

Optimiser la consommation d'énergie grâce au silicium dans les applications de détection sur microcontrôleur

Rasmus Christian Larsen, Directeur Support & Formation, Energy Micro

La conception d'une architecture basse consommation est un exercice qui nécessite une bonne compréhension du logiciel, de la plate-forme matérielle et de l'application afin de minimiser la consommation énergétique d'un système embarqué. Il est en effet facile de gaspiller de l'énergie si l'on ne trouve pas le juste équilibre entre les avantages et inconvénients respectifs d'un traitement essentiellement logiciel (et donc flexible) et une implantation matérielle, plus contraignante certes, mais souvent plus éco-efficace. D'autant que de subtiles modifications de l'implantation matérielle peuvent engendrer des variations importantes de la consommation énergétique globale.

La conception d'un microcontrôleur (MCU) basse consommation vise généralement à maintenir son activité à un niveau faible, afin que le processeur au cœur du système passe le plus clair de son temps en mode veille. Cette approche a prouvé sa grande efficacité au sein de compteurs d'eau, de gaz ou d'électricité, là où le cœur de processeur peut rester en mode veille pendant 99% du temps sur la totalité de son cycle de vie. Le processeur se réveille uniquement pour récupérer les données fournies par les capteurs, habituellement à des moments préprogrammés, ou pour réagir à de rares interruptions non prévues.

L'importance du niveau d'activité peut être appréciée par une analyse de la variation de consommation d'énergie d'un MCU en fonction de ses besoins énergétiques en mode veille. Le temps est en effet l'ennemi de toute application alimentée par pile dont l'autonomie doit être élevée, car même des niveaux de consommation apparemment faibles deviennent significatifs sur de longues périodes d'activité.

Durant les périodes d'activité, la puissance consommée par toute fonction logique est donnée par la formule CV^2f , où C est la capacité totale des chemins de données dans le circuit, V la tension d'alimentation et f la fréquence de fonctionnement. La formule s'est légèrement complexifiée à cause de la désactivation d'horloge (clock gating) dont l'usage s'est généralisé dans la plupart des cœurs de processeur modernes. Cette technique limite les activités de commutation dans les portes logiques non utilisées en désactivant le signal d'horloge dans ces circuits. Néanmoins, même approximative, la formule donne une bonne idée de la puissance consommée à l'état actif.

Les MCU basse consommation sont habituellement implantés avec des procédés de fabrication à faibles courants de fuite. Ces courants de fuite n'en constituent pas moins un élément de pertes énergétiques que l'on doit prendre en compte dans la consommation, pertes dont la valeur peut s'avérer significative pour tout bloc logique actif pendant une longue période de temps. La puissance dissipée par ces courants de fuite affiche une valeur constante, mais celle-ci tend à augmenter avec la température du fait des porteurs générés thermiquement dans les transistors. Au-delà du choix précautionneux d'un procédé de fabrication, le seul moyen de réduire cette puissance perdue est donc de désactiver le bloc logique.

Afin de fournir aux concepteurs divers moyens de réduire la consommation d'énergie, les MCU basse consommation disposent de plusieurs modes veille qui permettent de désactiver les différents blocs du composant les uns après les autres, après stockage de leurs états en mémoire non volatile ou dans des registres dédiés à faibles pertes, et ce jusqu'à extinction pratiquement complète du MCU. A titre d'exemple, un microcontrôleur EFM32 Gecko d'Energy Micro consomme 5mW en mode actif pour une fréquence d'horloge de 10MHz et une tension d'alimentation de 3V. Dans le mode veille le plus profond, le même composant consomme seulement 60 nW – soit une énergie d'un ordre de grandeur pratiquement six fois inférieur.

Pour déterminer la consommation d'énergie globale, il faut également considérer le coût énergétique du temps de réveil. Un certain laps de temps est en effet nécessaire pour que se stabilisent les boucles à verrouillage de phase (PLL) qui délivrent un signal d'horloge régulier au cœur, et pour que les contextes soient complètement restaurés. Dans cet intervalle de temps, les besoins en énergie vont augmenter graduellement, mais les parties « réveillées » du composant ne seront pas aptes à effectuer leurs tâches correctement tant que leur alimentation et leur contexte ne sont pas complètement rétablis. Dès lors, il est important de prendre en compte la durée de cette période de réveil lorsque l'on sélectionne un MCU basse consommation.

Lorsque le rapport cyclique est de 1%, c'est-à-dire lorsque le composant cadencé à 10MHz passe en mode actif durant 10ms toutes les secondes avec un temps de réveil de 2ms, un MCU donné pour 200 μ A/MHz - et doté d'un mode sommeil profond consommant 2 μ A - peut fonctionner pendant 12,7 années avec deux piles AA. Si le rapport cyclique passe à 2%, la durée d'autonomie sera réduite à sept ans seulement. Cependant, il faut bien avoir en tête la manière dont s'organise le cycle d'activité. Si l'objectif est d'améliorer le temps de réponse aux événements en réveillant le cœur du MCU deux fois par seconde, la durée de réveil prise dans sa globalité va doubler et réduire la durée d'autonomie des piles d'au moins six mois. Par contre, réduire le temps de réveil à seulement 2 μ s peut prolonger la durée de vie de plus d'un an, en gardant l'hypothèse d'un réveil du système deux fois par seconde.

[Fig 1]

Il apparaît donc clairement qu'il faut maintenir un faible rapport cyclique avec un temps de réveil réduit. Mais il n'est pas toujours possible de prévoir à quel moment les capteurs vont détecter un événement important. Prenons le cas, par exemple, d'une détection capacitive, technologie désormais répandue dans de nombreux secteurs industriels. Les capteurs capacitifs à hautes performances peuvent mesurer avec une résolution élevée la proximité, la position ou le déplacement d'un objet, l'humidité, le niveau d'un fluide ou l'accélération de n'importe quel objet conducteur. Depuis l'introduction par Apple de l'iPhone, les fabricants de produits grand public reconnaissent l'intérêt d'intégrer des dalles tactiles capacitives dans leurs systèmes pour offrir des interfaces utilisateur plus réactives et plus intuitives. Ces capteurs sont généralement des produits peu onéreux, mais ils exigent une puissance de calcul non négligeable au niveau du MCU, ce qui peut s'avérer difficile à implanter concrètement dans un système où la consommation doit rester à un niveau minimal.

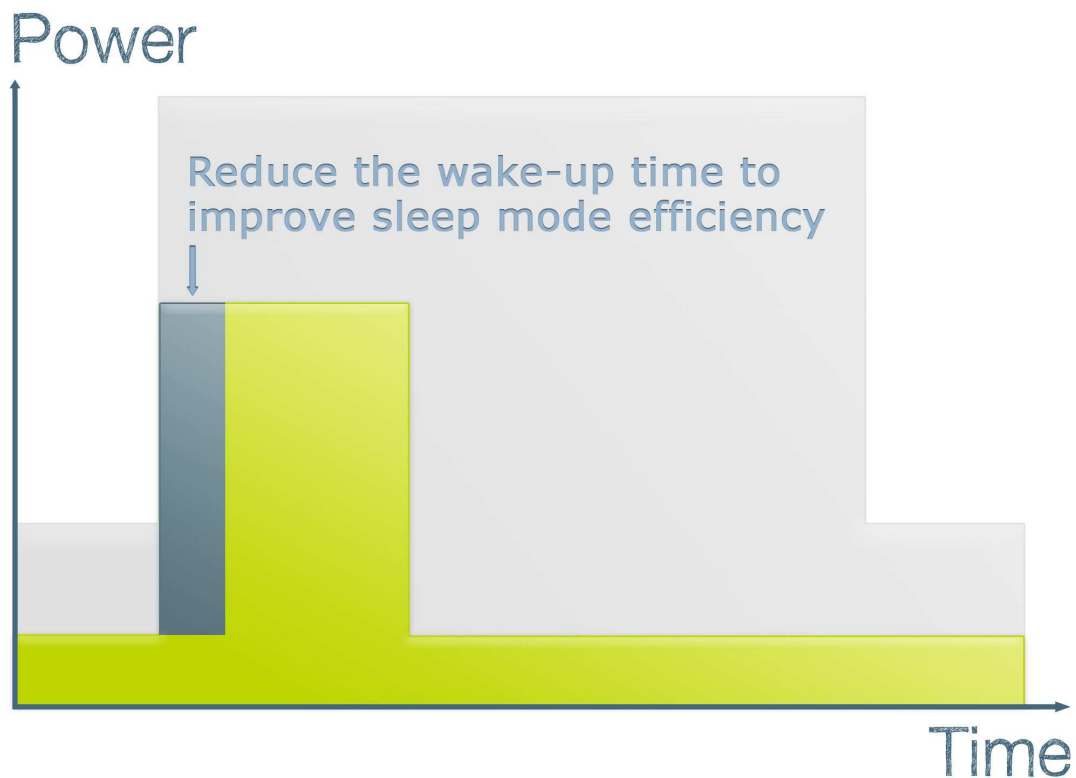


Figure 1: Période où le processeur est inefficace, le passage entre mode sommeil profond et mode actif ne peut être purement et simplement négligé. La réduction du temps de réveil permet de minimiser l'énergie consommée avant le moment où le CPU est enfin prêt à effectuer ses tâches.

La capacité électrique est une valeur qui décrit comment l'espace séparant deux conducteurs affecte le champ électrique qui s'établit entre eux. Lorsque deux électrodes métalliques sont placées l'une en face de l'autre et séparées par un espace, et lorsqu'une tension est appliquée sur l'une des électrodes, il se crée un champ électrique entre elles. Ce champ électrique est dû à la différence de charges électriques stockées sur la surface des électrodes. La quantité de charge détermine la valeur de l'intensité du courant qui doit être utilisée pour modifier la tension sur l'électrode.

Dans le cas où un capteur capacitif est implanté dans une interface utilisateur, c'est la surface de détection de la sonde qui forme l'électrode sous tension, tandis que le doigt (ou le stylet) de l'utilisateur constitue l'autre électrode. L'électronique de contrôle va alors faire varier en continu la tension de la surface de détection. C'est ce que l'on appelle la tension d'excitation. L'intensité du courant électrique requise pour modifier cette tension est mesurée par le circuit de détection et indique la valeur de la capacité électrique entre la sonde et la cible. Inversement, une quantité fixe de courant électrique peut être alternativement injectée puis pompée dans et hors de la sonde et ce sont les variations de la tension résultantes qui sont alors mesurées.

Dans les implantations typiques, le circuit de détection intègre un oscillateur RC libre qui génère une tension d'excitation variant à une fréquence plus ou moins constante. Lorsque le doigt d'un utilisateur s'approche du capteur, la fréquence de résonance globale du montage va être modifiée du fait de l'augmentation de la capacité électrique de l'élément de détection. Le MCU détecte cette variation de fréquence par le biais d'un

compteur. Si la valeur du compteur diffère significativement de la valeur nominale après un laps de temps donné, l'algorithme considérera qu'un utilisateur a touché l'écran.

L'algorithme de mesure nécessite donc deux composants principaux: un timer et un compteur. Le timer est tout d'abord programmé par le MCU pour décompter à partir d'une valeur donnée. Une fois arrivé à échéance, il génère une interruption vers le MCU. Pendant ce laps de temps, le compteur du MCU est incrémenté à chaque oscillation positive de l'oscillateur. Au moment où il reçoit l'interruption du timer, le MCU doit simplement vérifier la valeur du compteur pour déterminer si l'utilisateur a effectivement touché l'écran.

L'inconvénient de ce type d'implantation tient au fait que le cœur de processeur doit rester en mode actif durant toute la phase de détection afin de répondre aux interruptions et gérer le compteur. Pendant ce temps, le MCU peut n'avoir quasiment aucune autre tâche à effectuer, à part exécuter éventuellement une boucle à vide. Un moyen de réduire la consommation est de réveiller le MCU de manière périodique, le temps simplement que celui-ci sonde la dalle tactile.

Mais, dans le cas d'une interface utilisateur, le sous-système de détection ne peut pas prévoir le moment où l'utilisateur touchera une nouvelle fois l'écran. Bien que le microcontrôleur hôte puisse être plongé régulièrement en mode veille, des périodes de veille prolongées risquent d'affecter négativement le temps de réponse. De fait, cette option risque de réduire la fiabilité d'usage du produit. Tant que le système n'est pas complètement repassé en mode actif et n'est pas dans un état lui permettant d'effectuer des mesures précises, il se peut en effet que des touches ne soient pas correctement détectés.

[Fig 2]

D'un point de vue énergétique, il serait plus efficace de confier ces opérations au matériel et de réveiller le MCU uniquement lorsqu'un toucher a été dûment enregistré. En effet, l'usage de circuits dédiés ne génère pas une telle dépense énergétique. Un processeur consomme beaucoup plus d'énergie à transférer instructions et données vers les unités d'exécution qu'à effectuer les calculs proprement dits. Des chercheurs travaillant sur le projet de processeur basse consommation ELM de la Stanford University ont mesuré que les transferts d'instructions, à eux seuls, étaient responsables de 42% de la consommation électrique totale d'un processeur. Il faut également tenir compte des besoins en énergie nécessaires aux échanges de variables entre la mémoire et les registres, ainsi qu'aux allers-retours sur le chemin de données. Ces transferts de données, qui s'effectuent en général via des bus à haute capacité et qui nécessitent donc plus d'énergie, n'existent pas dans les implantations matérielles.

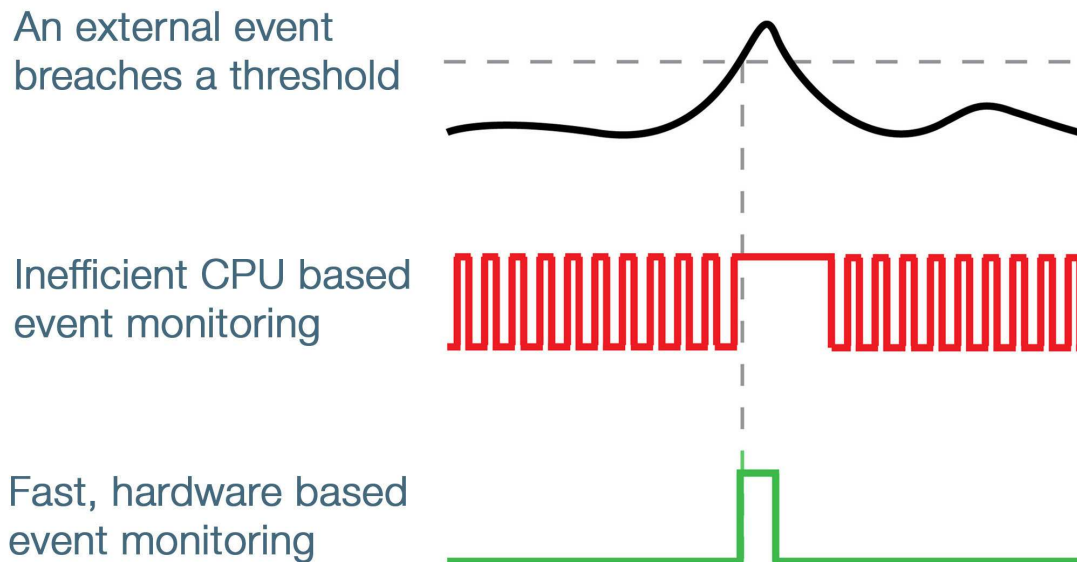


Figure 2: Comme la plupart des événements surgissent de manière asynchrone, les systèmes embarqués devraient être capables de réagir uniquement lorsque des niveaux de seuil sont atteints et sans qu'il soit besoin de gaspiller inutilement de l'énergie à interroger les capteurs à intervalles réguliers.

Dans le cas d'un sous-système de détection capacitive, on peut réaliser un circuit qui excite périodiquement l'oscillateur, déclenche un timer, puis décompte le nombre d'impulsions avant l'échéance de ce timer. Au terme du processus, si la valeur du compteur se situe sous un minimum préalablement fixé, le circuit peut générer une interruption qui réveille le MCU. Dans ce type d'implantation, il est inutile que le MCU passe régulièrement en mode actif, à moins qu'il n'ait besoin d'effectuer d'autres fonctions de gestion. Dès lors, le MCU sera actif uniquement au moment où il aura des tâches à réaliser, d'où une réduction de la consommation au minimum envisageable, compte tenu des actions de l'utilisateur.

Les capteurs inductifs posent le même genre de problème au niveau de la prévisibilité du processus de détection. Ces capteurs électroniques de proximité, qui sont capables de détecter la présence d'un objet conducteur, sont utilisés dans des applications classiques comme les détecteurs de métaux, les feux tricolores, les lave-autos et de multiples applications industrielles automatisées. Comme aucun contact n'est nécessaire, les capteurs inductifs sont notamment utilisés dans des environnements opérationnels difficiles où une poussière omniprésente risque d'obturer des capteurs de proximité optiques ou infrarouges.

Egalement appelés capteurs à courants de Foucault, les capteurs inductifs à hautes performances peuvent mesurer avec une résolution élevée la position et/ou les changements de position de n'importe quel objet conducteur. Les capteurs inductifs les moins chers s'utilisent comme des commutateurs de proximité qui délivrent un simple

signal on-off en présence ou non d'un objet conducteur. En termes de consommation énergétique, ce sont ces derniers qui posent le plus de problème au moment de l'implantation de l'algorithme de détection sur un MCU.

Les capteurs de proximité inductifs détectent les pertes magnétiques dues au courant électrique induit qui est généré sur une surface ou un objet conducteur par un champ magnétique externe. Lorsqu'un courant AC parcourt une bobine, celle-ci génère un champ magnétique alternatif. Si un objet conducteur passe à proximité du capteur, le champ magnétique crée des courants électriques sur cet objet à détecter, les fameux courants de Foucault.

L'un des moyens de créer une oscillation dans une inductance consiste à utiliser un circuit résonnant formé d'une bobine d'induction et d'un condensateur, de manière un peu similaire au circuit RC utilisé dans la détection capacitive (la bobine d'induction en plus).

Le condensateur stocke l'énergie sous la forme d'un champ électrique tandis que la bobine la stocke sous la forme d'un champ magnétique. Lorsque l'interrupteur est dans sa position initiale, le condensateur est chargé sous la tension d'alimentation DC. Une fois celui-ci entièrement chargé, l'interrupteur est placé dans sa seconde position. Le condensateur se retrouve alors monté en parallèle avec la bobine d'induction et se décharge à travers cette bobine. La tension aux bornes du condensateur commence à diminuer tandis que l'intensité du courant électrique qui parcourt la bobine commence à augmenter. Cette variation d'intensité génère un champ magnétique autour de la bobine. Lorsque le condensateur est entièrement déchargé, l'énergie qui y était préalablement stockée est désormais stockée dans la bobine d'induction.

Comme aucune tension externe n'est appliquée au circuit pour maintenir le courant électrique dans la bobine, le courant rebrousse chemin vers le condensateur qui va se retrouver chargé avec une polarité inverse à celle de la charge électrique initiale. Le cycle se répète alors continuellement avec un transfert d'énergie périodique entre les deux éléments du circuit. La polarité de la tension varie en fonction du sens des transferts d'énergie entre l'inductance et la capacité, générant de ce fait une tension et un courant alternatifs.

A chaque fois qu'il y a transfert d'énergie entre les deux éléments du circuit, des pertes apparaissent et les oscillations sont amorties à cause des composants résistifs du circuit qui dissipent l'énergie petit à petit. L'amplitude de l'oscillation décroît à chaque demi-cycle jusqu'à ce que l'énergie du circuit soit entièrement dissipée. Si un objet métallique est situé près de la bobine, ce phénomène d'amortissement est encore accéléré. En décelant cette variation d'amortissement, un algorithme de détection exécuté par le MCU peut alors déterminer la présence d'un objet dans la zone environnant le capteur. Cette action s'effectue habituellement en mesurant à intervalles réguliers une tension et en la comparant à une valeur de référence. Si la tension d'entrée passe sous cette valeur de référence, le MCU peut déclencher un événement dit de "proximité".

Comme dans le cas de la détection capacitive, une implantation logicielle nécessite ici que le processeur soit en mode actif pour générer le signal d'excitation et traiter les signaux d'entrée analogique du comparateur. Dans le cas d'un compteur d'eau ou de gaz, déterminer la fréquence de réveil du MCU pour que ce dernier prenne en compte le

moindre tour de turbine, peut toutefois s'avérer ardu. Par ailleurs, comme les robinets de gaz ou d'eau peuvent éventuellement rester fermés pendant de longues périodes, le MCU risque de passer en mode actif... pour s'apercevoir qu'aucune donnée utile n'est à traiter. [Fig 3]

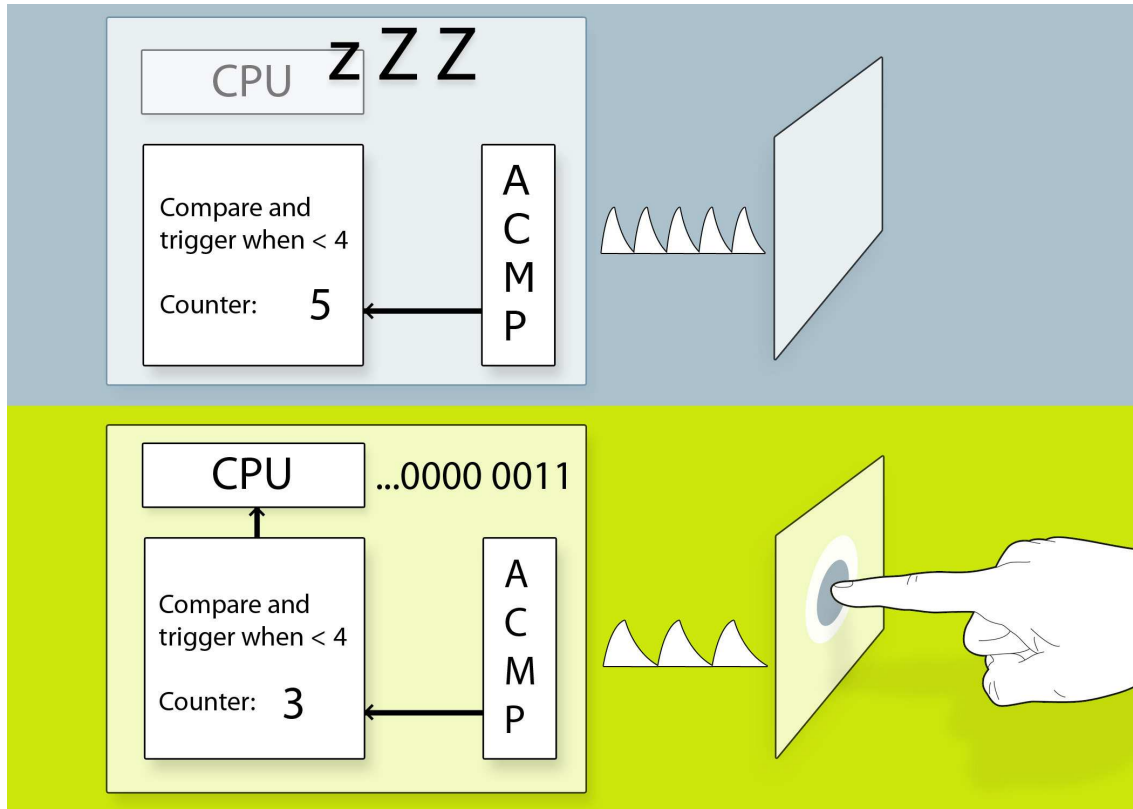


Figure 3: Le bloc matériel LESENSE du microcontrôleur est chargé de superviser le capteur capacitif de telle sorte que le CPU n'ait pas à se réveiller avant que la valeur du compteur ne passe sous le seuil de déclenchement du comparateur. Ainsi le CPU peut rester plus longtemps plongé en mode sommeil profond.

Intégrer le circuit de détection dans le silicium peut grandement réduire la consommation énergétique du système. Mais les composants matériels nécessaires aux détecteurs inductifs s'avèrent légèrement différents de ceux utilisés dans les détecteurs capacitifs. Certes, on y retrouve de nombreuses fonctions comme le comparateur analogique qui supervise la tension de sortie du capteur, ou les timers et compteurs nécessaires au calcul des temps de mesure et à la détection de proximité. Mais le capteur inductif requiert un convertisseur numérique/analogique (CNA) supplémentaire afin de fournir un niveau de tension autour duquel peut osciller le capteur. Il impose également un montage des composants différents. Dès lors, la conception d'un MCU devant supporter les deux types de détection risque de générer un surcoût.

Les similitudes entre les deux types d'interfaces de capteurs peuvent néanmoins être exploitées par une conception soignée du sous-système matériel. La clé ici est de bien comprendre que les principales différences tiennent à la façon dont sont interconnectés les différents composants individuels que sont les comparateurs, les CNA, les compteurs et les timers. Il faut également bien appréhender la manière dont sont articulés les algorithmes de détection pour MCU. Ces algorithmes sont généralement constitués d'états et de transitions entre ces états, ces transitions dépendant de conditions

logiques: une valeur de compteur qui est inférieure à une valeur donnée à l'échéance d'un timer par exemple, ou l'entrée du comparateur analogique qui passe sous un certain seuil avant que le timer n'arrive à son terme.

Ces conditions peuvent être gérées et contrôlées par une machine d'état qui, si elle est programmable, peut être configurée pour gérer un large éventail de fonctions de détection, et ce sans qu'il soit besoin de réveiller le cœur de processeur. C'est l'option qu'Energy Micro a choisi pour l'implantation du bloc périphérique LESENSE.

[Fig 4]

Mais de nombreuses machines d'état programmables ont l'inconvénient de ne pouvoir actionner et déclencher qu'un nombre figé de sorties à chaque transition entre états. Au sein du bloc LESENSE, les états et les conditions de transition sont programmés via des "descripteurs" qui peuvent être chaînés pour réaliser des opérations relativement complexes là où le besoin s'en fait sentir (mais ils peuvent aussi se limiter à de simples sorties si cela s'avère suffisant). Il est ainsi possible de bâtir des fonctions logiques relativement complexes sans avoir à solliciter l'intervention du cœur de processeur.

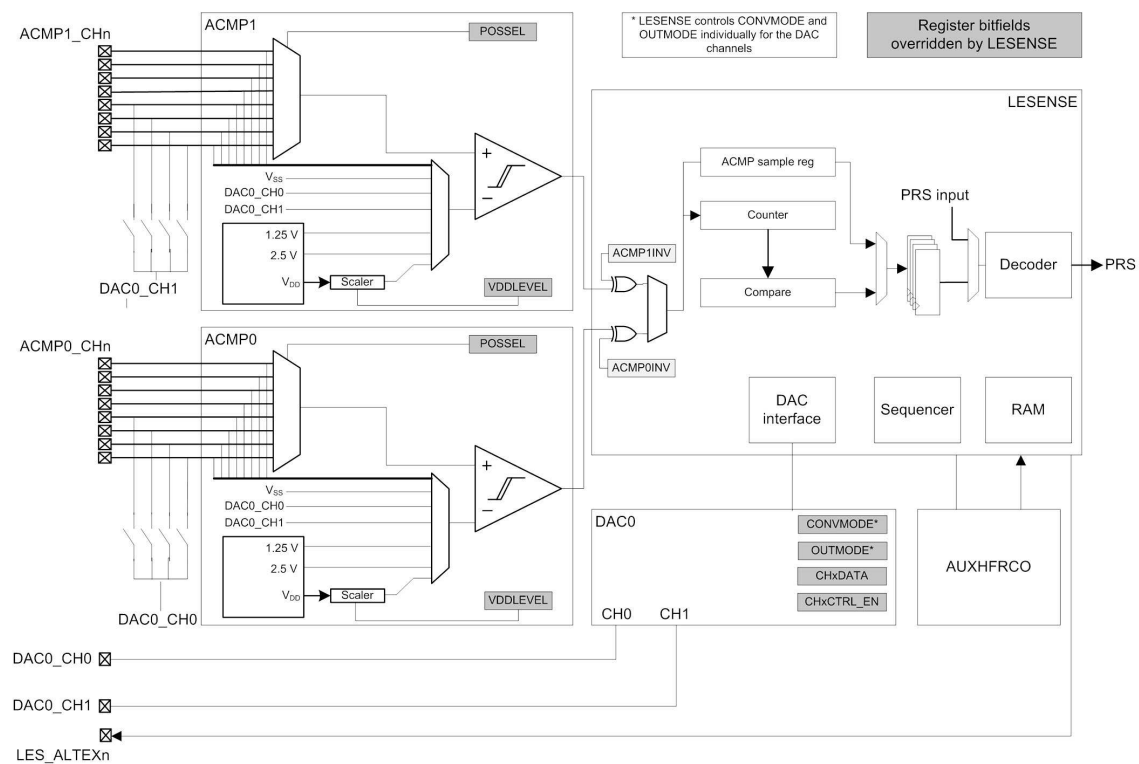


Figure 4: Les nouveaux blocs fonctionnels autonomes du MCU sont particulièrement importants pour un fonctionnement éco-efficace. L'interface pour capteurs basse consommation LESENSE peut être configurée pour superviser virtuellement n'importe quel capteur analogique sans intervention du CPU.

Grâce à l'ajout d'un décodeur configurable dans le périphérique, il est alors possible d'étendre les capacités de la machine d'état pour qu'elle puisse traiter les signaux en provenance de multiples capteurs. A titre d'exemple, le décodage de signaux en quadrature est très souvent employé dans le comptage de flux d'énergie ou dans d'autres types d'applications faisant appel à des capteurs de vitesse de rotation à base de techniques inductives. Avec le décodage en quadrature, les capteurs passent de manière

cyclique dans une succession d'états qui sont liés à la position desdits capteurs. Le décodeur offre ainsi la possibilité de choisir le jeu de capteurs le plus approprié afin de procéder à une analyse sans que le cœur de processeur ne soit sollicité. Dès lors, le MCU doit passer en mode actif uniquement lorsqu'une lecture positive a été détectée sur l'une des positions en quadrature et qu'un événement a été dûment enregistré.

En étudiant de manière globale les combinaisons matériel/logiciel potentiellement implantables dans un système architecturé autour d'un MCU – et en répartissant les fonctions entre l'un et l'autre - il est donc parfaitement envisageable de requérir la pleine flexibilité du logiciel uniquement lorsque c'est nécessaire. En requérant le logiciel pour traiter les données émises par le capteur uniquement lorsqu'il y a effectivement des données à traiter, la consommation d'énergie est maintenue au strict minimum.

www.energymicro.com