

L'USB en bref

Donner un sens au standard USB

Le Paquet d'installation

Tout appareil USB doit répondre aux paquets d'installation sur le canal de communication par défaut. Les paquets d'installation sont utilisés pour la détection et configuration de l'appareil et véhiculent des fonctions courantes telles que la mise en place de l'adresse de l'appareil USB, la demande d'un descripteur d'appareil ou la vérification de l'état d'une terminaison.

Un hôte USB conforme s'attend à ce que toutes les requêtes soient traitées dans une période maximale de 5 secondes. Il précise aussi des temps plus stricts pour des requêtes particulières :

Les requêtes standards d'appareils sans étage de données doivent être accomplies en 50 ms.

Les requêtes standards d'appareils avec un étage de données doivent commencer à renvoyer les données 500 ms après la requête.

- Chaque paquet de données doit être envoyé dans les 500 ms de la transmission réussie du paquet précédent.
- L'étage d'état doit être accompli dans les 50 ms après la transmission du dernier paquet de données.

L'instruction **SetAddress** (qui contient une phase de données) doit traiter l'instruction et retourner l'état dans ces 50 ms. L'appareil a donc 2 ms pour changer d'adresse avant que la prochaine requête ne soit envoyée.

Ces temporisations sont très acceptables même pour les appareils les plus lents, mais peuvent être restrictifs pendant les mises au point. 50 ms ne sont pas suffisants pour l'envoi de caractères de mise au point à 9600bds vers un port série asynchrone ou bien pour un émulateur/débogueur de faire du pas à pas ou d'arrêter l'exécution du programme pour examiner les registres internes. En conséquence, l'USB demande des méthodes différentes de mises au point des autres projets à base de microcontrôleur.

En lisant le DDK XP (Kit Développement), vous noterez que le driver contrôleur d'hôte a maintenant une instruction `USBUSER_OP_SEND_ONE_PACKET` qui comporte le commentaire suivant : " Cette API (Application Programme Interface) est utilisée pour appliquer le 'pas à pas' en tant qu'outil de développement pour une transaction USB ". Alors qu'un tel outil n'a pas encore été produit, nous ne pouvons qu'espérer en voir bientôt un sur le marché.

Chaque requête commence avec un paquet d'installation de 8 octets qui a le format suivant :

Décalage	Champ	Taille	Valeur	Description
0	bmRequestType	1	Bit-Map	D7 Direction de transfert de phase de données 0 = Hôte vers l'appareil 1 = Appareil vers Hôte D6..5 Type 0 =00= Standard 1 =01= Classe 2 =10= Constructeur 3 =11= Réservé D4..0 Destinataire 0 = Appareil 1 = Interface 2 = Terminaison 3 = Autre 4..31 = Reservé
1	bRequest	1	Valeur	Requête
2	wValue	2	Valeur	Valeur
4	wIndex	2	Index ou Décalage	Index
6	wLength	2	Compteur	Nombres d'octets à transférer s'il y a une phase de données

Le champ **bmRequestType** déterminera le sens de la requête, le type de la requête et le destinataire désigné. Le champ **bRequest** indique la requête formulée. **bmRequestType** est généralement analysé et l'exécution est raccordée à un numéro d'identificateurs comme un gestionnaire de requête d'un appareil standard, un gestionnaire de requête d'une interface standard, un gestionnaire de requête de terminaison standard, un gestionnaire de requête d'un appareil de classe, etc.... La façon d'exécuter le paquet d'installation vous appartient. D'autres pourraient choisir d'exécuter en premier **bRequest** puis de déterminer le type et le destinataire basé sur chaque requête.

Les requêtes standards sont communes à tous les appareils USB et seront détaillés dans les prochaines pages. Les requêtes de classe sont communes aux classes de drivers (pilote logiciel). Par exemple, tout appareil étant conforme à la classe HID aura un ensemble commun de requêtes spécifiques de classe. Ils différeront d'un appareil conforme à la classe communication et différeront encore d'un appareil conforme à la mémoire de masse (de grande capacité de stockage).

Et pour finir, les requêtes définies par le constructeur. Ce sont des requêtes que vous pouvez attribuer en tant que concepteur d'appareil USB. Elles sont normalement différentes d'un appareil à un autre, mais dépendent de votre réalisation et de votre imagination.

Une requête commune peut être dirigé vers des destinataires différents et basée sur les fonctions différentes exécutée par le destinataire. Par exemple une requête standard **GetStatus** peut être dirigée sur l'appareil, l'interface ou la terminaison. Quand elle est dirigée sur un appareil, ce dernier retourne des drapeaux (flags) indiquant l'état de la station d'activation et si l'appareil est auto-alimenté.

Toutefois si la même requête est dirigée sur l'interface il renvoie toujours Zéro, ou bien si elle était dirigée sur une terminaison, l'appareil retournerait le drapeau Halt pour la terminaison.

Les champs **wValue** et **wIndex** permettent le passage de paramètres avec la requête. **wLength** est utilisé pour préciser le nombre d'octets à transférer au cas où il existerait une phase de donnée.

Les Requêtes standard

La section 9.4 de la spécification USB détaille les requêtes " d'appareils standard " exigées qui doivent être appliquées pour chaque appareil USB. Le standard fournit un tableau unique regroupant des articles par requête. Considérant que la plupart des (micro)programmes analyseront le paquet d'installation par destinataire, nous choisirons de séparer les requêtes en fonction du destinataire pour une étude et une application plus simple.

Les Requêtes d'appareil standard

Il y a actuellement 8 requêtes d'appareil standard, chacune étant détaillée dans le tableau ci-dessous:

bmRequestType	bRequest	wValue	wIndex	wLength	Données
1000 0000b	GET_STATUS (0x00)	Zéro	Zéro	Deux	Etat de l'appareil
0000 0000b	CLEAR_FEATURE (0x01)	Sélecteur de fonction	Zéro	Zéro	Aucune
0000 0000b	SET_FEATURE (0x03)	Sélecteur de fonction	Zéro	Zéro	Aucune
0000 0000b	SET_ADDRESS (0x05)	Adresse de l'appareil	Zéro	Zéro	Aucune
1000 0000b	GET_DESCRIPTOR (0x06)	Type de Descripteur & Index	Zéro ou ID de Langues	Longueur du Descripteur	Descripteur
0000 0000b	SET_DESCRIPTOR (0x07)	Type de Descripteur & Index	Zéro ou ID de Langues	Longueur du Descripteur	Descripteur
1000 0000b	GET_CONFIGURATION (0x08)	Zéro	Zéro	1	Valeur de Configuration
0000 0000b	SET_CONFIGURATION (0x09)	Valeur de Configuration	Zéro	Zéro	Aucune

La requête **GetStatus** dirigée vers l'appareil retournera 2 octets pendant l'étape de données suivant le format :



Si D0 est à 1, ceci indique que l'appareil est auto alimenté. S'il est à 0, l'appareil est alimenté par le bus. Si D1 est à 1, l'activation à distance de l'appareil est validée et ce dernier peut réveiller l'hôte pendant le mode veille. Le bit d'activation à distance peut être positionné par les requêtes **SetFeature** et **ClearFeature** grâce à un sélecteur de fonction de `DEVICE_REMOTE_WAKEUP(0x01)`.

Les requêtes **Clear Feature** et **Set Feature** peuvent être utilisées pour positionner des fonctions booléennes. Quand le destinataire désigné est l'appareil, les 2 seuls sélecteurs de fonctions disponibles sont `DEVICE_REMOTE_WAKEUP` et `TEST_MODE`. Test mode permet à l'appareil de présenter diverses conditions. A leur égard, des précisions supplémentaires sont fournies dans la spécification USB, version 2.0.

Set Address est utilisé pendant l'énumération pour attribuer une adresse unique à l'appareil USB. L'adresse est précisée dans **wValue** et peut valoir au maximum 127. Cette requête est unique dans le sens où l'appareil ne positionnera pas son adresse tant que la phase d'état ne sera pas achevée (voir transfert de commande). Toutes les autres requêtes doivent être terminées avant la phase d'état.

Set Descriptor/Get Descriptor est utilisé pour renvoyer le descripteur indiqué dans **wValue**. Une requête pour le descripteur de configuration retournera le descripteur d'appareil et, tous les descripteurs d'interfaces et de terminaisons dans la même requête.

- Les descripteurs de terminaisons ne peuvent pas être accessibles directement par une requête de `GetDescriptor / SetDescriptor`.
- Les descripteurs d'interfaces ne peuvent pas être accessibles directement par une requête de `GetDescriptor / SetDescriptor`.
- Les descripteurs de chaînes incluent une identification de langues dans **wIndex** pour autoriser le support de plusieurs langues.

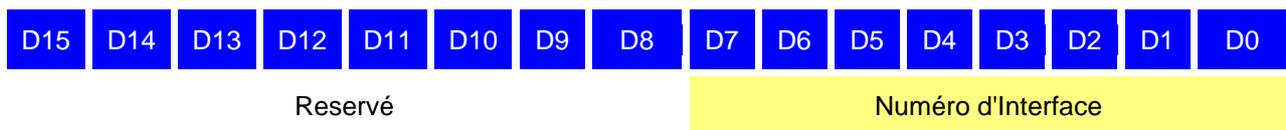
Get Configuration/Set Configuration est utilisé pour demander ou positionner la configuration de l'appareil actuel. Dans le cas d'une requête **GetConfiguration**, un octet sera renvoyé pendant la phase de donnée indiquant l'état de l'appareil. Une valeur zéro signifie que l'appareil n'est pas configuré et une valeur différente de zéro indique que l'appareil est configuré. **SetConfiguration** est utilisé pour valider un appareil. Il doit contenir la valeur de **bConfigurationValue** du descripteur de configuration voulu dans l'octet de poids faible de **wValue** pour sélectionner quelle configuration valider.

Les Requêtes d'Interface standard

La spécification actuelle définit 5 requêtes d'interface standard qui sont détaillées dans le tableau ci-dessous : Il est intéressant de noter que, seules 2 requêtes donnent quelque chose de compréhensible.

bmRequestType	bRequest	wValue	wIndex	wLength	Données
1000 0001b	GET_STATUS (0x00)	Zéro	Interface	Deux	Etat de l'Interface
0000 0001b	CLEAR_FEATURE (0x01)	Sélecteur de Fonction	Interface	Zéro	Aucune
0000 0001b	SET_FEATURE (0x03)	Sélecteur de Fonction	Interface	Zéro	Aucune
1000 0001b	GET_INTERFACE (0x0A)	Zéro	Interface	Un	Interface de remplacement
0000 0001b	SET_INTERFACE (0x11)	Positionnement alternatif	Interface	Zéro	Aucune

wIndex est normalement utilisé pour préciser l'interface de référence pour des requêtes liées à l'interface. Voir son format dans le schéma ci-dessous.



Get Status est utilisé pour retourner l'état de l'interface. Une telle requête à l'interface doit renvoyer 2 octets de valeur 0x00, 0x00. (Les 2 octets sont réservés pour une utilisation future).

Les requêtes **ClearFeature** et **SetFeature** peuvent être utilisées pour positionner des fonctions booléennes. Quand le destinataire désigné est l'interface, la spécification USB actuelle révision 2 précise qu'il n'y a pas de fonctions d'interface.

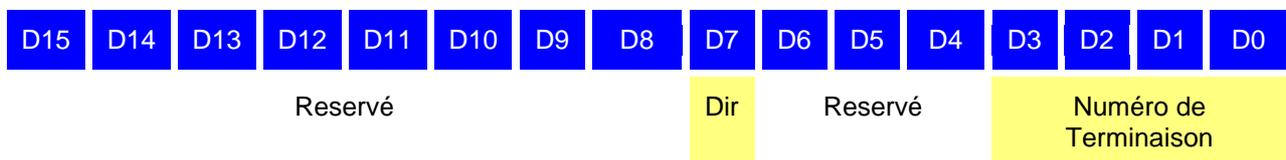
Get Interface et **Set Interface** règle le positionnement de l'interface de remplacement qui est décrit plus en détails dans le descripteur d'interface.

Les Requêtes de terminaisons standard

Les requêtes de terminaisons standard sont au nombre de 4, listées ci-dessous :

bmRequestType	bRequest	wValue	Windex	wLength	Données
1000 0010b	GET_STATUS (0x00)	Zéro	Terminaison	Deux	Etat de la Terminaison
0000 0010b	CLEAR_FEATURE (0x01)	Sélecteur de Fonction	Terminaison	Zéro	Aucune
0000 0010b	SET_FEATURE (0x03)	Sélecteur de Fonction	Terminaison	Zéro	Aucune
1000 0010b	SYNCH_FRAME (0x12)	Zéro	Terminaison	Deux	Numéro de Trame

Le champ **wIndex** est normalement utilisé pour préciser la terminaison de référence et la direction pour les requêtes liées à la terminaison. Voir son format dans le schéma ci-dessous.



GetStatus renvoie 2 octets indiquant l'état (Arrêté/Bloqué) d'une terminaison. Le format des 2 octets renvoyés est illustré ci-dessous.



ClearFeature et **SetFeature** sont utilisés pour positionner les fonctions de la terminaison. Le standard défini actuellement un sélecteur de fonction de terminaison : **ENDPOINT_HALT (0x00)** qui permet à l'hôte de bloquer et d'effacer une terminaison. Seules les terminaisons autres que la terminaison par défaut sont conseillées pour avoir cette fonctionnalité.

Une requête **SynchFrame** est utilisée pour rapporter une trame de synchronisation de terminaison.