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

TP0 : Prise en main de l'environnement de programmation du microcontrôleur (1h30)

Objectifs pédagogiques

- Découvrir l'interface et les fonctionnalités de l'environnement de développement intégré (IDE).
- Maîtriser la création et la configuration d'un projet pour un microcontrôleur cible.
- Comprendre les étapes de la chaîne de compilation (compilation, assemblage, édition de liens).
- Apprendre à programmer (flasher) la carte cible.
- S'initier aux outils de débogage de base (points d'arrêt, exécution pas à pas, visualisation des registres).

Matériel requis

- Carte de développement (ex: carte avec microcontrôleur PIC, AVR, STM32 selon l'établissement)
- Programmateurs/débogueurs (ex: PICkit, ST-Link, USBasp)
- Ordinateur avec IDE installé (MPLAB X, Keil, STM32CubeIDE, Arduino IDE selon la cible)
- Câbles USB de programmation
- Documentation technique du microcontrôleur (datasheet)

Prérequis

- Aucun prérequis technique spécifique pour ce TP d'introduction.

Déroulement détaillé

Phase 1 : Découverte de l'IDE (20 minutes)

Les étudiants lancent l'IDE et observent l'interface : barre de menus, barre d'outils, explorateur de projets, éditeur de code, fenêtre de sortie (compilation, messages). L'enseignant présente brièvement les différentes zones et leur utilité. Les étudiants identifient les paramètres de configuration globaux : sélection du compilateur, chemins d'accès aux outils.

Phase 2 : Création d'un nouveau projet (25 minutes)

Chaque étudiant crée un nouveau projet en suivant les étapes guidées :

- Choix du type de projet (standalone, application, etc.)
- Sélection du microcontrôleur exact (référence complète)
- Sélection du programmateur/débogueur utilisé
- Choix du compilateur (assembleur ou C selon l'orientation)
- Définition du nom du projet et de son emplacement de sauvegarde

Une fois le projet créé, les étudiants examinent la structure : fichiers sources, fichiers d'en-tête, fichiers de configuration du projet.

Phase 3 : Écriture du programme minimal (20 minutes)

Les étudiants écrivent un programme minimal en assembleur ou en C (selon le niveau) :

```
// Exemple en C pour un PIC
#include <xc.h>
void main(void) {
   // Initialisation (vide pour l'instant)
   while(1) {
      // Boucle infinie
      NOP(); // Instruction ne faisant rien
   }
}
```

chaque élément est expliqué : la directive d'inclusion, la fonction main, la boucle infinie indispensable pour éviter que le programme ne s'égare.

Phase 4 : Compilation et analyse des messages (15 minutes)

Les étudiants lancent la compilation du projet. Ils observent la fenêtre de sortie et apprennent à interpréter les messages :

- Messages d'information (compilation réussie)
- Avertissements (warnings) : à ne pas ignorer
- Erreurs (errors) : empêchent la génération du fichier exécutable

Ils localisent les fichiers générés : fichier hexadécimal (.hex), fichiers de listing, fichiers de débogage.

Phase 5 : Programmation de la carte cible (15 minutes)

Les étudiants connectent le programmateur à la carte et à l'ordinateur. Ils lancent l'outil de programmation intégré à l'IDE, sélectionnent le fichier .hex à charger, et déclenchent l'opération de programmation. Ils observent les étapes : effacement de la mémoire Flash, programmation, vérification. Un message de succès confirme la bonne opération.

Phase 6 : Premiers pas en débogage (15 minutes)

Les étudiants passent en mode débogage. Ils apprennent à :

- Placer un point d'arrêt (breakpoint) sur l'instruction NOP() dans la boucle
- Lancer l'exécution jusqu'au point d'arrêt
- Effectuer une exécution pas à pas (step over, step into)
- Ouvrir une fenêtre de visualisation des registres (Watch window)
- Observer les valeurs des registres généraux et des registres spéciaux (SFR)

Ils vérifient que le programme atteint bien le point d'arrêt et qu'il s'exécute correctement.