

# Master d'Informatique – 1ère année

## Réseaux et protocoles

### USB

Bureau S3-354

[Mailto:Jean.Saquet@unicaen.fr](mailto:Jean.Saquet@unicaen.fr)

<http://saquet.users.greyc.fr/M1/rezopro>

# USB - généralités

Le BUS USB fonctionne avec un maître (l'hôte) et des esclaves. (possibilité de négociation du rôle d'hôte en USB 2.0).

Les vitesses sont de 1,5 Mb/s ou 12 Mb/s (USB 1.0).  
Possibilité de 480 Mb/s en USB 2.0.

La documentation USB est impressionnante (650 pages, sans compter les annexes).

# USB - topologie

Le BUS USB est en étoile à étages. Il utilise des Hubs, éventuellement incorporés à des périphériques (ex. clavier).

Les alimentations des périphériques sont indépendantes, contrairement aux systèmes "daisy chaining" (SCSI, FireWire, ...). Les hubs permettent également de négocier les vitesses avec les périphériques.

# USB – adressage et principe

L'hôte n'a pas d'adresse (on communique toujours avec lui). Il y a 127 adresses possibles sur un Bus USB (l'adresse 0 est réservée).

Il s'agit d'un bus série à 4 fils, dont 2 pour l'alimentation (possibilité d'auto-alimentation des appareils).

Il utilise le "plug 'n play" : possibilité de brancher – débrancher les périphériques à chaud – en principe.

# USB – technique

Les connecteurs sont de type A ou B (hôte ou périphérique) Les Hubs ont un connecteur B et plusieurs de type A.

Les câbles A-A sont en principe interdits (sf pour connecter 2 ordinateurs en USB 2.0 avec négociation de l'hôte) et les "rallonges" également (pour respecter la longueur max. des câbles). Il y a également des "minis prises B" pour des périphériques de petite taille (appareils photos, ...)

# USB – communications

Elles s'effectuent dans un des modes suivants :

- contrôle
- interruption
- bloc
- isochrone

et ceci au moyen de TRANSACTIONS (tjs initiées par l'hôte) :

- Paquet Jeton
- Paquet Data (s'il y a lieu)
- Paquet Etat ou Handshake

# USB – champs des paquets

Sync : synchro des horloges (8bits)

Pid : Il y a 4 bits pour le PID, toutefois pour s'assurer qu'il a été reçu correctement, les 4 bits sont **complémentés** et répétés faisant un PID de 8 bits au total.

Addr : 7 bits. L'adresse 0 est utilisée en phase init.

FrameNumber : no de trame, sur 11 bits

Endp : Id de "terminaisons" 4 bits donc 16 terminaisons possibles par appareil

Crc sur 5 bits (jetons) ou 16 (autres paquets)

Eop : fin de paquet

## Utilisation de ces champs selon type de paquet

# Types de paquets

| Groupe                           | Valeur PID | Identificateur Paquet  |
|----------------------------------|------------|------------------------|
| Token<br>Jeton                   | 0001       | OUT Token              |
|                                  | 1001       | IN Token               |
|                                  | 0101       | SOF Token              |
|                                  | 1101       | SETUP Token            |
| Data<br>Données                  | 0011       | DATA0                  |
|                                  | 1011       | DATA1                  |
|                                  | 0111       | DATA2                  |
|                                  | 1111       | MDATA                  |
| Handshake<br>Poignée<br>de Mains | 0010       | ACK Handshake          |
|                                  | 1010       | NAK Handshake          |
|                                  | 1110       | STALL Handshake        |
|                                  | 0110       | NYET (No Response Yet) |
| Special                          | 1100       | PREamble               |
|                                  | 1100       | ERR                    |
|                                  | 1000       | Split                  |
|                                  | 0100       | Ping                   |

# USB – paquets jeton

Jetons : IN, OUT ou SETUP

L'hôte demande à lire ou à envoyer des données, ou à initialiser (installation, configuration)

Format : Sync, Pid, Addr, Endp, Crc5, Eop

# USB – paquets data

Data0 ou Data1 (même principe que bit alterné)  
Data2 et MDATA pour haute vitesse.

Format : Sync, Pid, Data, Crc16, Eop

Remarque : pas d'adresse ni endp dans ce paquet, car suit un paquet Jeton.

Taille max des données selon vitesse.

# USB – paquets handshake

Ack (réception OK)

Nack (pas prêt à recevoir)

Stall : bloqué

Format : Sync, Pid, Eop.

# USB – paquets début trame (SOF)

Trame 11 bits, envoyée régulièrement par l'hôte

Format : Sync, Pid, FrameNumber, Crc5, Eop

(trame de type jeton – il existe aussi des trames "spéciales")

# USB – appareils

Hôte, périphériques mais aussi hubs

Un hub détecte un branchement (tension sur les fils de données) et le signale à l'hôte sur interrogation de ce dernier.

De manière générale, un périphérique qui a des données à transmettre les met en mémoire tampon jusqu'à interrogation de l'hôte.

# USB – terminaisons

Emetteur ou récepteur de données (d'un périphérique ou hub)

Tout appareil a une terminaison zéro (reçoit les commandes et demandes d'état pour la phase d'énumération – reconnaissance des appareils)

Un canal de communication bidirectionnel est ouvert avec l'hôte sur cette terminaison zéro.

# USB – types de transfert - commandes

Utilisés pour commandes, états  
Essentiel dans la fonction d'énumération (installation)

Etape d'installation  
Etape de données  
Etape d'état

# USB – types de transfert - interruption

Toujours initiée par l'hôte (ce n'est donc pas une vraie interruption)

L'hôte doit interroger régulièrement toutes les terminaisons selon un taux défini dans le descripteur de terminaison

La donnée est mise en attente par l'appareil, et envoyée sur sollicitation de l'hôte.

# USB – types de transfert - isochrone

Utilisé par exemple pour trains de données audio/video  
Garantie de bande passante, de temps d'attente maximal  
Unidirectionnel, pas de reprise sur erreur (pas d'étape handshake)

Uniquement en pleine et haute vitesse

# USB – types de transfert - bloc

Utilisé pour grande quantité de données, pas urgentes  
(exemple type : impression)

Aucune garantie de temps d'attente, mais par contre  
garantie de livraison (détection d'erreurs et reprise)

Là encore, pleine et haute vitesse uniquement.

# USB – bande passante du bus

L'hôte gère le bus en allouant la plus grande partie de la bande passante aux transferts isochrones et d'interruption.

Il garde une partie pour les transferts de commande et ... le reste pour les transferts de blocs.

# USB – Installation

Tout appareil doit répondre aux paquets d'installation.

Un "appareil standard" doit comprendre 8 requêtes différentes : demande d'état, mise en place ou arrêt d'une fonction, affectation d'une adresse, lecture ou écriture d'un descripteur ou d'une configuration.

Par ailleurs chaque terminaison peut également être configurée.

# USB – Enumération (1)

Cette fonction est essentielle pour le "plug 'n play".

Lorsqu'on branche un appareil sur un port d'un hub, ce dernier s'en aperçoit à cause de tension sur les fils de données du bus. Il indiquera cet événement à l'hôte à la prochaine interrogation d'état (du hub) par l'hôte.

L'hôte pourra alors obtenir des infos plus précises et commander l'activation du port.

# USB – Enumération (2)

L'appareil branché doit accepter les messages envoyés à l'adresse 0. L'hôte assigne une adresse à l'appareil et la lui envoie.

L'appareil doit alors répondre à cette nouvelle adresse, sur son canal par défaut.

La phase de configuration peut alors avoir lieu.

# USB – Enumération (3)

Lorsque l'appareil est débranché, le hub s'en aperçoit et prépare l'information correspondante pour l'hôte.

A sa réception, ce dernier retirera alors l'appareil de sa liste, désactivera le port du hub.