

# **ASTORE : Autonomie des Systèmes embarqués par la Technologie de l'Ordonnancement temps réel et la Récupération de l'Énergie renouvelable**

Maryline Chetto<sup>1</sup>, Christophe Plot<sup>2</sup>

<sup>1</sup> IRCCyN / Université de Nantes, 2 Avenue du Professeur Jean Rouxel,  
44475 Carquefou cedex - France

<sup>2</sup> CRTTI – IUT de Nantes, 2 Avenue du Professeur Jean Rouxel,  
44475 Carquefou cedex - France

[Maryline.chetto@univ-nantes.fr](mailto:Maryline.chetto@univ-nantes.fr) [Christophe.plot@univ-nantes.fr](mailto:Christophe.plot@univ-nantes.fr)

**Résumé** – Dans la conception d'un système temps réel embarqué il existe plusieurs attributs dont le développeur doit tenir compte. Premièrement, le produit final doit être correct d'un point de vue fonctionnel. Deuxièmement, sa performance exprimée en termes de temps de réponse doit être satisfaisante et conforme aux spécifications. Et troisièmement, lorsqu'il utilise une batterie comme source d'alimentation, son fonctionnement doit s'adapter à la quantité d'énergie disponible. Comme l'altération d'un de ces attributs va altérer les deux autres, la conception d'un système embarqué passera par la considération conjointe des aspects temporels et énergétiques. Notre objectif est de prouver sur un démonstrateur l'efficacité d'une stratégie qui combine ordonnancement temps réel et gestion énergétique visant à assurer un fonctionnement perpétuel qualifié d'énergétiquement neutre.

**Mots clés** – Système embarqué, récupération d'énergie, ordonnancement, énergie solaire.

## **1. Introduction**

La gestion de l'énergie est devenue la problématique centrale des systèmes embarqués qui s'alimentent sur une source d'énergie autonome et ne peuvent se recharger sur le secteur. Ce type de système se retrouve aussi bien dans le domaine des produits grand public telle que la téléphonie mobile que dans les systèmes qualifiés de temps réel dur pour lesquels un manque d'énergie conduit à la panne fatale et engendre perte financière voire perte de vies humaines dans les applications les plus critiques.

Traditionnellement, la plupart des systèmes embarqués sont conçus pour fonctionner en « basse consommation ». Cependant, il existe une différence essentielle entre la technologie basse consommation et celle que l'on peut qualifier de « conduite par la consommation ». Selon cette dernière, l'objectif est de faire le meilleur usage de la puissance disponible ce qui revient pour l'ordonnanceur à exécuter les tâches temps réel dans des intervalles de temps tel que toutes les contraintes temporelles et énergétiques soient satisfaites. Aujourd'hui, nous disposons de batteries de très haute densité et de super-condensateurs. Cependant, la quantité maximum d'énergie qui peut être emmagasinée limite très sévèrement la durée de vie des applications. D'autre part, dans la grande majorité des applications sans fil tels que les réseaux de capteurs, recharger ou remplacer une batterie se révèle délicat voire impossible. Par conséquent, l'avenir est à l'utilisation des sources d'énergie alternatives présentes dans l'environnement.

La récupération de l'énergie environnementale (Environmental energy harvesting) apparaît comme une approche prometteuse car l'environnement est capable de fournir cette énergie d'une part en quantité suffisante et d'autre part sur une durée sinon infinie au moins sur une décennie.

Différentes technologies pour extraire l'énergie de l'environnement ont fait leurs preuves. Les plus connues sont certainement celles du solaire et du piézo-électrique.

La question clé attachée à la gestion de puissance d'un système à récupération d'énergie n'est pas de minimiser la consommation de l'énergie pour maximiser la durée de vie du système comme dans les matériels portables traditionnels [1]. Le problème revient à garantir un mode de fonctionnement qualifié d'«énergétiquement neutre» dans lequel à tout moment le système ne consomme jamais plus d'énergie que celle récupérée.

Les systèmes embarqués sont généralement temps réel c'est à dire assujettis à des calculs en temps contraints, en réponse à des stimuli environnementaux. Qu'il soit temps réel à contraintes strictes ou à contraintes souples, un système embarqué temps réel fonctionnera d'une façon énergétiquement neutre si ses contraintes d'exécution peuvent être respectées malgré les limitations énergétiques. Une approche évidente pourrait consister à construire un récupérateur d'énergie dont la puissance minimum instantanée est à tout instant au moins égale à la puissance maximale requise par toute tâche en exécution. Malheureusement une telle approche présenterait des inconvénients majeurs en termes de coût, de volume et de poids.

La plupart des systèmes embarqués construits jusqu'à maintenant ne gèrent pas efficacement l'énergie extraite et par conséquent utilisent un récupérateur plus volumineux ( par exemple panneau solaire de plus grande dimension) et plus cher que celui nécessité par l'utilisation d'une stratégie optimale de la gestion de l'énergie récupérée. Et c'est en particulier cet aspect qui jusqu'à présent a limité le champ d'utilisation des batteries dynamiquement rechargeables à partir de l'énergie ambiante.

Les premiers travaux de recherche sur ce sujet sont apparus il y a environ cinq ans et traitent principalement de la question suivante : comment adapter le fonctionnement d'un système temps réel en fonction de la quantité d'énergie récupérée ?

Dans cet article, nous considérons une architecture matérielle de type monoprocesseur en charge d'exécuter des tâches soumises à des échéances strictes et qui requièrent des quantités d'énergie différentes pour leur exécution. Notre application consiste donc en une unité de traitement, un récupérateur d'énergie ( de type panneau solaire) et une unité de stockage d'énergie (batterie ou super-condensateur).

Plus précisément la question à laquelle nous souhaitons apporter une réponse est : comment ordonnancer les tâches pour garantir perpétuellement leurs contraintes temporelles en exploitant de façon adéquate à la fois la ressource processeur et la ressource énergie ambiante ?

Une stratégie d'ordonnancement EDeg (Earliest Deadline with energy guarantee) a été proposée fin 2009 [2]. Cette dernière a fait l'objet d'une étude en simulation et doit être validée grâce à un démonstrateur [3].

Le reste de ce papier est organisé de la façon suivante: Le §2 donne un état de l'art. Dans le §3, nous introduisons la problématique de l'ordonnancement associée aux systèmes embarqués récupérant l'énergie et décrivons la stratégie que nous proposons. Le §4 donne les objectifs visés par le démonstrateur et décrit celui-ci. Nous concluons par le §5.

## **2. Etat de l'art**

### **2.1 Energie ambiante et ordonnancement temps réel**

La très grande majorité des travaux de recherche portant sur l'ordonnancement temps réel contraint par l'énergie visent à minimiser l'énergie consommée en vue de maximiser la longueur de l'intervalle de temps séparant deux recharges de batterie. Ils ignorent donc la problématique liée à un fonctionnement énergétiquement neutre qui se caractérise par la recharge continue de la batterie. Ces travaux portent sur des techniques de gestion dynamique de puissance (Dynamic Power Management : DPM) [4] [5] et des techniques utilisant la variation de tension et de fréquence (Dynamic Voltage and Frequency Selection :DVFS) [6] [7]. Bien que les approches DPM et DVFS soient efficaces et souhaitables, elles ne suffisent pas à satisfaire les contraintes spécifiques liées à la rechargeabilité dynamique d'une batterie ou d'un super-condensateur.

Le premier résultat significatif sur cette thématique a été présenté en 2006 par Moser et al. du Swiss Federal Institute of Technology de Zurich. Ils proposent une stratégie d'ordonnancement monoprocasseur préemptive appelée LSA (Lazy Scheduling Algorithm) et prouvent son optimalité [8] [9]. Ici, la performance d'un algorithme d'ordonnancement se mesure en pourcentage d'applications temps réel fiablement ordonnancées étant donné une unité de traitement, une unité de stockage d'énergie de capacité donnée et un profil de la source d'énergie. Dans ce travail, le système de traitement est supposé consommer la même puissance instantanée quelque soit la tâche exécutée. Cela signifie donc que durée d'exécution et énergie consommée par toute tâche sont reliés par un unique facteur, la puissance maximale de consommation du processeur. C'est cette hypothèse restrictive que nous avons supprimée dans nos travaux. En effet considérer que la puissance consommée par une tâche le long de son exécution reste constante n'est réaliste que si cette tâche ne nécessite qu'un unique circuit de puissance donnée. Or la réalité se révèle plus complexe car une tâche nécessite des accès mémoire, des communications radio,... et aura une consommation de puissance fluctuante au cours de son exécution en fonction de la circuiterie utilisée.

## 2.2 Energie ambiante et noyaux temps réel

Plusieurs noyaux temps réel, particulièrement adaptés aux systèmes alimentés par des batteries ont été conçus ces dernières années: MANTIS de l'université du Colorado [10], CONTIKI de Swedish Institute of Computer Science [11], NANO-RK de l'université de Carnegie Mellon [12] et TinyOS de l'université de Californie à Berkeley [13]. Ces noyaux à code source ouvert ont été conçus par la communauté universitaire pour des micro-contrôleurs disposant d'un espace mémoire restreint. Malheureusement, aucun d'eux ne fournissent des fonctionnalités de gestion de puissance qui tiennent compte conjointement des contraintes temporelles et de la récupération en ligne de l'énergie.

## 3. L'ordonnanceur

### 3.1 Introduction à la problématique

Dans notre projet, nous visons un système à récupération d'énergie qui se compose de quatre modules: une source d'énergie électrique (panneau solaire), une unité de stockage d'énergie (un super-condensateur), un module de traitement mono-coeur (un micro-contrôleur) et une liste de tâches temps réel. La principale caractéristique de ce système est que l'énergie disponible à tout moment se trouve limitée par la taille de l'unité de stockage.

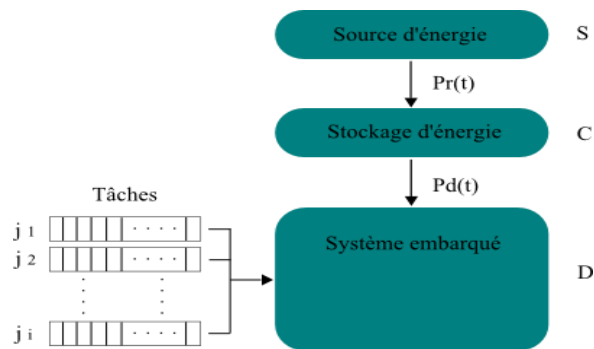


Figure 1 : Schéma type d'un système embarqué à récupération d'énergie

D'une façon synthétique, nous devons répondre aux questions suivantes :

- A quels instants, le système peut-il utiliser l'énergie et à quels instants doit-il rester inactif et ainsi permettre la recharge d'énergie?
- Si l'unité de stockage devient vide, l'application devra être interrompue. Comment éviter une telle situation de manque d'énergie ?

- Si l'unité de stockage devient pleine, l'énergie superflue sera gaspillée. Comment éviter une telle situation de surplus d'énergie ?
- Dans les systèmes temps réel où une violation d'échéance peut conduire à une dégradation de performance sévère, dans quel ordre devrions-nous traiter les tâches de façon à respecter les échéances?
- Comment dimensionner au mieux l'unité de stockage tenant compte des situations les plus pénalisantes de manque d'énergie?
- Comment implémenter une solution efficace en termes de complexité calculatoire?

Dans ce projet, les challenges sont donc de plusieurs ordres :

1. *logiciel* : en fournissant un environnement de programmation simple et efficace (micro-kernel) qui inclut la stratégie de gestion énergétique et d'ordonnancement dédiée à tout système temps réel à récupération d'énergie.
2. *méthodologique*: en fournissant une méthode et une boîte à outils qui intègre la stratégie précédente. L'aspect dimensionnement conduira à des économies en termes de coût (taille et nombre de super-condensateurs, taille du panneau solaire). Cette approche de conception revêt une grande importance pour la miniaturisation des appareils électroniques sans fil de nouvelle génération.
3. *matériel*: en validant par l'intermédiaire de l'expérimentation sur une application réelle, la technologie ainsi développée et la méthodologie de conception.

### 3.2 L'algorithme d'ordonnancement

La stratégie d'ordonnancement EDeg proposée repose sur les principes suivants :

- Les tâches sont exécutées selon la règle bien connue Earliest Deadline First (EDF). Cependant, avant d'autoriser une tâche à s'exécuter, nous devons nous assurer que cette exécution ne remet pas en question l'exécution fiable des tâches à venir compte tenu d'une part de leurs besoins en énergie et d'autre part de la quantité d'énergie disponible à l'instant courant additionnée de l'énergie à venir. Lorsque cette condition n'est pas satisfaite, le processeur doit rester inactif de façon à permettre la recharge d'énergie. Se pose alors le problème de déterminer la longueur maximale de l'intervalle de temps pour cette recharge.
- Nous introduisons deux notions: la laxité temporelle et la laxité énergétique qui sont des grandeurs dynamiques. La laxité temporelle représente la longueur de l'intervalle de temps pendant lequel le processeur peut rester inactif tout en garantissant le respect des contraintes temporelles. La laxité énergétique représente la quantité maximale d'énergie qui peut être consommée sans remettre en question l'exécution des tâches à venir. C'est l'évaluation en-ligne de ces grandeurs qui engendrera un overhead qu'il convient de parfaitement mesurer.

Ci-dessous, une description algorithmique de EDeg vous est présentée.

```

Algorithm EDeg
  while (1) do
    while PENDING=true do
      while E(t) > Emin and Slack_energy(t) > 0 do
        execute()
      end while
      while (E(t) < Emax and Slack_time(t) > 0) do
        wait()
      end while
    end while
    while PENDING=false do
      wait()
    end while
  end while

```

où l'on désigne par

PENDING : une variable booléenne, vraie s'il existe au moins une tâche en attente d'exécution,

$E(t)$  : la quantité d'énergie disponible à l'instant courant  $t$  dans l'unité de stockage,

$E_{\max}$  : la quantité maximale d'énergie stockable,

$E_{\min}$  : la quantité minimale d'énergie stockable,

Slack\_energy( $t$ ): la laxité énergétique du système à l'instant  $t$ ,

Slack\_time( $t$ ): la laxité temporelle du système à l'instant  $t$

Nous avons dans un premier temps mené une étude en simulation visant à évaluer cette stratégie, indépendamment du matériel. Un simulateur à temps discret a été développé en langage C. A des fins de comparaison, nous avons implémenté plusieurs heuristiques gloutonnes basées sur EDF pour les comparer à EDeg notamment en termes de pourcentage d'échéances satisfaites [3].

## 4. Le démonstrateur

### 4.1 Objectifs scientifiques

Le principal objectif du démonstrateur est de déterminer l'infrastructure nécessaire à l'implémentation de la stratégie.

Le second objectif est de prouver l'efficacité et la faisabilité de cette stratégie selon les deux points de vue suivants :

- *Performance temps réel*: Une première étape concerne l'évaluation en termes d'échéances satisfaites.

- *Efficacité énergétique*: Les simulations ont prouvé que notre méthode nécessite une source d'énergie moins puissante (donc des panneaux solaires de plus petite dimension) par comparaison avec un algorithme d'ordonnancement glouton. Cela signifie aussi que de plus petites unités de stockage d'énergie (super-condensateurs de plus faible capacité) suffiront pour aboutir à un niveau de performance similaire à celui d'un ordonnanceur glouton. En résumé, nous voulons évaluer précisément le gain obtenu en termes de poids, coût et volume malgré l'ajout d'une circuiterie rendue indispensable par la mesure en ligne de la quantité d'énergie disponible.

### 4.2 Verrous technologiques

La réussite face à ces challenges suppose de lever les verrous technologiques suivants:

- La stratégie EDeg requière des informations dynamiques précises sur la ressource énergétique caractérisée par une puissance fortement fluctuante. Se pose alors le problème de la fiabilité des mesures d'énergie disponible à partir de grandeurs de type intensité et tension. D'autre part, avec quelle résolution ces mesures doivent-elles être prises ?

- Dans nos précédents travaux, nous avons supposé une unité de stockage d'énergie idéale c'est à dire pour laquelle l'énergie stockée peut être utilisée sans limitation de temps car sans fuite. Dans quelle mesure une telle hypothèse reste-elle réaliste ? Et comment adapter la stratégie EDeg pour tenir compte des fuites d'énergie?

- De même, le temps et l'énergie consommés par l'exécution de l'ordonnanceur pourront introduire des surcoûts prohibitifs. Comment les prendre en compte ?

### 4.3 Construction d'une plateforme

La finalité du projet ASTORE est de disposer d'une plateforme de démonstration qui soit représentative d'une majorité des systèmes embarqués alimentés grâce à l'énergie environnementale, gardant à l'esprit les considérations suivantes :

- Démontrer l'intérêt de la récupération de l'énergie environnementale dans le domaine de l'embarqué

- Démontrer l'impact de l'ordonnanceur sur le respect des contraintes temporelles, sur la taille du panneau solaire et celle du super-condensateur.

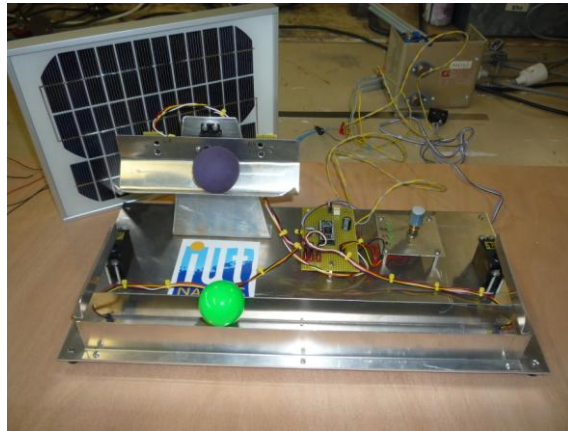


Figure 3 : le démonstrateur

La figure 3 présente le premier démonstrateur développé dans le cadre du projet ASTORE, il regroupe trois processus indépendants contrôlés par le même processeur. Le premier vise à maintenir en équilibre entre deux positions fixes une balle de ping-pong. Le second consiste à gérer des mouvements de balle en aller et retour, la balle étant renvoyée à chaque extrémité par deux actionneurs électriques. Le troisième consiste simplement à faire clignoter une diode électroluminescente, cette tâche pouvant être bloquante ou non. L'alimentation électrique des différents processus et du processeur est assurée par un panneau photovoltaïque associé à une batterie de super-condensateurs.

## 5. Conclusion

Il existe actuellement deux catégories de noyaux temps réel: dans la première, on retrouve tous les mécanismes nécessaires à l'ordonnancement sous contraintes temporelles, la synchronisation, la communication,... mais la gestion de l'énergie y est absente. Dans la seconde catégorie, nous retrouvons les systèmes d'exploitation qui proposent des techniques de type DPM ou DVFS pour minimiser l'énergie consommée et maximiser la durée de vie d'une application.

La nouveauté du projet ASTORE (Autonomie des Systèmes embarqués par la Technologie de l'Ordonnancement temps réel et la Récupération de l'Énergie renouvelable) tient dans la proposition de fonctionnalités nouvelles pour prendre en compte conjointement les aspects temporels des applications temps réel et les aspects énergétiques des systèmes autonomes alimentés par une source d'énergie renouvelable. Nous avons montré que ces fonctionnalités visent à adapter dynamiquement l'activité de l'unité de traitement au profil de la source énergétique de façon à garantir un fonctionnement perpétuel dit énergétiquement neutre. Un démonstrateur est actuellement en construction. Ce dernier utilise un panneau solaire et un super-condensateur pour la récupération et le stockage de l'énergie. Un microcontrôleur constitue l'unité de traitement. Le travail à venir (celui d'un Master recherche) consistera à concevoir un noyau temps réel intégrant notre stratégie d'ordonnancement et à l'intégrer au démonstrateur.

## 6. Bibliographie

- [1] M.T. Scmitz, B.M. Al-Hashimi and P. Eles “*System-Level Techniques for Energy Efficient Embedded Systems*”, Kluwer Academic Publishers, 194 pages, 2004.
- [2] M. Chetto and H. El Ghor, “*Real-time Scheduling of periodic tasks in a monoprocessor system with a rechargeable battery*” 17<sup>th</sup> IEEE International Symposium on Real Time Systems (RTSS 2009), wip session, Washington, 2-4 december 2009.
- [3] H. El Ghor, M. Chetto and R. Hage Chehade, “*A Real-Time Scheduling Framework for Embedded Systems with environmental energy harvesting*”, IRCCyN Technical report, 2010\_07, may 2010.
- [4] Lu, Y.-H., Benini, L., and De Micheli, G, “*Low-power task scheduling for multiple device*”, Proc. Int. Workshop HW/SW Co-design, 2006.
- [5] S. Liu, Q. Qiu, Q. Wu, “*Task merging for dynamic power management of cyclic applications in real-time multi-processor systems*”, in Proc of ICCD, 2006.
- [6] F. Yao, A. Demers, et al, “*A scheduling model for reduced CPU energy*” in IEEE symposium on Foundations of Comp. Science, 1995.
- [7] J. Luo and N. K. Jha, “*Static and dynamic variable voltage scheduling algorithms for real-time heterogeneous distributed embedded systems,*” In Proc. Of Int. Conf. on VLSI Design, pp.719-726, 2002.
- [8] C. Moser, D. Brunelli, L. Thiele, and L. Benini, “*Real-time scheduling with regenerative energy,*” in Proc. of the 18th Euromicro Conference on Real-time Systems (ECRTS06), pp 261-270, Dresden, Germany, October, 2006.
- [9] C. Moser, D. Brunelli, L. Thiele, L. Benini, “*Real-time scheduling for energy harvesting sensor nodes*”, Real-Time Systems, Volume 37 , Issue 3, Pages: 233 - 260, December 2007.
- [10] MANTIS HomePage, <http://mantis.cs.colorado.edu/> , University of Colorado at Boulder.
- [11] CONTIKI HomePage, <http://www.sics.se/contiki>
- [12] Nano-RK project HomePage, <http://www.nanork.org/>
- [13] TinyOS HomePage, <http://www.tinyos.net/>, University of California at Berkeley