

UN PASEO POR

USB-1

**Soporte Técnico OEM
FUJITSU ESPAÑA**

Marzo de 2000

INDICE

REFERENCIAS

INTRODUCCION

NIVEL FISICO

- Hubs
- Conexión y Desconexión en Caliente
- Protocolo

TIPOS DE TRANSFERENCIAS DE DATOS

- Transferencias de Control
- Transferencias Isócronas
- Transferencias de Interrupción
- Transferencias Bulk

MODELO LOGICO

CLASES USB

USB 2.0

REFERENCIAS

- Especificación de Bus Serie Universal Revisión 1.1. Compaq, Intel, Microsoft y NEC. 23 de Septiembre de 1998.

INTRODUCCION

USB es una especificación de las empresas Compaq, Intel, Microsoft y NEC, que describe un canal serie que soporta una gran variedad de periféricos de media y baja velocidad, con soporte integral para transferencias en tiempo real (isócronas) como voz, audio y vídeo comprimido, y que permite mezclar dispositivos y aplicaciones isócronas y asíncronas. Por lo tanto, entre los dispositivos USB más característicos se pueden citar teclados, ratones, joysticks, tabletas gráficas, monitores, modems, impresoras, escáneres, CD-ROMs, dispositivos de audio (como micrófonos o altavoces digitales), cámaras digitales y otros dispositivos multimedia.

La versión 1.1 (Sep/98) establece:

- Un acceso al bus gestionado directamente por el Controlador USB, para permitir transferencias isócronas y eliminar los tiempos de arbitración.
- Una velocidad de 12 Mbps (Full Speed o FS) y un subcanal de 1,5 Mbps (Low Speed o LS) para los dispositivos más lentos, como ratones y joysticks. La coexistencia en un mismo sistema de dispositivos FS y LS se maneja mediante conmutación automática y dinámica de velocidad entre unas transferencias y otras.
- Una conectividad excepcional, ya que puede manejar hasta 127 dispositivos simultáneamente que se pueden conectar y desconectar en caliente, sin tener que reiniciar el sistema.
- Una configuración automática de dispositivos, que elimina la necesidad de realizar configuraciones manuales por medio de puentes o conmutadores.
- La coexistencia de dispositivos isócronos y asíncronos. Los dispositivos isócronos se atienden en función del ancho de banda y latencia requeridos, y los asíncronos se atienden durante el tiempo restante no consumido por los dispositivos isócronos.
- Una distribución de alimentación desde el Controlador USB, que permite la conexión tanto de dispositivos alimentados desde el bus como autoalimentados.
- Una arquitectura fácilmente escalable para permitir la existencia de varios Controladores USB en un sistema.

USB se soporta en Win95 2.1 (4.00.950B) y superior, Win98, Win2000 y MacOS 8.5 y superior.

NIVEL FISICO

A nivel físico, USB utiliza un cable de 4 conductores para transmitir una señal diferencial (D+ y D-) y alimentación (VBus = 5V y GND) por medio de conexiones punto a punto. Los dispositivos LS van obligatoriamente equipados con un cable de longitud adecuada (hasta unos 3 m, dependiendo de sus características eléctricas), mientras que los FS pueden ir equipados con un cable o utilizar cables independientes de hasta 5 m (también dependiendo de sus características eléctricas).

La comunicación es bidireccional y semi-dúplex, y utiliza codificación autoreloj NRZI (la línea cambia de nivel si se transmite un 0 y no cambia si transmite un 1) con "bit stuffing" (inserción de un cero tras la transmisión de 6 unos, para asegurar transiciones en la línea y permitir que la PLL del receptor se mantenga sincronizada). Los dispositivos disponen de un transmisor diferencial, receptores diferencial y S/E y resistencias de terminación con los que pueden transmitir y detectar varios estados eléctricos distintos en la línea:

- Transmisión/Recepción diferencial de bits: Estados DIFF0 y DIFF1, denominados también estados J y K.
- SE0 (Single-Ended 0): Ambas señales D+ y D- a 0V. Se utiliza para detectar la conexión/desconexión de dispositivos, para indicar el EOP (fin de paquete) y para generar reset.
- IDLE: reposo o línea en alta impedancia, necesario para permitir transferencias semi-dúplex, detectar la conexión y desconexión de dispositivos y discriminar entre dispositivos FS y LS.
 - El SOP (principio de paquete) se indica mediante una transición IDLE a K.
 - El EOP (fin de paquete) se indica mediante una secuencia SE0 (2 bits) + J (1 bit) + IDLE.

- Detección de dispositivo y discriminación FS/LS: cuando el transmisor deja la línea en IDLE, si hay un dispositivo conectado su polarización fuerza un estado J (DIFF0 si LS ó DIFF1 si FS), y si no lo hay, la polarización del transmisor fuerza un estado SE0.
- Reset: transmisión de SE0 durante ≥ 10 ms.

Hubs

Dentro de la arquitectura USB, unos elementos esenciales y especiales son los Hubs (concentradores), que proveen conectividad (los dispositivos no se conectan entre sí directamente, sino cada uno a un hub), detectan la conexión y desconexión de dispositivos y si son FS o LS, generan alimentación hacia los dispositivos e incorporan la terminación de las líneas. Los hubs disponen de una conexión "upstream" hacia el ordenador y una o varias conexiones "downstream" hacia dispositivos u otros hubs (concentrando efectivamente varios puntos de conexión en uno sólo), de forma que se pueden encadenar varios Hubs para formar una topología en varios niveles. Como a los Hub se conectan los dispositivos en estrella, la topología USB se denomina Estrella en Niveles. USB permite hasta 6 niveles, y en el nivel 0 (Raíz o Root) se encuentra el Controlador USB, que controla todo el tráfico de información en el bus. Los Hubs podrán ir integrados en algunos dispositivos (del estilo de teclados, impresoras y monitores), y también estarán disponibles como elementos independientes. Normalmente los hubs serán autoalimentados, aunque bajo ciertas restricciones topológicas podrían utilizarse hubs alimentados desde el bus. El hub dispone de un Repetidor, para pasar información entre el puerto Upstream y los puertos Downstreams, y de un Controlador, que incorpora un juego de registros a través de los cuales el Controlador USB configura el hub y controla y monitoriza los puertos Downstream.

Conexión y Desconexión en Caliente

USB permite conectar un nuevo dispositivo sin tener que reiniciar el sistema. Cuando un Hub detecta una nueva conexión se lo comunica al Controlador USB, el sistema interroga al nuevo dispositivo, determina sus propiedades y posibilidades y lo configura. Adicionalmente el sistema carga el manejador adecuado al tipo de dispositivo con lo que el usuario puede empezar a trabajar con el mismo inmediatamente. Algunos sistemas operativos llaman a este proceso "Bus Enumeration".

Protocolo

El protocolo de nivel físico se basa en tokens (testigos). El controlador USB transmite tokens que incluyen la dirección del dispositivo destino, y el dispositivo que detecta su dirección en el token responde y lleva a cabo la transferencia de datos con el controlador. De esta manera, el Controlador USB maneja la parte más compleja del protocolo, generando los tokens de transferencias de datos a 12 Mbps o a 1,5 Mbps, y controlando la conexión lógica entre el sistema y las funciones internas de cada dispositivo. El controlador USB también maneja el consumo en el bus a través de las funciones Suspend/Continuar, por medio de las cuales controla los modos Reposo/Activo de los dispositivos. Esta arquitectura permite el diseño de dispositivos extremadamente simples y de bajo coste.

USB divide el tiempo en espacios de 1 ms denominados Tramas, durante las cuales se llevan a cabo las comunicaciones a través de Transacciones, las cuales se componen a su vez de Paquetes. Las Transacciones se componen de 3 fases: Token, Dato y Validación (Handshake):

- La fase de Token se compone de un paquete de token enviado por el Controlador USB, y siempre está presente en toda transacción. El paquete contiene los campos:
 - PID (identifica el tipo de paquete). Todos los PIDs van protegidos por bits redundantes,
 - Dirección del elemento destino (7 bits de dispositivo + 4 bits de elemento interno al dispositivo), y
 - CRC5.
- La fase de Datos (opcional) se compone de los paquetes de datos que se transfieren entre el Controlador USB y el dispositivo. Cada paquete se compone de los campos PID, Datos, y CRC16.
- La fase de Validación (opcional) se usa para indicar el resultado de la Transacción. Se compone sólo de un campo PID.

Adicionalmente, el Controlador USB indica el principio de cada Trama y la transmisión hacia dispositivos LS mediante tokens especiales.

TIPOS DE TRANSFERENCIAS DE DATOS

USB soporta 4 tipos de transferencias de datos:

- Control, para configuración y control de dispositivos y para manejo del bus.
- Isócrono, para transmisión de información con ancho de banda y latencia garantizados, necesario para aplicaciones como audio, telefonía y vídeo. Permite una comunicación periódica y continua entre el sistema y el dispositivo.
- Interrupción, para transferencias de pocos datos, no periódicas, de baja frecuencia pero con unos ciertos límites de latencia.
- Bulk, para transferencias de grandes cantidades de datos con dispositivos asíncronos, como impresoras, escáneres, cámaras de fotos (foto fija), etc.

Transferencias de Control

- Se desarrollan en 3 Transacciones:
 - Transacción de Configuración (Setup), en la que se envía al dispositivo un paquete que especifica la operación a ejecutar. Ocupa 8 bytes.
 - Cero o más Transacciones de Datos, en las que se transfieren los paquetes de datos en el sentido indicado por la Transacción de Configuración. La información útil por paquete puede ser de 8, 16, 32 ó 64 bytes para Endpoints FS, y de 8 bytes para Endpoints LS.
 - Transacción de Estado, en la que el receptor informa del estado final de la operación.
- Se procesan por medio de un mecanismo "best effort", según el cual el Controlador USB las va procesando en función del tiempo disponible en cada Trama. Como mínimo se reserva el 10% del tiempo de Trama, y se puede utilizar tiempo adicional siempre que las necesidades de los tráficos isócrono y de interrupción lo permitan.
- Incorporan mecanismos de detección de errores (CRC) y de recuperación/retransmisión de datos.

Transferencias Isócronas

- Sólo son utilizables por dispositivos FS.
- La información útil por paquete puede oscilar entre 1 y 1,023 bytes.
- En cada Trama se transfiere un paquete por cada conexión isócrona establecida.
- El sistema puede asignar como máximo el 90% del tiempo de Trama para transferencias isócronas y de interrupción. Si el sistema ya tiene asignado un tiempo de Trama de forma que no garantiza tiempo suficiente como para manejar una nueva conexión isócrona (transmitir un nuevo paquete por Trama), simplemente no se establece la conexión.
- Los posibles errores no se recuperan (la información que no llega a su tiempo, se descarta).

Transferencias de Interrupción

- Aseguran una transacción (paquete) dentro de un periodo máximo (los dispositivos FS pueden solicitar entre 1 y 255 ms, y los LS entre 10 y 255 ms de periodo máximo de servicio).
- Incorpora detección de errores y retransmisión de datos.
- La información útil por paquete puede oscilar entre 1 y 64 bytes para dispositivos FS y entre 1 y 8 bytes para dispositivos LS.
- El sistema puede asignar como máximo el 90% del tiempo de Trama para transferencias isócronas y de interrupción. Si el sistema no puede garantizar tiempo suficiente como para manejar una nueva conexión de interrupción (transmitir un nuevo paquete dentro del periodo máximo requerido), simplemente no se establece la conexión.

Transferencias Bulk

- Sólo son utilizables por dispositivos FS.
- Se procesan por medio de un mecanismo "good effort", en el que el sistema aprovecha cualquier ancho de banda disponible y en el momento en que esté disponible (en otras palabras, no se garantiza una latencia ni un ancho de

banda mínimos). Se puede utilizar el tiempo de Trama reservado y no consumido por transferencias de Control (10%).

- Incorporan mecanismos de control de errores para garantizar la entrega de datos.
- La información útil por paquete puede ser de 8, 16, 32 ó 64 bytes.

Estos 4 tipos de transferencias están disponibles como interfaces software que el sistema pone a disposición de los manejadores de dispositivo, estando los manejadores obligados a comunicarse con los dispositivos única y exclusivamente a través de estos 4 interfaces de programación. Esto viene a significar que un manejador de dispositivo USB jamás accede directamente al hardware del dispositivo, y por otro lado significa que todos los dispositivos USB deben cumplir necesariamente unas especificaciones básicas comunes, ya que deben gestionar adecuadamente los tipos de transferencias que soportan. Adicionalmente, los dispositivos USB se agrupan en Clases, de forma que todos los dispositivos de una misma Clase cumplen además con las especificaciones de dicha Clase, ya que la Clase incide directamente en la manera en que el software interactúa con el dispositivo.

MODELO LOGICO

Los dispositivos USB pueden tener una o más Configuraciones posibles, que definen distintas formas de funcionamiento. A nivel lógico, una determinada Configuración es un conjunto de Interfaces, donde cada Interfaz especifica qué partes del hardware del dispositivo se comunican con el sistema, donde cada una de estas partes de hardware se denomina Endpoint. En resumen, cada posible Configuración de un dispositivo USB es un conjunto de Interfaces y cada Interfaz es un conjunto de Endpoints. Los Endpoints son unidireccionales, y se direccionan por un número y por el sentido en que transfieren la información (IN (entrada) si transfieren información hacia el sistema, y OUT (salida) si transfieren información hacia el dispositivo).

La comunicación entre una aplicación y los distintos Endpoints de un dispositivo se realiza a través de USB por medio de unos caminos lógicos de transferencias de datos denominados Pipes, de forma que cada Pipe comunica la aplicación con un determinado Endpoint en el dispositivo. Las Pipes pueden ser de tipo Control (también denominadas de Mensaje), que son bidireccionales y con formato especificado por la norma, y de tipo Stream, que son unidireccionales (tipo FIFO) y con formato libre no especificado por la norma. Las Pipes de Control conectan la aplicación con un Endpoint de Control (formado por una pareja de Endpoints uno IN y otro OUT) para realizar transferencias bidireccionales de Control. Las Pipes Stream conectan la aplicación con un Endpoint para realizar transferencias unidireccionales Isócronas, Interrupción y Bulk.

Todos los dispositivos USB deben implementar los dos Endpoints 0 (IN y OUT) para permitir que el sistema pueda establecer la Pipe de Control por Defecto y pueda acceder a información de identificación y requisitos de configuración y pueda configurar el dispositivo. Adicionalmente, USB permite direccionar otros 15 Endpoints IN y 15 Endpoints OUT por dispositivo FS y otras 2 Pipes de Control y/o Interrupción por dispositivo LS. Estos Endpoints adicionales son opcionales y dependientes de los requisitos de implementación del dispositivo.

CLASES USB

Una Clase USB es un grupo de dispositivos (o interfaces) con atributos o características similares. Las especificaciones para cada Clase permiten el desarrollo de dispositivos que pueden controlarse por medio de un manejador adaptativo, es decir, que se configura según la Clase reportada por el dispositivo. Dos dispositivos (o interfaces) pertenecen a la misma Clase si por ejemplo utilizan una misma forma de comunicarse con el sistema, o si por ejemplo utilizan el mismo formato de datos.

Las Clases USB se usan principalmente para describir la manera en que los dispositivos (o interfaces) se comunican con el sistema, incluyendo los mecanismos de control y datos, y adicionalmente algunas Clases se usan para identificar en todo o en parte la funcionalidad del dispositivo (o interfaz). En este caso, la Clase se puede utilizar para identificar qué manejador debe controlar dicho dispositivo (o interfaz).

Adicionalmente, los dispositivos de una Clase pueden agruparse en Subclases, lo que facilita aún más el que los manejadores puedan explorar el bus y seleccionar todos aquellos dispositivos que pueda controlar.

USB 2.0

El Grupo de Promoción de USB 2.0, formado por Compaq, HP, Intel, Lucent, Microsoft, NEC y Philips, han anunciado un posible incremento en la velocidad hasta los 480 Mbps, es decir, 40 veces más rápido que USB 1.1. Será compatible con dispositivos USB 1.1 y utilizará los mismos cables y conectores. Sólo se necesitarán nuevos hubs algo más complicados que los actuales, ya que tendrán que comunicarse tanto con dispositivos 2.0 como con los actuales 1.1 FS y LS.