

---

REPUBLIQUE ALGERENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'enseignement supérieur et de la recherche scientifique  
Université Badji Mokhtar Annaba  
Faculté de technologie  
Département d'électronique



## Travaux Pratiques Mp Mc

### TP n° 1

#### Prise en main de l'environnement de programmation du microcontrôleur

Ces Travaux pratiques ont été réalisés pour le module TP Microprocesseurs et Microcontrôleurs de la troisième année licence automatique, Pour l'année universitaire 2024/2025 au département d'électronique, Université Badji Mokhtar ANNABA.

**Par Dr. MERABTI Nardjes**

**Objectif :** L'objectif capital de ce premier TP est de se familiariser avec les outils employés pour réaliser les TPs de ce module. Nous nous intéressons à la programmation du microcontrôleur PIC de Microship (**PIC16F84**) et aux environnements de conception et de simulation (MPLAB et Proteus ) et donc utiliser les fonctions de base de l'outil de développement MPLAB pour:

- La création d'un projet
- L'edition d'un programme
- La compilation, débogage et la simulation d'un programme avec Mplab

■ **Présentation générale du PIC 16F84 de Microchip** : est un microcontrôleur 8 bits développé par Microchip Technology, Il est largement utilisé dans les projets électroniques amateurs et professionnels en raison de sa simplicité, de sa faible consommation d'énergie et de sa disponibilité sur le marché.

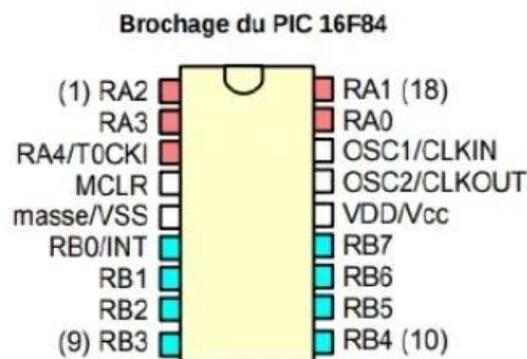


Figure 1 : Brochage du 16F84

Un **PORT** sert pour entrer et sortir des informations. Le 16F84 est doté de deux **PORTS A et B**.

- ❖ **Le PORTA** est constitué de **5 pins bidirectionnels: RA0 à RA4**.
- ❖ **Le PORT B** est constitué de **8 pins bidirectionnels: RB0 à RB7**.

La configuration des pins des PORTS A et/ou B en entrées ou en sorties est assurée par un registre de direction appelé **TRISA** associé au **PORT** et **TRISB** associé au **PORTB**

- **Ressources**

PC doté de logiciel MPLAB et du Proteus

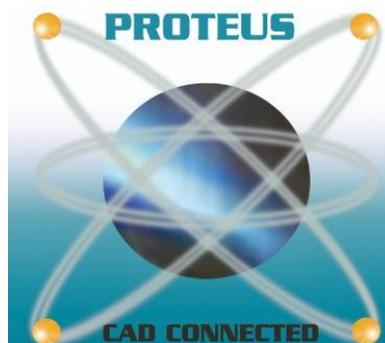
- **Logiciel MPLAB de Microchip:**



MPLAB est un environnement gratuit de programmation pour les microcontrôleurs de la famille PIC fourni par le constructeur Microchip. L'environnement intègre un éditeur d'assembleur, un débogueur complet ainsi que tous les outils permettant de gérer les programmeurs de composants. Il est gratuitement téléchargeable et utilisable.

Cet environnement de développement intégré (IDE) est indispensable pour écrire, compiler et déboguer un code pour le PIC16F84.

- **Proteus :**



C'est un très bon logiciel de simulation en électronique. C'est un éditeur de

schémas qui intègre un simulateur analogique, logique ou mixte. Il est capable de vérifier et simuler le comportement du PIC 16F84. En effet, le simulateur vous permettra de vérifier le fonctionnement de votre circuit avant même de le construire physiquement. Ceci permettra d'accélérer le prototypage et de réduire le coût de réalisation

Plus de détails dans l'annexe (*Cours de Proteus professional (ISIS & ARES)* Par TOURE

Med Lamine)

### ✧ **Les premières étapes :**

#### **1. Installation des logiciels :**

**MPLAB IDE :** Téléchargez une version sur le site de Microchip. Suivez les instructions d'installation.

**Proteus :** Procurez-vous la version étudiante ou professionnelle, selon vos besoins. L'installation est généralement simple.

Exemple 1:

#### ➤ **Création d'un nouveau projet avec « Project Wizard »**

Lancer logicielle MPLAB IDE à partir de l'icône qui se trouve sur votre bureau.

Cliquer sur Project >> Project Wizard ...

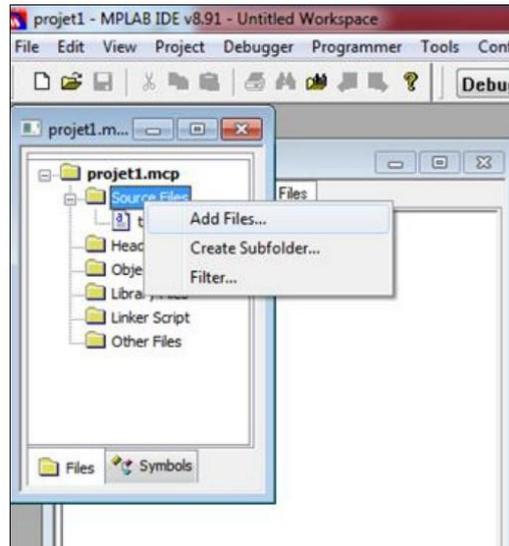
Cliquer sur Suivant, puis sélectionner PIC16F84A dans le menu déroulant Device.

Cliquer sur Suivant > puis sélectionner « Microchip MPASM Toolsuite » et «MPASM Assembler » afin de pouvoir programmer en Assembleur.

Cliquer sur Suivant > puis dans le champ « Project Name » saisir le nom du projet (ex projet\_tp1) , Cliquer sur Suivant > puis sur Terminer.

#### ➤ **Édition du programme**

Il faut ouvrir un fichier source (.asm) dans lequel on écrira le programme en langage assembleur : Bouton droit sur Source File, puis ajouter le fichier « TP1.asm »



Saisir le programme suivant et le sauvegarder, dans votre répertoire :

```
.....  
.....  
T1 EQU 0x30  
MOVLW 0xFF  
ADDLW B'01101010'  
MOVWF T1  
END  
.....  
.....
```

Ajouter des commentaires au programme pour montrer que vous avez bien compris les différentes instructions. Une ligne de commentaire commence par « ; »

➤ **Compilation**

Cliquer sur Project >> **Build All ...** ou sur l'icône de la barre de menu  ou encore **Ctrl+F10** pour compiler le projet complet.

Erreurs et warnings sont signalés dans la fenêtre Output ainsi que le résultat de la compilation **BUILD SUCCEEDED** ou **BUILD FAILED**.

NB: En double-cliquant sur une erreur ou un warning dans la fenêtre Output, vous serez amené directement à la ligne concernée dans votre programme. S'il n'y a pas des erreurs de compilation, on obtient le message « BUILD SUCCEEDED » et on aura la génération du fichier « TP1.hex ».

➤ **Simulation**

Choisissez l'outil de simulation MPLAB SIM comme "debugger". Exécutez le programme pas à pas et observez l'évolution des registres PCL, Status (Onglet View -> Special Function Register) Suivre les registres et les différentes variables (Onglet View -> Watch) : Ajouter, par le menu contextuel de la barre du nom des colonnes (clic droit) une colonne « Hex » afin d'obtenir les valeurs en hexadécimal du contenu des registres.

## 2. Création d'un nouveau projet MPLAB : pour faire clignoter une LED

- ✓ Ouvrez MPLAB X IDE.
- ✓ Créez un nouveau projet, en sélectionnant le PIC16F84 comme microcontrôleur.
- ✓ Écrivez votre premier programme simple : un clignotement de LED par exemple

## 3. Simulation dans Proteus :

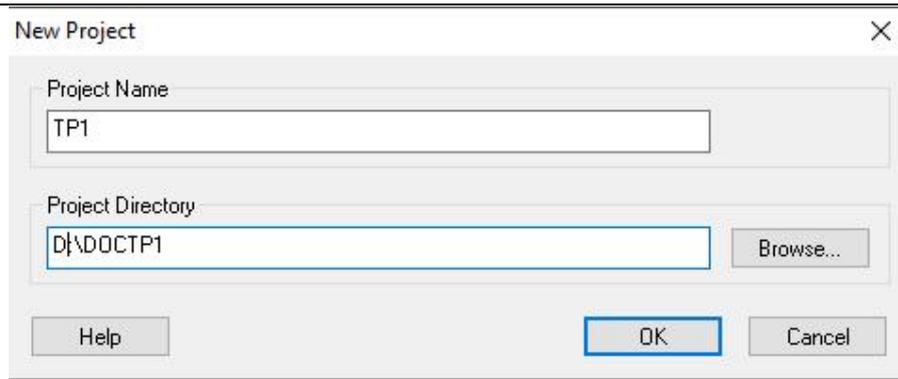
Créez un nouveau schéma dans Proteus.

- ✓ Placez les composants nécessaires : PIC16F84, résistances, LED, etc.
- ✓ Connectez les composants conformément à votre schéma.
- ✓ Importez le code compilé depuis MPLAB dans Proteus.
- ✓ Simulez votre circuit et observez le comportement de la LED.

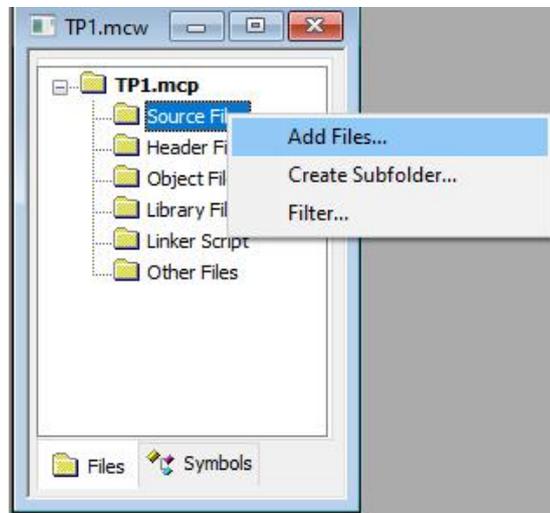
### ✧ Exemple d'application simple :

Pour réaliser un clignotant LED simple. un programme est écrit en assembleur pour faire allumer et éteindre une LED connectée à une broche du PIC à intervalles réguliers.

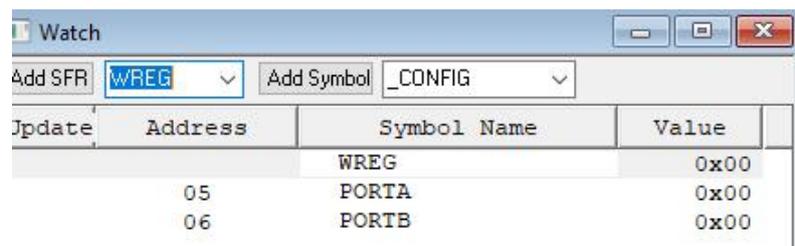
Cliquer sur **Project -> New**



**Avec le bouton droit ajouter le fichier Programme1.asm au projet TP1**

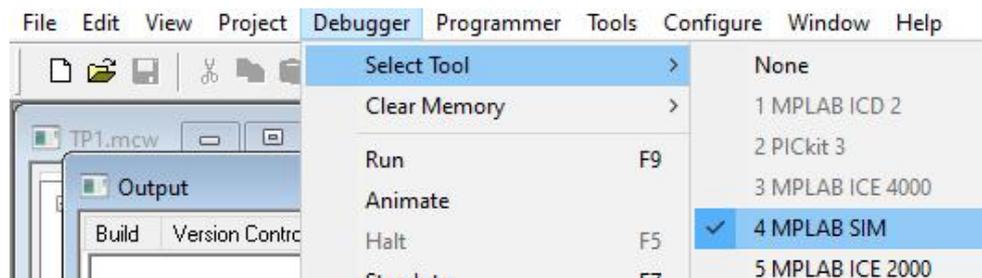


- ✓ Cliquer sur Configure -> Select Device et sélectionner PIC16F84
- ✓ Compiler le projet Project -> Build All
- ✓ Vérifier la création d'un nouveau fichier programme1.hex ouvrir ce fichier avec l'éditeur de texte BLOC-Notes ou NotePad , conclure.
- ✓ Visualiser les registres avec **View -> Watch**

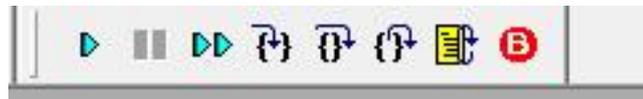


Update	Address	Symbol Name	Value
		WREG	0x00
	05	PORTA	0x00
	06	PORTB	0x00

## Sélectionner MPLAB SIM



Avec l'exécution pas à pas vérifier les valeurs des registre de la fenêtre Watch.



## AVEC PROTEUS OUVRIR LE PROJET Carte\_TP1

- Charger le programme1.hex dans le PIC
- Vérifier le fonctionnement du programme en lançant la simulation.

**Le schéma de la carte :**

