

File: TestSOM014D.c, Date: 04/03/2011, Time: 12:42:06

```

/*****
This program was produced by the
CodeWizardAVR V2.04.8a Standard
Automatic Program Generator
© Copyright 1998-2010 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

```

```

Project : PPE DESIGN
Version : 1
Date    : 03/03/2011
Author  : Philippe Mariano
Company : Lycée Pierre Emile Martin
Comments: Test des fonctionnalités de la carte SOM014-D

```

```

Chip type           : ATmega8535
Program type        : Application
AVR Core Clock frequency: 8,000000 MHz
Memory model         : Small
External RAM size    : 0
Data Stack size     : 128

```

```

*****/

```

```
#include <mega8535.h>

```

```
//-----

```

```
// Librairies non générées par Wizard (rajoutées manuellement)

```

```
// -----

```

```
#include <ssi2.h>           // Contient les fonctions Lire_BP, Affiche_LCD, Regdebit, Select_Seringue

```

```
#include <delay.h>          // Contient la fonction delay_ms

```

```
#include <stdio.h>          // Contient la fonction sprintf

```

```
#include "SOM014D.h"        // Contient les déclarations et les fonctions pour la gestion du module

```

```
    // SOM014D

```

```
// -----

```

```
// Déclaration pour le module SOM014-D

```

```
// -----

```

```
#define NumFichierMax      4           // Nombre de fichiers présents sur la carte SD

```

```
#define VolumeINIT          3           // Valeur initial de l'incrément de réglage du volume (0 à 7)

```

```
// -----

```

```
// Alphanumeric LCD Module functions

```

```
#asm

```

```
    .equ __lcd_port=0x15 ;PORTC

```

```
#endasm

```

```
#include <lcd.h>

```

```
// -----

```

```
// Declare your global variables here

```

```
//-----

```

File: TestSOM014D.c, Date: 04/03/2011, Time: 12:42:06

```
// type          nom          Description
//-----
enum EtatMenu{ChoixFichier, ReglageVolume, Sequence} // Etats du graphe des transitions "Menu"
Menu=ChoixFichier, Menu_1= ReglageVolume;
bit PlayStop = 0;
// -----
// =====
// Programme principal
// =====
void main(void)
{
//-----
// Declare your local variables here
//-----
// type          nom          Description
//-----
char display_buffer_ligne0[17]; // Tampon ligne 0 de l'afficheur LCD2x16
char display_buffer_ligne1[17]; // Tampon ligne 1 de l'afficheur LCD2x16
unsigned int NumFichier=0, NumFichier_1=1;
unsigned char Volume=4, Volume_1=1; // Incrément compris entre 0 et 7 pour le
// réglage du volume
unsigned char BP=0; // Bouton-poussoir actionné ou 0 ou 0xFF
unsigned char EtatSequence=0, EtatSequence_1=1;
//-----
// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=P State4=P State3=P State2=P State1=P State0=P
PORTA=0x3F;
DDRA=0x00;
//-----
// Port B initialization
// Func7=In Func6=In Func5=In Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=T State6=T State5=T State4=0 State3=0 State2=1 State1=0 State0=0
PORTB=0x06; // Impératif pour que le reset et le premier réglage s'effectue correctement !!!!!!!!!!!!!!!
DDRB=0x1F;
//-----
// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;
//-----
// LCD module initialization
lcd_init(16);
```

File: TestSOM014D.c, Date: 04/03/2011, Time: 12:42:06

```
//----- Message d'accueil -----  
lcd_clear();  
sprintf(display_buffer_ligne0, "  Test SOM014-D ");  
sprintf(display_buffer_ligne1, "  [03/03/2011]  ");  
Affiche_LCD(display_buffer_ligne0, display_buffer_ligne1);  
delay_ms(4000);  
  
//-----  
// Initialisation du module SOM014D  
//-----  
InitSOM014D();           // Mise à 1 de Reset et CLK + séquence de reset  
Som014DReglageVolume(VolumeINIT); // Initialisé à 50%  
//-----  
// Boucle principale  
//-----  
while (1)  
{  
    BP = Lire_BP();  
  
    switch (Menu)  
    {  
// Menu 1 : Sélection d'un des fichiers de la carte SD  
//-----  
        case ChoixFichier: if ((Menu_1 != Menu) || (NumFichier != NumFichier_1))  
        {  
            sprintf(display_buffer_ligne0, "[1] Fichier = %-u", NumFichier);  
            sprintf(display_buffer_ligne1, "OK+ SET- Ent>>> ");  
            Affiche_LCD(display_buffer_ligne0, display_buffer_ligne1);  
            NumFichier_1 = NumFichier;  
        }  
  
        switch (BP)  
        {  
            case OK: if (NumFichier < NumFichierMax-1) NumFichier = NumFichier + 1;  
                     else NumFichier = 0;  
            break;  
            case SET: if (NumFichier > 0) NumFichier = NumFichier - 1;  
                     else NumFichier = NumFichierMax-1;  
            break;  
            case INC: EtatSequence = 0; Menu = Sequence; break;  
            case DEC: Menu = ReglageVolume; break;  
            case ENTR: SendFrameToSom014D(NumFichier); break;  
            default: Menu = ChoixFichier;  
        }  
  
        Menu_1 = ChoixFichier;
```

```
    break;
// Réglage du volume
//-----
    case ReglageVolume: if ((Menu_1 != Menu) || (Volume != Volume_1))
    {
        sprintf(display_buffer_ligne0, "[2] Volume = %u", Volume);
        sprintf(display_buffer_ligne1, "OK+ SET- ENT>P ");
        Affiche_LCD(display_buffer_ligne0, display_buffer_ligne1);
        Volume_1 = Volume;
    }

    switch (BP)
    {
        case OK: if (Volume < VolumeMax) Volume = Volume + 1;
                 else Volume = VolumeMax;
                 Somol4DReglageVolume(Volume);

        break;
        case SET: if (Volume > 0) Volume = Volume - 1;
                  else Volume = 0;
                  Somol4DReglageVolume(Volume);

        break;
        case ENTR: SendFrameToSomol4D(PlayPause); break;
        case INC: Menu = ChoixFichier; break;
        case DEC: EtatSequence = 0; Menu = Sequence; break;
        default: Menu = ReglageVolume;
    }

    Menu_1 = ReglageVolume;

    break;
//-----
    case Sequence: if ((Menu_1 != Menu) || (EtatSequence_1 != EtatSequence))
    {
        sprintf(display_buffer_ligne0, "[3] Seq(Fich)=%u", NumFichier);
        sprintf(display_buffer_ligne1, "OK+ SET- ENT>P ");
        Affiche_LCD(display_buffer_ligne0, display_buffer_ligne1);
        EtatSequence_1 = EtatSequence;
    }

    // Début séquence

    if (PlayStop == 1)
    {
        switch(EtatSequence)
```

File: TestSOM014D.c, Date: 04/03/2011, Time: 12:42:06

```
{
    case 0 : // Dragster
        if (Busy == 1) break;
        else
        {
            delay_ms(50); // Attente après libération de Busy
            NumFichier = 0;
            SendFrameToSomol4D(NumFichier);
            delay_ms(50); // Attente avant relecture de Busy (tBusy min = 20ms)
            EtatSequence = 1;
            break;
        }
    case 1 : if (Busy == 1) break; else { delay_ms(50); NumFichier = 1; SendFrameToSomol4D(NumFichier); delay_ms(50); EtatSequence = 3; break; } // Tir
    case 2 : if (Busy == 1) break; else { delay_ms(50); NumFichier = 2; SendFrameToSomol4D(NumFichier); delay_ms(50); EtatSequence = 0; break; } // Salsa
    case 3 : if (Busy == 1) break; else { delay_ms(50); NumFichier = 3; SendFrameToSomol4D(NumFichier); delay_ms(50); EtatSequence = 2; break; } // Chien
}

switch (BP)
{
    case OK: if (Volume < VolumeMax) Volume = Volume + 1;
            else Volume = VolumeMax;
            Somol4DReglageVolume(Volume);

            break;
    case SET: if (Volume > 0) Volume = Volume - 1;
            else Volume = 0;
            Somol4DReglageVolume(Volume);

            break;
    case ENTR: if (PlayStop == 1) PlayStop = 0; else PlayStop = 1;
            SendFrameToSomol4D(PlayPause);

            break;
    case ECHAP: SendFrameToSomol4D(Stop); break;
    case INC: Menu = ReglageVolume; break;
    case DEC: Menu = ChoixFichier; break;
    default: Menu = Sequence;
}

Menu_1 = Sequence;

break;
} // fin du while
//-----
} // fin du while
```

File: TestSOM014D.c, Date: 04/03/2011, Time: 12:42:06

```
//-----  
} // fin du main  
//-----
```