

Électronique :

TP_Pic_01

Table des matières

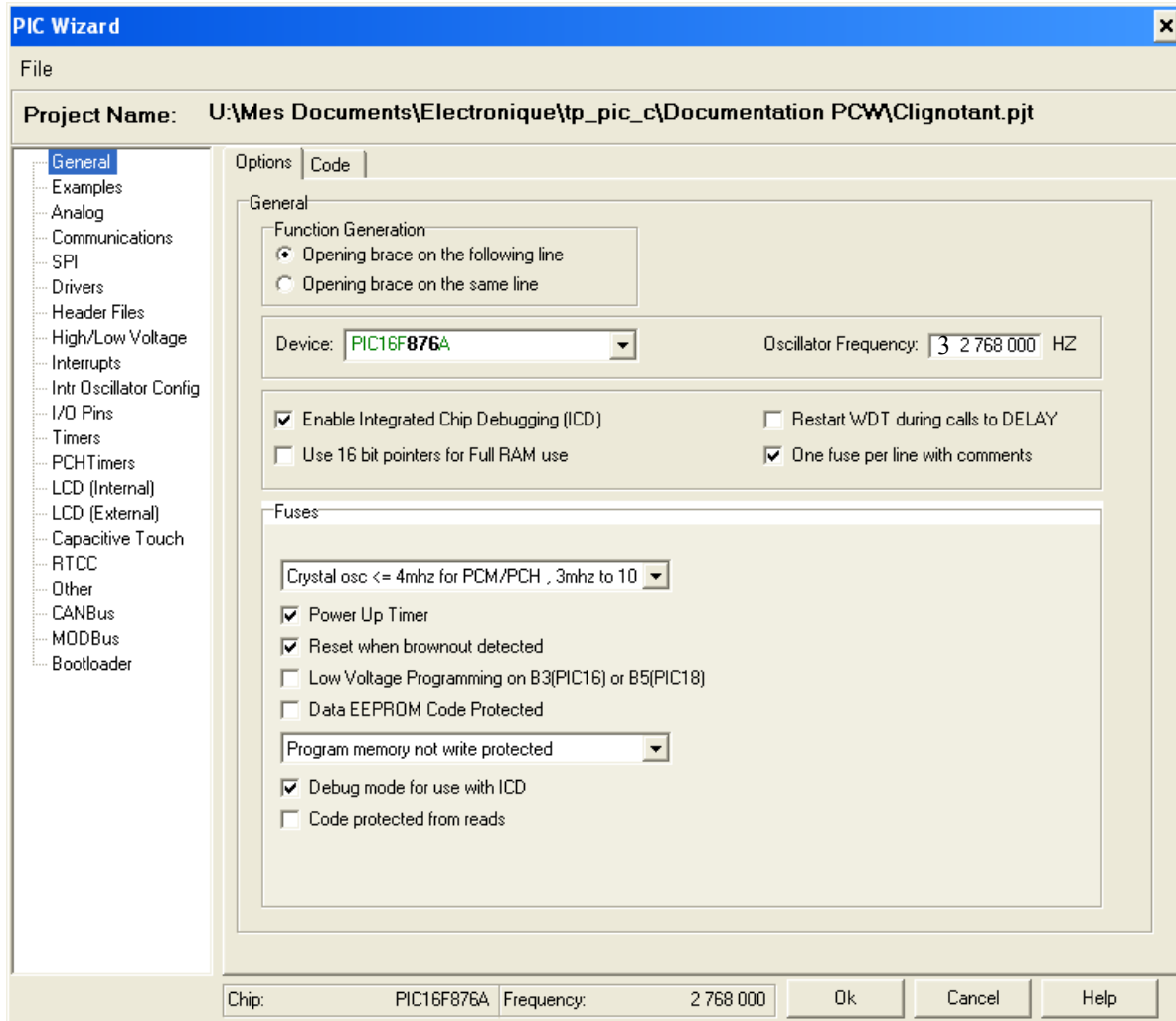
TP_Pic_01.....	1
1. Mise en œuvre matérielle :.....	3
2. Programme N°1 :.....	4
3. Modification du programme N°1(clignotant) :.....	7
4. Modification du programme N°2 notion de « PORT » :.....	8
5. Programme N°3 Notion de variable:.....	9

1. Mise en œuvre matérielle :

- Carte de développement : Circuit imprimé
- Cordon USB : adaptateur RJ/USB
- Interface de programmation : PC
- Logiciel de développement : PCW.exe
- Led/boutons poussoirs : sur la carte

2. Programme N°1 :

Documentation du logiciel PCW.



Configuration du projet en fonction de :

- la référence du PIC





PIC16F876A

- type de l'oscillateur



Il est écrit dessus la fréquence de l'oscillateur.

 Test_24_09_2015.c :Fichier **source** d'extension qui contient le code source en C

 Test_24_09_2015.h :Fichier **include** d'extension qui contient les paramètres liés au mot de configuration du PIC et dans lequel on pourra déclarer les fonctions utilisées dans le code source.

- Code du fichier **source** :

```
16
17 void main()
18 {
19     port_B_pullups(0xFF);
20
21     while(TRUE)
22     {
23         output_b(11);
24     }
25
26 }
```

- Code du fichier **include** :

```
#include <16F876A.h>
#device ICD=TRUE
#device adc=16

#FUSES NOWDT           //No Watch Dog Timer
#FUSES XT              //Crystal osc <= 4mhz for PCM/PCH , 3mhz to 10 mhz for PCD
#FUSES PUT            //Power Up Timer
#FUSES NOLVP          //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O
#FUSES DEBUG          //Debug mode for use with ICD

#use delay(clock=2768000)

#use FIXED_IO( B_outputs=PIN_B5,PIN_B0 )
#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8,stream=PORT1)
```

3. Modification du programme N°1(clignotant) :

Clignotement de 1s (1000ms)

Phase 1 : allumer LED

Phase 2 : éteindre LED

({boucle infinie})

```
void main()  
{  
    port_B_pullups(0xFF);  
  
    while(1)  
    {  
        Output_b(0x0B);  
        delay_ms(1000);  
        Output_b(0x00);  
        delay_ms(1000);  
    }  
}
```



4. Modification du programme N°2 notion de « PORT » :

Phase 1 : allumer toutes les LED

Phase 2 : éteindre toutes les LED

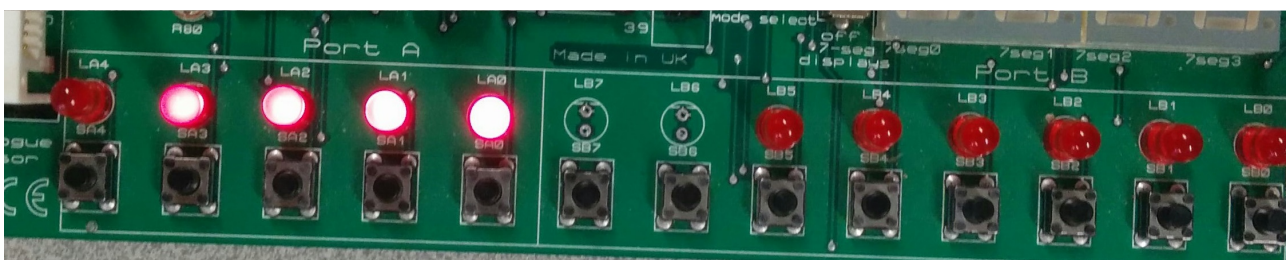
(0xFF) correspond aux 8 LED

```
void main()  
{  
    port_B_pullups(0xFF);  
    while(1)  
    {  
        Output_b(0xFF);  
        delay_ms(1000);  
        Output_b(0x00);  
        delay_ms(1000);  
    }  
}
```

PORT A :

Toutes les LED du port A allumées

```
void main()  
{  
    while(TRUE)  
    {  
        Output_a(0xFF);  
    }  
}
```



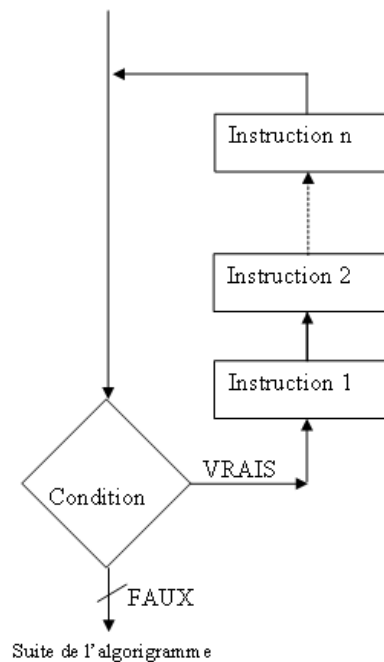
5. Programme N°3 Notion de variable:

- Utilisation d'une variable qui varie !

Le programme fait clignoter les leds du portb de plus en plus rapidement, de 1s de clignotement à 50ms. (Ici il faudrait mettre int16 comptage(16bits) , car là, 1000ms ne passe pas dans 8bits.)

```

1  #include <ma_variable.h>
2  int comptage ;
3
4  void main()
5  {
6
7      comptage = 1000;
8      port_B_pullups(0xFF);
9
10     while(comptage >= 50)
11     {
12         Output_b(0xFF);
13         delay_ms (comptage);
14         Output_b(0x00);
15         delay_ms (comptage);
16         comptage=comptage-25;
17     }
18 }
19
20 
```



- Lire le contenu du portA et l'affecter au PortB.

```

1  #include <affectation_A_a_B.h>
2
3
4  void main()
5  {
6      port_B_pullups(0xFF);
7      while(TRUE)
8      {
9          output_B(input_A());
10     }
11 }
12 
```

Ici quand nous appuyons sur un bouton du port A, une LED du port B s'allume.