TP2b – STR : Chronomètre Numérique

1. Introduction

La Figure 1 représente le schéma électronique d'un chronomètre numérique à base d'un microcontrôleur PIC18F4520. Les boutons poussoirs START STOP et CLEAR, permettent respectivement de commencer le comptage de temps ou de l'arrêter ou bien de remettre le compteur à zéro, tandis que les afficheurs 7 segments affichent les dixièmes de secondes, les secondes et les minutes.

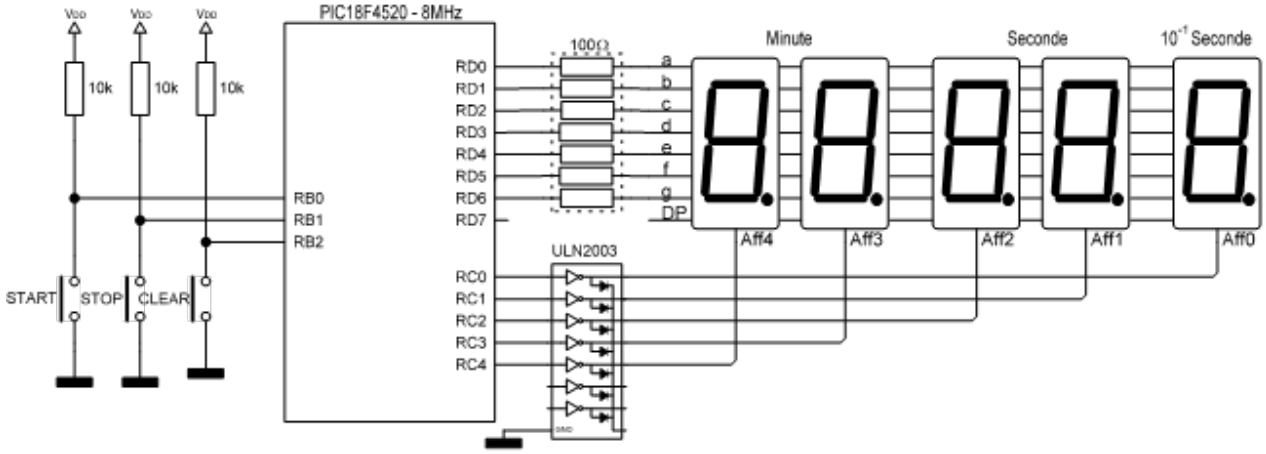

Figure 1 : schéma du chronomètre numérique

2. Démarche à suivre

On propose ici de décomposer le système selon le diagramme de la Figure 2.

- La fonction ***GererConsigne,*** : gère les événements issus des B. Poussoirs et met à jour la variable *ConsFlag*.
- La fonction ***Comptage*** : met à jour les variables *DixSec, Sec et Min* toutes les 100 ms.
- La fonction ***BinTo7SEG*** : convertit en 7Seg. les valeurs à afficher.
- La fonction ***Affiche*** : gère l'affichage multiplexé.

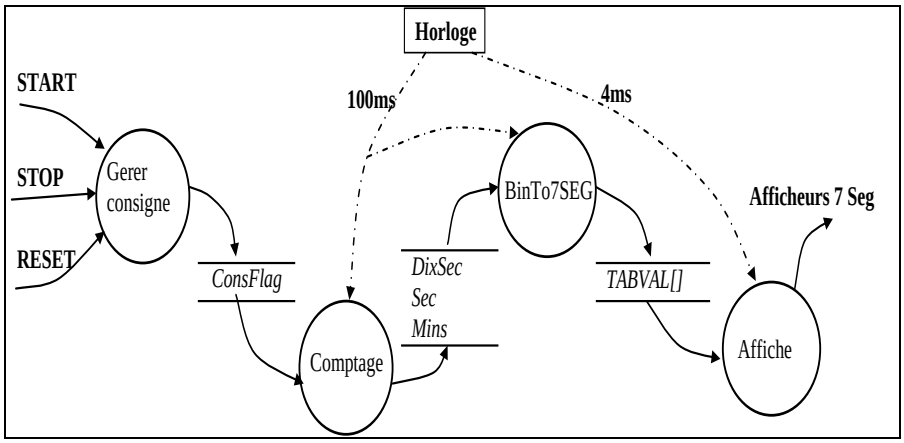

Figure 2: Diagramme d'identification des tâches

3. Travail demandé

Compléter le programme suivant :

```
#include <xc.h>
#pragma config OSC = HS, WDT = OFF, LVP = OFF
#define _XTAL_FREQ 8000000
#define START  PORTBbits.RB0
#define STOP   PORTBbits.RB1
#define CLEAR  PORTBbits.RB2
char CntChrono = 25;      // 100ms/4ms = 25
char DixSec=0 ;           // Dixièmes de seconde
char Sec=0 ;              // Secondes
char Min=0 ;              // Minutes
char ConsFlag = 0;        // ConsFlag = 0 : remise à 0 du compteur
                          // ConsFlag = 1 :    Comptage
                          // ConsFlag = 2 :    Arrêt
char T7Seg[]={0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F} ;
char AffSel = 0x01 ;      // sélection des afficheurs
char TabAff[4] ;          //tableau contenant les valeurs à envoyer aux affi.
char NumAff=0;            // numéro de l?afficheur
//--------------- Prototypes des fonctions -------------------------
void Comptage(char _ConsFlag);
void Bin7Seg(void);
void Affichage(void);
//---------------------------------------------------------------------
void __interrupt() isr_Timer2(void)
{
   PIR1bits.TMR2IF = 0 ;         // remise à 0 de l'indicateur
    Affichage();                 // balayage toutes les 4 ms
   CntChrono--;                  // décrimentation de la variable compteur
    if(CntChrono == 0){          // 100 ms ?
        Compléter …
    }
}
void Comptage(char _ConsFlag)
{
    Compléter …
}
void Affichage()
{   PORTA = 0;                   // Désélectionner tous les afficheurs
    PORTD = TabAff[NumAff];      // chargé le PORTD par le code 7 segments
    PORTA = AffSel;              // sélectionner l'afficheur appropiré
    AffSel = AffSel << 1;        // préparer la prochaine sélection
    NumAff++;                    // préparer l'indice de l'afficheur prochain
      if(NumAff > 4){            // si l'indice > 4 (NumAff : 0 - 4), préparer
                                 // le retour au premier afficheur
         NumAff = 0;
         AffSel = 0x01;
      }
}
void BinTo7Seg(void)
{   char uSec, dSec, uMin, dMin;
    TabAff[0] = T7Seg[DixSec];   // code 7 segments des dixèmes de seconde
    uSec = Sec%10;               // unité de seconde
    TabAff[1] = T7Seg[uSec];     // code 7 segments l'unite de seconde
    dSec = Sec/10;               // dizaine de seconde
    TabAff[2] = T7Seg[dSec];     // code 7 segments de dizaine de seconde
    uMin = Min%10;               // unité de minute
    TabAff[3] = T7Seg[uMin];     // code 7 segments l'unite de minute
    dMin = Min/10;               // dizaine de minute
    TabAff[4] = T7Seg[dMin];     // code 7 segments de dizaine de minute
```

```
}
void Init()
{   ADCON1 = 0x0F;              // Tous les ports en E/S
    TRISA = 0x00;               // PORTA en sortie
    TRISD = 0x00;               // PORTD en sortie
    TRISB = 0xFF;               // PORTB en entrée
    T2CON = 0x3D;               // timer 2 : prédiv = 4, Postdiv = 8
    PR2 = 249;
    PIE1bits.TMR2IE = 1 ;       // validation de l'IT du timer 2
    INTCONbits.PEIE = 1 ;       // validation IT des périphériques
    INTCONbits.GIE = 1 ;        // validation globale des IT
}
void main(void)
{   Init();
     while(1)
    {
     if( START == 0)
         ConsFlag = 1;
     if( STOP == 0)
         ConsFlag = 2;
     if( CLEAR == 0)
         ConsFlag = 0;
    }
return;
}
```