

# PLACA DEL MICROROBOT EXPERIMENTAL

6	<b>¡Error! Marcador no definido.</b>	
6	1	
6.1.	DIAGRAMA DE BLOQUES DE LAS PLACAS DE CONTROL DEL MICROROBOT EXPERIMENTAL.....	2
6.2.	LA FUENTE DE ALIMENTACIÓN .....	3
6.3.	DESCRIPCIÓN DEL CIRCUITO DE PROGRAMACIÓN DEL PIC16F877 .....	4
6.4.	COMO UTILIZAR EL PROGRAMA BOOTLOADER.....	6
6.5.	COMO REALIZAR LA CARGA DEL PROGRAMA EN EL MICROCONTROLADOR.....	8
6.6.	LA PLACA DE EXPNSIÓN .....	10

## 2.1. DIAGRAMA DE BLOQUES DE LAS PLACAS DE CONTROL DEL MICROROBOT EXPERIMENTAL

El diagrama de bloques de las placas del microrobot experimental, es el que se muestra en la Figura 1.

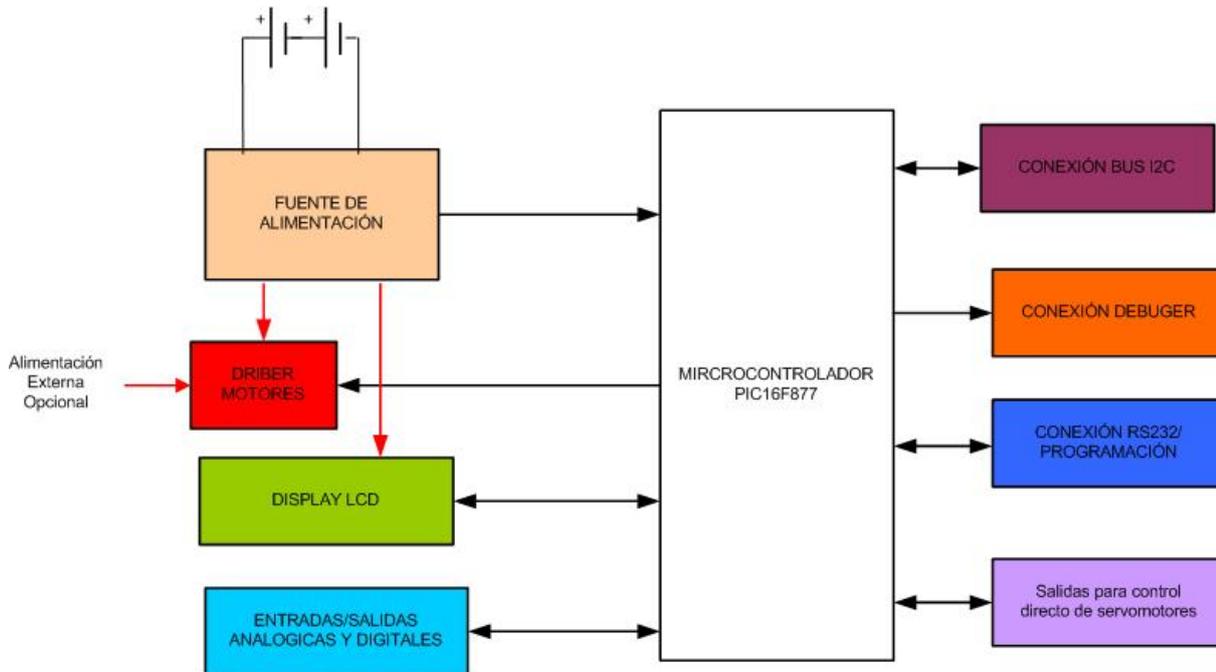


FIGURA 1.- DIAGRAMA DE BLOQUES DEL ROBOT EXPERIMENTAL

Las placas de circuito impreso que denominaremos Sumi y Expansión, contienen los bloques que se indican en la Figura 1, de la siguiente manera:

La placa “Sumi” contiene los siguientes elementos:

- Microcontrolador PIC16F877A (Anexo Microcontrolador)
- Fuente de alimentación
- Conexión RS 232
- Conexión I2C
- Conexión para el Debugger ICD2
- Entradas y salidas Digitales
- Conectores para ampliación de E/S

La placa de “Expansión” contiene:

- Un driver para motores
- 6 entradas Digitales con Trigger Schmitt
- LCD de 2 líneas 16 columnas.

Seguidamente describiremos los elementos que no se hayan descrito en capítulos anteriores.

## 2.2. LA FUENTE DE ALIMENTACIÓN

El sistema del microrobot experimental que hemos diseñado puede y debe trabajar con una tensión de  $5V \pm 10\%$ , por lo tanto, parece lógico que nuestra alimentación tenga un regulador de 5V como es el LM7805 y efectivamente así es como puede verse en el circuito de la figura 2, el cual nos permite alimentar el circuito perfectamente con una batería de de por ejemplo 9 o 12 V. No obstante en este curso utilizaremos como alimentación 4 baterías de NiMH capaces de entregar hasta 2.500 m/h, que conectaremos en serie y como cada célula es capaz de entregar tan solo 1,2V, la tensión máxima de alimentación sería de 4,8V, por lo tanto en esta aplicación para que funcione correctamente, nos hemos permitido poner el *jumper* JP1 para poder cuentear el regulador y que no actúe, solamente eliminaremos el puente si se piensa en otro tipo de alimentación que proporcione por lo menos 7,5V a la entrada.

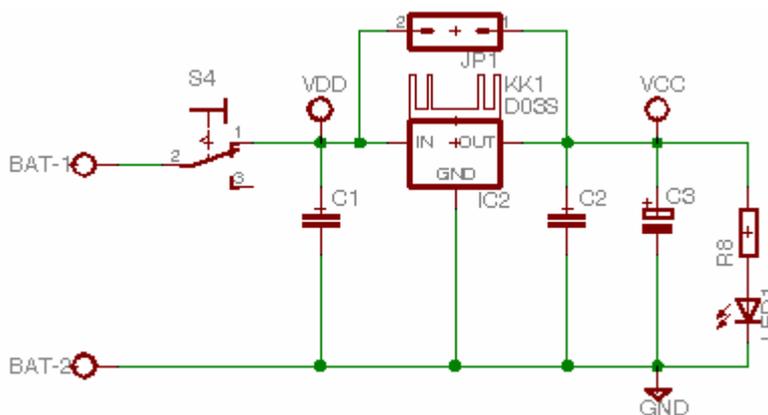


FIGURA 2.- FUENTE DE ALIMENTACIÓN

Por otra parte la tensión de alimentación de los motores de corriente continua, que se controlan desde la placa de expansión, pueden alimentarse directamente a las baterías o a la salida del 7805 de la placa Sumi, o alimentarse a otra tensión o a través de otras baterías que se conecten al conector PWR y poniendo el *jumper* JP1R de la placa de expansión en la posición PWR.

## 2.3. DESCRIPCIÓN DEL CIRCUITO DE PROGRAMACIÓN DEL PIC16F877

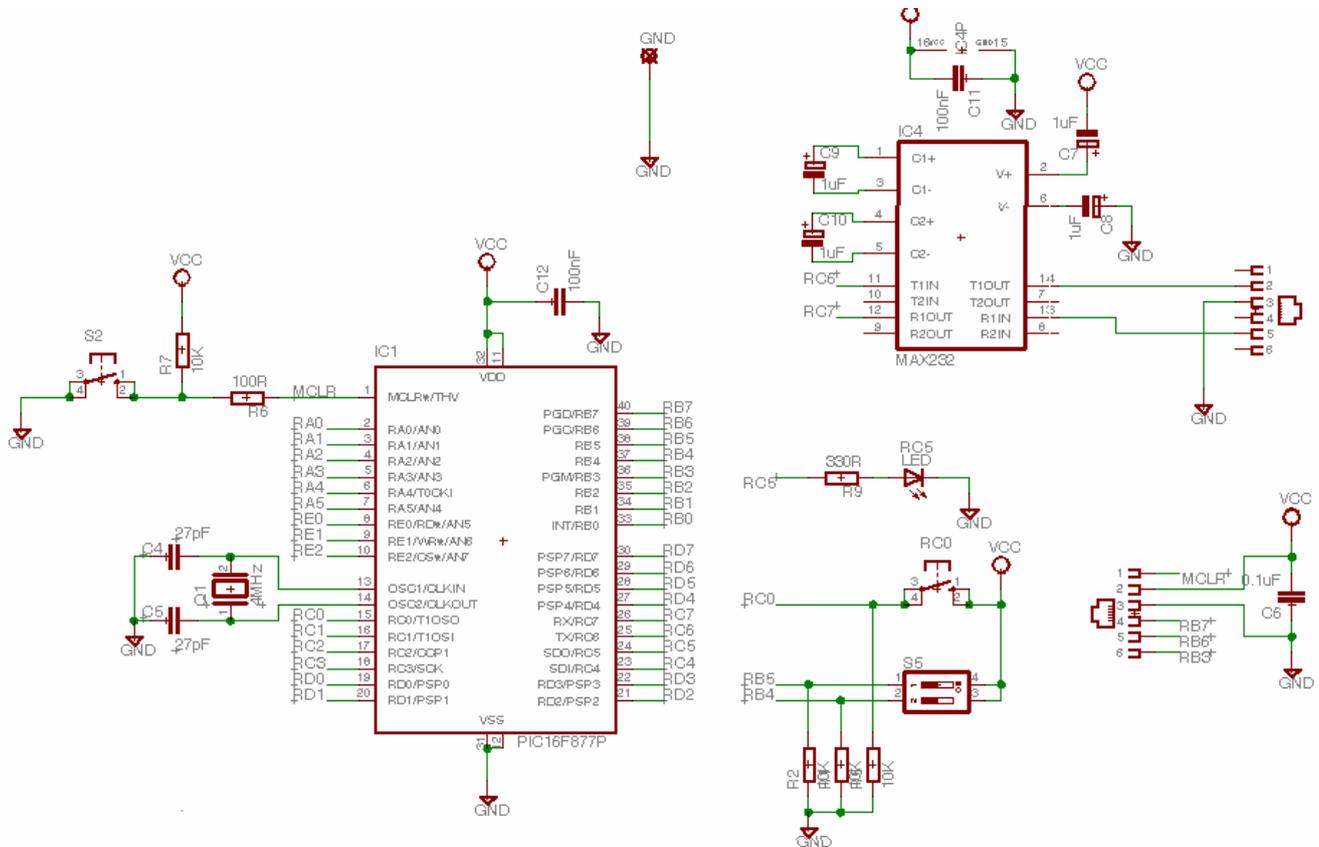


FIGURA 3.- CIRCUITO DE GRABACIÓN DEL PIC16F877

El circuito, como puede verse en la Figura 3, está formado por un microcontrolador PIC16F877A al que previamente le hemos cargado el programa bootloader, utilizando para ello el ICD2 conectado al conector RJ11 marcado con Debugger un ICD2 o utilizando antes de insertar el chip en el zócalo de la placa de circuito impreso, un programador universal para dispositivos programables o por ejemplo alguno de los programadores para los microcontroladores PIC que se pueden encontrar en el website [www.ic-prog.com](http://www.ic-prog.com).

El circuito consta de un microcontrolador PIC16F877A al que se le han conectado un cristal de cuarzo de 4MHz y los condensadores para el oscilador de 27 pF y un pulsador que cuando se activa pone un nivel bajo en la entrada MCLR para provocar un RESET del sistema. Además para poder conectar la USART del microcontrolador al ordenador, se ha utilizado un circuito adaptador de niveles MAX232, tal y como indica el fabricante en sus notas de aplicación, y por último se le ha dotado de un conector RJ-11 hembra de seis terminales, para poder conectar el circuito al puerto serie del ordenador, utilizando las líneas RC6 y RC7 del microcontrolador.

La conexión entre la placa del microrobot y el ordenador, es del tipo modem nulo, por lo que solo necesita tres líneas para realizarla, pero el cruce entre la transmisión y la recepción ya se ha hecho en la placa de circuito impreso, por lo que se han realizado las siguientes conexiones: La línea de masa conectada al terminal 5 (GND) del conector DB9 al 4 (GND) del RJ-11, las líneas 2 (RXD) del DB9 con el 5 (TXD) del RJ-11 y por último

la 3 (TXD) del conector DB9 con la 2(RxD) del RJ-11. Ver el diagrama de conexión de la Figura 6 y la Tabla 1.

Conector Macho RJ-11		Conector Hembra DB9		Colores
Pin	Función	Pin	Función	
2	RxD	3	TXD	Amarillo
5	TXD	2	RxD	Negro
4	GND	5	GND	Verde

Tabla 1.- Conexiones de los pines del conector RJ-11 con el DB9

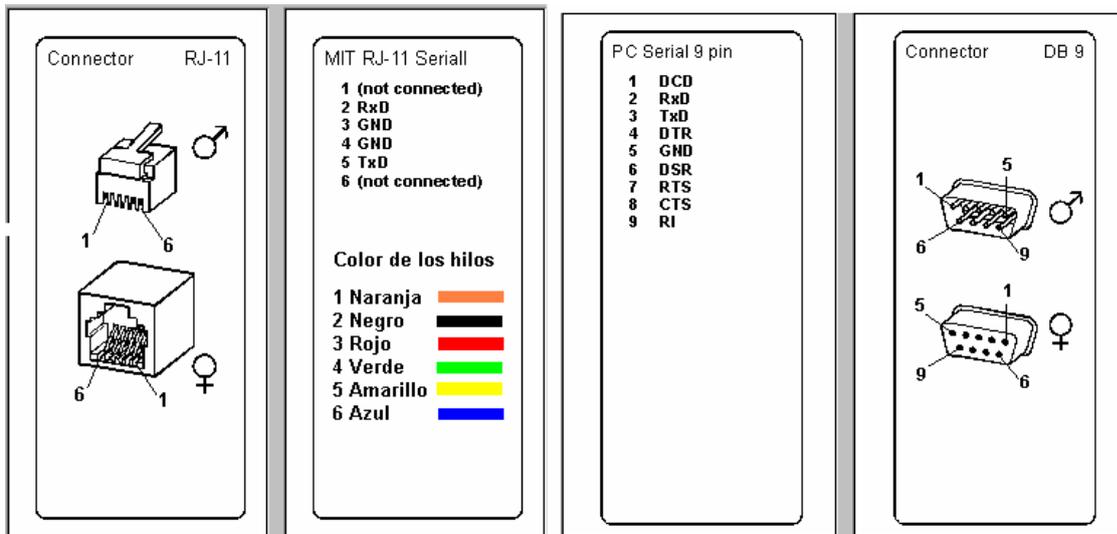


FIGURA 4.- PINES DE LOS CONECTORES RJ-11 Y DB9

Además la placa dispone de un conector RJ11, conectado a las líneas MCLR, RB3, RB6 y RB7 del microcontrolador, lo que permite realizar la conexión a un módulo ICD2 (Depurador en circuito), que como veremos es una herramienta de bajo costo que conectada al PC y en el mismo entorno del MPLAB, permite la visualización de los registros del microcontrolador en los puntos de ruptura que se establezcan. Pudiéndose ejecutar los programas en tiempo real o en modo paso a paso.



FIGURA 5.- CONEXIÓN DEL ICD A LA PLACA DE DESARROLLO

## 2.4. COMO UTILIZAR EL PROGRAMA BOOTLOADER

- 1°.- En primer lugar debemos obtener el programa en la siguiente web [http://www.ehl.cz/pic/pic\\_e.htm](http://www.ehl.cz/pic/pic_e.htm) , donde también se puede obtener el PIC donwloader V1.08.
- 2°.- Abrir el proyecto bootldr.pjt utilizando el programa Microchip MPLAB y cambiar los parámetros de la sección de usuario marcados dentro del ensamblador con las marcas <<<, tal y como se muestra en la Tabla 1, en la que hemos fijado el microcontrolador PIC16F877A, con un oscilador de 4MHz, una velocidad de transmisión de 9600 baudios y utiliza el pin 7 del PORTB como señal de disparo para ejecutar el bootloader cuando está a nivel bajo y correr el programa de usuario cuando está a nivel alto. Por tiempo entra en el programa de usuario después de haber pasado 0,3 segundos sin recibir la palabra de IDENTIFICACIÓN.

```

===== Sección fijada por el usuario =====

list p=16f877a
; <<< set type of microcontroller (16f873a or 16f876a)
; set same microcontroller in the project
#define ICD_DEBUG 0
; <<< if using MPLAB ICD Debugger, moves bootloader down 256 bytes to make room for it [0|1]

#define FOSC D'4000000' ; <<< set quartz frequency [Hz], max. 20 MHz
#define BAUD D'19200' ; <<< set baud rate [bit/sec]
#define BAUD_ERROR D'4' ; <<< set baud rate error [%]
#define TIME ; <<< set method of bootloader start PIN/TIME

; PIN : start on low level of trigger pin
; TIME: start on receive IDENT byte in TIMEOUT
#define TRIGGER PORTB,7 ; <<< only for PIN - set PORT_X,PIN_NR
#define TIMEOUT D'2' ; <<< only for TIME - set time [0.1s], max. 25 sec
#define WATCHDOGTIMER 0 ; <<< Watchdog timer default OFF/ON [0|1]
===== Fin de la sección fijada por el usuario =====

```

TABLA 2.- SECCIÓN DEL PROGRAMA BOOTLOADER QUE DEBE FIJAR EL USUARIO

- 3°.- Si se modifica alguno de los parámetros anteriores, debemos de volver a compilar el programa bootldr.asm.
- 4°.- Programar el microcontrolador con el archivo bootldr.hex utilizando cualquier programador de los utilizados tradicionalmente para programar los microcontroladores PIC, por ejemplo un programador tipo T-20 con el software ic-prog. Información sobre los mismos se puede obtener en la web [www.ic-prog.com](http://www.ic-prog.com)
- 5°.- Recompilar el programa de la aplicación para que no borre el bootloader. El bootloader usa los 256 bytes de la parte alta de memoria de programa (0x1F04-0x1FFF).

Llegado a este punto lo más aclaratorio es ver un ejemplo con un programa que utiliza este método de carga del programa de usuario en el microcontrolador.

El programa Parpadea.asm es un programa ejemplo que utiliza una la rutina Retardos.inc . Configura la línea RC5 como salida y enciende el LED conectado a dicha línea durante 200 ms, seguidamente apaga el LED durante otros 200 ms y repite el bucle de encender de nuevo el LED.

```

*****Parpadea.asm*****
;
;Enciende y apaga el LED 0 del puerto B
;
;
;*****DATOS*****
;
LIST      P=16F877A          ; Pocesador usado
INCLUDE <P16F877A.INC>      ; En este fichero se definen las etiquetas del PIC

CBLOCK 0x20

ENDC

;CODIGO*****
ORG      0x00
clrf    PCLATH
goto    Inicio

ORG      0x05
Inicio
bsf     STATUS,RP0          ;Selecciono el Banco 1.
movlw   b'00000110'
movwf   ADCON1              ;Puerto A digital
clrf    TRISB               ;Pongo como salidas el puerto B.
movlw   b'11011111'        ;Configura RC5 como salida y el resto como entrada
movwf   TRISC
bcf     STATUS,RP0          ;Selecciono el Banco 0.

Programa

bsf     PORTC,5              ;Enciendo el LED RC5
call    Retardo_200ms        ;Espero 200 milisegundos
bcf     PORTC,5              ;Apago el LED RC5
call    Retardo_200ms        ;Espero 1 segundo
goto    Programa            ;Salto a Programa

INCLUDE <RETARDOS.INC>

END

```

6°.- El mapa de memoria del Microchip PIC 16F876/877 con el bootloader instalado es el que se muestra en la Figura 7.

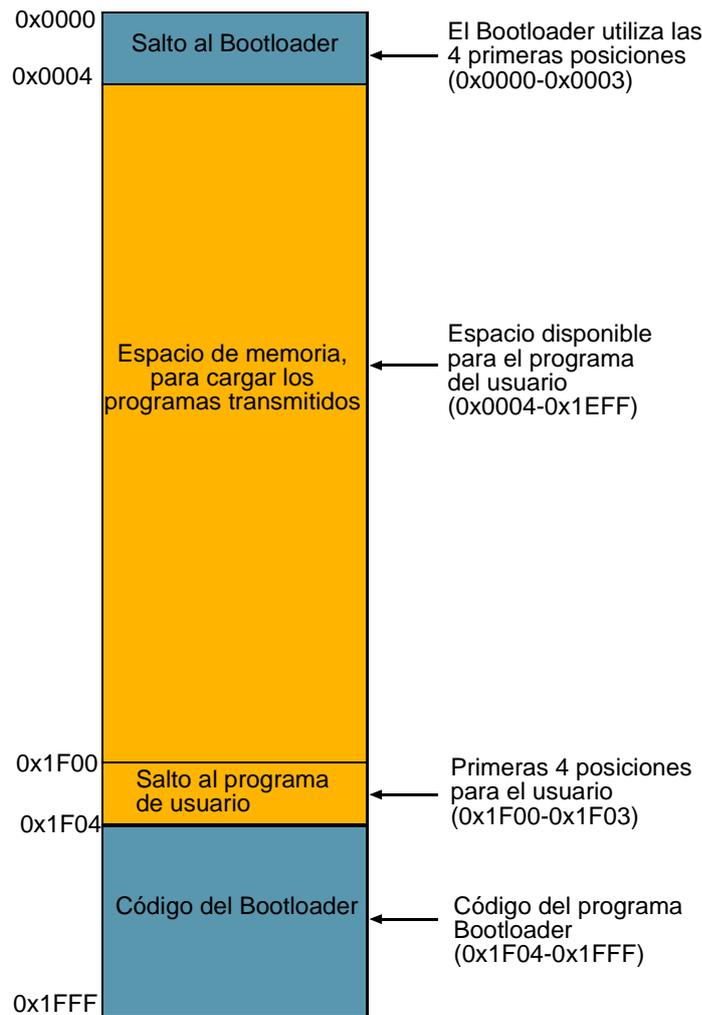


FIGURA 6.- MAPA DE MEMORIA DEL BOOTLOADER PARA UN MICROCONTROLADOR DE 8KBYTES

## **2.5. COMO REALIZAR LA CARGA DEL PROGRAMA EN EL MICROCONTROLADOR**

- 1º.- En primer lugar se conecta la placa de desarrollo al PC mediante un cable de modem nulo, tal y como se comento anteriormete.
- 2º.- Para realizar la carga del programa de usuario, vamos a utilizar un programa de "carga" freeware, se trata del PIC downloader de Petr Kolomaznik y que hemos conseguido en la web <http://www.microchip.com/> . Una vez instalado, al ejecutarlo, aparece una pantalla como la que se muestra en la Figura 7.

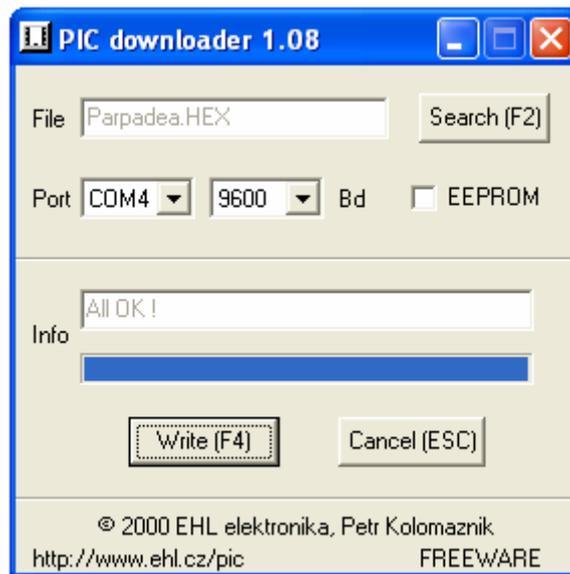


FIGURA 7.- PANTALLA DEL PROGRAMA PIC DOWNLOADER

- 3°.- Seleccionamos la velocidad de transmisión en baudios, en nuestro caso 9600, el puerto serie de comunicación, en el ejemplo hemos seleccionado el COM2 y buscamos el nombre del programa en código hexadecimal que queremos cargar en el microcontrolador, Parpadea.hex.
- 4°.- Este programa tiene las siguientes teclas de función rápida:
  - F2 - Búsqueda
  - F4 - Escribir
  - ESC - Cancelación
  - F10 - Salida
- 5°.- Seguidamente se pulsa el botón de RESET y el programa se carga en el microcontrolador PIC. Según se va realizando la carga, aparece un indicador de carga (línea azul) y al finalizar la carga suena una señal acústica.
- 6°.- Si volvemos a actuar manualmente sobre el pulsador de RESET, se ejecutará el programa de usuario.

**Características del programa PIC downloader 1.8 :**  
 Compatible con [el bootloader](#) de HI-TECH's [o de Shane Tolmie](#) para los microcontroladores PIC16F87x y PIC16F87xA.  
 Funcionamientos bajo Windows 9x/ME/NT/2000  
 Selección de la velocidad a partir del 2400 a 56000 Bd  
 Selección del puerto serie entre el COM1-6  
 Trabaja con datos de EEPROM en el archivo hexadecimal  
 No envía la palabra de los configuración a la posición 0x2007 o palabra de la identificación desde el bootloader  
 Mantiene las últimas configuraciones  
 Se incluye el código de fuente  
 El downloader del PIC se controla al realizar una señal de disparo del RESET en el microcontrolador  
 Se trabaja en ensamblador para todos aquellos programadores que no utilizan el lenguaje C. El bootloader es muy fácil de recompilar para otros microcontroladores actualizando el pin/tiempo y la frecuencias del cristal de cuarzo. Solo se necesita el programa MPLAB de Microchip que también es gratuito.  
 Se puede obtener en las siguientes webs: (<http://www.ehl.cz/pic/>) (<http://www.microchip.com/>)

## 2.6. LA PLACA DE EXPANSIÓN

Como ya hemos comentado, la placa de expansión dispone de un circuito integrado de tipo L293B que dispone de cuatro *drivers* emparejados de dos en dos por dos entradas de inhibición. La conexión realizada permite que cada pareja de *drivers* controle el sentido de giro de un motor tanto en sentido horario como antihorario a través de una única entrada de control, ya que la otra entrada siempre es la negada de la anterior al estar unido a través de un inversor. Por lo tanto si nos fijamos en el esquema de la figura 8, se puede comprobar que si RC1 está a nivel alto y RB2 también está a nivel alto, el motor gira en un sentido, debido a que la entrada 2 A tendrá el valor alto y la 1ª estará a nivel bajo, Por otro lado, si RB1 a nivel bajo, el motor gira en sentido contrario por que las líneas 1 A y 2 A tomaran los valores “0” y “1” respectivamente.

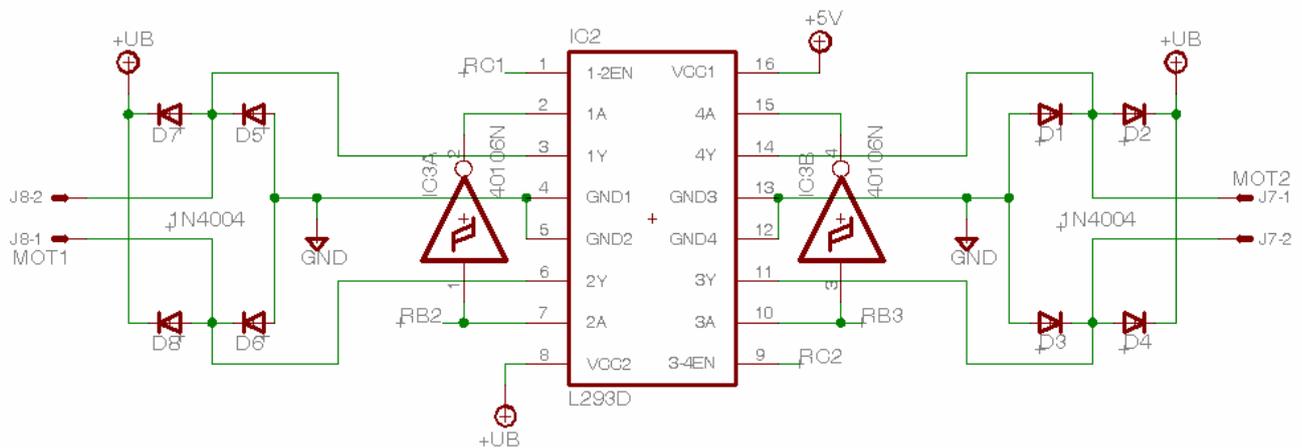


FIGURA 8.- DRIVERS DE CONTROL DE MOTORES

Además la placa dispone de seis entradas del tipo *Trigger Schmitt* con resistencias de polarización, para poder conectar a cada una de ellas directamente un sensor del tipo CNY70, un *bumper*, etc..

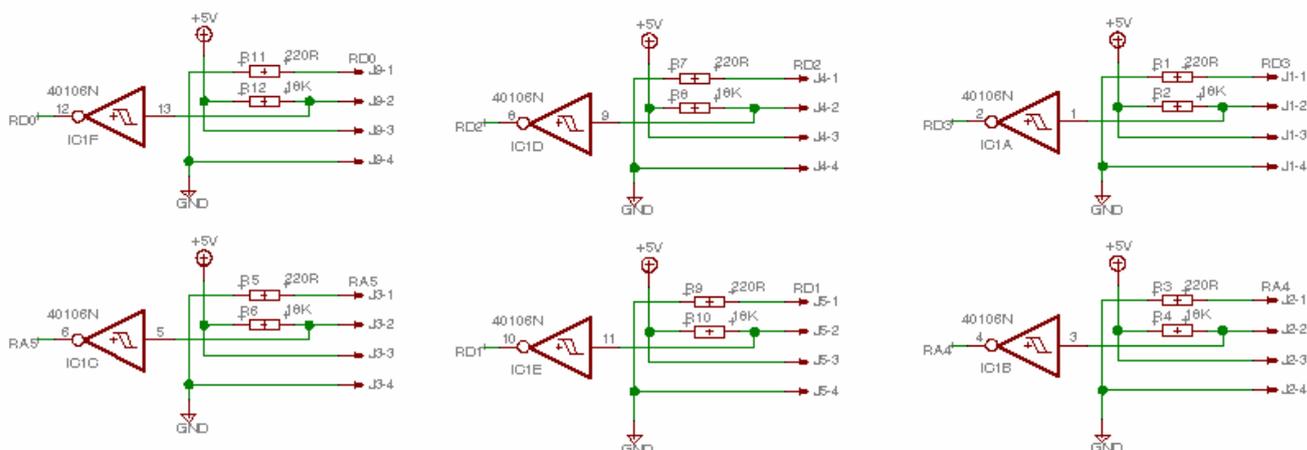


FIGURA 9.- ENTRADAS TRIGGER SCHMITT POLARIZADAS

Se ha incluido además una pantalla LCD de 2 líneas x 16 caracteres, conectado de manera que pueda ser controlado solo a través de un bus de datos de cuatro líneas (RD7:RD4) y las líneas de control RS, R/W y E conectadas s RE2:RE0 respectivamente.

Las rutinas de control del LCD están en el CD-ROM de la documentación con el nombre LCD\_4BIT.inc

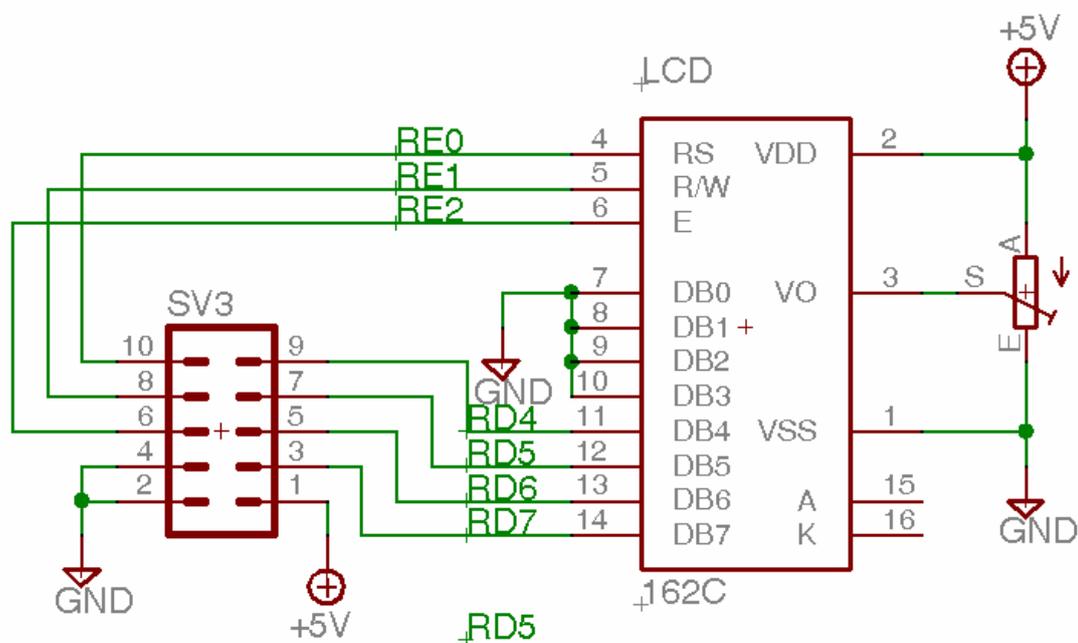


FIGURA 10.- CONEXIÓN DEL LCD