

```

                                aff.c
// aff.c   Programme Emetteur pour PIC16F628A   Robert Denoyel   Affichage mutiplexe
// Le PIC est en horloge interne sans Clock Out( 4 MHz)   #FUSES INTRC_IO //Internal RC Osc, no CLKOUT
// La patte MCLR est disponible pour le reset           #FUSES MCLR //Master Clear pin enabled
// Le multiplexage de l'affichage est géré par timer0 (extinction et allumage pendant 8,2ms)
// L'affichage affiche "TV"

/* Inclusions et équivalences */
#include "EmetSportTV 16F628.h"
#define selaff_gauche pin_B5 // Sélection de l'afficheur gauche
#define selaff_droit pin_B4 // Sélection de l'afficheur droit

/* Déclaration des variables globales */
int choix_affich=0; // Variable de choix d'afficheur (0 -->gauche )
int cathodes_gauche=0b00000000; // Mot de commandes des cathodes de l'afficheur gauche initialisé à éteint (1 --> allumé)
int cathodes_droit=0b00000000; // Mot de commandes des cathodes de l'afficheur droit initialisé à éteint (1 --> allumé)
int temp=0; // variable temporaire

/* Déclaration des prototypes */
void affiche(void); //pas de variable d'entrée et de sortie

/* Déclaration des fonctions */

/* Fonction affiche: affiche le contenu de cathodes_gauche ou cathodes_droit
sur l'afficheur gauche ou droit en fonction de choix_affich */
void affiche(void)
{
    if (choix_affich==1)
    {
        temp=cathodes_gauche;
        temp=temp & 0b00000001;
        if(temp!=0)output_low(Pin_A0); else output_high(Pin_A0);
        temp=cathodes_gauche;
        temp=temp & 0b00000010;
        if(temp!=0)output_low(Pin_A1); else output_high(Pin_A1);
        temp=cathodes_gauche;
        temp=temp & 0b00000100;
        if(temp!=0)output_low(Pin_A2); else output_high(Pin_A2);
        temp=cathodes_gauche;
        temp=temp & 0b00001000;
        if(temp!=0)output_low(Pin_A3); else output_high(Pin_A3);
        temp=cathodes_gauche;
        temp=temp & 0b00010000;
        if(temp!=0)output_low(Pin_A4); else output_high(Pin_A4);
        temp=cathodes_gauche;
        temp=temp & 0b00100000;
        if(temp!=0)output_low(Pin_A6); else output_high(Pin_A6);
        temp=cathodes_gauche;
        temp=temp & 0b01000000;
        if(temp!=0)output_low(Pin_B6); else output_high(Pin_B6);
        temp=cathodes_gauche;
        temp=temp & 0b10000000;
        if(temp!=0)output_low(Pin_B7); else output_high(Pin_B7);
        output_low(selaff_gauche); // validation de l'afficheur gauche
        output_high(selaff_droit); // Inhibition de l'afficheur droit
    }
    else
    {
        temp=cathodes_droit;
        temp=temp & 0b00000001;
        if(temp!=0)output_low(Pin_A0); else output_high(Pin_A0);
        temp=cathodes_droit;
        temp=temp & 0b00000010;
        if(temp!=0)output_low(Pin_A1); else output_high(Pin_A1);
        temp=cathodes_droit;
        temp=temp & 0b00000100;
        if(temp!=0)output_low(Pin_A2); else output_high(Pin_A2);
        temp=cathodes_droit;
        temp=temp & 0b00001000;
        if(temp!=0)output_low(Pin_A3); else output_high(Pin_A3);
        temp=cathodes_droit;
        temp=temp & 0b00010000;
        if(temp!=0)output_low(Pin_A4); else output_high(Pin_A4);
        temp=cathodes_droit;
        temp=temp & 0b00100000;
        if(temp!=0)output_low(Pin_A6); else output_high(Pin_A6);
        temp=cathodes_droit;
        temp=temp & 0b01000000;
        if(temp!=0)output_low(Pin_B6); else output_high(Pin_B6);
    }
}

```

aff.c

```
temp=cathodes_droit;
temp=temp & 0b10000000;
if(temp!=0)output_low(Pin_B7); else output_high(Pin_B7);
output_high(selaff_gauche); // Inhibition de l'afficheur gauche
output_low(selaff_droit); // validation de l'afficheur droit
}
}

/* déclaration du programme d'interruption de débordement du timer0 (muxliplexage de l'affichage=8,2ms) */
#include <avr/interrupt.h>
ISR(TIMER0_OVF_vect)
{
    disable_interrupts(INT_TIMER0); // inhibition des interruptions du timer0
    if (choix_affich==0)
        choix_affich=1;
    else
        choix_affich=0;
    affiche();
    enable_interrupts(INT_TIMER0); // validation des interruptions du timer0
}

/* Programme Principal */
void main()
{
    setup_timer_0(RTCC_INTERNAL,RTCC_DIV_32); //timer0 interruption au bout de 8,2ms
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);
    enable_interrupts(INT_TIMER0);
    enable_interrupts(GLOBAL);
    SET_TRIS_B(0b00001011); //E/S de PB
    SET_TRIS_A(0b10100000); //E/S de PA
    do
    {
        cathodes_gauche=0b01110000;
        cathodes_droit=0b00111110;
    }
    While( true);
}
```