



Descriptif de la pile USB Philog

La pile USB host *Philog* est totalement portable.

Pour simplifier les tâches d'adaptation à différents OS et plates-formes, toutes les fonctions dépendantes de l'OS ou du matériel sont isolées du code principal.

La pile est livrée sous forme binaire pour le programme principal et en code source pour le BSP et d'éventuels drivers. La licence est un droit d'usage et de copie illimitée.

Portage, maintenance et développements additionnels peuvent être assurés par *Philog*, notamment en ce qui concerne le développement de la partie DEVICE.

Caractéristiques :

- **USB 2.0**
Support de la norme USB 2.0, compatibilité avec les périphériques USB 1.1 garantie.
- **UHCI, OHCI, and EHCI Host Controllers**
Tous les contrôleurs host USB 1.1 and 2.0 EHCI, UHCI, OHCI sont supportés. Ainsi que les contrôleurs ISP1161, CY7C67300, CY7C67200.
- **HUB driver**
Chaque module, EHCI, OHCI ou autre intègre un driver pour les périphériques de classe HUB. L'API proposée par *Philog* permet de connaître la topologie exacte du bus et de piloter les ports des hubs unitairement.
- **Transfert en mode Control, Bulk, Interrupt, et Isochrone**
Tous les types de transferts peuvent être utilisés pour assurer un débit maximum avec les périphériques.
- **Low-, Full-, et High-Speed Data Transfer**
Les vitesses de 12 Mb/s sur USB 1.1 et 480 Mb/s sur USB 2.0 sont supportées.
- **Orientée embarqué**
Rapide et compacte : environ 30ko de RAM et 60ko de code par contrôleur. La pile USB génère son propre tick à 1ms.
Faible consommation en ressources temps réel : seuls 1 interruption, 3 sémaphores et 1 tâche sont nécessaires pour chaque contrôleur.
- **Plug-and-Play et Hot Plugging**
Enregistrement dynamique des drivers, gestion des connexions et retraits à chaud, supervision aisée de l'arborescence des périphériques.
- **Class Drivers**
La pile *Philog* comprend un driver de Host controller et un driver pour les périphériques de la classe HUB. Des drivers de classe sont aussi disponibles, soit sous forme de produits finis (CDC serial emulation, Still Image) ou sous formes d'implémentations partielles ne comprenant que la gestion de l'USB, l'interface haute étant développée à façon (Mass storage BOT, HID).



- **API de haut niveau**

Une librairie d'interface fournit aux développeurs de drivers de périphériques un accrochage simple sur la pile USB. Les drivers s'enregistrent dynamiquement et sont notifiés dès que leur périphérique est connecté.

- **Portabilité**

La pile USB ne nécessite qu'une simple tâche ou thread pour fonctionner. Tous les aspects de tâche, séquençement, temporisateurs... sont fondés sur des mécanismes internes. L'interruption 1ms du contrôleur est utilisée pour générer l'horloge interne de la pile. La pile USB est totalement écrite en C ANSI C et supporte les CPU little- et big-endian.