

Instituto Tecnológico Argentino Técnico en Hardware de PC		
Plan THP2A03B	Reservados los Derechos de Propiedad Intelectual	
Tema: Configuraciones Avanzadas I: Recursos Legacy	Archivo: CAP2A03BTHP0121.doc	
Clase N°: 21	Versión: 1.2	Fecha: 27/8/03

CONFIGURACIONES AVANZADAS 1

1 OBJETIVO

La instalación de nuevos componentes en una PC se ha simplificado mucho en la última generación de computadoras personales. No podemos decir lo mismo de las generaciones anteriores, donde el técnico debía planificar cuidadosamente la disponibilidad de los recursos, para lograr que el nuevo componente instalado funcione a pleno. Pero aunque la configuración haya evolucionado y simplificado, el principio de funcionamiento no ha variado en su esencia.

En esta clase, estudiaremos los elementos esenciales de toda configuración conocidos como recursos. Comenzaremos por la metodología de configuración de los componentes pertenecientes a la generación previa de computadoras, para luego progresar en nuestro conocimiento y conocer la metodología moderna en la clase próxima.

Lo estudiado en esta clase servirá como base fundamental para entender los sistemas de última generación, ya que como hemos dicho anteriormente la esencia es la misma.

Por otro lado, no debemos caer en el error de pensar que las generaciones anteriores de computadoras se han extinguido. Muy por el contrario, muchos clientes no han actualizado sus equipos, y debido a la situación actual del mercado informático argentino, posiblemente no deseen actualizarlos de inmediato. Pero sí desean mantenerlos en funcionamiento, y posiblemente modificar algunas configuraciones para incorporarles nuevos componentes.

Por ello, es indispensable que un técnico competente conozca a fondo esta tecnología, ya sea para mantener o ampliar viejos equipos, o para tener una base técnica sólida aplicable a las tecnologías más modernas.

2 LOS RECURSOS

El destino de una PC es amplio. Más aún si consideramos que cambiando o agregando algunos periféricos podemos darle una nueva aplicación. Basta por ejemplo agregar una interfaz sintonizadora y una antena, para poder ver TV con la computadora.

Integrar nuevos componentes, es como agregarle nuevos “brazos” a nuestro equipo para brindarle nuevas funcionalidades. Estos nuevos miembros incorporarán funciones nativas, y deberán estar comandados por el cerebro de la PC (la CPU o micro procesador). Es posible además que se interrelacionen con otros componentes periféricos, como por ejemplo la memoria, tanto para la toma o entrega de datos.

Muchos fabricantes independientes en el mundo, brindan soluciones de hardware para los usuarios de computadoras PC, pero no fabrican las computadoras, sólo periféricos. Estos periféricos deben ser lo suficientemente genéricos, como para poder ser instalados en una amplia gama de marcas y modelos de PCs, posibilitando al fabricante acceder a mayor cantidad posible de clientes.

En resumen, por un lado tenemos componentes periféricos que deben tener acceso al hardware de la PC para lograr su funcionalidad; y por otro lado tenemos computadoras que pueden ser muy diferentes entre sí. La pregunta es ¿cómo se pueden relacionar íntimamente componentes de diferentes orígenes sin caer rápidamente en incompatibilidades severas?

La respuesta está en el establecimiento de normas y métodos estandarizados, para el acceso a los componentes del sistema.

Básicamente estos métodos de acceso nos permiten:

- Que la CPU pueda entregar o recibir información del periférico.
- Que el periférico pueda introducir o tomar información de la memoria del sistema.
- Que el periférico le pueda avisar a la CPU que dispone de nuevos datos; que ha completado una tarea, o que necesita un servicio.

Cada uno de estos métodos debe establecer múltiples vías de comunicación, para permitir la concurrencia de varios periféricos, sin interferirse mutuamente. ***El conjunto de vías y métodos de acceso, conforman los recursos del sistema.***

Para que la CPU pueda entregar o recibir información de los periféricos, las computadoras disponen de puertas de entrada y salida de datos. Se han definido **1024** direcciones de acceso en los primeros diseños; en los actuales se disponen de **65.536**, conocidas mejor como ***puertas de E/S*** (en inglés, *Input Output ports* o ***I/O ports***).

Para que el periférico pueda introducir o tomar información de la memoria del sistema, existen los canales de acceso directo a memoria (en inglés, *Direct Memory Access* o ***DMA***)

El mecanismo de aviso que usan los periféricos con la CPU se lo conoce como petición de interrupción (en inglés, *Interrupt ReQuest* o ***IRQ***).

2.1 LAS PUERTAS DE ENTRADA Y SALIDA

Repasemos brevemente cómo funciona una computadora. La CPU posee básicamente tres buses: ***direcciones, datos y control.***

El ***bus de datos*** llega por igual a todos los dispositivos del sistema, es bidireccional y su función es la de transportar datos, desde o hacia la CPU respectivamente.

El ***bus de direcciones*** es controlado por la CPU, y permite activar un dispositivo en particular entre todos los existentes en el sistema. Para ello, la CPU coloca en el bus una combinación única de unos y ceros (una dirección). Un dispositivo (*y sólo uno*) debe activarse con esta combinación. A partir de allí, la comunicación está establecida entre ese dispositivo y la CPU.

El ***bus de control*** es un canal especial, por el cual la CPU indica cuál es la operación que desea realizar, como por ejemplo leer o escribir un dato sobre el dispositivo seleccionado. Esto es sólo un ejemplo, ya que las funciones del bus de control son varias, y en esta clase justamente veremos otra de estas funciones: ***diferenciar entre una operación con la memoria o con un periférico.***

El bus de direcciones de una CPU Pentium III por ejemplo, tiene **32 bits**. Con esa cantidad de bits, se pueden generar 4.294.967.296 combinaciones de unos y ceros (equivalente a **2³²**). Cada una de estas combinaciones, puede utilizarse para activar un componente, como por ejemplo una celda de una memoria. Generalmente éste es el destino de las **2³²** combinaciones posibles de este bus, y por eso a este espacio se lo conoce como ***direcciones de memoria.***

El bus de datos también llega a los demás periféricos. Sin embargo el espacio de direcciones reservado para ellos, es distinto al de la memoria. Para comprender mejor esto, usaremos el esquema de la *figura 21.1*.

En dicha figura observamos un esquema simplificado de los buses mencionados, donde hemos detallado la relación de los mismos con la memoria y los periféricos.

El **bus de datos** llega tanto a la memoria como a los periféricos. Esto permite que la CPU se pueda comunicar indistintamente con cualquiera de ellos.

Vemos además que los *32 bits* del **bus de direcciones** llegan a la memoria, pero sólo *10 o 16 Bits* (esto depende del diseño de la PC) llegan a los periféricos.

El **bus de control**, llega a un decodificador que detecta combinaciones especiales de unos y ceros generados por la CPU. Por este bus, la CPU puede indicar entonces, si la combinación presente en los *32 bits* de direcciones, es una *dirección válida de memoria* o se trata de una combinación únicamente válida para un periférico. En este último caso, la validez se establece para los primeros *10 o 16 Bits* del bus de direcciones, y *el resto de los bits es ignorado*.

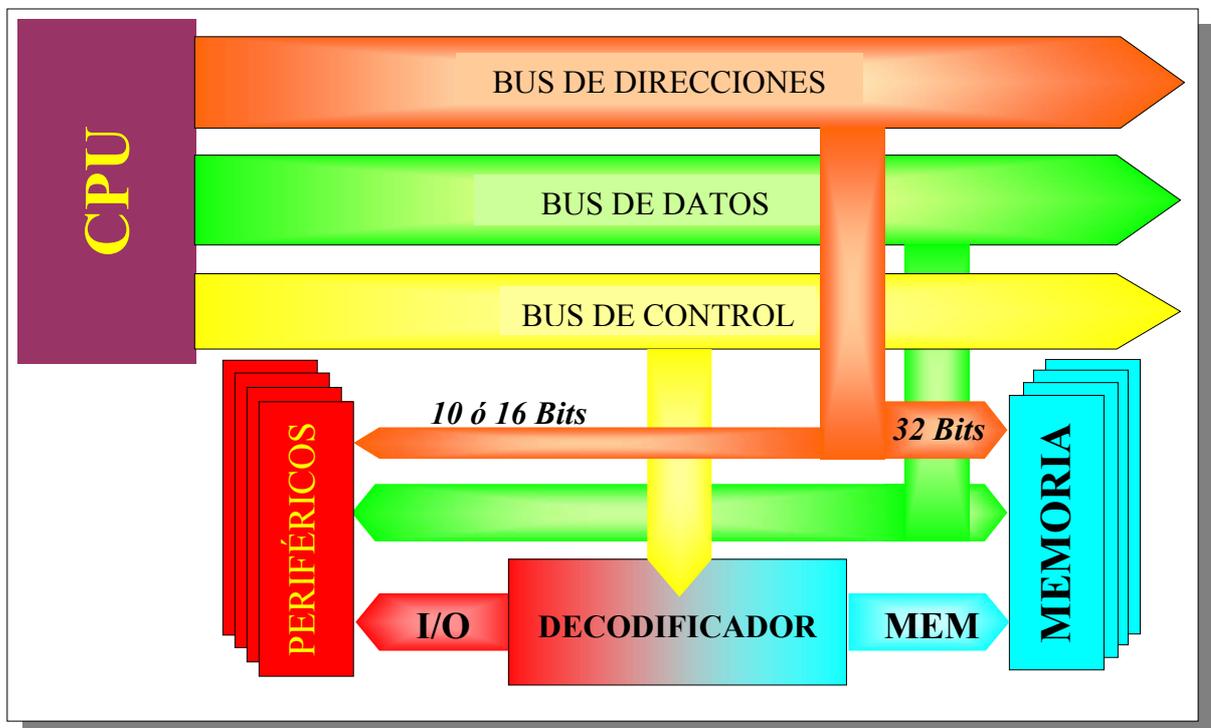


Figura 21.1

Con *10 Bits* de direcciones se pueden generar 1024 combinaciones (2^{10}) y con *16 Bits*, se pueden generar 65.536 combinaciones posibles (2^{16}).

En resumen, es el **bus de control** quien determina si la dirección presente en el bus de direcciones debe aplicarse a la memoria o a un periférico. Esto significa que por ejemplo la dirección 378 presente en el bus de direcciones, puede estar activando la celda 378 de la memoria o al periférico 378, dependiendo de la combinación presente en el bus de control.

Si la dirección se aplica a un periférico, se la denomina **puerto de entrada y salida (I/O Port)**, sino, se la denomina **dirección de memoria**.

2.2 CANALES DE ACCESO DIRECTO A MEMORIA

Normalmente las transferencias de datos desde los periféricos hacia la memoria y viceversa, los realiza la CPU. Esta transferencia si bien es eficiente, mantiene ocupada a la CPU, quitándole tiempo de procesamiento para otras tareas.

Desde los primeros diseños de PC, se ha incluido un componente adicional, capaz de controlar y arbitrar transferencias: *el controlador de acceso directo a memoria (DMA)*.

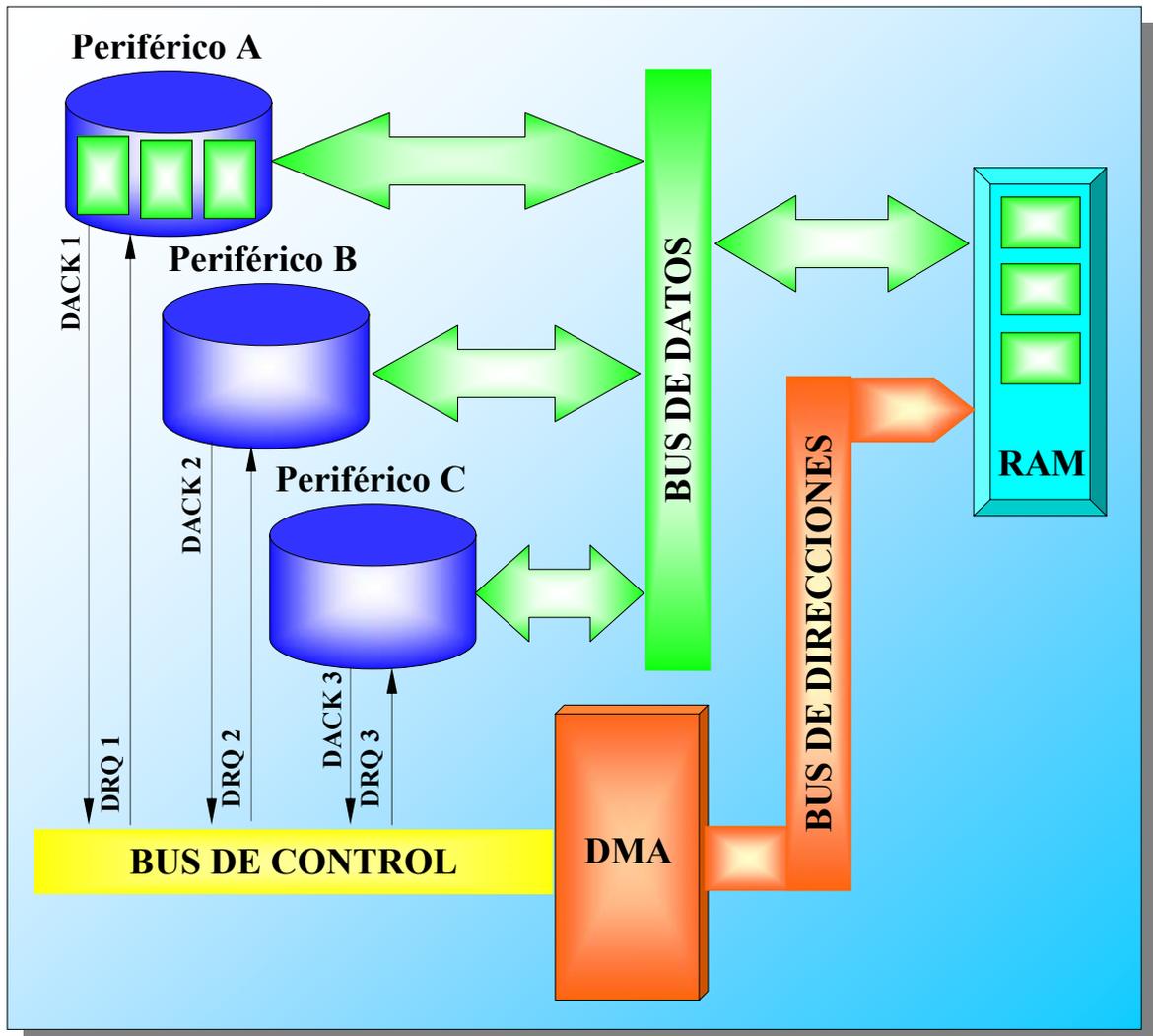


Figura 21.2

En la *figura 21.2*, podemos observar la relación existente entre los buses, periféricos, memoria RAM y el controlador de accesos directos a memoria (*DMA*). Nótese que en la figura no está presente la CPU, ya que esta última no interviene en la transferencia en sí; sólo interviene en la configuración de la transferencia.

Cuando una transferencia debe llevarse a cabo, el chip DMA le solicita a la CPU el control del bus. La CPU cuando completa la tarea en curso, le avisa al chip DMA que puede hacerse cargo del bus e iniciar la transferencia. Supongamos que el periférico "A" es quien debe realizar la transferencia. Entonces el chip DMA coloca en el bus de direcciones, la dirección inicial de la transferencia, activando la celda correspondiente en la memoria RAM.

Luego le envía una señal **DRQ 1** al periférico por el bus de control. El periférico al recibir la señal, coloca el dato en el bus de datos que será almacenado directamente en la celda activa de la memoria. Luego el periférico le envía la señal **DACK 1** al chip DMA por el bus de control, quien inmediatamente activa la próxima dirección de memoria, repitiéndose el procedimiento hasta completar la transferencia. Finalmente toma nuevamente el control del sistema la CPU.

Cada periférico que debe realizar transferencias por DMA, debe tener **con exclusividad**, un par de líneas **DRQ** (*Data ReQuest, solicitud de datos*) y **DACK** (*Data ACKnowledge, dato reconocido*). Estas líneas las emplea para dialogar con el chip DMA, como se ha indicado anteriormente. **A cada par de señales DRQ y DACK se la conoce como canal DMA.** En la *figura 21.2* por ejemplo, podemos ver la conexión de **tres canales**.

Los primeros diseños de PC, contaban con un chip DMA que podía manejar hasta cuatro canales. Con el diseño del la PC AT, se incluyó un segundo chip DMA conectado en cascada con el primero, agregando de este modo tres canales adicionales y haciendo entonces un total de siete.

2.3 LA PETICIÓN DE INTERRUPCIÓN (IRQ)

Los periféricos disponen de un mecanismo para alertar a la CPU que necesitan servicio. Por ejemplo, cuando movemos el mouse, la CPU debe enterarse del suceso, para recoger del mouse la información del desplazamiento (dirección y velocidad), y actualizar la posición del puntero en la pantalla.

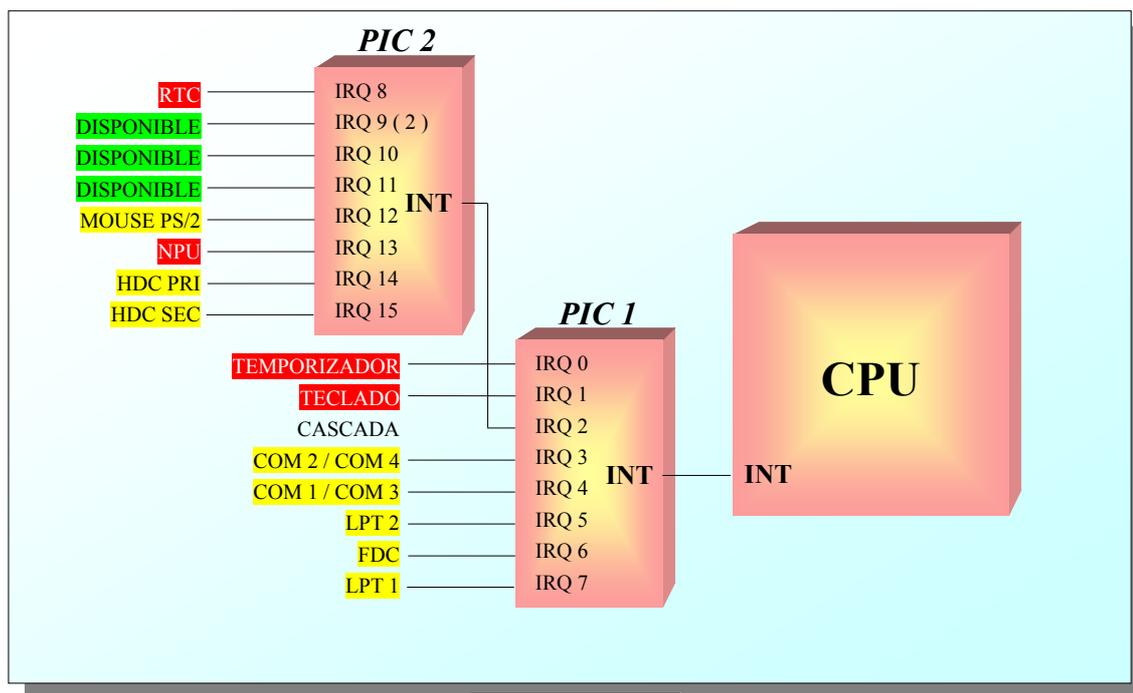


Figura 21.3

Como en una PC, puede haber varios periféricos conectados, y varios de ellos podrían requerir la atención de la CPU simultáneamente, se ha establecido un mecanismo de niveles de prioridades, pues obviamente la CPU no los puede atender a todos al mismo tiempo.

En la PC XT existían sólo ocho niveles de interrupción, controlados por un chip conocido como **PIC** (*Programmable Interrupt Controller, Controlador de Interrupciones Programable*).

Cuando IBM diseñó el modelo siguiente (PC AT), incluyó un segundo PIC en cascada con el primero como se muestra en la *figura 21.3*, ampliando el sistema a 15 niveles en total. Se identifican numéricamente desde el **IRQ 0** al **IRQ 15**.

Los niveles agregados (del 8 al 15) son controlados por el segundo PIC, cuya salida ingresa al primer PIC por el nivel 2. El viejo nivel 2 fue reasignado al nivel 9 del segundo PIC.

Las prioridades van desde la del IRQ0 (la más alta), al IRQ7 (la más baja). La prioridad de los niveles entre el 8 y el 15, quedan comprendidos entre el IRQ1 y el IRQ3, ya que el aviso a la CPU pasa por la cascada de controladores, asignada al viejo nivel 2 en el primer PIC (ver *figura 21.3*) como se mencionó anteriormente.

Este sistema de interrupciones se utiliza en los dispositivos ISA, y ***esta tecnología no permite que dos dispositivos estén conectados al mismo nivel de IRQ.***

Algunos periféricos que utilizan interrupciones forman parte del motherboard, son indispensables para la operación de la PC, y por ello desde fábrica se les ha asignado un nivel de IRQ, no estando disponible para ningún otro periférico que instalemos. Es el caso de los niveles:

IRQ 0: Asignado a un cronómetro interno del motherboard

IRQ 1: Asignado al controlador del teclado

IRQ 2: No disponible por ser la conexión cascada con el segundo PIC

IRQ 6: Asignado al controlador de la disquetera (**FDC - Floppy Disk Controller**, controlador de discos flexibles)

IRQ 8: Asignado al reloj/calendario del motherboard (**RTC - Real Time Clock**, reloj de tiempo real)

IRQ 13: Asignado al co-procesador matemático, que desde el 80486 viene incorporado dentro de la CPU, también conocido como **NPU (Numeric Processor Unit**, unidad de procesamiento numérico.

IRQ 14: Asignado al controlador de discos rígidos IDE primario.

Los niveles 6, 12 y 14 generalmente no están disponibles, pues lo ocupan el controlador de disquetera, el mouse PS/2 y el controlador de discos rígidos IDE primario, respectivamente.

Los condicionalmente disponibles serían los IRQ 3, 4, 5, 7 y 15 si los dispositivos que normalmente usan esos niveles no están instalados; y los que generalmente están disponibles, son IRQ 9, 10, 11

3 MECANISMO DE CONFIGURACIÓN

Hasta ahora hemos considerado que los recursos los usan los periféricos, y esto es en realidad una simplificación. Si avanzamos un poco más en detalles, los periféricos generalmente se relacionan con la CPU a través de una interfaz. En el caso de los periféricos internos del motherboard, la interfaz está implementada también en el mismo motherboard.

Es entonces sobre la interfaz donde se asignan los recursos. En el caso del mouse PS/2, por ejemplo, la interfaz está incluida en el motherboard. Por lo tanto la posible configuración de los recursos se debe buscar allí.

Salvo para aquellos dispositivos que tienen una asignación fija por diseño, como por ejemplo el RTC o el Temporizador, los fabricantes deben brindar flexibilidad en las posibilidades de configuración. Recordemos que no es posible compartir un canal de **DMA**, o un **IRQ ISA**, o un **I/O Port**, y que la cantidad de recursos disponibles es relativamente baja. Teniendo alternati-

vas en la configuración de los recursos, hay mayores posibilidades de instalar exitosamente los componentes en la PC.

Pensemos que si el fabricante no diera ninguna alternativa en la configuración, no sería posible instalar algunas combinaciones de dispositivos. Por ejemplo, supongamos que los fabricantes de placas de sonido le asignaran en forma fija el IRQ 10 a esos dispositivos, y que los fabricantes de módems también lo hicieran al mismo recurso. Podrían instalarse por separado, pero no podrían estar ambos simultáneamente en una PC. Por otro lado, no existen infinitos recursos como para brindarle una posibilidad fija y distinta a cada fabricante.

Habiendo alternativas, sería posible elegir en cada dispositivo una configuración que no entre en conflicto con otro componente del sistema. Bueno, en realidad de eso se trata nuestra labor: **Localizar una configuración válida que no entre en conflicto con los demás componentes instalados.**

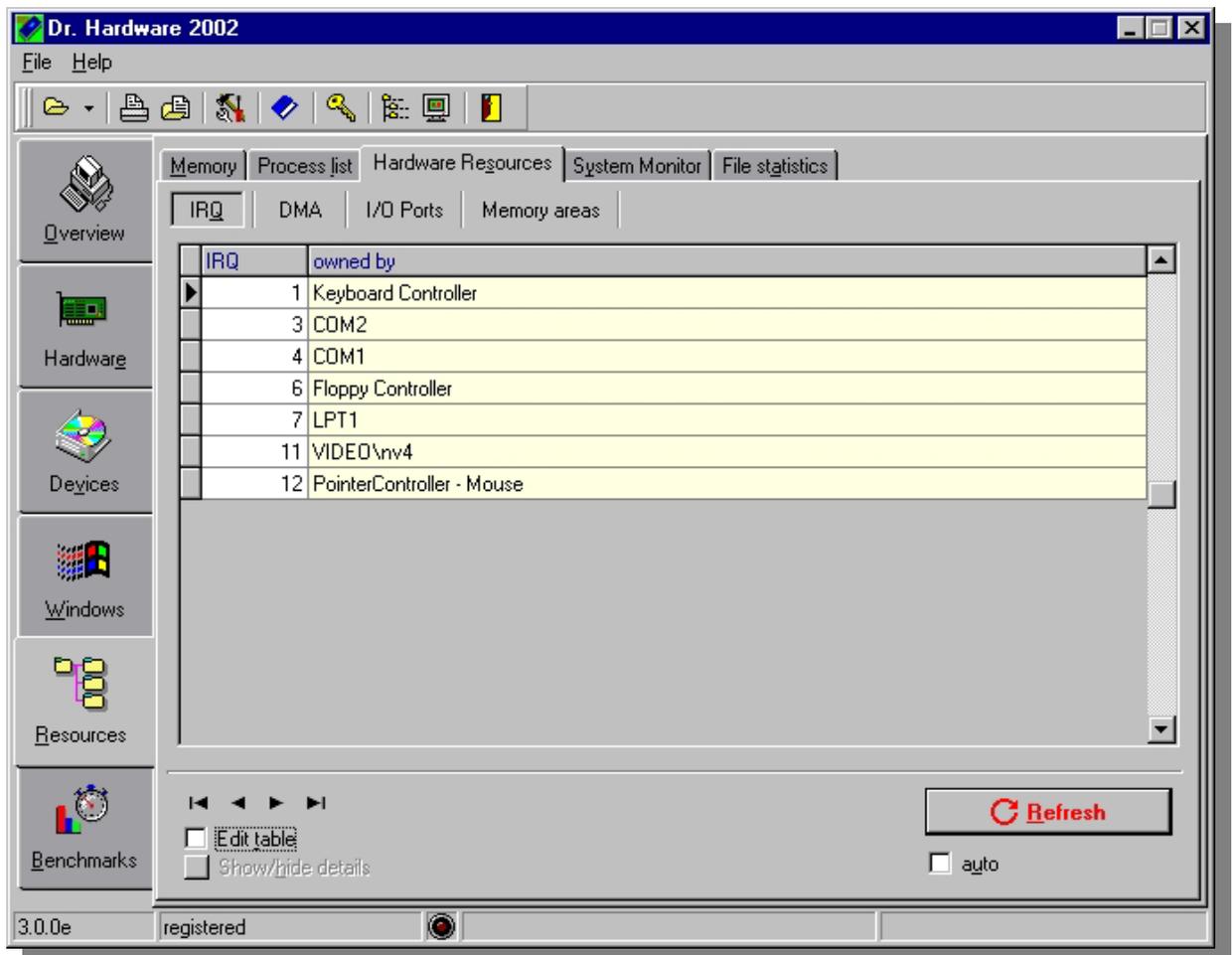


Figura 21.4

Existen programas utilitarios informativos, como el que observamos en la *figura 21.4*, que nos permite investigar rápidamente cuáles son los que están actualmente en uso en la PC, y poder asignar algún recurso vacante al componente que se desea instalar. El software se puede obtener de la siguiente URL: <http://www.dr-hardware.com>.

En dicha figura, se puede observar que por ejemplo, el IRQ5 o el IRQ10 están libres.

Una vez localizado el o los recursos vacantes que necesitamos, procederemos con la configuración de la interfaz del dispositivo.

La pregunta inmediata que surge es: ¿Cómo se configura la interfaz? Pues bien, tratándose de dispositivos ISA, como por ejemplo un módem, generalmente la configuración pasa por mover algunos jumpers y/o DIP Switch. Por ejemplo en la *figura 20.5*, vemos un módem ISA donde está señalado con flechas la ubicación de los jumpers y DIP Switch.

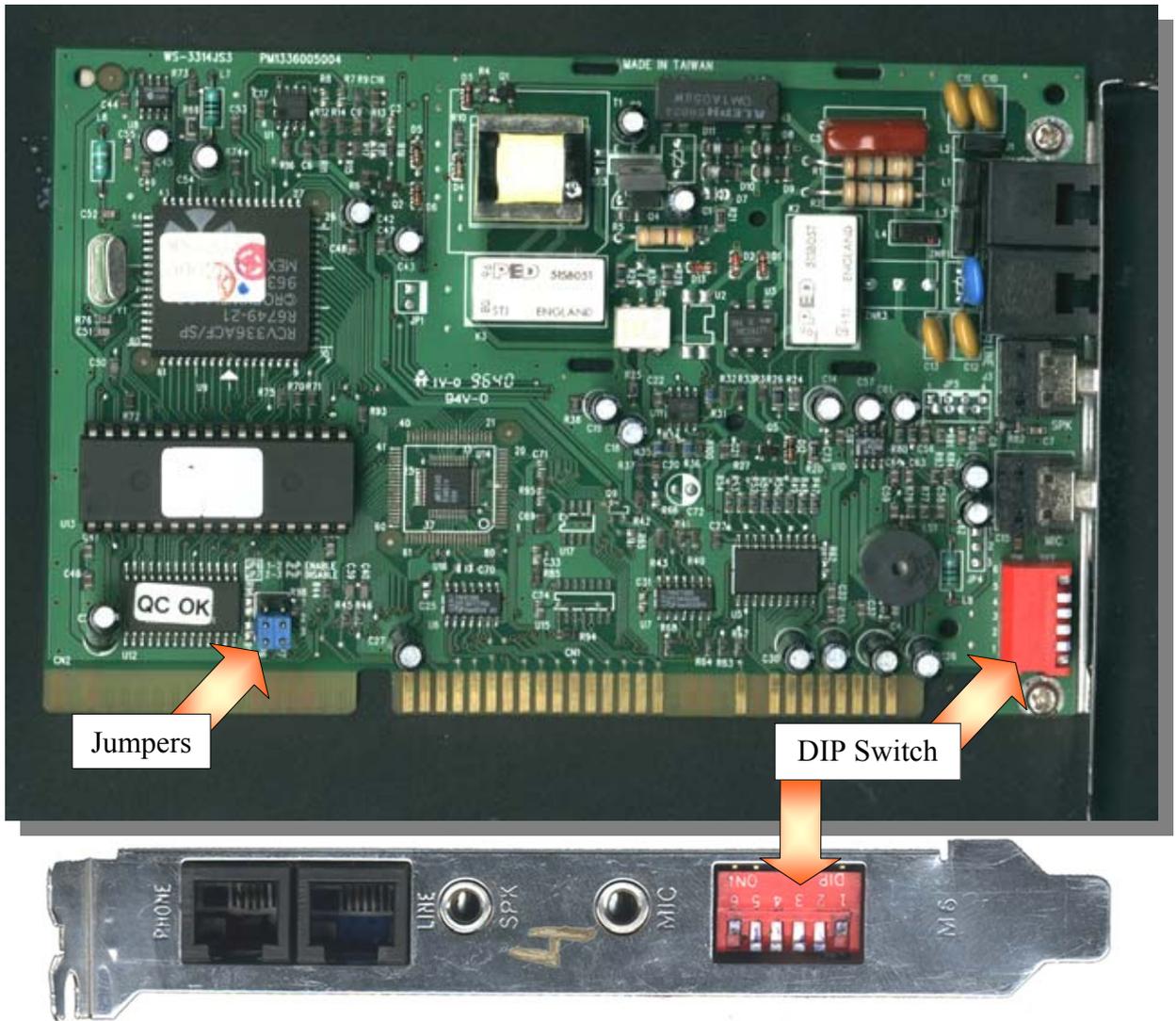


Figura 21.5

La información necesaria para saber en qué posición debe colocarse los jumpers y DIP Switches, debe obtenerse del manual del fabricante.

En el manual del módem de la figura, encontramos una tabla que nos indica por ejemplo cómo debemos colocar los jumpers y el DIP Switch para que utilice el IRQ 5, como se puede apreciar en la *Tabla 21.1* (la línea correspondiente se encuentra resaltada).

En la columna de la izquierda, se observa la dirección de I/O (**I/O port**) asignado: **3E8**. Es normal que los manuales nos indiquen las direcciones en hexadecimal. Sin embargo esto no es algo que deba preocuparnos, ya que esté expresado en la base que sea, no debe haber otro dispositivo en el sistema que use la misma dirección.

Dirección I/O Puerto COM	IRQ	DIP 1	DIP 2	DIP 3	DIP 4	DIP 5	DIP 6	JP6	JP7
03F8	4	ON	ON	OFF	OFF	ON	OFF	2-3	2-3
02F8	3	OFF	ON	OFF	ON	OFF	OFF		
03E8	5	ON	OFF	OFF	OFF	OFF	ON		
	4	ON	OFF	OFF	OFF	ON	OFF		
02E8	2	OFF	OFF	ON	OFF	OFF	OFF		
	3	OFF	OFF	OFF	ON	OFF	OFF		

Tabla 21.1

PUERTOS SERIALES		
Identificación	I/O port (hexadecimal)	IRQ
Com 1	03F8	4
Com 2	02F8	3
Com 3	03E8	4
Com 4	02E8	3

Tabla 21.2

PUERTOS PARALELOS		
Identificación	I/O port (hexadecimal)	IRQ
LPT 1	0378	7
LPT 2	0278	5
LPT 3	03BC	7

Tabla 21.3

En la *tabla 21.2*, observamos las asignaciones tradicionales de direcciones de puertos de entrada y salida para las interfaces de comunicación 1 a 4 (COM1 a COM4) y en la *tabla 21.3*, observamos las configuraciones tradicionales de los puertos paralelos. *En nuestro caso, al asignar la dirección 3E8 al módem, le hemos asignado el puerto de comunicación COM3.*

Si consultamos nuevamente con nuestro software informativo (*figura 21.6*), vemos que esta asignación es feliz, ya que no hará conflictos con otros puertos de comunicación instalados en esa máquina. En la misma, ya están instalados los puertos COM1 (3F8 - IRQ4) y COM2 (2F8 - IRQ3).

Si estudiamos detenidamente la *tabla 21.1*, vemos que la alternativa que hemos elegido es una que no hace conflicto con otros dispositivos instalados en la PC.

Por ejemplo *la primera línea* establece la configuración hacia el puerto I/O **3F8** y le asignaría el **IRQ4**. En ambas asignaciones tendríamos conflicto con la interfaz serie COM1 ya instalada en la máquina (ver *figura 21.6*).

La segunda alternativa que nos brinda el fabricante del módem, hace conflicto con la interfaz serie COM2 instalada en la PC.

La cuarta alternativa no hace conflicto con ningún puerto I/O, pero hace conflicto con el IRQ4, que según la *figura 21.5* lo usa al COM1 instalado en la PC.

La quinta alternativa asigna el puerto I/O **2E8**, y el **IRQ 2**, **siendo esta la segunda configuración posible**, ya que ambos recursos están disponibles. Recordemos que el viejo IRQ2 fue reasignado como IRQ9 desde el modelo AT. Es decir que la configuración así como se indica, estará usando en realidad el IRQ9 que ingresa por el PIC2. Esto no es ningún inconveniente, pero hay que tenerlo en cuenta, porque una vez instalado el dispositivo físicamente, nuestro software de información nos dirá que está usando el IRQ9, no en el IRQ2; y si no tenemos claro la causa, puede confundirnos.

La sexta alternativa de la *tabla 21.1*, nos muestra que tendríamos conflicto con el IRQ3, que está en uso por la interfaz de comunicación serie COM2 según muestra la *figura 21.6*.

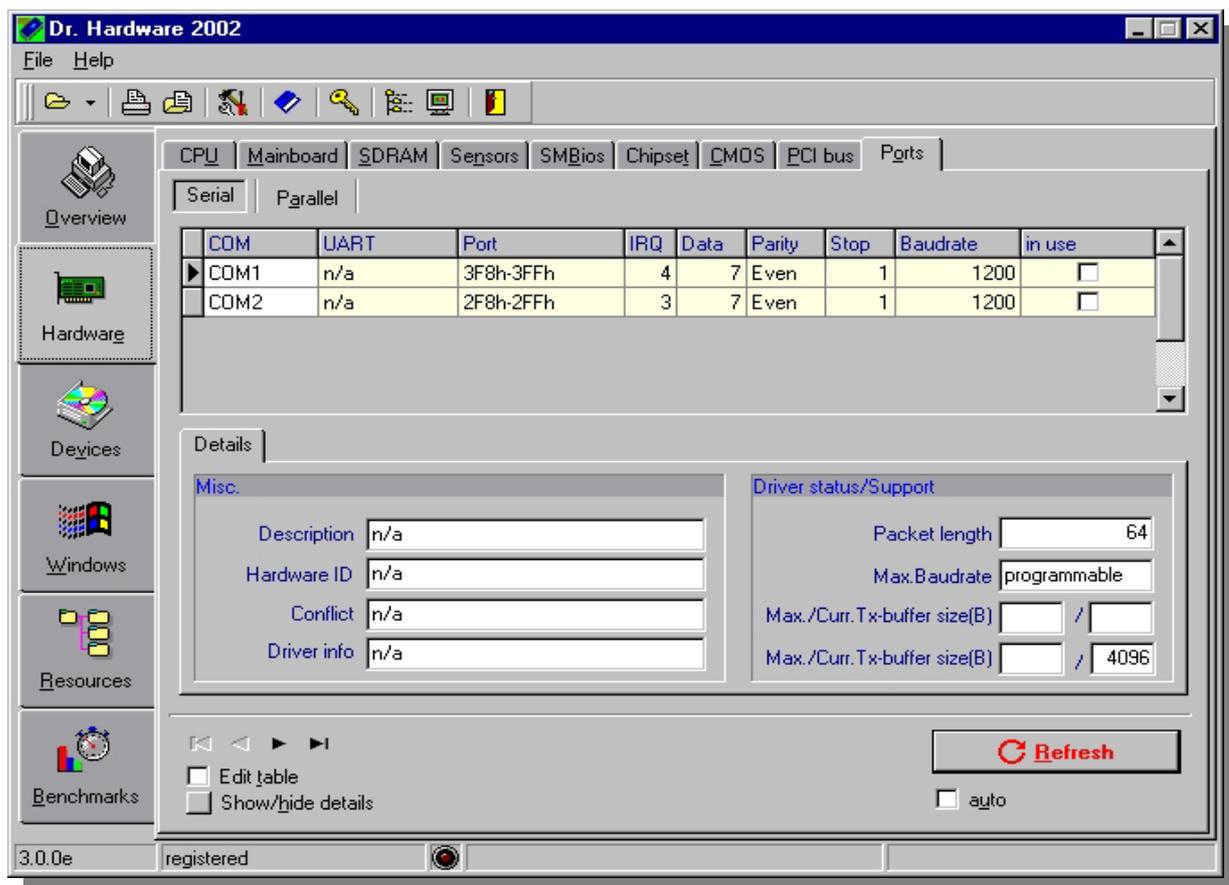


Figura 21.6

CUESTIONARIO CAPITULO 21

1.- *¿Por qué algunos sistemas pueden usar 65.536 I/O ports y otros no?*

2.- *¿Puede existir memoria RAM en la dirección 378 si está presente el puerto paralelo LPT1? ¿Por qué?*

3.- *La interfaz de un escáner, según el fabricante, requiere el IRQ12. ¿Qué debería tener en cuenta antes de incorporarla?*

4.- *Al incorporar un módem ISA en el COM3 dejó de funcionar el mouse. ¿Cuál es la causa? ¿Cómo procedería para solucionar el problema?*

5.- *¿Quién es el que gobierna el bus de direcciones, durante una transferencia por DMA?*

6.- *Suponiendo que se está llevando a cabo una transferencia DMA desde un periférico hacia la memoria, ¿quién es el que escribe el dato sobre la memoria, el controlador del periférico o el chip DMA?*
