

TD 2 - Utilisation avancée de MPLAB

Objectifs :

- utilisation des fonctionnalités avancées de MPLAB (simulation, watch, ...)
- programmation assembleur : tests de bits, branchements, sous-programmes.

1 Manipulation guidée

A partir du programme de la semaine dernière, créer un nouveau fichier (TD.2A.s), et saisissez dans la partie "boucle principale" le programme suivant :

```

1 loop:  bset  PORTA,#3
2         bclr  PORTA,#3
3         bra   loop

```

Que fait ce programme (relire TD1) ? Effectuez le build, et vérifiez qu'il n'y a pas d'erreurs.

1.1 Exécution sur la cible réelle

1. Sélectionner le programmeur ICD2, et transférer le programme dans la carte. En exécutant le programme (Run), que constate-t-on ? Peut-on observer le clignotement des leds ? Pourquoi ?
2. Positionner un point d'arrêt sur l'une des lignes (double clic sur la ligne : un B apparaît dans la marge). Relancer l'exécution, et vérifier que l'exécution s'arrête. Continuer en pas à pas, et vérifier le fonctionnement.
3. Ouvrir la fenêtre "Watch" (menu View). Cette fenêtre permet d'observer à chaque arrêt de l'exécution (point d'arrêt ou pas-à-pas) la valeur des registres de contrôles du processeur, ou des variables du programme.
4. Dans la liste déroulante, choisir PORTA, et cliquer sur Add SFR : le registre apparaît dans la fenêtre, avec sa valeur actuelle.
5. Exécuter le programme en pas-à-pas, et vérifier l'évolution de la valeur.
6. Cette fenêtre sert également en écriture : cliquer sur le champ "Value", et saisissez 0x00f0, validez puis 0x000f, tout en observant l'état des leds sur la carte.

1.2 Exécution en simulation

On souhaite profiter de ce programme pour vérifier le temps d'exécution d'un programme. Dans le jeu d'instruction du processeur (p.192 de la doc), on peut trouver le nombre de cycles de chaque instruction (colonne "# of cycles"). On utilisera ensuite le simulateur pour mesurer la durée d'exécution.

1. Rechercher dans le jeu d'instruction du PIC le nombre de cycles d'horloge utilisés pour chacune des instructions de la boucle, et les noter dans le cadre de gauche.
2. Sélectionner le simulateur : menu Debugger → Select Tool → MPLAB SIM
3. Menu Debugger → Settings : Dans l'onglet Osc / Trace, cochez "Trace All", et donner une fréquence de 4 MHz comme fréquence d'horloge du processeur (identique à celle de la carte Explorer 16).
4. Validez l'affichage de l'analyseur logique : Menu View → Simulator Logic Analyser
5. Dans la fenêtre "Logic Analyser", sélectionner la broche 3 du port A, en cliquant sur "Channels"
6. Lancer la simulation en cliquant sur le bouton "Animate", puis "Halt" (rapidement !). Vérifier dans l'analyseur logique l'état de RA3, et mesurer avec le curseur la durée d'un cycle complet. La valeur obtenue est-elle conforme au résultat attendu ?

Note vous pouvez :

- effacer les chronogrammes de la fenêtres via un clic-droit → Clear ;
- ré-initialiser l'origine à zéro en cliquant sur le bouton "Reset" (barre d'outils principale) ;
- zoomer et utiliser un curseur horizontal via la barre d'outil de la fenêtre "Logic Analyser".

2 Travail à effectuer

1. Proposer un programme (TD.2B.s) qui va réaliser un "chenillard" (led allumée se décalant de gauche à droite).
2. Modifier ensuite le programme pour un appui sur S3 qui puisse alternativement provoquer la mise en route et l'arrêt.

Compléments assembleur

1. Appel d'un sous programme : `call etiquette`
2. Retour au prog. appelant : `return`
3. Décrémentation : `dec Wn, Wm ; Wm = Wn - 1`
4. Répétition instruction suivante : `repeat #n`
5. Rien (No OPeration) : `nop`