay dos diodos led, el de power (D8) y el del calentador (D12). Como siempre hay que tener cuidado con la polaridad; la patilla larga es el ánodo por lo que se debe introducir en el pin indicado por un signo positivo (+).

6. Una de las tareas más delicadas es colocar y soldar los encapsulados TO92 (transistores Q1-Q8 y sensor de temperatura U9). La dificultad estriba en que estos dispositivos tienen los pines muy juntos y además conviene dejar cierta distancia (1-2 mm) entre el encapsulado y la placa. Por lo tanto, es interesante utilizar un hilo de estaño lo más fino posible. Un truco para dejar una distancia de uno o dos milímetros entre el encapsulado y la placa, consiste en colocar

el transistor pegado a la placa doblando 45 grados las patillas exteriores; una vez dejemos el transistor boca abajo quedará separado exactamente la distancia justa por su propio peso. Por lo tanto, colocaremos todos los transistores de esta manera, le daremos la vuelta a la placa y los soldaremos con el mayor cuidado posible.

7. Los siguientes elementos a colocar van a ser todos los conectores menos el BNC (J1, JF1, JF2, JP1, JP5-JP9, P1, P2), los jumpers (SW2, SW6, SW19, SW23) y las tiras de pines (LCD D13 y teclado SW5). Los conectores no requieren un cuidado especial. Los conectores de apriete van en piezas de tres pines, por lo que para hacer el conector JP8 tendremos que unir dos piezas de tres pines (encajan perfectamente por una ranura presente al efecto). El conector BNC (J2) se ha dejado para más tarde por ser bastante alto.

Los jumpers se hacen cortando las tiras de 40 pines macho que se suministran. La parte más pequeña es la que se introduce en el PCB y la larga es la que queda fuera para colocar los capuchones de selección.

El teclado matricial y el LCD se conectan a la placa a través de unas tiras de pines hembra. Para el LCD tendremos que cortar 14 pines y para el teclado 8 pines. Es conveniente cortar por en medio de un pin para que no se estropeen el resto, de esta manera perdemos un pin pero nos aseguramos que la tira de 14 u 8 quedará completa.

El teclado ya tiene su conector de pines macho, pero los pines son demasiado largos por lo que habrá que recortarlos un poco a fin de que la altura del teclado sea de unos 12 mm.

El LCD no dispone de su conector de pines macho, por lo que habrá que ponérselo. Para ello cortaremos una tira de 14 pines macho y se la soldaremos.

Es importantísimo, especialmente en el LCD y el teclado, que las tiras de pines queden rectas, de lo contrario ni el LCD ni el teclado se podrán montar en el PCB. Una forma de evitar que queden torcidos es soldando sólo uno de los pines de la tira hasta conseguir que quede vertical, una vez conseguida la verticalidad se puede soldar el resto de pines de la tira. Si se sueldan todos los pines y la tira queda torcida, va a ser muy difícil (prácticamente imposible) devolverla a su sitio.

8. Por altura le corresponde el turno a los condensadores electrolíticos de 100 uF. Los condensadores a colocar son por tanto el C1

y C4 (100 uF alimentación). Es importante respetar la polaridad de estos condensadores, de lo contrario se pueden producir averías en la tarjeta.

9. Los conmutadores serán los siguientes elementos a colocar. Hay conmutadores de tres tipos:
  - a. **2pos3pin:** Son conmutadores de palanca de 2 posiciones (derecha-izquierda) y tres pines. Según la posición de la palanca conectan el pin central con uno de los lados. Curiosamente, si la palanca está a la derecha, el pin que se conecta con el central es el izquierdo y viceversa. Por esta razón, si se sustituyen los conmutadores por jumpers, las etiquetas de la serigrafía quedan al revés (a no ser de que unas nuevas etiquetas, Ej. pegatinas, hayan sustituido a las del PCB). SW3, SW7, SW8, SW10, SW12, SW14, SW17, SW20, SW21.
  - b. **3pos3pin:** Son conmutadores de tres posiciones (derecha-centro-izquierda) y tres pines. Son iguales que los anteriores pero tienen una posición más, la central, que desconecta el pin central de ambos pines laterales. De esta manera es posible deshabilitar completamente las entradas analógicas y digitales. SW9, SW11, SW13, SW16.
  - c. **2pos2x3pin:** Es un conmutador doble, equivale a dos de 2pos3pin colocados en paralelo y accionados por una única palanca. En este caso se ha utilizado para deshabilitar las señales RB6 y RB7 durante la programación del microcontrolador PIC. SW18

Aunque el funcionamiento es el mismo, conviene, al menos por estética, colocar los conmutadores lo más verticales posible. Si no se dispone de una almohadilla para hacer presión, realizaremos la misma operación que para las tiras de pines, soldaremos un solo pin moviendo el conmutador hasta que quede vertical, y luego soldaremos el resto de pines.

10. Cristal de 6 MHz Y1. Resulta interesante dejar una separación de 0.5 mm entre el chasis del cristal y la placa, aunque si se pone pegado tampoco pasa nada y es algo más robusto.
11. Los potenciómetros R18 y R20 (entradas analógicas) pueden ser los siguientes elementos a montar. Se pueden distribuir dos tipos de potenciómetros con la placa. Lo

normal es que se suministre un potenciómetro como los de ajuste (R34, R35) pero al que se le ha incluido un mando para poderse ajustar con los dedos. Si este es el caso se monta en su sitio y ya está.

Puede ocurrir, en algún caso, que se suministre un potenciómetro algo mayor cuya característica es que los tres pines se encuentran delante. En este caso se suelda tal cual, pero es necesario realizar una conexión entre el pin central de este nuevo potenciómetro y el pin central que ocuparía el potenciómetro normal de ajuste en la parte de atrás. Este puente se puede hacer con alguna de las patillas que se han cortado de alguno de los componentes discretos soldados con anterioridad.

12. Los penúltimos elementos a montar serán el conector vertical BNC (J2) y el condensador electrolítico de 1000 uF (C2). El conector BNC sólo tiene una posición posible, pero con el condensador hay que tener mucho cuidado de colocarlo con la polaridad correcta.
13. Por último hay que colocar los chips de alimentación U3 (L7812) y U4 (L7805). Antes de soldarlos se les atornillarán los radiadores que se suministran. Hay que atornillarlos en el agujero central del radiador. Una vez hecho esto los colocaremos de manera que el radiador toque directamente la placa del PCB, de esta manera, y una vez soldados, conseguiremos que tanto el radiador como el chip no se muevan. Si una vez soldado existe poca estabilidad en el radiador, podremos afianzarlo atornillándolo de nuevo o moviendo hacia abajo las tiras del radiador en contacto con la placa.

Una vez soldados todos los componentes, incluida la tira en el LCD, montaremos los chips en sus zócalos respectivos, los displays de 7 segmentos en sus zócalos respectivos y montaremos el LCD y el teclado. Para esto último utilizaremos los tornillos y tuercas de 2 mm y los separadores de plástico.

### 3.3 Ajustes finales

Antes de empezar a utilizar la placa resulta conveniente colocar todos los interruptores y jumpers de configuración en un lugar que interfiera lo mínimo posible con el funcionamiento normal de la tarjeta. También los potenciómetros se deben ajustar de manera que su funcionamiento sea correcto.

1. **Jumpers:** El SW2 selecciona entre RA4 y T0K, como normalmente utilizaremos esta señal como entrada digital, colocaremos el capuchón sobre RA4, en cualquier momento

se puede dejar esta señal libre quitando el capuchón o colocándolo en la opción TOK, que al tratarse de una señal externa estará normalmente desconectada.

El SW6 está junto al anterior y selecciona entre RB0 e INT. Normalmente pondremos el capuchón sobre RB0 para que funcionen la mayoría de elementos de la placa. Si se desea dejar esta señal al aire, o que funcione el botón de interrupción o que se pueda conectar una interrupción externa, entonces lo pondremos sobre la opción INT.

El grupo SW19 está formado por dos jumpers y se deben cambiar ambos para seleccionar el bus I2C o USB tal como viene indicado en la serigrafía. Si se desean dejar libres las señales RC3 y RC4 quitaremos el capuchón de los jumpers. Normalmente lo dejaremos en modo I2C.

El SW23 se dejará al aire. En muy pocos casos resulta necesario ponerlo a una tensión u otra. En la sección dedicada al interfaz USB se explica con mayor detalle el uso de este jumper.

EL SW24 se puede conectar a BUF para que la salida analógica se realice a través del buffer.

El SW25, SW26 y SW28 los podemos dejar al aire si no se va a utilizar el LCD. Si se utiliza el LCD entonces lo normal será poner estos jumpers en el puerto A (RA0, RA1 y RA2) pues los programas de ejemplo suponen que el LCD está conectado de esa manera.

El JP10 lo podemos dejar conectado siempre que estemos seguros de que la señal RC7 no interfiere con otros circuitos. Si no vamos a usar el RS232 lo mejor es quitarlo.

- 2. Conmutadores:** Los conmutadores de habilitación de dispositivos los pondremos en posición de apagado (OFF). Estos conmutadores son el SW3, SW14, SW20 y SW21. El conmutador de alimentación (SW22) también lo dejaremos apagado.

El conmutador de programación (SW18) lo pondremos en posición 'normal'.

Los conmutadores de selección analógico/digital (SW9, SW11, SW13 y SW16) los pondremos en su posición central (deshabilitado o 'libre').

Los conmutadores de 0/1 (SW8, SW10, SW12 y SW17) los pondremos a cero o uno pero todos al mismo lugar.

- 3. Potenciómetros:** El R35 sirve para ajustar el contraste del LCD. Si la pantalla aparece

negra girar hacia la derecha y si no aparece absolutamente nada girar hacia la izquierda.

El R34 sirve para ajustar la tensión de referencia del conversor PCF8591. Por defecto el potenciómetro está a cero, por lo que no funcionará el conversor. A fin de tener un ancho margen de tensión de salida, resulta conveniente ajustar esta tensión de referencia lo más alta posible, pero sin llegar a 5 voltios, ya que el conversor se satura mucho antes. Una posición correcta es aquella en la que la flecha del potenciómetro apunta hacia el Noreste, es decir, a 45° entre arriba y la derecha, mirado siempre la placa de frente.

## 4 El programa EduMic

La tarjeta EduMic no sería un buen sistema de desarrollo si no dispusiera de un software adecuado para su programación. Se ha intentado realizar un programa de descarga y programación que fuera sencillo de utilizar pero que al mismo tiempo tuviera muchas posibilidades. También se ha intentado que sea lo más compatible posible con los sistemas operativos existentes actualmente sobre PC (Windows, NT y Linux).

Las características básicas de este software son las siguientes:

- Soporta casi todos los sistemas operativos (Linux, Windows95, Windows98, Windows2000, WindowsNT, WindowsMe).
- El software es de libre distribución.
- Admite la especificación en el fichero HEX tanto de la memoria de programa como de la memoria de datos (EEPROM), así como la palabra de configuración y el identificador. El identificador y EEPROM sólo están disponibles para la familia 16.
- La palabra de configuración se puede establecer en el propio programa seleccionando las opciones que se deseen.
- Admite la especificación de un identificador para el microcontrolador PIC (sólo familia 16).
- El programa sirve tanto para escribir el microcontrolador PIC como para leerlo.
- Se permite la edición de los datos (programa y EEPROM) antes de enviarlo al micro.
- Tiene la característica de poder desproteger chips protegidos contra lectura/escritura en

microcontroladores PIC basados en Flash. Una vez desprotegido el micro pierde toda su información pero puede ser reutilizado. Esta función reinicia completamente el microcontrolador PIC.

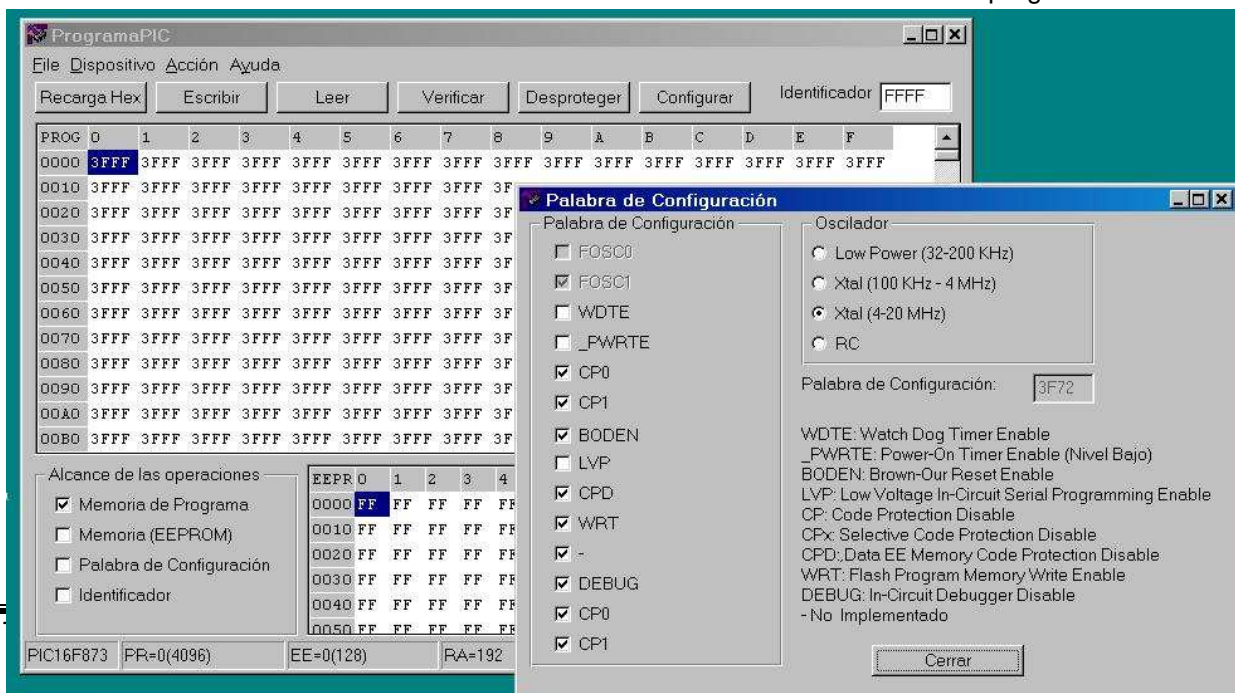
- Dispone de un fichero de configuración para poder incorporar nuevos microcontroladores PIC y funcionalidades según vayan apareciendo sin necesidad de descargar nuevas versiones del software. El propio usuario puede modificar este fichero.

### 4.1 Instalación

La instalación es tan sencilla como bajar el fichero correspondiente a nuestro sistema operativo de la página web donde se encuentra (<http://edumic.uv.es/>), o bien ejecutar directamente el fichero desde el CD que se distribuye con la tarjeta. En cualquier caso hay que desinstalar previamente cualquier versión que ya se tenga de este software.

La instalación difiere un poco según el sistema operativo de que se trate:

- **Windows98 y similares:** La instalación sólo requiere ejecutar el programa de instalación sin más. Una vez instalado ya se puede empezar a trabajar.
- **Windows2000 y NT:** En este caso la instalación difiere de la anterior, ya que es necesario instalar un driver para el puerto paralelo. De la instalación de este driver se encarga el propio programa de instalación, pero para que se pueda instalar correctamente es necesario que sea el administrador quien ejecute el programa de instalación. Una vez instalado se debe reiniciar Windows, ya que de lo contrario el driver no estará disponible. Una vez reiniciado Windows el programa funcionará



perfectamente.

- **Linux:** La instalación hay que hacerla de forma manual. Siendo root hay que copiar los ficheros ejecutables y librería en un directorio que se encuentre en el PATH (por ejemplo /usr/local/bin). El fichero EduMic.ini que viene con la distribución hay que copiarlo en /etc. Por último hay que activar el bit de cambio de usuario (s) para que sea root quien ejecute el programa cuando cualquier usuario lo invoque (esto es necesario para poder acceder al puerto paralelo de forma sencilla).

La desinstalación del programa en Windows se hace como con cualquier otro programa, es decir, en el panel de control elegiremos agregar o quitar programas y lo podremos eliminar.

En linux la desinstalación consiste en borrar los ficheros que se hubieran añadido.

## 4.2 La ventana del programa

En la ventana del programa podemos ver una serie de botones de manera que se puede acceder de forma rápida a cualquier función.

En esta ventana principal podemos distinguir también dos ventanas en forma de tabla. La ventana más grande es la memoria de programa y nos permite ver y cambiar el contenido en hexadecimal de la memoria de programa del microcontrolador PIC. La ventana más pequeña es la memoria de datos y en ella podemos ver y cambiar los valores de la memoria EEPROM de los microcontroladores PIC.

Arriba a la izquierda se encuentra una pequeña entrada desde la que podemos ver y cambiar el identificador de 4 cifras en hexadecimal del microcontrolador PIC.

Abajo a la derecha nos aparece el alcance de las operaciones, es decir, cada vez que se realice una función (escribir, leer, etc.) sólo se realizará sobre la opción u opciones que se hubieran seleccionado y que son: memoria de programa, memoria de datos (EEPROM), palabra de configuración e identificador.

## 4.3 Funciones del programa

### 4.3.1 Menú Archivo

- **Abrir Hex:** Carga en el programa un fichero en formato HEX de Intel. Aunque el programa no está preparado para entender toda la sintaxis del formato HEX, todavía no se ha encontrado ningún programa que no se haya podido cargar.
- **Guardar como:** Esta opción sirve para guardar el contenido de la memoria de

programa, EEPROM, identificador y palabra de configuración en un fichero en formato HEX. Es especialmente útil si estamos leyendo un microcontrolador PIC y queremos guardar su contenido en un fichero. Sólo disponible para la familia 18.

- **Salir:** Abandona el programa y sale.
- **Nombre de fichero:** En la zona inferior suelen aparecer los últimos ficheros que se abrieron para poderlo cargarlos con facilidad. Existe en la ventana de programa un botón que realiza la misma función de cargar el último fichero HEX utilizado.

### 4.3.2 Menú Dispositivo

En este menú encontramos una lista con todos los dispositivos y familias que el programa puede grabar. Las diferencias entre unas familias y otras no son demasiado grandes, se refieren sobre todo a tamaños de memoria de programa y palabra de configuración. Si un dispositivo no aparece en la lista podemos utilizar el que más se parezca y lo más probable es que se programe sin ningún problema.

La lista de dispositivos se toma de un fichero de configuración que el usuario puede editar, de manera que resulta sencillo añadir cualquier microcontrolador PIC o familia que pueda aparecer nueva. También es útil cuando se necesitan los parámetros específicos de un microcontrolador PIC para poder programarlo correctamente. Hay más adelante una sección dedicada a este fichero de configuración donde se explica cómo añadir microcontroladores PIC o modificar los ya existentes.

### 4.3.3 Menú Acción

En este menú se recogen todas las operaciones y funciones que el programa puede realizar sobre el microcontrolador PIC insertado en la tarjeta. Todas estas acciones tienen su botón correspondiente en la ventana de programa, por lo que tenemos dos medios de acceder a ellas. Las principales funciones son las siguientes:

- **Blank Check:** Para dispositivos basados en EPROM (OTP) conviene comprobar que el micro se encuentre borrado. Esto se realiza automáticamente al hacer una escritura, pero también se puede hacer a mano.
- **Verificar:** Verifica que el contenido del microcontrolador PIC coincide con el contenido del programa, EEPROM, configuración e identificador. Realmente esta operación no es necesaria puesto que la propia escritura ya comprueba que se ha realizado correctamente (salvo en la familia 16F7X donde la verificación se realiza tras la escritura).

- **Escribir:** Vuelca el contenido de la memoria de programa, EEPROM, configuración e identificador al microcontrolador PIC. Estos contenidos se han podido leer de un fichero HEX (esto es lo habitual) o se han podido crear modificando las casillas en las ventanas correspondientes. Al igual que todas las operaciones, el alcance dependerá de lo seleccionado.
- **Leer:** Como su propio nombre indica permite leer el contenido del microcontrolador PIC, tanto de la memoria de programa, como de la EEPROM como de la palabra de configuración y el identificador. Una vez leído se puede grabar a un archivo. Dependiendo de la selección del 'Alcance de las operaciones' se leerá una cosa u otra o todas.
- **Desproteger:** Esta función permite reiniciar completamente el microcontrolador PIC, incluso si se encuentra protegido contra lectura. Esta operación sólo funciona con los dispositivos basados en Flash. Esta función desprotege el microcontrolador para volver a ser utilizado, pero no permite descubrir su contenido, es decir, esta operación borra permanentemente todo lo que hubiera en el microcontrolador PIC.
- **Configurar:** Esta opción nos abre una ventana nueva donde nos aparece la palabra de configuración correspondiente al microcontrolador PIC que hemos elegido. Esta palabra de configuración es lo que más diferencia unos micros PIC de otros. En el fichero EduMic.ini podemos especificar la palabra de configuración por defecto para cualquier microcontrolador.

En la ventana de configuración aparecen todos los bits para poderlos modificar. Los dos bits más bajos son los que sirven para especificar el tipo de reloj, y no son accesibles porque se ha creado una cuadro de opciones (arriba a la derecha) para facilitar la elección de los 4 tipos de reloj.

El resto de bits puede cambiar (y mucho) de un micro a otro, por lo que se ha preferido dejarlos libres con un breve comentario a su lado. Un subrayado delante del nombre de cada bit indica que es activo a nivel bajo, de lo contrario es activo a nivel alto. A continuación se enumeran la mayoría de los bits de configuración que nos podemos encontrar:

- **WDTE:** (*Watch Dog Timer Enable*) Habilita el temporizador de perro guardián. Sólo es útil si se quiere detectar que el micro se queda colgado.
- **\_PWRTE** (*Power-On Timer Enable*) Habilita el temporizador de espera al

encender el micro. Si no estamos seguros de lo que tarda en estabilizarse la tensión de alimentación, puede resultar interesante su activación. Es activo a nivel bajo.

- **BODEN** (*Brown-Out Reset Enable*) Se realiza automáticamente un reset cuando la alimentación cae por debajo de un umbral, normalmente entre 4 y 4.5 voltios.
- **LVP** (*Low Voltage In-Circuit Serial Programming Enable*) Permite la programación en bajo voltaje (5 V). Cuando está habilitada, el pin RB3 sólo sirve para programación por lo que no se puede usar en otra cosa, conviene por tanto mantener esta opción desactivada a no ser que realmente se quiera utilizar la programación de bajo voltaje.
- **CP, CPx** (*Code Protect Disable*) Sirve para proteger el código de programa (a 0 protege a 1 desprotege). En muchos micros la protección se puede hacer por zonas; en estos casos aparece CP con un número al lado indicando la zona. Dependiendo de cada micro esta zona es diferente, hay que leerse el manual en estos casos. Lo normal es que aparezcan estos bits de protección en varios bits, en estos casos todos los bits deben ponerse iguales (0 ó 1), de lo contrario el micro podría funcionar mal.
- **CPD** (*Data EE Memory Code Protection Disable*) Deshabilita la protección de código de la memoria EEPROM.
- **WRT** (*Flash Program Memory Write Enable*) Habilita la posibilidad de que se pueda escribir la propia memoria Flash de programa. Esta función sólo está disponible en los dispositivos con memoria Flash de programa.
- **DEBUG** (*In-Circuit Debugger Disable*) Deshabilita la posibilidad de depuración del programa.
- **MPEEN** (*Memory Parity Error Enable*) Habilita la detección de errores de paridad en la memoria.
- Un signo (-) indica que ese bit no está utilizado y que su valor no importa.

La palabra de configuración de la familia 18 es tan compleja que sólo se dispone de la posibilidad de introducirla directamente, sin llegar al nivel de bit.

Tal y como se ha comentado anteriormente, existe un botón para cada una de estas



operaciones accesible desde la propia ventana del programa.

Conviene recalcar una vez más que las operaciones anteriores sólo se van a realizar sobre los elementos que se hayan seleccionado en la zona de 'Alcance de las operaciones', esto incluye el borrado y verificación del microcontrolador PIC y todo el resto de operaciones, salvo la de desprotección que naturalmente desprotege todo.

#### **4.4 El fichero de configuración**

A fin de poder actualizar el programa con la aparición de nuevos microcontroladores PIC, pero sin tener que reinstalarlo todo, existe un fichero de configuración con los parámetros que definen el programa y los microcontroladores PIC. Este fichero se llama EduMic.ini y es fácilmente accesible desde Windows gracias a un acceso directo en el menú Inicio>Programa>EduMic. En estos sistemas el fichero se encuentra en el mismo lugar que el ejecutable. En los sistemas basados en Linux el fichero se encuentra en /etc y sólo root puede cambiarlo, a no ser que haya habilitado permisos de escritura para todos los usuarios.

En las siguientes líneas se encuentran los parámetros del programa así como la definición de un microcontrolador PIC de ejemplo:

Los números en hexadecimal se especifican

```
[Config]
port=$378
lastfile1=
...
[PIC16F873]
ProgMem = 4096
EEProm = 128
RAM = 192
PalConf = CP1 CP0 DEBUG - WRT CPD LVP BODEN CP1 CP0 _PWRTD WDTE FOSC1 FOSC0
PalConfDef = $3F72
Programming = Flash
```

con un \$ delante. Las secciones en el fichero ini comienzan con su nombre entre corchetes.

La primera sección, [Config], contiene los parámetros del programa que incluyen el puerto paralelo, los últimos ficheros visitados, el último directorio y el último micro utilizado.

Normalmente el puerto de la impresora suele ser el \$378, pero puede ocurrir en algunos sistemas que este puerto sea el \$278 o el \$3BC. Si el que viene por defecto no funciona probaremos a modificar el fichero de configuración cambiando el puerto por cualquiera de los valores anteriores.

El resto de secciones en el fichero corresponde cada una con un micro concreto, o también con

un tipo más general de microcontroladores PIC. El nombre de sección (entre corchetes) es el nombre que luego aparece en el menú 'Dispositivo', siguiendo el mismo orden que el fichero de configuración.

En el ejemplo que se está analizando sólo se ha incluido la sección dedicada al PIC16F873. El primer parámetro es el ProgMem que se refiere al número de bytes que tiene la memoria de programa. EEPROM contiene el número de bytes de la memoria EEPROM de datos. RAM contiene los bytes de la RAM, y es informativo pues no tiene ninguna influencia en el programa EduMic. PalConf contiene las 13 etiquetas de la palabra de configuración; tal como se pongan aquí aparecerán luego en la ventana de configuración de la palabra. PalConfDef contiene el valor recomendado para la palabra de configuración. Para la familia 18 esta palabra recomendada consta de 7 valores separados por un espacio. Se han elegido valores apropiados para la placa EduMic. Por último, Programming contiene el modo de programación, en concreto hay 5 tipos: Flash, Flash1, Flash2, OTP, 18f2xx0 y 18fx2xx8, estos dos últimos son para la familia 18 mientras que el resto son para la 16.

Los parámetros ProgMem y EEPROM son utilizados para saber los límites de las ventanas de programa y datos. También se utilizan para poner límites en el momento de leer un microcontrolador PIC o al escribirlo.

En el fichero de configuración que se suministra aparecen tipos generales de microcontroladores PIC, si bien se han incluido algunas entradas de los micros específicos más utilizados o que se suministran en la placa, como los PIC16F876A, PIC16F876, PIC16F84 o el nuevo PIC18F2550.

Del resto de microcontroladores PIC sólo se han incluido las familias. El criterio utilizado para elegir los parámetros ProgMem y EEPROM ha sido simple: de cada familia se ha puesto el mayor valor posible. Aunque esto podría causar problemas al escribir (la escritura se hace cíclica al llegar al límite real del micro) en la práctica no tiene por qué ocurrir nada, ya que sólo se escriben los bytes que vienen especificados en el fichero HEX. En el único caso en que abrirá problemas, es en el caso que se lea un micro y

luego se escriba, ya que al leer el límite de la memoria será ProgMem y no lo que realmente ocupe el código pues no se sabe. Este caso es bastante improbable, ya que lo habitual es leer el código a partir de un fichero HEX.

La palabra de configuración suele ser la misma dentro de una misma familia, y sólo en alguna familia rara puede que haya alguna diferencia, especialmente en la protección selectiva del código, ya que al no tener la misma cantidad de memoria las zonas protegibles pueden cambiar.

---

## 4.5 Consideraciones particulares

---

Si bien el programa y la tarjeta están preparados para programar la mayoría de los microcontroladores PIC de Microchip, existen algunas familias, especialmente las que van apareciendo nuevas, que pueden plantear problemas a la hora de su programación.

Probar todos los micros es imposible. Por otro lado la tarjeta EduMic no es un *programador* en el sentido técnico de la palabra pues un verdadero programador tiene más funciones que esta tarjeta que apenas se limita a descargar un programada en el micro. La función de la tarjeta EduMic es sobre todo la de hacer experimentos con micros regrabables, si bien puede funcionar y programar casi cualquier tipo de micro PIC.

Algunos tipos de microcontroladores PIC, incluso basados en Flash, pueden ser difíciles de grabar si no se toman las medidas necesarias. Algunos micros que permiten la grabación a baja tensión (la familia 16F62X por ejemplo) son difíciles de grabar puesto que si todos los pines de RB no están a cero entran en modo grabación a baja tensión en vez de a alta tensión que es justamente el que utiliza la tarjeta EduMic.

Para estos dispositivos hay que asegurar que estos pines estén a nivel bajo (en concreto, para la familia 16F62X basta con que RB4 esté a nivel bajo). En general podemos conseguir esto 'encendiendo' los dispositivos periféricos como display/led o el display de 4 cifras (poniendo los interruptores a ON). Si de esta manera hubiera aún problemas al escribir, habría que poner a GND el pin de RB4 para asegurar un cero en dicho pin.

Para no tener este problema deberemos deshabilitar la función LVP poniendo a cero el bit correspondiente de la palabra de configuración del microcontrolador PIC, de esta manera ya se podrá programar siempre sin ningún problema.

Aparte de este problema, de fácil solución, nos podemos encontrar también con micros difíciles de grabar, es decir, que graban algunas posiciones pero no todas, dando errores en

algunas. A veces el microcontrolador PIC se ha escrito bien pero falla al leer y en otras sí que ocurre que se ha escrito mal cierta posición. En estos casos hay que repetir el proceso de escritura hasta que la verificación sea correcta.

Por último decir que la tarjeta EduMic es un grabador de prototipos o de pruebas, pero nunca se le puede considerar como un grabador de producción, puesto que sólo verifica a la tensión nominal y no a las mínimas y máximas.

### 4.5.1 Problemas con la palabra de configuración

En algunos microcontroladores PIC se han encontrado problemas al grabar la palabra de configuración. En realidad la grabación se realiza bien pero falla al verificar. En estos casos simplemente se vuelve a escribir y ya está; la mayoría de las veces esto resuelve el error de verificación.

En otras ocasiones es posible que se grabe mal la palabra de configuración bloqueando el micro contra escritura. En estos casos la única solución es desproteger el micro (es una posibilidad del programa EduMic) y volver a intentar la grabación de la palabra de configuración, a ser posible sin que se programe nada más del micro.

### 4.5.2 Programación a bajo voltaje (LVP)

Esta característica está presente en casi todos los micros PIC basados en Flash y permite grabar el microcontrolador con la tensión de 5 V. Sin embargo, utilizan un pin para este menester por lo que este pin queda inutilizado. Curiosamente los microcontroladores PIC vienen con la programación a bajo voltaje por defecto, por lo que si queremos utilizar todos los pines deberemos configurar el micro para que no utilice el LVP. Esto se hace desde la palabra de configuración y el valor a cargar está precisamente incluido en el recomendado para el dispositivo que se trate. Por lo tanto **es importante que antes de utilizar un micro PIC grabemos la palabra de configuración recomendada.**

---

## 4.6 Otros programas de descarga

---

El interfaz hardware de programación de la tarjeta EduMic permite la utilización de otros programas de descarga más actualizados aunque algo engorrosos de utilizar al no estar específicamente diseñados para los micros PIC.

Este es el caso del software IC-Prog que soporta un gran número de dispositivos y también es de libre distribución. Este programa se puede encontrar en <http://www.ic-prog.com/>

## 5 Programas de Ejemplo

Antes de realizar ningún experimento propio, el usuario debería programar el microcontrolador PIC con los programas de ejemplo que vienen en la distribución para ver si funcionan correctamente. Esta prueba y el ajuste de la placa se han realizado ya antes de ser enviada, pero conviene hacerla una vez más por si la placa se hubiera dañado durante el transporte, además son una buena introducción para el usuario de la tarjeta EduMic.

### 5.1 Descripción de los programas de ejemplo

Estos programas se han escrito en C y han sido compilados con el C2C. Se distribuyen el fuente en C, y los resultados en ASM y HEX. Para no perder tiempo se recomienda cargar directamente el fichero HEX.

Los ejemplos que se presentan utilizan el módulo objeto del programa y los módulos únicamente necesarios para su prueba. De esta manera quedan claras cuáles son las instrucciones involucradas en cada módulo y le permiten al usuario de la tarjeta adentrarse poco a poco en la programación de los microcontroladores PIC. Los ejemplos del final incluyen en un solo programa la mayoría de módulos, por lo que resultan más apropiados en caso de sólo querer comprobar el funcionamiento de la tarjeta y sus periféricos.

En todos los ejemplos supondremos la configuración de partida por defecto, es decir, todos los dispositivos deshabilitados, salvo el interruptor de alimentación, el conmutador de programación en su posición 'NORMAL', los conmutadores de las entradas analógicas/digitales en su posición central 'LIBRE', los jumpers del USB/I2C en su posición de 'I2C', el jumper de RB0/INT en su posición 'RB0' y el jumper RA4/T0K en su posición 'RA4'.

#### 5.1.1 Salidas digitales al display simple y barra de leds (DigOut)

La salida más simple se realiza a la barra de leds en común con el display simple de 7 segmentos. En los ficheros DigOut.c, DigOut.asm y DigOut.hex se encuentran las fuentes en diferentes formatos.

Lo que hace este programa es encender uno a uno los leds de la barra de leds. Luego los enciende uno a uno pero sin apagar el anterior, con lo que al final se quedan todos encendidos. Luego se sacan los números del 0 al F por el

display de 7 segmentos. Cuando acaba vuelve a empezar.

**Configuración:** debe encenderse el interruptor display/led (SW14).

#### 5.1.2 Entradas digitales (DigInp)

Los ficheros DigInp.\* muestran el funcionamiento de las entradas digitales presentes en las señales RA0-RA4. Lo que hace el programa es encender el led correspondiente de la barra de leds según se active o no la entrada digital correspondiente. Es decir, si RA0 lo ponemos a 1 con el interruptor SW7, entonces se enciende el led de RB0 (izquierda en la barra de leds).

**Configuración:** debe encenderse el interruptor display/led (SW14) y los interruptores de selección analógico/digital deben ponerse a digital (derecha).

#### 5.1.3 Entradas analógicas (Analnp)

Los ficheros Analnp.c, Analnp.asm y Analnp.hex son el ejemplo más simple de prueba de las entradas analógicas del micro. En este ejemplo se convierte la señal analógica del puerto AN2 (potenciómetro) y se escala de 0 a 15 para mostrarla por el display simple de 7 segmentos (puerto B).

Más adelante se presentan variantes más complejas que de un golpe prueban varios de los circuitos de la tarjeta, pero se ha incluido éste para dejar claras las instrucciones propias de la conversión A/D.

**Configuración:** debe encenderse el interruptor display/led (SW14) y los interruptores de selección analógico/digital deben ponerse a analógico (izquierda).

#### 5.1.4 Display múltiple 4 dígitos (Display4)

Este programa saca un número de 4 cifras por el display múltiple de 7 segmentos. Como enciende un dígito cada vez y el 74LS48 no suministra la suficiente corriente, la intensidad del número que aparece no es muy alta.

El número que aparece en el display es la conversión de analógico a digital de la entrada AN2 (potenciómetro de la izquierda). Lo que se debe ver es un número que va de 0000 a 1023, ya que la conversión es de 10 bits.

**Configuración:** debe encenderse el interruptor display (SW3) y los interruptores de selección analógico/digital deben ponerse a analógico (izquierda).

### 5.1.5 LCD de 2x16 caracteres (LCD)

El programa LCD.c necesita de un include (lcd.h) donde se han reunido las principales funciones de acceso al LCD. Lo que hace este programa es sacar por la pantalla del LCD el mensaje 'Hola Mundo!!!'. Es muy sencillo pero muestra el funcionamiento a solas del LCD; más adelante se explotan más ampliamente todas las posibilidades del LCD.

**Configuración:** basta con habilitar el LCD poniendo el interruptor LCD (SW21) a ON y los jumpers SW25, SW26 y SW28 en su posición izquierda para seleccionar RA0, RA1 y RA2. Es importante que los conmutadores de entradas analógicas/digitales estén en su posición deshabilitada, o sea, en el centro (esta debería ser la opción por defecto y por tanto no era necesario volverlo a indicarlo aquí).

### 5.1.6 Teclado matricial 4x4 (TecladoLCD)

Este programa muestra el funcionamiento del teclado matricial. Utiliza una rutina que devuelve el código de la tecla pulsada. Esta rutina se puede usar tal cual en cualquier otro programa que utilice el teclado.

Lo que hace el programa es sacar por el LCD la tecla que se hubiera pulsado, es decir, muestra el código de la tecla (0..F), por lo que no tiene por qué coincidir con la serigrafía que aparece en cada botón.

**Configuración:** exactamente la misma que en el ejemplo anterior (el teclado no necesita de una configuración especial, es compatible con cualquier otro periférico).

### 5.1.7 Comunicación RS232 (RS232)

Este programa muestra el funcionamiento del puerto de comunicaciones RS232. Necesita de un microcontrolador que tenga una UART interna (como por ejemplo el 16F873 y tantos otros).

Lo que hace el programa es enviar por el puerto serie el código de la tecla que se esté pulsando. Además este código aparece también por el LCD. Para poderlo probar hay que conectar la tarjeta con el PC a través de un cable serie cruzado. Luego hay que abrir un terminal en el PC utilizando el programa que se encuentra en *Inicio>Programas>Accesorios>Comunicaciones>hyperterminal* Una vez abierto tendremos que especificar el puerto (COM1 o COM2) la velocidad (9600) los bits de datos (8), especificar que no hay paridad, los bits de parada (1) y que el control de flujo es por hardware. Una vez configurado el terminal y en modo de recepción se podrán ver por la pantalla del PC las teclas que se vayan pulsando.

**Configuración:** la misma que en el ejemplo anterior y además hay que conectar el puente JP10. Si se necesita deshabilitar la comunicación serie porque se está utilizando algún periférico externo y hay conflicto, entonces hay que desconectar este jumper JP10.

Bus I2C: memoria EEPROM, PCF8582 (I2Ceeeprom)

Este programa muestra el funcionamiento del bus I2C, en este caso se comunica con el chip PCF8582 (EEPROM) para leer y escribir bytes.

Lo que hace el programa es leer el teclado mandando cada tecla pulsada a la EEPROM; en el momento se detecta la tecla 0xF, se lee todo lo que se hubiera enviado a la EEPROM. Mientras se escriben las teclas a la EEPROM, éstas aparecen en la primera línea del LCD. Cuando se leen las teclas de la EEPROM, éstas se muestran en la segunda línea del LCD. Si todo ha ido bien ambas líneas deben coincidir.

**Configuración:** La misma que en el ejemplo anterior (LCD habilitado) y además hay que verificar que los dos jumpers de SW19 están en la posición I2C (opción por defecto).

### 5.1.8 Bus I2C: conversor digital/análogo, PCF8591 (I2Canaout)

Este programa muestra el funcionamiento del bus I2C, en este caso se utiliza el conversor digital/análogo del chip PCF8591 para obtener un diente de sierra por la salida analógica (BNC).

El diente de sierra se ha implementado mediante un contador de byte que al rebosar vuelve a empezar. Para ver su funcionamiento habrá que conectar un osciloscopio en el BNC.

Este experimento es útil para calibrar la tensión de referencia (R34 VREF) del conversor; mediante este potenciómetro se puede regular la amplitud. Si se quiere tener la máxima amplitud moveremos el cursor del potenciómetro de manera que aumente la amplitud, pero a partir de 4 voltios (4.5) es posible que la señal ya no sea un diente de sierra porque se recorta por arriba, en ese momento habremos llegado a la máxima amplitud y no deberemos aumentar más VREF.

**Configuración:** no necesita una configuración especial, ya que sólo se utiliza el bus I2C y éste no interfiere con ningún dispositivo interno a la placa, salvo el USB, por lo que los jumpers SW19 deben estar en posición I2C (opción por defecto).