

L'USB en bref

Donner un sens au standard USB

Les types de terminaisons.

La spécification du Bus Série Universel définit 4 types de transferts ou de terminaisons,

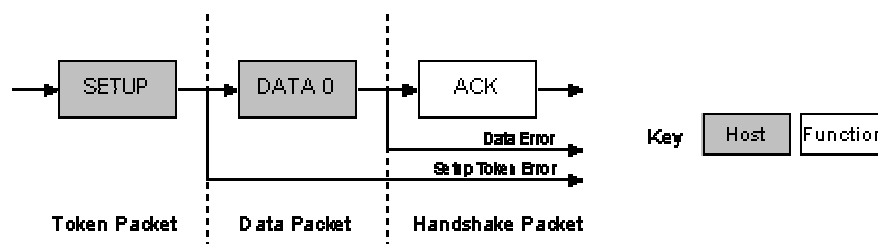
- **Transferts de commande**
- **Transferts d'interruption**
- **Transferts Isochrone**
- **Transferts en Bloc** (BULK)
- **Gestion de la bande passante.**

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. Il 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.

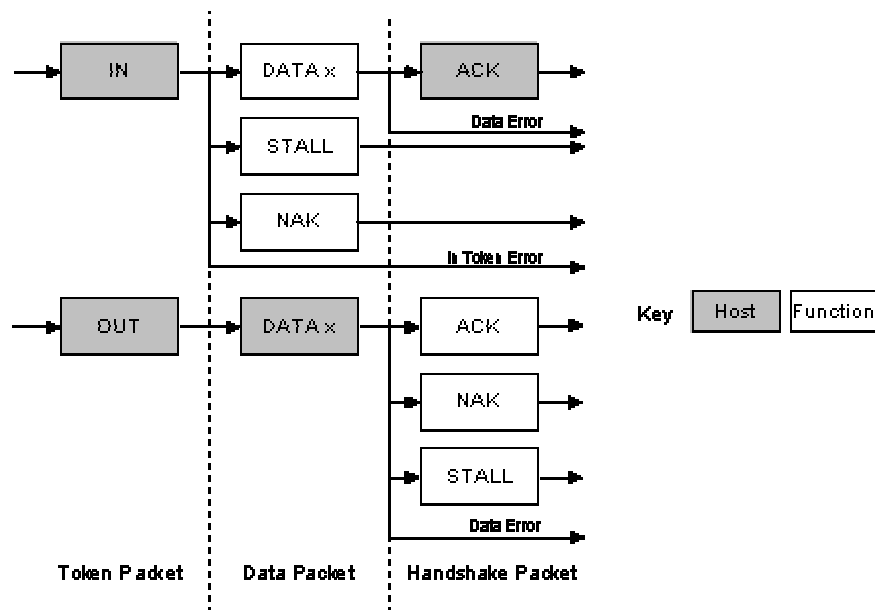
- L'**étape d'installation** se passe lorsqu'une demande est envoyée. Elle est composée de 3 paquets. Le jeton (token) d'installation envoyé le premier est celui qui contient l'adresse et le numéro de la terminaison. Le paquet de données est envoyé après et a toujours un type PID de Data0 et inclut un 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 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.

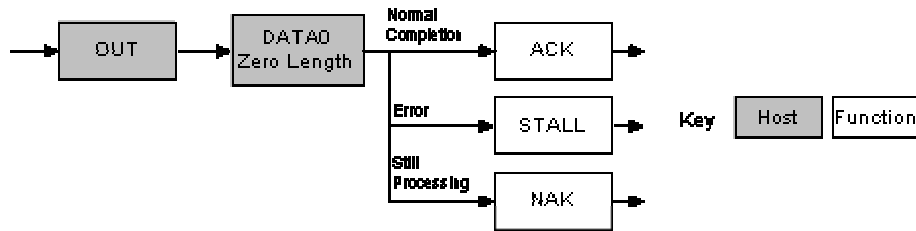
L'étape de données comporte 2 scénarios différents selon la direction du transfert de données.

- **IN** (Entrée): Quand l'hôte est prêt à recevoir les données de commande, il émet un jeton (token) IN. Si la fonction reçoit le jeton IN avec une erreur, c'est à dire que le PID ne correspond pas avec les bits inversés du PID, il ignore donc le paquet. Si le jeton est reçu correctement, l'appareil peut soit répondre avec un paquet de données contenant les données de commande à envoyer, soit un paquet "d'arrêt" signalant que la terminaison a eu une erreur soit un paquet NAK signalant à l'hôte que la terminaison fonctionne, mais provisoirement n'a pas de données à envoyer.

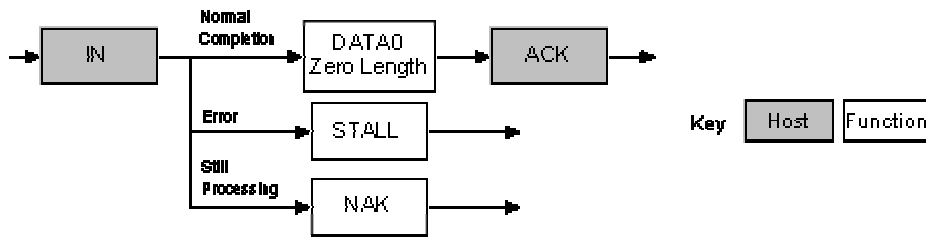


- **OUT** (Sortie): Quand l'hôte a besoin d'envoyer à l'appareil un paquet de données de commande, il émet un jeton OUT suivi par un paquet de données contenant les données de commande comme "charge utile" (payload). Si une partie du jeton OUT ou du paquet de données est altéré alors la fonction ignore le paquet. Si le buffer de terminaison de la fonction était vide et qu'il a cadencé les données dans le buffer de terminaison, il produit un ACK avisant l'hôte qu'il a bien reçu les données. Si le buffer de terminaison n'est pas vide à cause du traitement du paquet précédent, alors la fonction retourne un NAK. Toutefois si la terminaison comporte une erreur et que son bit "halt" ai été positionné, elle retourne un STALL (Bloqué).
- L'**étape d'état** 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.

IN (Entrée): Si l'hôte envoie un (ou des) jeton(s) IN pendant l'étape de données pour recevoir des données, alors l'hôte doit valider la bonne réception de ces données. Ceci est réalisé par l'hôte qui envoie un jeton OUT suivi par un paquet de données de longueur nul. La fonction peut maintenant rendre compte de son état dans l'étape poignée de mains. Un ACK indique que la fonction a achevé la commande et qu'elle est maintenant prête à accepter une autre commande. Si une erreur s'est produite pendant l'exécution de cette commande, alors la fonction émettra un STALL (Bloqué). Toutefois si la fonction continue l'exécution, elle retourne un NAK indiquant à l'hôte la nécessité de répéter l'étape d'état ultérieurement.



- **OUT (Sortie):** Si l'hôte envoie un (ou des) jeton(s) OUT pendant l'étape de données pour transmettre des données, la fonction validera la bonne réception des données en envoyant un paquet de longueur nul en réponse à un jeton IN. Toutefois si une erreur intervenait, cela déboucherait sur un STALL ou si la fonction était encore occupé à traiter les données, cela déboucherait sur un NAK demandant à l'hôte de retenter l'étape d'état ultérieurement.



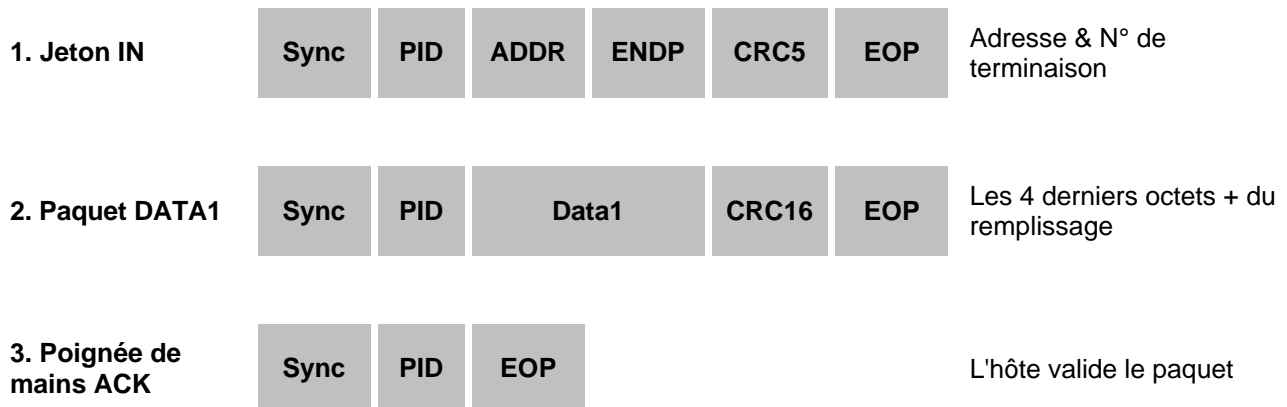
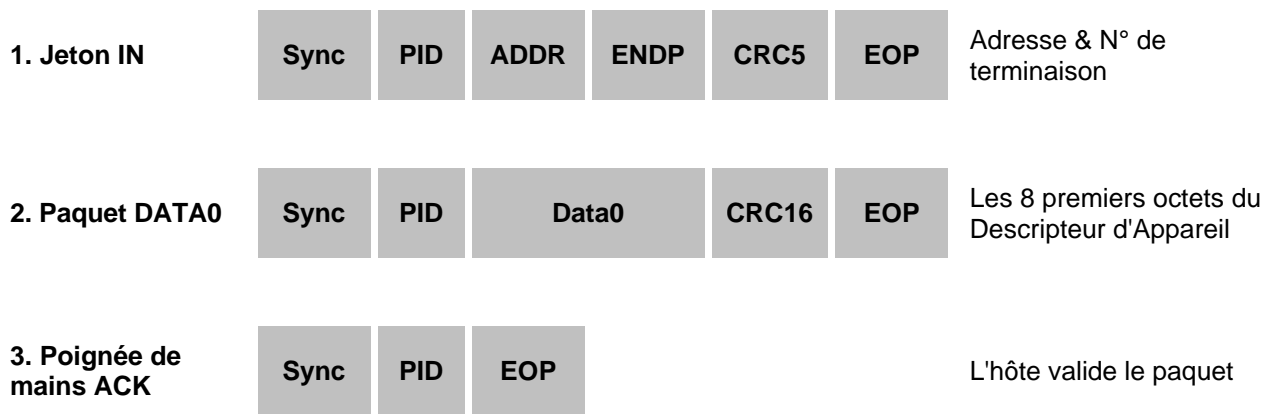
Transferts de Commande: Le coeur du problème.

Maintenant, voyons comment tout cela s'accorde? Supposons par exemple que l'hôte veuille demander un descripteur d'appareil pendant l'énumération. Les paquets qui sont envoyés sont les suivants:

L'hôte enverra un jeton d'installation disant à la fonction que le paquet suivant est un paquet d'installation. Le champ adresse fixera l'adresse de l'appareil à qui l'hôte a demandé le descripteur. Le numéro de la terminaison devrait être un zéro, indiquant une ligne défectueuse. L'hôte enverra alors un paquet DATA0. Celui-ci aura 8 octets de "charge utile" (payload) correspondant à la Demande de Descripteur d'Appareil comme souligné au chapitre 9 de la spécification USB. La fonction USB annoncera alors que le paquet d'installation a été lu correctement et sans aucune erreur. Si le paquet était reçu altéré, l'appareil ignorerait tout simplement ce paquet. L'hôte renverra alors le paquet après un court délai.

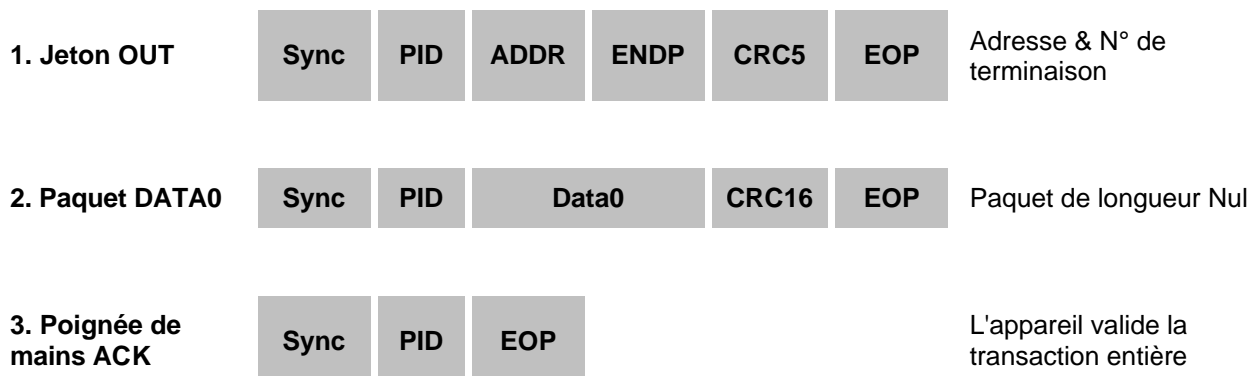
1. Jeton d'installation	Sync	PID	ADDR	ENDP	CRC5	EOP	Adresse & N° de terminaison
2. Paquet DATA0	Sync	PID	Data0	CRC16	EOP	Demande de Descripteur d'Appareil	
3. Poignée de mains ACK	Sync	PID	EOP			L'appareil valide le paquet d'installation	

Les 3 paquets ci-dessus représentent la première transaction USB. L'appareil USB décodera maintenant les 8 octets reçus, et déterminera que c'était une demande de Descripteur d'Appareil. L'Appareil tentera d'envoyer le Descripteur d'Appareil, qui sera la transaction USB suivante.



Dans ce cas nous assumons que la taille maximale de "charge utile" (payload) est de 8 octets. L'hôte envoie le jeton IN, disant à l'Appareil qu'il peut maintenant envoyer des données pour cette terminaison. Comme la taille maximale du paquet est de 8 octets, nous devons scinder les 12 octets du Descripteur d'Appareil en morceaux avant de pouvoir les envoyer. Chaque morceau doit être de 8 octets excepté la dernière transaction. L'hôte validera chaque paquet de données que nous lui enverrons.

Une fois que le Descripteur d'Appareil est envoyé, il s'ensuit une transaction d'état. Pour le cas où les transactions seraient réussies, l'hôte enverra un paquet de longueur Nul indiquant que l'ensemble de transactions était correct. La fonction répond alors à ce paquet de longueur Nul indiquant son état.



Transferts d'Interruption

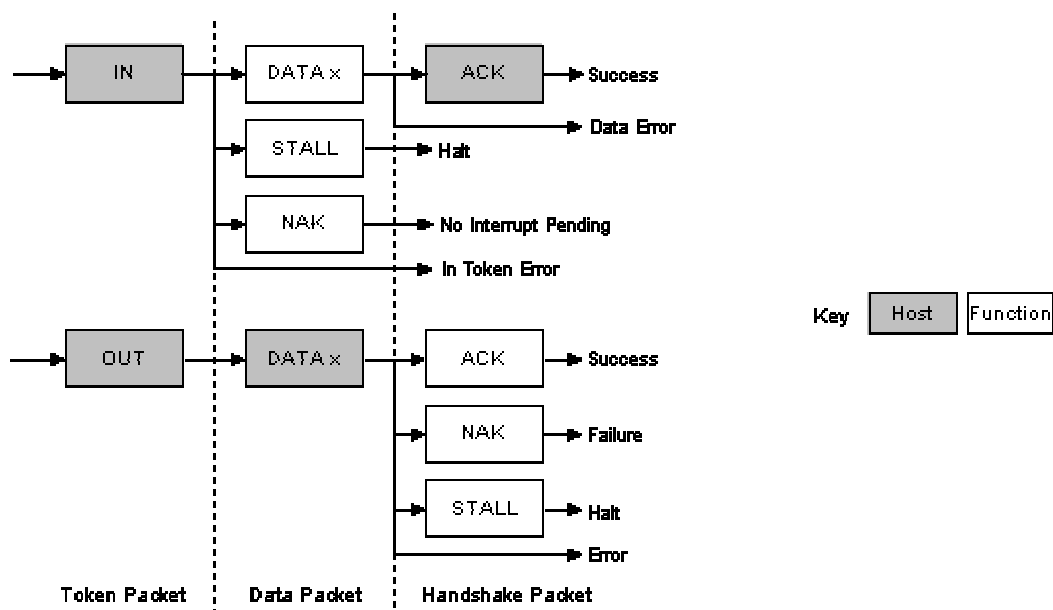
Celui qui eu une expérience de demandes d'interruption sur microcontrôleurs saura que les interruptions sont générées par l'appareil. Toutefois sous USB, si un appareil demande l'attention de l'hôte, il doit attendre que l'hôte l'interroge avant de signaler qu'il a besoin d'une attention urgente!

Transferts d'Interruption

- Temps de retard (ou Latence)garanti
- Ligne de flux - Unidirectionnel
- Détection d'erreur et nouvel essai sur période suivante

Les transferts d'interruptions sont communément non périodiques, les petits appareils qui ont "initié" la communication ont besoin de temps de retard limité.

- La taille maximale de "charge utile" (payload) pour des appareils basse vitesse est de 8 octets
- La taille maximale de "charge utile" (payload) pour des appareils pleine vitesse est de 64 octets
- La taille maximale de "charge utile" (payload) pour des appareils haute vitesse est de 1024 octets



Le diagramme ci-dessus montre le format d'une transaction d'interruption d'entrée IN et d'interruption de sortie OUT.

IN: L'hôte interrogera périodiquement la terminaison d'interruption. Le taux d'interrogation est spécifié dans le descripteur de terminaison qui sera examiné plus tard. Chaque interrogation obligera l'hôte à envoyer un jeton IN. Si le jeton IN est altéré, la fonction ignore le paquet et continue la surveillance du Bus pour de nouveaux jetons.

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

présente quand l'hôte interrogeait la terminaison d'interruption avec un jeton IN, alors la fonction signale cet état en envoyant un NAK. Si une erreur se produisait sur cette terminaison, un STALL (Bloqué) serait envoyé en réponse à un jeton IN.

OUT: Quand l'hôte veut envoyer à l'appareil les données d'interruptions, il émet un jeton OUT suivi par un paquet Data contenant les données d'interruption. Si une partie du jeton OUT ou du paquet Data est altéré alors la fonction ignore le paquet. Si le tampon de terminaison de la fonction était vide et qu'il ait cadencé les données dans le tampon de terminaison il émettrait un ACK prévenant l'hôte qu'il a reçu correctement les données. Si le tampon de terminaison n'est pas vide à cause du traitement d'un paquet précédent, alors la fonction retourne un NAK. Toutefois si une erreur se produisait à cause de la terminaison et que son bit d'arrêt (Halt) ait été positionné, elle renverrait un STALL (Bloqué).

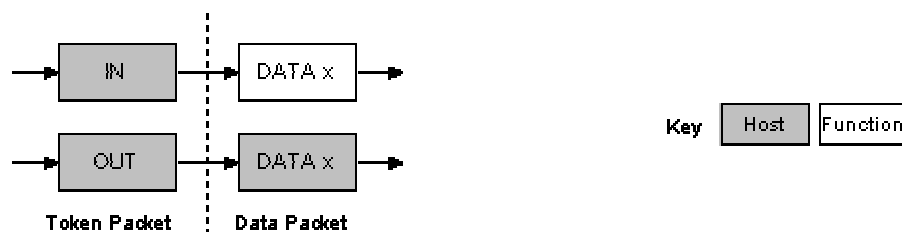
Transferts Isochrones

Les Transferts Isochrones se produisent continuellement et périodiquement. 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 pleine et haute vitesse.
- Pas de données de basculement (basculage, cachées, de commutation) **toggleing**?????.

La taille maximale de données en " charge utile " (payload) est spécifié dans le descripteur de terminaison d'une terminaison Isochrone. Cela peut être un maximum de 1023 octets pour un appareil pleine vitesse et 1024 octets pour un appareil haute vitesse. Comme la taille maximale de données en " charge utile " va effectuer des exigences de bande passante du Bus, il est prudent de spécifier une taille de " charge utile " modérée. Si vous utilisez de grandes " charges utiles " il peut être aussi plus intéressant de spécifier une série d'interfaces alternatives avec des tailles de charges utiles Isochrones variables. Si pendant l'énumération l'hôte ne peut pas valider votre terminaison Isochrone nommée à cause des restrictions de bande passante, il reste une solution de repli préférable à un échec complet. Les données qui ont été envoyées sur une terminaison Isochrone peuvent être inférieures à la taille pré négociée et peuvent varier en longueur de transaction en transaction.



Le diagramme ci-dessus montre le format d'une transaction Isochrone IN et OUT. Les transactions Isochrones n'ont pas d'étape de poignée de mains et ne peuvent pas rendre compte des erreurs ou des conditions STALL / HALT ((Bloqué) / Arrêt).

Transferts en Bloc

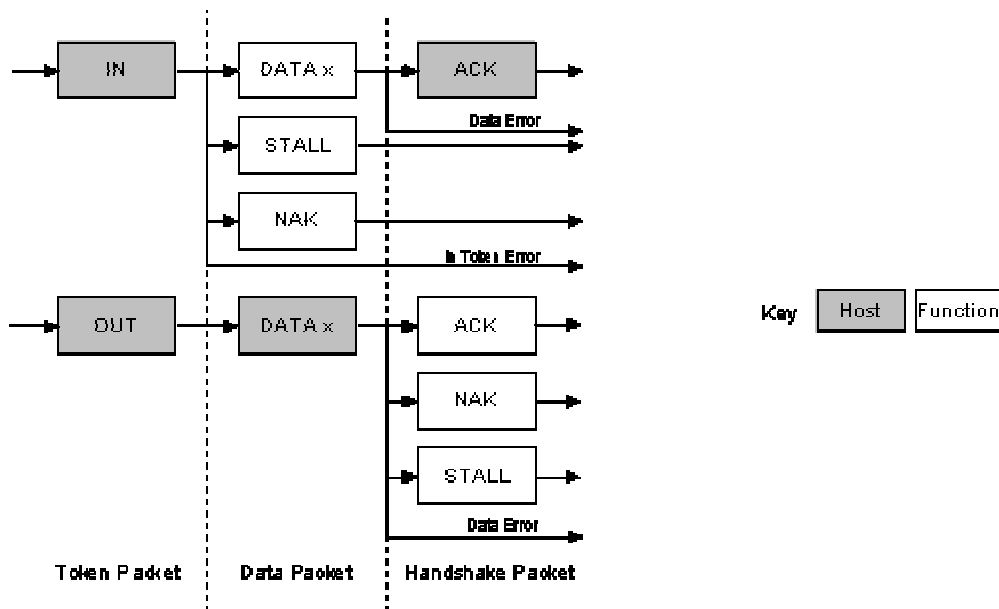
Les Transferts en Bloc peuvent être utilisés pour de grandes quantités de données sporadiques. De tels exemples pourraient inclure un travail d'impression envoyé à une imprimante ou une image provenant d'un scanner. Les Transferts en Bloc se prémunissent de correction d'erreurs sous la forme d'un champ 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 de la non garantie 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 des modes pleine et haute vitesse.

Les Transferts en Bloc sont seulement supportés par des appareils pleine et haute vitesse. Pour des terminaisons pleine vitesse, la longueur maximale du paquet en Bloc est soit 8, 16, 32 ou 64 octets. Pour des terminaisons haute vitesse, la longueur maximale du paquet peut aller jusqu'à 512 octets. Si la charge utile des données est inférieure à la taille maximale du paquet, elle n'a pas besoin d'être remplie avec des zéros. Un transfert en Bloc est considéré comme complet quand il a transféré la quantité exacte de données demandées, ou un paquet inférieur à la taille maximale de la terminaison, ou un paquet de longueur zéro.



Le diagramme ci-dessus montre le format d'une transaction en Bloc IN et OUT.

IN: Quand l'hôte est prêt à recevoir des données en Bloc, il émet un jeton IN. Si la fonction reçoit le jeton IN avec une erreur, il ignore le paquet. Si le jeton est reçu correctement, la fonction peut soit répondre avec un paquet DATA contenant les données en Bloc à envoyer ou bien un paquet Stall signalant que la terminaison a eu une erreur ou un paquet NACK signalant à l'hôte que la terminaison travaille, mais provisoirement n'a pas de données à envoyer.

OUT: Quand l'hôte veut envoyer à la fonction un paquet de données en Bloc, il émet un jeton OUT suivi par un paquet DATA contenant les données en Bloc. Si une partie du jeton OUT ou du paquet DATA est altérée, alors la fonction ignore le paquet. Si le tampon de terminaison de la fonction est vide et qu'il a cadencé les données dans le tampon de terminaison, il émet un ACK prévenant l'hôte qu'il a reçu correctement les données. Si le tampon de terminaison n'est pas vide à cause du traitement d'un paquet précédent, alors la fonction retourne un NAK. Toutefois si la terminaison a eu une erreur et que son bit d'arrêt a été positionné, elle retourne un Stall (Bloqué).

Gestion de la bande passante.

L'hôte est responsable de la bande passante du Bus. Elle est faite par l'énumération (dénombrement) lors de la configuration des terminaisons Isochrones et d'interruptions et durant le fonctionnement du Bus. La spécification place des limites sur le Bus, l'autorisant à allouer plus de 90% pour des transferts périodiques (Interruption et Isochrone) sur un Bus pleine vitesse. Sur des Bus hautes vitesses, cette limitation se réduit à moins de 80% d'une micro-trame qui peut être allouée pour des transferts périodiques.

Aussi vous pouvez assez rapidement voir que si vous avez un Bus hautement saturé avec des transferts périodiques, les 10% restants sont laissés pour les transferts de contrôle et une fois qu'ils ont été attribués, les transferts en Bloc prendront ce qui reste.