

Systemes informatiques II

LINGI1113

Professeur : Marc Lobelle
Assistants : Christoph Paasch, Fabien Duchêne

Projet 2 : exercice sur machine nue



Arnaud Schils (SINF13BA)
Julien Bergiers (SINF13BA)

Année académique 2011-2012

• Documentation pour l'utilisateur

Le réveil est composé de 2 boutons, MENU/NEXT (bouton du haut) et ADD/CHOOSE (bouton du bas) et d'un écran LCD pour afficher l'heure, l'état de l'alarme et le menu.

Lors de l'allumage du réveil, il vous est demandé d'indiquer l'heure. Pour modifier les minutes, il suffit simplement de presser le bouton ADD et ensuite, NEXT pour passer aux heures. Par défaut, l'alarme est désactivée. Si vous désirez l'activer/désactiver dans l'écran d'affichage de l'heure, pressez le bouton ADD (il vous est aussi possible de la modifier dans le menu).

L'ouverture du menu à partir de l'écran d'affichage de l'heure s'effectue en pressant le bouton MENU. Après avoir parcouru les différentes sections du menu en pressant le bouton NEXT, vous pouvez entrer dans une de celles-ci en pressant le bouton CHOOSE. Ensuite, comme expliqué ci-dessus, la modification des minutes et heures s'effectue en utilisant les boutons ADD et NEXT.

Lorsque le réveil sonne, il suffit de presser un des deux boutons pour le désactiver.

• Documentation pour l'installateur

Deux programmes ont été conçus : le premier permet de déterminer la fréquence d'horloge du pic, le second est le réveil. Les deux programmes se trouvent dans leur propre dossier, dans le fichier test.c. Nous avons gardé le nom des fichiers du dossier exemple « testPic » fourni afin de pouvoir réutiliser le makefile.s

Il suffit donc pour installer un des programmes de lancer la commande make dans son dossier respectif et de transférer le fichier compilé sur le pic par la procédure habituelle.

• Documentation pour le programmeur

a) Programme déterminant la fréquence du pic :

Afin d'atteindre une meilleure précision pour la réalisation du programme « réveil » il nous fallait d'abord trouver la fréquence du pic. Le programme find_freq affiche le nombre d'instructions effectuées par le pic en une seconde.

Pour trouver ce nombre nous avons configuré le timer1 pour qu'il overflow lorsqu'une seconde s'est écoulée et le timer0 pour qu'il overflow après une seconde (ceci est possible car nous connaissons approximativement la fréquence du pic, 25 ou 40

MHz). Notons que pour ce faire nous avons utilisé le prescaler du timer0 (réglé sur un ratio de 1:256).

Nous lançons le timer1 et le timer0 l'un après l'autre. Une fois une seconde écoulée le timer1 overflow et génère une interruption. Dans la méthode d'interruption nous lisons la valeur du timer0.

La fréquence du pic est alors facilement déterminable : $F_{osc} = \text{nbre_instr} * 4 * 256$

Le prescaler étant réglé sur 256 et chaque instruction étant exécutée en 4 cycles par le microprocesseur du pic.

Cette mesure n'est cependant pas parfaite pour plusieurs raisons :

- Le timer0 continue à tourner pendant le passage du contexte principal à celui de la routine d'interruption.
- Le prescaler réglé sur 256 induit une erreur sur la mesure de 255 (lorsqu'on lit la valeur du timer0 le timer était peut-être à 1 instruction de la prochaine incrémentation du timer0, la valeur lue est donc inférieure de 255 à ce qu'elle devrait être) .

En utilisant cette méthode nous avons trouvé une fréquence de 26,2MHz.

b) Programme réveil :

Le programme comporte les fonctions suivantes:

- Réglage de l'heure (au démarrage du pic et à n'importe quel moment à partir du menu).
- Activation / désactivation du réveil avec choix de l'heure de l'alarme (à partir du menu) .

A l'aide d'un State Diagram, nous avons identifié 9 états importants (qui sont repris au tout début du code dans les #define). Ces états ont permis de gérer toutes les éventualités et de rendre notre code lisible et logique. Nous les retrouvons dans la gestion des interruptions levées par les boutons ainsi que dans la mise à jour de l'écran LCD.

La gestion de l'heure se fait par interruption. Le timer0 est configuré par le prescaler (1:128) et sa valeur initiale pour overflow toutes les secondes. Une fois que le timer0 a overflow on le remet à sa valeur initiale (14364). Lors de cette opération un manque de précision est possible, on écrase le temps écoulé lors du passage du contexte du programme à celui de la procédure d'interruption.

Peut-être l'attente active permettrait d'avoir plus de précision. Cependant nous avons choisi d'utiliser les interruptions plutôt que l'attente active car cette dernière fait travailler le microprocesseur et gaspille donc des cycles CPU. De plus cette technique nous paraît un peu plus « propre ».