

# TD Timers

## Questions

1. Quelle est la différence entre un Compteur et un Timer ?
2. Quel est le rôle du prédiviseur ?
3. Quel est le bit qui permet de sélectionner le mode compteur ou Timer dans le module TIMER1 ?
4. Pour  $F_{osc} = 8MHz$ , déterminer la période de débordement maximale du TIMER0, TIMER1 et TIMER2.
5. Pour  $F_{osc} = 20MHz$ , et un prédiviseur de 2 ; déterminer la période maximale de débordement du TIMER0.
6. Pour  $F_{osc} = 16MHz$ , combien d'impulsions doit compter le TIMER1 pour avoir une période de débordement de 20ms ?
7. Trouvez le contenu du registre T1CON pour programmer le TIMER1 en mode synchrone avec un prédiviseur de 4 et utilisant un oscillateur à quartz comme horloge externe.
8. A l'aide du TIMER0 en mode 16 bits, sans prédivision. Déterminer le mot de commande et la valeur de préchargement du registre TMR0 pour obtenir une période de débordement de 1ms. Supposons que le microcontrôleur PIC18F4520 est cadencé à 8 MHz.
9. Pour  $F_{osc} = 8MHz$ , Trouvez le contenu de T2CON et PR2 du TIMER2 pour obtenir une période de débordement de 10ms.
10. Sans prédivision, Quelle est la durée d'un cycle, si on précharge TMR1H par 100 et TMR1L par 100 ? Le microcontrôleur PIC18F4520 est cadencé à 20 MHz.

## Exercice 1

On se propose de simuler le fonctionnement d'une minuterie avec un microcontrôleur PIC18F4520 cadencé à 8MHz. Le temps d'allumage de la LED connectée à la broche RC1 est fixé à 30s. Le bouton poussoir est connecté à la broche RB0.

1. Déterminer le mot de commande et la valeur à précharger dans le registre TMR0.
2. Ecrire le programme en langage XC8.

## Exercice 2

Nous voulons utiliser le TIMER0 pour générer sur la broche RC1 un signal carré de période 1s (500ms au niveau haut + 500ms au niveau bas). Le microcontrôleur PIC18F4520 est cadencé à 8MHz.

1. Déterminer le mot de commande et la valeur initiale du TIMER0.
2. Pour une valeur de prédivision de 256
  - a) Calculer la valeur initiale du TIMER0.
  - b) Déterminer l'erreur commise sur une période.

## Exercice 3

Pour une fréquence de quartz 8 MHz et en utilisant le TIMER2 ; écrire une séquence d'instructions pour générer des interruptions périodiques toutes les 8 ms.

## Exercice 4

Nous voulons générer une interruption toutes les secondes. Cette interruption est utilisée comme base de temps d'une montre numérique. Le TIMER3 du microcontrôleur PIC18F4520 est attaqué par un signal d'horloge provenant d'un quartz de 32,768kHz connecté aux broches T1OSI et T1OSO.

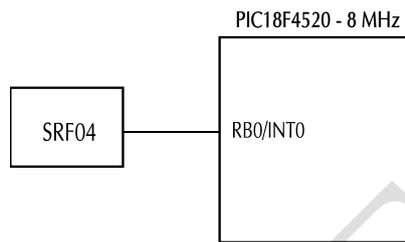
1. Déterminer le mot de commande et la valeur initiale du registre TMR3.
2. Ecrire le corps de la routine d'interruption et la séquence de configuration du TIMER3.

## Exercice 5

La mesure de la fréquence consiste à compter le nombre de période du signal à mesurer pendant une seconde. Le signal à mesurer est appliqué à l'entrée RC0/T13CKI. La fréquence de l'oscillateur principal  $F_{osc} = 8MHz$

1. En utilisant une variable compteur, quelle est la fréquence maximale qu'on peut mesurer (Timer1 sans prédivison) ? il faut bien noter que la fréquence du signal à mesurer ne peut pas dépasser  $F_{osc}/4$ .
2. La valeur de la fréquence doit être chargée dans la variable globale FreqVal1 . Quel est le type de cette variable ?
3. Ecrire le programme du fréquencesmètre numérique, vous pouvez utiliser le TIMER0 pour générer la base de temps d'une seconde.

### Exercice 6



Le détecteur à ultrason « SRF04 » est utilisé en robotique pour mesurer la distance entre le robot et l'obstacle le plus proche. A chaque mesure, le détecteur délivre à sa sortie une impulsion de largeur proportionnelle à la distance. Connaissant la vitesse de déplacement de son (environ 340m/s) ; il est aisé de déduire qu'une impulsion de largeur 58 $\mu$ s correspond bien à une distance de 1cm.

1. Comment configurer la timer2 pour avoir une période de débordement de 58 $\mu$ s, sachant que le microcontrôleur PIC18F4520 est cadencé à 8MHz.
2. Quelle est la portée du détecteur si la largeur maximale de l'impulsion est de 18ms.
3. La valeur de la distance est chargée dans la variable globale « Distance ». Quel est le type de cette variable.

Ecrire le programme complet du télémètre à ultrason.

## Corrigés

### Réponse aux questions

- La différence réside au niveau de l'horloge :
  - Horloge externe : mode compteur
  - Horloge interne : mode Timer
- Le prédiviseur permet d'augmenter la période de débordement.
- Le bit T0CS du registre T0CON
- Période de débordement maximale pour chaque compteur :
 
$$\text{TIMER0} : T_{Dmax} = \frac{4}{8} 10^{-6} \times 256 \times 2^{16} = 8,388 \text{ s}$$

$$\text{TIMER1} : T_{Dmax} = \frac{4}{8} 10^{-6} \times 8 \times 2^{16} = 262,144 \text{ ms}$$

$$\text{TIMER2} : T_{Dmax} = \frac{4}{8} 10^{-6} \times 16 \times 2^8 \times 16 = 32,768 \text{ ms}$$
- La période de débordement maximale est obtenue en mode 16bits
 
$$T_{Dmax} = \frac{4}{40} 10^{-6} \times 2 \times 2^{16} = 13,1072 \text{ ms}$$
- $Nb. Impulsions = \frac{T_D}{T_h} = T_D \times F_h = 20 \cdot 10^{-3} \times \frac{16}{4} 10^6 = 80000 \text{ Impulsions}$
- T1CON = 0b10101011
- $\frac{T_D}{T_h} = (2^{16} - TMR0Init) = 1000/0,5 = 2000$ , d'où la valeur de précharge :
 
$$TMR0Init = 65536 - 2000 = 63536$$
, cette valeur doit être chargée dans les registres TMR0H et TMR0L.
 
$$TMR0H = \frac{63536}{256} = 248 \text{ et } TMR0L = 63536 - 248 \times 256 = 48$$

Le mot de commande T0CON = 0b100X1XXX ; on peut prendre la valeur 0b10001000
- $\frac{T_D}{T_h} = Prediv \times (PR2 + 1) \times Postdiv = 10000/0,5 = 20000$   
 En prenant Prediv = 16, PR2 = 124 et Postdiv = 10 :
 
$$T_D = 0,5 \times 16 \times (124 + 1) \times 10 = 10 \text{ ms}$$

$$T2CON = 0b0100111X$$
- La durée d'un cycle = période de débordement
 
$$TMR1Init = 100 \times 256 + 100 = 25700$$

$$T_D = T_h \times (2^{16} - TMR1Init) = \frac{4}{20 \cdot 10^6} (65536 - 25700) = 7,967 \text{ ms.}$$

### Exercice 1

- $T_D = T_h \times Prediv \times (2^{16} - TMR0Init)$  ; cherchons la période de débordement maximale :
 
$$T_{Dmax} = T_h \times Prediv_{Max} \times 2^{16}$$

$$= 0,5 \times 256 \times 65536 = 8,388 \text{ s} ; T_{Dmax} < T = 30 \text{ s, nous sommes obligés d'utiliser une}$$
 variable compteur :  $T = Count \times T_D$  ; nous pouvons choisir  $T_D = 3 \text{ s}$  pour avoir une valeur entière de Count (Count = 10).
 
$$T_D = T_h \times Prediv \times (2^{16} - TMR0Init) = 3 \text{ s.}$$

$$T_h = 0,5 \mu\text{s, pour } Prediv = 128 \text{ et } TMR0Init = 18661.$$
 Mot de commande : T0CON = 0b10000110 et TMR0Init = 18661
- Programme

```
#include <xc.h>
#define _XTAL_FREQ 8000000
#pragma config OSC = HS, WDT = OFF, LVP = OFF
#define S1 PORTBbits.RB0
#define LED PORTCbits.RC1
```

```

#define TMR0Init 18661
char Count ;
void delay__30s(void){
    T0CONbits.TMR0ON = 1 ;
    for(Count = 0 ; Count < 10 ; Count++){
        TMR0 = TMR0Init;
        INTCONbits.TMR0IF = 0;
        while(INTCONbits.TMR0IF == 0) continue;
    }
    T0CONbits.TMR0ON = 0 ;
}
void main(void) {
    ADCON1 = 0x0F ;
    TRISBbits.TRISB0 = 1;          // RB0 en entrée
    TRISCbits.TRISC1 = 0;          // RC1 en sortie
    T0CON = 0x06;                  // Mot de commande (TMR0ON = 0)
    while(1){
        if(S1 == 1) {
            LED = 1;                // LED allumée
            delay__30s();           // Attente 30s
            LED = 0;                // LED éteinte
        }
    }
}

```

## Exercice 2

1. Le TIMER0 doit fonctionner en mode Timer (horloge interne) avec une période de débordement de 500ms.

$$T_D = \frac{4}{8} 10^{-6} \times Pre\ div \times (2^{16} - TMROInit) = 0,5s$$

$$Pre\ div \times (2^{16} - TMROInit) = \frac{500000}{0.5} = 1000000$$

Nous avons deux inconnus *Prediv* et *TMROInit*. Nous fixons *Prediv* et nous cherchons la valeur de *TMROInit*.

Pour *Prediv* = 8  $\rightarrow 2^{16} - TMROInit = \frac{1000000}{8} = 125000 > 2^{16} - 1$  valeur inacceptable.

Pour *Prediv* = 16  $\rightarrow 2^{16} - TMROInit = \frac{1000000}{16} = 62500 < 2^{16}$ , d'où *TMROInit* = 3036

Toutes les valeurs de prédivison supérieures à 8 sont valables.

TOCON = 0b10000011

2. Pour *Prediv* = 256

a)  $2^{16} - TMROInit = \frac{1000000}{256} = 3906,25$  Nous devons prendre la partie entière, ceci influe légèrement sur la précision de la période de débordement.

$$TMROInit = 65536 - 3906 = 61630$$

b) Calculons alors la période réelle :

$$T_D = 0,5 \times 256 \times 3906 = 499,968ms$$

L'erreur commise :  $\Delta T_D = 2 \times (500 - 199,968) = 64\mu s$ .

## Exercice 3

Nous devons calculer la valeur à écrire dans le registre PR2 :

$$T_h = \frac{4}{F_{osc}} = \frac{4}{8} 10^{-6} = 0,5\mu s$$

$$T_D = T_h \times Prediv \times (PR2 + 1) \times Postdiv$$

$$\Rightarrow PR2 = \frac{T_D}{T_h \times Prediv \times Postdiv} - 1 = \frac{16000}{Prediv \times Postdiv} - 1$$

Il suffit de prendre  $Prediv = 16$  et  $Postdiv = 4$  Pour avoir  $PR2 = 249$

Le mode de commande  $T2CON = 0x1E$  ou  $0x1F$ .

```
PR2 = 249 ; // charger PR2 par 249
RCONbits.IPEN = 1 ; // valider la logique de priorité
PIR1bits.TMR2IF = 0 ; // remettre l'indicateur d'inter. à 0
PIE1bits.TMR2IE = 1 ; // valider l'inter. du TIMER2
INTCONbits.PEIE = 1 ; // valider l'inter. des périphériques
T2CON = 0x1E ; // activer TIMER2, Préddiv=16 et Postdiv=4
INTCONbits.GIE = 1 ; // valider l'inter. globale
```

#### Exercice 4

1. Mode de commande :

$$T_D = T_h \times Prediv \times (65536 - TMR3Init) ; T_h = \frac{1}{32768} s$$

$$Prediv \times (65536 - TMR3Init) = \frac{T_D}{T_h} = 32768$$

Pour  $Prediv = 1$  ;  $TMR3Init = 32768 = 0x8000$

Le bit  $T10SCEN$  doit être positionné à 1 pour activer l'oscillateur.

$T3CON = 0x03$

2. Configuration du TIMER3 en mode interruption

```
void __interrupt() isr_Timer3(void){
    PIR2bits.TMR3IF = 0 ;
    TMR3H = TMR3H + 0x80 ; // il suffit d'initialiser TMR3H
    :
}

:
T1CONbits.T10SCEN = 1 ; //activer l'oscillateur
TMR3H = 0x80 ; // Initialiser TMR3
TMR3L = 0x00 ;
PIR2bits.TMR3IF ; // Remettre l'indicateur à 0
PIE2bits.TMR3IE ; // valider l'inter. du TIMER3
T3CON = 0x03 ; // horloge externe, TMR3ON
INTCON = 0xC0 ; // GIE =1 et PEIE = 1.
:
```

#### Exercice 5

1. La fréquence maximale  $F_{MAX} = \frac{F_{OSC}}{4} = 2MHz$ .
2. `FreqVal` est de type `unsigned long`.
3. Programme du fréquencemètre

Puisque  $TMR1$  est un registre de 16 bits, il peut déborder plusieurs fois en une seconde. Il peut déborder au maximum  $\frac{2000000}{65536} \approx 30$  fois. Le nombre de débordements doit être compté dans une variable.

```
#include <xc.h>
#define _XTAL_FREQ 8000000
#pragma config OSC = HS, WDT = OFF, LVP = OFF
char T1OV_Cnt;
unsigned long FreqVal;
void __interrupt (high_priority) T1_ISR(void)
{
    if(PIR1bits.TMR1IF){
        PIR1bits.TMR1IF = 0;
```

```
        T1OV_Cnt++;           // compte le nombre de débordements
    }
}
void delay_1s (void){
    int i;
    TMR0 = 3036;
    INTCONbits.TMR0IF = 0;
    while(!(INTCONbits.TMR0IF));
}
void main (void) {
    T0CON = 0x85 ;
    TMR1 = 0x00;
    PIR1bits.TMR1IF = 0;
    RCONbits.IPEN = 1;
    IPR1bits.TMR1IP = 1;
    PIE1bits.TMR1IE = 1;
    T1CON = 0x83;
    INTCON = 0xC0;
    while(1){
        TMR1 = 0x00;
        PIR1bits.TMR1IF = 0;
        T1OV_Cnt = 0;
        delay_1s ();
        FreqVal = (uint32)T1ov_cnt * 65536 + TMR1;
    }
}
```