

Conexión y programación con el puerto paralelo

Las comunicaciones en paralelo se realizan mediante la transferencia simultánea de todos los bits que constituyen el dato (byte o palabra). Presentan la ventaja de que la transmisión puede ser más rápida. Sin embargo, las comunicaciones en paralelo no pueden ser implementadas para grandes distancias debido a que no es viable la conexión física de todas las líneas necesarias.

Las comunicaciones en paralelo propiamente dichas no han sido normalizadas, lo que sí se reconoce es la *norma Centronic*; para la conexión del PC a la impresora, mediante el envío simultáneo de 8 bits de datos (un byte), además de un conjunto de líneas de protocolo (*handshake o intercambio*). La operación más frecuente en la que interviene el puerto paralelo del PC es en el envío de datos a la impresora.

Los antiguos circuitos integrados que se incluían en las tarjetas de interface del puerto paralelo no permitían la recepción de datos, sólo estaban diseñados para el envío de información al exterior. Las versiones recientes de estas tarjetas de interface de puertos paralelo sí **permiten la recepción de datos** y dan la posibilidad, por ejemplo, de **intercambiar información entre PC a través del puerto paralelo**, siempre que se utilice el software adecuado.

La **norma Centronics** hace referencia a las características de la conexión entre un interface de puerto paralelo y una impresora. Las líneas son *latcheadas*, esto es, mantienen siempre el último valor establecido en ellas mientras no se cambien expresamente y los niveles de tensión y de corriente coinciden con los niveles de la lógica TTL, cuyos valores típicos son:

-Tensión de nivel alto:	5 V.
-Tensión de nivel bajo:	0 v.
-Intensidad de salida máxima:	2.6 mA.
-Intensidad de entrada máxima:	24 mA.

La norma Centronics establece el nombre y las características de 36 líneas eléctricas para la conexión entre el PC y la impresora.

En realidad, para la transferencia de las señales de datos y de control a través de la tarjeta de interface paralelo **sólo** se requieren **18 líneas**, las restantes son líneas de masa que se enrollan alrededor de los cables de señal para proporcionarles **apantallamiento** y protección contra interferencias. Por esto, las citadas tarjetas suelen incorporar un conector hembra DB-25, mientras que prácticamente todas las impresoras incorporan un conector hembra tipo Centronics macho de 36 pines,

Los cables comerciales para la conexión paralelo entre el PC y la impresora tienen una longitud de 2 metros, aunque no es recomendable que tengan una longitud superior a 5 metros si se desea una conexión fiable y sin interferencias.

En la siguiente tabla se describen todas las líneas del estándar Centronics, con indicación de su denominación y el número de pin que le corresponde, tanto en el conector tipo Centronics de 36 pines como en el conector DB-25. En esa tabla se indica que las 8 líneas correspondientes a los bits de datos (D0 a D7) son líneas de salida, pues así lo establece el estándar Centronics, sin embargo y sobre todo en las implementaciones más recientes, la circuitería asociada al interface del puerto paralelo puede ser tal que las líneas de datos pueden ser leídas desde el PC y, por tanto, ser consideradas como líneas **bidireccionales**, aunque sea en determinadas condiciones y con el software adecuado.

Descripción de los pines del Puerto paralelo:

N.º de pin		Señal	Sentido	Descripción
DB-25	DB-36		PC-PRN	
1	1	/STR	→	STROBE. Validación del dato (activa a nivel bajo). Un nivel L indica a la impresora que el dato es válido.
2	2	D0	→	Bit 0 de datos.
3	3	D1	→	Bit 1 de datos.
4	4	D2	→	Bit 2 de datos.
5	5	D3	→	Bit 3 de datos.
6	6	D4	→	Bit 4 de datos.
7	7	D5	→	Bit 5 de datos.
8	8	D6	→	Bit 6 de datos.
9	9	D7	→	Bit 7 de datos.
10	10	/ACK	←	ACKNOWLEDGE. (Activa a nivel bajo). Un nivel L indica que la impresora está en disposición de recibir un nuevo dato.
11	11	BSY	←	BUSY. Ocupada. Un nivel H indica que la impresora está ocupada y no puede recibir datos.
12	12	PAP	←	PAPER END. Sin papel. Un nivel H indica que la impresora se ha quedado sin papel.
13	13	OFON	←	ON LINE. Conectada. Un nivel H indica que la impresora está conectada y en línea.
14	14	/ALF	→	AUTO LINE FEED. Cambio de línea automático (Activa a nivel bajo). Un nivel L indica a la impresora que cuando reciba un retorno de carro debe hacer también un cambio de línea automáticamente.
15	32	/ERR	←	ERROR. (Activa a nivel bajo). Un nivel L indica que se ha producido un error en la impresora (buffer lleno, impresora fuera de línea, etc.).
16	31	/INI	→	INITIALIZE PRINTER. Inicialización (Activa a nivel bajo). Un nivel L inicializa o provoca un reset en la impresora (si la impresora lo admite).
17	36	/DSL	→	SELECT. (Activa a nivel bajo). Un nivel L selecciona o pone on line la impresora (si la impresora lo admite).
18-25	19-30 33	Masa		Referencia de tensión para las señales.
—	16	0V		—
—	17	Chasis		Conexión al chasis del equipo.
—	18	+Vcc		Tensión de +5 V.
—	34, 35	—		No utilizada.

El puerto paralelo en un PC

Todos los ordenadores tipo PC están equipados, al menos, con una tarjeta de interface paralelo, frecuentemente junto a un interface serie. Como sistema operativo, el DOS puede gestionar hasta cuatro interfaces de puertos paralelo, LPT1 , LPT2 , LPT 3 y LPT4, además, reserva las siglas PRN como sinónimo del LPT 1, de modo que puede ser tratado como un archivo genérico. En el byte **0040:0011** del BIOS almacena el número de interfaces de puertos paralelo que se hayan instalado en el equipo. La dirección de entrada/salida de cada uno de los puertos paralelo y el número de puertos instalados en un PC se muestra en la pantalla inicial de arranque del equipo es frecuente, casi estandard que las direcciones de los dos primeros puertos paralelo sean las siguientes:

LPT1 = 0x378 Hexadecimal

LPT2 = 0x278 Hexadecimal

Las tarjetas del puerto paralelo tiene una estructura muy simple; consta de *tres registros: de control, de estado y de datos*. Todas las señales que intervienen en el puerto tienen asociado un bit en uno de esos registros, de acuerdo con las funciones asignadas a cada línea en particular.

- **El registro de datos**

Es de tipo latch de 8 bits, que puede ser leído y escrito desde el procesador. Es el registro donde el procesador, en operaciones de salida (OUT), pone el dato que se quiere enviar a la impresora y su **dirección** coincide con la **dirección base del puerto paralelo** (**0x 378** en LPT 1). En la Figura 32.1 se muestra la distribución de los bits de este registro y los pines asociados a cada uno de ellos en el conector DB-25.

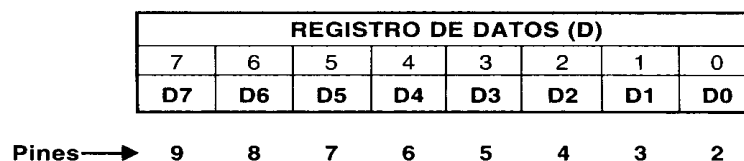
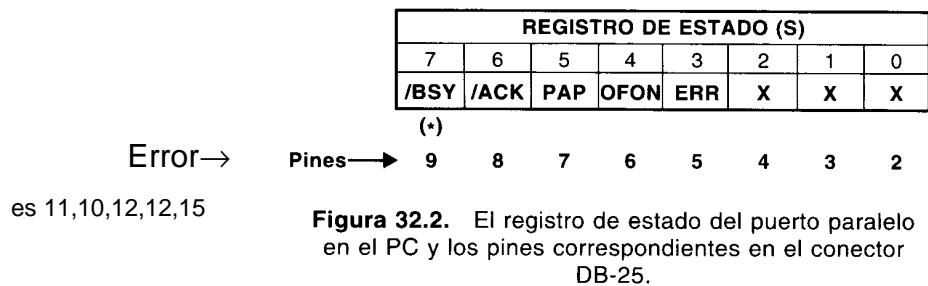


Figura 32.1. El registro de datos del puerto paralelo en el PC y los pines correspondientes en el conector DB-25.

- **El registro de estado**

El registro de estado indica la situación actual de la impresora conectada al puerto, de acuerdo con los niveles de tensión que tengan las líneas ACK, BSY, PAP y OF/ON , lo que permite controlar el comportamiento de la impresora.

Se trata de un **registro de entrada** (Lectura) de información, su dirección se obtiene sumando 1 a la dirección base del puerto (**0x379** en LPT1).



Los bits de este registro se designan según se muestran en la Figura 32.2, en la que el símbolo «/» delante del nombre del bit indica que es *activo a nivel bajo*. Pero el bit 7 además (/ BSY) del registro de estado (bit 7) es invertido por el hardware y, por tanto, la línea tiene un nivel complementado al que aparece en ese bit.

El significado que tienen los bits de este registro es el siguiente:

- ◆ Si el bit 7 (/BSY → Busy) está a 0, significa que la impresora está ocupada (buffer de impresión lleno, procesando información, pendiente de inicializar, etc.).

- ◆ El bit 6 (/ACK → Acknowledge) indica que se ha producido una transferencia correcta: cuando del puerto paralelo se transfiere un byte a la impresora, la impresora activa la línea ACK de reconocimiento del carácter y, como consecuencia, el bit ACK del registro de estado pasa a nivel bajo; cuando el bit ACK está a nivel alto, significa que la impresora está ocupada y no se pueden realizar envíos.

- ◆ El bit 5 (PAP → Paper) si está a 1, señala que la impresora **no** dispone de papel.

- ◆ El bit 4 (OF/ON → Line Off) indica cuando está a 1, que la impresora no está en línea.
- ◆ El bit 3 (ERR) si está a 0, indica que se ha producido un error de impresora (mal funcionamiento, falta de papel, impresora fuera de línea ...).

Los bits 0,1 y 2 no se utilizan.

- **El registro de control**

El **registro de control** permite controlar las transferencias de información con la impresora, y puede ser escrito y leído desde el microprocesador. Es un registro de entrada/salida cuya dirección se obtiene sumando 2 a la dirección base del puerto (0x37A en L PT 1). Los bits de este registro se designan en la Figura 32.3, donde el símbolo «/» delante del nombre del bit indica que es *activo a nivel bajo*.

REGISTRO DE CONTROL (C)							
7	6	5	4	3	2	1	0
X	X	X	IRQ	DSL	/INI	ALF	STR
				(*)		(*)	(*)
				17	16	14	1 ← Pines

Figura 32.3. El registro de control del puerto paralelo en el PC y los pines correspondientes en el conector DB-25.

El símbolo (*) indica que los bits STR, ALF y OSL del registro de control son invertidos por el hardware con relación a las líneas correspondientes al cable de conexión, por lo que el nivel de los bits 0,1 y 3 del registro es complementado con relación a las líneas correspondientes.

El significado que tienen los bits de este registro es el siguiente:

- ◆ El bit 4 (IRQ) es el que permite controlar la generación de interrupciones de tipo hardware desde el puerto paralelo. Si este bit está a 1, el interface paralelo puede generar la petición de interrupción IRQ7 (en LPT1), que se corresponden con la interrupción 0x0Fh respectivamente del procesador 80X86. Esta petición de interrupción se produce cuando se da una transición H→L en la línea ACK.

◆ El bit 3 (DSL) : La mayoría de las impresoras paralelo IBM-compables, no utilizan esta línea y son activadas con un pulsador de on-line. El bit 2 (INI) produce una inicialización de la impresora (es poco utilizado).

◆ Si el bit 1 (ALF) está a nivel alto, la impresora produce automáticamente un *cambio de línea* (LF) cada vez que recibe un *retorno de carro* (CR).

◆ El bit 0 (STR) controla la línea que permite validar el dato existente en el registro de datos. La puesta a 1 del bit STR genera un impulso corto que indica a la impresora que el carácter del registro de datos es válido y debe ser aceptado. Así pues, cada vez que se precise enviar un carácter, no basta con ponerlo en el registro de datos, sino que hay que hacer un reset en el bit STR del registro de control y validar el dato volviendo a poner un 1 en ese bit.

◆ Los bits 5, 6 y 7 no se utilizan.

Entradas y salidas por el puerto paralelo

Al hablar de operaciones de entrada y salida por el puerto paralelo no debe olvidarse que, inicialmente, este elemento se desarrolló de acuerdo con el estándar Centronics con el fin, casi exclusivo, de que el PC pudiese enviar datos en paralelo a la impresora conectada, no se pensó en la posibilidad inversa: que el PC pudiese recibir datos a través de ese puerto.

Las operaciones de entrada y salida de información a través del puerto paralelo en el PC las realizaremos **gestionando el puerto paralelo en el nivel de registros**, es decir, programando directamente los circuitos integrados o chips que constituyen la tarjeta de interface, lo cual permitirá aprovechar al máximo todas las posibilidades que ofrezca realmente el hardware de la tarjeta de interface.

Características E/S

Cuando usamos el puerto paralelo para otro cometido distinto al original, solo podemos hablar de 12 líneas de salida de información desde el ordenador:

- pines del 2 al 9 → registro de datos
- pines 1,14, 16 y 17 → registro de control

15 líneas de entrada al mismo:

- pines 2-9→ registro de datos
- pines 10,11,12,13 y 15 → registro de estado

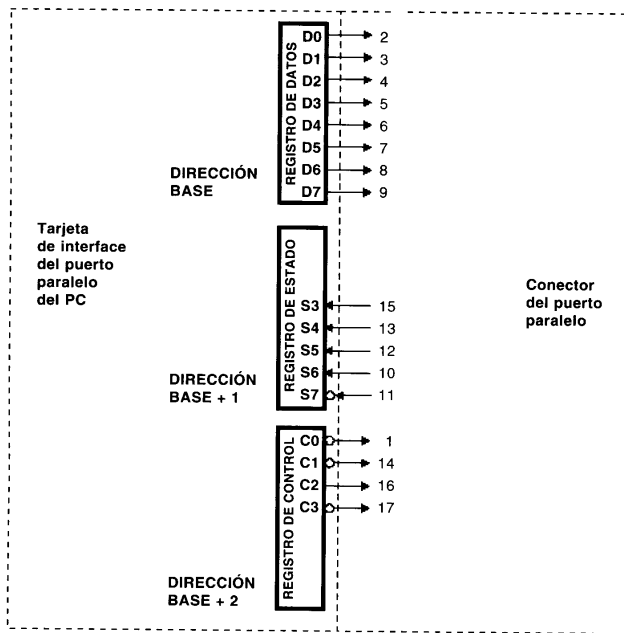


Figura 32.4. Líneas en el conector DB-25 del puerto paralelo.

Esto hace del puerto paralelo un interface de comunicación con el exterior bastante flexible.

El **registro de estado** es de sólo lectura. Cuando se **lee** este registro, lo que se recibe es el estado lógico de los pines 10, 11, 12, 13 y 15 del conector DB-25 (el bit S 7 contiene el complemento del estado

de la línea). Los tres bits de menor peso (S0-S2) no se utilizan y, habitualmente, se encuentran a nivel alto

El **registro de control** es parcialmente bidireccional. Cuando se **escribe** en los cuatro bits de menor peso (C0 - C3) lo que se hace es establecer el nivel lógico de los pines C 2 de forma directa y C0, C 1 y C 3 de forma complementada.

Los tres bits de mayor peso (C 5 C 7) no se utilizan.

De forma experimental, se ha podido comprobar que, sólo en algunas tarjetas de interface paralelo, el bit C 6 del registro de control influye en la configuración, de modo que si C 6 = 0 las l líneas de datos se configuran como ENTRADAS y si C 6 = 1 las líneas de datos se configuran como SALIDAS. Otras tarjetas, sobre todo si son bidireccionales, no permiten el cambio de nivel de ese bit.

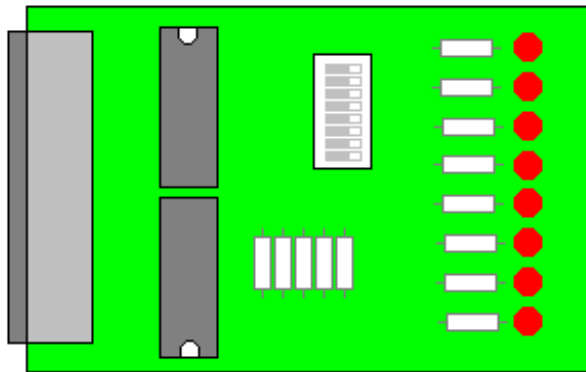
El **registro de datos** es de tipo latch de lectura y de escritura, de modo que cuando se realiza una operación de escritura (OUT) el dato se carga en los bits correspondientes y las líneas asociadas del conector tienden a alcanzar la tensión correspondiente a ese estado.

En algunas ocasiones las líneas de datos de la tarjeta de interface paralelo (Centronics) son bidireccionales, pero la **etapa de salida** se ha construido mediante buffers con transistores en **colector abierto**. En este caso, el hecho de que las operaciones de entrada y salida se hagan por las mismas líneas, condiciona notablemente el proceso de lectura, ya que con esa configuración electrónica de las líneas de datos (D0 - D7), los valores lógicos leídos **dependerán del nivel lógico presente en el registro y del valor de tensión en la línea** (que no tienen por qué coincidir) de acuerdo con lo mostrado en la Tabla 32.2.

Tabla 32.2. Niveles lógicos y de tensión en el puerto paralelo.

Nivel lógico de los bits en el registro de datos	Nivel de tensión en los pines del conector	Nivel lógico leído en una operación de lectura del puerto paralelo
L	L	L
L	H	L
H	L	L
H	H	H

A la vista de la Tabla 32.2, lo que se deduce es que, **si se va a realizar una operación de lectura** sobre el puerto paralelo, lo que se va a **leer realmente** es la operación **AND lógica entre el nivel lógico del registro y el nivel lógico de la línea**, lo que implica que, **si se desea realizar una lectura real** del estado de las líneas, deberá **escribirse antes el dato 0xFF** en el registro de datos del puerto paralelo.



La tarjeta de E/S que se va a utilizar es un simulador de impresora, donde se pueden activar los 8 bits de datos y realizar lecturas del registro de control.

Las instrucciones en Turbo C que dan acceso a los puertos están en la librería **conio.h** y son, para *escribir*,

outp (puerto ,dato), y para *leer* **inp** (puerto).

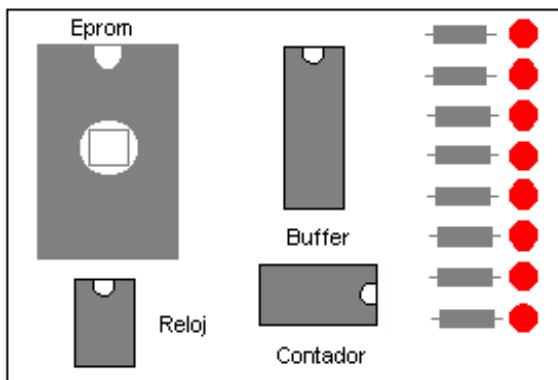
La tarjeta consta de 2 búferes para aislar las conexiones desde al PC y el exterior. Cada Led representa 1 bit del registro de datos, siendo el Led superior el correspondiente a D0 y el inferior al D7. Debemos pensar que si tenemos 8 bits de datos, cada uno de ellos se encendería de acuerdo a la potencia de 2 que nos da su peso, así los datos 1,2,4,8,16,32,64 y 128, encenderían cada uno de los Leds D0 a D7. Si queremos activar el Led de D3 escribiríamos la siguiente instrucción en C:

```
outp ( 0x378, 8)
```

↑_{instrucción} ↑_{puerto} ↑_{dato}

Uso de la tarjeta como secuenciador

Si pensamos en la estructura real de un circuito digital que realice una secuenciación veremos que se asemeja a este diagrama:



En una memoria Eprom guardamos los datos y mediante un contador fijo vamos leyendo las distintas posiciones. Este circuito simple tiene el inconveniente poder realizar sólo una secuencia programada, aunque con alguna lógica adicional podemos incluir varios programas y seleccionarlos a

posteriori. Vemos que es necesario un circuito contador y también presenta el

problema de modificar el programa de la secuenciación, que debe ser extraída la memoria, borrarla completamente, y regrabar el nuevo programa y todos los anteriores.

Si utilizamos una tarjeta de E/S nos podemos olvidar de la Eprom e implementarla mediante una matriz. Podemos declarar una matriz de caracteres, cuyos valores están dentro del rango de 8 bits y se puede inicializar la matriz con el programa o secuencia que necesitemos. Luego mediante bucles del tipo **for** o **while** se puede recorrer la matriz, e incluso variar el recorrido y el programa en función de ciertas condiciones.

Un ejemplos :

- **Recorrido del puerto:**

```
#include <conio.h>
void main (void)
{
    char i=0;
    for (i=0;i<256;i++)
        outp (0x378,i);
}
```

- **Parpadeo de todos los leds de forma simultánea durante 10 veces:**

```
#include <conio.h>
void main (void)
{
    char i;
    for (i=0;i<10;i++)
        outp (0x378,255);
        outp (0x378,0);
}
```

- **Encendido secuencial ascendente – descendente**

```
#include <conio.h>
void main (void)
{
    char datos [ ]={1,2,4,8,16,32,64,128};
    char i;
    outp (0x378,0);
    for (i=0,i<8,i++)                // Recorrido ascendente
        outp (0x378,datos[i]);
    outp (0x378,0);
    for (i=7,i<0,i--)                // Recorrido descendente
        outp (0x378,datos[i]);
}
```

Cálculo y conversión de señales analógicas

Medir una temperatura con un PC es relativamente fácil. Necesitamos un sensor que traduzca los grados centígrados a un voltaje y seguidamente convertir ese voltaje en una secuencia de ceros y unos, lo que se conoce como conversión analógica/digital (AD). El problema es que, al igual que ocurre con muchos ordenadores, en un portátil de 1,5 Kg. y 25 cm. de largo no se puede insertar ninguna tarjeta comercial, lo que obliga a una profunda reflexión para encontrar soluciones. Para complicar más las cosas, tampoco se dispone de ninguna fuente de alimentación a mano, a excepción de las baterías, Las cuales no podemos utilizar.

Finalmente debemos proporcionar unos valores fiables y útiles. Así, un conversor AD de 10 bits, nos proporciona, en un rango de 0 a 100 grados, una resolución de una décima de grado, aunque también dependerá de los componentes.

Uno de los posibles convertidores que cumplen todos nuestros requerimientos es el LTC 1092 de *Linear Technology*. Su funcionamiento es el siguiente:

El integrado LTC 1092 comienza su ciclo cuando la línea de selección (CS) se pone a cero. Después de unos 2 ciclos de estabilización el integrado proporciona una conversión por aproximaciones sucesivas, para después transmitir el resultado mediante 10 pulsos de reloj (1 por bit). Esta señal será generada por el ordenador. La forma de hacerlo es fácil. Se pone una de las salidas del puerto paralelo alto, se espera un poco, después se vuelve a poner a nivel bajo y volvemos a esperar otro poco; y así sucesivamente. En medio de este proceso leemos el valor que nos proporciona el convertidor. La frecuencia de muestreo debe ser entre 10KHz y 500KHz. El resto de las líneas del puerto paralelo se situarán a nivel alto para que todas ellas en paralelo puedan alimentar al circuito.

Otros componentes:

Todos los convertidores AD necesitan una tensión de referencia que es aquella que equivale al fondo de escala del convertidor. Si el circuito es de 10 bits podemos alcanzar hasta $2^{10}-1 = 1023 \rightarrow V_{ref}$. y $0 \rightarrow 0$ voltios. Mediante el integrado LM336 de National generamos una tensión estable y prácticamente inmune a cambios de temperatura centrada en 2.49v y gracias a una resistencia ajustable regulamos a 1 voltio. (Técnicamente el circuito puede ajustarse hasta medir tensiones de 2.5v de fondo de escala). Ya sólo nos queda por describir el último componente, que es el sensor de temperatura. Se trata del LM35 que ofrece una tensión /°C de 0.01 voltio, así $0^{\circ} \rightarrow 0$ voltios y $25^{\circ} \rightarrow 0,25$ voltios. Como el fondo de escala es de 1 voltio la temperatura máxima es de 100 grados.

Ahora se requiere de un programa en el PC con el fin de transformar las señales analógicas en digitales. Como podemos ver en listado adjunto podemos direccionar el puerto paralelo realizando una operación de salida (OUT) en puerto correspondiente, que por defecto es 0x378h para el LPT1

Recordemos que **todas las salidas deben ponerse a nivel alto, exceptuando aquellas donde van a existir variaciones.** La programación del chip LTC1092 es relativamente sencilla.

```

read_ad_one_mtm10()
{
  int base=0x378,val=0xff;
  int q,n,b=512,tot=0;

  #define LOW outp(base,val-3);fwait(10);
  #define HIGH outp(base,val-1);fwait(10);

  outp(base,val-2); fwait(10);

  LOW; HIGH; LOW; HIGH;

  /* conversión
  for (q=0; q<10; ++q) {
    LOW; HIGH;
  }
  */

  /* recogida de datos */
  for (q=0;q<10;++q) {
    LOW; HIGH;
    n=(inp(base+1)&128);
    n=(n)?b:0;
    tot+=n;
    b>>=1;
  }
  return tot;
}

```

Si seguimos el esquema, el bit 0 del puerto corresponde con la señal CS mientras que el bit 1 corresponde con la señal de reloj. Para la entrada de datos se utiliza el bit 7 del registro de estado (pin 11). Tenemos dos macros, denominadas HIGH y LOW que determinan el estado del reloj.

El algoritmo opera de la siguiente forma a la hora de realizar una conversión: en primer lugar, con el reloj en estado bajo, procedemos a activar la línea CS, que debe pasar del estado alto de reposo al bajo de comienzo de actividad. Deberá mantenerse en este estado bajo hasta terminar el ciclo, momento en el cual se alzarán. Seguidamente, se produce un ciclo de reloj, es decir , se pone el reloj a nivel bajo y acto seguido a nivel alto.

En este ciclo no se realiza ninguna operación. A partir de este momento, todo está preparado para recibir los bits uno a uno. En cada ciclo de reloj leeremos el puerto de entrada recibiendo un bit. Dicho de otra manera, subimos el bit de reloj, lo

bajamos y leemos un bit en el puerto. Esta operación la realizaremos diez veces, obteniendo así diez bits, de forma que nos llega **primero el de mayor peso (D9)** [en el programa b=512 porque son 10 bits] y finalmente el de menor (D0).

El resultado obtenido puede verse influenciado por inestabilidades fruto de la electrónica, por lo que es recomendable el empleo de un filtro que estabiliza la lectura. En general, las medidas ofrecidas por el tienen una gran estabilidad, aunque es posible que en ocasiones se pueda producir algún "ruido" externo que estropee el valor de la señal.

Para minimizar estos problemas, se suele emplear algún algoritmo que obtenga la media de tres o cuatro lecturas consecutivas, lo que suaviza la respuesta eliminando picos indeseados.

Bibliografía utilizada:

- Técnicas de programación
Cap.32.2 Ed. McGraw Hill
- Revista PC World, Octubre'94. Sección de programación