

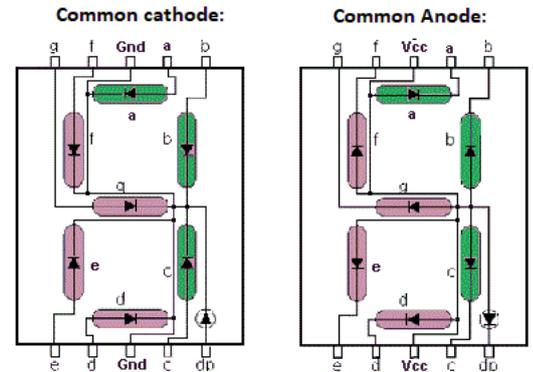
TP N°03 Les Afficheurs 7 segments

1. Objectif

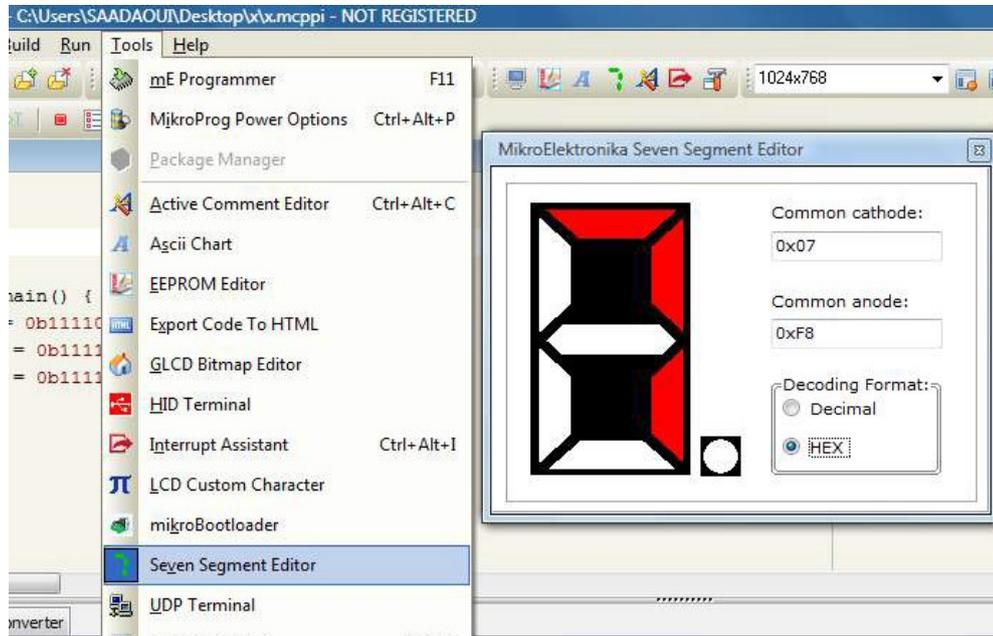
L'objectif de ce troisième TP est d'apprendre à utiliser les afficheurs 7 segments avec un microcontrôleur PIC.

2. Introduction

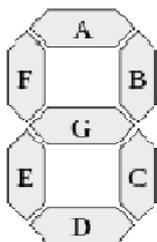
Un afficheur est constitué de 7 segments (LEDs) identifiés par les lettres de *A* à *G*. Chaque segment est rattaché à l'une des broches du μC . Il existe deux types d'afficheur, anode commune *AC* et cathode commune *CC* (noter le sens des diodes et l'alimentation des afficheurs). Pour les *AC* les diodes sont allumés par la présence d'un état *0* et pour les *CC* par un état *1*. Donc pour afficher un chiffre on doit présenter sur les broches du circuit le code 7 segments correspondant



MikroC PRO vous permet de former ce code et ce en allant dans le menu: **Tools ---> Seven Segment Editor**



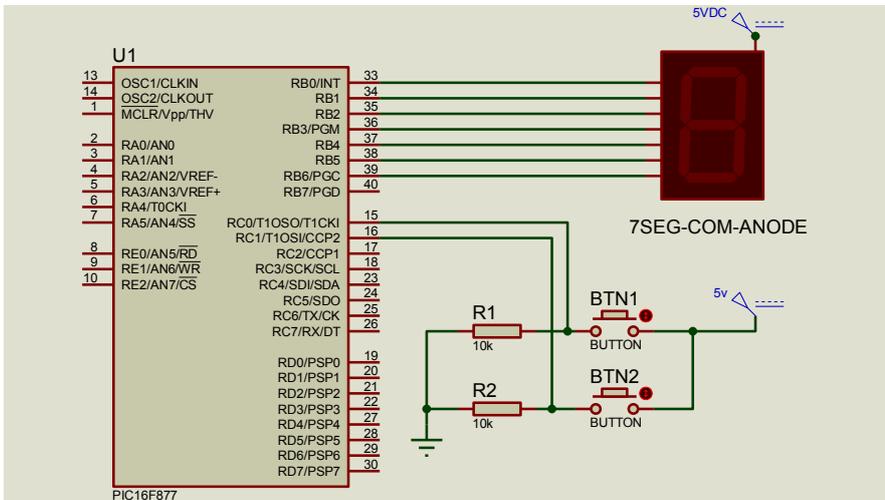
➤ **2.1.** En vous aidant de cet outil, complétez le tableau suivant pour le cas d'afficheur a anode commun :



Affichage	bit7 DP	bit6 G	bit5 F	bit4 E	bit3 D	bit2 C	bit1 B	bit0 A	Hexa
0	0	0	1	1	1	1	1	1	0x3F
1	0								
2	0								
3	0								
4	0								
5	0								
6	0								
7	0								
8	0								
9	0								

3. Travail demandé :

- **3.1.** Sous ISIS, réaliser le circuit ci-dessous liant le port B du PIC 16F877 avec un afficheur 7 segments (AC),
- **3.2.** Ecrire le programme suivant sur mikroC PRO en commentant chaque ligne, générer le fichier *.hex* et lancer la simulation.



```

void main() {
    trisb = 0b00000000 ;
    while (1)
    {
        portb = 0xC0 ;
        delay_ms (1000) ;
        portb = 0xF9 ;
        delay_ms (1000) ;
    }
}

```

- **3.2.** Expliquer Ce fonctionnement en se basant sur le code.

- **3.3.** Ecrire un code qui permet d'afficher le nombre 7.

- **3.4.** Jusqu'à présent on n'a pas utilisé les boutons.

Dans le code ci-dessous, on va utiliser les deux boutons poussoirs **BTN1** et **BTN2**. Lorsqu'on appui sur **BTN1** on affiche 6. Et lorsqu'on appui sur **BTN2** on affiche 9, sinon on affiche un 0.

```

void main() {
    TRISB = 0 ; TRISC = 0xFF ;
    PORTB = 0 ; PORTC = 0 ;

    while (1)
    {
        if ( portc.f0 == 1 )
            portb = 0x82 ;
        else if ( portc.f1 == 1 )
            portb = 0x90 ;
        else portb = 0xC0 ;
    }
}

```

- **3.5.** Ce programme ci-dessous permet d'afficher de 0 à 9 à une cadence de 500ms en déclarant un tableau tab[10] contenant les 10 codes de 7 seg des chiffre de 0 à 9 et avec une structure for repérer dans la table le code à envoyer sur le portB tout les 500 ms.

```

int i ;
char tab[10] = { 0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0x80, 0x90 } ;
void main()
{
    PORTB = 0 ; // Initialize PORTC
    TRISB = 0 ;
    while (1) {
        for ( i=0 ; i<= 9 ; i++ )
        {
            portb = tab[i] ;
            delay_ms (500) ;
        }
    }
}

```

- **3.6.** Modifier ce programme pour les nombres de 9 à 0 (décomptage) avec une cadence de 300ms.

- **3.7.** Écrire un programme qui permet d'incrémenter le chiffre affiché en appuyant sur un bouton poussoir **BTN1** et le décrémenter en appuyant sur le bouton **BTN2**.