

– *Formation EasyPIC* – projets à réaliser

1. Matériel nécessaire

Tous ces exercices ont été testés avec la carte EasyPIC4 de MikroElektronika, sous sa configuration initiale (PIC 16F877A cadencé à 8 MHz).

Sauf contre-indication, aucun cavalier ou interrupteur n'est à modifier.

Les références fournies sont relatives à la documentation du 16F877A (code DS39582). Il existe cependant une seconde documentation importante, celle de la famille 16F8xx (code DS33023). Enfin, l'onglet « **QHelp** » de MikroC est souvent très utile !

On considère que les apprenants ont déjà travaillé avec le langage C ou un autre langage de haut niveau ; les objectifs se limitent donc à une mise en œuvre du processeur, de la carte de développement et de l'IDE.

1. Lien entre BP et DEL

Objectifs :

- mettre en œuvre les entrées du port A (ainsi que leur configuration) ;
- mettre en œuvre les sorties du port B (ainsi que leur configuration).

Programme à réaliser :

Allumer les DEL RB0 et RB1 en fonction des entrées RA0 et RA1 (un BP appuyé entraîne l'allumage de la DEL correspondante).

Références à utiliser :

- table 4-1 : port A functions ~ page 43 ;
- table 4-2 : summary of registers associated with port A ~ page 43 ;
- table 4-3 : port B functions ~ page 45 ;
- table 4-4 : summary of registers associated with port B ~ page 45 ;
- register 11-2 : ADCon1 register ~ page 128.

Remarques :

- ne pas oublier le réglage de ADCON1 ;
- pour tous les ports, le registre de direction est nommé TRISx. Un bit à 1 place la broche correspondante en entrée, un bit à 0 la place en sortie.

Solution :

Voir le fichier « formation_bp_del.c ».

2. Clignotement d'une DEL

Objectifs :

- mettre en œuvre les sorties du port A (ainsi que leur configuration) ;
- mettre en œuvre les bibliothèques fournies.

Programme à réaliser :

Faire clignoter la DEL RA0 à une fréquence d'approximativement 1 Hz. Faire en sorte qu'un appui sur le BP RA1 double la fréquence de clignotement.

Références à utiliser :

- table 4-1 : port A functions ~ page 43 ;
- table 4-2 : summary of registers associated with port A ~ page 43 ;
- register 11-2 : ADCon1 register ~ page 128.

Remarque :

Ne pas oublier le réglage de ADCON1.

Solution :

Voir le fichier « formation_clignotement.c ».

3. Ecriture et lecture en EEPROM

Objectifs :

- mettre en œuvre les sorties du port B (ainsi que leur configuration) ;
- mettre en œuvre la EEPROM ;
- mettre en œuvre les bibliothèques fournies.

Programme à réaliser :

Stocker les valeurs !128 (%10000000) à !143 (%10001111) en EEPROM, puis les rappeler une par une par pas de 0,5 s. Leur rappel se matérialisera par l'allumage séquentiel des DEL du port B.

Références à utiliser :

- table 4-3 : port B functions ~ page 45 ;
- table 4-4 : summary of registers associated with port B ~ page 45.

Remarque :

Respecter une temporisation de 20 ms entre une écriture et une lecture (nombre pessimiste car il semble que 5 à 6 ms suffisent).

Solution :

Voir le fichier « formation_eeprom.c ».

4. Gestion MLI

Objectifs :

- mettre en œuvre la gestion de la MLI (port C) ;
- mettre en œuvre les bibliothèques fournies.

Programme à réaliser :

Générer un signal PWM de fréquence 1 kHz en RC2. Réaliser une boucle infinie qui permette de faire varier le rapport cyclique entre 10 % et 70 % par paliers de 10 %, puis entre 70 % et 10 %

par paliers de 5 %. L'évolution se fera toutes les 500 ms.

Solution :

Voir le fichier « formation_pwm.c ».

5. Voltmètre

Objectifs :

- mettre en œuvre le module CAN ;
- mettre en œuvre l'afficheur alphanumérique.

Programme à réaliser :

Lire la valeur fournie par la broche RA3, la traiter et l'afficher sur l'afficheur alphanumérique. Faire suivre la valeur de la lettre V. Dans un premier temps, ne pas se soucier de la fréquence de rafraîchissement de la valeur ; éventuellement, par la suite, proposer un rafraîchissement de l'ordre du hertz.

Remarques :

- il est arrivé que le LCD ne réagisse pas correctement aux ordres prévus (notamment en ce qui concerne le positionnement des informations à afficher, après avoir utilisé le GLCD). Dans ce cas, déconnecter le câble USB de l'EasyPIC, puis son alimentation (le cas échéant), et reconnecter le tout ;
- cavalier JP15 à placer pour utiliser le potentiomètre P1.

Solution :

Voir le fichier « formation_voltmeter.c ».

6. Oscilloscope

Objectifs :

- mettre en œuvre le module CAN ;
- mettre en œuvre l'afficheur graphique.

Programme à réaliser :

Lire la valeur fournie par la broche RA3, la traiter et l'afficher sur l'afficheur graphique sous la forme d'un chronogramme. Ne pas se soucier de la fréquence de rafraîchissement de la valeur.

Remarques :

- sans aucune gestion des calibres, on peut afficher un signal compris entre 0 et 5 V, avec une fréquence de l'ordre du hertz ;
- cavalier JP15 à placer pour utiliser le potentiomètre P1.

Solution :

Voir le fichier « formation_can_scope.c ».

7. Génération d'un signal de 1 kHz de précision

Objectifs :

- mettre en œuvre le module *timer* ;
- mettre en œuvre les interruptions logicielles.

Programme à réaliser :

Générer, sur RA0, un signal TTL de fréquence 1 kHz et de rapport cyclique 50 % en utilisant le *timer* interne (*timer0*) et ses interruptions de débordement.

Ensuite, grâce à un oscilloscope, ajuster la fréquence pour avoir un maximum de précision.

Références à utiliser :

- table 4-1 : port A functions ~ page 43 ;
- table 4-2 : summary of registers associated with port A ~ page 43 ;
- register 11-2 : ADCon1 register ~ page 128 ;
- register 5-1 : Option_Reg register ~ page 54 ;
- table 5-1 : registers associated with timer0 ~ page 55 ;
- register 2-3 : IntCon register ~ page 24.

Remarque :

La différence entre calcul et réel vient du fait que le temps des instructions autour de l'interruption n'a pas été pris en compte ; il est difficile d'être précis temporellement en langage de haut niveau.

Solution :

Voir le fichier « formation_1_khz.c ».