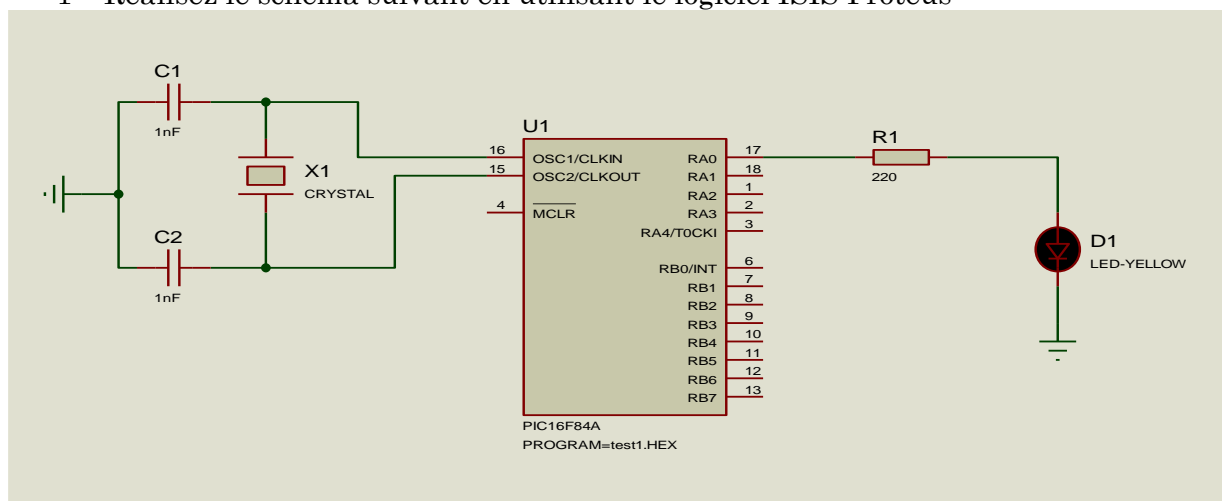


TP N° 1

Programmation assembleur PIC 16F84a

I – Schéma ISIS

- 1- Réalisez le schéma suivant en utilisant le logiciel ISIS Proteus



- 2- Créer un répertoire et enregistrer le schéma suivant dans le répertoire.
- 3- ouvrir un fichier .txt et écrire le code suivant :

```
LIST      p=16F84A  ; Processeur 16F84A
#include <p16F84A.inc> ; Définition des registre internes du 16F84
__CONFIG _CP_OFF & _WDT_OFF & _HS_OSC ; Option de programmation
```

```
bsf STATUS , RP0 ; sélection de la bank 1
movlw H'00'
movwf TRISA ; configurer port A en sortie
```

```
LED bcf STATUS, RP0 ; sélection de la bank 0
movlw b'00001'
movwf PORTA ; envoyer 5v dans RA0
goto LED ; rester dans la boucle
END
```

- 4- Enregistrer le fichier sous l'extention .asm.
- 5- Dans le menu source de ISIS ajouter votre fichier .asm.
- 6- Exécuter le fichier sous ISIS.

2- LED Clignotante.

```
;;;;;; *** Configuration
LIST    p=16F84A    ; Processeur 16F84A
#include <p16F84A.inc> ; Définition des registre internes du 16F84
_CONFIG _CP_OFF & _WDT_OFF & _HS_OSC ; Option de programmation

;;;;;; *** Définitions
RA2 EQU 0x02    ; PortA, sortie RA2

;;;;;; *** Variables
CBLOCK 0x0C
    vTempo1, vTempo2;
ENDC

;;;;;; *** Vecteurs
ORG    H'0000'    ; Déclaration du Reset
GOTO   Start

; * Tempo de clignotement qui utilise deux variables pour obtenir une tempo longue

Tempo  MOVLW  0xFF    ; Charge W avec 0xFF
        MOVWF  vTempo2    ; pour charge la vTempo2
_Tempo2 MOVWF  vTempo1    ; et la vTempo1
_Tempo1 DECFSZ vTempo1, f    ; Décrémente vTempo et met le résultat dans vTempo
        GOTO   _Tempo1    ; puis saute l'instruction suivante si 0.
        DECFSZ vTempo2, f
        GOTO   _Tempo2
        RETURN

; * Programme principal

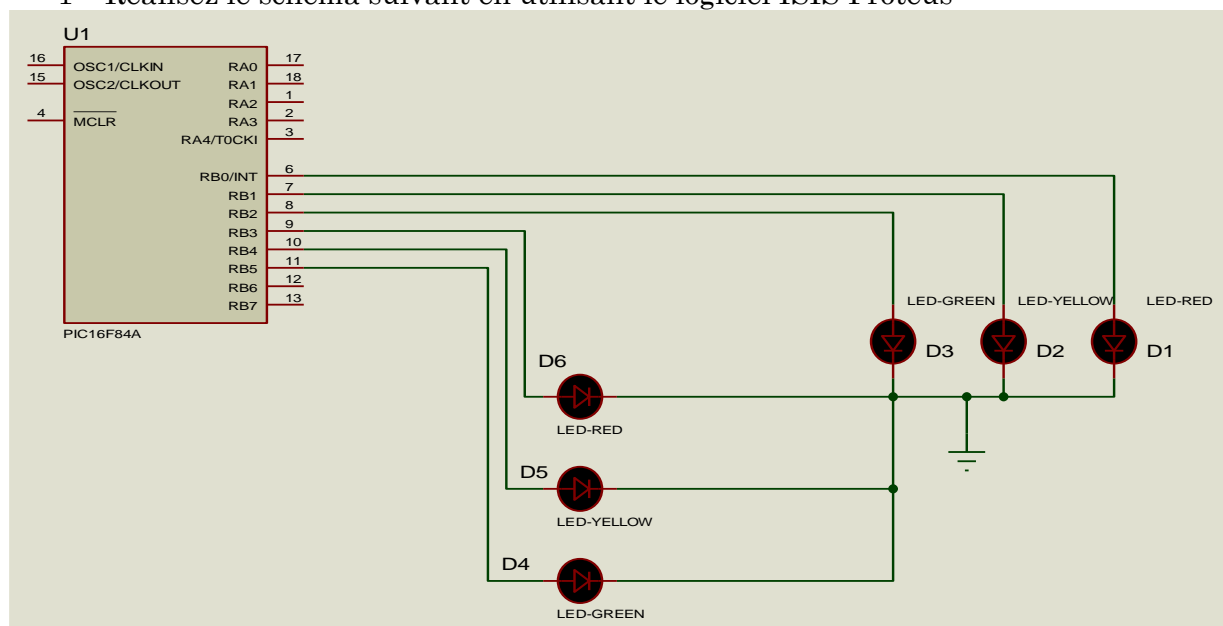
Start  BSF STATUS, RP0 ; Acces Bank1
        MOVLW  0xFB    ; TRISA.b2 à 0 pour RA2 en sortie
        MOVWF  TRISA
        BCF   STATUS, RP0 ; Acces Bank0
        BSF   PORTA, RA2 ; Allume la Led
        CALL  Tempo
        BCF   PORTA, RA2 ; Eteint la Led
        CALL  Tempo
        GOTO  Start    ; Reboucle au départ
        END
```

TP N° 2

Programmation Micro C PIC 16F84a

I – Feu Rouge

1- Réalisez le schéma suivant en utilisant le logiciel ISIS Proteus



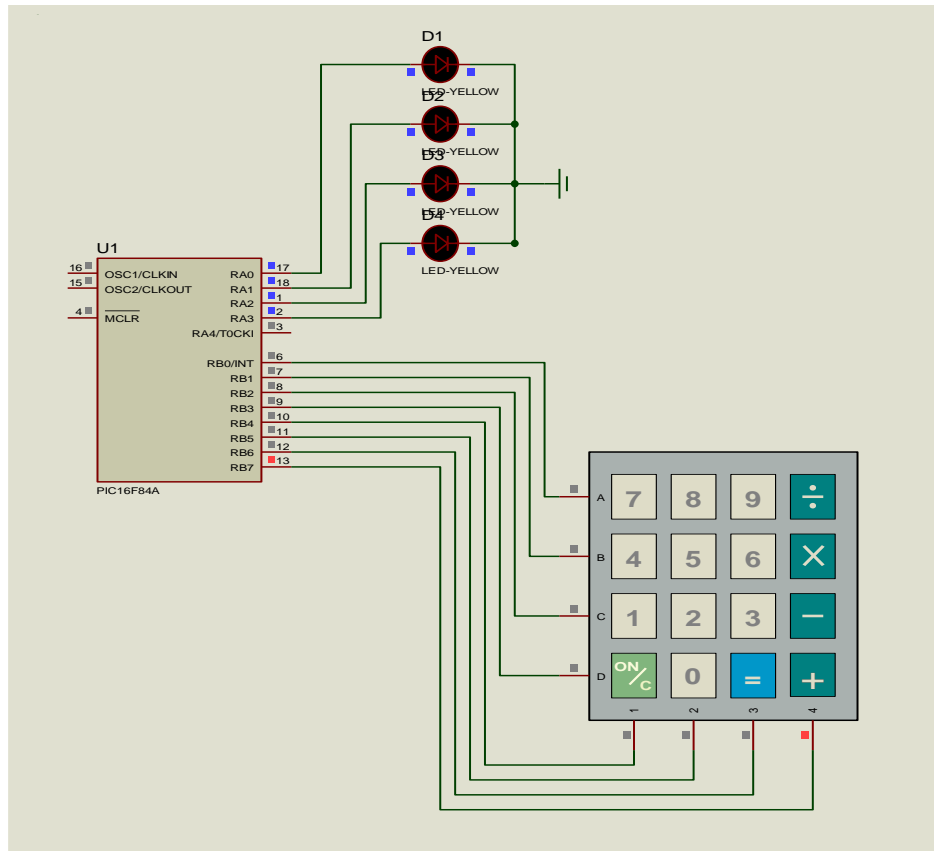
- 2- Créer un répertoire et enregistrer le schéma suivant dans le répertoire.
- 3- Ouvrir MikroC Pro
- 4- Choisissez le type de Microcontrôleur (16F84a) et réglez la fréquence d'horloge sur 1Mhz.
- 5- Configurer le port B en sortie par la commande `TRISB=0x00 ;`
- 6- Mettre une tension 0V dans la sortie du port B.
- 7- Faire une boucle for infinie.


```
for (;;)
{
}

```
- 8- Allumer la LED rouge D1 et verte D4 par l'instruction `PORTB=0b00100001`
- 9- Mettre un temporisateur de 5s utilisant l'instruction `delay_ms(5000) ;`
- 10- Allumez la LED rouge D1 et jaune D5 avec un temps de 0.5 s.
- 11- Allumez la LED rouge D1 et éteindre la LED jaune D5 avec un temps de 0.5 s.
- 12- Allumer la LED rouge D6 et verte D3
- 13- Allumez la LED rouge D6 et jaune D2 avec un temps de 0.5 s.
- 14- Allumez la LED rouge D1 et éteindre la LED jaune D2 avec un temps de 0.5 s.
- 15- Reboucler....

II – Clavier 4x4

- 1- Réalisez le schéma suivant en utilisant le logiciel ISIS Proteus
Avec le clavier KEYPAD SMALL-CALC



- 2- Réaliser le programme suivant utilisant le logiciel MikroC

```
char Q;  
char Keypadport at PORTB;  
  
void main()  
{  
    TRISA=0x00;  
    PORTA=0;  
    for(;;)  
    {  
        Q=Keypad_key_click();  
        switch (Q)  
        {  
            case 1: PORTA=1;    break;  
            case 2: PORTA=2;    break;  
            case 3: PORTA=3;    break;  
            case 4: PORTA=4;    break;  
            case 5: PORTA=5;    break;  
            case 6: PORTA=6;    break;  
            case 7: PORTA=7;    break;  
            case 8: PORTA=8;    break;  
            case 9: PORTA=9;    break;  
            case 10:PORTA=10;   break;  
            ...  
        }  
    }  
}
```