



UNIVERSITE D'ANTANANARIVO
ECOLE SUPERIEURE POLYTECHNIQUE
DEPARTEMENT GENIE MECANIQUE ET PRODUCTIQUE
DEPARTEMENT GENIE ELECTRIQUE
UFR : GENIE INDUSTRIEL

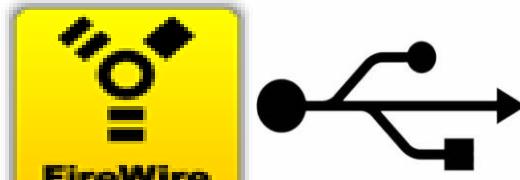
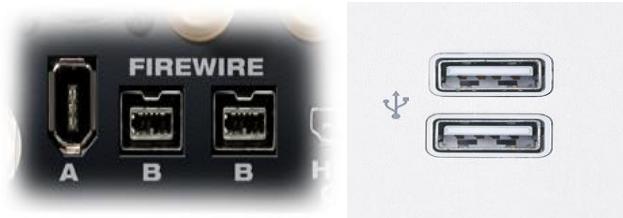


MEMOIRE DE FIN D'ETUDES
en vue de l'obtention
du DIPLÔME de LICENCE en GENIE ELECTRIQUE

N° d'ordre

Intitulé

« COMPARAISON DES PORTS USB ET FIREWIRE »



Présenté par :

- RAKOTOARIMANANA Avotra Mahefarivo
- RAKOTOARISOA ANDRIAMANANA Narisoa Olivier

Directeur de mémoire : Monsieur RAVALOMANANA

Date de soutenance : 25 Mars 2015

Lieu de soutenance : bloc technique Vontovorona

Promotion 2014



UNIVERSITE D'ANTANANARIVO
ECOLE SUPERIEURE POLYTECHNIQUE
DEPARTEMENT GENIE MECANIQUE ET PRODUCTIQUE
DEPARTEMENT GENIE ELECTRIQUE
UFR : GENIE INDUSTRIEL



MEMOIRE DE FIN D'ETUDES
en vue de l'obtention
du DIPLÔME de LICENCE en GENIE ELECTRIQUE

N° d'ordre

Intitulé

« COMPARAISON DES PORTS USB ET FIREWIRE »

Présenté par :

- RAKOTOARIMANANA Avotra Mahefarivo
- RAKOTOARISOA ANDRIAMANANA Narisoa Olivier

Directeur de mémoire : Monsieur RAVALOMANANA

Enseignant Chercheur à l'ESPA

Président du jury : Monsieur RAKOTONIAINA Solofo Hery

Maître de conférences à l'ESPA

Membres du jury :

- ✚ Monsieur RAMAROZATOVO Vonjy
Maître de conférences à l'ESPA
- ✚ Monsieur RALAIVAO Haja
Enseignant Chercheur à l'ESPA

Promotion 2014

REMERCIEMENTS

Nous rendons grâce à Dieu tout puissant pour son amour et sa bonté, de nous avoir donné la force et la santé durant la réalisation de ce mémoire de fin d'études.

Nous adressons nos vifs remerciements aux personnes suivantes sans qui ce travail de mémoire n'aurait pas pu être réalisé :

- ⊕ Monsieur ANDRIANARY Philippe Antoine, Directeur de l'Ecole Supérieure Polytechnique d'Antananarivo qui nous a autorisé de passer trois années d'études afin d'obtenir le diplôme de licences es-sciences techniques en Génie Industriel.
- ⊕ Mr Grade, responsable de l'unité d'enseignement informatique
- ⊕ Monsieur , grade et qui a accepté de présider ce mémoire.
- ⊕ Monsieur RAKOTONIAINA Solofo Hery, Chef de département Génie Électrique, Maître de conférences à l'ESPA.
- ⊕ Monsieur RAKOTOMANANA Charles Rodin, Chef de département Génie Mécanique et Productique.
- ⊕ Monsieur RAVALOMANANA, Enseignant chercheur à l'ESPA et Directeur de mémoire qui, malgré ses préoccupations, a accepté de nous encadrer pédagogiquement, de nous donner des conseils, et qui nous a beaucoup aidé pour l'achèvement de ce présent mémoire.
- ⊕ Messieurs les membres de jury, , Enseignants Chercheurs au sein de la filière Génie Electrique et de l'Ecole.
- ⊕ Tous les enseignants de la filière Génie Electrique, de l'Ecole Supérieure Polytechnique d'Antananarivo pour toutes les connaissances qu'ils nous ont fournies durant notre formation.
- ⊕ Tous les personnels de l'Ecole Supérieure Polytechnique d'Antananarivo.
- ⊕ Toute notre famille pour leurs sacrifices et leurs soutiens bienveillants durant ces longues années afin qu'on puisse arriver à ce niveau.
- ⊕ Enfin tous ceux qui de loin ou de près ont contribué à la réalisation de ce travail.

QUE DIEU VOUS BENISSE !!!

TABLES DES MATIERES

REMERCIEMENTS	I
TABLES DES MATIERES	II
LISTE DES FIGURES.....	V
LISTE DES TABLEAUX	VI
INRODUCTION GENERALE	1
CHAPITRE 1. L'ETUDE THEORIQUE DU PORT UNIVERSAL SERIAL BUS (USB)	2
1.1 GENERALITES SUR L'USB	3
<i>1.1.1 Introduction</i>	<i>3</i>
<i>1.1.2 Origine de l'USB.....</i>	<i>3</i>
<i>1.1.3 La naissance et l'évolution de la norme USB.....</i>	<i>4</i>
<i>1.1.4 Facilité d'utilisation.....</i>	<i>4</i>
1.2 VITESSE DE TRANSFERT DE L'USB.....	7
<i>1.2.1 Définitions des différentes vitesses.....</i>	<i>7</i>
<i>1.2.2 Raison des trois vitesses de l'USB.....</i>	<i>7</i>
<i>1.2.3 Domaine d'utilisation des différentes vitesses</i>	<i>8</i>
<i>1.2.4 Les débits de l'USB.....</i>	<i>8</i>
1.3 LE MATERIEL	8
<i>1.3.1 Les connecteurs</i>	<i>8</i>
<i>1.3.2 L'électrique</i>	<i>11</i>
<i>1.3.3 Identification de la vitesse</i>	<i>12</i>
<i>1.3.4 Alimentation (VBUS).....</i>	<i>13</i>
<i>1.3.5 Courant de veille</i>	<i>14</i>
<i>1.3.6 Accès au mode veille</i>	<i>15</i>
<i>1.3.7 Taux de transfert de données</i>	<i>15</i>
1.4 LE BUS USB PROPREMENT DITE.....	16
<i>1.4.1 Principe du bus USB.....</i>	<i>16</i>
<i>1.4.2 Topologie du Bus USB</i>	<i>16</i>
<i>1.4.3 Protocole USB.....</i>	<i>17</i>

1.4.4 Type de paquet USB.....	18
1.5 NOTIONS IMPORTANTES SUR L'USB.....	20
1.5.1 Architectures de transmissions des données (entrées/sorties).....	20
1.5.2 Les terminaisons (ou les types de transferts)	21
1.5.3 Les descripteurs USB.....	24
1.5.4 L'énumération	26
CHAPITRE 2. LA TECHNOLOGIE DU PORT FIREWIRE (IEEE 1394 ou I.LINK)	29
2.1 LA THEORIE DU BUS FIREWIRE	30
2.1.1 Historique.....	30
2.1.2 Les normes FireWire	30
2.1.3 Le connecteur FireWire.....	31
2.1.4 Brochage du FireWire	32
2.1.5 Architecture.....	32
2.1.6 Caractéristiques techniques.....	33
2.2 FONCTIONNEMENT DU BUS FIREWIRE	33
2.2.1 La topologie.....	33
2.2.2 Les types de transferts.....	34
2.2.3 Modes de transmissions	34
2.3 CARACTERISTIQUES	36
2.3.1 Le protocole.....	36
2.3.2 La couche physique	37
2.3.3 L'arbitrage	38
2.3.4 La Couche de Lien (Link Layer).....	39
2.3.5 La Couche de Transaction	40
CHAPITRE 3. LA COMPARAISON DES DEUX PORTS.....	41
3.1 INTRODUCTION.....	42
3.2 LES DIFFERENCES ENTRE USB ET FIREWIRE	42
3.3 CES AVANTAGES.....	44

3.3.1 Le port USB.....	44
3.3.2 Le bus FireWire	45
3.4 CES INCONVENIENTS	46
 3.4.1 De l'USB	<i> Erreur ! Signet non défini.</i>
 3.4.2 Du FireWire	<i> Erreur ! Signet non défini.</i>
3.5 EXEMPLE DE COMPARAISON ENTRE USB2.0 ET IEEE 1394	46
 3.5.1 Performance.....	46
 3.5.2 Tableau comparatif.....	47
3.6 CONCLUSION	<i> Erreur ! Signet non défini.</i>
CONCLUSION GENERALE	48
REFERENCES BIBLIOGRAPHIQUES	A

LISTE DES FIGURES

<i>Figure 1 : Symbole d'un USB</i>	3
<i>Figure 2 : Architecture de devices USBs avec HUB.....</i>	6
<i>Figure 3 : (1) le connecteur « type A » et (2) le connecteur « type B »</i>	9
<i>Figure 4 : Les mini-connecteurs A et B.....</i>	10
<i>Figure 5 : Les micro-connecteurs A et B</i>	11
<i>Figure 6 : Appareil pleine vitesse avec résistance de rappel état haut branché sur D+.....</i>	13
<i>Figure 7 : Appareil basse vitesse avec résistance de rappel état haut branché sur D-.....</i>	13
<i>Figure 8 : La topologie du bus USB</i>	17
<i>Figure 9 : Exemple de branchement respectant la topologie du Bus USB</i>	17
<i>Figure 10 : Le protocole USB</i>	18
<i>Figure 11 : Format du paquet Jeton</i>	19
<i>Figure 12 : Format du paquet Data.....</i>	19
<i>Figure 13 : Format du paquet Handshake</i>	20
<i>Figure 14 : Format du paquet SOF</i>	20
<i>Figure 15 : Notion de l'E/S.....</i>	21
<i>Figure 16 : Diagramme hiérarchique des descripteurs</i>	26
<i>Figure 17 : Symbole du FireWire</i>	30
<i>Figure 18 : Type 1394a-1995.....</i>	31
<i>Figure 19 : Type 1394a-2000.....</i>	31
<i>Figure 20 : Type 1394b Bêta.....</i>	31
<i>Figure 21 : Type 1394b Bilingual</i>	31
<i>Figure 22 : Architecture de l'IEEE-1394</i>	32
<i>Figure 23 : Topologie du FireWire</i>	34
<i>Figure 24 : Modèle de communication</i>	36
<i>Figure 25 : Le protocole FireWire.....</i>	37
<i>Figure 26 : La couche de Lien</i>	39

LISTE DES TABLEAUX

<i>Tableau 1 : Débits des normes USB.....</i>	8
<i>Tableau 2 : Caractéristiques des 4 types de transfert.....</i>	24
<i>Tableau 3 : Descriptions des descripteurs USB.....</i>	25
<i>Tableau 4 : Brochages</i>	32
<i>Tableau 5 : les différences entre l'USB et le FireWire</i>	44
<i>Tableau 6 : Récapitulation de la comparaison de l'USB2.0 et le FireWire</i>	47

NOTATIONS
ABREVIATIONS

ACK	ACKnowledge; Validation
EOP	End Of Packet
FPGA	Field Programmable GateArray
IEEE	Institute of Electrical and Electronics Engineers
IRP	Input-output Request Packet
IRQ	Interrupt Request
NAK	No AcKnowledge; Pas de validation
NRZI	Non Return to Zero Invert.
PID	Packet ID
Ping	S'assure d'une bonne connexion
PREamble	Synchronisation initiale
SETUP	Configuration
SOF	Start Of Frame
Split	Partager
STALL	Bloqué
USB	Useless Serial Bus

INTRODUCTION GENERALE

La technologie de l'information à propos des transferts de données ne cesse de croître depuis quelques années. Des concepteurs consacrent leurs recherches pour apporter les meilleurs matériels dans la cadre des ports et des périphériques comme sur les ordinateurs, camera, appareils photos, ...

En informatique, la transmission et échanges des paquets ou des fichiers de données nécessitent les ports USB, FIREWIRE et d'autres pour les faire véhiculer via câble. D'ailleurs, les deux ports USB et FIREWIRE ont leurs fonctions spécifiques et conditions d'adaptation sur les outils qui les utilisent. Les normes, les protocoles aussi font parties des particularités de ces ports.

La connaissance de la diversité de l'USB et FIREWIRE nous apportes à la comparaison des deux et donne ainsi des résultats concluants. C'est aussi de bien connaître, d'enrichir et de se développer dans ce domaine. Ceci nous a conduits à entreprendre ce mémoire qui s'intitule : « **CONCEPTION ET COMPARAISON DES PORTS USB ET FIREWIRE** ».

Pour mieux cerner le sujet, on a divisé cet ouvrage en trois grands chapitres :

- ✚ Le premier chapitre rappelle l'étude théorique du port Universal Serial Bus ou USB
- ✚ Le deuxième chapitre concerne la technologie du port FIREWIRE (IEEE 1394 ou I.LINK)
- ✚ Le dernier chapitre illustre la comparaison des deux ports

CHAPITRE 1. L'ETUDE THEORIQUE DU PORT UNIVERSAL SERIAL BUS (USB)

1.1 GENERALITES SUR L'USB

1.1.1 *Introduction*

La démocratisation d'appareils numériques grand public nécessitant des taux de transfert soutenus (caméscope, scanner, appareils photos numériques, etc...), a obligé les constructeurs à développer des interfaces plus rapides, plus universelles tout en pouvant y raccorder un maximum de ces périphériques hétéroclites.

Nous allons étudier ici l'interface USB qui représente une nouvelle liaison série synchrone. Il a été conçu pour remplacer les nombreux ports externes d'ordinateur lents et incompatibles, c'est le successeur des antiques liaisons **RS-232** et **parallèle**, mais avec des performances nettement meilleures...

La technologie d'interface série actuelle (horloge à cadence plus élevée) permet une transmission plus rapide que par le passé. Le bus USB est maintenant un standard sur tous les PC, et est généralement utilisé pour brancher les imprimantes, scanners, modems et de nombreux appareils stockant des données (disque dur, clé USB...).

1.1.2 *Origine de l'USB*

Tout au début, avant la standardisation il y avait des problèmes avec la compatibilité de l'USB, car chaque fabricant de carte mère inventait son propre protocole, il n'y avait pas encore de norme. C'est pourquoi on était très sceptique sur ce nouveau protocole à ses débuts. La dénomination USB qui est « **Universal Serial Bus** » a dérivée vers le nom « **Useless Serial Bus** ».



Figure 1 : Symbole d'un USB

Le bus USB est donc réellement né de l'alliance en 1994 de sept partenaires industriels (Compaq, DEC, IBM, Intel, Microsoft, NEC et Northern Telecom). C'est eux qui ont commencé à créer la norme USB.

Le bus USB a été conçu à l'origine pour faciliter les transferts de données en particulier définir une connectique « **universelle** » et « **Plug & Play** », utilisable aussi bien pour une souris que pour un modem ou un moniteur où l'on va voir ci-après.

Conçu également pour répondre au besoin d'intégration entre le monde du PC et celui du téléphone et enfin pour répondre au besoin d'extensions multiples en dehors du PC.

1.1.3 *La naissance et l'évolution de la norme USB*

La spécification ou norme USB version 1.0 opérationnelle est sorti, après plusieurs années de développement, en janvier 1996, les premiers produits ont vu le jour en fin 1997. A cette époque, le manque de composants USB a retardé la mise à jour de la norme. La version 1.0, est toute la première donc, n'est plus utilisée et n'a jamais été très utilisée, c'était en quelque sorte une version de test. En septembre 1998 paraît la dernière version officielle (version 1.1), c'est celle que l'on utilise actuellement. La version, complète, « officielle », est donc la **version 1.1**.

Puis en avril 2000, est sortie la spécification 2.0. Cette nouvelle norme supporte toutes les caractéristiques de l'USB 1.1 et optimise l'utilisation de la bande passante. Elle a été révisé en 2002, la **version 2.0** présentée ne s'applique actuellement qu'aux applications de type vidéo, télécom et transmission haute vitesse pour lesquelles les drivers voient le jour actuellement.

Et en 2008, c'est au tour de l'**USB 3.0** de voir ses spécifications publiées. Elle introduit des communications à vitesse supérieure ou « Super Speed ».

En l'absence de logo la meilleure façon de déterminer s'il s'agit de périphériques USB à bas ou haut débit est de consulter la documentation du produit dans la mesure où les connecteurs sont les mêmes.

1.1.4 *Facilité d'utilisation*

Le principal but du bus USB est la facilité d'utilisation qui se traduit suivant de multiples critères que voici :

a. *La souplesse sur la norme*

Tout d'abord on peut dire que le protocole USB est une norme très souple ; une interface unique suffit pour commander plusieurs types de périphériques, il suffit juste de posséder le bon driver.

b. *Le plug & play*

Un point fort de l'USB est sa configuration automatique, on l'appelle aussi le « **plug & play** ». Cela signifie que si l'utilisateur connecte un périphérique USB, Windows détecte automatiquement ce périphérique et charge le driver approprié s'il est disponible dans les fichiers de Windows.

Si ce n'est pas le cas, Windows demande d'installer le disque (CD-Rom) contenant ce driver afin qu'il le copie dans son répertoire et ceci se fait une seule fois. Lors du prochain branchement, le

périphérique USB recharge automatiquement son driver, cette étape est alors transparente pour l'utilisateur.

Il faut noter aussi qu'il n'est pas nécessaire, avec le protocole USB, de lancer un fichier d'installation ou de redémarrer (rebooter) le PC avant d'utiliser le périphérique. Il n'y a pas non plus dans le protocole USB à faire le choix de l'adresse du port comme par exemple pour une liaison série, une adresse dynamique est allouée à chaque fois que l'on branche un autre périphérique, la norme USB est capable de fournir 127 adresses dynamiques différentes (codée sur 7 bits) c'est pour cela qu'on entend toujours parler de brancher jusqu'à 127 périphériques (en théorie) sur le bus USB.

c. La série d'adresse et l'IRQ

Le PC alloue donc une série d'adresses de port et une commande d'interruption (**IRQ**) pour les interfaces USB.

Contrairement aux autres périphériques qui ne sont pas USB, pour chaque ajout d'un périphérique il faut lui associer une adresse de port, souvent un IRQ ainsi qu'une carte d'extension si le PC n'en possède plus ou pas assez. Avec les périphériques USB, tous ces problèmes sont inexistants.

d. Le HUB

Un atout majeur du bus USB est qu'il est très facile à connecter, on n'a pas besoin d'ouvrir le PC à chaque fois que l'on veut rajouter un périphérique, les anciens PC possèdent au minimum deux ports USB (de nos jours il y en a davantage). On peut aussi connecter un HUB sur un port déjà existant pour pouvoir connecter encore plus de périphériques USB.

Le HUB joue en quelque sorte le rôle de multiplexeur (connexion de plusieurs périphériques à un même câble) mais aussi de répétiteur, d'amplificateur, de contrôleur du signal et de fournisseur de courant. C'est un multiplieur de port USB. Actuellement certains moniteurs, claviers ... possèdent des **HUBs** intégrés. On peut, évidemment, reconnecter d'autres HUBs aux HUBs cité précédemment.

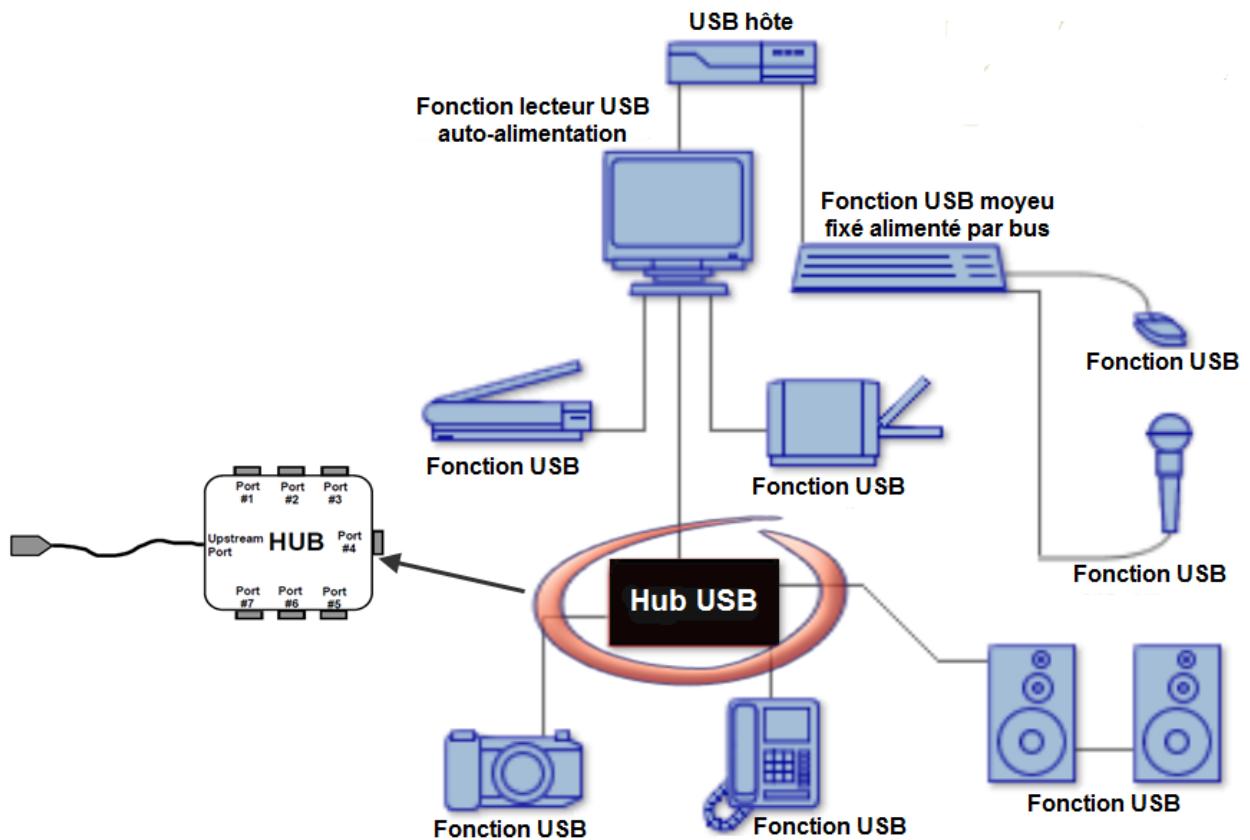


Figure 2 : Architecture de devices USBs avec HUB

e. Le Hot pluggable

Un autre avantage du bus USB est la connectique. En effet les câbles possèdent deux extrémités bien différentes pour ne pas se tromper lors du branchement, de plus les câbles USB sont très compacts comparé aux câbles pour les liaisons parallèles ou séries.

On dit également que le bus USB est « **Hot pluggable** », c'est à dire que l'on peut connecter et déconnecter un périphérique USB tout en ayant le PC allumé. L'Operating System installé sur le PC (Windows ...) le reconnaît ainsi immédiatement.

f. Le Power conservation

L'USB possède aussi une caractéristique très intéressante : c'est le mode veille lorsque l'on n'utilise pas le périphérique. On l'appelle aussi « **Power conservation** ». En effet le bus USB se met en suspend après 3 ms ou il n'est plus utilisé. Pendant ce mode, le composant ne consomme que $500\mu\text{A}$. Enfin, le dernier point fort pour l'USB est que cette norme permet d'alimenter le périphérique directement avec le PC. Il n'est pas nécessaire d'alimenter le périphérique avec une alimentation extérieure, le PC suffit.

1.2 VITESSE DE TRANSFERT DE L'USB

1.2.1 *Définitions des différentes vitesses*

D'après les différentes versions de l'USB cité ci-dessus, il existe en général 3 types de vitesses.

Pour le :

 **USB 1.1:** Pour cette version la vitesse se divise en deux

- Le mode lent (**Low Speed**) à 1.5 [Mbits/s]. C'est pour des périphériques transférant peu de données. Il est moins sujet aux perturbations Electro-Magnétique (EMI) réduisant ainsi le coût des composants utilisés. Les périphériques USB utilisent un quartz de 12 [MHz], pour un bon fonctionnement, la précision de cette fréquence est importante.
- Le mode rapide (**Full Speed**) à 12 [Mbit/s].

 **USB 2.0:** L'**High Speed** à 480 [Mbit/ s].

Il est 40 fois plus rapide que 'USB 1.1, il y a une compatibilité descendante partielle qui dépend des périphériques.

 **USB 3.0:** Le **Super Speed** à 4,8 [Gbit/s] (soit environ 600 [Mo/s]).

Cette nouvelle technologie aura une compatibilité descendante avec l'USB2.0 (mais pas avec l'USB1.1), mais on ne gagnera pas en performances avec des matériels anciens, bien entendu. A comparer à l'USB 2.0 : 60 Mo/s (480[Mbps]) soit un rapport de 10, il autorisera ainsi un transfert très rapide de masses de données de plus en plus grandes. Il mettra moins d'une minute à vider complètement les plus grosses cartes mémoires disponibles. L'USB 3.0 n'apportera guère de nouveautés au niveau connectique pour l'utilisateur.

1.2.2 *Raison des trois vitesses de l'USB*

Le Low Speed a été introduit pour deux raisons essentielles, le premier est que les périphériques USB ne sont pas cher, c'est à dire que tout le monde peut se le permettre. La deuxième raison est pour avoir des souris USB un peu plus pratiques. En effet les câbles USB pour le Low Speed n'ont pas besoin d'être blindés et, de ce fait, sont très souples. Le Full Speed a été conçu pour remplacer les liaisons séries et parallèles.

Le High Speed est une vitesse supplémentaire qui a été introduit lors de la mise à jour de la norme USB 2.0.

Ainsi, le Super Speed aussi c'est pour permettre de mettre en valeur la puissance du bus USB.

1.2.3 *Domaine d'utilisation des différentes vitesses*

Les applications Low Speed concernent essentiellement des périphériques interactifs (claviers, souris, consoles), mais aussi des afficheurs, des lecteurs (de carte à puce) et des applications en automatismes (mesure, capteurs) appelées à se développer.

En mode Full Speed on va trouver la téléphonie, les modems, les disques, les imprimantes, les fax ainsi que les scanners, certains lecteurs de carte à puce et le domaine multimédia (jeux, audio, vidéo limitée).

Et pour les 2 autres (High Speed et le Super Speed), il peut s'appliquer à tous les domaines cités précédemment et voir plus. De plus, ces deux derniers c'est pour la raison de puissance au niveau du transfert et la mise à jour des normes correspondants aux différents nouveaux appareils actuels.

1.2.4 *Les débits de l'USB*

Il faut noter que dans la norme USB le débit n'est pas proportionnel à la vitesse. L'USB « Low Speed » qui est consacré aux périphériques soit disant lents est limités à échanger aux maximum 8 octets toutes les 10 [ms], ce qui correspond à un débit maximum de 800 [octets/sec] soit 6400 [bits/s]. Par certains moyens détournés il est possible d'espérer atteindre 8 [Ko/s], mais c'est sans garantie.

L'USB « Full Speed » qui est consacré aux HUBs et aux périphériques non lents peut échanger jusqu'à 1024 octets toutes les ms, soit un débit de [1 Mo/s].

Version	USB 1.0	USB 1.1	USB 2.0	Wireless USB	USB 3.0	USB 3.1
Année	1996	1998	2000	2005	2008	2013
Débit	1,5 Mbit/s 0,19 Mo/s	12 Mbit/s 1,5 Mo/s	480 Mbit/s 60 Mo/s	480 Mbit/s 60 Mo/s	5 Gbit/s 600 Mo/s	10 Gbit/s 1,2 Go/s

Tableau 1 : Débits des normes USB

1.3 LE MATERIEL

1.3.1 *Les connecteurs*

a. *Type A et type B*

Tous les appareils ont une connexion amont vers l'hôte et tous les hôtes ont une connexion aval vers l'appareil. Les connecteurs amont et aval ne sont pas interchangeables mécaniquement, éliminant

ainsi les connexions de rebouclage interdite aux Hubs comme pour un port aval connecté à un port aval.

Il y a généralement 2 types de connecteurs, appelé **type A** dont la forme est rectangulaire et servant généralement pour des périphériques peu gourmands en bande passante (clavier, souris, webcam, etc.) et le **type B** dont la forme est carrée et utilisés principalement pour des périphériques à haut débit (disques durs externes, etc.) présenté ci-dessous.



Figure 3 : (1) le connecteur « type A » et (2) le connecteur « type B »

Les prises mâles de type A sont toujours tournés vers l'Amont. Les prises femelles de type A se trouveront généralement sur les hôtes et les Hubs. Par exemple, les prises femelles de type A sont courantes sur les cartes mères des ordinateurs et les Hubs. Les prises mâles de type B sont toujours connecté vers l'aval et par conséquent les prises femelles de type B se trouvent sur les appareils.

Il peut être intéressant de trouver des câbles de type A vers type A avec un câblage direct et une matrice de changeur de genre USB dans certains magasins d'ordinateurs.

C'est en contradiction avec la spécification USB. Les seuls appareils avec prise type A vers prise type A sont des ponts que l'on utilise pour brancher 2 ordinateurs entre eux. D'autres câbles prohibés sont les extensions

USB qui ont une prise mâle à une extrémité (soit de type A ou de type B) et une prise femelle à l'autre. Ces câbles ne respectent pas les exigences de longueur de câble de l'USB.

Numéro de broches	Couleurs des câbles	Fonction
1	Rouge	VBUS (5 volts)
2	Blanc	D-
3	Vert	D+
4	Noir	Noir Masse

Tableau 1.01: Les brochages des connecteurs

Nous allons déjà voir le cas de l'USB 1.1 et 2.0 de plus près car c'est en fait l'origine :

Alors, en 1 nous avons la borne **+5[V], 100[mA]** maximum

- En **2, D-**, qui permet de transférer les données.
- En **3, D+**, qui permet également de transférer les données.
- En **4, Ground**, c'est-à-dire le **0[V]**.

On utilise des couleurs standards pour les fils intérieurs des câbles USB de façon à faciliter l'identification des fils d'un constructeur à un autre. La normalisation précise les différents paramètres électriques pour les câbles. Il peut être intéressant de lire en détail la documentation sur l'USB 1.0 d'origine qui est incluse.

b. Les mini et micro-connecteurs

Chaque type (A ou B) existait dans les deux genres (mâle ou femelle), ce qui fait qu'il existait au départ quatre connecteurs. En octobre 2000, devant le développement des appareils compacts (téléphones portables, appareils photo numériques), une mise à jour de la norme USB 2 introduit une version miniature du connecteur B :

- Le **mini-B**. Elle est fonctionnellement équivalente au connecteur B, mais de dimensions nettement réduites.

En décembre 2001, l'USB 2 introduit le connecteur **mini-AB**, utilisé dans le cadre de l'extension « *On-The-Go* » qui ajoute la fonctionnalité pair à pair (peer to peer) à l'USB. Il permet aux appareils compatibles de jouer indifféremment le rôle d'hôte ou celui de périphérique, contrairement à l'USB classique où l'hôte se branche obligatoirement sur un connecteur de type A et le périphérique sur un connecteur de type B.



Figure 4 : Les mini-connecteurs A et B

Les appareils mobiles s'étant encore réduits, les connecteurs mini-B sont devenus à leur tour trop gros. En janvier 2007 :

- Le nouveau connecteur **micro-B** est annoncé. Il est non seulement plus fin que le mini-B, mais également prévu pour supporter un grand nombre de cycles de connexion/déconnexion (jusqu'à 10 000¹¹), ce qui le rend particulièrement bien adapté aux appareils mobiles souvent branchés/débranchés (tablettes tactiles, smartphones, etc.). Pour les mêmes raisons, en avril

2007, une nouvelle norme micro-AB vient remplacer la norme mini-AB, qui est officiellement dépréciée le mois suivant.



Figure 5 : Les micro-connecteurs A et B

c. La nouvelle norme : Le type C

Un nouveau connecteur est introduit dans la norme le 11 août 2014 : le **type C**, destiné à remplacer tous les connecteurs précédents. Il a la particularité d'être réversible, c'est-à-dire qu'il n'a plus de sens haut/bas. Outre l'aspect pratique, il est compatible à la fois avec le standard USB 3.1 (qui porte le débit maximal théorique à 10 Gbit/s) et l'USB Power Delivery (jusqu'à 100 watts dans les deux sens).

1.3.2 *L'électrique*

A moins qu'on ne concevait le silicium pour un appareil ou émetteur / récepteur USB ou bien pour un hôte ou Hub USB, il n'y a pas grand-chose à retenir des particularités électriques. On va préciser brièvement les points essentiels ici-même :

Comme nous le savons déjà, l'USB utilise une paire de transmission différentielle (cf annexe) pour les données. Celle-ci étant codé en utilisant le NRZI et est garni de bits pour assurer les transitions adéquates dans le flot de données.

Sur les appareils à vitesse basse et pleine un '1' différentiel est transmis en mettant **D+** au-dessus de 2,8[V] grâce à une résistance de 15[kΩ] relié à la masse et **D-** en dessous de 0,3[V] avec une résistance de 1,5k ohms relié à 3,6[V]. D'autre part un différentiel '0' correspond à D- plus grand que 2,8[V] et D+ inférieur à 0,3[V] avec les mêmes résistances de rappel état haut/bas adéquates.

Le récepteur défini un différentiel '1' avec D+ plus grand de 200[mV] que D- et un différentiel '0' avec D+ plus petit de 200[mV] que D-. La polarité du signal est inversée en fonction de la vitesse du BUS. En conséquence les états référencés par les termes 'J' et 'K' sont utilisés pour signifier les niveaux logiques. En vitesse basse, un état 'J' ai un différentiel '0'. En vitesse haute, un état 'J' ai un différentiel '1'.

Les émetteurs / récepteurs USB comprendront à la fois des sorties différentielles et uniques (non complémentaires). Certains états de BUS USB sont indiqués par des signaux à sorties uniques

(single ended zero ou SE0) sur D+, D- ou les deux. Par exemple un zéro à sorties uniques ou SE0 peut être utilisé pour signifier la réinitialisation d'un appareil s'il est maintenu plus de 10[ms]. On génère un SE0 en maintenant D+ et D- en position basse (inférieur à 0,3[V]). Les sorties uniques et différentielles sont importantes d'être notées si vous utilisez un émetteurs / récepteurs et un FPGA comme appareil USB. Vous ne pouvez pas vous contenter simplement d'échantillonner la sortie différentielle.

Le BUS basse et pleine vitesse a une impédance caractéristique de $90[\Omega]$ +/-15%. Il est donc important d'observer la documentation technique lorsque vous sélectionnez les résistances des caractéristiques électriques séries pour D+ et D- afin d'équilibrer l'impédance. Toute documentation technique devrait spécifier ces valeurs et tolérances.

Le mode vitesse haute (480 Mbits/s) utilise un courant constant de 17,78[mA] pour demander de réduire le bruit.

1.3.3 *Identification de la vitesse*

Un appareil USB doit indiquer sa vitesse en mettant soit D+ ou D- à 3,3[V]. Un appareil pleine vitesse, représenté plus bas utilisera une résistance de rappel rattaché à D+ pour se signaler comme tel. Ces résistances de rappel à l'extrémité de l'appareil seraient aussi utilisés par l'hôte ou Hub pour détecter la présence d'un appareil connecté à son port. Sans résistance de rappel, l'USB suppose qu'il n'y a rien de connecté au BUS. Certains appareils possèdent cette résistance intégré sur le silicium, pouvant être connecté ou non sous commande micro-programmée, d'autres exigent une résistance externe.

Par exemple Philips Semi-conducteur a une technologie Soft connectTM. Lors d'une première connexion au BUS, elle permet au microcontrôleur d'initialiser la fonction USB de l'appareil avant de valider la résistance de rappel d'identification de vitesse, indiquant ainsi qu'un appareil est relié au BUS.

Si la résistance de rappel était connectée à VBUS, alors celle-ci indiquerait qu'un appareil est branché au BUS dès que la prise est insérée. L'hôte peut alors tenter de réinitialiser l'appareil et demander un descripteur au moment où le microprocesseur n'a pas encore commencé à initialiser la fonction USB de l'appareil.

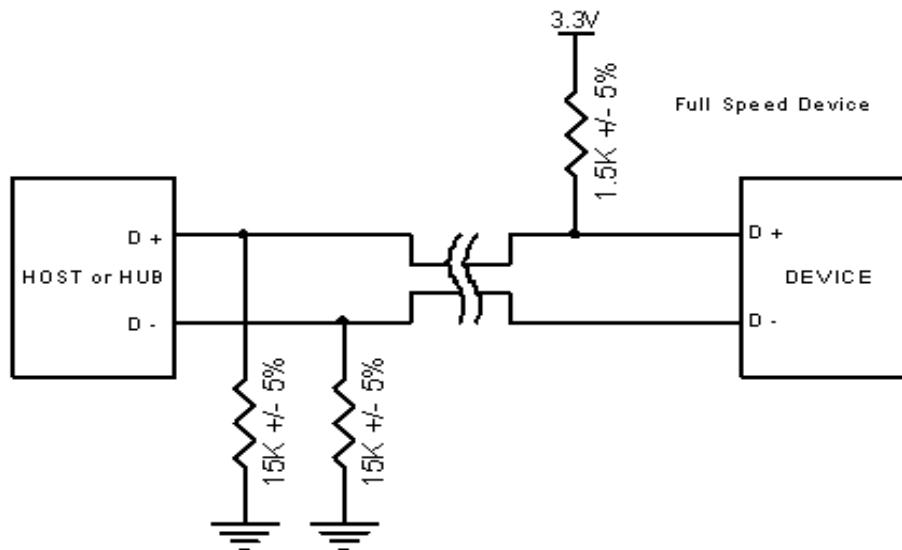


Figure 6 : Appareil pleine vitesse avec résistance de rappel état haut branché sur D+

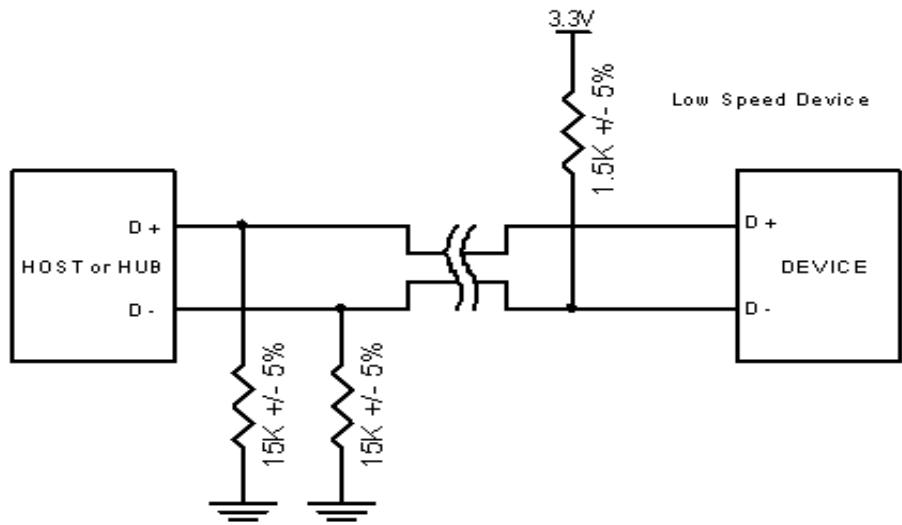


Figure 7 : Appareil basse vitesse avec résistance de rappel état haut branché sur D-

1.3.4 Alimentation (VBUS)

Un des avantages de l'USB réside dans ces appareils alimentés par le Bus. Ceux-ci obtiennent leur alimentation à partir du Bus et ne demande aucune prise externe et câble additionnel.

Cependant beaucoup de gens se focalisent sur cette option sans prendre en compte au préalable tous les critères nécessaires.

Un appareil USB précise sa consommation électrique exprimé en unité de 2[mA] dans le descripteur de la configuration que nous examinerons plus tard. Un appareil ne peut pas augmenter sa consommation électrique plus qu'il n'est précisé pendant l'énumération, même s'il perd de la puissance externe. Il existe 3 classes de fonctions USB,

- Les fonctions alimentées par le Bus à basse puissance (Low-Power)
- Les fonctions alimentées par le Bus à haute puissance (High-Power)
- Les fonctions auto alimentées (Self-powered)

Les fonctions alimentés par le Bus à basse puissance tire toute leur puissance de VBUS et ne peuvent en tirer que la charge d'une unité. La spécification USB défini une charge d'unité à 100[mA]. Les fonctions alimentées par le Bus à basse puissance peuvent aussi être conçus pour travailler à une tension de VBUS tombant à 4,4[V] et montant à un maximum de 5,25[V] mesuré à la prise amont de l'appareil. Pour beaucoup d'appareil 3,3[V], des régulateurs LDO sont obligatoires.

Les fonctions auto alimentées peuvent tirer jusqu'à une unité et faire dériver le reste de leur alimentation d'une source extérieure.

Si cette source extérieure venait à manquer, il doit y avoir des réserves en place de manière à ne pas tirer plus d'une unité de charge du Bus. Les fonctions auto alimentées sont plus faciles à concevoir au niveau de la spécification car il ne peut guère y avoir de problèmes en ce qui concerne l'alimentation électrique. La charge alimentée par le Bus à une seule unité permet la détection de l'énumération d'appareil sans avoir besoin d'une alimentation principale/secondaire du secteur.

Les autres considérations de VBUS sont l'appel de courant qui doit être limité. L'appel de courant est soutenu au niveau de la capacité de tête de votre appareil entre VBUS et la masse. Les spécifications précisent par conséquent que la capacité de découplage maximum que vous pouvez avoir sur votre appareil est de 10[μ F]. Quand vous déconnectez l'appareil après que le courant soit passé par le câble USB inductif, une grande tension de retour peut se produire sur l'extrémité ouverte du câble. Pour éviter ceci, on recommande une capacité de découplage VBUS minimum de 1[μ F].

1.3.5 *Courant de veille*

Le mode Veille est obligatoire sur tous les appareils. Pendant son temps d'action, d'autres contraintes surviennent. Le courant maximum de veille est proportionnel à l'unité de charge. Pour un appareil d'une unité de charge (par défaut) le courant de veille maximum est de 500[μ A]. Ceci comprend le courant dû aux résistances de rappel sur le Bus. Au niveau du Hub, D- et D+ possèdent des résistances de rappel niveau bas de 15 [k Ω]. Pour des raisons de consommation électrique, la résistance de rappel niveau bas de l'appareil est montée en série avec la résistance de rappel niveau haut de 1,5[k Ω], totalisant ainsi une charge de 16,5[k Ω] sur VTERM habituel de 3,3[V]. Par conséquent cette résistance draine un courant de 200[μ A] avant même que l'on commence.

Une autre considération pour bon nombre d'appareils est le régulateur de 3,3V. Un grand nombre d'appareil USB fonctionne sur 3,3[V]. Le PDIUSBD11 en est un exemple. Les régulateurs linéaires

sont habituellement tout à fait inefficaces avec des courants de repos moyen de l'ordre de 600[μ A], par conséquent on aura besoin de régulateurs plus efficace et donc plus cher. Dans la majorité des cas, on doit ralentir ou arrêter les horloges pour que la consommation du microcontrôleur tombe en dessous de la limite de 500[μ A].

Toutefois en fonctionnement normal, si on essaye de dépasser les 100[mA] ou la charge permise qui vous est indiqué, alors le Hub ou Hôte le détecte et déconnecte l'appareil, dans l'intérêt de l'intégrité du Bus.

1.3.6 *Accès au mode veille*

Un appareil USB entrera en veille lorsqu'il n'y a aucune activité sur le Bus pendant plus de 3[ms]. Il dispose ensuite de 7[ms] de plus pour éteindre l'appareil et ne prendre que le courant de veille désigné, ne prenant ainsi que le courant de veille nominal à partir du Bus 10[ms] après que l'activité du Bus ce soit arrêté. Afin de le maintenir connecté à un Hub ou à un Hôte mis en veille, l'appareil doit encore fournir de l'alimentation à ces résistances de rappel de sélection de vitesse pendant le mode veille.

L'USB possède un démarrage de trames de bits ou bien d'entretien qui sont envoyés périodiquement sur le Bus. Ceci empêche un Bus inutilisé d'entrer dans le mode veille en l'absence de données. Comme ci-dessous :

- ✚ Un Bus haute vitesse enverra une trame toutes les 125.0 [μ s] \pm 62.5 [ns].
- ✚ Un Bus pleine vitesse enverra une trame toutes les 1.000 [ms] \pm 500 [ns].
- ✚ Un Bus basse vitesse aura un dispositif d'entretien qui est un EOP (End Of Packet ou Fin De Paquet) toutes les 1[ms] simplement en l'absence de données basse vitesse.

Le terme veille Global (Global Suspend) est utilisé lorsque le Bus USB entier entre collectivement dans le mode veille. Cependant les appareils sélectionnés peuvent être mis en veille en ordonnant au Hub sur lequel l'appareil est aussi connecté. On fait référence à cette opération comme mode "veille sélective". L'appareil reprendra son fonctionnement quand il recevra tout signal qui n'est pas en attente. Si un appareil possède une mise en service de réveil retardé, alors il devra signaler à l'Hôte de reprendre à partir du mode veille.

1.3.7 *Taux de transfert de données*

Un autre domaine qui est souvent négligé est la tolérance de l'horloge USB. Elle est précisée dans la spécification USB 7.1.11 :

- Les données haute vitesse sont cadencées à 480.00 [Mb/s] avec une tolérance de transmission de données de ± 500 ppm ;
- Les données pleine vitesse sont cadencés à 12.000 [Mb/s] avec une tolérance de transmission de données de ± 0.25 % ou 2,500 ppm ;
- Les données basse vitesse sont cadencées à 1.50 [Mb/s] avec une tolérance de transmission de données de ± 1.5 % ou 15,000 ppm.

Cela permet aux résonateurs d'être utilisés pour des appareils faibles vitesse et faible coût, mais d'être exclus pour des appareils pleins et hauts vitesse.

1.4 LE BUS USB PROPREMENT DITE

1.4.1 *Principe du bus USB*

Le bus USB est un bus fonctionnant sur la hiérarchie, commandé par un host unique. Le host utilise un protocole Maître/Esclave pour communiquer avec les périphériques USB. Cela signifie que c'est le host qui décide du transfert des données et que les différents périphériques ne peuvent pas établir de connexion entre eux tant que le maître n'a pas donné l'autorisation. On peut dire que cela peut être un inconvénient par rapport aux autres protocoles mais il ne faut pas oublier que l'USB a été conçu avec des compromis de coût et de performance. Le fait que le bus USB fonctionne avec le protocole Maître/Esclave résout implicitement ces problèmes comme par exemple les problèmes de collision ou d'arbitrage de périphériques.

1.4.2 *Topologie du Bus USB*

La topologie du port est une topologie « étoile série » ou encore « tiered star » qui tolère jusqu'à 5 niveaux de concentrateurs. La spécification limite le nombre de périphérique à 127. Un dispositif conforme à la norme 2.0 peut être relié à un HUB 1.1 mais dans ce cas le trafic sera celui défini par la norme 1.1. C'est toujours le même pour les autres normes. Les divers cas possibles sont précisés sur la figure ci-dessous. Notons qu'à l'initialisation d'un "device" (ou d'une fonction selon la terminologie utilisée par le groupement USB) celui-ci reçoit une adresse. Les transferts Full Speed déclenchés par le "host" parcourent tous les tronçons, mais seule la fonction adressée va évidemment répondre. Ainsi si l'il s'agit d'un ordre d'impression il sera reçu par tous les périphériques, mais seule l'imprimante le comprendra.

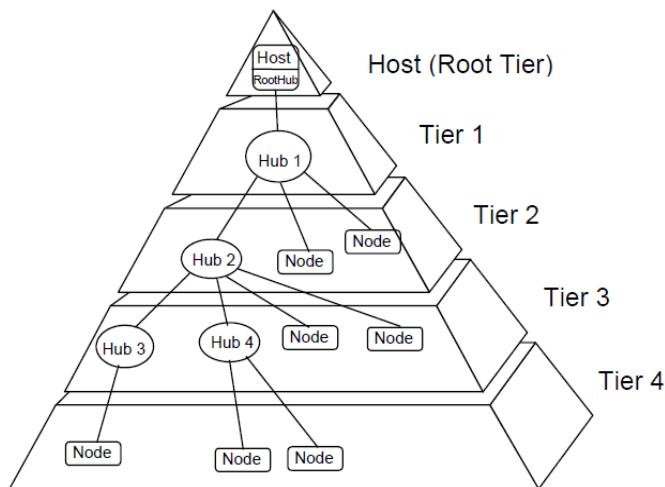


Figure 8 : La topologie du bus USB

Et voici en image un exemple de branchement avec la topologie étoile de l'USB.

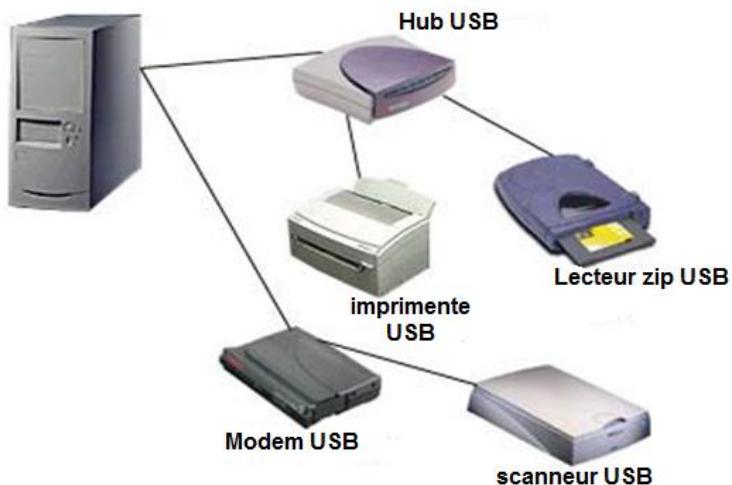


Figure 9 : Exemple de branchement respectant la topologie du Bus USB

1.4.3 Protocole USB

Le protocole USB est, comme tous les autres protocoles un protocole à encapsulation. Le fonctionnement de l'USB est présenté sous forme d'un système séparé en 3 couches (les 3 couches basses du modèle standard OSI). Cette représentation permet à chaque intervenant dans la réalisation d'un élément USB de bien délimiter son travail en respectant les spécifications attendues par les autres intervenants. Le synoptique ci-dessous traduit de façon imagée les liens entre les différents éléments.

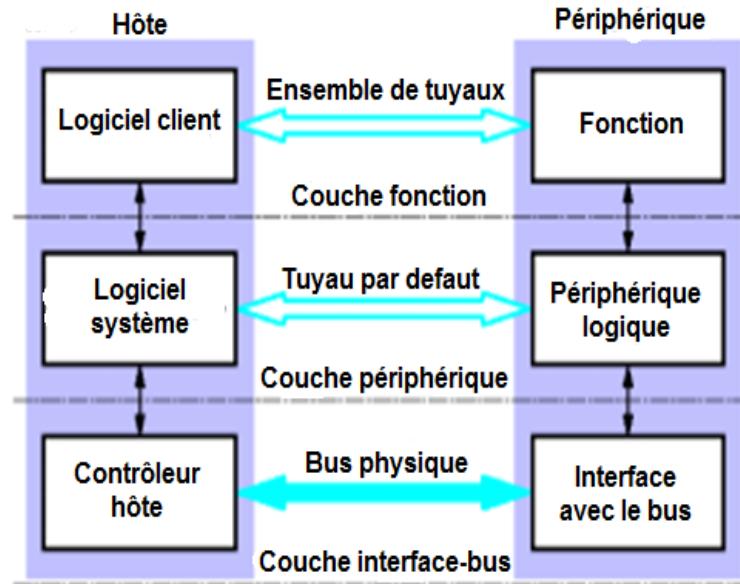


Figure 10 : Le protocole USB

Le protocole de communication est qualifié de « **host scheduled token based protocol** » : le host initie toutes les transactions. Le premier paquet est produit par le host pour décrire ce qui va suivre, si la transaction de données sera en lecture ou en écriture, ce que sera l'adresse de l'appareil et la terminaison (mode de transfert) désignée. Le dernier contient donc une grande partie de l'information sur la nature de la communication.

En détail, un client envoie au système USB une requête transformée en **IRP** (Input-output Request Packet) par le driver USB. Le host gère la liste des IRP (énumération). Chacune se concrétise par la succession d'une ou de plusieurs transactions (s'il faut fragmenter les données) sur le bus.

1.4.4 Type de paquet USB

Contrairement à la RS232 et des interfaces sérielles similaires où le format des données envoyées n'est pas défini, l'USB est composé de plusieurs couches de protocoles. Une fois qu'on comprit ce qui se passe, on doit uniquement inquiéter des couches supérieures. En fait la plupart des Circuits Intégrés contrôleur d'USB s'occuperont de la couche inférieure, la rendant ainsi presque invisible au regard du concepteur final. Les données sur le BUS USB sont transmises avec le bit LSB en premier.

L'USB a quatre types différents de paquet : les paquets jetons (Token) indiquent le type de la transaction qui va suivre, les paquets de données contiennent la charge utile, les paquets « Handshake » ou « poignée de mains » sont utilisés pour valider les données ou rapporter les erreurs et ainsi les paquets début de trame (SOF) indiquent le commencement d'une nouvelle trame.

a. Les paquets jetons

Il y a 3 sortes de paquets Jetons :

- **In** : Informe l'appareil USB que l'hôte veut lire des informations.
- **Out** : Informe l'appareil USB que l'hôte veut envoyer des informations.
- **Setup** : Utilisé pour commencer les transferts de command

Les paquets jetons doivent se conformer au format suivant :

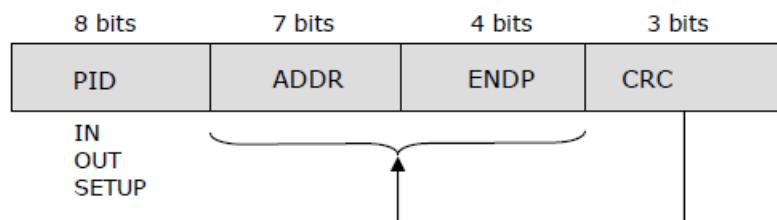


Figure 11 : Format du paquet Jeton

b. Les paquets de données

Il y a 2 sortes de paquets de données, chacun étant capable de transmettre plus de 1024 octets de données :

- Data0
- Data1

Le mode haute vitesse définit 2 autres PIDs de données, DATA2 et MDATA. Les paquets de données ont le format suivant :

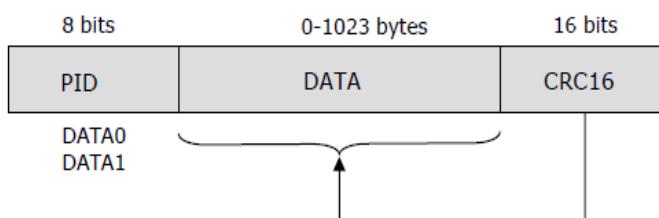


Figure 12 : Format du paquet Data

- La taille maximale de données " charge utile " pour les appareils basse vitesse est de 8 octets.
- La taille maximale de données " charge utile " pour les appareils pleine vitesse est de 1023 octets.
- La taille maximale de données " charge utile " pour les appareils haute vitesse est de 1024 octets.
- Les données doivent être envoyées en multiple d'octets.

c. Les paquets « Handshake »

Il y a 3 sortes de paquets " poignée de mains " qui font simplement partie du PID.

- ✚ **ACK** : validant que le paquet a été reçu correctement.
- ✚ **NAK** : rapporte que temporairement l'appareil ne peut ni envoyer ou recevoir des données. Aussi utilisé pendant les transactions d'interruptions pour avertir l'hôte qu'il n'a pas de données à envoyer.
- ✚ **STALL (Bloqué)** : L'appareil se retrouve dans un état qui va exiger l'intervention de l'hôte.

Les paquets " poignée de mains " ont le format suivant,

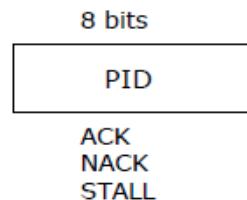


Figure 13 : Format du paquet Handshake

d. Les paquets SOF

Le paquet SOF composé d'une trame de 11 bits est envoyé par l'hôte toutes les $1\text{ms} \pm 500\text{ns}$ sur un bus Full Speed vitesse ou bien toutes les $125\mu\text{s} \pm 0,0625\mu\text{s}$ sur un bus High Speed.

Les paquets SOF ont le format suivant :

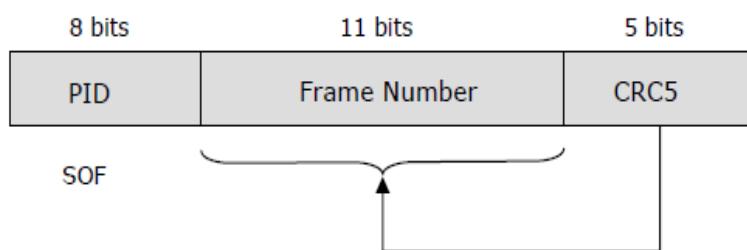


Figure 14 : Format du paquet SOF

1.5 NOTIONS IMPORTANTES SUR L'USB

1.5.1 Architectures de transmissions des données (entrées/sorties)

La fonction d'un ordinateur est le traitement de l'information (fonction réalisée au niveau de la mémoire et l'unité central ou UC). L'ordinateur acquiert cette information et restitue cette information au moyen d'entrées/sorties (E/S).

- ✚ Les échanges d'informations entre les périphériques, le processeur central, la mémoire centrale



Figure 15 : Notion de l'E/S

- ⊕ Les périphériques : dispositifs matériels permettant d'assurer les échanges d'informations en entrée et en sortie entre l'ordinateur et l'extérieur ou de stocker de manière permanente des informations comme clavier, souris, imprimantes, ...
- ⊕ Communication entre les modules du processeur et les périphériques :
 - Le but est d'adapter la diversité des périphériques (débit, tps de réponse, format des données, etc.) à une interface commune obéissant aux normes adoptées par le constructeur
 - Double nature : La communication avec la mémoire centrale et le microprocesseur c'est-à-dire au travers de bus dits bus d'extension (ISA, USB, PCI, ...) et la communication avec les périphériques ; le pilotage.

1.5.2 *Les terminaisons (ou les types de transferts)*

L'USB définit quatre types de transferts :

1.5.2.1 TRANSFERTS DE COMMANDE

Les Transferts de commande sont régulièrement utilisés pour les opérations de commande et d'état. Ils sont essentiels pour installer un appareil USB avec toutes les fonctions d'énumération qui seront exécutés en utilisant les Transferts de commande. Ils surviennent généralement en paquets directs et par rafales qui sont initiés par l'hôte et utilisent le meilleur rendement de livraison.

La longueur du paquet du Transfert de commande pour appareil basse vitesse doit être de 8 octets, les appareils pleine vitesse autorise une taille de paquet de 8, 16, 32 ou 64 octets et les appareils haute vitesse doivent avoir une taille de paquet de 64 octets.

Un Transfert de commande peut avoir plus de 3 étapes :

- ⊕ **L'étape d'installation** : il se passe lorsqu'une demande est envoyée. Elle est composée de 3 paquets. Le jeton (token) d'installation envoyé le premier est celui qui contient l'adresse et le numéro de la terminaison. Le paquet de données est envoyé après et a toujours un type PID de Data0 et inclut un paquet d'installation qui détaille le type de la demande. Nous détaillerons le paquet d'installation plus tard. Le dernier paquet est une poignée de mains utilisé pour valider la bonne réception ou pour indiquer une erreur. Si la fonction reçoit

correctement la donnée d'installation (CRC et PID etc...Ok) elle répond avec ACK, sinon elle ignore la donnée et n'envoie pas un paquet de poignée de mains. Les fonctions ne peuvent pas émettre un paquet STALL ou NAK en réponse à un paquet d'installation.

- ⊕ **L'étape de données facultative** qui consiste en un ou plusieurs transferts IN (Entrée) ou OUT (sorties). La demande d'installation indique la quantité de données qui doit être envoyée dans cette étape. Si elle dépasse la taille maximale du paquet, les données seront envoyées en plusieurs transferts, chacune ayant la longueur maximale du paquet à l'exception du dernier paquet.
- ⊕ **L'étape d'état** qui rend compte des états de l'ensemble de demandes et cette fois encore change selon la direction du transfert. Le rapport d'état est toujours réalisé par la fonction.

1.5.2.2 TRANSFERTS D'INTERRUPTION (OU INTERRUPT)

Le transfert par interruption est utilisé par les appareils ayant peu de données à transmettre, mais ayant des données qui doivent être transmises rapidement (exemple: clavier ou souris). Les interruptions sont détectées par polling (interrogation successives) de la part de l'hôte. La fréquence de polling requise est donnée par les descripteurs de l'appareil. Ce mode de transfert est également compatible avec le Low et Full Speed.

Ce type de transfert est une expérience de demandes d'interruption sur microcontrôleurs saura que les interruptions sont générées par l'appareil. Toutefois sous USB, si un appareil demande l'attention de l'hôte, il doit attendre que l'hôte l'interroge avant de signaler qu'il a besoin d'une attention urgente. Si une interruption a été mise en attente par l'appareil, la fonction enverra un paquet Data contenant des données ayant rapport à l'interruption quand il recevra le jeton IN. Sur des réceptions au niveau de l'hôte, celui-ci retournera un ACK. Toutefois si les données sont altérées, l'hôte ne mentionnera aucun état. Si, d'autre part, une condition d'interruption n'était pas.

1.5.2.3 TRANSFERTS ISOCHRONES

Le transfert isochrone est un mode de transfert pour lequel les données sont transmises à l'intérieur d'une trame et elles sont synchronisées avec les débuts de trames. Il est utilisé pour des transferts nécessitant un flux régulier de données comme par exemple les caméras ou les téléphones. Ce mode de transfert est uniquement compatible avec la vitesse Full USB.

Ils contiennent généralement des informations à durée de vie critique, tel des trains de données audio ou vidéo. S'il y avait un retard ou une reprise de données dans un flot de données audio, alors on

pourrait s'attendre à de l'audio par intermittence contenant des signaux transitoires. Le battement (rythme) ne serait plus synchronisé. Toutefois si un paquet ou une trame se perdait, il est vraisemblable que l'auditeur ne le remarquerait même pas.

Les transferts Isochrones fournissent :

- ✚ Un accès garanti à la bande passante USB.
- ✚ Un temps d'attente limité.
- ✚ Des flux de données - Unidirectionnel.
- ✚ La détection d'erreur via le CRC, mais sans reprise ou garantie de livraison.
- ✚ Seulement des modes pleines et haute vitesse.
- ✚ Pas de données de basculement (basculement, cachées, de commutation)

1.5.2.4 TRANSFERT EN BLOC (OU BULK)

Les Transferts en Bloc peuvent être utilisés pour de grandes quantités de données sporadiques. De tels exemples pourraient inclure un travail d'impression envoyé à une imprimante ou une image provenant d'un scanner.

Les Transferts en Bloc se prémunissent de correction d'erreurs sous la forme d'un champ CRC16 sur les données " charge utile " et sur les mécanismes de détection et de retransmission d'erreurs qui assure la transmission et la réception de données de manière infaillible. Les Transferts en Bloc utiliseront une bande passante de réserve non attribuée sur le Bus après que toutes les autres transactions aient été allouées.

Si le Bus est occupé avec de l'Isochrone et/ou de l'interruption, les données en bloc peuvent alors s'écouler doucement sur le Bus. En conséquence, les transferts en bloc devraient seulement être utilisés pour des communications insensibles au temps du fait du non garanti du temps d'attente.

Les Transferts en Bloc :

- ✚ Utilisés pour de grandes quantités de données sporadiques.
- ✚ Détection d'erreurs via le CRC, avec la garantie de livraison.
- ✚ Pas de garantie de bande passante ou du temps d'attente minimum.
- ✚ Des flux de données - Unidirectionnel.
- ✚ Seulement les modes pleines et haute vitesse.

Ce mode de transfert est uniquement compatible avec le Full USB. Ce mode est réservé pour les gros transferts de données (imprimantes...). Le débit est variable et dépend de la disponibilité. Ce mode assure la reprise sur les erreurs.

Voici un tableau récapitulatif des caractéristiques des 4 types de transfert :

	Control Transfert autorisé aux périphériques low speed	Isochronous Transfert interdit aux périphériques low speed	Interrupt Transfert autorisé aux périphériques low speed	Bulk Transfert interdit aux périphériques low speed
Mode	Message (format prédefini)	Stream (format non prédefini)	Stream (format non prédefini)	Unidirectionnelle (format non prédefini)
Direction	Bidirectionnelle	Unidirectionnelle	Unidirectionnelle	Unidirectionnelle
Contrainte sur la taille maximale du bloc de données	Full speed : 8, 16, 32, 64 octets par trame Low speed : 8 octets par trame	1023 octets full speed par trame	Full speed : 64 octets par trame Low speed : 8 octets pas trame	Full speed : 8, 16, 32, 64 octets par trame
Accusé de réception, reprise sur erreur	Oui	Non	Oui	Oui
Bandé réservée	10% de la trame « best effort »	90% de la trame « guaranteed »		Non « good effort »

Tableau 2 : Caractéristiques des 4 types de transfert

1.5.3 *Les descripteurs USB*

⊕ Définition

On peut définir les descripteurs comme étant des blocs d'informations pré formatés. Tous composants USB doit obligatoirement posséder les descripteurs standards. Tous les transferts d'informations durant cette phase d'énumération se font suivant le type Control. Il va de soi que tout composant USB doit pouvoir être capable de supporter ce type de transfert. Nous verrons par la suite que ce n'est pas le cas pour tous les autres types de transfert que nous définirons.

⊕ Son rôle

Tout le processus d'énumération se fait grâce aux descripteurs qui sont rassemblés dans un fichier texte (fichier assembleur par exemple). En général, un fichier assembleur, qui est ensuite programmé dans le système USB.

Lorsque l'on connecte ou déconnecte un périphérique, celui-ci fournit à l'hôte toutes les informations nécessaires à son identification, c'est à dire les descripteurs.

Ils sont très utiles pour l'hôte puisqu'il peut, de ce fait, à pour rôle de connaître les caractéristiques périphériques comme par exemple la puissance utile, le type de périphérique, le dispositif de transfert des données, le module de gestion ... etc.

⊕ Composition des descripteurs USB

Tous les descripteurs relèvent d'un schéma commun. Le premier octet précise la longueur du descripteur, tandis que le second octet indique le type de descripteur. Si la longueur du descripteur est plus petite que ce que définit la spécification, alors l'hôte doit l'ignorer. Toutefois si la taille est plus grande que prévue, l'hôte ignorera les octets supplémentaires et ne commencera à rechercher le prochain descripteur qu'à la fin de celui-ci.

Décalage	Champ	Taille	Valeur	Description
0	bLength	1	Nombre	Taille du descripteur en octets
1	bDescriptionType	1	Constante	Type de descripteur
2	...	n		Début des paramètres destinés au descripteur

Tableau 3 : Descriptions des descripteurs USB

 **Type des descripteurs :**

- **Les descripteurs d'Appareils** ou « *Device descriptor* » : Ce type de descripteur donne les informations générales. C'est le premier descripteur que vient lire le host.
- **Les descripteurs de Configurations** ou « *Configuration descriptor* » : Un descripteur de configuration renseigne sur les différents états dans lequel peut se trouver le composant USB. Ces descripteurs de configuration définissent par exemple l'origine de l'alimentation.
- **Les descripteurs d'Interfaces** ou « *Interface descriptor* » : Ils peuvent être vus comme un "en tête" ou un regroupement de terminaisons dans un groupe fonctionnel exécutant une simple fonction pour l'appareil.
- **Descripteurs de Terminaisons** ou « *Endpoint descriptor* » : Ce dernier indique la direction du transfert (IN ou OUT), ses types de transfert (ISOCHRONOUS, BULK, INTERRUPTION ou CONTROL), ainsi que d'autres informations.

Ces quatre types de descripteurs sont indispensables. Voici le diagramme hiérarchique de ces descripteurs :

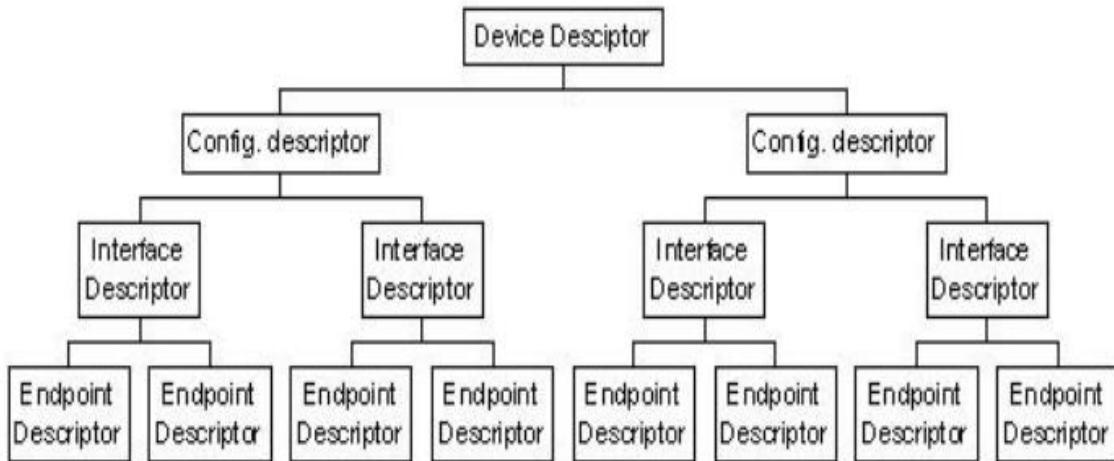


Figure 16 : Diagramme hiérarchique des descripteurs

1.5.4 L'énumération

✚ Définition

Le terme «**énumération**» désigne un processus USB par lequel le système identifie et configure le périphérique en lui donnant une adresse unique. On peut dire aussi que c'est la manière de déterminer l'appareil qui vient juste d'être branché au bus et les paramètres dont il a besoin, comme la consommation électrique, le nombre et le type de terminaison, la classe du produit, ... Donc c'est une gestion dynamique de la connexion et de la déconnexion des périphériques reliés à un bus USB.

✚ Fonctionnement de l'énumération

Lors du démarrage de l'ordinateur, celui-ci interroge tous les périphériques USB connectés à son bus et leur assigne à chacun une adresse. De même lorsqu'on connecte un périphérique USB à chaud, l'ordinateur (hôte) détecte l'ajout du nouvel élément grâce au changement de la tension entre les fils D+ et D- et le même processus se produit.

Débute alors l'étape d'énumération qui permet d'identifier le périphérique (type, constructeur, nom et version) et de le configurer en déterminant le driver (pilote) le plus approprié parmi ceux que possèdent le système d'exploitation. Les drivers génériques donnent accès aux fonctions de base, mais des fonctions avancées peuvent manquer.

Ainsi à la connexion, l'ordinateur envoie un signal d'initialisation au périphérique pendant 10 [ms], puis lui fournit du courant grâce aux fils GND et VBUS (jusqu'à 100 [mA]). Le périphérique est alors alimenté en courant électrique et récupère temporairement l'adresse par défaut (l'adresse

0).L'étape suivante consiste à lui fournir son adresse unique et définitive (c'est la fin de la procédure d'énumération).

Pour cela, l'ordinateur interroge les périphériques déjà branchés pour connaître la leur et en attribue une au nouveau, qui en retour s'identifie (clé mémoire, appareil photo, souris...). L'hôte, disposant de toutes les caractéristiques nécessaires est alors en mesure de charger le pilote (driver) approprié.

L'Enumération sous Windows

- L'hôte ou Hub détecte la connexion d'un nouvel appareil via les résistances de rappel de l'appareil reliées sur les 2 fils de données. L'hôte attend au moins 100 [ms], le temps que la prise soit insérée complètement et que l'alimentation de l'appareil soit stabilisée.
- L'hôte émet un " **reset** " mettant l'appareil dans l'état par défaut. L'appareil peut maintenant répondre à l'adresse zéro par défaut.
- L'hôte (sous MS Windows) demande les 64 premiers octets du descripteur d'appareil.
- Après avoir reçu les 8 premiers octets du descripteur d'appareil, l'hôte émet immédiatement un autre reset sur le bus.
- L'hôte émet maintenant une commande SetAddress, mettant l'appareil dans l'état adressable.
- L'hôte demande la totalité des 18 octets du descripteur d'appareil.
- Puis il demande les 9 octets du descripteur de configuration pour déterminer la taille totale.
- L'hôte demande les 255 octets du descripteur de configuration.
- L'hôte demande l'un des descripteurs de chaînes s'ils étaient indiqués.

A la fin de l'étape 9, " Windows " demandera un driver (pilote logiciel) pour votre appareil. Il est alors courant de le voir redemander tous les descripteurs avant d'émettre une requête SetConfiguration. Le processus d'énumération ci-dessus est courant dans Windows 2000, Windows XP,...

L'étape 4 embarrasse souvent les gens qui écrivent des microprogrammes pour la première fois.

L'hôte demande les 64 premiers octets du descripteur d'appareil, aussi lorsque l'hôte met à zéro votre appareil après avoir reçu les 8 premiers octets, il est tout à fait naturel de penser qu'il y a un problème soit au niveau du descripteur d'appareil soit dans la façon dont votre microprogramme manipule la requête.

Cependant, comme vous le diront beaucoup de gens, si vous insistez sur la mise en œuvre de la commande SetAddress, elle sera récompensée par la demande suivante de 18 octets pleins du descripteur d'appareil.

Mémoire de fin d'études

Généralement quand il y a un problème avec le descripteur ou sur la façon dont il est envoyé, l'hôte tentera de le lire 3 fois avec de longues pauses entre les requêtes. Après la troisième tentative, l'hôte abandonne signalant une erreur au niveau de l'appareil.

CHAPITRE 2. LA TECHNOLOGIE DU PORT FIREWIRE (IEEE 1394 ou I.LINK)

2.1 LA THEORIE DU BUS FIREWIRE

2.1.1 *Historique*

Le bus FireWire qui a d'autre nom IEEE 1394-1995 (nom de la norme à laquelle il fait référence) ou encore i-Link a été mis au point à la fin de l'année 1995 afin de fournir un système d'interconnexion permettant de faire circuler des données à haute vitesse en temps réel.

La société Apple lui a donné le nom commercial « FireWire », qui est devenu le plus usité. Sony lui a également donné le nom commercial d'i-Link, tandis que Texas Instrument lui a préféré le nom de Lynx. Il s'agit ainsi d'un port, équipant certains ordinateurs, permettant de connecter des périphériques (notamment des caméras numériques) à très haut débit. Les connecteurs et câbles FireWire sont repérables grâce à leur forme, ainsi qu'à la présence du logo suivant :



Figure 17 : Symbole du FireWire

2.1.2 *Les normes FireWire*

La norme gère les CDs, les disques dures, les imprimantes, les scanneurs, les caméras numériques.... Le branchement des périphériques peut se faire à chaud et le débit autorisé atteint les 800Mo/s.

Il existe différentes normes FireWire permettant d'obtenir les débits suivants :

⊕ IEEE 1394a (ou version 1)

- IEEE 1394a-S100: 100 [Mbit/s]
- IEEE 1394a-S200: 200 [Mbit/s]
- IEEE 1394a-S200: 400 [Mbit/s]

⊕ IEEE 1394b (ou version 2)

- IEEE 1394b-S800: 800 [Mbit/s] (connecteur à 9 broches)
- IEEE 1394b-S1200: 1200 [Mbit/s]
- IEEE 1394b-S3200: 3200Mbit/s

Cette norme permet au départ de doubler le débit de la norme 1394a (800 Mbit/s au lieu de 400 Mbit/s) toutes en préservant la compatibilité avec la norme antérieure. Un nouveau codage est utilisé (codage 8b-10b). Grâce à IEEE 1394b, la transmission des données et du signal d'arbitrage se fait simultanément, réduisant les temps d'attente lors de l'envoi de petits paquets de données

2.1.3 *Le connecteur FireWire*

Le connecteur du FireWire s'est inspiré du connecteur du Nitendo Game Boy qui, faute d'être élégant, résiste aux enfants. Il existe différents connecteurs FireWire pour chacune des normes IEEE 1394.

- La norme IEEE 1394a définit deux connecteurs :

- **Les connecteurs 1394a-1995 :**



Figure 18 : Type 1394a-1995

- Les **connecteurs 1394a-2000** appelés **mini-DV** : ils sont utilisés pour les caméras vidéo numériques DV (Digital Vidéo)



Figure 19 : Type 1394a-2000

- La norme IEEE 1394b définit deux types de connecteurs dessinés de façon à ce que les prises 1394b-Beta puissent s'enficher dans les connecteurs Beta et Bilingual mais que les prises 1394b Bilingual ne puissent s'enficher que dans les connecteurs Bilingu.

- **Les connecteurs 1394b Bêta**



Figure 20 : Type 1394b Bêta

- **Les connecteurs 1394b Bilingual :**



Figure 21 : Type 1394b Bilingual

2.1.4 Brochage du FireWire

La longueur de câble maximale prise en charge est 4,5 [m]. Les câbles sont dotés de prises à 4,6 et 9 broches. Les câbles équipés de connecteurs à 6 ou 9 broches servent au transport d'alimentation et de données, d'un périphérique à l'autre. Les câbles à 4 broches servent uniquement au transport de données.

N°	6 fils	4 fils
1	VCC (12V)	TPB-
2	Masse (0V)	TPB+
3	TPB-	TPA-
4	TPB+	TPA+
5	TPA-	
6	TPA+	

Tableau 4 : Brochages

Deux brochages distincts existent en s400 et s800 : Le format à 6 broches permettant l'alimentation des périphériques et le format à 4 broches sans alimentation. Le format à quatre broches est celui des PC portables et des caméscopes à bandes mini DV. En s800 les connecteurs ont 9 broches. s400 et s800 sont compatibles : On peut connecter un périphérique s800 avec un s400 en utilisant un câble 9 broches vers 6 broches.

2.1.5 Architecture

Comme la montre la figure ci-dessous, la norme IEEE-1394 emploie une connexion « peer-to-peer ». Les réseaux « peer-to-peer » utilisent les participants de puissance connectée plutôt que de compter sur un petit, le nombre de serveurs concentrée. L'avantage de cette stratégie est qu'IEEE-1394 fournit des taux de données soutenus sans nécessiter un ordinateur hôte pour l'interconnexion entre les périphériques.

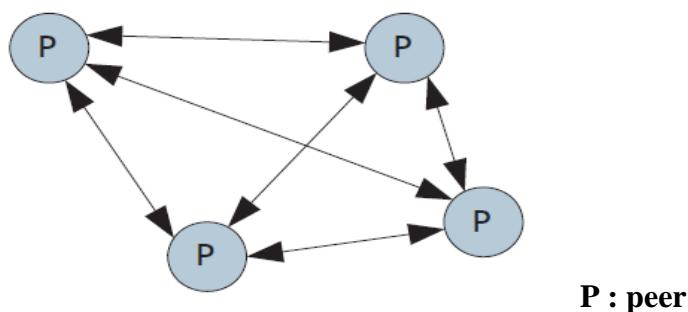


Figure 22 : Architecture de l'IEEE-1394

2.1.6 *Caractéristiques techniques*

Alimentation

- Gamme de tensions d'entrée : de 8 à 33 [V]
- Puissance d'entrée : environ 0,3 [W]
- Gamme de tensions de sortie : de 12 à 30 [V] (selon le produit)
- Puissance de sortie : au moins 7 [W] par port

Sécurité de l'alimentation

L'interface FireWire des ordinateurs Apple comprend des circuits de protection contre les défaillances et la surtension de configuration automatique. S'il n'est pas possible d'alimenter le périphérique à partir de cette interface ou s'il s'éteint de manière inattendue, un câble défectueux ou une défaillance du périphérique peut être à l'origine de ce problème. Si tel est le cas, il faut débrancher le périphérique immédiatement.

Fonctionnalité de répéteur

La fonctionnalité de répéteur intégrée aux ordinateurs Apple équipés d'au moins deux ports permet aux données FireWire de circuler entre les périphériques externes par le biais de l'ordinateur. La fonctionnalité de répéteur est disponible même lorsque l'ordinateur est en mode veille ou est éteint, à condition qu'il soit connecté à une source d'alimentation secteur. Lorsqu'il s'agit d'un ordinateur portable fonctionnant sur batterie, en mode veille ou éteint, la fonctionnalité de répéteur est disponible si au moins un périphérique FireWire utilise le câble FireWire pour alimenter l'ordinateur.

2.2 **FONCTIONNEMENT DU BUS FIREWIRE**

2.2.1 *La topologie*

D'une topologie très proche de celle du l'USB, le FireWire se démarque par sa bande passante bien supérieure. Il ne nécessite pas de HUBs FireWire, le branchement de plusieurs périphériques ne le nécessitant pas. Les cartes FireWire pour micro-ordinateurs possèdent également des ports pour une utilisation interne. Le FireWire supporte aussi le transport de deux flux vidéo temps réel avec une qualité broadcast en simultanée, possède également des ports pour une utilisation interne et aucune supervision de bus central n'est requise.

Le FireWire a donc comme topologie flexible, c'est-à-dire les périphériques peuvent être Daisy-Chained (réseau en marguerite).

Grâce à ses performances, l'IEEE-1394 est destinée particulièrement à relier : les ordinateurs, les produits manipulant de l'audio, des images et de la vidéo, les imprimantes et les scanners, les disques durs, les caméscopes... il peut prendre en charge jusqu'à 63 appareils.

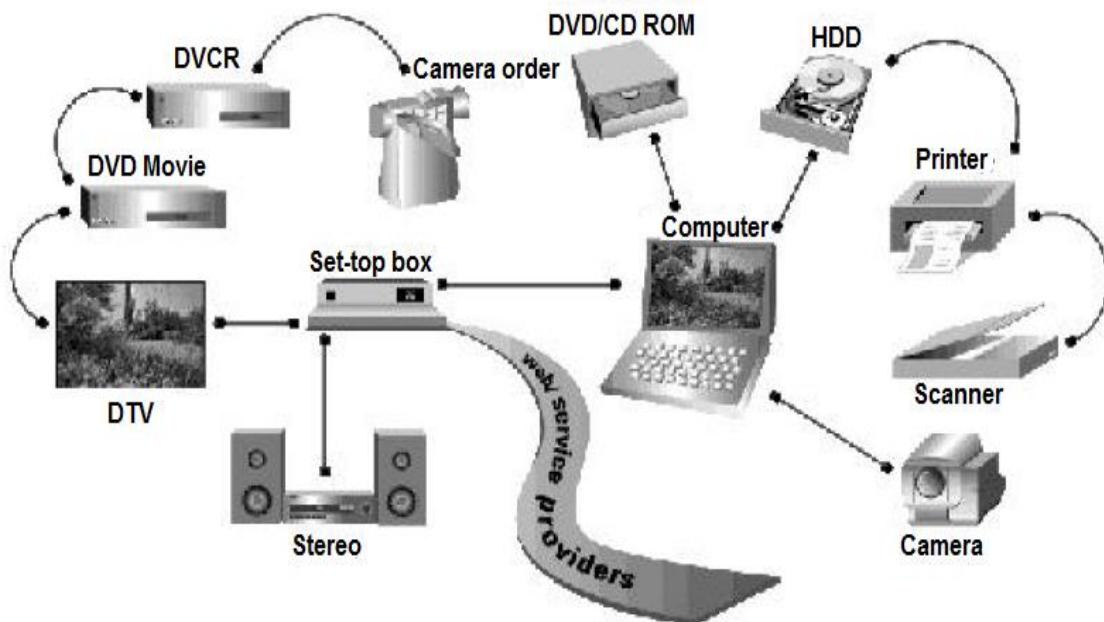


Figure 23 : Topologie du FireWire

2.2.2 Les types de transferts

Pour le port FireWire, il existe 2 principaux types de transferts :

LINK :

- Link Request : Le contrôleur de lien effectue une requête à la couche physique.
- Link Transmit : Le contrôleur de lien envoie des données par paquet à la couche physique pour transmission.
- Link Receive : reçoit des données par paquet de la couche physique (du bus IEEE1394).

PHY :

- PHY Status : donne des informations sur l'état de la couche physique.

Il est à noter que les transferts s'effectuent à la même vitesse.

2.2.3 Modes de transmissions

La couche physique supporte les différentes vitesses de transmission suivant le matériel la composant. La couche de lien définit deux modes de communication « **asynchrone et isochrone** »

2.2.3.1 Le mode de transmission asynchrone

C'est la transmission de paquets à intervalles de temps variables. Il garantit la bonne réception des données par accusé de réception. Ce temps de latence ne peut pas être quantifié car il dépend de taux d'utilisation du bus 1394 par d'autres transmissions pour d'autres appareils communicants entre eux. Ce paquet de données peut être envoyé à une adresse d'un appareil connecté au réseau ou à toutes les adresses. Bref,

- ✚ Ils ciblent un nœud spécifique avec une adresse explicite
- ✚ La quantité de la bande passante utilisable sur le bus n'est pas connue précisément
- ✚ Vérification d'erreurs et mécanismes de retransmission possibles

Pour toute application ne tolérant pas d'erreur dans les données (exemple : disque dur)

2.2.3.2 Le mode de transmission isochrone

Il envoi de paquets de données de taille fixe à intervalle de temps régulier .Il réserve, pour la transmission, un espace-temps de dimension particulière et cyclique, toutes les 125[μs]. Depuis un appareil, un espace-temps isochrone est garanti. Les communications isochrones sont prioritaires aux asynchrones de sorte que la bande passante pour les communications isochrones est assurée. Ainsi, la communication isochrone entre deux appareils ou plus est assimilable à un canal. Une fois qu'un canal a été établit, l'appareil demandeur est garanti d'avoir espace-temps demandé à chaque cycle. C'est ce mode de transmission que l'on choisit pour le transport de données vidéo ou toutes autres données qui ont besoin d'avoir une transmission garantie en "temps réel". En somme donc :

- ✚ Il n'y a pas de correction d'erreur ni de retransmission
- ✚ cadencé grâce aux deux fils d'horloge
- ✚ 80% de la largeur de la bande passante peut être utilisée
- ✚ Allocation de la bande passante est suivie grâce à un nœud sur le bus.

Leurs domaines applications sont sur l'audio et la vidéo. Il représente ainsi le meilleur choix pour l'informatique temps-réel et avec une tolérance de l'erreur.

2.3 CARACTERISTIQUES

2.3.1 *Le protocole*

La norme IEEE 1394 s'appuie sur le modèle OSI et spécifie trois couches : les deux couches basses physique et liaison qui correspondent aux couches 1 et 2 du modèle OSI et la couche transaction qui fait office d'interface entre la couche application et la couche liaison pour le mode asynchrone :

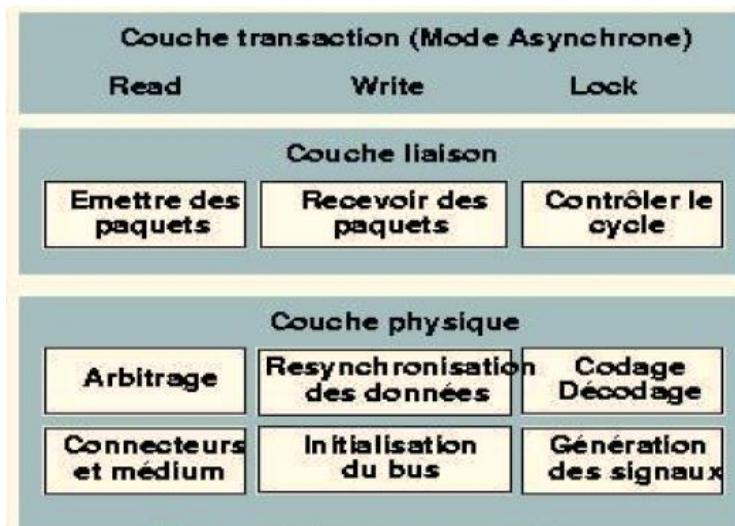


Figure 24 : Modèle de communication

Le protocole de communication IEEE- 1394 se découpe en trois couches distinctes: la couche physique, la couche liaison et la couche transaction :

- ✚ **La couche physique** est responsable de la transmission de bits sur le réseau. Elle gère aussi l'arbitrage du bus afin d'éviter des collisions de données.
- ✚ **La couche liaison** gère la communication de messages d'un point à l'autre du réseau. Elle est responsable des adresses, du contrôle d'erreur et de la gestion haut niveau des accès sur le bus (Cycle de contrôle).
- ✚ **La couche transaction** fournit une interface entre une application externe et le bus FireWire. Elle gère également les réponses aux requêtes, les acquittements, les retransmissions, la puissance et les ressources du bus.
- ✚ La gestion du bus série gère les registres d'adresses, les canaux isochrones, la bande passante et la puissance.

Voici sur la figure ci-dessous ce qu'on vient de citer précédemment :

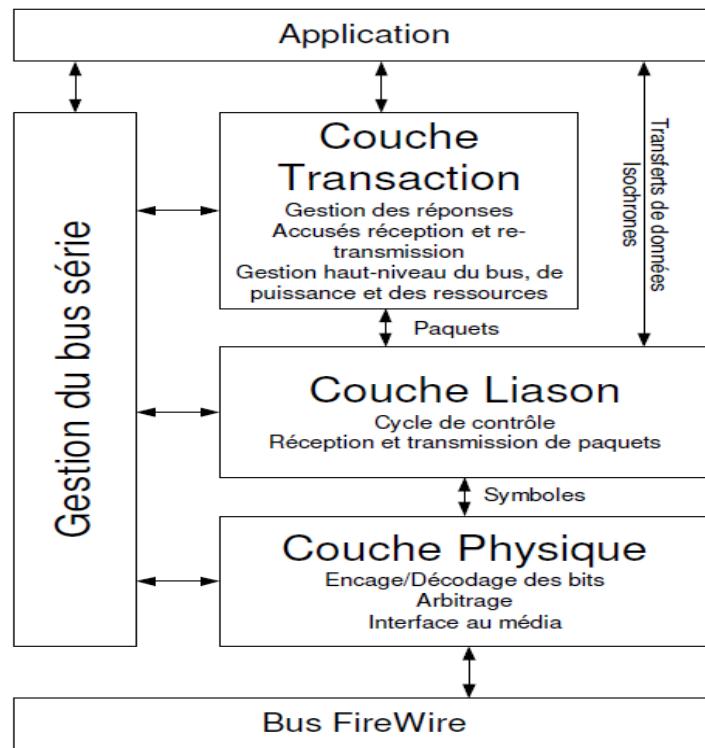


Figure 25 : Le protocole FireWire

2.3.2 La couche physique

La couche physique est responsable du transfert de bits. Les fréquences de transmissions sont 98.304[Mbit/s] ou des multiples de cette fréquence (196.608[Mbits] et 393.216[Mbits] pour 1394a et jusqu'à 32 fois 98.304[Mbit/s] - environ 3,2[Gbits/s] pour 1394b). Le 1394a transmet les données en half-duplex (dans un sens, puis dans l'autre). Le 1394b transmet les bits en full-duplex (transmission simultanée dans les deux sens). La couche physique est responsable aussi de l'arbitrage du bus.

De plus la couche physique permet la détection automatique de nouveau nœud et de l'attribution automatique d'adresse.

Les appareils du réseau sont connectés de point-à-point. Les lignes entre deux appareils sont indépendantes des autres lignes.

Un bus est créé parce que chaque appareil retransmet les signaux reçu sur un de ses ports de branchement vers les autres ports. Cette fonction de répéteur doit être active même quand l'appareil est fermé.

2.3.3 *L'arbitrage*

C'est le niveau où on détermine automatiquement qui est l'arbitre du bus. L'arbitre détermine qui communiquera sur les 2 pairs de fils du bus. L'arbitrage du bus est effectué en quatre phases distinctes:

- ⊕ **Reset** : Reset du bus, forcé lorsqu'un appareil est ajouté ou retiré.
- ⊕ **Identification de l'arborescence du réseau**: Chaque nœud détermine qui est ses parents (l'appareil sur lequel un nœud est connecté) et qui sont ses enfants (les appareils qui sont connectés sur un nœud). Un nœud est proclamé racine de l'arbre du réseau (nœud qui n'a pas de parent).
- ⊕ **Attribution automatique d'IDs**: Chaque nœud obtient un numéro et partage de l'information sur ses caractéristiques avec les autres.
- ⊕ **Arbitrage Normal**: Détermine qui transmet et qui reçoit des données sur le bus. Le mode d'arbitrage dépend de la version du protocole.

2.3.3.1 Arbitrage Normal 1394a

Dans l'arbitrage 1394a, la racine du réseau arbitre les communications. Les nœuds les plus proches de la racine ont une plus grande priorité que les nœuds les plus éloignés. Il s'agit d'une priorité naturelle : pour obtenir le droit de transmettre, tous les enfants font des requêtes à leur parent. Les parents retransmettent cette requête jusqu'à la racine. La racine, quant à elle, accorde le droit de transmettre à la première requête reçue. Comme elle recevra en premier les requêtes des nœuds qui en sont les plus près, ces derniers acquièrent plus facilement le contrôle du bus. Lorsque deux nœuds sont à égale distance de la racine, celui avec le plus petit ID a la plus grande priorité.

2.3.3.2 Arbitrage Normal 1394b

Dans l'arbitrage 1394b, l'arbitre est le dernier nœud ayant transmis dans la sous-action. Il est appelé BOSS dans la littérature 1394b. Les requêtes de temps de bus chevauchent les transferts de données. Les requêtes sont transmises continuellement dans toutes les directions où ne circulent pas les données. Le BOSS détermine qui aura le droit de transmettre selon la priorité des nœuds. Encore une fois, il y a la priorité naturelle vue précédemment et deux niveaux de priorités pour chaque appareil : « **urgent et normal** ». Un appareil ayant des requêtes urgentes se verra accordé le bus avant un appareil ayant des requêtes normales.

2.3.4 La Couche de Lien (Link Layer)

Le temps sur le bus FireWire est découpé en tranches de 125 [us] appelées cycle. Après chaque début de cycle, des blocs de données isochrones sont transmis. Ils sont suivis de sous actions asynchrones. Avant l'envoi de chaque paquet sur le bus, il existe une séquence d'arbitrage afin de déterminer qui pourra transmettre. Le nœud gagnant envoie son paquet après la séquence d'arbitrage. Après chaque paquet, il y a un délai afin de permettre à la destination du message de l'acquitter et de répondre au besoin (intervalle d'acquittement).

Une séquence arbitrage-paquet-acquittement ou arbitrage-paquet-acquittement-réponse-acquittement est appelée une sous action. Entre chaque sous action et entre chaque bloc de données isochrones (également des sous actions), il existe un délai pour assurer que tous les nœuds commencent la phase d'arbitrage. Ce délai est plus court pour les sous actions isochrones (intervalles isochrones) que pour les sous actions asynchrones (intervalles asynchrones), car il n'y a pas d'acquittement individuel pour les données isochrones. Chaque paquet est constitué: d'une entête contenant la source et la destination du message, des données, et d'un checksum afin de valider les données.

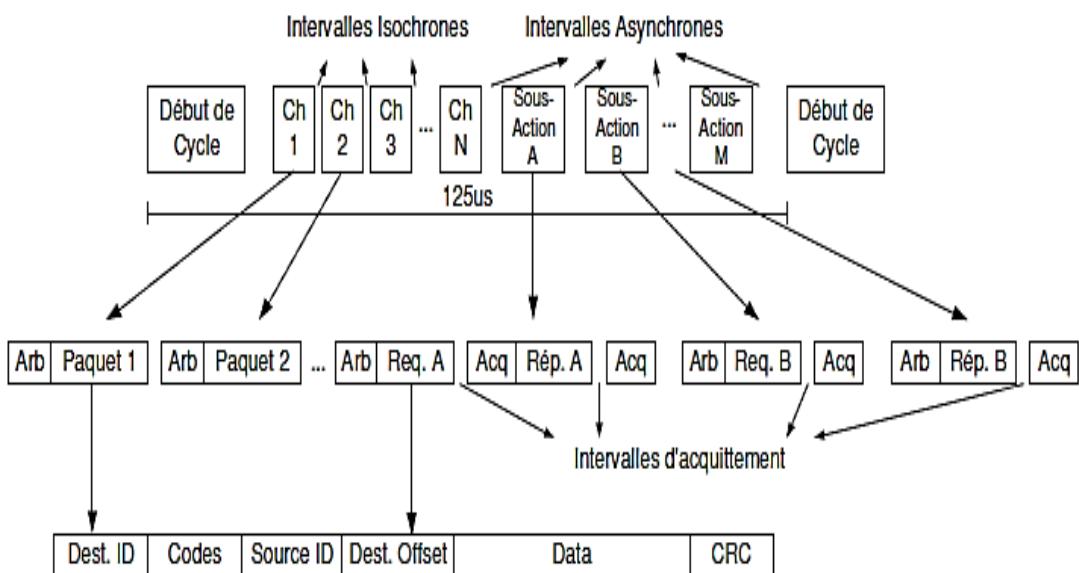


Figure 26 : La couche de Lien

2.3.5 *La Couche de Transaction*

La couche de transaction est responsable de la transmission et de la réception de paquets asynchrones. C'est dans cette couche que sont gérés les requêtes, les réponses, les acquittements, les retransmissions. La couche transaction supporte plusieurs types de transactions:

- ⊕ **Split transactions** : Il s'agit d'une séquence requête-délai-acquittement-délai- réponse-acquittement. La requête et sa réponse sont deux transactions séparées.
- ⊕ **Concatenated transaction** : Il s'agit d'une séquence requête-délai-acquittement- réponse-acquittement. Le répondeur ne relâche pas le contrôle du bus après l'acquittement: il envoie sa réponse tout de suite.
- ⊕ **Unified transaction** : Il s'agit d'une séquence requête-délai-acquittement qui n'a pas de réponse, utilisée lors d'écriture.

CHAPITRE 3. LA COMPARAISON DES DEUX PORTS

3.1 INTRODUCTION

Lorsque l'ordinateur personnel a été introduit sur le marché pour la première fois, la configuration du PC a été un véritable mystère pour les utilisateurs d'ordinateurs. Les utilisateurs n'ont pas compris pourquoi ils devaient avoir un connecteur différent pour chaque appareil et les différences entre un le port souris, un port d'imprimante, le port de jeu ou d'autres. Les câbles à raccorder à ceux-ci ports étaient encore plus confus. Le nombre différent de broches (mâles et femelles) a encore plus difficile de savoir quel câble est allé avec les ports. Les fabricants étaient le développement de technologies qui ont été visant à réaliser des profits immédiats sans donner examen des problèmes de compatibilité et simplicité. Pour rendre ces ports fiables à l'utilisateur, les connexions externes et les câbles ont finalement été codés en couleur par les fabricants pour éliminer toute possibilité de brancher le connecteur au mauvais port. Après la norme PC99 réglementait la couleur pour chaque port, les utilisateurs d'ordinateurs sont devenus moins intimidés de brancher le câble à bon port par eux-mêmes.

3.2 LES DIFFÉRENCES ENTRE USB ET FIREWIRE

Ils peuvent aussi bien être des bus série qui sont utilisés pour la connexion de l'appareil et le transfert de données, mais USB et FireWire ne sont pas aussi semblables tels qu'ils apparaissent à première vue. Alors que les deux technologies offrent la connectivité et de la puissance des ordinateurs et des périphériques, ils ont conçu autour de deux normes distinctes, de sorte qu'ils ont des façons différentes de faire le travail. Ces deux bus présente beaucoup de différence tel qu'au niveau :

⊕ **Conception :** Apple a créé à l'origine de la norme FireWire et Intel la norme USB. USB est beaucoup plus commun et ordinaire.

⊕ **Connexion périphérique :** La principale différence entre les deux est que FireWire est fait pour gérer plus de données que USB, notamment les informations audio et visuelle.

USB est un bus série basée sur l'hôte, ce qui signifie qu'il est utilisé pour connecter des périphériques (tels que les imprimantes, les scanners, appareils photo et lecteurs MP3,...) à un «hôte» (votre PC). Dans la technologie basée sur l'hôte, un ordinateur doit toujours être présent à agir comme «intermédiaire» par lequel les périphériques communiquent. FireWire, d'autre part, est classé comme « peer-to-peer ».

Ce qui signifie qu'il permet à deux ou plusieurs dispositifs périphériques d'interagir sans être relié à un ordinateur (un exemple d'une connexion « peer-to-peer » serait deux caméras liée via FireWire, indépendant d'un PC).

- **Vitesse :** Même si USB est plus largement utilisé que le FireWire, ses capacités de vitesse ne peuvent pas comparer. Les taux de transfert de données de l'interface USB vont de 1,5 Mbits par seconde (basse vitesse) à 480 Mbits / s (haute vitesse). Il y avait un bref moment où la première version de FireWire (qui fonctionne à jusqu'à 400 Mbits / s) et Speed USB haut coïncidaient, et les deux étaient effectivement en concurrence pour un certain temps. Mais cela n'a pas duré longtemps; quand FireWire 800 a frappé le marché à 800 Mbits par seconde, ce était débit de données supérieur de près de deux fois plus rapide que USB haute vitesse.
- **Compatibilités :** Une des plus grandes forces de l'interface USB est son utilisation très répandue dans les ordinateurs d'aujourd'hui. Bien que d'un port FireWire est souvent réservé uniquement pour Mac, la plupart des ordinateurs (Mac et PC) ont des ports au moins un couple USB qui en fait un choix sûr et populaire pour le matériel externe. USB et FireWire ont fait informatiques et électroniques numériques beaucoup plus pratique pour les utilisateurs. Les deux types d'interfaces sont remplaçables à chaud, et les deux ont une place dans le monde d'aujourd'hui, des dispositifs interopérables interconnectées.
- **Distance :** Des cordons plus longs sont disponibles, mais les technologies techniquement parlant, FireWire et USB ont été conçus pour fonctionner via des câbles n'est plus que de 4,5 et 5 mètres, respectivement. Ces mesures sont très proches, et peuvent sembler mettre l'USB et le FireWire au coude-à-coude dans la course pour la capacité de la distance, mais le concours ne s'arrêtent pas là. Si on prend des extensions de chaînage et de moyeu marguerite en compte, le FireWire bat l'USB dans le département de distance car les câbles FireWire peuvent être chaînés (extrémité directement relié à la fin) jusqu'à une distance de 72 mètres, tandis que les câbles pleine vitesse de l'USB ne peuvent être prolongés (avec l'aide de concentrateurs USB) à une longueur maximale de 30 mètres.
- **Puissance :** l'USB et le FireWire sont deux périphériques fournissent de l'électricité, mais ils le font sur des niveaux complètement différents. Avec une puissance nominale de 5 volts, USB est orienté vers des dispositifs à faible consommation, mais l'électronique de grande puissance bénéficié de la capacité de 30 volts de FireWire.

Ainsi, les deux bus série semblent similaires, ils sont destinés à satisfaire différents besoins de la bande passante et des coûts. L'IEEE 1394 peut transférer davantage des données dans un laps de temps donné, mais beaucoup plus cher que l'USB en raison de son protocole plus complexe et le taux de signalisation. Voici un tableau qui résume les différences entre eux :

	FireWire IEEE 1394a, 1394b	USB Vers. 1.0 / 1.1 / 2.0
Débit binaire [Mo/s]	400[Mb/s], 800[Mb/s]	1,2, 12, 480 [Mb/s]
Le nombre maximum de appareils connectés	63	127
Hot-swap?	Oui	Oui
Plug-and-Play?	Oui	Oui
Longueur de câble entre dispositifs	4,5- 100 mètres	5 mètres
Embarqué ligne électrique	Oui	Oui
Périphériques	D-Caméscopes D-caméras Set-Top Boxes HDTV DVD-ROM, RAM Disques durs Imprimantes Scanners	Claviers Souris PC Moniteurs Joysticks DVD-ROM, RAM Basse résolution D-caméras CD-ROM à basse vitesse, RW Modems Imprimantes Scanners
Coût relatif	Supérieur	Inférieur

Tableau 5 : les différences entre l'USB et le FireWire

3.3 CES AVANTAGES

3.3.1 *Le port USB*

- ✚ En général il est **standard** : Les périphériques USB peuvent être utilisés sur presque n'importe quel ordinateur, parce que presque tous les ordinateurs contiennent des ports USB. Cela signifie que de nombreuses entreprises fabriquent des appareils qui utilisent la technologie et que qu'on doit presque toujours être en mesure de connecter les périphériques via USB.
- ✚ Faible coût de l'interface, alimentation possible des dispositifs vis câble, indépendance vis-à-vis des machines hôtes. Ce norme permet aussi d'alimenter le périphérique directement avec le PC. Il n'est pas nécessaire d'alimenter le périphérique avec une alimentation extérieure, le PC suffit.
- ✚ Le port USB est fiable et sécuritaire, c'est à dire la détection et correction automatique d'erreurs. Il offre aussi plusieurs vitesses possibles et 4 types de transferts.
- ✚ Il est facile à utiliser : tout d'abord on peut dire que le protocole USB est très souple ; une interface unique suffit pour commander plusieurs types de périphériques, il suffit juste de posséder le bon driver.

Ensuite un point fort de l'USB est sa configuration automatique, on l'appelle aussi la plug and Play. Cela signifie que si l'utilisateur connecte un périphérique USB, Windows détecte automatiquement ce périphérique et charge le driver approprié s'il est disponible dans le fichier Windows. Il faut noter qu'il n'est pas nécessaire, avec le protocole USB, de lancer un fichier d'installation ou de redémarrer le PC avant d'utiliser avant le périphérique

- ⊕ Un autre avantage du bus USB est la connectique : les câbles possède deux extrémités bien différentes pour ne pas se tromper l'hors du branchement, de plus les câbles USB sont très compactes comparer au câbles pour la liaison parallèle ou série.
- ⊕ La transmission en **paquets de données** : la nécessité d'envoyer des paquets de données pour terminer la transmission avant que le dispositif peut recevoir d'autres données. Ce qui influer sur les performances et la stabilité dans une certaine mesure.
- ⊕ Sur le **coût** : les dispositifs USB sont généralement à un prix inférieur que les dispositifs FireWire, de sorte qu'ils peuvent être plus attrayants pour les utilisateurs sur un budget.

3.3.2 *Le bus FireWire*

- ⊕ La norme IEEE 1394 est plus rapide et plus flexible. Il mieux adapter pour le transfert vidéo et d'autre utilisation ou la vitesse est très important.
- ⊕ Grace a la topologie « peer to peer » du FireWire on peut communiquer avec un autre appareil directement sans passer avec le host comme avec l'USB. On peut donc envoyer l'information a plusieurs périphériques a la fois.
- ⊕ Supportent **une bande passante** plus élevée : Il en résulte la possibilité d'utiliser plusieurs entrées et sorties, ainsi que des performances accrues et de la stabilité au niveau transfert.
- ⊕ La transmission en **flux de données** : Il en résulte une synchronisation et des performances plus stables, c'est-à-dire qu'il peut transférer des données dans les deux directions en même temps.
- ⊕ La capacité de mettre **en cascade** ou **en chaîne** : Ceci signifie que plusieurs périphériques FireWire de la même famille peuvent être connectés ensemble pour les entrées et sorties supplémentaires.

Exemple : On peut connecter deux projets FireStudio PreSonus ensemble pour un total de 20 entrées et 20 sorties.

- ⊕ Il permet d'atteindre des débits réels proches des débits théoriques. Comme un disque dur interfacé en FireWire 400 peut atteindre 40 à 45 [Mo/s] (sur 50 Mo/s théoriques).
- ⊕ Le FireWire consomme moins de CPU et est nettement plus rapide.

3.4 QUELQUES INCONVENIENTS

- Il n'y a aucune forme de chipset compatible avec les périphériques USB.
- Certains ordinateurs ne peuvent pas être mis à niveau pour soutenir FireWire. Pour un ordinateur portable qui n'a pas de Cardbus, PCMCIA ou slot ExpressCard, il n'y a pas moyen d'ajouter FireWire au système. D'autre part, si l'ordinateur est un ordinateur de bureau, mais qui n'a aucune PCI disponible ou PCI Express, on n'est pas en mesure d'ajouter un FireWire sans enlever un autre appareil.
- La plupart des ordinateurs modernes n'ont pas ports FireWire intégré. On peut alors mettre à jour leur ordinateur pour soutenir FireWire, mais cela nécessiterait l'achat de matériel supplémentaire.
- La norme FireWire n'est pas prise en charge par les southbridge (l'une des deux puces du chipset d'une carte mère), ce qui nécessite automatiquement une puce supplémentaire ; et donc augmente les coûts
- Le FireWire n'est adaptée réellement qu'aux disques durs et à la vidéo, ce qui limite son utilisation, et les appareils à cette norme sont généralement plus onéreux.

3.5 EXEMPLE DE COMPARAISON ENTRE USB2.0 ET IEEE 1394

3.5.1 *Performance*

Basé sur des débits publiés, il a été suggéré que l'USB 2.0 (480 [Mb/s]) offre des performances plus élevé que la norme IEEE-1394a (400 [Mb/s]). Lorsqu'on prend à leur valeur nominale, les taux semblent certainement étayer cette affirmation. Dans la demande pratique, cependant, indique le contraire.

En fait, le processeur et l'hôte augmenté les frais généraux de contrôle attribuables à la topologie maître-esclave de l'USB 2.0 et fait réduire son débit soutenu à des taux inférieurs à ceux de l'IEEE-1394a (voir tableau) affiche les résultats d'un test de référence menée par USB-Ware avec un disque dur externe IDE.

Il existe de nombreux exemples de différences de performance entre USB 2.0 et IEEE-1394a disponible sur Internet. Bien que les exploits indiquent une différence de paramètres entre ordinateurs Mac et Windows, le débit de données le plus rapide soutenu de l'USB 2.0 est encore plus lent que le rythme soutenu de l'IEEE-1394a.

Et d'autres, les débits de données IEEE-1394a du PowerBook pour lire et écrire sont et 35 [Mo/s] de 38[MB/s] respectivement. En comparant le taux de données maximum de l'USB 2.0 atteint sur

Windows avec celle de l'IEEE-1394a sur l'ordinateur Mac (puisque les taux de Windows n'ont pas été publiés pour IEEE-1394a), IEEE-1394a surpassé encore USB 2.0 de 15% en lecture et de 29% pour l'écriture.

3.5.2 **Tableau comparatif**

Le tableau ci-dessous présente un aperçu rapide sur les différences entre USB 2.0 et IEEE-1394a. Une différence entre les deux technologies d'interconnexion, à savoir, la capacité de périphériques d'alimentation, est d'une importance particulière. IEEE 1394a offre 16[W] de puissance à travers le bus, tandis que l'USB 2.0 fournit seulement ~ 2.4 [W]. Par conséquent, USB 2.0 nécessite même les appareils photo numériques les plus élémentaires d'être alimenter.

Point	USB 2.0	FireWire
Data Rate Imprimé	480 [Mbits/s]	400 [Mbits/s]
Taux de données soutenues (Lecture)	33 [MB/s]	38[MB/s]
Taux de données soutenues (Ecriture)	27[MB/s]	35 [Mo/s]
Architecture	maître-esclave	Peer-to-Peer
Conçu pour :	La commodité	La vitesse
Grandes avantages	Standard sur 90% des PC	Les taux donnés soutenues pour les appareils vidéo ou autres medias sont parfaits
Plus grands inconvénients	Nécessite une alimentation externe pour la haute performance comme sur les Caméras CCD,...	Il n'est pas généralement standard sur les PC

Tableau 6 : Récapitulation de la comparaison de l'USB2.0 et le FireWire

CONCLUSION GENERALE

Bref, cette étude nous a permis d'améliorer nos connaissances sur les appareillages informatiques. Plus précisément les ports USB et FireWire. Nous avons réussi à étudier la mise en œuvre de ces derniers. L'utilisation du port USB et FireWire de l'ordinateur devient actuellement incontournable au niveau de transmission de données. Et presque la totalité des enceins ports commence à disparaître.

Est-ce que FireWire « gagne » la guerre d'interface face à l'USB, c'est parce qu'il n'est pas nécessaire pour un gagnant. On s'attend que le FireWire et l'USB coexister paisiblement dans les futurs ordinateurs. Le port FireWire et USB remplacera le foule de connecteurs qui se trouve à l'arrière du PC d'aujourd'hui

Donc, on peut réserver l'USB pour les périphériques de faible bande passante (souris, clavier, modem...) et l'utilisation de FireWire pour connecter à la nouvelle génération de haut débit informatique, de l'électronique et la nécessité des produits de grande puissance.

Ainsi, il est possible de transférer un film (ex : DivX) en 1 seconde avec les versions de ces deux ports (USB3.0 ou FireWire 3200). Or tout ceci n'est que théorie car en pratique c'est toujours un peu différent, les débits sont toujours plus bas. Mais nous pouvons imaginer que dans un future proche le transfert de données devient de plus en plus adapté à toutes sorte d'appareils (informatique, électrique, nanotechnologies,...), mais avec quelle type de port ?

REFERENCES BIBLIOGRAPHIQUES

- [1] ACQUIER Bernard “**L’USB en bref**” chapitre I,14 juin
- [2] Anderson, Don “FireWire System Architecture” 2nd Edition (IEEE1394a),spring 2002
- [3] Eric Magarotto “support de cours de transmission et Acquisition de données” Licence EEA-IE, version proviso ire de 15 décembre 2013
- [4] Etienne Tremblay “Ordinateur structure et applications” Université Laval, cours 20, IEEE 1394
- [5] FWDG 1.0Ta “FreWire Desing Guide” 03 fevrier 2010
- [6] Jean-Philipe PERIN “L’ordinateur et ses périphériques” Université Stendhal, Département informatique Pédagogique
- [7] Les ABM”Généralités sur le microorinatuer et ses composant” Chapitre 1, Mars 2003
- [8] Mathieu KUHN “La norme USB en quelque mots” 21 septembre 2002

ANNEXE 1

Les différents LOGO de l'USB



La version 1.0/1.1



La version 2.0



La version 3.0

ANNEXE 2

La transmission différentielle et le codage en NRZI

- ❖ Le terme adéquat est "**signal symétrique**" : cela s'utilise quelquefois en audio et c'est ce qui a été choisi pour l'USB. A l'émission le signal [S] est multiplié par -1 (déphasage de PI) ce qui donne [-S], puis on transmet à la fois [S] sur le fil D+ et [-S] sur le fil D-.

- A la réception on amplifie la différence, ce qui donne:

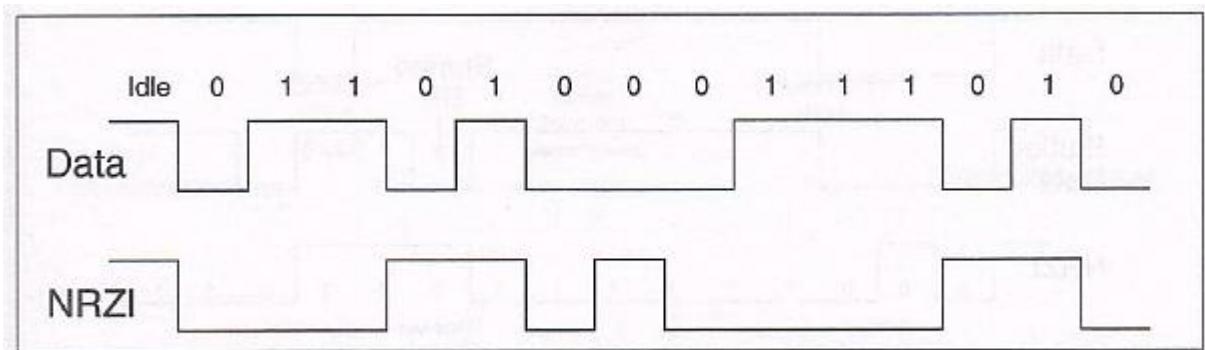
$$[S] - [-S] = [S+S] = [2 * S]$$

- Lors du parcours, tout signal parasite [P] s'additionnera de façon identique sur [S] et [-S], ce qui donne [S+P] et [-S+P]
- A la réception on amplifie la différence:

$$[S+P] - [-S+P] = [S+P+S-P] = [2 * S] ; \text{ et le signal parasite a donc disparu.}$$

- ❖ Pour transmettre les données, l'USB utilise le codage NRZI (Non-Retour à Zéro Inversé). Le principe de ce codage est simple, Un «1» logique est représenté par un non changement d'état en NRZI et un «0» logique est représenté par un changement d'état. Le codage va encore plus loin et utilise le Bit Stuffing ; c'est le fait de mettre un «0» après 6 «1» logique consécutif pour forcer une transition dans le code NRZI (pour éviter les pertes de données).

Ci-dessous un exemple ce codage NRZI. Ce type de codage est uniquement utilisé pour le transport à travers le cordon USB. Il doit être décodé lors de la réception pour pouvoir retraiter les données.



. Principe du Codage NRZI

ANNEXE 3

Explication des différents champs des paquets d'une trame USB

- **Sync** : Ce champ est le début du paquet. Il fait de 8 bits de long pour la basse et pleine vitesse ou 32 bits pour la haute vitesse est utilisé pour synchroniser l'horloge du récepteur avec celle de l'émetteur/récepteur. Les 2 derniers bits indiquent l'endroit où le champ PID commence.
- **PID (Paquet ID)** Ce champ est utilisé pour identifier le type de paquet qui est envoyé. Le tableau suivant montre la désignation des PIDs :

Groupe	Valeur PID	Identificateur Paquet
Token	0001	OUT Token
	1001	IN Token
	0101	SOF Token
	1101	SETUP Token
Data	0011	DATA0
	1011	DATA1
	0111	DATA2
	1111	MDATA
Handshake	0010	ACK Handshake
	1010	NAK Handshake
	1110	STALL Handshake
	0110	NYET (No response Yet)
Special	0000	PREambule
	1100	ERR
	1000	Split
	0100	Ping

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. Le format résultant figure ci-dessous :

PID0	PID1	PID2	PID3	nPID1	nPID1	nPID2	nPID3

- **ADDR** : Le champ adresse détermine à quel appareil le paquet est destiné. Sa longueur de 7 bits, lui permet de supporter 127 appareils.
- **ENDP** : Le champ de terminaison est composé de 4 bits, autorisant 16 terminaisons possibles. Les appareils Low Speed, toutefois peuvent seulement avoir 2 terminaisons additionnelles au-dessus du canal de communication par défaut (4 terminaisons maximales)
- **CRC** : Les Contrôles à Redondance Cyclique sont exécutés sur les données à l'intérieur du paquet de charge utile. Tous les paquets jetons ont un CRC de 5 bits tandis que les paquets de données ont un CRC de 16 bits.
- **EOP** : Fin de Paquet. Signalé par une sortie unique zéro (SE0) pendant une durée approximative de 2 bits suivie par un " J " d'une durée de 1 bit.



Auteurs :

RAKOTOARIMANANA Avotra Mahefarivo
RAKOTOARISOA ANDRIAMANANA Narisoa Olivier

Contacts :

033 02 812 95
033 12 297 16

E-mails :

ravotramahefa02@gmail.com
rakotoarisoanarisoaa@gmail.com

Etablissement :

Ecole Supérieure Polytechnique d'Antananarivo

Filière :

Génie Electrique

RESUME

Les USB et FireWire sont deux normes pour les échanges des données. Ils ont tous ses caractéristiques spécifiques et présentent tous des avantages et des inconvénients. Ces derniers ont aussi leurs fonctions particulières dans des différents domaines d'application.

ABSTRACT

USB and FireWire ports are two standards for the transfer and exchange of data. They all have their specific characteristics and all have advantages and disadvantages. These also have their specific functions in different application areas.

Mots clés : USB, FireWire, Périphérique, IEEE 1394, hôte

Directeur de mémoire : Monsieur RAVALOMANANA

Nombre de pages : 47

Nombre de figures : 26

Nombre de tableaux : 06