ÉCOLE POLYTECH'NICE SOPHIA
SI5
2014

Julien DeAntoni

Systèmes et applications embarqués

Programmation micro-contrôleur

Avec le RTOS AvrX

Introduction

Nous allons programmer la plaque arduino UNO R2 équipée d'un micro contrôleur ATMEL atmega328p avec un système d'exploitation temps réel (RTOS [1]). Un des plus petit RTOS fonctionnant sur le micro-contrôleur à notre disposition est AvrX, un RTOS très léger développé par Larry Barello (larry@barello.net). Malgré sa petite taille, il reprend les objets classiques d'un RTOS (tâche, sémaphore, ordonnanceur à priorité, ...)
La première étape va être de récupérer les sources modifiées de l'OS (adaptées à arduino et avr-g++) sur mon site personnel (http ://www-sop.inria.fr/aoste/personnel/Julien.Deantoni/enseignements/iam03/).
Uniquement si néceessaire [2], compiler le code situé dans le sous dossier avrx sous forme d'une bibliothèque (avrx.a) grâce au Makefile fourni Vous devrai aussi compiler les librairies arduino en allant dans le sous dossier *arduinoLibAndCore* et en compilant la bibliothèque (make libres.a).
Puisque nous avons peu de temps alloué, nous allons faire des programmes "par l'exemple". Ainsi, après avoir compilé AvrX vous testerez les exemples fournis dans le zip sur ma page internet (ledBlink_rtos.c). Ensuite vous pourrez les modifiez pour atteindre les objectifs donnés dans la suite.

1 Led Blink Multi-tâches

Cet exemple est fourni dans le .zip téléchargé pécédemment. Après vous être assuré que les branchements sont les même que ceux de la semaine précédente, compilez et testez ce programme sur la carte de développement.
La led sur D8 doit clignoter plus rapidement que celle sur D11.

Maintenant regardez votre code, comprenez le et posez vous quelques questions. En particulier, appréciez la facilité de réalisation d'une attente non active dans une tâche. Selon vous, ces tâches sont-elles périodiques ? Si non, pourquoi, si oui sur quelle période ?
Enlevez maintenant les attentes sur timer de la tâche 1. Que se passe t'il ? Changez ensuite la priorité de la tâche 1 pour qu'elle soit plus grande que celle de la tâche 2. Que pouvez vous conclure ?

2 Interruption logicielle

Maintenant, on désire scruter un bouton poussoir branché sur D9 lors de l'allumage de la led en D8. Si le bouton est appuyé, on lance le clignotement de la led sur D11 pour 5 cycles et on arrete ensuite le

1. Real Time Operating System
2. c'est à dire après un test sans recompiler

clignotement jusqu'au prochain appuie sur le bouton poussoir. Notez que si l'on reste pendant trois cycles d'allumage appuyé sur le bouton poussoir, la led branchée sur `D11` doit clignoter pendant 15 cycles.

3 Un "vrai" programme ?

On désire maintenant réagir en fonction de la luminosité ambiante. Pour ce faire on utilise un capteur de luminosité. Ce capteur renvoie une tension analogique qui augmente en fonction de la luminosité. Comme on est informaticien on n'aime pas beaucoup les tensions continues. Il est donc nécessaire de convertir cette valeur en numérique. Nous utiliserons donc le convertisseur analogique numérique présent dans le micro contrôleur (pages 252 à 268 des datasheets). Vous brancherez le capteur de luminosité sur la borne `A5`. Il vous sera nécessaire de paramètrer le convertisseur. voici les paramètres à lui appliquer :

- tension de référence égale à AV_{cc} → `REFS0`=1 et `REFS1`=0
- utilisation de `ADC1`
- fréquence de conversion à 125KHz (*prescaler* à 128 dans le registre `ADCSRA`)
- trigger automatique, *free running mode* (registre `ADCSRA` et `ADCSRB`)

Une fois paramètré, vous pourrez selectionné et demarrer le convertisseur (registre `ADCSRA`).
La valeur peut alors être lue dans `ADCW`. Cette lecture se fera dans une tâche périodique.
Si la valeur représentant la luminosité ambiante descend en dessous de 250, vous lancerez le clignotement de la led sur D11. Ce clignotement s'arretera sur deux conditions :

- soit la valeur représentant la luminosité ambiante remonte au dessus de 600 pendant plus de 5 secondes ;
- soit le bouton poussoir branché sur D9 est actionné. La détection d'une telle action se fera par l'utilisation d'une interruption matérielle externe et non par scrutation.

Si vous détectez un manque dans la spécification, proposez une solution.

Avant de vous lancez à corps et âmes perdus dans la programmation n'oubliez pas de réfléchir aux différentes tâches nécessaires, aux communication entre ces tâches, etc...

4 Que se passe t'il ?

il y a un programme nommé LCD_rtos.c sur la page du cours. Essayez de comprendre ce programme puis lancez le après avoir branché le LCD sur le port 2 de la carte. Que se passe t'il ? pourquoi ? Résoudre le problème si problème il y a...