

d'implémentation (malgré la généricité de certaines solutions). L'intégration de ces fonctionnalités nécessite le développement d'un support et souvent l'acquisition de ces connaissances de bout en bout. S'adapter à un système d'horloge d'un microcontrôleur AVR à un MSP430 demande déjà du travail. Imaginez le temps et l'énergie nécessaire s'il faut faire de même avec l'USB au plus bas niveau...

Il existe cependant une alternative : Linux. Avec un code excessivement audité et robuste, le noyau Linux intègre de manière fiable et documentée une masse très importante de fonctionnalités qui ne nécessitent alors que quelques ajustements pour être adaptée à une plateforme. Il n'est plus nécessaire de développer l'ensemble du système mais uniquement de le porter sur la nouvelle plateforme. Un problème est cependant à prendre en considération : le noyau Linux est « taillé » pour un cahier des charges minimal qui n'est pas celui auquel peut répondre un microcontrôleur 16 ou 32 bits comme un Cortex M. Un certain nombre d'éléments

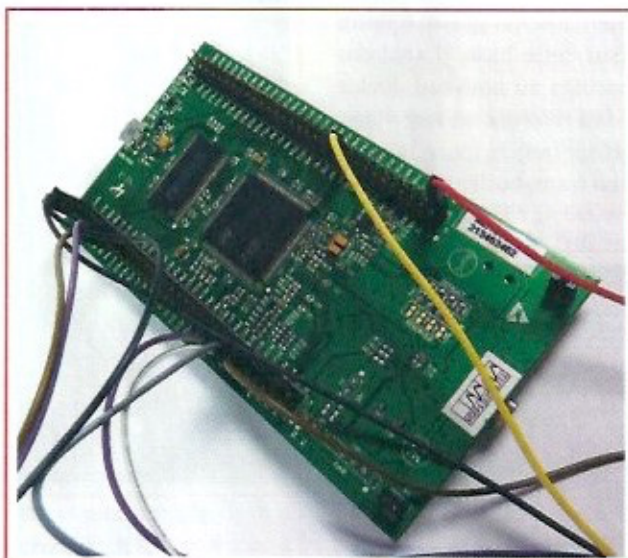
doivent, en effet, être présents comme une certaine quantité de mémoire ou encore certaines fonctionnalités du processeur comme la présence d'un MMU (*Memory Management Unit*).

C'est là le point de différence majeur entre Linux et μ Clinux. Ce dernier est destiné aux architectures dites *MMU-less*. μ Clinux a vu le jour en 1998 avec pour plateforme principale de démonstration, le processeur Motorola 68000 du PalmPilot III. Des portages Coldfire et ARM ont rapidement vu le jour, ainsi que le passage des développements d'une base Linux 2.0 à 2.4 puis 2.6. On retrouve ainsi μ Clinux dans bon nombre de produit du routeur à la caméra IP en passant par toutes la *appliances* multimédias.

μ Clinux intègre tout ce qui fait la force de Linux comme par exemple la pile IP, l'ensemble des systèmes de fichiers supportés de base, le sous-système USB, la facilité de portage des applications, une masse importante de bibliothèques, outils et applications, etc. Tout cela, sans support MMU, objet même de l'existence d' μ Clinux.

Attention cependant, car cela n'est pas une mince affaire sans MMU :

- Pas de mémoire virtuelle et donc une nécessité pour le noyau d'utiliser



d'autres mécanismes pour régler le problème de fragmentation mémoire.

- Pas de protection mémoire, tout est partagé. Une application peut utiliser et corrompre la mémoire du noyau ou celle des autres applications. Cette gestion repose ainsi plus que sur le développeur et la qualité de son code. Dans le cas d'un système embarqué, ce genre de chose est normalement de mise car une application qui *segfault* signifie de toutes façons des problèmes pour les utilisateurs.
- Pas de *swapping*. Là encore ceci n'est pas vraiment un problème étant donné que ce genre de fonctionnalité n'a pas lieu d'être dans l'embarqué (enfin... normalement).

- Pas de changement de contexte car pas de VM et d'espaces d'adressage séparés entre noyau et application. Ceci signifie plus un avantage qu'un handicap car synonyme de gain en rapidité.

- Pas d'appel système **fork** ou du moins pas dans le sens strict puisque **vfork** est disponible et, dans les grandes lignes, partage un fonctionnement très similaire au **fork** de Linux.

μ Clinux est donc parfaitement adapté si l'on cherche à disposer de toutes les fonctionnalités de Linux en contrepartie des limitations imposées qui, rappelons-le, sont également présentes avec d'autres OS pour une même plateforme modestement équipée.

Bien que ne bénéficiant pas d'une communauté aussi importante que celle de Linux, le projet μ Clinux est toujours actif. Ainsi, des mises à jour et correctifs sont apportés à chaque nouvelles version 2.6 du noyau Linux, même si le projet semble désynchronisé et que le site web officiel n'est plus vraiment tenu à jour. En réalité, bon nombre de développements se font maintenant hors du projet principal. Comme nous le verrons plus loin, la version pour STM32 qui nous intéresse ici est basée sur un μ Clinux maintenu par Emcraft Systems, elle-même basée sur un Linux 2.6.33 qui ne semble pas exister dans les dépôts officiels μ Clinux.

Attention cependant à ne pas tomber dans le travers séduisant de vouloir, quoi qu'il en coûte, disposer de Linux (ou μ Clinux) sur une plateforme qui ne dispose pas des ressources nécessaires. Il existe bien des microcontrôleurs chez bien des constructeurs et rester « bloqué » sur un cahier des charges ou une BOM trop rigide n'est généralement pas une bonne chose. S'il vous reste 64 Ko de mémoire pour vos