Étude, réalisation et applications d'une maquette polyvalente de développement ARM7 enseignement-recherche

Pascal Varoqui, Gilbert Pradel
pascal.varoqui@ens-cachan.fr, gilbert.pradel@ens-cachan.fr
École Normale Supérieure de Cachan, département EEA
61 avenue du président Wilson, 94235 Cachan Cedex

RESUME : Cet article présente une expérience de transfert bidirectionnel recherche-enseignement en informatique industrielle. L'expérience accumulée dans le cadre du projet ANR RobAutiSTIC [1] et la confrontation des besoins spécifiques en matière de maquettes d'expérimentation en recherche et dans l'apprentissage de l'informatique industrielle au niveau licence/maîtrise ont guidé nos choix dans la réalisation d'une carte polyvalente autour d'un processeur à cœur ARM7 [2]. Cette carte a été conçue pour être compatible avec la base robotique du projet RobAutiSTIC et servir de support à la mise en place d'un enseignement en architecture des systèmes embarqués et enfouis et en systèmes multi-tâche temps réel. Cet enseignement s'insère dans la préparation à l'agrégation de génie électrique de l'ENS de Cachan et dans la formation M2P en Génie des Systèmes Informatiques de l'université Paris XI, centre d'Orsay. Le transfert initial recherche vers enseignement s'est accompagné d'un retour de l'enseignement vers la recherche : certains objectifs pédagogiques ont conduit à mettre en œuvre de nouvelles fonctionnalités et à des simplifications qui ont bénéficié au robot du projet RobAutiSTIC.

Mots clés : informatique industrielle, robotique, dispositif pédagogique, transfert enseignement-recherche, retour d'expérience.

1 INTRODUCTION

La double casquette d'enseignant-chercheur conduit facilement à utiliser des matériels très différents dans le cadre de l'informatique industrielle lorsque les choix techniques sont faits de façon indépendante entre les activités pédagogiques et de recherche. En effet, les contraintes sont différentes : d'une part la nécessité de mettre en œuvre des maquettes pédagogiques et solides, pour lesquelles la modernité ou la performance sont des priorités de faible niveau ; d'autre part des projets de recherche qui nécessitent des technologies actuelles et performantes pour satisfaire des contraintes imposées par l'objet de la recherche.

La conséquence est une séparation assez nette à la fois dans le domaine matériel et logiciel : des maquettes uniquement orientées enseignement dont la technologie peut-être ancienne mais robuste et par ailleurs des cartes dédiées à des applications de recherche à la technologie actuelle mais développées sans souci de polyvalence ni souvent de pérennité. Ces écarts matériels génèrent des incompatibilités entre les codes des programmes étudiés dans la partie pédagogique et ceux mis au point dans la partie recherche. Un enseignant peut utiliser en travaux pratiques des maquettes basées sur des processeurs 8 bits d'une conception trentenaire tout en mettant en œuvre dans ses activités de recherche des processeurs ARM de dernière génération pour satisfaire des contraintes temps réel hors de portée d'un antique 8051. De cette séparation, de la différence des systèmes en usage dans ces deux domaines, découle une synergie difficile entre les activités d'enseignement et de recherche.

Ce constat est-il réellement une fatalité inévitable ?

Cet article va d'abord présenter une comparaison des contraintes à satisfaire dans la conception d'une carte d'expérimentation micro-informatique dans le cadre de

l'enseignement et celles imposées par son utilisation dans nos domaines de recherche. Ensuite nous présenterons les choix effectués pour satisfaire ces contraintes dans la réalisation d'une carte mère ARM7 polyvalente, quelques exemples d'utilisations dans les domaines de l'enseignement et de la recherche.

Nous concluons par un bilan après deux années d'utilisation de ce matériel.

2 VERS UNE CONVERGENCE DU CAHIER DES CHARGES

2.1 Compatibilité des contraintes d'utilisation

Les contraintes peuvent sembler inconciliables et il y a effectivement une logique qui conduit à l'état de fait évoqué plus haut : la pédagogie a moins besoin de s'encombrer d'actualité, et les processeurs les plus modernes sont aussi les plus difficiles à aborder de par leur complexité, fruits de plusieurs décades d'études accumulées. D'un autre côté il serait illusoire de croire que l'ancienneté des composants utilisés soit un gage de qualité pédagogique, et l'actualité des composants et thèmes abordés constitue une motivation pour les élèves. Sachant que la programmation se fait dans la très grande majorité des cas en langage C, la complexité du langage machine et de la structure d'un processeur actuel n'est pas réellement un frein à son utilisation en TP. Bien au contraire ses riches possibilités d'entrées/sorties et périphériques ouvrent la porte à de nouvelles manipulations (I2C, Timer 32 bits, SPI, PWM, SD Card, ...). Du côté de l'utilisation "recherche", la puissance de traitement et la polyvalence sont deux paramètres capitaux pour accueillir les projets en cours ou à venir qui comportent toujours un noyau temps réel et des entrées/sorties nombreuses et complexes.

La mise au point du code est une activité incontournable et travailler sur une plateforme éprouvée et déjà

équipée d'une interface utilisateur de type clavier/écran embarquée facilite la mise au point.

2.2 Besoins pédagogiques

Une mise à plat des besoins à satisfaire pour l'enseignement de l'informatique industrielle à des élèves de licence et maîtrise a permis de dégager les caractéristiques nécessaires suivantes :

- un processeur d'actualité (abandon des 8/16 bits au profit d'un 32 bits),
- quelques entrées/sorties faciles à mettre en oeuvre pour les premiers TP : leds, interrupteurs, buzzer,
- des interfaces utilisateur plus complexes : afficheur LCD, clavier matricé,
- des périphériques plus inhabituels permettant à la fois d'aborder des problèmes particuliers et d'apporter un côté ludique : télémètres à ultrason, servo-moteur de télécommande,
- un accès à des bus externes et des E/S classiques : I2C, SPI, UART,
- des possibilités de gestion de l'énergie (mise en veille, contrôle de la fréquence de travail, désactivation de périphériques inutilisés),
- un système d'interruptions vectorisées performant,
- un interfaçage avec des maquettes robotiques existantes,
- utilisation d'un exécutif temps réel multitâche,
- environnement de développement facile à prendre en main et efficace,
- facilité de mise en œuvre et robustesse contre les erreurs de manipulation.

2.3 Besoins recherche

Les besoins sont spécifiques à l'objet de la recherche, et la carte idéalement polyvalente prête à faire face à toutes les contraintes n'existe pas. Nous nous sommes centré sur nos domaines d'activité pour définir les points suivants :

- puissance de calcul et faible consommation pour une utilisation robotique embarquée,
- souplesse de communication avec d'autres systèmes et interfaces : possibilité d'utiliser au maximum les E/S du système grâce à une architecture de base qui mobilise le moins d'E/S possible,
- utilisation d'un exécutif multitâches temps réel.

2.4 Autres éléments pris en compte

Quelques autres considérations ont déterminé notre choix du système de développement :

- mise au point in-situ par liaison JTAG
- système d'exploitation Linux et Windows
- coût raisonnable

Par ailleurs, l'expérience accumulée dans les activités antérieures de recherche dans le domaine de la robotique a aussi guidé le choix de l'architecture du système

et du cœur du CPU : un "System On Chip" (SOC) à cœur ARM.

3 LES CHOIX EFFECTUES

La confrontation des besoins en enseignement et en recherche a permis de définir les éléments intégrés à la carte et ses possibilités de connexion à des périphériques externes.

3.1 Compatibilité avec le projet RobAutiSTIC

Fig. 1 : Robot GIPY du projet RobAutiSTIC

Un travail de recherche dans le cadre du projet ANR RobAutiSTIC a conduit à la conception et à la réalisation d'un robot jouet équipé de diverses modalités (mouvements, sonores, lumineuses) destiné à servir de facilitateur des interactions entre un enfant autiste et son environnement (figure 1). Notre volonté de convergence enseignement-recherche nous a dicté les limites dimensionnelles de la carte étudiée, ses points de fixation et le connecteur d'interface avec la base robotique du projet RobAutiSTIC (figure 2).

Fig. 2 : Embase du robot GIPY

3.2 Aspect pédagogique

L'association de la carte développée avec cette embase robotique (figure 3) permet de proposer facilement des TP axés sur les problèmes des systèmes embarqués et la gestion d'un petit robot. Elle offre en outre une plateforme d'étude qui profite au développement de RobAutiSTIC.

En examinant l'offre des environnements de développement en 2006, nous avons fait le choix de la chaîne de développement croisé CrossStudio de la société Rowley. Celle-ci proposait un lot comprenant une carte CrossFire 2138 basée sur le LPC2138 [2] et l'environnement de développement associé à un tarif très compétitif pour ses caractéristiques [8]. CrossStudio se compose des compilateurs gcc et g++, d'un éditeur de lien, d'un moniteur de mise au point, le tout intégré dans une interface graphique ergonomique. La liaison avec la carte ARM7 et son alimentation se font par un simple cordon USB, l'interface USB-JTAG étant intégrée à la carte CrossFire.

Fig. 3 : Carte ARM7 montée sur le châssis du robot

4 APPLICATIONS PEDAGOGIQUES

Dans cette section, nous présentons 3 objets d'étude particuliers abordés en enseignement. Ils ont été proposés à des élèves de M2P "génie des systèmes d'informatique industrielle" de l'université Paris XI, centre d'Orsay.

4.1 Mise en évidence des limitations du système d'interruptions du processeur

4.1.1 Présentation des objectifs

La plupart des enseignements sur les interruptions abordent les principes (événement à arrivée aléatoire, vectorisation, sauvegarde et restitution du contexte d'exécution du programme) et grâce à des exercices les mettent en évidence. Il est plus rare de proposer aux élèves un dispositif qui pousse le système d'interruption à ses limites : quelle est la fréquence maximale d'occurrence des interruptions admissible par le processeur dans un contexte donné, compatible avec la bonne exécution des tâches ? Comment visualiser cette mise en

défaut et comparer les performances des différents types d'interruptions (IRQ, FIQ) ?

Pour répondre à ces interrogations, ce TP permet de soumettre le processeur à des interruptions externes issues d'un GBF, d'observer son temps de réponse et la bonne exécution des tâches en s'aidant d'observations oscilloscopiques. Il constitue une introduction à la série de TP du module systèmes temps réel embarqués.

4.1.2 Déroulement des manipulations

Le processeur exécute en tâche de fond un programme assurant la mise à l'état haut répétitive d'une sortie A. La procédure d'interruption invoquée sur l'arrivée d'un front sur une entrée B du système provoque la mise à 0 de A ainsi que le basculement d'une sortie C (réalisation d'un diviseur de fréquence par deux, figure 4). L'observation simultanée de ces signaux permet de quantifier le temps de réponse de l'interruption et les possibilités d'exécution de la tâche de fond et de la tâche d'interruption.

Fig. 4 : schéma de principe

Le code utilisé est très simple pour faciliter la compréhension des phénomènes. La figure 5 en présente les parties principales rédigées en C dans un soucis de clarté et de fidélité aux conditions habituelles de mise en œuvre de ce type de système. Une écriture en assembleur aurait permis de meilleures performances mais sortait du cadre du TP.

```
// réponse à EINT1 (entrée B)
void div_freq(void) {
    IOOCLR = sortie_A; //A à 0
    IOOPIN ^= sortie_C; //basculement sortie diviseur de fréq.
    EXTINT=EXTINT_EINT1; //acquiescement
}
...
// tâche de fond
int main (void) {
    init(); //init matériel et logiciel
    while (1) {
        IOOSET = sortie_A; //A à 1
    }
}
```

Fig. 5 : Extraits du code source utilisé

Une première mise en œuvre à fréquence d'interruption faible permet d'appréhender le fonctionnement du système et d'en relever les temps caractéristiques (figure 6). Dans tous les chronogrammes qui suivent, on représente dans l'ordre les signaux A, C, B.

Fig. 6 : Interruption isolée

Une augmentation de la fréquence du signal B permet de bien visualiser le fonctionnement du diviseur de fréquence sur la sortie C et l'augmentation du temps passé dans la routine d'interruption par rapport à la tâche de fond (signal A), comme l'illustre la figure 7.

Fig. 7 : Interruption répétée

Au cours de ce TP il est demandé aux élèves de mesurer les temps correspondants aux différentes étapes qui mènent à l'exécution de la procédure d'interruption depuis l'apparition de la requête d'interruption, et de justifier leurs variations.

Une augmentation supplémentaire de fréquence fait apparaître plus clairement les limites du système. D'abord le temps d'exécution de la tâche de fond se réduit au point de ne plus permettre avec certitude l'exécution complète d'un tour de sa boucle infinie entre deux interruptions (figure 8).

Fig. 8 : Fréquence élevée

En augmentant encore la fréquence, le temps machine alloué à la tâche de fond diminue et peut même s'annuler (figure 9).

Fig. 9 : Limite de fonctionnement atteinte

Un accroissement supplémentaire de fréquence dégrade le fonctionnement du système. La division de fréquence ne peut plus être assurée correctement car une nouvelle interruption peut survenir avant complétion du traitement de la précédente : certains fronts montants de B échappent au diviseur de fréquence. On constate incidemment que cela redonne un peu de temps machine à la tâche de fond avant le front actif suivant (figure 10).

Les élèves sont invités à relever les valeurs limites de fonctionnement et les valeurs mini et maxi du temps de réponse à l'interruption en justifiant cette variation constatée.

Fig. 10 : Limite de fonctionnement correct dépassée

Ces mesures, malgré leur imprécision, permettent de fixer l'ordre de grandeur de la réponse du processeur et de ses possibilités de traitement :

Le temps de réponse du système à une interruption extérieure (entre le signal d'entrée et la première action conséquente sur les sorties) est légèrement inférieur à la microseconde.

Il est irréaliste de traiter par interruption un signal d'entrée de fréquence supérieure à 500 kHz car même avec un programme d'interruption très court la tâche de fond ne peut quasiment plus s'exécuter.

4.2 Mise en évidence du taux de charge d'un système multi-tâche

Après avoir examiné des contraintes liées aux performances du système, les étudiants sont sensibilisés au problème du temps machine alloué à chaque tâche : il ne suffit pas d'écrire du code, encore faut-il qu'il ait la possibilité d'être exécuté en fonction de la charge du système et des différentes priorités.

Lors de l'utilisation d'un système multi-tâche, il est intéressant de pouvoir quantifier le temps machine utilisé par chacune des tâches et par les routines d'interruption. Une méthode simple a été mise au point pour mettre ces temps en évidence dans les TP. Le principe

est le suivant : la tâche de fond est uniquement consacrée à l'incréméntation permanente d'un compteur. Lorsque aucune autre tâche n'est exécutée, tout le temps processeur est consacré à l'exécution de cette tâche de fond, ce qui permet au compteur d'atteindre une valeur de référence N_{max} dans un intervalle de temps donné. Cette valeur est établie à l'initialisation du système, juste après avoir fixé les conditions de fonctionnement du processeur (fréquences des bus internes). Lorsque le système multitâche est lancé, la valeur N atteinte par le compteur dans le même intervalle de temps est plus faible. Le rapport $R = N/N_{max}$ fournit une valeur représentative de l'oisiveté (inverse de la charge) du processeur à un instant donné. La charge du processeur en pourcentage est donnée par :

$$\text{Charge}_{(\%) } = 100 * (1-R)$$

La visualisation de cette valeur sur l'afficheur LCD permet de suivre en permanence le taux de charge du système pour différentes conditions de fonctionnement ou de comparer l'efficacité de plusieurs implémentations des mêmes fonctionnalités.

En pratique, la référence de temps pour cette mesure est fournie par le périphérique d'horloge temps réel (RTC) intégré au processeur. Ce périphérique fournit des interruptions dont la fréquence est indépendante du paramétrage des différentes horloges utilisées dans le fonctionnement du processeur et des autres périphériques.

Cette méthode développée pour l'enseignement a été ensuite réutilisée dans les activités de recherche. Elle a permis par exemple de mesurer le gain apporté par des optimisations en assembleur dans un codec audio GSM, ou de préciser les conditions de fonctionnement de routines d'accès à une carte SD utilisée dans RobAutoSTIC.

La mesure du taux de charge fournit une information immédiatement exploitable par l'utilisateur. Cette approche globale de la charge du système vient compléter la méthode oscilloscopique de mesure de performances dynamiques pour montrer aux étudiants les contraintes temporelles liées à sa puissance de calcul auxquelles un système temps réel doit faire face.

4.3 Comparaison de solutions matérielles et logicielles : adéquation algorithme-architecture

L'embase robotisée utilisée comporte des capteurs incrémentaux sur chaque roue d'une résolution permettant un positionnement au $10^{\text{ème}}$ de millimètre. Cette finesse de positionnement est un atout pour mouvoir avec précision un stylo fixé au robot sur une feuille de papier. Par contre, dans le cas d'un déplacement rapide, la gestion par interruptions de ces incréments conduit à une surcharge du système mis en évidence par le dispositif vu en 4.2. L'expérimentation a montré qu'il était préférable de ne pas dépasser la moitié de la vitesse maximale des moteurs dans le cas de cette gestion logicielle des compteurs.

Cette limitation peut être évitée par l'exploitation des compteurs matériels dont est dotée la carte (figure 11). Ces derniers sont incrémentés ou décrémentés selon les

mouvements des capteurs de rotation. L'accès à la valeur de ces compteurs est disponible à tout moment à travers une liaison rapide SPI. Ce comptage matériel permet un contrôle du positionnement avec un taux de charge beaucoup plus faible.

Fig. 11 : Compteurs et circuits logiques

Ce TP permet de bien mettre en évidence ce qui doit être du ressort du logiciel (règles de déplacement, traitement des informations des capteurs) et du matériel (traitement simple de signaux rapides), illustrant ainsi l'adéquation algorithme-architecture. Il montre aussi qu'une solution simple et séduisante sur le papier peut s'avérer inadaptée si les capacités de traitement du système sont mal évaluées.

5 EXTENSIONS

Des connecteurs d'extensions permettent d'accéder aux signaux de la carte CPU par l'intermédiaire d'un câble en nappe ou par enfichage direct d'un module d'extension.

Certains signaux ne sont pas libres car le fonctionnement des entrées-sorties intégrées en consomme quelques-uns. Afin de libérer au maximum les ports dans les applications qui le nécessitent, un certain nombre de fonctions intégrées peut être désactivé par écriture dans un registre du bus externe. Ainsi, les compteurs des capteurs incrémentaux, les capteurs à ultrason et le buzzer peuvent libérer les entrées/sorties correspondantes. Seul le bus externe ne peut être désactivé, clavier et LCD sont ainsi toujours disponibles sans occupation supplémentaire de broches.

5.1 Extensions pédagogiques

La première carte d'extension (à gauche sur la figure 12) a été réalisée pour le premier TP présenté : elle délivre les créneaux à fréquence variable et permet de prélever facilement les signaux qui nous intéressent.

Fig. 12 : Carte d'extension sur carte ARM7

D'autres cartes sont en cours d'étude. En particulier une carte de gestion d'énergie pour contrôler la charge et la décharge des batteries au plomb du robot et une carte SD Card+MP3. Cette dernière, construite autour d'un circuit codec VLSI [7], permet de réaliser un "balladeur" MP3 et de mettre en lumière nombre de points d'étude intéressants :

- dialogue avec un DSP externe utilisant les bus dédiés SPI et propriétaires nécessaires,
- fluidification d'un flot de données par l'utilisation adéquate de mémoire tampon,
- utilisation de bibliothèques libres pour l'implémentation du système de fichiers VFAT [6].

La gestion de l'énergie est un des points clefs d'un système embarqué. C'est aussi un thème d'étude transversal au département EEA. La carte mesure en permanence le courant traversant la batterie et la tension à ses bornes pour en déduire l'énergie soutirée en fonctionnement autonome ou injectée lors de la charge.

5.2 Extensions recherche

L'utilisation de cette carte n'a pas donné lieu au développement d'interfaces spécifiques lors de l'utilisation en recherche. En effet, les thèmes de recherches se succèdent et il est inutile de pérenniser ou dupliquer une maquette d'étude une fois les solutions matérielles et logicielles définies. Cette carte a été utilisée dans plusieurs TER pour la mise au point de maquettes de sous-systèmes qui sont ou seront intégrés dans divers projets dont RobAutiSTIC.

6 CONCLUSION

Depuis 2007, la carte (figure 13) a été utilisée comme support dans de nombreuses activités pédagogiques en TP et TER :

- systèmes temps réels embarqués et enfouis [3][4][5] (formation M2P génie des systèmes d'informatique industrielle, Université Paris XI-Orsay),
- préparation à l'agrégation génie électrique de l'ENS de Cachan,
- travaux d'étude et de recherche (M1 IST, Université Paris XI-Orsay) sur les sujets suivants :
  - détection des interactions enfant autiste-robot par mesure de la distance et de l'orientation de la tête de l'enfant,
  - système de détection de chute pour personne atteinte d'atrophie des membres inférieurs,
  - diffusion de comptines au format MP3 stockées sur SD Card pour le robot GIPY,
  - système d'enregistrement simultané du son et de coordonnées GPS, etc.

Le principal retour d'expérience en enseignement a trait à l'aspect concret des manipulations proposées. L'intérêt des participants s'est focalisé principalement sur la mise en évidence du fonctionnement du système d'interruption et l'évaluation de ses temps caractéristiques.

De même l'évaluation de la charge du processeur a permis de concrétiser des notions souvent floues dans l'esprit des étudiants.

L'ensemble de ces travaux a pour origine un transfert de connaissance recherche-enseignement. Les contraintes d'un projet de recherche ont été prises en compte et introduites dans un cadre pédagogique. En retour, des travaux d'élèves ont permis de tester la faisabilité de dispositifs qui sont ou seront intégrés dans des projets de recherche.

Fig. 13 : Carte ARM7 sans l'embase robotique

Référence ANR

Le projet RobAutiSTIC est financé par l'ANR dans le cadre de l'appel PSIROB06-174075

Bibliographie

- [1] <http://robautistic.ibisc.univ-evry.fr>.
- [2] http://www.nxp.com/acrobat_download/usermanuals/UM10120_1.pdf.
- [3] J.P. PEREZ, "Systèmes temps réels", *Dunod*, ISBN10 : 2-04-019724-9
- [4] Christian Bonnet, Isabelle Demeure, "Introduction aux systèmes temps réel", *Hermès - Lavoisier*, EAN13 : 9782746200166
- [5] F. Cottet, E. Grolleau, "Systèmes temps réels de contrôle-commande, conception et implémentation", *Dunod/L'Usine Nouvelle*, coll technique et ingénierie, EAN13 : 9782100078936
- [6] http://elm-chan.org/fsw/ff/00index_e.html
- [7] <http://www.vlsi.fi/fileadmin/datasheets/vlsi/vs1053.pdf>
- [8] <http://www.rowley.co.uk/arm/index.htm>