

TP N° **1**

Prise en main de l'environnement de développement pour le PIC1684A

1.1 Objectifs

- Se familiariser avec l'environnement du logiciel Proteus ISIS.
- Utiliser le compilateur MikroC PRO pour programmer un PIC.
- Comprendre l'architecture matérielle et logicielle du microcontrôleur PIC16F84A.

1.2 Introduction

La simulation des systèmes embarqués exige l'utilisation de deux logiciels, le premier est un logiciel dédié à la simulation des circuits électronique et le deuxième pour la programmation des PICs, dans ce contexte nous allons utiliser durons ces Travaux Pratiques les softwares : Proteus ISIS et MikroC PRO.

1.2.1 Proteus ISIS

ISIS Proteus est un logiciel de simulation électronique édité par la société « Lab Center Electronics ». Il intègre des simulateurs analogique, logique et mixte pour tout type de circuit électronique. Grâce à des modules additionnels, ISIS est également capable de simuler le comportement d'un microcontrôleur (PIC, Atmel, Intel 8051, ARM, Motorola HC11...) et son interaction avec les composants qui l'entourent, c'est cette dernière fonctionnalité d'ISIS qui nous intéresse dans ce TP. La figure 1.1 illustre la fenêtre principale du logiciel ISIS.

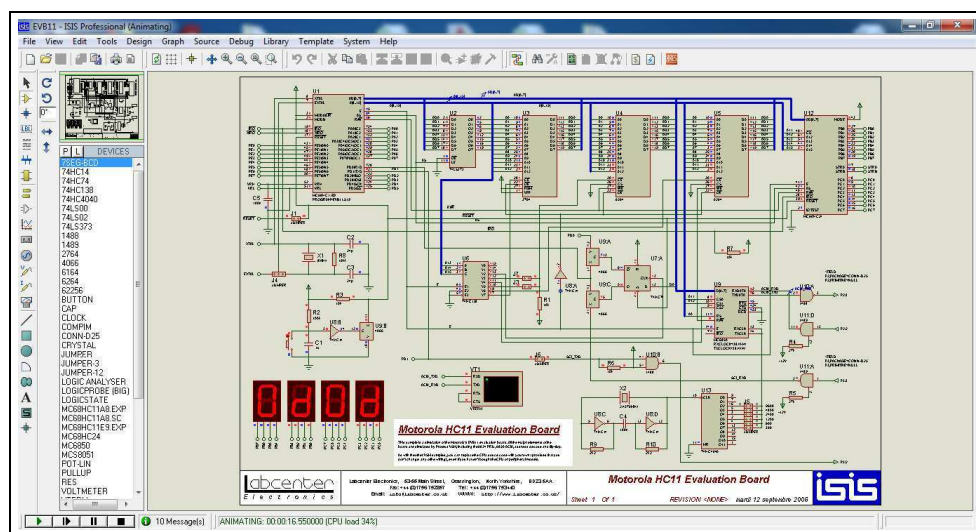


FIGURE 1.1: Fenêtre Proteus ISIS

1.2.2 MikroC PRO

Le « mikroC PRO » est un compilateur pour PIC conçu par la société « Mikroelektronika ». Il comporte plusieurs outils intégrés (mode simulateur, terminal de communication, gestionnaire 7 segments,...). Il a une capacité à pouvoir gérer la plupart des périphériques rencontrés dans l'industrie (BusI2C, 1Wire, SPI, RS485, Bus CAN, cartes compact Flash, signaux PWM, afficheurs LCD et 7 segments...), de ce fait il est un outil de développement incontournable et puissant. Il contient un large ensemble de bibliothèques de matériel, de composant et une documentation complète. La figure 1.2 illustre la fenêtre principale du logiciel MikroC PRO.

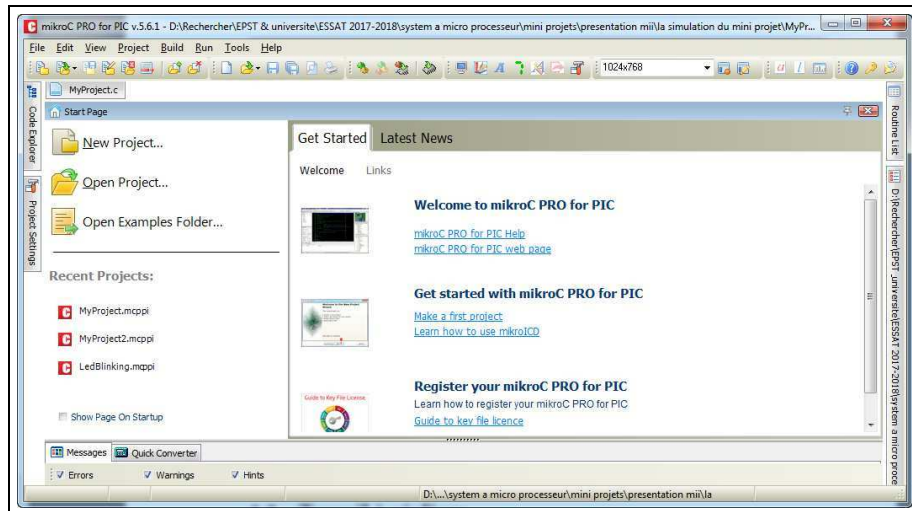


FIGURE 1.2: Fenêtre Mikro C PRO

1.3 Travail à réaliser

Dans ce qui suit, nous allons réaliser quelques applications basiques à base du microcontrôleur PIC16F84A, ces exemples vont nous permettre de bien se familiariser avec l'environnement de simulation ISIS et le compilateur mikroC PRO.

1.3.1 Application N° 1 : Allumer une LED

Le but de cette première application est d'allumer une LED de couleur rouge connectée à la broche RB0 du port B du PIC16F84A, selon les étapes suivantes :

1. Lancer ISIS et le Compilateur mikroC PRO.
2. Réaliser le circuit de test (voir figure 1.3) sous ISIS :
3. Ouvrir un nouveau projet sous mikroC PRO, écrire le programme ci-dessous :
4. Compiler votre programme, et générer le fichier .hex correspondant à votre programme.
5. Dans ISIS, on double clique sur le microcontrôleur et on injecte le fichier .hex généré dans l'étape précédente.

6. Par la suite lancer la simulation sous ISIS (le bouton play du panneau de contrôle de l'animation), la LED sera normalement allumée.

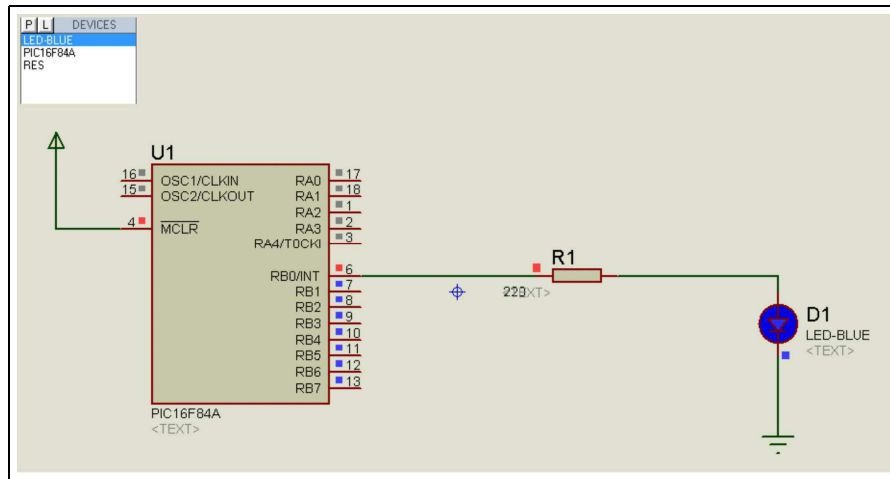


FIGURE 1.3: Schéma électronique « commande une LED par le PIC16F84A »

```
Void main() {  
    trisb = 0b11111110; // configuration du pin RB0 en sortie (1 : Entrée, 0 : Sortie)  
    while(1) // boucle infinie  
    {  
        {portb = 0b00000001;} // la led connectée au pin RB0 sera allumée  
    }  
}
```

Dans les applications qui suivent, nous allons refaire les mêmes étapes de 1 à 6, afin de réussir la simulation.

1.3.2 Application N°2 : Clignotement d'une LED

Dans cette application, nous allons transformer notre programme pour faire clignoter la LED. La LED s'allume pendant une seconde et s'éteint pendant la seconde suivante. Le même circuit de la première application sera utilisé. Dans ce programme, nous utiliserons la fonction `delay_ms()` permettant d'ajouter un délai en millisecondes.

Le programme correspondant à cette application est le suivant :

```
void main() {
trisb= 0b11111110; // configuration du pinRB0 en sortie (1 : Entrée, 0 : Sortie)
while(1)           // boucle infinie
{ portb = 0b00000001; // allumer la led connectée au pin RB0
delay_ms(1000);      // pause d'une seconde
portb = 0b00000000; // éteindre la led connectée au pin RB0
delay_ms(1000);      // pause d'une seconde
}}
```

1.3.3 Application N°3 : Clignotement de plusieurs LED

Le but de cette application est de faire clignoter d'une manière alternée 4 LED de couleur rouge connectées au port B du PIC16F84A correspondant aux broches allant de RB0 jusqu'à RB3. Les LED s'allument pendant une demi-seconde et s'éteignent pendant la demi-seconde suivante.

1. Réaliser le circuit de test (figure 1.4) sous ISIS.
2. Ecrire le programme illustré par la figure 1.5 sous mikroC PRO.

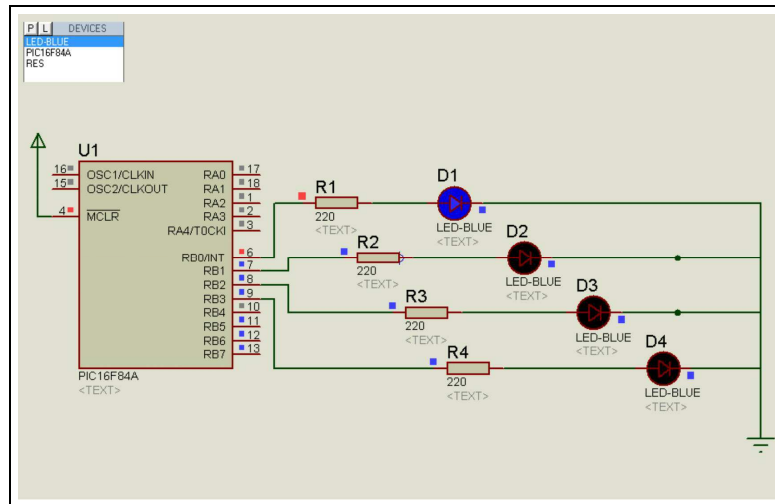


FIGURE 1.4: Schéma électronique « Clignotement de plusieurs LED »

```
void main() {
    trisb = 0x10;          // configuration des pins RB0 jusqu'à RB3 en sortie
    while(1)              // boucle infinie
    { portb = 0b00000000;
      delay_ms(500); portb = 0b00000001;
      delay_ms(500); portb = 0b00000010;
      delay_ms(500); portb = 0b00000100;
      delay_ms(500); portb = 0b00001000;
      delay_ms(500); }
}
```

FIGURE 1.5: Code source pour faire clignoter d'une manière alternée 4 LEDs

Dans l'exemple précédent on a allumé les LED en agissant sur le port en entier, cependant, nous pouvons accéder individuellement à chaque bit en utilisant les identifiants F0, ..., F7. Par exemple pour allumer la première LED, on écrit `Portb.F0 = 1`.

1. Ecrire un autre programme illustré par la figure 1.6 qui permet de faire clignoter les LED paires pendant une seconde et les LED impaires pendant la seconde suivante.

1.3.4 Application N°4 : Un seul feu tricolore

Le but de cette application est de simuler la commande d'un seul feu tricolore à l'aide du PIC16F84A. Le fonctionnement du feu tricolore est détaillé comme suit :

```
void main() {  
  trisb= 0x10; // configuration des pins RB0 jusqu'à RB3 en sortie  
  while(1) // boucle infinie  
  { portb = 0b00000000;  
    delay_ms(500); portb.f0 = 1;  
    delay_ms(500); portb.f0 = 0; portb.f1 = 1;  
    delay_ms(500); portb.f1 = 0; portb.f2 = 1;  
    delay_ms(500); portb.f2 = 0; portb.f3 = 1;  
    delay_ms(500); }  
}
```

FIGURE 1.6: Code source pour faire clignoter les LEDs paires et impaires d'une manière alternée

- Allumer le feu vert pendant 30 secondes.
- Allumer le feu orange pendant 04 secondes.
- Allumer le feu rouge pendant 20 secondes.
- Reprendre le cycle du début.

1.3.5 Application N°5 : Deux feux tricolores synchronisés

Dans cette application, vous allez simuler un cas réel des feux de croisement composés de deux feux tricolores synchronisés (voir figure 1.7). Les deux feux permettront l'régulation de la circulation d'un carrefour à deux voies (route principale et route secondaire).



FIGURE 1.7: Deux feux tricolores synchronisés.

TP N°1. Prise en main de l'environnement de développement pour le PIC1684A

Le fonctionnement des deux feux tricolores est détaillé comme suit :

- Allumer le feu vert pendant 40 secondes sur la route principale et le rouge sur la route secondaire.
- Allumer le feu orange pendant 05 secondes sur la route principale et toujours le rouge sur la route secondaire.
- Allumer le feu rouge pendant 25 secondes sur la route principale et le feu vert sur la route secondaire.
- Allumer le feu orange pendant 04 secondes sur la route secondaire et toujours le rouge sur la route principale.
- Reprendre le cycle du début