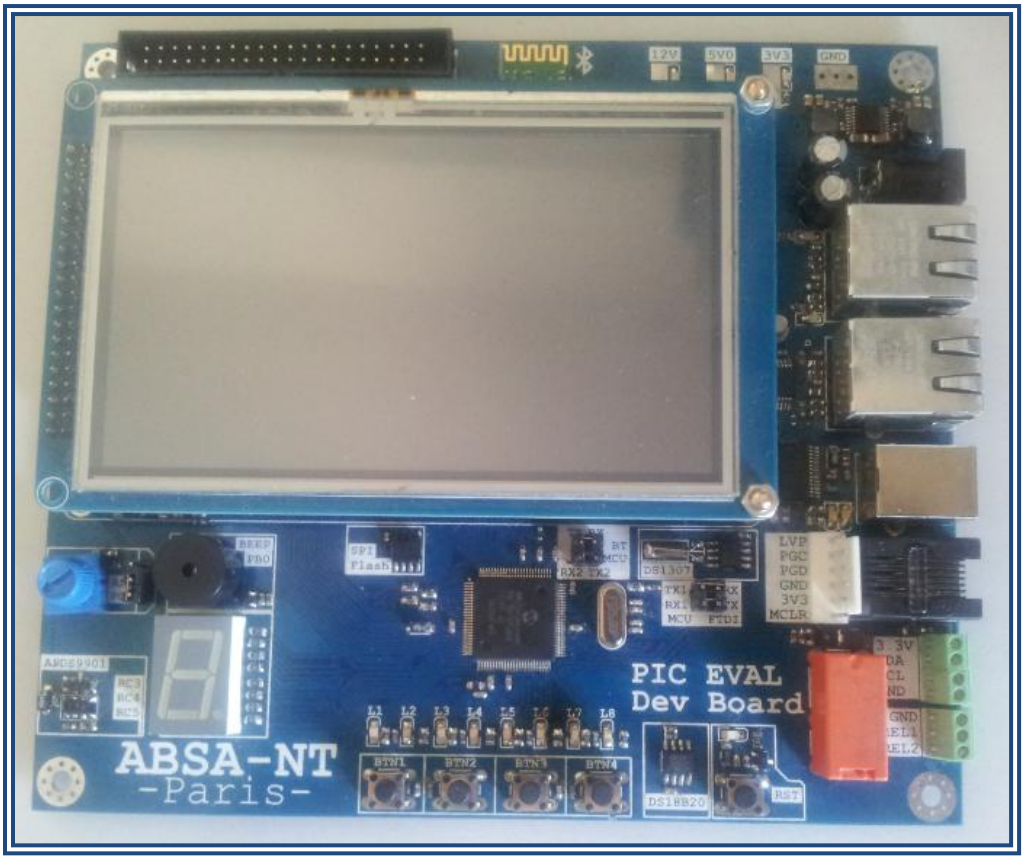
ABSA-NT
Développement
Electronique Informatique

PIC EVAL Dev Board

PIC18F97J60



TP2 :

Gestion cabine Ascenseur par plateforme PIC EVAL-ANFA

Pour répondre aux questions et justifier vos réponses, vous pouvez faire des copies d'écran ou des schémas.

I. Objectifs du TP

Le but de ces manipulations est de simuler le fonctionnement d'un ascenseur de 4 étages en utilisant les boutons poussoirs, leds, et l'écran LCD touch screen ; le but final étant de simuler la montée et descente de l'ascenseur sur 4 étages, en effet, il suffira d'appuyer sur un étage et cela avec des interruptions générées par un simple appui sur les boutons poussoirs.

Résultat final :

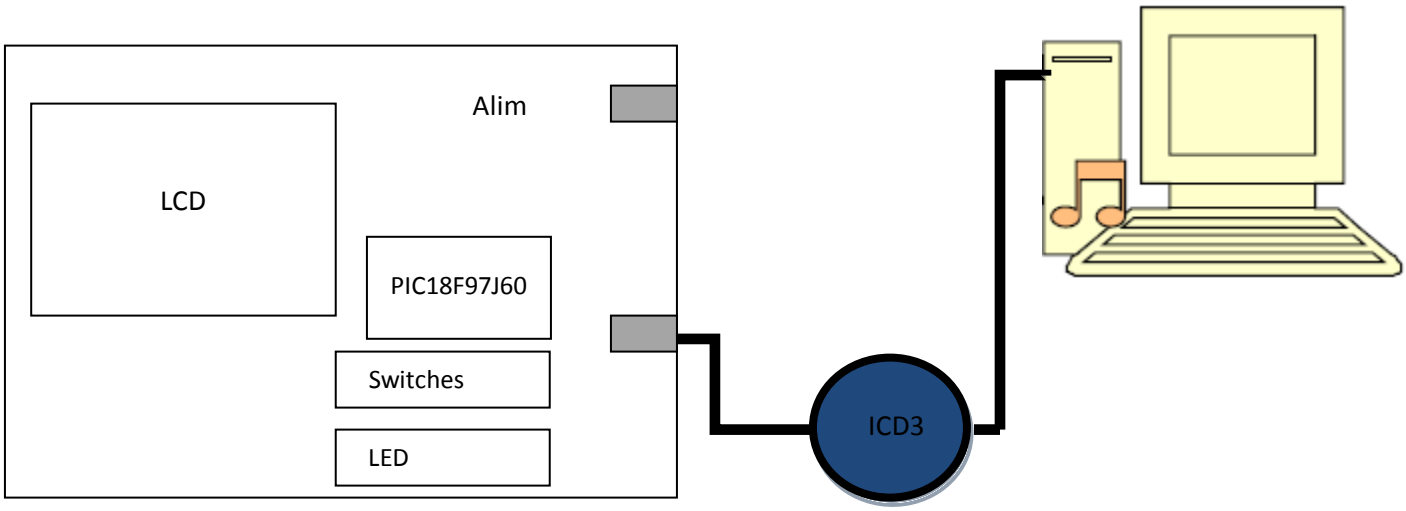
1. Matériel à votre disposition

Certains des éléments suivants sont présents sur votre plan de travail.

- Une carte Pic Eval
- Un ICD3
- Un câble USB
- Un câble d'alimentation pour la carte

2. Montage du TP

Il faut impérativement suivre les instructions du fichier 'MPLAB_Manuel.pdf' pour éviter tout dysfonctionnement.



2. Présentation de la carte PIC EVAL

La carte de développement «PIC EVAL» est un outil de développement adapté pour la programmation et l'expérimentation avec le Micro-contrôleur PIC18F97J60 de chez Microchip.

PIC EVAL intègre :

- 1 connecteur de programmation mémoire Pic.
- 1 connecteur USB pour la communication PC.
- 1 afficheur 7 segments et 8 leds.
- 1 Capteur de proximité et lumière.
- 1 module Bluetooth / Android.

- 1 microcontrôleur PIC 18F67J60.
- 1 mémoire flash 16 MB.
- 1 écran tactile touch screen.
- 1 capteur de température.
- 1 horloge temps réel RTC.
- 2 Relais.
- 1 serveur web.
- 1 liaison Ethernet Maitre.
- 1 liaison Ethernet esclave.
- 1 liaison I2C.

3. Présentation de MPLAB

Ouvrir le projet TP_Ascenseur (File > Open Project > TP_Ascenseur)

1. Présentation générale

MPLAB est un Environnement de Développement Intégré (IDE) qui permet le développement logiciel des microcontrôleurs PIC et les contrôleurs de signal numériques dsPIC de la société Microchip.

MPLAB IDE permet :
- De créer le code source à l'aide de l'éditeur intégré.
- D'assembler, compiler et lier les fichiers sources qui peuvent provenir de langages différents. Un assembleur, un "linkeur" et un gestionnaire de librairies sont fournis avec MPLAB. Un compilateur C est vendu à part par Microchip; des outils de tierces parties peuvent aussi être utilisés.
- De déboguer le code exécutable en observant le déroulement du programme à l'aide du simulateur fourni, de l'émulateur temps réel ICE 2000 ou de l'ICD2 (in circuit debugger) ou encore l'ICD3 développés par Microchip. Des outils de tierces parties peuvent aussi être utilisés.
- D'effectuer des mesures temporelles avec le simulateur ou l'émulateur.
- De voir les variables grâce à des fenêtres d'observation (watch windows).

2. Prise en main de Mplab (Optionnel)

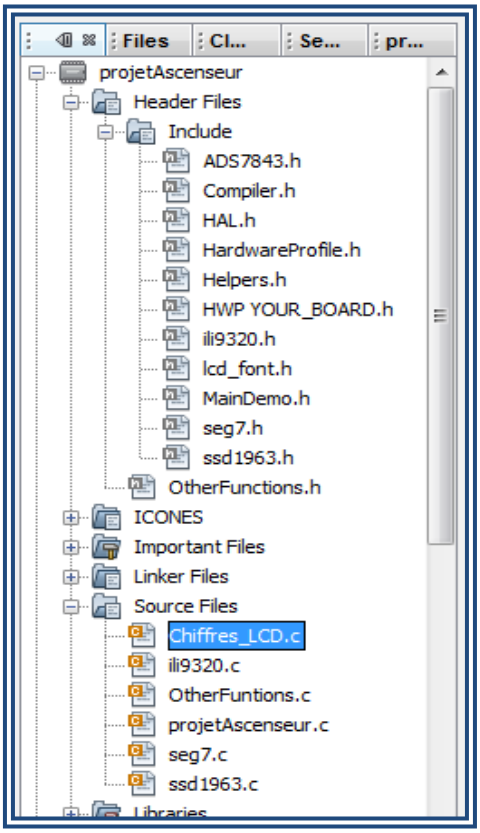
Il faut impérativement avoir fait au préalable le TP1 Prise en main de l'environnement de programmation pour la carte PIC EVAL-ANFA.

4. Projet Ascenseur

1. Écran Lcd touchscreen

Nous allons tout d'abord nous intéresser à la prise en main de l'écran LCD.

Vous avez à votre disposition les fichiers 1963.c et 1963.h où vous avez des fonctions pour commander le LCD.



Vous avez par exemple les fonctions :

void ssd1963_PutText(unsigned int x,unsigned int y,unsigned char *pString,unsigned int charColor,unsigned int bkColor) qui écrit une chaîne de caractères (*pString) sur l'écran LCD.

void ssd1963_ClearZone(unsigned int x,unsigned int y,unsigned int dX, unsigned int dY,unsigned int Color) qui dessine une zone rectangulaire de couleur Color.

2. Programme

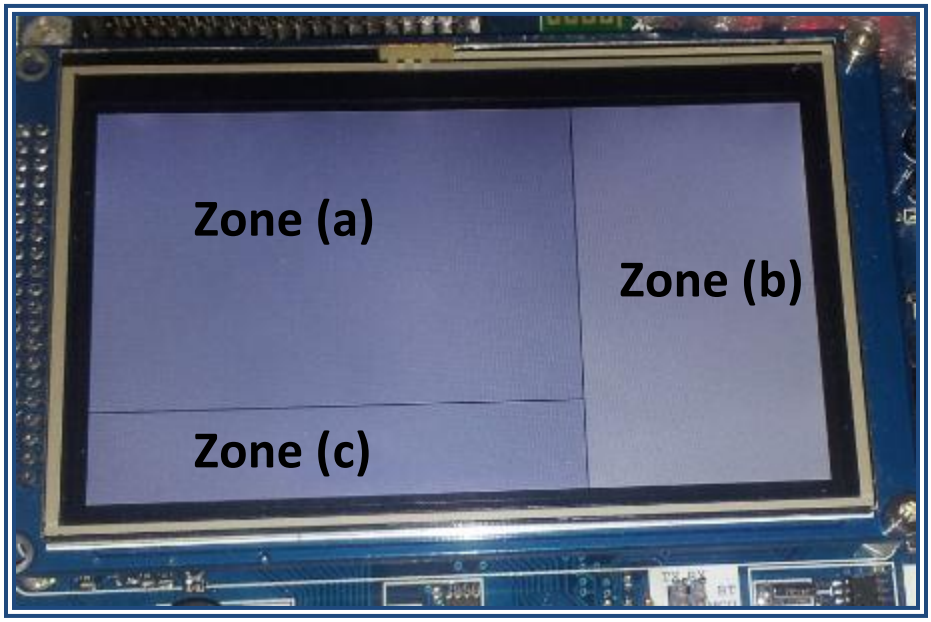
Le programme de la Pic Eval utilise l'écran LCD. Ce programme active l'écran LCD, prend la couleur gris clair et affiche au milieu 'Hello world'.

Votre fonction **main ( )** pour ce projet sera constitué de la fonction suivante :

```
void main(void)
{

   InitializeBoard();

    // Couleur de l'ecran
   ssd1963_ClearZone(0, 0, 480, 272, LightGrey);

   // Ligne affichee sur l'ecran LCD
   ssd1963_PutText(200, 120, Hello, Black, LightGrey);

   while(1)
   {
   }

}
```

Compiler le projet en entier (Run> Clean and Build Project). Lancer l'application dans la Pic Eval (Run > Run Project) et vérifier le fonctionnement.

Q1. En vous appuyant sur le programme donné, réaliser un programme qui divise l'écran en 3 parties de différentes couleurs comme la figure ci-dessous.

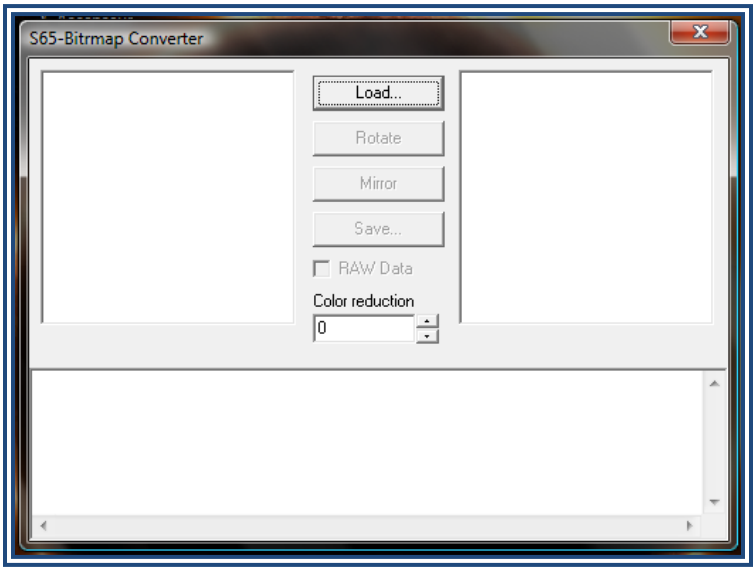


Logiciel Convert

Maintenant on va s'intéresser à l'insertion des boutons des différents étages. Pour cela vous avez le logiciel Convert. (C:\Tp_Ascenseur\ Programme_Convertisseur_IMAGE) qui convertit une image en un tableau de caractères.

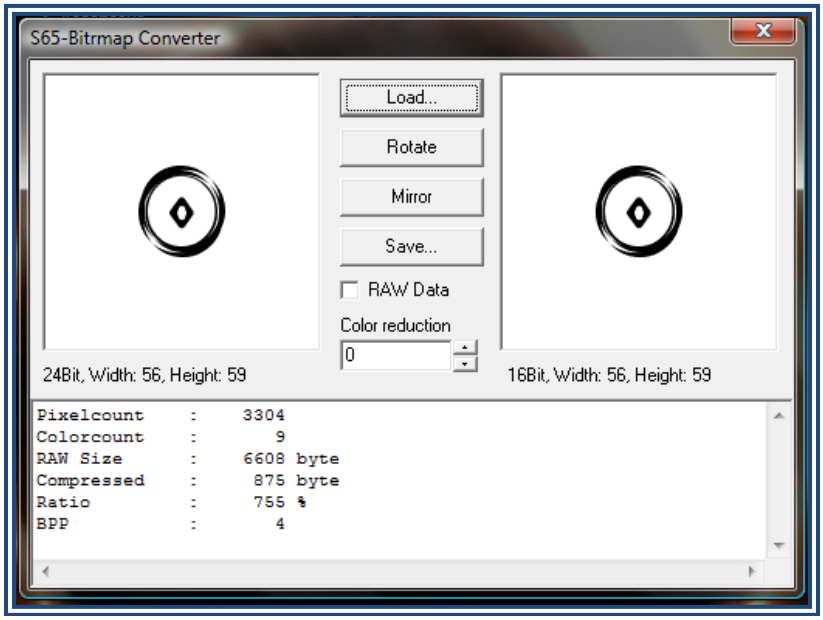
Etape 1 :

Lancez le logiciel Convert.



Etape 2 :

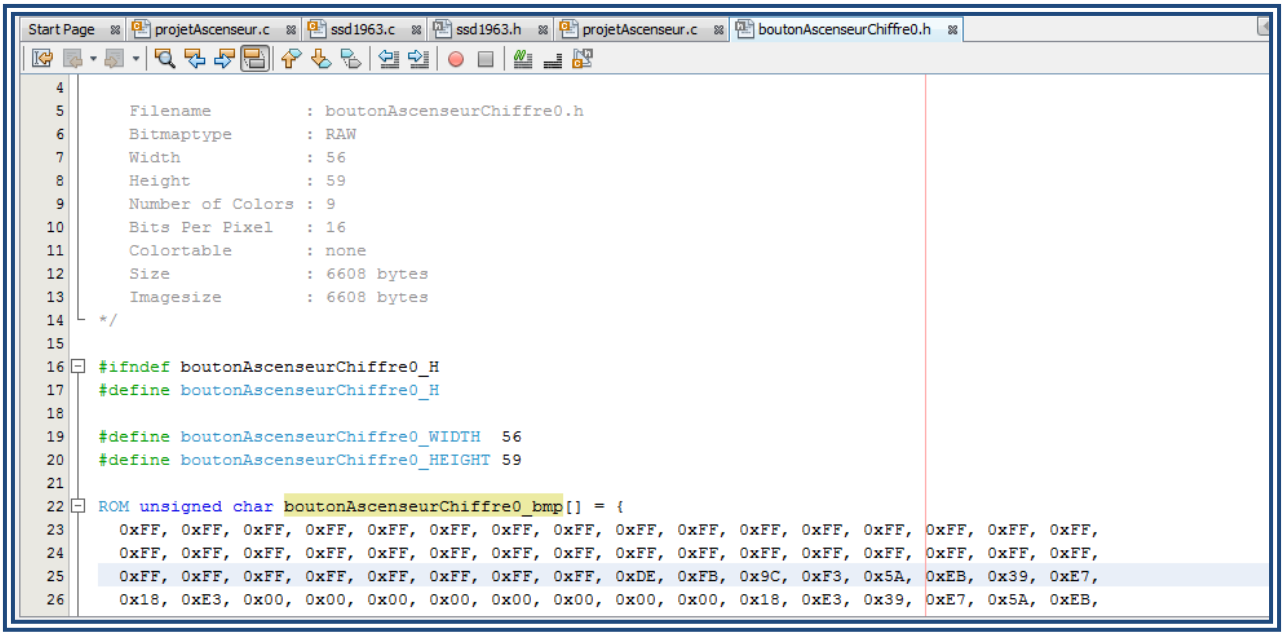
Cliquez sur ‘Load…’ pour ouvrir l’image que vous voulez convertir. Vous devez avoir l’image suivant s’afficher :



Etape 3 :

Cochez l’option ‘RAW Data’ et cliquez sur ‘Save…’, donner le nom du fichier et le répertoire où l’enregistrer.
Une fois le fichier converti donné, intégrez le dans le projet.
Pour ce faire créez un nouveau fichier (File > New File) appelez le par exemple boutonAscenseurChiffre0.h et mettez y le code donné par le logiciel Convert.

Vous aurez ceci :



Pour faire appel à ce fichier, vous avez à votre disposition la fonction :

void ssd1963_DrawPicture(unsigned int X,unsigned int Y,unsigned int size_X,unsigned int size_Y,rom unsigned char *pic) voir description dans le fichier ssd1963.c.

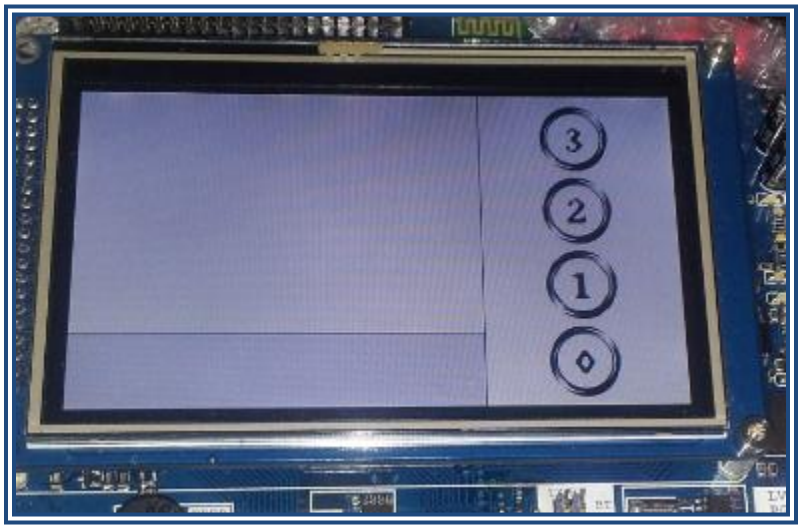
Pour notre exemple, on peut utiliser la fonction comme tel :

```
void main(void)
{

   InitializeBoard();

    // Couleur de l'ecran
   ssd1963_ClearZone(0, 0, 480, 272, LightGrey);

   // Ligne affichee sur l'ecran LCD
   ssd1963_PutText(200, 120, Hello, Black, LightGrey);

   //Affichage Bouton 0
   ssd1963_DrawPicture(370, 205, boutonAscenseurChiffre0_WIDTH,
         boutonAscenseurChiffre0 _HEIGHT, boutonAscenseurChiffre0_bmp);

   while(1)
   {
   }
}
```

Q2. Modifiez le programme en ajoutant l'affichage du bouton 0 sur l'écran LCD dans la zone (b).

Q3. En vous appuyant sur la question 1 et 2, affichez sur le coté droit de l'afficheur les 4 boutons pour les différents étages.

Résultat voulu :



3. Utilisation de la touchscreen

Nous allons maintenant utiliser la touchscreen, pour cela vous avez une fonction donnée :

```
void Decode_TS(unsigned int x, unsigned int y) // Interruption écran tactile
{
   if((x > 330)&&(x < 430))
   {

         if((y > 0)&&(y < 68)) //a modifier en fonction de vos boutons
         {
            // ZONE Bouton 3

         }
         if((y > 68)&&(y < 136)) //a modifier en fonction de vos boutons
         {
            // ZONE Bouton 2

         }
         if((y > 136)&&(y < 204)) //a modifier en fonction de vos boutons
         {
            // ZONE Bouton 1

         }
         if(y > 204) //a modifier en fonction de vos boutons
          {
            // ZONE Bouton 0

         }

   }
}
```

Q4. Dans le programme principal, créez une fonction qui, lorsqu'on appuie sur un des boutons, il reconnait bien le bouton où on a appuyé (faites allumer la LED0 pour le bouton 0 par exemple ...).

On va maintenant ajouter une zone blanche dans la zone gauche haute (zone (a)) pour afficher l'étage où l'on est.

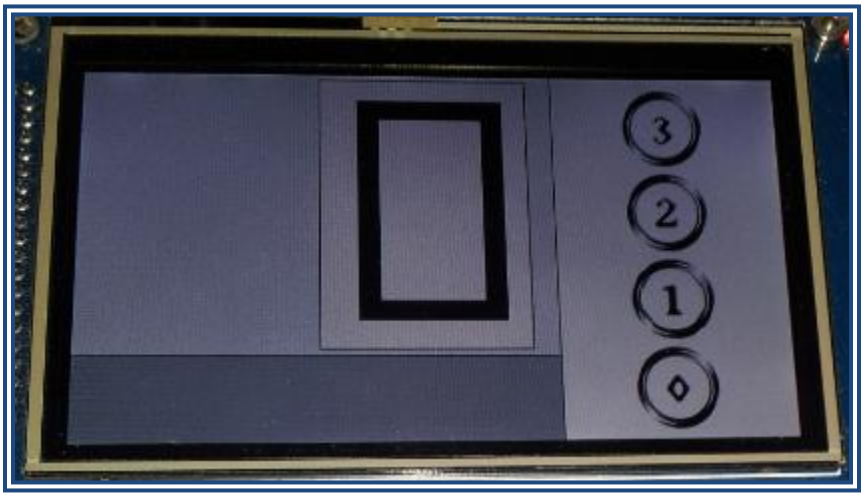
Code de la zone blanche :

```
//contour de la zone blanche ou apparaitra l étage ou on est
   ssd1963_PutLine(159,4,'X',142,Black);      // trait horizontal haut
   ssd1963_PutLine(159,205,'X',142,Black);    // trait horizontal bas
   ssd1963_PutLine(159,4,'Y',202,Black);      // trait vertical gauche
   ssd1963_PutLine(301,4,'Y',202,Black);      // trait vertical droite
   ssd1963_ClearZone(160, 5, 140, 200, White);
```

Q5. Rajoutez ce bout de code dans le programme et vérifiez le fonctionnement.

Q6. En utilisant la fonction ssd1963_ClearZone(), dessinez des chiffres de 0 jusqu'à 3, chiffres qui seront affichés dans la zone blanche de la question précédente. Créez un nouveau fichier que vous appellerez Chiffres_LCD.c et mettez y des fonctions pour chaque chiffre, par exemple number_zero() pour le chiffre zéro, number_two pour le deuxième, etc.

Exemple de chiffre :



Q7. Modifiez le programme de la question Q4 pour que lorsqu'on appuie sur un bouton, le numéro de l'étage demandé soit indiqué dans la zone blanche.

On veut maintenant simuler la montée et descente de l'ascenseur via des flèches, une flèche vers le haut pour une montée et une flèche vers le bas pour la descente.

Q8. Créer une fonction qui dessine une flèche vers le haut.

Rappel : Une flèche est un rectangle avec un triangle…

Q9. Créer une fonction qui dessine une flèche vers le bas.

Vous avez à votre disposition des fonctions qui permettent de gérer la montée ou descente d'un étage, ces fonctions sont dans le fichier OtherFuntions.c.

Pour faciliter l'insertion de ces fonctions dans le main, vous aurez à votre disposition les bouts de code à ajouter dans le fichier Question9.c

Fonction TIMER0_ISR modifiée :

```
void TIMER0_ISR(void) // Interruption Timer0
// <editor-fold defaultstate="collapsed" desc="user-description">
{
  if(INTCONbits.TMR0IF)
  {
        ////////////////////////////////////////////////////////////////////////////
    if(flagAttenteAscenseurOuvert==1)
    {
      compteurTimer0AscenseurOuvert++;
      if(compteurTimer0AscenseurOuvert==100) // Si on passe 100 fois dans l'interruption du Timer
      {
        compteurTimer0AscenseurOuvert=0;
        flagAttenteAscenseurOuvert=0;
        flagFinAscenseurOuvert=1;
      }
    }
       ////////////////////////////////////////////////////////////////////////////
    if(ascenseurOn==1)
    {
      compteurTimer0ChangementEtage++;
      if(compteurTimer0ChangementEtage==100) // Si on passe 100 fois dans l'interruption du Timer
      {
        compteurTimer0ChangementEtage=0;      // On remet à 0 pour recompter
        flagChangementEtage=1;                // On lance le flag qui simule l'arrivée à un nouvel étage
      }
    }
    else
    {
      compteurTimer0ChangementEtage=0;
    }

……
```

Q10. En vous appuyant sur la question précédente, c*réer un programme qui lorsqu'on monte il dessine la flèche vers le haut (question Q6) et pour la descente la flèche vers le bas (question Q7) à gauche de la zone blanche.*

Q11. On veut maintenant que, lors d'une montée ou descente, dans la zone du bas (zone (c)), on affiche le texte suivant : 'Prochain arret : x' où x est l'étage où l'on veut aller.

Q12. Ajoutez une image d'interdiction de fumer dans la zone du bas à droite. Pour cela utiliser le logiciel Convert. (voir parti Logiciel Convert).

Résultat voulu :