

Introduction aux Systèmes et Logiciels Embarqués

Programmation micro-contrôleur

Avec FreeRTOS

Introduction

Nous allons programmer la plaque arduino UNO R2 équipée d'un micro contrôleur ATMEL atmega328p avec un système d'exploitation temps réel (RTOS¹). Ce micro-contrôleur est relativement complet mais nous n'expérimenterons dans ce TP qu'une partie de ses possibilités.

Un des RTOS les plus connus fonctionnant sur le micro-contrôleur est FreeRTOS, un RTOS très complet dont la documentation se trouve ici : <http://www.freertos.org/> (voir section). Il reprend les objets classiques d'un RTOS (tâche, sémaphore, ordonnanceur à priorité, ...) et beaucoup d'autres objets plus rares que je vous laisserai découvrir avec enthousiasme en dehors des séances.

reference API for FreeRTOS 10.3.1

La version de FreeRTOS qui a été portée sur Arduino est la : 10.3.1. Son API est référencée ici : <https://www.freertos.org/a00106.html>

1 Led Blink Multi-tâches

Cet exemple est fourni dans le fichier zip téléchargé sur ma page web. Pour savoir comment brancher les leds sur l'arduino, regardez dans le code. Compiler ce programme pour la carte de développement (*make*). Ensuite, *uploader* l'exécutable dans l'arduino en utilisant *make upload*. Vous pouvez aller regarder le makefile pour avoir une idée de ce qui est fait.

Les deux leds doivent clignoter, la rouge deux fois plus lentement que la verte.

Maintenant regardez votre code, comprenez le et posez vous quelques questions. Selon vous, ces tâches sont-elles périodiques ? Si non, pourquoi, si oui sur quelle période ?

Maintenant donnez une plus faible priorité à la tâche de la led verte. Que se passe t'il de visible ? d'invisible ?

Enlevez maintenant, un à un le delai de chacune des tâches. Que se passe t'il ? Changez ensuite la priorité de la tâche 2 pour qu'elle soit plus grande que celle de la tâche 1. Que pouvez vous conclure ?

info en vrac : vTaskDelayUntil

1. Real Time Operating System

2 Interruption logicielle

Maintenant, on désire scruter un bouton poussoir branché sur $D4$. Si le bouton est appuyé, on lance le clignotement de la led verte pour 5 cycles et on arrête ensuite le clignotement jusqu'au prochain appui sur le bouton poussoir. Notez que si l'on appuie trois fois de suite sur le bouton poussoir, la led verte doit clignoter pendant 15 cycles. Pensez à bien regarder le nom de cet exercice.

infos en vrac :xSemaphoreCreateCounting

3 Interruption logicielle bis

On désire simuler un interrupteur avec un bouton poussoir. Pour ce faire, on affichera sur le LCD la valeur de l'interrupteur (i.e., "button is on" ou "button is off"). De plus on allumera la led rouge lorsque l'état du bouton est "on".

Concernant le bouton poussoir, toujours branché sur $D4$, on désire le scruter. Si le bouton est appuyé une première fois, l'état est considéré comme "on", une deuxième fois comme "off", etc. Afin de visualiser la fréquence de scrutation du bouton, la led verte doit changer d'état à chaque scrutation. La scrutation devra se faire toutes les 100ms.

4 Interruption logicielle v2

périodiquement, toutes les 4 secondes, faire clignoter la led rouge pour 5 cycles de chacun 300ms (150ms allumée, 150ms éteinte) et on arrête ensuite le clignotement jusqu'à la prochaine période. De plus, toutes les 2 secondes on veut changer l'état de la led verte. Enfin, lors du démarrage du troisième cycle de clignotement de la led rouge, on désire émettre un **bref** (environ 50ms) BIP sur le buzzer branché en $D7$ (sur le simulateur le buzzer est remplacé par une led pour une question de facilité).

5 Interruption logicielle et matérielle

Maintenant, on désire faire clignoter une LED rouge à une vitesse prédéfinie au démarrage. En même temps on souhaite réagir à l'appui sur le bouton poussoir branché en $D4$ (interruption). Lors de l'appui sur le bouton, la vitesse de clignotement doit être diminuée de $1/3$.

6 Un "vrai" programme ?

On désire maintenant réagir en fonction de la luminosité ambiante. Pour ce faire on utilise un capteur de luminosité utilisé lors des premiers TDs. La lecture de la luminosité 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 rouge. Ce clignotement s'arrêtera 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 $D4$ 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...