

## Programme

```
/******
```

```
This program was produced by the  
CodewizardAVR V1.25.3 Evaluation  
Automatic Program Generator  
© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.  
http://www.hpinfotech.com
```

```
Project : PWM Test program  
Version : x  
Date : ??  
Author : Marc Poupi  
Company : IUT GEII  
Comments:
```

```
Chip type : ATmega8535  
Program type : Application  
Clock frequency : 16,000000 MHz  
Memory model : Small  
External SRAM size : 0  
Data Stack size : 128  
*****/
```

```
#include <mega8535.h>
```

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

```
#include <stdio.h>  
#include <delay.h>  
// Declare your global variables here
```

```
void main(void)  
{  
// Declare your local variables here  
//unsigned char i=0x20;  
//unsigned char tampon[20];  
int a;  
  
// 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=T State4=T State3=T State2=T State1=T State0=T  
PORTA=0x00;  
DDRA=0x00;  
  
// Port B 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  
PORTB=0x00;  
DDRB=0x00;  
  
// Port C initialization  
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out  
Func0=Out  
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0  
PORTC=0x00;  
DDRC=0xFF;  
  
// Port D initialization
```

### Programme

```
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out
Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTD=0x00;
DDRD =0x31; // 0b00110001;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OCO output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 16000,000 kHz
// Mode: Ph. correct PWM top=00FFh
// OC1A output: Non-Inv.
// OC1B output: Inverted
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0xB1;
TCCR1B=0x01;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x30;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: off
// INT1: off
// INT2: off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// LCD module initialization
lcd_init(16);

lcd_gotoxy(1,1);
```

```

lcd_putsf("Mise en marche");

a=0;
while (1)
{
    // Place your code here

    // faire défiler la fleche pour selectionné le mode voulut
    if (PINB.5==0 )
    {
        a++;
        if(a==1)
        {
            lcd_init(16);
            lcd_gotoxy(0,0);
            lcd_putsf("->Chrono1:");           // affichage Chronomètre 1
            lcd_gotoxy(12,0);
            lcd_putsf(",");
            lcd_gotoxy(15,0);
            lcd_putsf("s");
            lcd_gotoxy(0,1);
            lcd_putsf(" Chrono2:");         // affichage Chronomètre 2
            lcd_gotoxy(12,1);
            lcd_putsf(",");
            lcd_gotoxy(15,1);
            lcd_putsf("s");
            lcd_gotoxy(0,2);
            lcd_putsf(" vit1:");           //affichage vitesse du kart 1
            lcd_gotoxy(9,2);
            lcd_putsf(",");
            lcd_gotoxy(12,2);
            lcd_putsf("km/h");
            lcd_gotoxy(0,3);
            lcd_putsf(" vit2:");         //affichage vitesse du kart 2
            lcd_gotoxy(9,3);
            lcd_putsf(",");
            lcd_gotoxy(12,3);
            lcd_putsf("km/h");
        }
        else
        {
            if(a==2)
            {
                lcd_gotoxy(0,1);
                lcd_putsf("->");
                lcd_gotoxy(0,0);
                lcd_putsf(" ");
                lcd_gotoxy(0,2);
                lcd_putsf(" ");
                lcd_gotoxy(0,3);
                lcd_putsf(" ");
            }
            else
            {
                if(a==3)
                {
                    lcd_gotoxy(0,2);
                }
            }
        }
    }
}

```

## Programme

```

lcd_putsf("->");
lcd_gotoxy(0,0);
lcd_putsf(" ");
lcd_gotoxy(0,1);
lcd_putsf(" ");
lcd_gotoxy(0,3);
lcd_putsf(" ");
}
else
{
  if(a==4)
  {
    lcd_gotoxy(0,3);
    lcd_putsf("->");
    lcd_gotoxy(0,0);
    lcd_putsf(" ");
    lcd_gotoxy(0,2);
    lcd_putsf(" ");
    lcd_gotoxy(0,1);
    lcd_putsf(" ");
  }
  else
  {
    if(a==5) // défilement du menu pour affichage température
    {
      lcd_init(16);

      lcd_gotoxy(0,0);
      lcd_putsf(" Chrono2:"); // affichage Chronomètre 2
      lcd_gotoxy(12,0);
      lcd_putsf(",");
      lcd_gotoxy(15,0);
      lcd_putsf("s");
      lcd_gotoxy(0,1);
      lcd_putsf(" vit1:"); //affichage vitesse du kart 1
      lcd_gotoxy(9,1);
      lcd_putsf(",");
      lcd_gotoxy(12,1);
      lcd_putsf("km/h");
      lcd_gotoxy(0,2);
      lcd_putsf(" vit2:"); //affichage vitesse du kart 2
      lcd_gotoxy(9,2);
      lcd_putsf(",");
      lcd_gotoxy(12,2);
      lcd_putsf("km/h");
      lcd_gotoxy(0,3);
      lcd_putsf("->Ta:"); // affichage températures
      lcd_gotoxy(7,3);
      lcd_putsf(" Ts:");
      lcd_gotoxy(13,3);
      lcd_putsf(" \xdfc");

      a=0;
    }
  }
}
}
}

}

}

}

}

}
}
delay_ms(200); // système anti rebon

```

## Programme

```
// affichage grand afficheur
/*do
{
  if (PINB.7==1)
  {
    if(a==1)
    {
      // affichage chrono
    }
    else
    {
      if(a==2)
      {
        // affichage vitesse 1
      }
      else
      {
        if(a==3)
        {
          // affichage vitesse 2
        }
      }
    }
  }
}
}
while(PINB.7==1);

// programme a déplacer dans les différents affichages

// acquisition des valeur numérique pour affichage

    sprintf(tampon,"%d",variable tu chrono1 avant virgule 2 chiffres);
    lcd_gotoxy(0,11);
    lcd_puts(tampon);

    sprintf(tampon,"%d",variable tu chrono1 après virgule : 2 chiffres);

    lcd_gotoxy(0,14);
    lcd_puts(tampon);

    sprintf(tampon,"%d",variable tu chrono2 avant virgule : 2 chiffres);

    lcd_gotoxy(1,11);
    lcd_puts(tampon);

    sprintf(tampon,"%d",variable tu chrono2 après virgule : 2 chiffres);

    lcd_gotoxy(1,14);
    lcd_puts(tampon);

chiffres); sprintf(tampon,"%d",variable de vitesse 1 avant virgule : 2
chiffres);

    lcd_gotoxy(2,8);
    lcd_puts(tampon);

chiffres); sprintf(tampon,"%d",variable de vitesse 1 après virgule : 2
```

## Programme

```
        lcd_gotoxy(2,11);
        lcd_puts(tampon);
chiffres); sprintf(tampon,"%d",variable de vitesse 2 avant virgule : 2
        lcd_gotoxy(3,8);
        lcd_puts(tampon);
chiffres); sprintf(tampon,"%d",variable de vitesse 2 après virgule : 2
        lcd_gotoxy(3,11);
        lcd_puts(tampon);
        if(a==5)
        {
            sprintf(tampon,"%d",variable tu chrono2 avant virgule : 2 chiffres);

            lcd_gotoxy(0,11);
            lcd_puts(tampon);

            sprintf(tampon,"%d",variable tu chrono2 après virgule : 2 chiffres);

            lcd_gotoxy(0,14);
            lcd_puts(tampon);
chiffres); sprintf(tampon,"%d",variable de vitesse 1 avant virgule : 2
        lcd_gotoxy(1,8);
        lcd_puts(tampon);
chiffres); sprintf(tampon,"%d",variable de vitesse 1 après virgule : 2
        lcd_gotoxy(1,11);
        lcd_puts(tampon);
chiffres); sprintf(tampon,"%d",variable de vitesse 2 avant virgule : 2
        lcd_gotoxy(2,8);
        lcd_puts(tampon);
chiffres); sprintf(tampon,"%d",variable de vitesse 2 après virgule : 2
        lcd_gotoxy(2,11);
        lcd_puts(tampon);

        sprintf(tampon,"%d",variable de température air);

        lcd_gotoxy(3,6);
        lcd_puts(tampon);

        sprintf(tampon,"%d",variable de température sol);

        lcd_gotoxy(3,12);
        lcd_puts(tampon);
    }
*/
}
```

## Programme

}