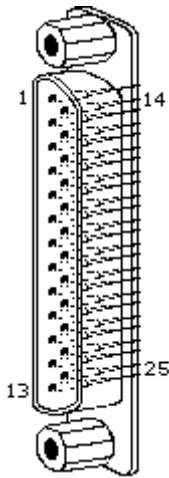

Control Básico del Puerto Paralelo



Desde sus comienzos el puerto paralelo ha sido la principal interfaz para la conexión de periféricos a la computadora, esto se debe a su versatilidad y a que todas las computadoras poseen al menos uno. Pese a su creciente popularidad siempre fue un reto manejar este puerto debidamente por causa de la gran cantidad de sucesivas modificaciones que se le han realizado al diseño original.

Físicamente el puerto paralelo es identificado por el usuario por su característico conector de 25 pines (hembra). Es importante diferenciar bien el conector del puerto paralelo de otros conectores, el texto está orientado totalmente al conector descrito (DB25), aunque es posible encontrar viejas placas controladoras con un conector de 36 pines tipo Centronics; se debe tener especial precaución al identificar el conector ya que algunos, si bien cada vez menos, dispositivos SCSI utilizan el mismo tipo de conector para su enlace con dispositivos externos. Es posible que una computadora posea más de un puerto paralelo, generalmente se pueden tener hasta tres, aunque Windows y otros programas tienen capacidad para manejar hasta cuatro de ellos; se debe tener en cuenta que el hecho de que estén instaladas las placas controladoras, no significa que los puertos estén en condiciones de ser utilizados, para esto deben estar configurados apropiadamente.

Debido a los cambios que se le han hecho al puerto paralelo, se definieron a lo largo de su historia distintos estándares, por lo que debemos conocer por lo menos de forma breve cada una de las normas para saber como es el puerto con que estamos trabajando.

Cada puerto paralelo posee una dirección Base, es decir, una dirección con la que se identifica al puerto, y según el tipo de puerto de que se trate, podrá tener además otra direcciones (Base+1, Base+2, Base+3, etc.). Actualmente las direcciones por defecto para los puertos son las siguientes:

Puerto	Dirección base
LPT1	0x378
LPT2	0x278
LPT3	0x3BC

El BIOS (Basic Input/Output Services) construye una tabla en memoria cuando arranca la computadora desde la dirección 40:08h a 40:0Dh que contiene las direcciones base de cada puerto paralelo instalado, leyendo esta tabla de manera ordenada se puede conocer la dirección base de cada puerto. En lenguaje C la tabla se puede leer como se muestra a continuación.

```
Peek(0x40 , 6 + 2 * LPTnum);
```

Tipos de Puerto

Es posible encontrar los siguientes tipos de puertos, por orden de aparición.

1.- SPP: Es el puerto más sencillo, tiene tres buses o registros, en la dirección Base: bus de datos, éste es de 8 bits y todos sus pines son salidas; en la dirección Base+1: bus de estado, en éste registro sólo 5 pines se encuentran cableados al conector, y todos ellos son entradas, en las impresoras son los que llevan la información de Papel Atascado, Falta de Papel, etc., una de las entradas (el bit 7) está complementada; por último, en la dirección Base+2: bus de control, este bus tiene cuatro salidas, de las cuales tres están negadas.

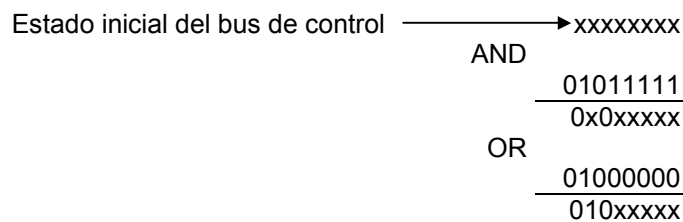
- Dirección Base + 0: registro o bus de datos
- Dirección Base + 1: registro o bus de estado
- Dirección Base + 2: registro o bus de control

	LPT1	LPT2	LPT3
Bus de Datos	0x378	0x278	0x3BC
Bus de Estado	0x379	0x279	0x3BD
Bus de Control	0x37A	0x27A	0x3BE

2.- Bidireccional (PS/2): Este puerto tiene las mismas características que el SPP con la diferencia que el bus de datos es bidireccional, es decir que cambiando el estado de algunos bits del bus de control se puede actuar sobre el bus de datos haciendo que trabaje en su totalidad como entradas o como salidas (no se puede configurar algunos pines como entrada y otros como salida).

- Entrada: para configurar el bus de datos como entradas, se deben poner en nivel alto el bit 5 y 7 del bus de control y en nivel bajo el bit 6 del bus de control.
- Salida: para configurar el bus de datos como salidas, se deben llevar a nivel bajo el bit 5 y 7 del bus de control y a nivel alto el bit 6 del bus de control.

Para hacer esto utilizaremos el concepto de máscara: supongamos que el bus de control tiene contenido X (suponiendo que el registro es de un byte: xxxxxxxx), entonces, a los bits que queramos poner en cero, les haremos una operación AND, con un número que tenga cero en los bits del mismo peso, y para poner a nivel alto, lo que debemos hacer es realizar de forma análoga una OR. En el ejemplo se configura el bus de datos como entrada.



3.- EPP (Enhanced Parallel Port): Este tipo de puerto tiene la característica de ser tan rápido como el bus del sistema (ISA), alcanzando velocidades de transferencia de hasta 1Mb por segundo, fue desarrollado por Intel, Xircom y Zenith y tuvo amplia aceptación por el resto de los fabricantes, quienes comenzaron a construir sus puertos de tipo EPP, aunque no totalmente compatibles con los creados por Intel, lo que llevó a que se produjera un estándar denominado EPP 1.7, que tiempo después se combinó con el estándar IEEE 1284, que describe los puertos bidireccionales de alta velocidad para impresoras y se formó un estándar que actualmente se denomina IEEE 1284 EPP. Un puerto paralelo del estilo IEEE1284 soporta múltiples modos: SPP, bidireccional PS/2, EPP y ECP. Otra particularidad de este puerto es que posee otras cinco direcciones (desde Base+3 a Base+7), aunque esto hace que no pueda existir un tercer puerto en la dirección 3BCh, ya que de ser así se solaparía con las direcciones asignadas a video. Es por esto que cuando se creó el puerto EPP, la dirección que antes solía ser el primer puerto, pasó a ser el tercero.

4.- ECP (Extended Capabilities Port): Puede, al igual que el EPP, escribir y leer a la velocidad del bus, fue desarrollado por Microsoft y Hewlett Packard, y se distingue del anterior por tener DMA (Direct Memory Access), FIFO (First In First Out) y compresión de datos. A su vez puede emular los puertos del tipo SPP y bidireccional, y si bien la norma no contempla los EPP, algunos fabricantes suelen usar algún bit no utilizado por ECP para la configuración del puerto como EPP. El puerto ECP agrega además los registros desde Base+400h; a Base+402h, en el cual está el registro ECR (Extended Control Register), mediante el cual se pueden configurar los diferentes modos, el significado de los bits es el siguiente:

Bit	Significado
7:5	000 SPP (Compatible ISA)
	001 Bidireccional (compatible PS/2)
	010 Centronics rápido (Compatible ISA FIFO)
	011 ECP
	100 Reservado (EPP)
	101 Reservado
	110 Prueba
	111 Configuración
4	Deshabilita interrupciones de ERROR
3	Habilita DMA
2	Deshabilita el servicio de interrupción FIFO/Terminal Count
1	Sólo lectura (FIFO lleno)
0	Sólo lectura (FIFO vacío)

ACLARACIÓN: Si el bit 3 es puesto a uno y el 2 a cero, se generará una interrupción en DMA - Terminal Count, por lo que este bit deberá ser puesto a cero para limpiar la bandera de interrupción TC.

Si el puerto se encuentra configurado en el modo SPP o bidireccional se puede cambiar a otro modo, por otro lado, si se está en otro modo se debe pasar primero a SPP o bidireccional y luego al modo que se desee; a su vez, si estamos en el modo 010 ó 111 y el bit de dirección está en cero, debemos esperar a que la FIFO se borre antes de cambiar a los modos 000 o 001.

En el modo 111, la dirección Base+400h será el registro de configuración A y la dirección Base+401h será el registro de configuración B, aunque no se explicará cada bit de estos registros puesto que escapa a los contenidos exigidos del curso.

Detección del Tipo de Puerto

Como ya se ha dicho el puerto puede ser de tipo SPP (común), bidireccional (compatible con el estándar IBM PS/2), EPP o ECP. Para conocer que tipo de puerto es el que se tiene se debe comenzar testeando el de mayor complejidad y seguir en orden hasta el de menor complejidad.

PRECAUCIÓN: Si la dirección base del puerto que se va a evaluar es 3BCh, no se debe realizar la prueba de puerto ECP ni EPP, puesto que estas pruebas implican alterar registros correspondientes a direcciones asignadas a video.

Detección de Puerto ECP

Para detectar si el puerto es de tipo ECP se debe leer el registro de control extendido (ECR) en la dirección Base+402h y verificar que el bit 0 esté en uno y el bit 1 esté en cero; estos bits no deben ser susceptibles de los cambios producidos en los bits 0 y 1 del bus de control, por lo que otra prueba a realizar es poner en bajo el bit 0 y en alto el bit 1 del bus mencionado, hecho esto, verificar que los bits del registro extendido de control no haya cambiado; por último, los bits 0 y 1 del registro extendido de control son de sólo lectura, lo que implica que no será posible hacerlos cambiar de estado, entonces podemos escribir un dato que conmute los estados de los bits 0 y 1 y luego leer el estado de esos bits, como lo que se lee es lo que hay en el registro y no lo que se envió, podemos decir que si los bits no han cambiado de estado y se han verificado las pruebas anteriores, estamos en presencia de un puerto ECP.

Detección de Puerto EPP

Si el puerto fallase en las pruebas para determinar si es ECP, debemos proseguir efectuando pruebas para comprobar, ahora, si es EPP. Para esto, se pone en nivel bajo el bit 7 del bus de estado y el bit de Timeout (bit 0 del registro de estado), debido a la diversidad de fabricantes existen distintos métodos para limpiar este último bit, por eso debemos contemplar todos ellos para asegurar el funcionamiento en la mayor cantidad de máquinas posible, lo

primero que haremos será escribir el bit dos veces con un uno y luego pasarlo a cero, podemos verificar ahora que el bit en cuestión esté en nivel bajo, de no ser así, ya podemos descartar que se trate de un puerto de tipo EPP. La segunda parte de la prueba consiste en escribir un dato en el registro Base+3, y al no haber ningún periférico que responda a la escritura realizada, el bit de Timeout deberá cambiar su estado lógico a un uno, de no ser así debemos seguir con las pruebas para determinar si el puerto es bidireccional o SPP.

Detección de Puerto Bidireccional (PS2 Compatible)

En esta prueba lo que haremos será configurar el bus de datos como entrada y verificar que al sacar un dato por el puerto no sea retenido.

Detección de Puerto SPP

Para detectar si un puerto es SPP verificamos la retención de los datos enviados, para lo cual enviamos un par de valores al bus de datos y comprobamos (en la dirección base) que los valores leídos sean lo que se escribieron, es decir que el dato enviado haya sido retenido por el puerto.

El Puerto Paralelo a Nivel Físico

	Dirección del puerto (hexadecimal)			Bit	Pin DB-25	Entrada o Salida	Función Especial Asignada al Pin
	LPT1	LPT2	LPT3				
Bus de Datos	378	278	3BC	0	2	Entrada/Salida	-
				1	3	Entrada/Salida	-
				2	4	Entrada/Salida	-
				3	5	Entrada/Salida	-
				4	6	Entrada/Salida	-
				5	7	Entrada/Salida	-
				6	8	Entrada/Salida	-
				7	9	Entrada/Salida	-
Bus de Estado	379	279	3BD	0	NC	Lectura/Escritura	-
				1	NC	-	-
				2	NC	-	-
				3	15	Entrada	Error
				4	13	Entrada	Printer On line
				5	12	Entrada	Out of Paper
				6	10	Entrada	Acknowledge
				7	11	Entrada Complementada	Busy
Bus de Control	37A	27A	3BE	0	1	Salida Complementada	Strobe
				1	14	Salida Complementada	Line Feed
				2	16	Salida	Initialize Printer
				3	17	Salida Complementada	Select Printer
				4	NC	-	-
				5	NC	-	-
				6	NC	-	-
				7	NC	-	-

Cuando se transmite información por un cable multipar (con varios conductores aislados), cada conductor genera un campo electromagnético que influye en los otros conductores, para reducir tal efecto el cable se confecciona enroscando un conductor sobre su

retorno, de manera que los campos generados por cada conductor se anulen entre si y no afecten a los otros; a su vez estos pares trenzados se enroscan nuevamente para evitar la interferencia con otros cables. También hay otras opciones que complementan la antes mencionada a la hora de transmitir datos por un cable: para poder aumentar la longitud del cable es necesario que el mismo posea mayor inmunidad al ruido, especialmente los de baja frecuencia como los de la red eléctrica, lo que se logra con una malla metálica que envuelve a todos los conductores conectada a masa (chasis de la computadora), pero cuando la longitud es considerable y los datos se transmiten a gran velocidad el cable debe ser considerado como una línea de transmisión, con todas las propiedades que esta posee, es entonces donde se observa la necesidad de que el cable tenga un terminador para evitar los rebote de señal hacia su fuente.

Cuando se diseñó el puerto paralelo también se dividieron las masas según las señales como lo indica la siguiente tabla, aunque actualmente es muy común observarlas todas unidas, lo que no está del todo mal, puesto que los circuitos de salida contienen todos la misma masa, ya que con el nivel de integración de hoy en día todos los buffer de salida están prácticamente en un mismo integrado, y de esta manera la corriente tomará el camino con menor impedancia para su retorno.

Pin 18	Retorno de masa correspondiente a: C0 y D0
Pin 19	Retorno de masa correspondiente a: D1 y D2
Pin 20	Retorno de masa correspondiente a: D3 y D4
Pin 21	Retorno de masa correspondiente a: D5 y D6
Pin 22	Retorno de masa correspondiente a: D7 y S6
Pin 23	Retorno de masa correspondiente a: C3
Pin 24	Retorno de masa correspondiente a: S7
Pin 25	Retorno de masa correspondiente a: C2

Interrupciones

La mayoría de los puertos paralelos son capaces de detectar interrupciones generadas por el periférico que se encuentre conectado. Para que el microprocesador detecte la interrupción el puerto debe tener asignado un Interrupt Request Level (IRQ). Generalmente se utiliza IRQ7 para LPT1 e IRQ5 para LPT2, aunque este último suele estar asociado con la placa de sonido, por lo que es posible observar que los puertos tengan configurado otro IRQ.

Canales DMA

Los puertos de tipo ECP utilizan Direct Memory Access (DMA) para una transferencia a mayor velocidad, hay que remarcar que utilizando DMA, mientras se produce la transferencia el microprocesador está libre para ejecutar otra acción, el rango que utiliza el puerto paralelo es de cero a tres.

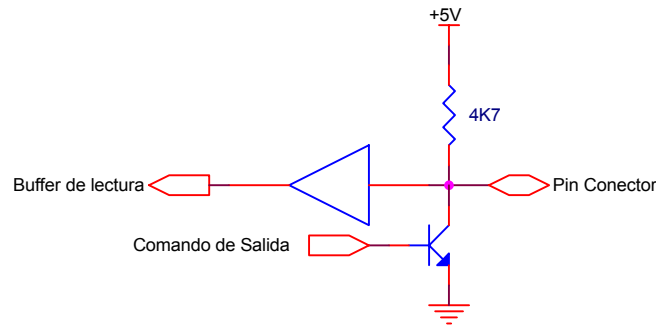
El Puerto y su Hardware

El puerto trabaja con lógica TTL de baja potencia de tipo Schottky, aunque en puertos viejos es posible encontrar algunos que trabajen con salidas de tipo colector abierto. La corriente drenada del puerto puede tomar valores de hasta 2,6mA y la entregada hasta un máximo de 24mA, aunque debido a la cantidad de fabricantes y a la falta o desconocimiento de normas es conveniente no acercarse al máximo que aquí se menciona. En cuanto a las tensiones de salida, el nivel bajo corresponde una tensión entre 0V y 0,8V, y para el nivel alto la tensión de salida es de 2,4V a 5,5V.

En algunos de los puertos se pueden encontrar también capacitores de 2,2nF entre las líneas de E/S y masa, lo que ayuda a establecer los tiempos de subida y de bajada de la señal, especialmente cuando se transmite a altas velocidades donde es muy importante la detección del flanco.

En el bus de control, generalmente, las salidas son de tipo TTL a colector abierto, con resistencias de colector de 4,7K Ω y teniendo en cuenta que lo que se lee por programa es el estado del pin del puerto, es posible utilizar este bus como entradas, aunque esto no es

recomendable, ya que como se ha mencionado antes no hay una norma firme respecto a la electrónica interna; en el caso de querer hacerlo hay que poner a cero los bits del bus de control que se vayan a utilizar como entrada, de esta manera los transistores de salida quedan al corte, entonces, al quedar el colector sin nivel de masa se puede aplicar tensión al pin sin provocar daño alguno.



En lenguaje C:

```
int inportb(int port)
    esta función entrega un valor entero correspondiente a la lectura del
    puerto almacenado en la variable entera port.
void outportb(int port, int value)
    esta función escribe el valor almacenado en la variable entera value en
    el puerto almacenado en la variable entera port.
```

Además es necesario conocer las funciones para operar con bits:

$\sim X$	NOT X
$X \& Y$	X AND Y
$X Y$	X OR Y
$X \wedge Y$	X X-OR Y
$X \gg n$	Desplazamiento hacia derecha de X , n posiciones
$X \ll n$	Desplazamiento hacia izquierda de X , n posiciones

Funciones para la configuración del bus de datos como entrada y como salida:

```
void BusDatosEntrada(int Base)
{
    outportb(Base+2,(inportb(Base+2)|160) & 191);
    return;
};

void BusDatosSalida(int Base)
{
    outportb(Base+2,(inportb(Base+2)|64) & 95);
    return;
};
```