

OT Setre - TP1

MSP430 : Prise en main de la chaîne de compilation

Durée encadrée : 4 heures

But du TP

Le but de ce TP est de vous familiariser avec l'environnement de développement basé sur `gcc` et `insight`, et de comprendre les mécanismes de base du fonctionnement du MSP430.

Les commandes que vous aurez à exécuter : pour compiler : `msp430-gcc`, pour insight : `msp430-insight`.

1 Affichage de leds

Récupérez le fichier `tp1.tar` (instruction données sur place) extrayez l'archive. L'archive comporte trois répertoires avec des programmes pour le MSP430 permettant : d'afficher une led à intervalles régulier, de faire fonctionner le bargraph de la carte interface et de faire fonctionner le LCD. Nous étudierons essentiellement le premier, les autres vous seront utiles pour la suite de l'OT.

1.1 LED et Timer

Placez vous dans le répertoire `timer-ot2007`. Éditez le fichier `main.c` et parcourez le pour comprendre ce qu'il fait. On n'analysera pas précisément la fonction `set_timer()` pour l'instant, cette fonction initialise un timer qui envoie une interruption régulièrement.

1. En vous reportant au plan de la carte, repérez le port où est connectée la LED.
2. Que fait la fonction `delay(unsigned int d)` ?
3. Que fait la fonction `prvTickISR(void)`, en quoi est-elle différentes des autres fonctions.
4. Que doit faire ce programme ?
5. Compiler le programme (commande `make`), analyser les différentes commandes effectuées, quel est le fichier exécutable à télé-charger sur le MSP430 ?

1.2 Exécution sur la carte

(si les cartes sont là, sinon passer au suivant)

1. Vérifier les connexions entre le PC et la carte.
2. Dans une fenêtre `bash`, lancer la commande : `msp430-gdbproxy --port=2159 msp430`
Cette commande démarre un processus qui fait l'interface entre d'une part la carte connectée par le JTAG sur le port parallèle et d'autre part une connexion TCP sur le port 2159 du PC (actuellement, `gdbproxy` attend qu'un processus se connecte sur ce port).
3. Lancer, depuis un autre shell `msp430-insight` sur l'exécutable compilé.
4. Dans l'onglet `File/Target Setting`, choisissez `Remote/TCP` remplissez le champ `hostname` avec la valeur `localhost` et le champ `Port` par 2159. Connectez le simulateur (onglet `Run/Connect to target`), observez les résultat dans la fenêtre de `gdbproxy`.
5. Faites apparaître la console `gdb` de `msp430-insight` (onglet `View/Console`). Dans la console `gdb` taper `monitor erase all`. Depuis `msp430-insight` chargez le programme (onglet `Run/Download`).
6. Lancez le programme (onglet `Run/Run`). GDB a mis un point d'arrêt au début du `main`, continuez l'exécution et vérifiez que la fréquence de changement des chiffres correspond à ce que vous avez calculé.

1.3 Simulation avec `msp430-insight`

1. Quittez et relancez `msp430-insight`
2. Dans l'onglet `File/Target Setting`, choisissez `simulation`
3. Connectez (onglet `Run/Connect to target`)
4. Exécutez le programme (`Run/Run`), pouvez-vous vérifier qu'il fonctionne correctement ?

1.4 Simulation avec `msp430-insight` et `wsim-ot2007`

`wsim-ot2007` est un simulateur du MSP430 qui intègre certains périphériques présent sur la carte ainsi que sur la carte interface parmi lesquels le timer, la led, le bargraph et le LCD. `wsim-ot2007` peut être connecté à `msp430-gdb` par le même mécanisme que lorsque `msp430-insight` se connecte à la carte : connections TCP.

1. Lancer `wsim-ot2007` avec la commande suivante : `wsim-ot2007 -ui -gdb=2159 executable.elf`
Le programme se met en attente d'une connexion sur le port 2159.
2. Lancez `msp430-insight executable.elf`
3. Dans l'onglet `File/Target Setting`, choisissez `Remote/TCP` avec `localhost` comme `hostname` et 2159 comme port.
4. Connectez-vous, Exécutez le programme. Vous pouvez maintenant vérifier qu'il fonctionne correctement. Le débugage en simulation évite toute la manipulation de recharge du programme à chaque compilation, il permet aussi de débbuger sans avoir physiquement la carte.

2 Programmation du timer

Nous allons étudier la programmation du timer, nous référenceront les pages du manuel utilisateur du MSP430f149 (fichier `MSP430x14xxusersguide.pdf`). Le MSP430 contient 4 timers qui fonctionnent sur le même principe : ils sont initialisés à une certaine valeur, ils incrémentent (ou décrémentent) un registre à chaque top d'horloge et envoient une interruption lorsque ce registre atteint une certaine valeur. Le *watchdog timer* envoie systématiquement un reset, il est utilisé pour maintenir les systèmes en vie, si le programme ne réinitialise pas régulièrement le timer, le MSP est rebouté. Les timers *timer_A* et *timer_B* peuvent être utilisés par le MSP.

Le timer_A (doc chapitre 11) est contrôlé par un registre nommé TACTL, le registre qui "compte" est nommé TAR (TBCTL et TBR pour le timer_B). La première instruction de la fonction `timerA_start()` assure que le timer est arrêté (mise à 0 du registre TACTL). La deuxième instruction : `(TACTL = TASSEL_1)` sélectionne l'horloge utilisée pour incrémenter le timer, ici l'horloge ACLK. L'instruction suivante : `TACTL |= TACLRL` initialise le compteur à 0. Le registre TACCRO est celui qui est comparé au compteur, ici il est mis à la valeur : $4 * 10^6 / 100 = 40000$, soit une interruption toutes les $40000 / (4 * 10^6) = 10^{-2} s$ (10 millisecondes).

Le registre TACCTLO (capture/compare control register) positionne la manière dont doit être *capturée* l'horloge (front montant, front descendant,...) et si le compteur doit émettre une interruption. L'instruction `TACCTLO = CCIE` le positionne dans le mode capture et émission d'interruption. L'instruction `TACTL |= ID_x` indique si l'horloge doit être divisée (incrément du compteur tout les x tops d'horloge), ici elle n'est pas divisée. Ensuite on re-vérifie que le timer est éteint, puis on lance le compteur en mode "up" : incrémentation jusqu'à la valeur de TACCRO.

- Modifiez le programme pour que la LED clignote différemment (un S.O.S en morse par exemple).

3 Fonctionnement du bargraph et des boutons

Dans le répertoire `ot2007-1eds`, le programme permet de faire fonctionner le bargraph présent sur la carte par l'intermédiaire des boutons. exécutez le programme en simulation avec `wsim-ot2007`, les boutons R0, R1, R2, R3 sont les lettres 'a', 'z', 'e', 'r'. Il faut appuyer assez longtemps sur chacun des boutons. commencez par appuyer sur les 4 boutons puis suivez l'exécution en appuyant indifféremment sur les 4 boutons.

4 Fonctionnement du LCD

Dans le répertoire LCD, le programme propose des primitives permettant d'afficher une chaîne de caractère sur le LCD de la carte. le LCD n'est pas encore implémenté dans

`wsim-ot2007`, exécutez donc ce programme directement sur la carte. Analysez le code source.