



## A partir de :

MPLAB et carte de simulation ISIS CARTE\_TEST\_PIC18F4620.DSN

DataSheet PIC18F4620

Programme flashIT\_demo\_horloge.c (voir annexe)

Programme init\_horloge\_interne.c (voir annexe)

### 1) MODE HORLOGE EXTERNE

En mode oscillateur externe, la fréquence de l'oscillateur du PIC sur ISIS est définie dans les propriétés du composant.

Pour choisir la valeur de l'horloge externe, sur carte de simulation CARTE\_TEST\_PIC18F4620.DSN, faire clic-droit sur le composant puis dans les propriétés configurer la fréquence de l'oscillateur externe à 4Mhz (FOSC), quelle sera alors le temps de cycle machine (TCY) ?.

Le programme flashIT fait clignoter la LED, la demi-période est fixée par le TIMER0.

- Dé-commenter la ligne `#pragma config OSC = XT`
- Commenter la ligne `init_horloge_interne();`
- Retirer le fichier `init_horloge_interne.c` du projet s'il est présent.

Que signifie `#pragma config OSC = XT` ( voir dans Help – Topics – PIC18 config settings ) ?

A partir de la valeur placée dans le registre T0CON, donnez la demi-période sur la LED, puis la période.

Tester maintenant le programme flashIT.c et vérifier la valeur de cette période à l'aide de l'oscilloscope.

Modifier la fréquence de l'oscillateur externe en 8Mhz. Quelle période de clignotement doit-on obtenir ? Vérifier expérimentalement.

### 2) MODE HORLOGE INTERNE

En mode oscillateur interne la fréquence FOSC sur ISIS est définie dans le programme, elle est basée sur l'oscillateur interne 8Mhz du PIC, la fréquence définie dans les propriétés du composant n'est plus considérée.

Commenter la ligne `#pragma config OSC = XT`

Dé-commenter la ligne `init_horloge_interne();`

Ajouter le fichier `init_horloge_interne.c` au projet. (ce fichier sera maintenant compilé avec flashIT.c)

Ouvrir le fichier `init_horloge_interne.c`, le lire

Rechercher dans MPLAB help-topics-PIC config settings la signification des prama

```
#pragma config OSC = INTIO67
```

```
#pragma config WDT = OFF
```

```
#pragma config LVP = OFF
```

```
#pragma config PBADEN = OFF
```

Rechercher dans le datasheet du PIC les registres OSCCON et OSCTUNE. (Paragraphe OSCILLATOR CONFIGURATIONS )

Quels sont les rôles des bits IRCF<sub>x</sub>, PLEN, SCS<sub>x</sub>

Indiquer en fonction des valeurs programmée dans la fonction void `init_horloge_interne(void)` la fréquence FOSC et FCY (FCY = FOSC/4) du PIC. En déduire la période attendue sur la LED. Vérifier expérimentalement.

Desactiver la PLL et configurer maintenant le registre OSCON de manière à disposer d'une horloge interne (FOSC) de 4Mhz et mesurer la période sur la LED

Configurer maintenant le registre OSCON et OSCTUNE de manière à disposer d'une horloge interne à 125Khz et mesurer la période sur la LED. Conclure.



## Programme init horloge interne.c

```
// initialisation horloge PIC
#pragma config OSC = INTIO67 //pour INTRC-OSC2 as RA6, OSC1 as RA7
#pragma config WDT = OFF //pour watch dog timer disable
#pragma config LVP = OFF //pour low voltage program disable

void init_horloge_interne(void)
{
#if defined(__18F4620)|| defined (__18F2620) OSCCONbits.IRCF2=1;
    OSCCONbits.IRCF1=1;
    OSCCONbits.IRCF0=1;
    OSCTUNEbits.PLEN=1;
    OSCCONbits.SCS1=0;
    OSCCONbits.SCS0=0;
    ADCON1=0x0F;
}
}
```

## Programme flashIT.c

```
// CD 09/2011
// FLASHIT.c clignotement de la LED

#include <p18f4620.h>

#pragma config PBADEN=OFF
#pragma config OSC = XT // a commenter si horloge interne

#define LED PORTBbits.RB1
#define TRISLED TRISBbits.TRISB1

#pragma interrupt traiteIT
void traiteIT(void)
{
    if(INTCONbits.TMR0IF)
    {
        LED = !LED; //bascule LED
        INTCONbits.TMR0IF = 0;
    }
}

#pragma code it=0x08
void saut_sur_spIT(void)
{
    _asm goto traiteIT _endasm
}
#pragma code

void main()
{
//    init_horloge_interne(); // decommenter si horloge interne
    TRISLED = 0;
    LED = 0;
    T0CON = 0x82;
    INTCONbits.TMR0IE = 1;
    INTCONbits.GIEH = 1;

    while(1);
}
}
```