

```

/*****
This program was produced by the
CodewizardAVR V1.24.0 Standard
Automatic Program Generator
© Copyright 1998-2003 HP InfoTech s.r.l.
http://www.hpinfotech.ro
e-mail:office@hpinfotech.ro

```

```

Project : SCOPE DMX512
Version : 2
Date création : 12/02/2007
Révisions le : 20/12/07, 06/02/08
Author : Philippe Mariano
Company : LYCEE Pierre Emile MARTIN
Comments:
Réception d'une trame DMX512 et affichage
de 24 canaux consécutifs sur un LCD

```

```

Chip type      : ATmega32
Program type   : Application
Clock frequency : 16,000000 MHz
Memory model   : Small
External SRAM size : 0
Data Stack size : 512

```

```

*****/

```

```

#include <mega32.h>
#include <delay.h>
#include <stdlib.h>
#include <string.h>
#include <ssi.h>          // Sous-programme de désérialisation et de décodage de la
                          // trame issue du clavier PS2 et sous-programme Affiche_LCD
#include <tpavnum.h>      // Table de transcodage du clavier numérique utilisé
                          // Cette table devra être adapté au clavier utilisé
#include <pavenum.h>      // Gestion d'un clavier numérique type PS2

```

```

// Alphanumeric LCD Module functions
#asm
.equ __lcd_port=0x15
#endasm
#include <lcd.h>

```

```

// Caractère
#define Caractere_Vide      0x20
#define Caractere_Pulse     0
#define Caractere_FlecheH   1
#define Caractere_FlecheB   2
#define Caractere_Plein     0xFF

```

```

// Paramètres de transmission
#define RXB8 1
#define TXB8 0
#define UPE 2
#define OVR 3
#define FE 4
#define UDRE 5
#define RXC 7

#define ERREUR_TRAME (UCSRA &(1<<FE))
#define DONNEE_RECUE UDR
#define NOMBRE_CANAUUX 24
#define VRAI 1
#define FAUX 0

#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<OVR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

// Adresse CGRAM de l'afficheur LCD
// -----
#define Add_Pulse_CGRAM 0x40 // @0 Caractère pulse
#define Add_1barre_CGRAM 0x48 // @1 Caractère 1 barres
#define Add_2barres_CGRAM 0x50 // @2 Caractère 2 barres
#define Add_3barres_CGRAM 0x58 // @3 Caractère 3 barres
#define Add_4barres_CGRAM 0x60 // @4 Caractère 4 barres
#define Add_5barres_CGRAM 0x68 // @5 Caractère 5 barres
#define Add_6barres_CGRAM 0x70 // @6 Caractère 6 barres
#define Add_7barres_CGRAM 0x78 // @7 Caractère 7 barres
#define Add_FlecheH_CGRAM 0x48 // @1 Caractère flèche haut
#define Add_FlecheB_CGRAM 0x50 // @2 Caractère flèche bas

// Declare your global variables here
/*-----*/
type Désignation Commentaires
/*-----*/

// Communication avec le clavier PS2
// Le programme principal se synchronise avec la routine d'interruption située dans la bibliothèque ssi.lib grace aux deux
// variables ci-dessous
unsigned char g_key_ASCII='E'; // Contient le code ASCII de la touche pressée (à initialiser avec un
// caractère n'existant pas sur le clavier utilisé)
unsigned char g_flag_char = 1; // g_flag_char = 1 lorsqu'une touche est appuyée
// g_flag_char = 0 lorsqu'une touche a été relâchée
unsigned char g_Sens_transfert = 0; // variable globale indiquant le sens de
// communication µC <--> clavier que doit gérer la routine
d'interruption

```

```

// receptel
unsigned int g_Trame_TX;

// Paramètres utilisés en mode réception
unsigned int Table_Donnee[NOMBRE_CANAUUX] ; // Contient les valeurs des canaux reçues
unsigned int compteur_canaux = 0, canal_base = 1, nouveau_canal_base = 0, Table_canal[3]={0,0,0};
unsigned char drapeau_reception = FAUX; // Indicateur de rafraîchissement de la table des canaux reçus

enum TMenu {Menu_Da,Menu_Db,Menu_Dc,Menu_Recepteur_D11,Menu_Recepteur_D12,Menu_Recepteur_D13,Menu_Recepteur_D14,
            Emetteur,Diagnostic,Saisie_Canal_base,Aff_Bargraph,Aff_Decimal,Aff_Pourcent,Aff_Hexa,Aff_Binaire,Aff_max_min
            } Menu=Menu_Da,Menu_1=Menu_Db;// Etats du graphe tests

// Constantes en flash
//-----
// Caractères personnalisés placés dans la mémoire CGRAM du LCD
const unsigned char Pulse_Barre_Graphe[8] = {0x00, 0x0E, 0x0A, 0x0A, 0x0A, 0x0A, 0x1B, 0x00}; // @0 CGRAM
const unsigned char Barre_graphe_niveau1[8] = {}; // @1 CGRAM à compléter
const unsigned char Barre_graphe_niveau2[8] = {}; // @2 CGRAM à compléter
const unsigned char Barre_graphe_niveau3[8] = {}; // @3 CGRAM à compléter
const unsigned char Barre_graphe_niveau4[8] = {}; // @4 CGRAM à compléter
const unsigned char Barre_graphe_niveau5[8] = {}; // @5 CGRAM à compléter
const unsigned char Barre_graphe_niveau6[8] = {}; // @6 CGRAM à compléter
const unsigned char Barre_graphe_niveau7[8] = {}; // @7 CGRAM à compléter
const unsigned char flecheHaut[8] = {}; // @1 CGRAM à compléter
const unsigned char flecheBas[8] = {}; // @2 CGRAM à compléter

// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
//static unsigned int compteur_canal;
unsigned char Tampon_Donnee;
unsigned int position;

if ERREUR_TRAME {
    compteur_canaux = 0;
    Tampon_Donnee = DONNEE_RECUE; // mise à zéro de RXC
}
else {
    if ((compteur_canaux >= canal_base) && (compteur_canaux <= canal_base + NOMBRE_CANAUUX - 1)){
        position = compteur_canaux - canal_base;
        Tampon_Donnee = DONNEE_RECUE;
        Table_Donnee[position] = Tampon_Donnee;
        if (compteur_canaux == canal_base + NOMBRE_CANAUUX - 1)
            drapeau_reception = VRAI;
    }
    else
        Tampon_Donnee = DONNEE_RECUE; // mise à zéro de RXC

    compteur_canaux += 1;
}
}

```

```
}
```

```
// Standard Input/Output functions
#include <stdio.h>
```

```
void main(void)
```

```
{
// Declare your local variables here
```

```
/*-----
type          Désignation          Commentaires-----*/
```

```
char display_buffer_ligne0[17];          // Tampon ligne 0 de l'afficheur LCD
char display_buffer_ligne1[17];          // Tampon ligne 1 de l'afficheur LCD
unsigned char i;
unsigned char *chaine; // espace[]=" ";
unsigned char Saisie_Numero_Canal = VRAI, Numero_Canal_est_modifie = FAUX, pos_X = 12;
unsigned int Nb_barre = 0;
```

```
// Input/Output Ports initialization
```

```
// Port A initialization
```

```
// Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In Func7=In
// State0=T State1=T State2=T State3=T State4=T State5=T State6=T State7=T
PORTA=0x00;
DDRA=0x00;
```

```
// Port B initialization
```

```
// Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In Func7=In
// State0=T State1=T State2=T State3=T State4=T State5=T State6=T State7=T
PORTB=0x00;
DDRB=0x1f;
```

```
// Port C initialization
```

```
// Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In Func7=In
// State0=T State1=T State2=T State3=T State4=T State5=T State6=T State7=T
PORTC=0x00;
DDRC=0x00;
```

```
// Port D initialization
```

```
// Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In Func7=In
// State0=T State1=T State2=T State3=T State4=T State5=T State6=T State7=T
PORTD=0x00;
DDRD=0x00;
```

```
// Timer(s)/Counter(s) Interrupt(s) initialization
```

```
TIMSK=0x00;
```

recepte1

```
// USART initialization
// Communication Parameters: 8 Data, 2 Stop, No Parity
// USART Receiver: On
// USART Transmitter: Off
// USART Mode: Asynchronous
// USART Baud rate: 250000
UCSRA=0x00;
UCSRB=0x90;
UCSRC=0x8E;
UBRRH=0x00;
UBRRL=0x03;

// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Low level
// INT1: Off
// INT2: Off
GICR|=0x40;
MCUCR=0x00;
MCUCSR=0x00;
GIFR=0x40;

// LCD module initialization
lcd_init(16);

// Chargement initial de la mémoire CGRAM de l'afficheur avec les
// caractères personnalisés
for (i=0;i<8;i++)
{
    lcd_write_byte(Add_Pulse_CGRAM + i, Pulse_Barre_Graphe[i]); // @0 Caractère pulse
    // @1 Caractère flèche haut à compléter
    // @2 Caractère flèche bas à compléter
    // @3 Caractère 3 barres à compléter
    // @4 Caractère 4 barres à compléter
    // @5 Caractère 5 barres à compléter
    // @6 Caractère 6 barres à compléter
    // @7 Caractère 7 barres à compléter
}
// Global enable interrupts
#asm("sei")

while (1)
{
    switch (Menu)
    {
```

```
// -----
// Menus déroulants
case Menu_Da : if (Menu_1 != Menu)
{
    sprintf(display_buffer_ligne0,"[1] Recepteur");
    sprintf(display_buffer_ligne1,"[3] Emetteur");
    Affiche_LCD(display_buffer_ligne0,display_buffer_ligne1);
    lcd_gotoxy(15,0); // Affichage flecheHaut
    lcd_putchar(Caractere_FlecheH);
    lcd_gotoxy(15,1); // Affichage flecheBas
    lcd_putchar(Caractere_FlecheB);
}

if (g_flag_char == 1) // Si action sur touche clavier PS2
{
    switch(g_key_ASCII)
    {
        case '8': Menu = Menu_Dc; g_key_ASCII = 'E'; break; // touche flèche haut
        case '2': Menu = Menu_Db; g_key_ASCII = 'E'; break; // touche flèche bas
        case '1': Menu = Menu_Recepteur_D11; g_key_ASCII = 'E'; break;
        case '3': Menu = Emetteur; g_key_ASCII = 'E'; break;
        default: Menu = Menu_Da;
    }
}
Menu_1 = Menu_Da;

break;

case Menu_Db : if (Menu_1 != Menu)
{
    sprintf(display_buffer_ligne0,"[3] Emetteur");
    sprintf(display_buffer_ligne1,"[5] Diagnostic");
    Affiche_LCD(display_buffer_ligne0,display_buffer_ligne1);
    lcd_gotoxy(15,0); // Affichage flecheHaut
    lcd_putchar(Caractere_FlecheH);
    lcd_gotoxy(15,1); // Affichage flecheBas
    lcd_putchar(Caractere_FlecheB);
}

if (g_flag_char == 1) // Si action sur touche clavier PS2
{
    switch(g_key_ASCII)
    {
        case '8': Menu = Menu_Da; g_key_ASCII = 'E'; break; // touche flèche haut
        case '2': Menu = Menu_Dc; g_key_ASCII = 'E'; break; // touche flèche bas
        case '5': Menu = Diagnostic; g_key_ASCII = 'E'; break;
        case '3': Menu = Emetteur; g_key_ASCII = 'E'; break;
        default: Menu = Menu_Db;
    }
}
}
```

```

Menu_1 = Menu_Db;
break;
case Menu_Dc : if (Menu_1 != Menu)
{
    sprintf(display_buffer_ligne0,"[5] Diagnostic ");
    sprintf(display_buffer_ligne1,"[1] Recepteur ");
    Affiche_LCD(display_buffer_ligne0,display_buffer_ligne1);
    lcd_gotoxy(15,0); // Affichage flecheHaut
    lcd_putchar(Caractere_FlecheH);
    lcd_gotoxy(15,1); // Affichage flecheBas
    lcd_putchar(Caractere_FlecheB);
}

if (g_flag_char == 1) // Si action sur touche clavier PS2
{
    switch(g_key_ASCII)
    {
        case '8': Menu = Menu_Db; g_key_ASCII = 'E'; break; // touche flèche haut
        case '2': Menu = Menu_Da; g_key_ASCII = 'E'; break; // touche flèche bas
        case '5': Menu = Diagnostic; g_key_ASCII = 'E'; break;
        case '1': Menu = Menu_Recepteur_D11; g_key_ASCII = 'E'; break;
        default: Menu = Menu_Dc;
    }
}
Menu_1 = Menu_Dc;
break;
// -----
case Menu_Recepteur_D11 : if (Menu_1 != Menu)
{
    sprintf(display_buffer_ligne0,"[5] Regl. canal ");
    sprintf(display_buffer_ligne1,"[7] Bargraphe ");
    Affiche_LCD(display_buffer_ligne0,display_buffer_ligne1);
    lcd_gotoxy(15,0); // Affichage flecheHaut
    lcd_putchar(Caractere_FlecheH);
    lcd_gotoxy(15,1); // Affichage flecheBas
    lcd_putchar(Caractere_FlecheB);
}

if (g_flag_char == 1) // Si action sur touche clavier PS2
{
    switch(g_key_ASCII)
    {
        case '5': Menu = Saisie_Canal_base; g_key_ASCII = 'E'; break;
        case '7': Menu = Aff_Bargraph; g_key_ASCII = 'E'; break;
        case '8': Menu = Menu_Recepteur_D14; g_key_ASCII = 'E'; break; // touche flèche haut
        case '2': Menu = Menu_Recepteur_D12; g_key_ASCII = 'E'; break; // touche flèche bas
        default: Menu = Menu_Recepteur_D11;
    }
}

```

receptel

```
    }
  }
  Menu_1 = Menu_Recepteur_D11;
break;

case Menu_Recepteur_D12 : if (Menu_1 != Menu)
{
  sprintf(display_buffer_ligne0,"[7] Bargraphe  ");
  sprintf(display_buffer_ligne1,"[9] Max - Min  ");
  Affiche_LCD(display_buffer_ligne0,display_buffer_ligne1);
  lcd_gotoxy(15,0);      // Affichage flecheHaut
  lcd_putchar(Caractere_FlecheH);
  lcd_gotoxy(15,1);      // Affichage flecheBas
  lcd_putchar(Caractere_FlecheB);
}

if (g_flag_char == 1) // Si action sur touche clavier PS2
{
  switch(g_key_ASCII)
  {
    case '9': Menu = Aff_max_min; g_key_ASCII = 'E'; break;
    case '7': Menu = Aff_Bargraph; g_key_ASCII = 'E'; break;
    case '8': Menu = Menu_Recepteur_D11; g_key_ASCII = 'E'; break; // touche flèche haut
    case '2': Menu = Menu_Recepteur_D13; g_key_ASCII = 'E'; break; // touche flèche bas
    default: Menu = Menu_Recepteur_D12;
  }
}
Menu_1 = Menu_Recepteur_D12;
break;

case Menu_Recepteur_D13 : if (Menu_1 != Menu)
{
  sprintf(display_buffer_ligne0,"[9] Max - Min  ");
  sprintf(display_buffer_ligne1,"[0] Menu Racine ");
  Affiche_LCD(display_buffer_ligne0,display_buffer_ligne1);
  lcd_gotoxy(15,0);      // Affichage flecheHaut
  lcd_putchar(Caractere_FlecheH);
  lcd_gotoxy(15,1);      // Affichage flecheBas
  lcd_putchar(Caractere_FlecheB);
}

if (g_flag_char == 1) // Si action sur touche clavier PS2
{
  switch(g_key_ASCII)
  {
    case '9': Menu = Aff_max_min; g_key_ASCII = 'E'; break;
    case '0': Menu = Menu_Da; g_key_ASCII = 'E'; break;
    case '8': Menu = Menu_Recepteur_D12; g_key_ASCII = 'E'; break; // touche flèche haut
    case '2': Menu = Menu_Recepteur_D14; g_key_ASCII = 'E'; break; // touche flèche bas
```



```

                                receptel
                                default: Menu = Menu_Recepteur_D13;
                                }
                                }
Menu_1 = Menu_Recepteur_D13;
break;

case Menu_Recepteur_D14 : if (Menu_1 != Menu)
{
    sprintf(display_buffer_ligne0,"[0] Menu Racine ");
    sprintf(display_buffer_ligne1,"[5] Regl. canal ");
    Affiche_LCD(display_buffer_ligne0,display_buffer_ligne1);
    lcd_gotoxy(15,0);          // Affichage flecheHaut
    lcd_putchar(Caractere_FlecheH);
    lcd_gotoxy(15,1);          // Affichage flecheBas
    lcd_putchar(Caractere_FlecheB);
}

if (g_flag_char == 1) // Si action sur touche clavier PS2
{
    switch(g_key_ASCII)
    {
        case '5': Menu = Saisie_Canal_base; g_key_ASCII = 'E'; break;
        case '0': Menu = Menu_Da; g_key_ASCII = 'E'; break;
        case '8': Menu = Menu_Recepteur_D13; g_key_ASCII = 'E'; break; // touche flèche haut
        case '2': Menu = Menu_Recepteur_D11; g_key_ASCII = 'E'; break; // touche flèche bas
        default: Menu = Menu_Recepteur_D14;
    }
}
Menu_1 = Menu_Recepteur_D14;
break;

case Emetteur : if (Menu_1 != Menu)
{
    sprintf(display_buffer_ligne0,"--Emet INACTIF--");
    sprintf(display_buffer_ligne1,"[0] Menu Racine ");
    Affiche_LCD(display_buffer_ligne0,display_buffer_ligne1);
}

if (g_flag_char == 1)
{
    switch(g_key_ASCII)
    {
        //case '': g_key_ASCII = 'E'; break;
        case '0': Menu = Menu_Db; g_key_ASCII = 'E'; break;
        //case '2': Menu = Diagnostic; g_key_ASCII = 'E'; break;
        default: Menu = Emetteur;
    }
}
Menu_1 = Emetteur;

```

receptel

```
break;

case Diagnostic : if (Menu_1 != Menu)
{
    sprintf(display_buffer_ligne0,"--Diag INACTIF--");
    sprintf(display_buffer_ligne1,"[0] Menu Racine ");
    Affiche_LCD(display_buffer_ligne0,display_buffer_ligne1);
}

if (g_flag_char == 1)
{
    switch(g_key_ASCII)
    {
        //case ':': PORTB.0 = 1; g_key_ASCII = 'E'; break;
        case '0': Menu = Menu_Dc; g_key_ASCII = 'E'; break;
        //case '2': Menu = Recepteur; g_key_ASCII = 'E'; break;
        default: Menu = Diagnostic;
    }
}
Menu_1 = Diagnostic;

break;

// -----
// -----
// Début des sous-menus du mode Recepteur
// -----
// Saisie du canal de base
case Saisie_Canal_base :
    if (Menu_1 != Menu)
    {
        sprintf(display_buffer_ligne0,"Canal base= ");
        sprintf(display_buffer_ligne1,"Etr>Acq Del>Esc ");
        Affiche_LCD(display_buffer_ligne0,display_buffer_ligne1);
        for(i=12;i<15;i++)
        {
            lcd_gotoxy(i,0);
            lcd_putchar('*');
        }
        lcd_gotoxy(12,0);
        _lcd_ready();
        _lcd_write_data(displayOn3); // Curseur clignotant
    }

    if (g_flag_char == 1)
    {
        //Action sur une touche du clavier PS2
        switch (Saisie_Numero_Canal) // Saisie_Numero_Canal = VRAI à l'initialisation
        { // VRAI si numéro de canal est en cours de saisie ou doit être saisie
            Page 10
```

Table_canal[2];

```
receptel
case VRAI : if (g_key_ASCII == ' ') //Traitement touche enter
{ // Le numéro du canal saisi a été acquitté par la touche 'Enter'
  if (Numero_Canal_est_modifie == VRAI)
  { // Le numéro saisi et acquitté est différent du numéro de canal mémorisé
    // => calcul de la nouvelle valeur du canal de base (en décimal)
    nouveau_canal_base = 100*Table_canal[0] + 10*Table_canal[1] +

    if ((nouveau_canal_base > 0) && (nouveau_canal_base<489))
    { // Le numéro de canal est correct
      canal_base = nouveau_canal_base; // Mémorisation
      lcd_gotoxy(15,0); // Affichage flecheHaut
      lcd_putchar(Caractere_FlecheH);
      lcd_gotoxy(15,1); // Affichage flecheBas
      lcd_putchar(Caractere_FlecheB);
      Saisie_Numero_Canal = FAUX; // On sort de la saisie
    }
    else
    { // Le numéro du canal est faux
      for(i=12;i<15;i++)
      {
        Table_canal[i-12] = 0; // La table recevant les valeurs
        lcd_gotoxy(i,0); lcd_putchar('*');// entrées au clavier
      } // est remise à zéro
      lcd_gotoxy(12,0);pos_X=12; // Position du curseur
      _lcd_ready(); // Réinitialisation de la saisie du canal
      _lcd_write_data(displayOn3);
    }
    Numero_Canal_est_modifie = FAUX;
  }
}

else if (g_key_ASCII == '.') // Traitement touche 'Echap'
{ // On quitte la saisie du numéro du canal avec la touche 'Del'
  for(i=12;i<15;i++)
  { // La table recevant les chiffres du canal est remise à zéro
    Table_canal[i-12] = 0;
    lcd_gotoxy(i,0); lcd_putchar('*');
  }
  Numero_Canal_est_modifie = FAUX; Saisie_Numero_Canal = FAUX; pos_X=12;
}

else if ((g_key_ASCII >= 0x30) && (g_key_ASCII<=0x39)) // Traitement touches 0 à 9
{ // Saisie du numéro du canal, chiffre par chiffre jusqu'à acquittement par 'Enter'
  lcd_putchar(g_key_ASCII);Table_canal[pos_X-12] = g_key_ASCII - 0x30;
  if (pos_X<14) pos_X++; else { pos_X=12; Numero_Canal_est_modifie = VRAI;}
  lcd_gotoxy(pos_X,0);
}

g_key_ASCII = 'E';// Caractère consommé
```

```

receptel

break;

case FAUX : // Sortie de la saisie du canal
    Saisie_Numero_Canal = VRAI; Menu = Menu_Recepteur_D11; g_key_ASCII = 'E';
break;

} // fin switch Saisie_code
} // fin du if g_flag_char

Menu_1 = Saisie_Canal_base;

break;

case Aff_Bargraph :
    if (Menu_1 != Menu)
    {
        for (i=0;i<8;i++) // Caractère 1 barre à la place du caractère flèche Haut dans la CGRAM
        { // Caractère 1 barre à la place du caractère flèche Bas dans la CGRAM
            lcd_write_byte(Add_1barre_CGRAM + i, Barre_graphe_niveau1[i]); // @1 Caractère 1 barre
            lcd_write_byte(Add_2barres_CGRAM + i, Barre_graphe_niveau2[i]); // @2 Caractère 2 barres
        }
        lcd_clear();
        itoa(canal_base,chaîne); // Affichage du numéro du canal au format CDU
        if (canal_base < 10) {lcd_putchar('0');lcd_putchar('0');lcd_puts(chaîne);}
        else if ((canal_base > 9) && (canal_base < 100)) {lcd_putchar('0');lcd_puts(chaîne);}
        else lcd_puts(chaîne);
        lcd_gotoxy(0,1);lcd_putchar(Caractere_Pulse); // Affichage impulsion
    }

    if (drapeau_reception == VRAI)
    { // La table des canaux a été mise à jour
        for (i=0;i<24;i++)
        { // Affichage des canaux sous la forme d'un baregraphe // Temps d'exécution ?
            Nb_barre = (float)(Table_Donnee[i])/28.33;
            if (i<12) lcd_gotoxy(i+4,0); // Positionnement du canal sur l'afficheur
            else lcd_gotoxy(i-8,1);
            if (Nb_barre == 0) lcd_putchar(Caractere_Vide);
            else if (Nb_barre == 8) lcd_putchar(Caractere_Plein);
            else lcd_putchar(Nb_barre);
        }
        drapeau_reception = FAUX; // L'afficheur sera rafraîchi après la prochaine mise à jour de la table
    }

    if (g_flag_char == 1)
    {
        if (g_key_ASCII == '.') // Sortie du menu bargraphe par 'Echap'
        {
            Menu = Menu_Recepteur_D11; g_key_ASCII = 'E';
            for (i=0;i<8;i++) // Restauration des caractère flèche
            { // dans la CGRAM

```

```

                                receptel
                                lcd_write_byte(Add_FlecheH_CGRAM + i, flecheHaut[i]); // @1 Caractère flèche haut
                                lcd_write_byte(Add_FlecheB_CGRAM + i, flecheBas[i]); // @2 Caractère flèche bas
                                }
                            }
                    }
    Menu_1 = Aff_Bargraph;
break;

case Aff_max_min : if (Menu_1 != Menu)
{
    sprintf(display_buffer_ligne0,"Max_min");
    sprintf(display_buffer_ligne1,"Entr>Acq");
    Affiche_LCD(display_buffer_ligne0,display_buffer_ligne1);
    lcd_gotoxy(15,0);
    lcd_putchar(0);
    lcd_gotoxy(15,1);
    lcd_putchar(1);
}

if (g_flag_char == 1)
{
    switch(g_key_ASCII)
    {
        //case '.': PORTB.0 = 1; g_key_ASCII = 'E'; break;
        case '2': Menu = Aff_Bargraph; g_key_ASCII = 'E'; break;
        case '8': Menu = Saisie_Canal_base; g_key_ASCII = 'E'; break;
        default: Menu = Aff_max_min;
    }
}
Menu_1 = Aff_max_min;

break;
}
}
}

```