

## TP1+TD1 Solution

### Exercice 1

Avant de résoudre cet exercice laissez moi définir la méthode de travail dans ce module.

### Méthode de résolution

Cette section concerne la méthode à suivre pour résoudre n'importe quel exercice des TD du Module Microcontrôleur. Elle peut être utilisée pour la conception des projet avancés


Pour résoudre un exercice quelconque il y'a deux parties :

- Conception du circuit électronique pour réaliser la simulation avec ISIS Proteus
- Ecrire le programme (code source° en langage MiKroc pour rendre le microcontrôleur intelligent.

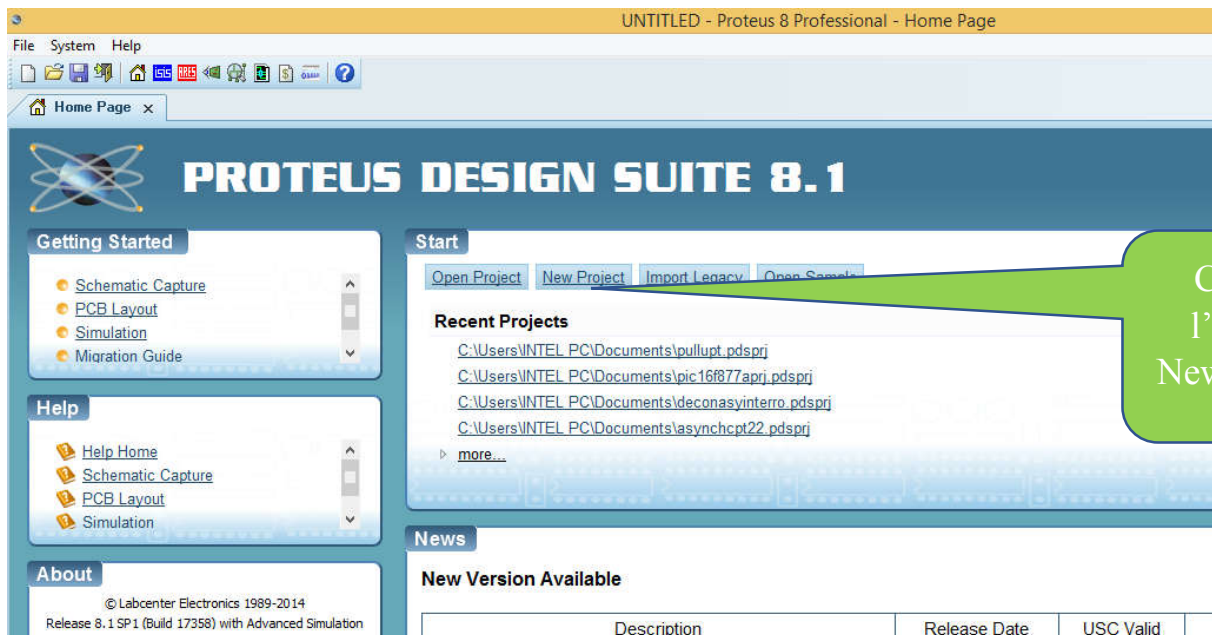
Dans les parties suivantes nous expliquons en détail les parties précédentes avec des capture d'écran afin de rendre la conception par microcontrôleur facile.

### Partie 1 : Conception du circuit électrique avec ISIS

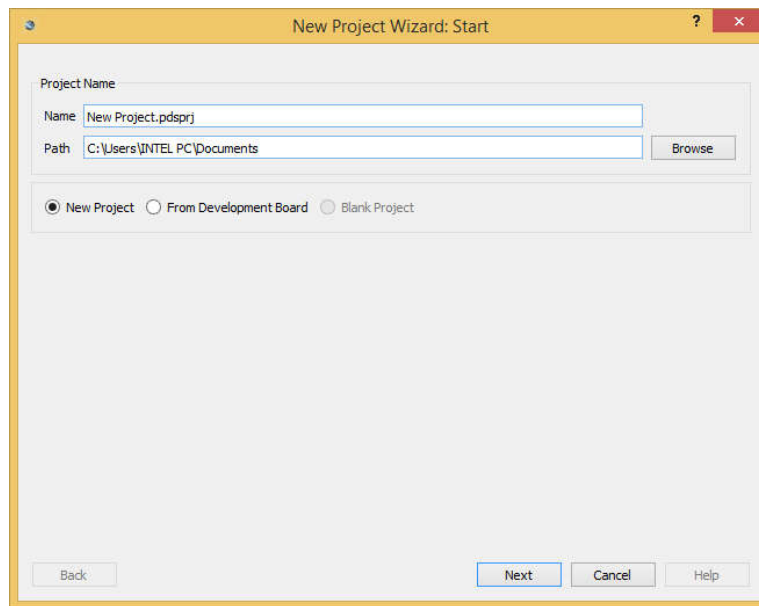


Double cliquer sur l'icone  pour lancer le software Proteus

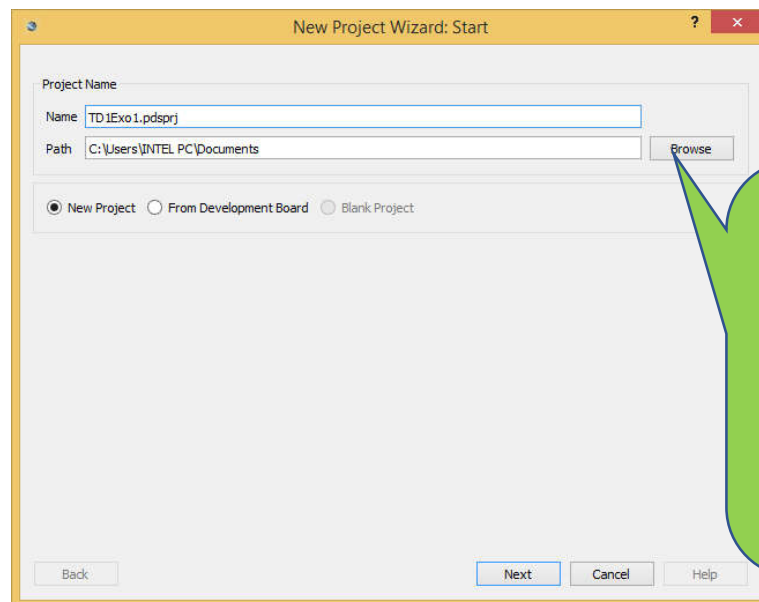
Une fois la fenêtre suivante est ouverte



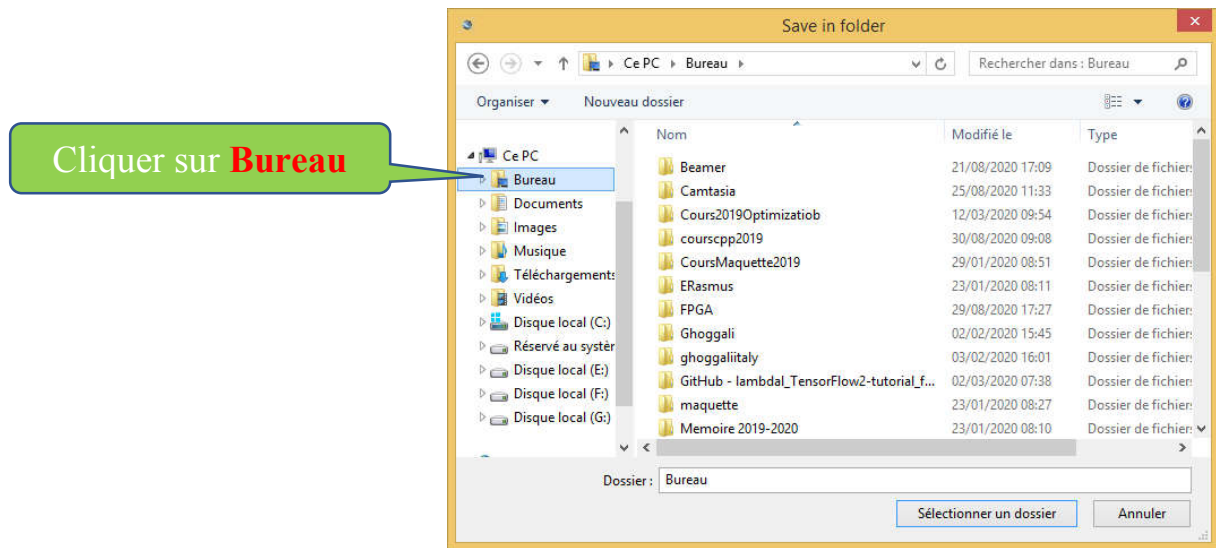
Une Nouvelle boîte de dialogue s'ouvre pour nous permettre de choisir le Nom, Le répertoire dans lequel nous allons sauvegarder les fichiers de notre projet :



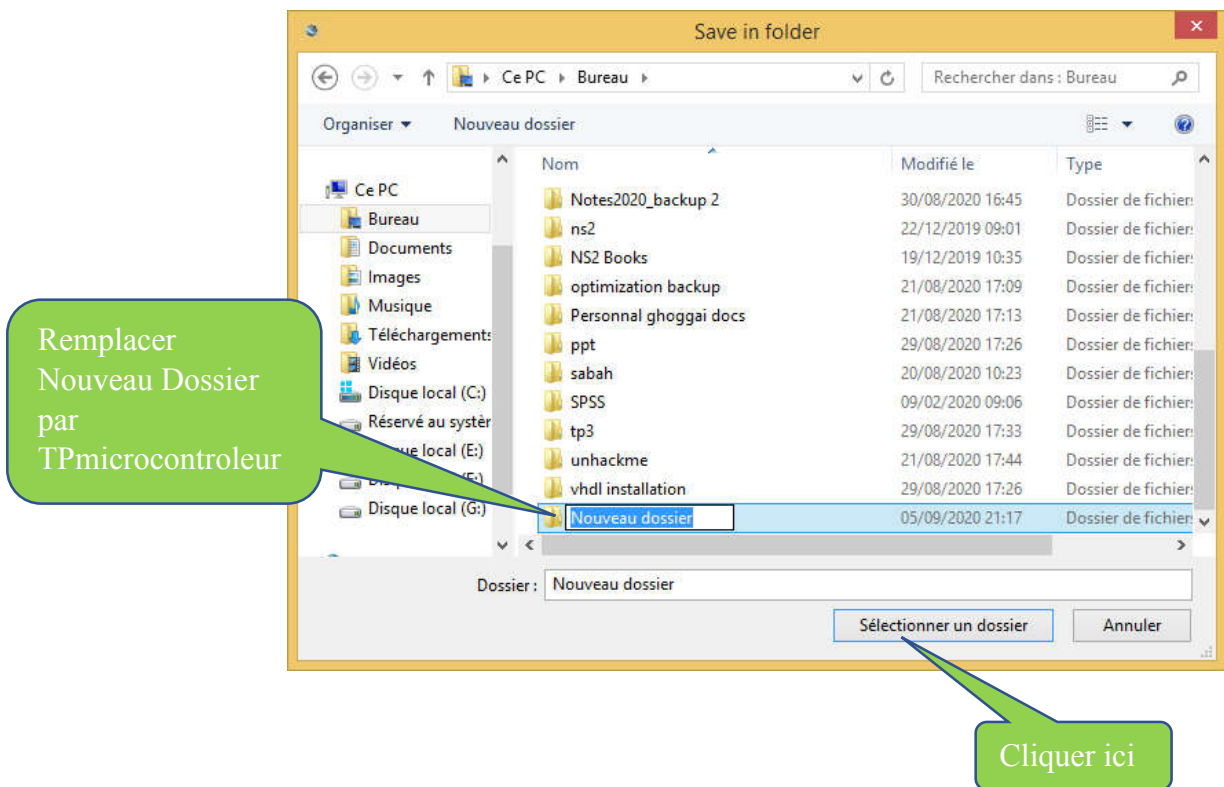
Remplacer le Champ Name Par **TDExo1.pdsprj** comme indiqué sur la figure suivante



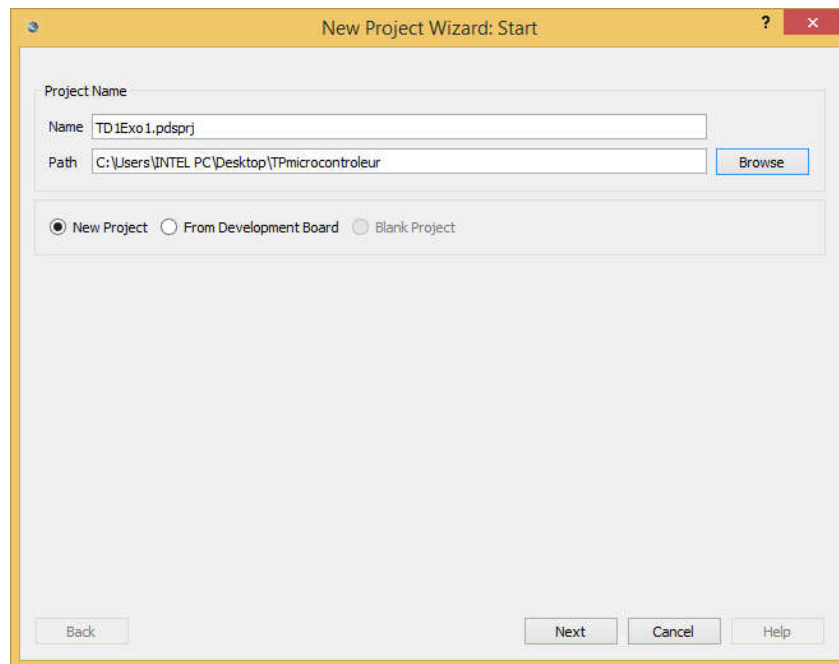
La boîte de dialogue suivante s’ouvre, cliquer sur **Bureau** comme indiqué sur la figure



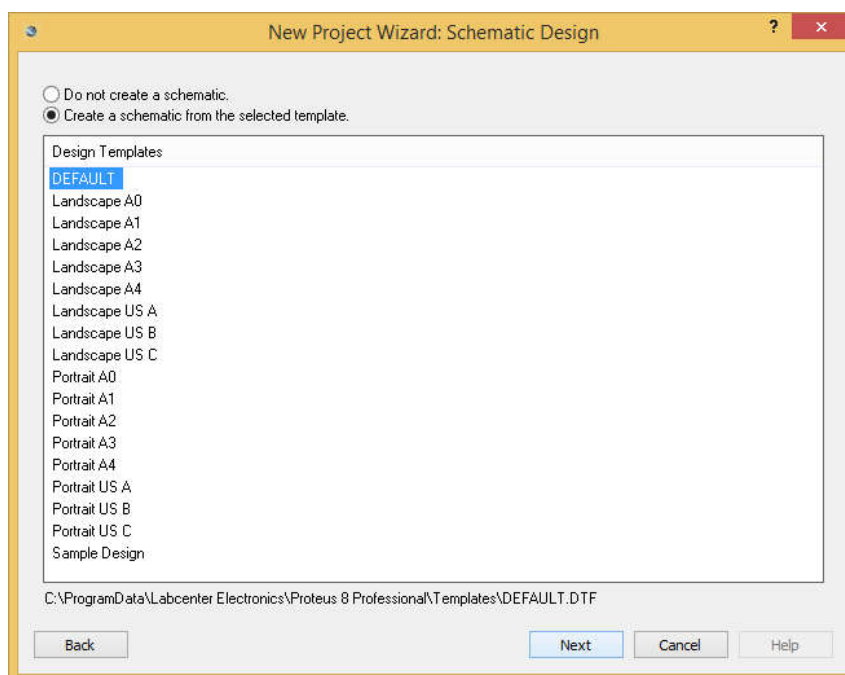
Pour obtenir la figure suivante



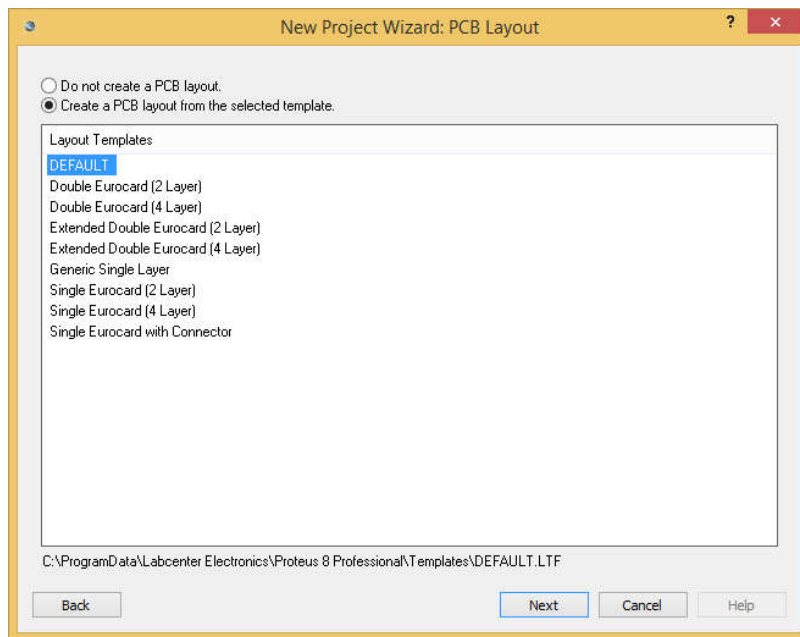
Une fois ces opérations sont achevées la boîte de dialogue suivante s’affiche



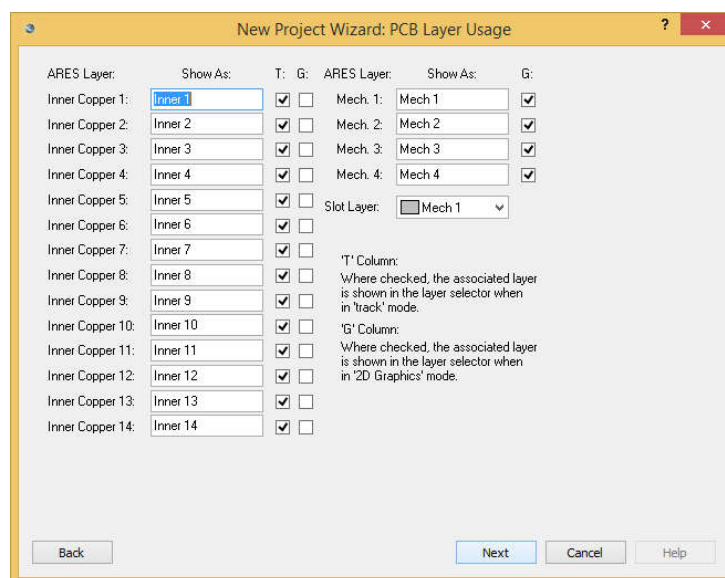
Cliquer sur Next



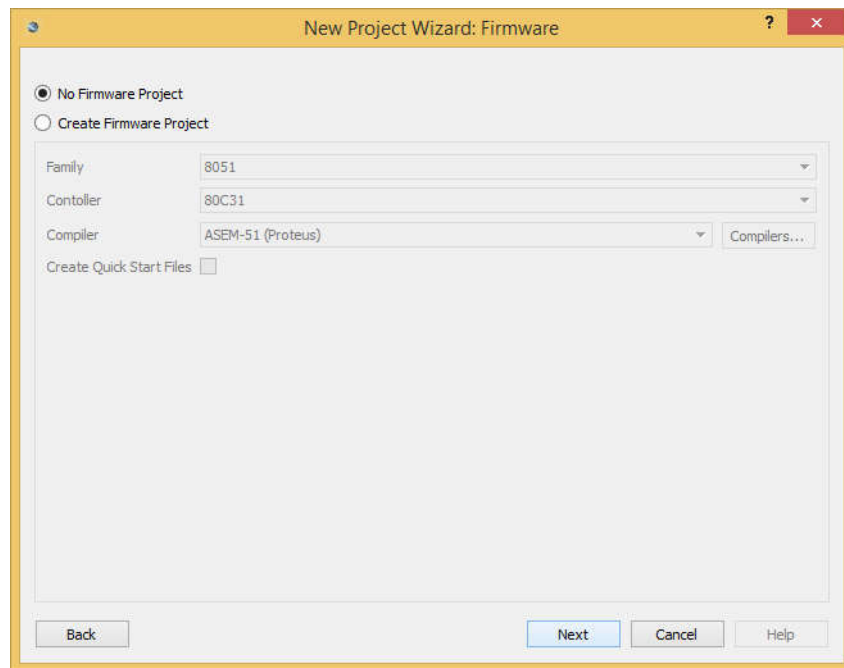
Cliquer sur **Create a schematic from the selected template** ensuite **Next**



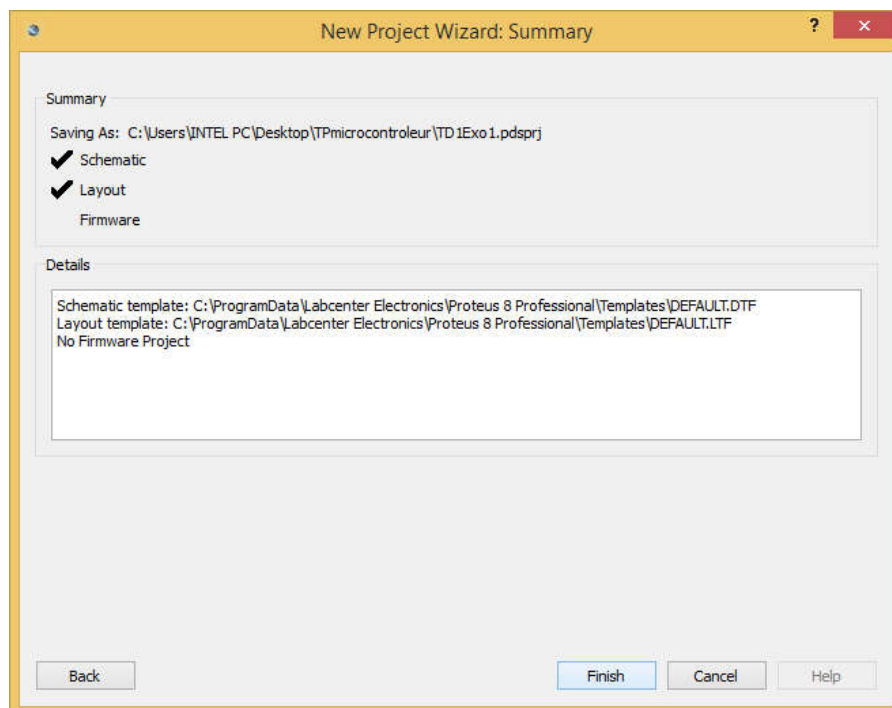
Cliquer sur **Create a PCB from Layout from the selected template** ensuite **Next**



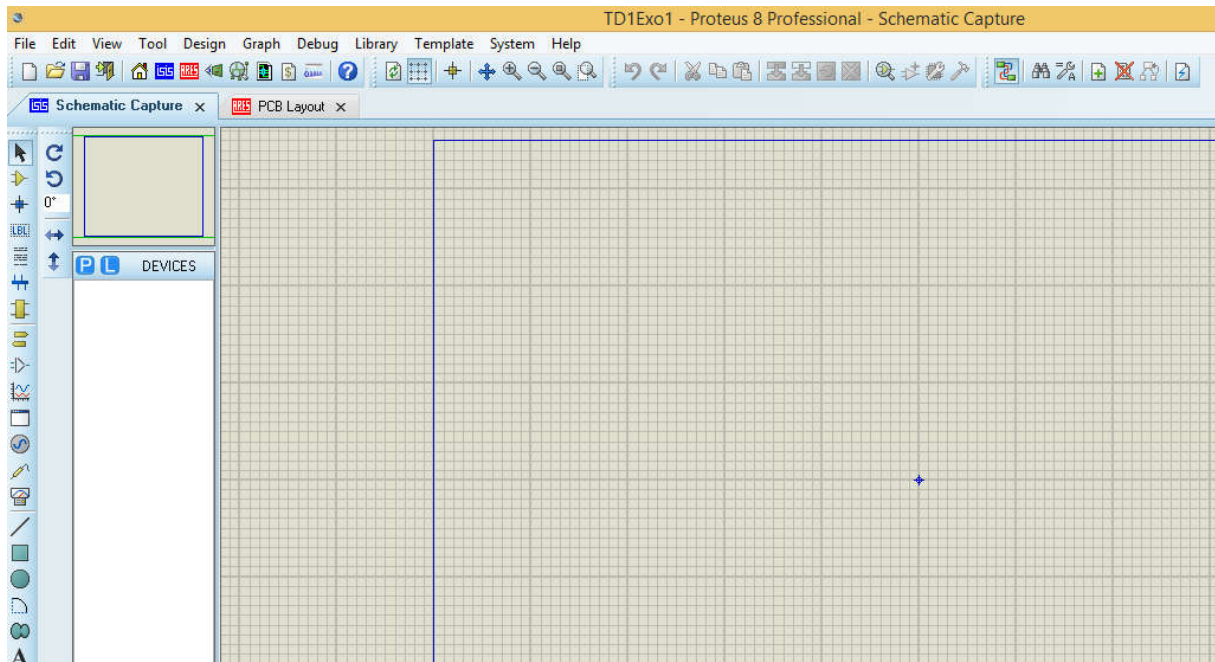
Cliquer sur **Next**



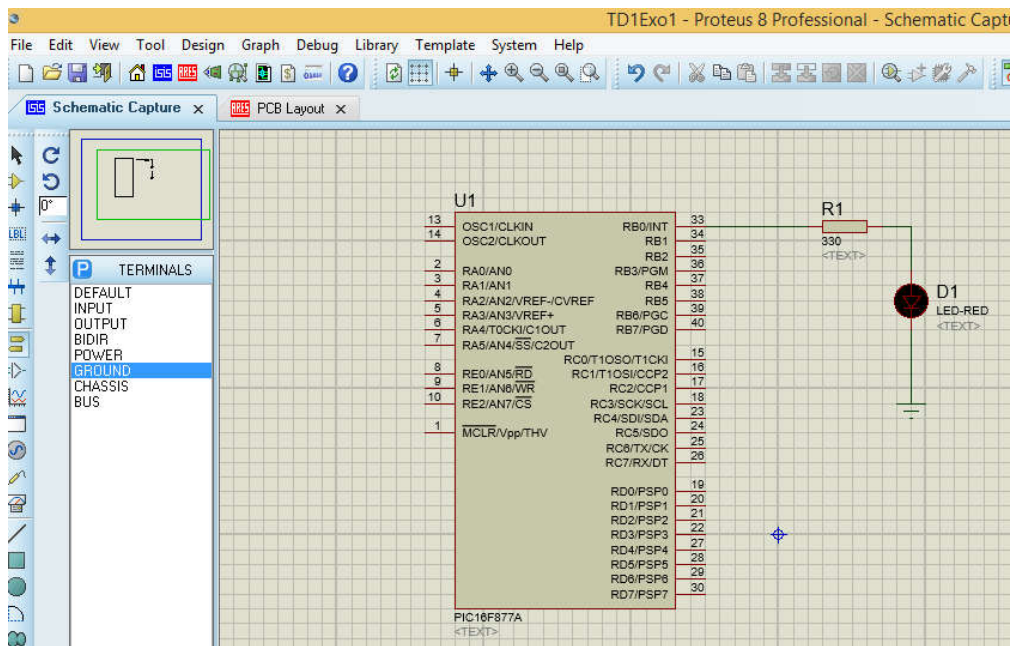
Cliquer sur Next



Cliquer sur Finish pour obtenir la figure suivante



Saisir le schema de l'exercice 1 du TD1 Comme indique sur la figure suivante

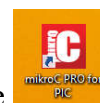


**Note :** N'oublier pas de Sauvegarder votre travail pendant la conception de temps en temps.

Une fois arriver à ce point, nous allons maintenant développer le programme C qui permet de répondre au besoin de l'exercice.

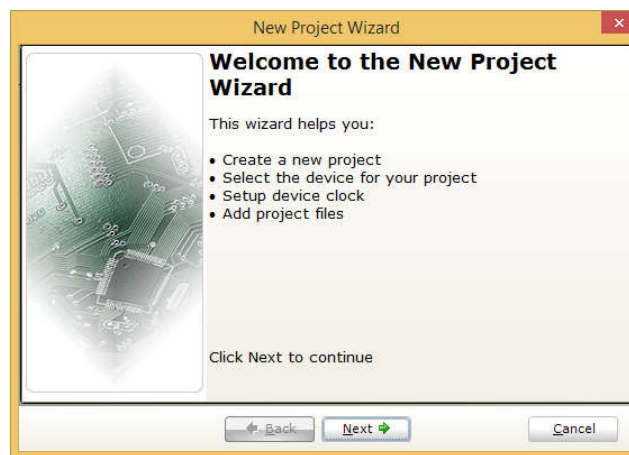
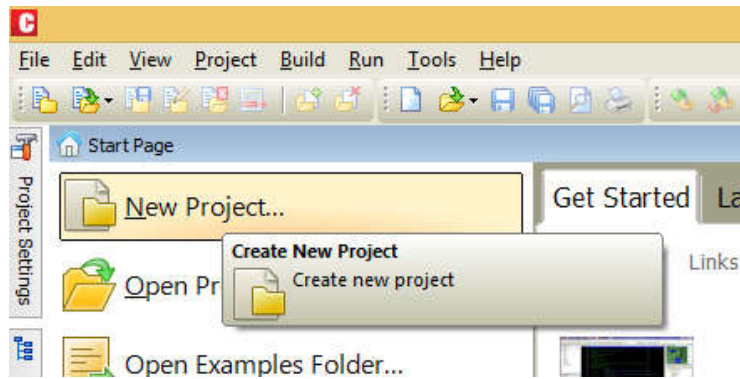
Deuxième partie : code source C avec Mikroc pro

Réduire la fenêtre de Proteus ISIS ensuite double cliquer sur l'icône suivante

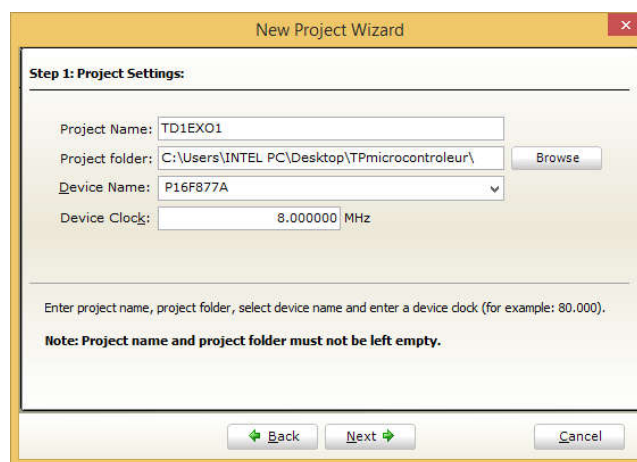


Créer le fichier projet comme suit :



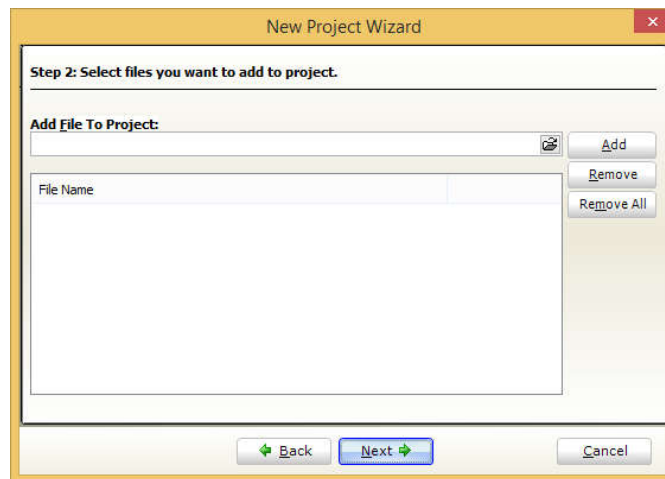


Cliquer sur Next

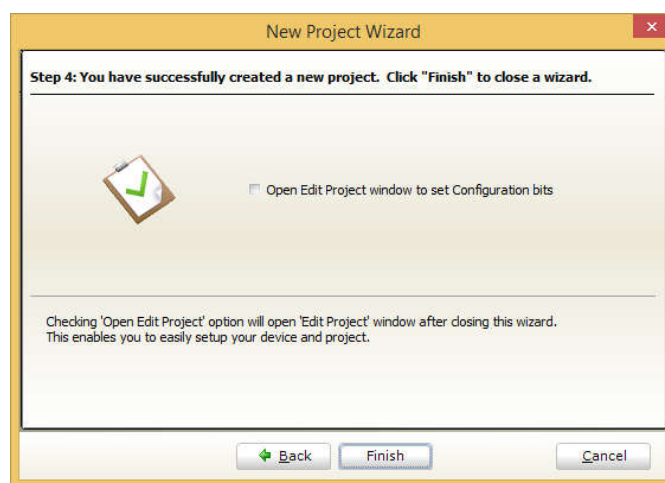


Cliquer sur Next





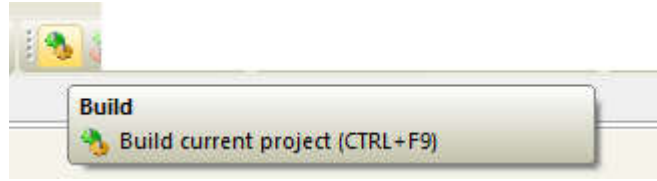
Cliquer sur Next 2 fois ensuite finish



Puis saisir le code C comme sur indiquer ci-dessous.

```
void main() {
    TRISB=0; //les pins du PORTB sont configurer comme sorties
    PORTB=0; //Mettre la valeur 00000000 dans les pins du PORTB
    while (1) //Boucle infinie
    {
        PORTB.RB0=0; //LED connecter au pin RB0 est etteinte
        delay_ms(1000); //Atternder un delqis de 1s=1000milliseconde
        PORTB.RB0=1; //LED Maintenant est allumée
        delay_ms(1000); //Atternder un delqis de 1s=1000milliseconde
    }
}
```

Sauvegarder et appuyer sur l'icône suivante

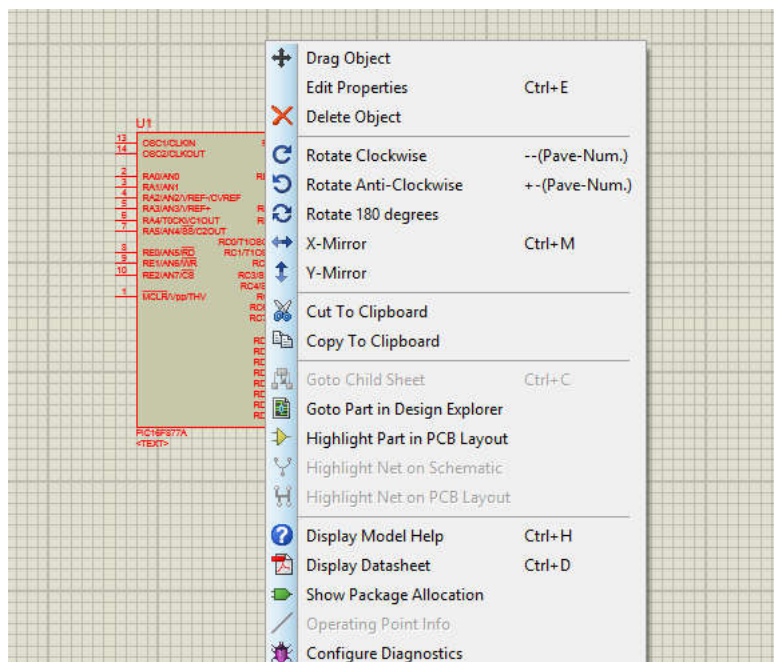


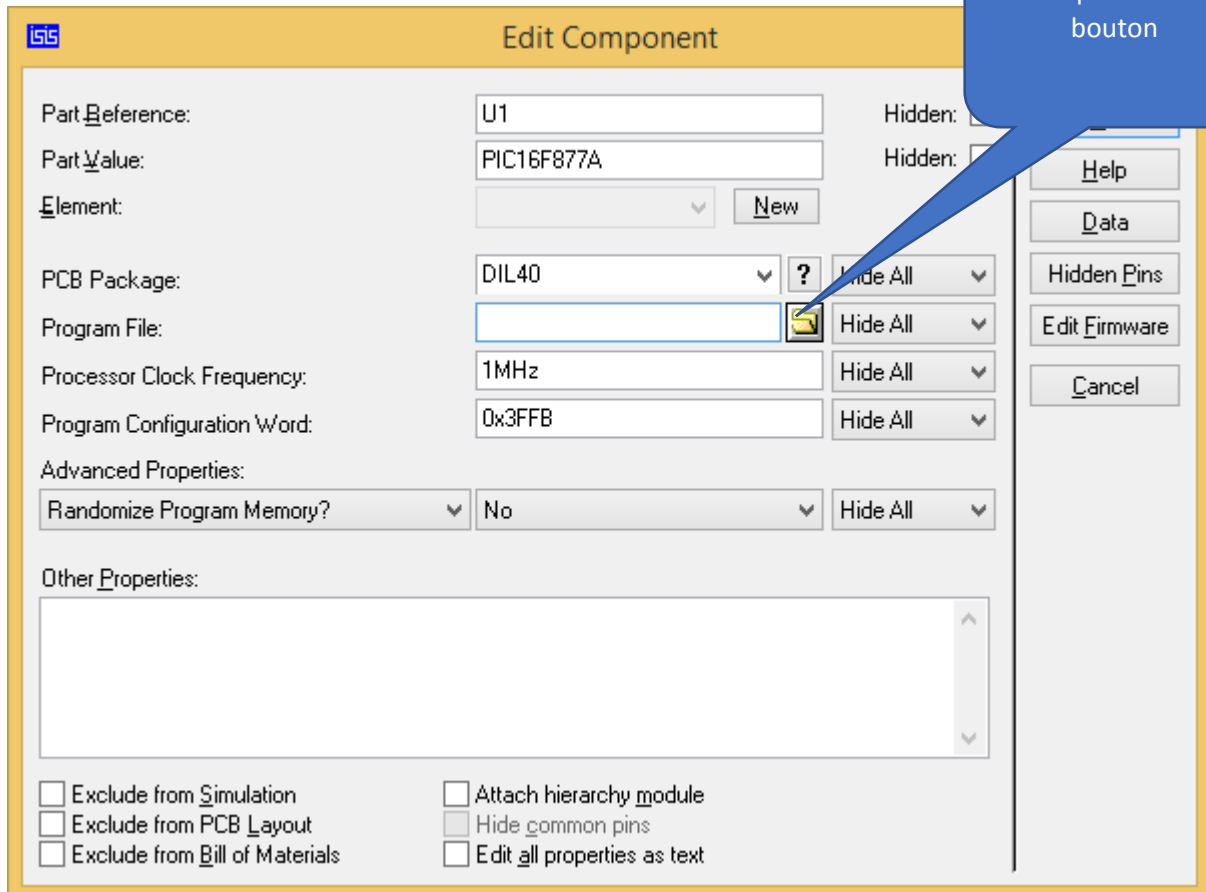
pour compiler le code C. Par le mot compiler on veut dire transformer le code C en Code Hex. Si votre programme ne contient aucune erreur la compilation se terminera par succes comme sur la figure suivante

Line	Message No.	Message Text
0	1144	Used RAM (bytes): 3 (1%) Free RAM (bytes): 349 (99%)
0	1144	Used ROM (program words): 56 (1%) Free ROM (program words): 8136 (99%)
0	125	Project Linked Successfully
0	128	Linked in 79 ms
0	129	Project 'TD1EXO1.mcppi' completed: 188 ms
0	103	Finished successfully: 05 sept. 2020, 21:55:32

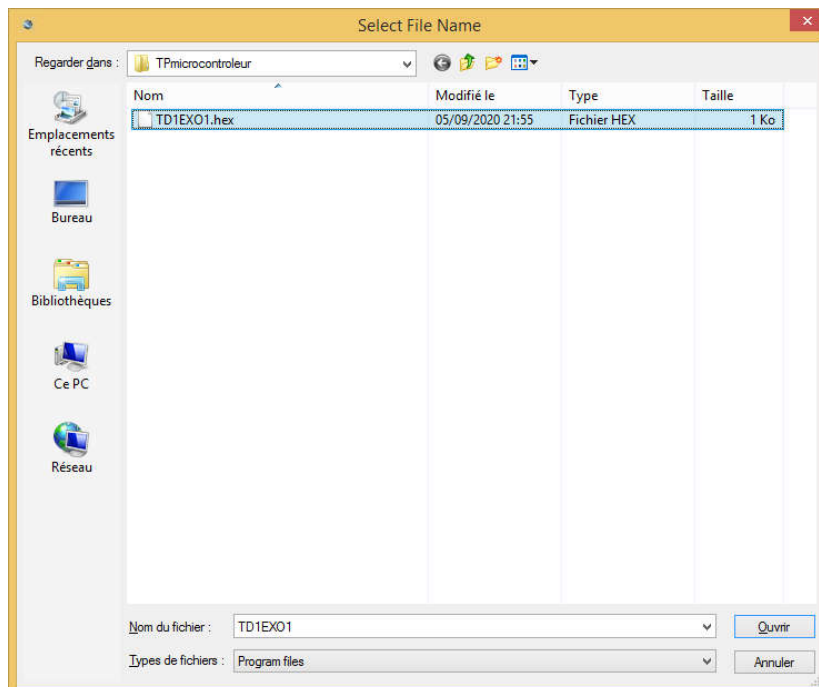
Revenons maintenant au Poteus ISIS afin d'intégrer le fichier .hex dans la mémoire flash du microcontrôleur et tester notre circuit

Cliquer avec le bouton droit de la souris ensuite choisir Edit Properties





Remplacer Processor Clock Frequency par 8MHz ensuite cliquer sur le bouton jaune.



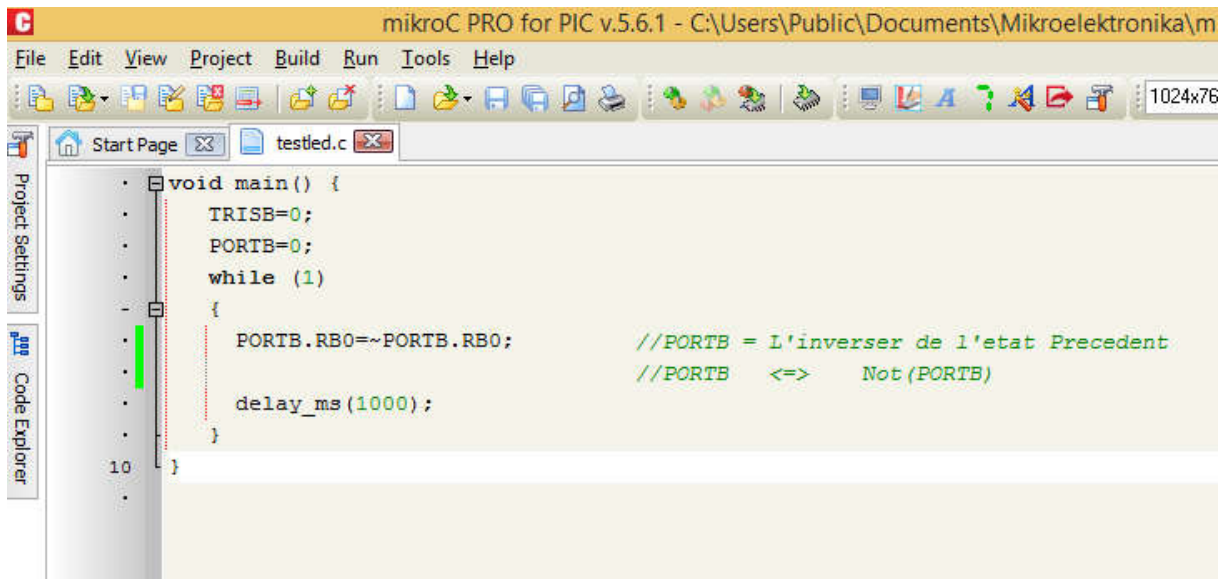
Choisir TD1EXO1.hex ensuite Quvrir ensuite Ok

Exécuter la simulation par l'appuie sur le bouton suivant



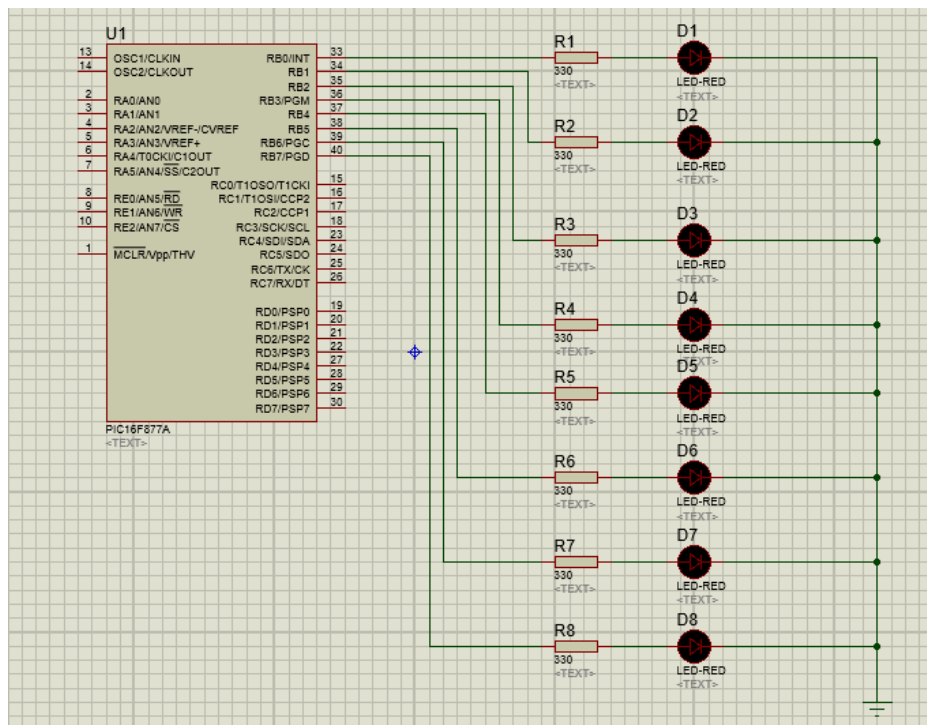
2<sup>ème</sup> Solution de l'exercice 1 :

L'exercice 1 peut être résolu en utilisant le symbole de négation (~) (porte NOT) de la manière suivante



**Exercice 2 :**

Réaliser le schéma suivant



Saisir le code suivant

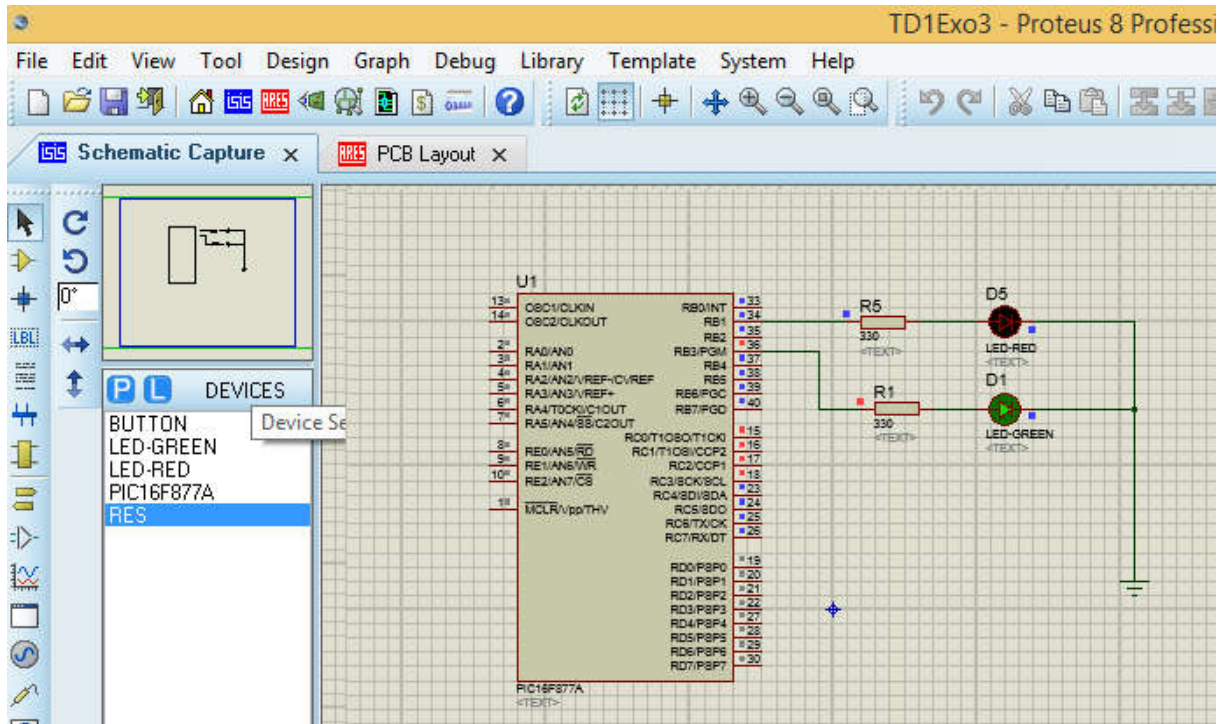
```

void main() {
    TRISB=0;
    PORTB=0;
    while (1)
    {
        PORTB=~PORTB;           //PORTB = L'inverser de l'etat Precedent
                                //PORTB <=> Not(PORTB)
        delay_ms(1000);
    }
}
    
```

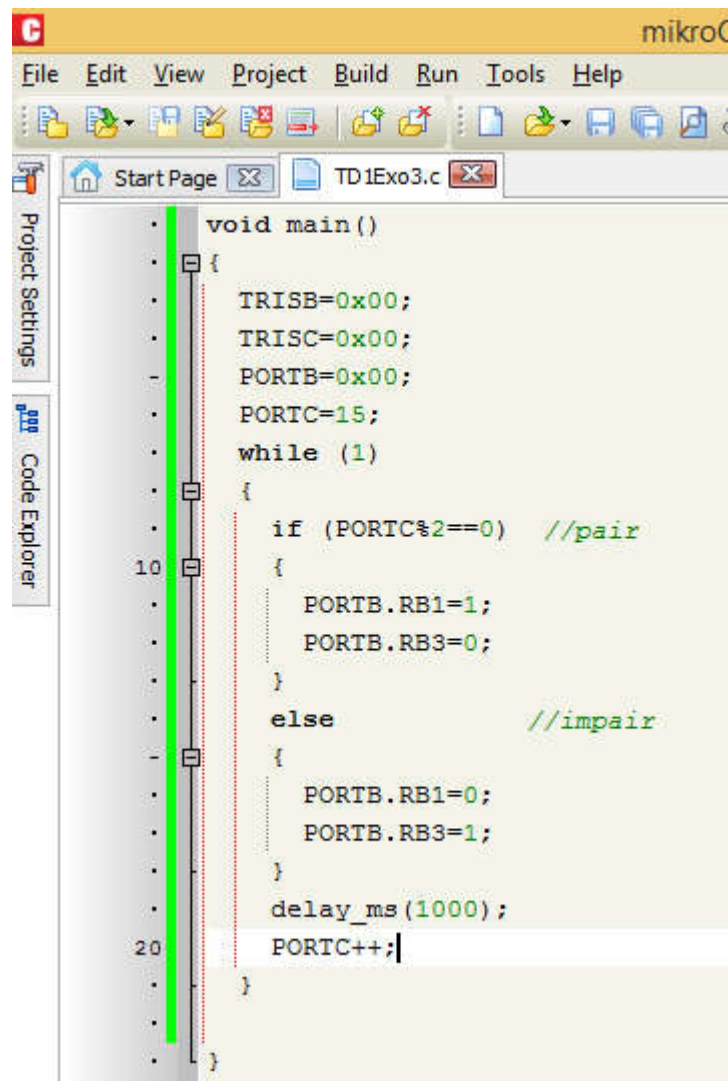
Suivre les mêmes étapes expliquer dans l'exercice 1

### Exercice 3

Saisir le schéma comme ci-dessous



Saisir le code C suivant

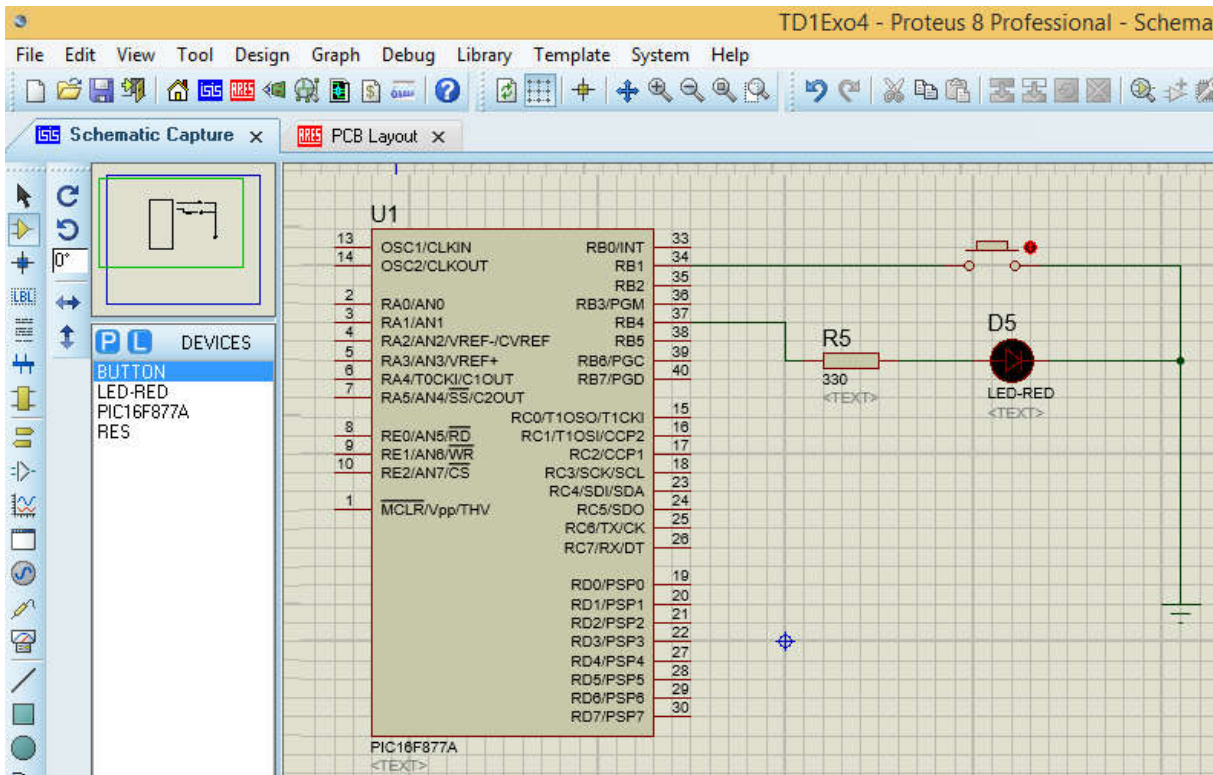


```
void main()
{
    TRISB=0x00;
    TRISC=0x00;
    PORTB=0x00;
    PORTC=15;
    while (1)
    {
        if (PORTC%2==0) //pair
        {
            PORTB.RB1=1;
            PORTB.RB3=0;
        }
        else //impair
        {
            PORTB.RB1=0;
            PORTB.RB3=1;
        }
        delay_ms(1000);
        PORTC++;
    }
}
```

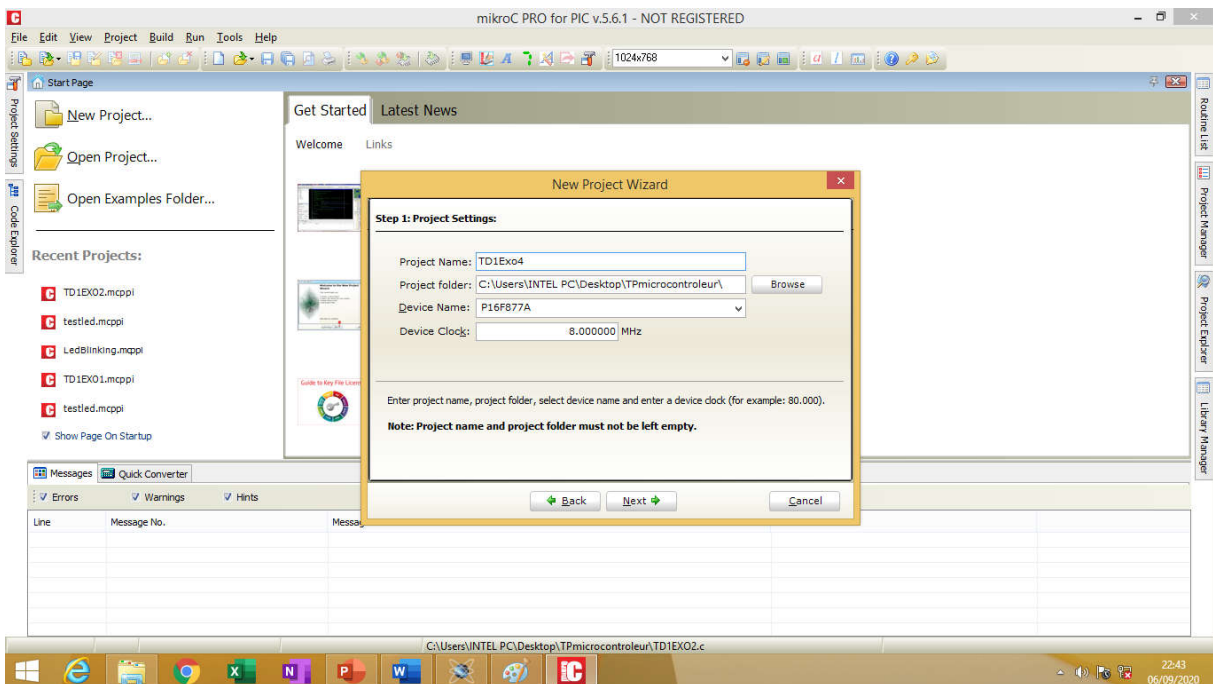


### Exercice 4

Après la création du projet comme indiqué dans les étapes de l'exercice 1, saisissez le schéma suivant

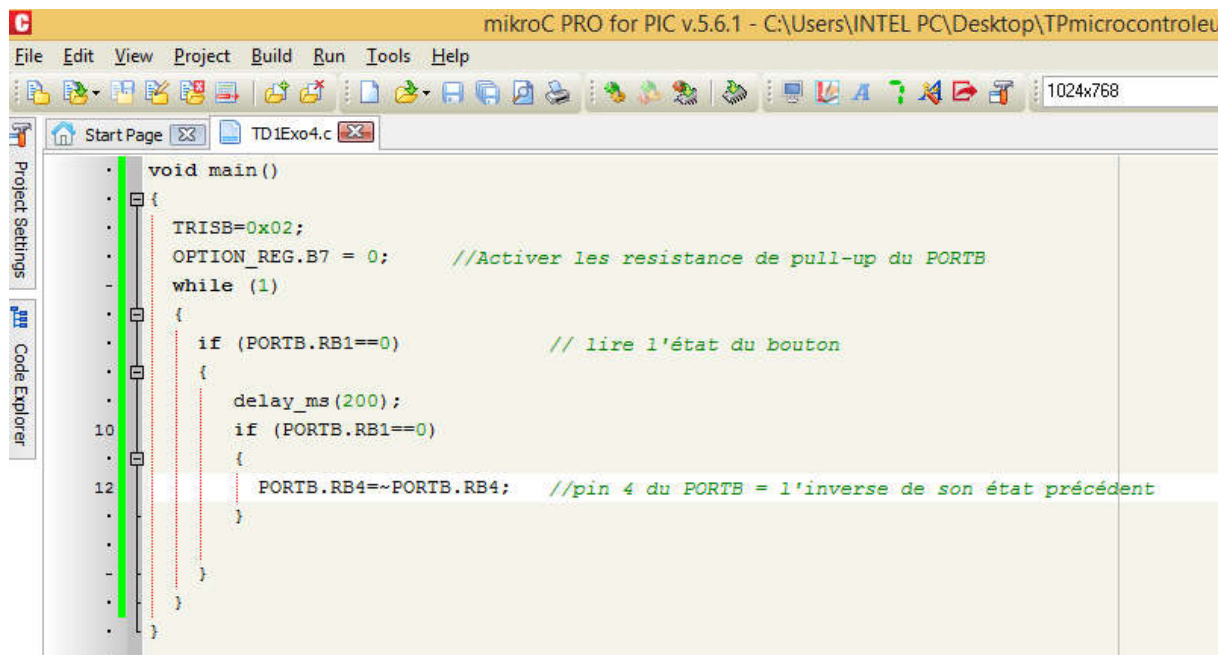


Ensuite créer un nouveau projet dans MikroC pro et nommé le TD1Exo4 comme indiqué sur la figure suivante





Ensuite Next, Next, Next et enfin Finish. Saisir le code source C suivant



```
void main()
{
    TRISB=0x02;
    OPTION_REG.B7 = 0;    //Activer les resistance de pull-up du PORTB
    while (1)
    {
        if (PORTB.RB1==0)    // lire l'état du bouton
        {
            delay_ms(200);
            if (PORTB.RB1==0)
            {
                PORTB.RB4=~PORTB.RB4;    //pin 4 du PORTB = l'inverse de son état précédent
            }
        }
    }
}
```

## Exercice 5

A vous de le faire