

Série d'exercices 4 : Les systèmes de développement (IDE)

Exercice 1 : Les étapes du cycle de développement

Remettez dans l'ordre les 5 étapes suivantes du cycle de développement avec un IDE comme MPLAB

:

1. Exécution / débogage
2. Édition de liens (linking)
3. Création du projet
4. Compilation / assemblage
5. Écriture du code source

Exercice 2 : Vocabulaire de l'IDE

Associez chaque terme à sa définition :

- **IDE**
- **Linker**
- **Hex file**
- **Programmeur**
- **Breakpoint**

Définitions :

- A. Fichier exécutable à charger dans le microcontrôleur
- B. Outil qui combine plusieurs fichiers objets en un exécutable
- C. Environnement regroupant éditeur, compilateur, débogueur
- D. Point d'arrêt pour suspendre l'exécution
- E. Outil matériel ou logiciel pour transférer le code vers la mémoire du μC

Exercice 3 : Points d'arrêt et exécution pas à pas

Vous déboguez un programme sur MPLAB. Le compteur principal (variable `compteur`) doit s'incrémenter de 1 à 10. Mais il bloque à 5.

Voici le code suspect :

```
int compteur = 0;
for(int i=0; i<10; i++) {
    compteur++;
    if(compteur == 5) {
```

```
while(1); // boucle infinie
```

```
}
```

```
}
```

- Où placeriez-vous un point d'arrêt pour diagnostiquer le problème ?
- Que montre l'exécution pas à pas ?
- Corrigez le code.

Exercice 4 : Visualisation des registres

Sur un PIC16F877, vous voulez surveiller le registre PORTA et le registre TRISA en debug.

- Quels sont les rôles de ces deux registres ?
- Lors de l'exécution pas à pas, vous voyez $PORTA = 0x05$ et $TRISA = 0x00$. Que se passe-t-il si vous écrivez $PORTA = 0x02$?
- Comment visualiser ces registres dans MPLAB X IDE ?

Exercice 5 : Chargement du fichier dans la mémoire

Après compilation, vous obtenez un fichier projet.hex.

- Quelle est son utilité ?
- Citez deux méthodes pour le charger dans le microcontrôleur avec MPLAB.
- Que signifie l'erreur « Target device not found » lors du chargement ?

Exercice 6 : Mise en situation – Projet complet

Énoncé :

Vous devez développer un clignotement de LED sur une carte PIC avec MPLAB. Décrivez précisément les actions à mener pour :

- Créer le projet
- Écrire le code source
- Compiler sans erreur
- Charger le programme
- Débugger (la LED clignote trop vite, on ne la voit pas)

Code de base :

```
#include <xc.h>
#define _XTAL_FREQ 4000000
void main() {
    TRISB0 = 0;
    while(1) {
        RB0 = 1;
        __delay_ms(10);
        RB0 = 0;
        __delay_ms(10);
    }
}
```

Proposez une solution pour ralentir le clignotement.

Solution Série 4

Exercice 1 :

Ordre correct :

1. Création du projet
2. Écriture du code source
3. Compilation / assemblage
4. Édition de liens
5. Exécution / débogage

Exercice 2 :

IDE → C

Linker → B

Hex file → A

Programmeur → E

Breakpoint → D

Exercice 3 :

- a) Placer un point d'arrêt à la ligne `if(compteur == 5)` ou à l'intérieur du `while(1)`.
- b) Le programme entre dans la boucle infinie quand compteur atteint 5.
- c) Corrigé : supprimer le `while(1)`; ou le remplacer par une condition correcte.

```
int compteur = 0;
```

```
for(int i=0; i<10; i++) {
```

```
    compteur++;
```

```
    // if(compteur == 5) { while(1); } // erreur supprimée
```

```
}
```

Exercice 4 :

- a) TRISA configure les directions (0 = sortie, 1 = entrée). PORTA contient les valeurs sur les broches en sortie ou lit l'état en entrée.
- b) TRISA = 0x00 → toutes les broches de PORTA sont en sortie. Écrire PORTA = 0x02 met la broche RA1 à 1, RA0 à 0.
- c) Dans MPLAB X : Fenêtre → Debugging → Variables, ou ajouter un watch sur PORTA et TRISA. Ou utiliser la vue SFR (Special Function Registers).

Exercice 5 :

a) projet.hex contient le code machine et les données en format hexadécimal Intel, prêt à être programmé dans la mémoire flash du μ C.

b) Méthodes :

- Utiliser un programmeur matériel (ex. PICkit 4, ICD 4) via MPLAB IPE.
- Utiliser le débogueur intégré (ex. PICkit 3 en mode programmation).
- c) L'IDE ne détecte pas le microcontrôleur : vérifier l'alimentation, la connexion du programmeur, le choix du bon outil dans les paramètres du projet.

Exercice 6 :

1. Créer le projet :

- Fichier → Nouveau projet → Microchip Embedded → Standalone Project
- Choisir la famille (ex. PIC16F) et le modèle exact du μ C
- Choisir le programmeur (ex. PICkit 4)
- Choisir le compilateur XC8

2. **Écrire le code** : copier le code fourni dans main.c

3. **Compiler** : cliquer sur l'icône « Make and Build Project » (marteau). Vérifier l'absence d'erreurs dans la fenêtre Output.

4. **Charger** : cliquer sur « Make and Program Device » (flèche verte)

5. **Déboguer** :

- La LED clignote trop vite (10 ms).
- Solution : augmenter les délais, ex. `__delay_ms(500)` pour 0,5 seconde.
- En mode debug : placer un point d'arrêt sur `RB0 = 1` et exécuter pas à pas pour voir la LED s'allumer, puis passer à l'étape suivante.

Code corrigé :

```
#include <xc.h>
#define _XTAL_FREQ 4000000
void main() {
    TRISB0 = 0;
    while(1) {
        RB0 = 1;
        __delay_ms(500);
        RB0 = 0;
```

Université Djilali BOUNAAMA, Khemis Miliana
مليانة خميس بونعامه جيلالي جامعة
Faculté des Sciences et de la Technologie
والتكنولوجيا العلوم كلية



```
    __delay_ms(500);  
  }  
}
```