Utilisation des capacités du noyau USB

Par Gordon Lunn, Future Devices Technology International (FTDI)

L'USB (Universal Serial Bus, ou bus série universel) est considéré depuis longtemps comme l'un des piliers de l'environnement PC, dans la mesure où il permet d'interfacer des périphériques, des appareils grand-public, ou des équipements de stockage, à un PC hôte, tout en permettant de les alimenter. Plus récemment, un marché s'est développé au-delà de cette sphère, pour répondre à une demande d'intégrer des fonctionnalités USB à d'anciens systèmes embarqués, pour les doter de caractéristiques ou de fonctionnalités plus évoluées.

Cette prolifération d'USB dans les systèmes électroniques pour l'industrie, le médical, le commerce de détail ou le transport, signifie que de plus en plus d'ingénieurs sont confrontés aux problèmes de conception et d'implantation d'USB. Ils doivent décider de comment procéder au mieux pour déployer l'USB dans le cadre de leurs projets, tout en ayant très que peu de connaissances pratiques sur le sujet. Pourtant, cela peut potentiellement empiéter considérablement sur le produit final, en termes de coût, de lancement et de performances globales.

A première vue, simplement passer de la solution à microprocesseur précédente, à une autre avec des caractéristiques semblables mais dotée d'une capacité USB intégrée, ou s'appuyant sur un circuit associé adapté, semble être l'approche la plus évidente. Cependant, un re-design matériel ou logiciel peut s'avérer

nécessaire pour intégrer le nouveau microprocesseur ou le nouveau microcontrôleur dans le système. Cela peut également nécessiter la création de nouveaux drivers complexes, la mise en œuvre de processus de debugging complexes ou de procédures de test plus sophistiquées, qui se traduiront par un temps d'études supérieur à l'estimation initiale, un lancement retardé, et un coût global augmenté.

Une autre approche consiste à utiliser un dispositif semiconducteur plus spécialisé USB, qui soit non seulement conçu pour assurer une connectivité USB, mais aussi qui soit doté des capacités de traitement nécessaires pour se substituer au microprocesseur ou microcontrôleur de départ. L'utilisation d'un circuit USB hautes performances du commerce, et du firmware associé, permet de simplifier radicalement l'ajout de la fonctionnalité d'USB aux designs existants.

Conscients de cette tendance croissante vers une plus grande utilisation de l'USB au-delà des applications grand-public, certains fabricants de semiconducteurs tournés vers l'avenir ont commencé à produire des CI capables d'adresser ce segment du marché. C'est dans ce contexte que FTDI a été amené à lancer le contrôleur USB hôte/esclave haut débit à deux canaux, Vinculum II. En s'appuyant sur les retours de sa base client, l'équipe de conception de la société a choisi de développer une solution USB plus intégrée, offrant davantage de flexibilité, tout en permettant de réduire la nomenclature globale, d'accélérer la conception et le debugging, d'utiliser des piles logicielles intégrées (côté hôte et côté périphérique), et de réduire l'encombrement sur la carte.

La création d'un noyau propriétaire, par opposition à l'approche plus traditionnelle consistant à utiliser des blocs IP standard fournis par un fabricant connu, a permis d'obtenir plusieurs avantages. Le raisonnement derrière ce

choix était de réduire les besoins mémoire du système, et de libérer de la capacité de traitement pour d'autres tâches, améliorant ainsi les performances. Ainsi, la solution est bien mieux optimisée pour une application donnée, et tire le meilleur parti du silicium disponible, tout en s'affranchissant du coût de royalties pour obtenir un prix plus attractif. Quant au besoin de création de drivers par l'utilisateur, celui-ci a pu aussi être évité. Ceci est clairement intéressant, puisque cela permet d'économiser un nombre important de jours x hommes, et ainsi de réduire considérablement les efforts et les heures de bureau d'études.

Le coeur CPU 16 bits du contrôleur Vinculum II, conçu par FTDI, s'appuie sur l'architecture Harvard. Il dispose de 256 kO de mémoire Flash et de 16 Ko de RAM. Il est accompagné d'un RTOS très efficace, doté d'une structure de driver extensible. Un ensemble d'outils spécifique est utilisée au lieu d'outils génériques. En associant l'offre Vinculum II à un environnement de conception en C, les ingénieurs sont en mesure de développer leur propre micrologiciel d'application.

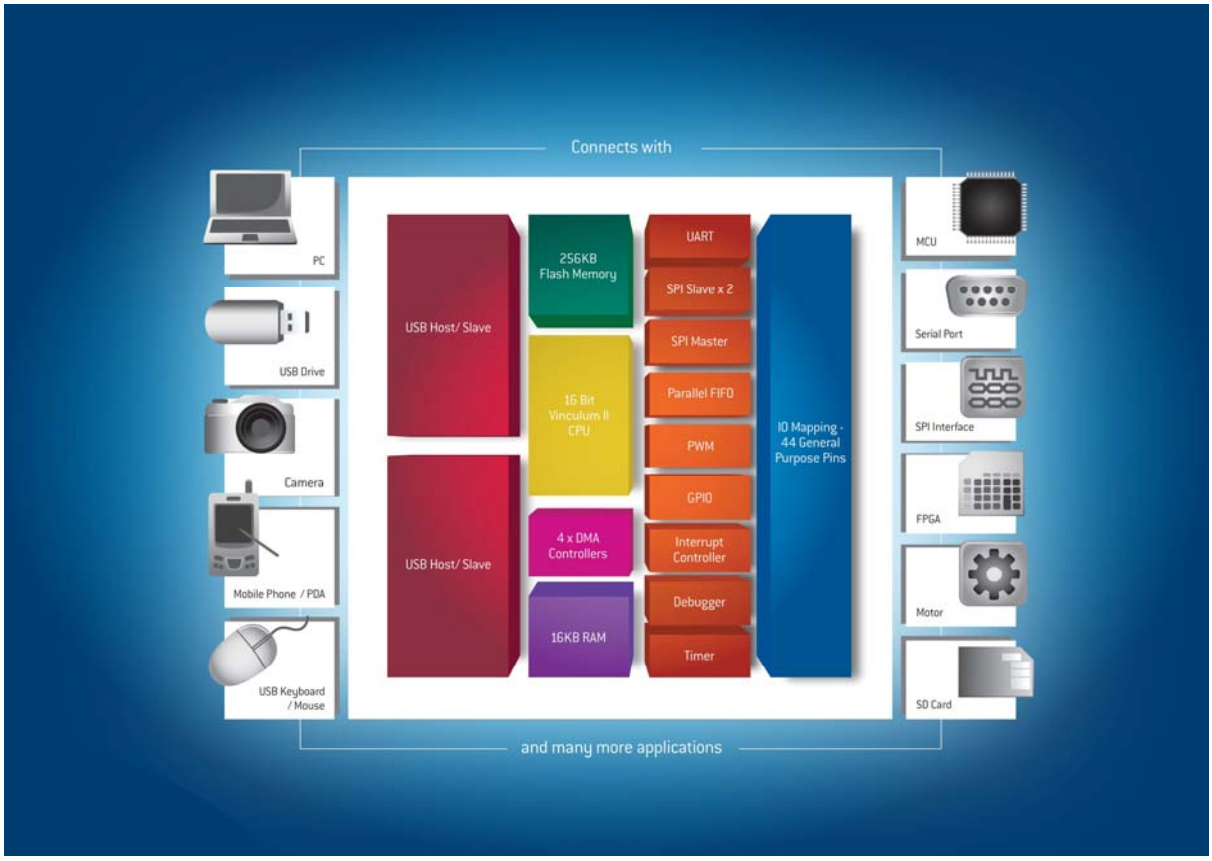

Figure 1: Schéma fonctionnel du contrôleur Vinculum II

L'approche intégrée, plus centrée sur l'USB, décrite dans cet article, présente sans aucun doute des avantages pour ceux qui souhaitent introduire la connectivité USB dans un ancien système, en permettant aux équipes techniques de surmonter plusieurs défis majeurs auxquels elles doivent faire face. Cette approche permet de réutiliser une grande partie du design initial, et d'éviter de perdre du temps en reprenant tout depuis le début. Elle permet aussi aux concepteurs de ne pas avoir à se préoccuper de modifications complexes. Les équipes peuvent élaborer leurs projets sans connaissance approfondie du logiciel et du matériel, ou des finesses théoriques de l'USB. Les concepteurs peuvent développer leur système selon une méthodologie classique : écriture, réalisation, debugging… sans se préoccuper de ce qui se passe "sous le capot". Cette solution USB flexible et performante, permet de réduire de manière significative le temps de développement et les coûts, en intégrant des piles

logicielles OS/Microcontrôleur libres de droits, tout en accélérant le lancement des produits.

En résumé, si l'on utilise un noyau universel, il faudra très souvent implanter les différentes fonctions USB spécifiques au niveau logiciel. L'approche noyau propriétaire décrite ici, permet d'implanter plusieurs de ces fonctions au niveau matériel.