

Une nouvelle direction pour l'économie d'énergie

Andreas Koller, VP Marketing et Ventes, Energy Micro, www.energymicro.com

Pour qu'un équipement puisse fonctionner sur batterie pendant plusieurs années voire plusieurs décennies, atteindre les plus hauts niveaux d'efficacité énergétique est impératif. Pour cela, il ne suffit pas d'améliorer de façon marginale le degré d'intégration du microcontrôleur (MCU), ou d'apporter des modifications mineures à l'architecture de base du processeur. Des évolutions radicales dans la conception du MCU sont indispensables si l'on souhaite satisfaire les exigences de performances et de consommation, souvent antinomiques, imposées par les concepteurs de produits éco-énergétiques sur des secteurs comme le comptage d'eau, de gaz ou d'électricité, la domotique, la sécurité ou les équipements médicaux portatifs.

La société Energy Micro s'est dégagée de tout a priori pour concevoir le microcontrôleur basse consommation EFM32 Gecko ainsi que les outils logiciels et matériels associés (Fig. 1). Elle a ainsi pu développer un composant qui consomme un quart seulement de l'énergie nécessaire aux traditionnels MCU 8 bits, 16 bits et 32 bits, et qui garantit un allongement considérable de la durée de vie des batteries actuelles. En d'autres termes, les concepteurs d'équipements peuvent désormais, avec un MCU aussi peu gourmand en énergie, réduire le coût et/ou les dimensions des batteries. Dans le cas de produits comme les systèmes de sécurité ou les compteurs d'énergie, la fréquence, le coût et l'empreinte carbone des opérations de maintenance peuvent également être minimisés.

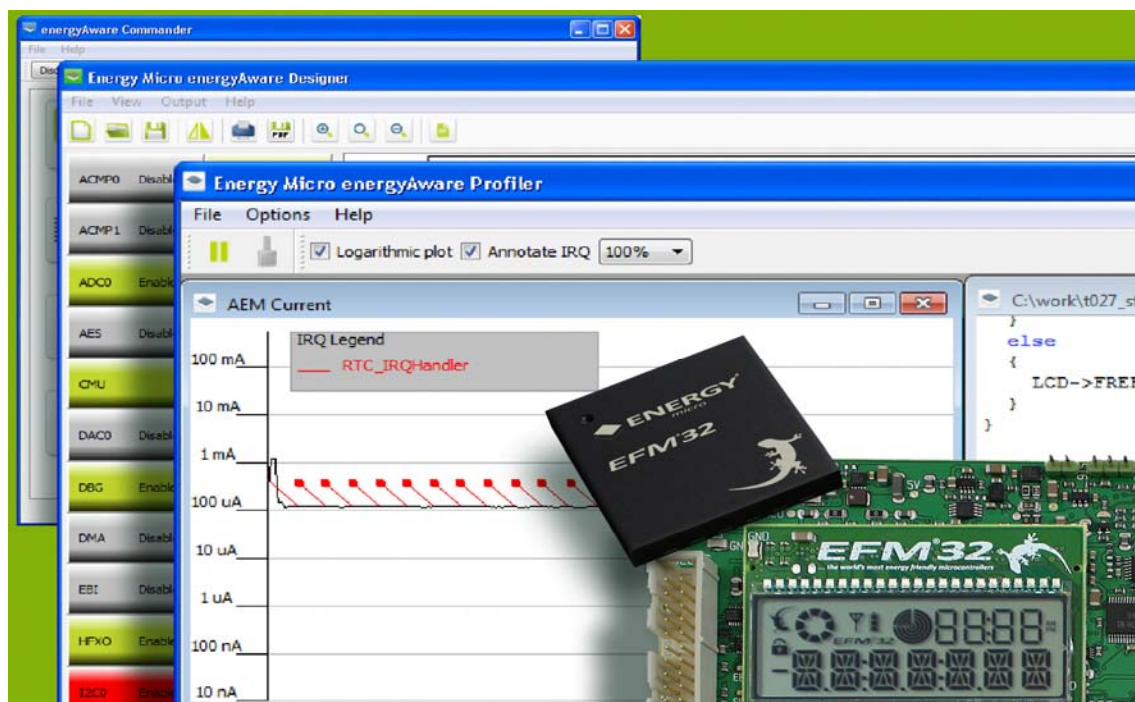


Figure 1: Le microcontrôleur EFM32 Gecko et l'environnement Simplicity Studio d'Energy Micro sont tous deux taillés pour réduire la consommation d'énergie.

Doter un MCU de telles références en matière de basse consommation n'a rien d'une tâche aisée. Il a fallu des années de développement et de réelles innovations pour atteindre cet objectif. Sur le site Web d'Energy Micro, vous pouvez consulter une description des technologies mises en œuvre au sein de l'EFM32 sous le titre « Les 10 facteurs qui font de l'EFM32 32 bits le microcontrôleur le plus éco-énergétique au monde ». En réalité, il en existe certainement bien plus de dix. Laissons de côté les batailles de spécifications liées à « l'ultra basse consommation ». Toute batterie ayant une charge électrique limitée, c'est la manière dont le MCU va utiliser cette énergie dans le temps qui importe. Il faut que l'énergie consommée (produit de la puissance consommée par la durée de fonctionnement) soit minimisée, aussi bien pendant les périodes de veille que pendant les périodes d'activité du microcontrôleur.

Architecturé autour d'un cœur de processeur ARM Cortex-M3, le MCU EFM32 a été conçu pour que sa consommation à l'état actif soit réduite de manière significative. Selon les résultats de benchmarks, un EFM32 cadencé à 32MHz et alimenté en pratique sous 3V peut correctement exécuter un code applicatif à partir de la mémoire flash en consommant seulement 180μA/MHz.

Tout ceci est parfait, mais le temps nécessaire au MCU pour exécuter une tâche a aussi un impact décisif sur l'efficacité énergétique. D'où l'usage du Cortex-M3 32 bits. Comme ce cœur est doté d'une capacité de calcul plus importante que les circuits 8 bits et 16 bits, il suffit d'un nombre bien plus faible de cycles d'horloge pour accomplir les tâches, et les périodes d'activité s'avèrent par conséquent nettement plus courtes. En réduisant au strict minimum ces périodes d'activité, le MCU 32 bits peut être plongé plus longtemps dans des modes sommeil profond.

Il faut également oublier le vieil adage qui veut que les processeurs 32 bits soient incapables de rester en mode veille en consommant moins de 1μA. Cette assertion est généralement battue en brèche si l'on applique des techniques de conception basse consommation appropriées. En consommant 0,9μA seulement en mode sommeil profond, l'EFM32 est capable de fournir toutes les fonctions de base comme l'horloge temps réel, le mode rétention pour la RAM et le CPU, la détection de sous-tensions (BOD, Brown-Out Detection) et la réinitialisation à la mise sous tension (POR, Power-On Reset).

Comme c'est souvent le cas dans les applications cibles susmentionnées, les périodes d'activité du MCU sont extrêmement courtes et le microcontrôleur peut rester en mode sommeil profond pendant 99% du temps. C'est pourquoi la consommation de courant durant les phases d'inactivité est un aspect particulièrement important à prendre en compte dans l'amélioration de l'efficacité énergétique globale.

Cet avantage est néanmoins perdu si la durée de réveil du MCU, c'est-à-dire le temps nécessaire pour passer du mode sommeil profond au mode actif, est trop importante. La raison en est simple. Pendant ce laps de temps, il existe toujours une période de réveil où le processeur doit attendre la stabilisation des oscillateurs et du système d'alimentation avant de démarrer l'exécution du code. Comme aucun traitement n'est possible pendant ce temps, l'énergie consommée durant la phase de réveil est de l'énergie perdue. Il est donc indispensable de réduire la durée de cette phase de réveil pour diminuer la consommation d'énergie globale.

De plus, les applications MCU affichent des contraintes temps réel. Par conséquent, la durée de réveil doit être généralement réduite au minimum afin que le MCU puisse répondre à un événement extérieur dans un laps de temps donné. Comme le temps de réponse exigé par la plupart des applications est inférieur à la durée de réveil de nombreux MCU actuels, tout est souvent fait pour que le système n'entre jamais en mode sommeil profond. Une solution guère adaptée aux besoins des applications éco-énergétiques.

Pour satisfaire cette contrainte, l'EFM32 met en œuvre un ensemble de techniques de conception qui réduisent à seulement 2µs le temps de réveil à partir d'un mode sommeil profond. Le niveau d'énergie dépensée avant que le CPU ne s'attaque au traitement des tâches est donc réduit au minimum. S'il est indispensable de contrôler et d'optimiser pleinement les économies d'énergie, les concepteurs de systèmes ont également besoin de la flexibilité que peut offrir le choix entre plusieurs modes de consommation bien pensés. Comme le montre le Tableau 1, l'EFM32 dispose de plusieurs modes de fonctionnement qui permettent aux concepteurs d'ajuster à tout moment les ressources utilisées afin de maximiser l'efficacité énergétique.

<i>EFM32 with 3V power supply running real application from Flash memory</i>	EM0 Run Mode	EM1 Sleep Mode	EM2 Deep Sleep Mode	EM3 Stop Mode	EM4 Shutoff Mode
Current consumption	180 µA/MHz	45 µA/MHz	0.9 µA	0.6 µA	20 nA
Wake-up time	-	0	2 µs	2 µs	160 µs
Wake-up events	Any	Any	32 kHz peripherals	Async IRQ, I2C slave, Analog Comparators, Voltage Comparator	Reset
CPU	On	-	-	-	-
High frequency peripherals	Available	Available	-	-	-
Low frequency peripherals	Available	Available	Available	-	-
Asynchronous peripherals	Available	Available	Available	Available	-
Full CPU and SRAM retention	On	On	On	On	-
Power-on Reset/Brown-out Detector	On	On	On	On	On

Tableau 1: Les différents modes de consommation de l'EFM32 contribuent à maximiser l'efficacité énergétique.

Certes, aux yeux de certains observateurs, ces modes de consommation peuvent apparaître comme définis de manière grossière. Mais, au sein de chacun de ces modes, il est possible d'ajuster finement les ressources en activant ou en désactivant différents périphériques. D'une façon ou d'une autre, les modes de consommation de l'EFM32 contribuent à éviter le moindre gaspillage d'énergie.

Evidemment, au sein d'un MCU basse consommation, les blocs de fonctions périphériques doivent aussi être conçus pour consommer le moins possible... et l'EFM32 ne constitue pas une exception à la règle. Le microcontrôleur intègre par exemple un CAN 12 bits huit canaux qui consomme 350µA à pleine résolution pour une fréquence d'échantillonnage de 1Méch./s. L'EFM32 embarque par ailleurs un contrôleur LCD 4x40 segments qui se satisfait de seulement 550nA avec fonctions clignotement, animation, contraste et élévateur de tension (voltage boost) intégrées. Le MCU dispose aussi d'une UART à basse énergie, une véritable UART avec horloge 32kHz consommant seulement 150nA pour un débit de 9600 bauds.

Pour réduire encore la consommation, une autre innovation de taille dans la conception du MCU consiste à laisser les périphériques fonctionner de manière autonome sans intervention du CPU. Les périphériques de l'EFM32 ont été pensés en ce sens afin que le CPU puisse se concentrer sur des tâches de haut niveau ou, tout bonnement, se placer en mode veille. Dans les deux cas, l'économie d'énergie est au rendez-vous.

L'EFM32 pousse cette notion d'autonomie encore plus loin. L'architecture du MCU intègre en effet une structure d'interconnexion programmable, du nom de Peripheral Reflex System (Fig. 2), qui permet aux périphériques de communiquer entre eux sans intervention du CPU, caractéristique synonyme d'une réduction supplémentaire de la consommation.

Mais un MCU particulièrement éco-énergétique ne garantit pas à lui seul le niveau de consommation d'énergie le plus faible qui soit. Disposer d'outils adéquats, capables d'identifier et de supprimer les pertes d'énergie dès les premières étapes de prototypage, peut contribuer grandement à réduire la consommation globale du produit final.

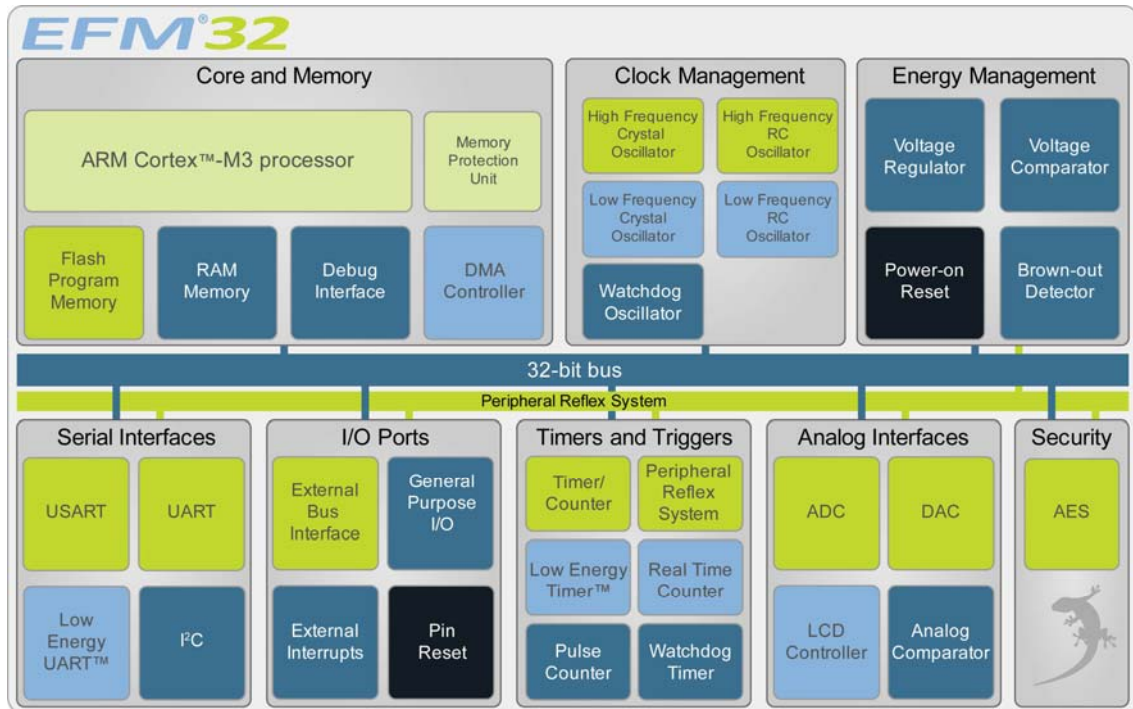


Figure 2: L'architecture du microcontrôleur EFM32 Gecko avec son système d'interconnexion original de périphériques Peripheral Reflex System. A l'occasion du salon Electronica 2010, Energy Micro a dévoilé son intention de développer Simplicity Studio, un environnement de développement graphique complet pour les microcontrôleurs EFM32. Cet environnement fournira un accès à toutes les

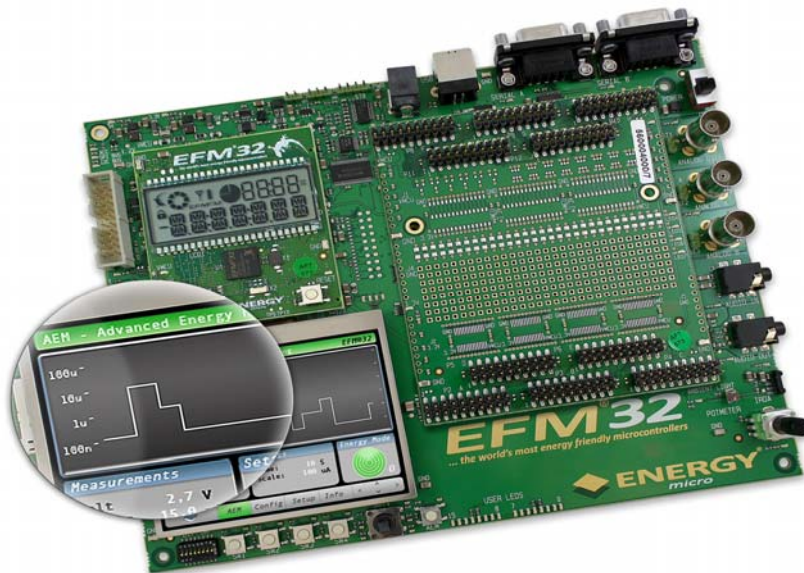


Figure 3: Le kit de développement du microcontrôleur EFM32 Gecko intègre l'outil de gestion évoluée de la consommation AEM (Advanced Energy Monitoring).

informations, toute la documentation et tous les outils dont les ingénieurs en charge du matériel, du firmware et du logiciel ont besoin pour développer des systèmes embarqués de façon plus efficace et plus rapide. La plupart de ces outils sont déjà disponibles. Les kits de développement de l'EFM32 sont fournis avec un système de gestion évoluée de la consommation (AEM, Advanced Energy Monitoring), un outil qui mesure en continu le courant électrique consommé. Ces mesures sont exploitées pour décrire avec précision la consommation au cours du temps et optimiser des cas d'usage pratiques en fonctionnement à basse consommation (Fig. 3).

Utilisé avec l'outil logiciel de « débogage de l'énergie » energyAware Profiler, l'outil AEM permet à l'utilisateur d'identifier sur une courbe d'énergie le code source en cours d'exécution à un instant donné. Avec cette fonction, l'ingénieur peut donc repérer toute partie du programme qui induit une forte consommation d'énergie, optimiser en conséquence le code, et gérer les économies d'énergie d'une manière plus précise que jamais.

www.energymicro.com