

# **UN PASEO POR**

## **USB 2.0**

**Soporte Técnico OEM**

**FUJITSU ESPAÑA**

**Marzo de 2003**

# INDICE

**Referencias**

**Introducción**

**Interfaz Eléctrico**

**Concentradores (Hubs)**

**Identificación de la velocidad**

**Tramas y microtramas**

**Endpoints High-Speed y High Bandwidth**

**Protocolo I: Paquetes y Transacciones**

- **Paquete de Token SOF y Número de Microtrama**
- **Paquetes de Datos**
- **Paquete de Handshake NYET**
- **Paquete Especial de Handshake ERR**
- **Paquetes Especiales de Token para Transacciones Split**
- **Paquete Especial de Token PING para Control de Flujo**

**Protocolo II: Pipes y Transferencias**

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

**ANEXOS**

- **Resumen de tipos y características de Pipes y Transferencias.**
- **Descomposición de las Transferencias en Transacciones y Paquetes.**
- **Estructura de los Paquetes.**

### Referencias

- Especificación de Bus Serie Universal Revisión 1.1. Compaq, Intel, Microsoft y NEC. 23 de Septiembre de 1998.
- Especificación de Bus Serie Universal Revisión 2.0. Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC y Philips. 27 de Abril de 2000.
- Un paseo por USB-1. Soporte Técnico OEM, Fujitsu España. 27 de Marzo de 2000

### Introducción

La norma USB 1.1 definía dos modos de transferencia a 12 Mbps (full-speed) y a 1,5 Mbps (low-speed). USB 2.0 añade un nuevo modo a 480 Mbps, denominado high-speed.

Los dispositivos high-speed también soportan una mínima funcionalidad en modo full-speed, de forma que cuando se conectan a un puerto que está trabajando en modo full-speed, pueden (al menos):

- Detectar y procesar el reset, y
- Aceptar, procesar y responder adecuadamente a las funciones estándar de asignación de dirección y de configuración, así como a las de lectura de la información descriptiva del dispositivo y de sus posibles configuraciones (descriptores).

Esta funcionalidad mínima permite que el sistema pueda detectar, identificar y configurar a los dispositivos high-speed aunque se conecten a puertos full-speed. El dispositivo high-speed puede o puede no soportar su total funcionalidad cuando trabaja en modo full-speed.

Por su parte, los concentradores high-speed soportan en sus puertos downstream los tres modos de funcionamiento (low, full y high-speed), de manera que admiten la conexión de cualquier dispositivo o concentrador USB 1.x y USB 2.0. Asimismo, soportan en sus puertos upstream los modos full y high-speed, de manera que se pueden conectar a cualquier concentrador o controlador USB 1.x o USB 2.0. Esto quiere decir que en un mismo sistema USB pueden coexistir dispositivos y concentradores USB 1.x y USB 2.0 (Low, Full y High Speed).

### Interfaz Eléctrico

Para transmitir en modo high-speed, el transmisor activa una fuente de corriente interna, derivada a partir de su fuente de alimentación positiva, y dirige dicha corriente hacia una de las dos líneas de datos por medio de un conmutador de corriente de alta velocidad. De esta manera, el transmisor genera los estados J y K high-speed en el cable.

Esta conmutación dinámica de corriente sobre ambas líneas de datos D+ y D-, sigue las mismas reglas de codificación NRZI y de inserción de bit (bit-stuffing) ya utilizadas en los modos full-speed y low-speed. El estado J se obtiene dirigiendo la corriente sobre la línea D+, mientras que el estado K se obtiene dirigiendo la corriente sobre la línea D-.

En modo high-speed, tanto el transmisor como el receptor activan unas resistencias de terminación entre cada línea y masa (45 Ohms +/- 10 %), de forma que el valor nominal de la corriente (17,78 mA) produce un voltaje nominal en la línea de 400 mV. El voltaje diferencial nominal (D+ - D-) es, por lo tanto, de +400 mV para el estado J y de - 400 mV para el estado K.

### Concentradores (Hubs)

Al igual que en USB 1.x, los concentradores siguen siendo el elemento clave sobre el que se construye el bus USB, y los concentradores USB 2.0 tienen mucho que ver con algunas de las mejoras introducidas en USB 2.0.

Los concentradores y controladores de host USB 2.0 incorporan una nueva funcionalidad (respecto a los concentradores USB 1.x) que consigue un uso mucho más óptimo del bus cuando se conectan dispositivos de velocidad inferior a la máxima. Para entenderlo mejor, vamos a compararlo con el caso de un bus USB 1.x:

- En un bus USB 1.x (full-speed), el tráfico entre el host y los dispositivos low-speed se produce a velocidad low-speed. Es decir, tanto el host como todos los concentradores intermedios transmiten en modo low-speed. Esto quiere decir que todo el bus está ocupado durante todo el tiempo que dura la transmisión low-speed (1 bit a velocidad low-speed ocupa el bus durante un tiempo equivalente a 8 bits a velocidad full-speed). Esto supone una gran penalización en el ancho de banda disponible para los dispositivos full-speed.
- En un bus USB 2.0 (high-speed), el host y todos los concentradores USB 2.0 intermedios transmiten siempre en modo high-speed, y sólo el tráfico entre el dispositivo full o low-speed y el concentrador USB 2.0 al que está conectado se produce a velocidad full o low-speed. De esta manera se pueden solapar las transferencias full y low-speed con transferencias high-speed con otros dispositivos, lo que minimiza el impacto que las transmisiones a velocidades full y low-speed tienen sobre el ancho de banda disponible para los dispositivos high-speed.

Para ello, los concentradores USB 2.0 disponen de un nuevo elemento: el Traductor de Transacciones (TT). En total, un concentrador USB 2.0 se compone de:

- Un Controlador. Permite la comunicación con el host, de forma que el host puede configurar el concentrador y monitorizar y controlar sus puertos.
- Un Repetidor. Es un conmutador entre los puertos upstream y downstream, que se utiliza como repetidor de paquetes cuando la velocidad en ambos lados es la misma (ya sea high, full o low-speed).
- Uno (mono-TT) o varios (multi-TT) Traductores de Transacciones. Se usan para “traducir” las transacciones high-speed a transacciones full o low-speed. Se puede decir que el concentrador mantiene aislados los segmentos del bus que usan full y low-speed de aquellos otros segmentos donde sólo se transmite en high-speed.

De esta manera, cada concentrador USB 2.0 funcionando en modo high-speed, básicamente añade uno o más buses USB 1.x (full/low-speed), disponiendo cada uno de ellos de sus 12 Mbps de ancho de banda:

- Un concentrador mono-TT sólo dispone de un “traductor” común para todos los puertos, por lo que todos los dispositivos full/low-speed conectados a dicho concentrador comparten un ancho de banda de 12 Mbps (el TT sirve a todos ellos pero de uno en uno).
- Un concentrador multi-TT dispone de un “traductor” para cada puerto, por lo que cada dispositivo full/low-speed conectado al mismo, dispone de un ancho de banda de 12 Mbps (los distintos TTs pueden trabajar en paralelo, lo que permite efectivamente tráfico en paralelo en todos los puertos full/low-speed).

Arriba se ha dicho “cada concentrador USB 2.0 funcionando en modo high-speed”. Hay que hacer esta puntualización porque un concentrador USB 2.0 conectado a un puerto USB 1.x no funciona en modo high-speed, sino en modo full-speed. En este caso no se hace nunca uso del TT (ya que no hay transacciones high-speed que traducir), y el concentrador USB 2.0 se comporta igual que si fuera USB 1.x (full-speed):

- Un concentrador USB 2.0 conectado a un puerto high-speed, siempre se comunica en modo high-speed en su puerto upstream, mientras que permite la conectividad en sus puertos downstream tanto de dispositivos high-speed (a través del repetidor), como de dispositivos full/low-speed (a través del o de los TTs).
- Un concentrador USB 2.0 conectado a un puerto full-speed, siempre se comunica en modo full-speed en su puerto upstream, y deshabilita el modo high-speed en todos sus puertos downstream. Quiere decir que los dispositivos high-speed conectados a este concentrador van a funcionar en modo full-speed. En este caso, las transferencias siempre se canalizan a través del repetidor, que se comporta como un repetidor USB 1.x (full/low-speed).

A consecuencia de todo esto, los dispositivos y concentradores high-speed:

- Se comportan como dispositivos high-speed si se conectan a puertos high-speed.

- Se comportan como dispositivos full-speed si se conectan a un puerto USB 1.x o bien a un concentrador USB 2.0 que tenga deshabilitado el modo high-speed (típicamente es el caso cuando el concentrador USB 2.0 está conectado a un puerto USB 1.x).

### Identificación de la velocidad

Los dispositivos USB 1.x identifican su velocidad eléctricamente, mediante una resistencia de pull-up en la línea D+ (full-speed) o D- (low-speed).

En cambio los dispositivos high-speed se identifican en principio como full-speed (mediante un pull-up en la línea D+), y durante el proceso de reset ejecutan un protocolo de bajo nivel (determinadas secuencias de señales eléctricas de niveles especiales) a través del cuál determinan si están conectados a un puerto full-speed o high-speed. Si el puerto es high-speed, detectará y responderá al protocolo iniciado por el dispositivo. Sólo si el concentrador y el dispositivo ejecutan el protocolo, se establece una comunicación high-speed entre ellos.

### Tramas y microtramas

USB 1.x dividía el tiempo en tramas de 1 milisegundo. Adicionalmente, USB 2.0 define un tiempo de microtrama de 125 microsegundos.

Al igual que en USB 1.x se reservan ciertos porcentajes del tiempo de trama, para dar servicio a las distintas transacciones de Control, Interrupción e Isócronas full/low-speed, en USB 2.0 también se reservan ciertos porcentajes del tiempo de microtrama, para dar servicio a los distintos tipos de transacciones high-speed.

### Endpoints High-Speed y High Bandwidth

En USB 1.x, el tiempo máximo reservado en cada trama a cada endpoint Isócrono o de Interrupción es el de una transacción por trama.

USB 2.0 soporta dos tipos de endpoints Isócronos y de Interrupción:

- Endpoints de ancho de banda normal, que precisan hasta 1,024 bytes por microtrama (una transacción), y
- Endpoints de alto ancho de banda (High Bandwidth), que precisan más de 1,024 bytes por microtrama, hasta un máximo de 3,072 bytes (3 transacciones).

### Protocolo I: Paquetes y Transacciones

USB 2.0 mantiene la arquitectura centralizada definida en USB 1.x, en la que el host es el iniciador de todas las transferencias que se producen en el bus. Los dispositivos deben mantenerse a la espera hasta recibir del host un paquete especial (token) dirigido a él, indicando el tipo de transferencia a realizar.

USB 2.0 también mantiene la estructura de la transacción, formada en base a la secuencia de paquetes Token-Data-Handshake, definida en USB 1.x. Básicamente USB 2.0 añade algunos tipos de paquetes nuevos, para implementar las nuevas funciones y protocolos que se han incorporado.

En el grupo de paquetes de Token, siguen existiendo los definidos en USB 1.x (IN, OUT, SOF y SETUP), aunque USB 2.0 reutiliza el tipo SOF:

- USB 1.x utiliza el token SOF para indicar el Principio de Trama (una vez cada milisegundo). USB 2.0 usa este mismo token para indicar el Principio de Microtrama (una vez cada 125 microsegundos).

En el grupo de paquetes de Datos, aparte de los dos tipos definidos en USB 1.x (DATA0 y DATA1), USB 2.0 define dos nuevos tipos:

- DATA2, empleado en transacciones isócronas high-speed high-bandwidth, y

- MDATA, empleado en transacciones isócronas high-speed high-bandwidth y en transacciones Split (las transacciones se describen más adelante)..

En el grupo de paquetes de Handshake, aparte de los tres tipos definidos en USB 1.x (ACK, NAK y STALL), USB 2.0 define un nuevo paquete:

- NYET, significa “Not YET” (todavía no), y se emplea en los protocolos Split y de control de flujo PING.

Por último, en el grupo de paquetes Especial, USB 2.0 define tres nuevos tipos de paquetes, reutilizando uno de ellos (ERR) el código asignado por USB 1.x al paquete especial PRE. Se recuerda aquí la descripción del paquete PRE, para mostrar que no hay posibilidad de error en cuanto a que un mismo código identifique a dos paquetes distintos:

- PRE (Preámbulo): Definido en USB 1.x para indicar que a continuación se va a transmitir en modo low-speed. Sólo lo puede transmitir un host USB 1.x (en modo full-speed) inmediatamente antes del campo de token de la transacción low-speed.
- ERR (Error): Se usa para indicar errores en transacciones Split high-speed. Reutiliza el código del tipo de paquete PRE. Sólo lo puede transmitir un concentrador USB 2.0 (en modo high-speed) en el campo de handshake de una transacción Split.
- SPLIT: Lo transmite el host en el campo de token de una transacción Split.
- PING: Lo transmite el host en el campo de token de una transacción de control de flujo PING.

### • Paquete de Token SOF y Número de Microtrama

En USB 1.x, el número de la trama es un campo de 11 bits que el host incrementa una vez por trama, y que envía sólo en el paquete SOF (Start-Of-Frame) al principio de cada trama.

USB 2.0 utiliza este mismo campo para indicar el número de microtrama. En este caso, el host no envía todos los bits del número de microtrama (14 bits), sino solamente los 11 bits más altos. Como cada trama (1 ms) contiene 8 microtramas (125 us), el efecto de lo anterior es que el host USB 2.0 envía durante 8 microtramas consecutivas un paquete SOF con el mismo número de trama. Los dispositivos full-speed reciben un paquete SOF cada milisegundo con un número de trama secuencial. Los dispositivos high-speed reciben un paquete SOF cada 125 microsegundos, pero con el mismo número de trama en cada grupo de 8 consecutivos.

Los dispositivos high-speed que necesiten llevar un control del número de microtrama, pueden detectar cuándo un paquete SOF tiene un número de trama distinto del anterior, e iniciar un contador interno de microtrama (de 3 bits) a 0. Cada uno de los siguientes 7 paquetes SOF con el mismo número de trama serán los correspondientes a las microtramas 1 a 7.

### • Paquetes de Datos

USB 2.0 sigue haciendo un uso de los paquetes DATA0 y DATA1, como parte del mecanismo de detección de errores Data Toggle en los casos definidos en USB 1.x; adicionalmente USB 2.0 define los nuevos paquetes DATA2 y MDATA, de nuevo con la función de detección de errores en los nuevos casos definidos en USB 2.0. El uso de los paquetes de datos en los nuevos casos definidos en USB 2.0 es:

- Las transacciones de Interrupción high-speed high bandwidth realizadas en la misma microtrama, hacen uso del mecanismo Data Toggle (uso alternativo de paquetes DATA0 y DATA1).
- En transacciones Isócronas-IN high-speed high-bandwidth, el dispositivo envía los datos utilizando las siguientes secuencias de paquetes:
  - En el caso de 1 transacción por microtrama, se usa el paquete DATA0.
  - En el caso de 2 transacciones por microtrama, se usa la secuencia de paquetes DATA1-DATA0, y
  - En el caso de 3 transacciones por microtrama, se usa la secuencia de paquetes DATA2-DATA1-DATA0.

- En transacciones Isócronas-OUT high-speed high-bandwidth, el host envía los datos utilizando las siguientes secuencias de paquetes:
  - En el caso de 1 transacción por microtrama, se usa el paquete DATA0.
  - En el caso de 2 transacciones por microtrama, se usa la secuencia de paquetes MDATA-DATA1, y
  - En el caso de 3 transacciones por microtrama, se usa la secuencia de paquetes MDATA-MDATA-DATA2.
- En transacciones Start-Split de Interrupción-OUT, el host alterna los paquetes DATA0 y DATA1 en la manera habitual (data toggle).
- En transacciones Complete-Split de Interrupción-IN, el concentrador también alterna los paquetes DATA0 y DATA1, siempre que la transacción en el bus full/low-speed se haya completado durante una microtrama. Cuando se da el caso de que la transacción en el bus full/low-speed abarca dos microtramas, la respuesta del concentrador en la primera transacción Complete-Split usa el paquete MDATA, para indicar al host que faltan datos por enviar, y que debe ejecutar otra transacción Complete-Split en la siguiente microtrama, para obtener el resto de los datos (que se transmitirán mediante un paquete DATA0 o DATA1, según corresponda en la secuencia Data Toggle).
- En transacciones Start-Split Isócronas-OUT, el host siempre usa el paquete DATA0.
- En transacciones Complete-Split Isócronas-IN, el concentrador usa el paquete MDATA para todos los paquetes excepto para el último, en que usa el paquete DATA0 (la transacción full-speed puede abarcar varias microtramas, por lo que los paquetes MDATA informan al host que debe seguir enviando transacciones Complete-Split en cada microtrama, hasta que reciba un paquete DATA0).

La máxima cantidad de información útil que se puede transferir en un paquete de datos depende de la velocidad del dispositivo y del tipo de endpoint.

### • Paquete de Handshake NYET

Se utiliza sólo en modo high-speed, y se puede transmitir en dos circunstancias:

- Lo puede transmitir un endpoint high-speed tipo Bulk-OUT o de Control, como respuesta durante el protocolo de control de flujo PING.
- También lo puede transmitir un concentrador high-speed como respuesta a una transacción Split, cuando el concentrador todavía no ha concluido la transacción full/low-speed con el dispositivo, o bien cuando el concentrador no puede atender la transacción en ese momento.

### • Paquete Especial de Handshake ERR

Se utiliza sólo en modo high-speed, y lo puede transmitir un concentrador high-speed en el campo de handshake de una transacción Complete-Split, para informar de un error en la transacción full/low-speed.

### • Paquetes Especiales de Token para Transacciones Split

Aparte de las transacciones definidas en USB 1.x (Control, Bulk, Interrupción e Isócronas), USB 2.0 define las transacciones Split. Sólo los controladores de host y los concentradores USB 2.0 deben soportar este nuevo tipo de transacción, ya que es invisible a los dispositivos.

Este tipo de transacción es el que permite la comunicación con dispositivos full/low speed conectados a concentradores que funcionan en modo high-speed. El host comienza una transacción Split cuando envía al concentrador, en modo high-speed, toda la información necesaria para que el concentrador ejecute ahora una transacción full/low-speed con el dispositivo. Toda la información queda almacenada en el concentrador, en el TT correspondiente al puerto al que está conectado el dispositivo (o en el único TT disponible si se trata de un concentrador mono-TT). Mientras el concentrador ejecuta esta transacción con el dispositivo full/low-speed, el bus queda libre para ejecutar nuevas transacciones high-speed con otros dispositivos.

La respuesta del dispositivo full/low-speed queda almacenada a su vez en el TT del concentrador, disponible para cuando el host posteriormente indique al concentrador que la envíe, en modo high-speed.

USB 2.0 define el token especial SPLIT para llevar a cabo las transacciones Split. Este es el único paquete de token de 4 bytes, a diferencia de los paquetes de token normales de 3 bytes.

USB 2.0 define dos transacciones Split que hacen uso del token especial SPLIT:

- **Start Split:** La utiliza el host para enviar al concentrador, en modo high-speed, toda la información necesaria para que el concentrador ejecute ahora una transacción full/low-speed con el dispositivo.
- **Complete Split:** La utiliza el host para leer del concentrador, en modo high-speed, la respuesta del dispositivo full/low-speed.

Un campo en el propio paquete SPLIT identifica el tipo de transacción.

### • Paquete Especial de Token PING para Control de Flujo

Un caso bastante frecuente y que produce una gran pérdida de ancho de banda útil en un bus USB 1.x, se da cuando los dispositivos full/low-speed contienen endpoints tipo Bulk-OUT y de Control que necesitan un tiempo para procesar los datos recibidos, de forma que no pueden momentáneamente recibir nuevos datos hasta que se desocupe el buffer de recepción.

En USB 1.x, esta situación la controla el dispositivo devolviendo un handshake NAK en la transacción OUT en la que el host ha transmitido un nuevo paquete de datos. Este handshake indica que el dispositivo no ha podido recibir los datos porque no tiene espacio suficiente en su buffer de recepción, y el host debe reintentar la transmisión posteriormente. Desafortunadamente, para cuando el dispositivo informa que no tiene espacio, la mayor parte del tiempo de la transacción ya se ha consumido, ya que el paquete de datos se ha transferido íntegramente. Esto produce una pobre utilización del bus cuando se suceden múltiples transacciones OUT con respuesta negativa (NAK) por parte del dispositivo.

USB 2.0 define un nuevo protocolo de control de flujo más eficiente, denominado PING, que debe ser utilizado por los endpoints de tipo Bulk-OUT y de Control. Los endpoints de Control deben soportar este protocolo en las fases de Datos y de Estado, pero no en la fase de Setup. Este nuevo protocolo evita la transmisión de paquetes de datos hasta que el host sabe que el dispositivo puede aceptarlos.

El host pregunta al dispositivo high-speed mediante el paquete especial de token PING, y el dispositivo puede contestar con un handshake ACK, para indicar que tiene espacio para recibir un nuevo paquete de datos, o con un handshake NAK, para indicar que no tiene espacio. El host pregunta periódicamente mediante paquetes PING hasta que recibe un handshake ACK, en cuyo caso procede a la transmisión del paquete de datos.

Una vez que se produce la recepción de un paquete de datos, el dispositivo puede contestar con un handshake ACK, para indicar la correcta recepción del paquete y que tiene espacio para el siguiente paquete, o con un handshake NYET, para indicar la correcta recepción del paquete y que no tiene espacio para el siguiente paquete.

El host puede seguir transmitiendo paquetes de datos en tanto que el dispositivo siga contestando con ACK. En el momento en que el dispositivo conteste con NYET, el host debe volver al proceso de preguntar al dispositivo mediante paquetes PING antes de enviar un nuevo paquete de datos.

## Protocolo II: Pipes y Transferencias

USB 1.x define las vías de comunicación entre las aplicaciones que se ejecutan en el host (clientes) y los distintos endpoints en los dispositivos USB (servidores), y las denomina “pipes”. Cuando un dispositivo USB se conecta a un sistema, y el sistema lo reconoce y lo configura, el dispositivo queda organizado como un cierto conjunto de endpoints de distintos tipos (existen 4 tipos de endpoints). Entonces el sistema establece todas las vías de comunicación (pipes) necesarias entre el sistema y cada uno de los endpoints disponibles en dicha configuración. El dispositivo puede implementar varias posibles configuraciones, con distintos conjuntos de endpoints de distintos tipos en cada una de ellas. El sistema elige una cierta configuración en función de la funcionalidad particular que se precise del dispositivo.

Existen 4 tipos de endpoints (Bulk, Control, Interrupción e Isócrono) y 2 tipos de pipes (Control o Mensaje y Stream). Las posibles combinaciones son:

- Pipe de Control o Mensaje. Es una vía de comunicación bidireccional entre el host y dos endpoints de Control en un dispositivo USB. Un endpoint es de Salida y el otro es de Entrada, de forma que se pueda establecer la comunicación bidireccional. Todos los dispositivos USB disponen de dos endpoints de Control en la dirección 0, uno de entrada y uno de salida, de manera que el sistema siempre puede establecer una pipe de Control con el dispositivo, incluso antes de configurarlo (se denomina Pipe de Control por Defecto, y es la única pipe que se puede establecer antes de configurar al dispositivo). A través de esta pipe, el sistema puede leer del dispositivo toda la información descriptiva necesaria para enterarse del tipo de dispositivo, posibles configuraciones, protocolos que soporta, número y tipos de endpoints que soporta en cada posible configuración, etc. Esta información descriptiva son los Descriptores.
- Pipe Stream. Es una vía de comunicación unidireccional entre el host y un endpoint de los tipos Bulk, Interrupción o Isócrono. Si un dispositivo necesita transferencias bidireccionales de un tipo de endpoint concreto, el sistema debe establecer dos pipes, una de salida (con un endpoint de salida) y otra de entrada (con un endpoint de entrada).

Las diferencias básicas entre las transferencias USB 1.x y USB 2.0 son:

- Tamaños máximos de los paquetes de datos en cada tipo de transferencia, y
- Reserva de tiempo de microtrama para las transferencias de Control, de Interrupción e Isócronas.

### • Transferencias de Control

Las transferencias de Control proporcionan control de flujo y una entrega de datos garantizada y libre de errores.

Todos los dispositivos full, high y low-speed pueden incorporar endpoints de Control, y por lo tanto pueden hacer uso de las transferencias de Control. Todos implementan, al menos, un endpoint de salida y uno de entrada en la dirección 0, para poder establecer la Pipe de Control por Defecto.

Las transferencias de Control se componen de 3 transacciones denominadas Setup-Dato-Estado. Los tamaños máximos del paquete de datos durante la transacción de datos son:

- Full-speed: 8, 16, 32, ó 64 bytes
- High-speed: 64 bytes
- Low-speed: 8 bytes.

USB hace una gestión “best effort” para ir dando curso a las distintas transferencias de Control pendientes en cada momento en todas las pipes de Control establecidas con todos los dispositivos. Para ello se hace la siguiente reserva del tiempo de trama o microtrama:

- En un bus full/low-speed, la reserva es del 10% del tiempo de trama.
- En un bus high-speed, la reserva es del 20% del tiempo de microtrama.

Las reglas definidas por USB para el envío de las transferencias pendientes son:

- Si el tiempo de trama o microtrama utilizado por las transferencias de Control pendientes es inferior al reservado, el tiempo restante puede utilizarse para transferencias Bulk.
- Si hay más transferencias de Control pendientes que tiempo reservado, pero hay tiempo adicional en la trama o microtrama no consumido por transferencias de Interrupción o Isócronas, entonces el host puede utilizar dicho tiempo adicional para enviar nuevas transferencias de Control.
- Si hay más transferencias de Control pendientes que tiempo disponible en una trama o microtrama, el host selecciona cuáles se procesan, quedando el resto pendientes para una próxima trama o microtrama.

Los endpoints de Control high-speed soportan el protocolo de control de flujo PING en las transacciones de Dato y Estado de salida.

### • Transferencias Isócronas

Las transferencias Isócronas están diseñadas para soportar aquellos dispositivos que precisan una entrega de datos a velocidad constante, y en la que no importa la pérdida eventual de información. Esto es necesario para aplicaciones en que la información de tiempo va implícita en la propia velocidad de transmisión/recepción de datos (isocronismo).

Para ello, las transferencias Isócronas proporcionan:

- Ancho de banda garantizado
- Latencia limitada
- Velocidad de transferencia de datos constante garantizada a través de la pipe
- En caso de error en la entrega, no se reintenta la transmisión
- Sin control de flujo

Sólo los dispositivos high y full-speed pueden incorporar endpoints Isócronos.

Las transferencias Isócronas se componen sólo de transacciones de datos. Las frecuencias y los tamaños de los paquetes de datos son:

- Full-speed: 1 transacción por trama de hasta 1,023 bytes.
- High-speed: 1 transacción por microtrama de hasta 1,024 bytes.
- High-speed high-bandwidth: 2 ó 3 transacciones por microtrama de hasta 1,024 bytes cada una.

La gestión que hace USB para garantizar las transferencias es la de establecer o no la pipe en función de que haya suficiente tiempo libre de trama o microtrama para realizarlas. Para ello, los endpoints Isócronos indican qué cantidad de información como máximo debe transferir la pipe en cada trama o microtrama, de forma que el sistema USB puede calcular si hay suficiente tiempo o no para acomodar la pipe, y en función de eso la establece o no.

La reserva de tiempo de trama o microtrama para acomodar transferencias Isócronas y de Interrupción es como máximo el tiempo no reservado para transferencias de Control. El sistema USB puede ir estableciendo pipes Isócronas y de Interrupción con distintos dispositivos hasta agotar dicha reserva:

- Full-speed: Hasta un 90% del tiempo de trama.
- High-Speed: Hasta un 80 % del tiempo de microtrama.

### • Transferencias de Interrupción

Las transferencias de Interrupción están diseñadas para soportar aquellos dispositivos que precisan enviar o recibir datos de manera no frecuente, pero con ciertos límites de latencia.

Para ello, las transferencias de Interrupción proporcionan:

- Tiempo máximo de servicio (latencia) garantizado
- Reintento de transferencia en el siguiente periodo, en caso de eventual fallo en la entrega.

Todos los dispositivos high, full y low-speed pueden incorporar endpoints de Interrupción.

Las transferencias de Interrupción se componen sólo de transacciones de datos. Los tamaños de los paquetes de datos son:

- Low-speed: hasta 8 bytes.
- Full-speed: hasta 64 bytes.
- High-speed: hasta 1,024 bytes.
- High-speed high-bandwidth: 2 ó 3 transacciones por microtrama de hasta 1,024 bytes cada una.

La gestión que hace USB para garantizar las transferencias es la de establecer o no la pipe en función de que haya suficiente tiempo libre de trama o microtrama para realizarlas. Para ello, los endpoints de Interrupción indican qué cantidad de información como máximo debe transferir la pipe en cada transacción, así como el tiempo máximo entre transacciones, de forma que el sistema USB puede calcular si hay suficiente tiempo o no para acomodar la pipe, y en función de eso la establece o no.

El tiempo máximo entre transacciones (tiempo de latencia máximo) especificado por cada dispositivo puede ser:

- Low-speed: de 10 a 255 ms
- Full-speed: de 1 a 255 ms
- High-speed: de 125 us a 4'096 seg.

La reserva de tiempo de trama o microtrama para acomodar transferencias Isócronas y de Interrupción es como máximo el tiempo no reservado para transferencias de Control. El sistema USB puede ir estableciendo pipes Isócronas y de Interrupción con distintos dispositivos hasta agotar dicha reserva:

- Full y Low-speed: Hasta un 90% del tiempo de trama.
- High-Speed: Hasta un 80 % del tiempo de microtrama.

### • **Transferencias Bulk**

Las transferencias Bulk están diseñadas para soportar aquellos dispositivos que precisan enviar o recibir grandes cantidades de datos, con latencias que pueden tener amplias variaciones, y en que las transacciones pueden utilizar cualquier ancho de banda disponible.

Para ello, las transferencias Bulk proporcionan:

- Acceso al bus en función del ancho de banda disponible.
- Reintento de transferencias en caso de errores de entrega.
- Entrega garantizada de datos, pero sin garantía de latencia máxima ni de ancho de banda.

Las transferencias Bulk se realizan relativamente rápidas si el bus dispone de mucho ancho de banda libre, pero en un bus USB con mucho ancho de banda reservado, pueden alargarse durante periodos de tiempo relativamente grandes.

Sólo los dispositivos high y full-speed pueden incorporar endpoints Bulk.

Las transferencias Bulk se componen sólo de transacciones de datos. Los tamaños de los paquetes de datos son:

- Full-speed: 8,16, 32 y 64 bytes.
- High-speed: 512 bytes.

USB hace una gestión “good effort” para ir dando curso a las distintas transferencias pendientes en cada momento en todas las pipes Bulk establecidas con todos los dispositivos. Las transferencias de Control tienen preferencia sobre las Bulk, por lo que las transferencias Bulk se realizan siempre que no haya otro tipo de transferencias que hacer en una trama o microtrama.

Los endpoints Bulk-OUT high-speed soportan el protocolo de control de flujo PING.

## **ANEXOS**

- **Resumen de tipos y características de Pipes y Transferencias.**
- **Descomposición de las Transferencias en Transacciones y Paquetes.**
- **Estructura de los Paquetes.**

La aplicación envía una IRP (I/O Request Packet) a una Pipe: El sistema USB la ejecuta en una o más Transferencias					
PIPE	TRANSFERENCIA	ENTREGA	DISPOSITIVO	PAYLOAD (bytes)	CARACTERISTICAS
CONTROL (MENSAJE)	Control	Best Effort	High Speed	64	Libre de Errores Entrega Garantizada Control de Flujo
			Full Speed	8/16/32/64	
			Low Speed	8	
STREAM	Bulk	Good Effort	High Speed	512	
			Full Speed	8/16/32/64	
			Low Speed		
	Interrupción	Periodo garantizado	High Speed	<= 1,024 (125 us - 4 s)	
			Full Speed	<= 64 (1 - 255 ms)	
			Low Speed	<= 8 (10 - 255 ms)	
	Isócrona	1, 2 ó 3 paquetes/microtrama	High Speed	<= 1,024	Ancho de Banda Garantizado Latencia Limitada Velocidad Constante de Datos Sin Retransmisiones
Full Speed			<= 1,023		

DESCOMPOSICIÓN DE LAS TRANSFERENCIAS EN TRANSACCIONES Y PAQUETES					
TRANSFERENCIA	TRANSACCIÓN	DESCOMPOSICIÓN DE LA TRANSACCIÓN EN PAQUETES			
		TOKEN (desde Host)	DATA (desde Origen de datos)	HANDSHAKE (desde Destino de datos)	
CONTROL	SET UP	SETUP	DATA0 desde Host	ACK / Nada desde Dispositivo	
	DATA (opcional)	IN / OUT / PING	Igual que Bulk (DATA1 + toggling)	Igual que Bulk	
	STATUS (Ver NOTA)	IN	DATA1 (0 bytes) desde dispositivo = CMD Completo		ACK / Nada desde HOST
			STALL (fallo en CMD) desde dispositivo		No existe
		OUT (FS / LS)	DATA1 (0 bytes) desde Host		ACK / NAK / STALL / Nada desde dispositivo
			PING (HS)	ACK / NAK / STALL / Nada desde dispositivo	
BULK	DATOS	IN	DATA0/1 desde dispositivo	ACK / Nada desde HOST	
			NAK / STALL / Nada desde dispositivo	No existe	
		OUT	DATA0/1 desde Host	ACK/NAK/STALL/NYET/Nada desde dispositivo	
		PING	ACK / NAK / STALL / Nada desde dispositivo	No existe	
INTERRUPCIÓN	DATOS	IN	DATA0/1 desde dispositivo	ACK / Nada desde HOST	
			NAK / STALL / Nada desde dispositivo	No existe	
		OUT	DATA0/1 desde Host	ACK / NAK / STALL / Nada desde dispositivo	
ISÓCRONA	DATOS	IN / OUT	DATA0 / DATA1 / DATA2 / MDATA	No existe	
NOTA: La transacción de STATUS se produce en la dirección contraria a la transacción DATA anterior (si no existe transacción DATA, se produce en dirección IN)		LEYENDA DE HANDSHAKE			
		NAK = No datos para transmitir / No Interrupción pendiente	ACK = Paquete OK (PID & bit stuff & CRC)		
		STALL = No puede transmitir	NAK = No puede aceptar datos. Host debe reenviar		
		Nada = Error en PID o bit stuff o CRC en paquete recibido	STALL = No puede recibir		

ESTRUCTURA DE LOS PAQUETES									
SYNC	PID (Packet ID)	DIR dispositivo	DIR endpoint	SPLIT info	Nº de trama	DATA	CRC	EOP	
8 bits (FS/LS)	4 bits PID + 4 bits /PID	7 bits	4 bits	12 bits	11 bits (Timestamp)	0 - 1,024 B (HS)	5 bits (Token)	2xSE0 + 1xJ (FS/LS)	
							0 - 1,023 B (FS)		8xJ / 8xK (HS; error bit stuff)
32 bits (HS)							8 B (LS)	16 bits (Data)	40xJ / 40xK (HS SOF; error bit stuff)
X	TOKEN OUT	X	X				X	X	
	TOKEN IN	X	X				X	X	
	TOKEN SOF				X		X	X	
	TOKEN SETUP	X	X				X	X	
	DATA0					X	X	X	
	DATA1					X	X	X	
	DATA2					X	X	X	
	MDATA					X	X	X	
	HANDSHAKE ACK							X	
	HANDSHAKE NAK							X	
	HANDSHAKE STALL							X	
	HANDSHAKE NYET							X	
	ESPECIAL TOKEN PRE								
	ESPECIAL HANDSHAKE ERR							X	
	ESPECIAL TOKEN SPLIT	DIR concentrador			X		X	X	
ESPECIAL TOKEN PING	X	X				X	X		