

TP Horloge Interne (P18F26K22)

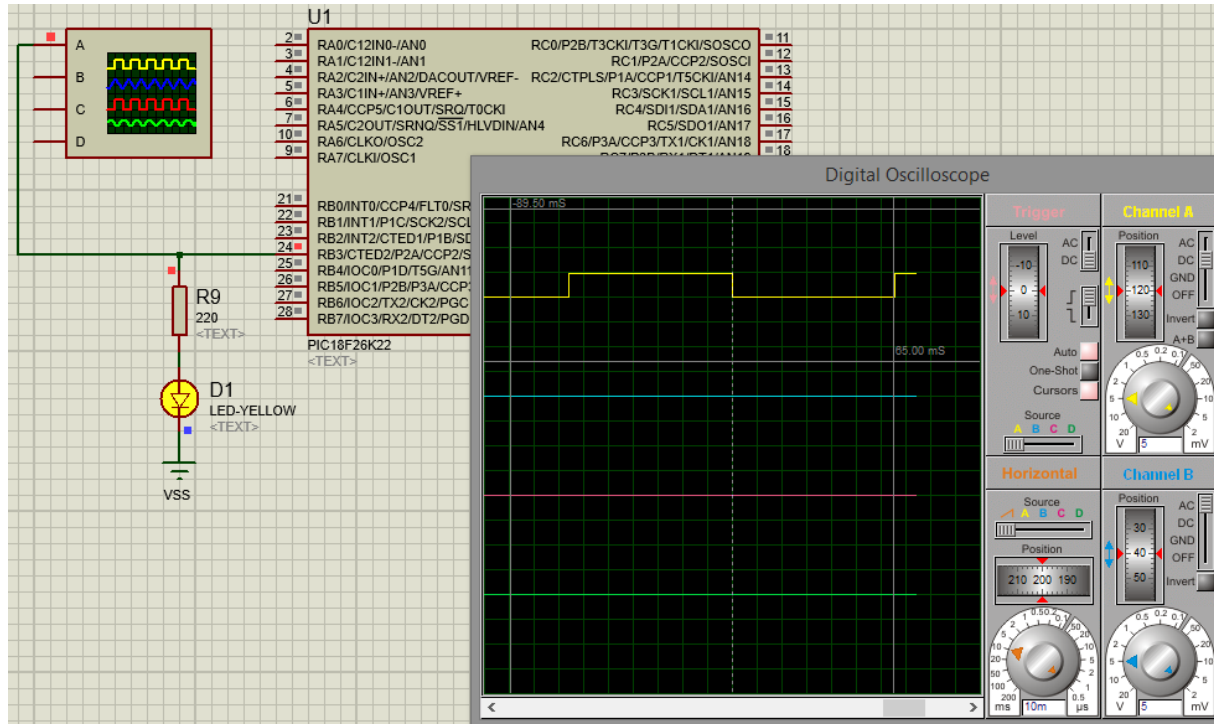


A partir de :

MPLAB et carte de simulation ISIS TP_Horloge_Interne.pdsprj
DataSheet PIC18F26K22
Programme flashit_TP_horloge_XC8_P18F26K22.c (voir annexe)

Réaliser un projet MPLABX avec le fichier source flashit_TP_horloge_XC8_P18F26K22.c, ISIS comme debugger avec le schéma TP_Horloge_Interne.pdsprj

Le configure active le TIMER0 en mode 16bits pour compter l'horloge interne FCY (FOSC/4) du microcontrôleur avec un pré-diviseur par 8. Un débordement a lieu tous les 65536 cycles fois 8.



- a) Vérifier le fonctionnement. L'oscilloscope doit indiquer une demi-période de 65mS, il y a une imprécision dans cette mesure, la valeur exacte est 65,536mS (Remarquer que $65,536\text{mS} = 2^{16} \mu\text{S}$)

En mode oscillateur interne la fréquence FOSC est définie dans le programme, elle est basée sur l'oscillateur interne 16Mhz du PIC18F26K22.
Elle peut être un sous multiple ou un multiple de 16MHz.

- b) Rechercher dans le datasheet du PIC les registres OSCCON et OSCTUNE. (Paragraphe OSCILLATOR CONFIGURATIONS)
Quels sont les rôles des bits IRCF_x, PLEN, SCS_x ?

- c) Indiquer en fonction des valeurs programmée dans l'exemple du TP dans IRCF_x, PLEN, SCS_x la fréquence FOSC et FCY (FCY = FOSC/4) du PIC.
d) Quelle est la configuration du TIMER0 (TMR0) si T0CON=0x82 ?

Le TIMER0 provoque une interruption lors de son débordement (overflow)

- e) En déduire la période attendue sur la LED. Vérifier expérimentalement.
f) Désactiver la PLL qui multiplie FOSC par 4, vérifier la nouvelle demi-période de débordement (4 fois plus longue soit $65536 \times 4 = 262,144\text{mS}$)
g) Modifier les bits IRCF de manière à obtenir une période de 1,048S

TP Horloge Interne (P18F26K22)



```
// CD 10/2014
// FLASHIT.c clignotement de la LED sur P18F26K22
// FOSC=32MHz, FCY=8MHz
// IT sur TMR0 16bits avec prédiviseur par 8. Horloge TMRO=8MHz/8=1MHz
// Le TMR0 a une horloge de 1MHz, T=1uS. Débordement et IT toutes les
// 2^16uS soit 65536uS.

#include <xc.h>

// Horloge interne et autres configurations
#pragma config FOSC = INTIO67, FCMEN = OFF, PLLCFG=OFF // CONFIG1H
#pragma config IESO=OFF, PWRTEN=OFF, BOREN=OFF, WDTEEN=OFF, CCP2MX=PORTC1 // CONFIG2H
#pragma config PBADEN=OFF, T3CMX=PORTC0, P2BMX=PORTC0, CCP3MX=PORTB5 // CONFIG3H
#pragma config MCLRE=EXTMCLR, STVREN=OFF, LVP=ON, XINST=OFF, DEBUG=OFF // CONFIG4H
#pragma config CP0 = OFF, CP1 = OFF, CP2 = OFF, CP3 = OFF // CONFIG5L
#pragma config CPB = OFF, CPD = OFF // CONFIG5H
#pragma config WRT0 = OFF, WRT1 = OFF, WRT2 = OFF, WRT3 = OFF // CONFIG6L
#pragma config WRTB = OFF, WRTC = OFF, WRTD = OFF // CONFIG6H
#pragma config EBTR0 = OFF, EBTR1 = OFF, EBTR2 = OFF, EBTR3 = OFF // CONFIG7L
#pragma config EBTRB = OFF

#define LED PORTBbits.RB3
#define TRISLED TRISBbits.TRISB3

void interrupt traiteIT(void) // aucun paramètre pour un SP d'IT, son appel est asynchrone
{
    if(INTCONbits.TMR0IF) //vérifie un débordement sur TMR0
    {
        LED = ~LED; //bascule LED
        INTCONbits.TMR0IF = 0; //efface le drapeau d'IT
    }
}

void main()
{
    // Configuration horloge interne
    OSCCONbits.IRCF2=1;
    OSCCONbits.IRCF1=1;
    OSCCONbits.IRCF0=0;
    OSTUNEbits.PLLEN=1; // PLL (x4) FOSC=32MHz
    OSCCONbits.SCS1=0; // osc interne sur primary oscillator
    OSCCONbits.SCS0=0;
    ANSELB=0; // PB numérique
    TRISLED = 0;
    LED = 0;
    TOCON = 0x82; //Active le TIMER0 - prescaler 1:8
    INTCONbits.TMR0IE = 1; //Autorise interruption sur TMR0
    INTCONbits.GIEH = 1; //autorise toutes les IT démasquées

    while(1); // une boucle infinie, tout fonctionne en IT
}
```