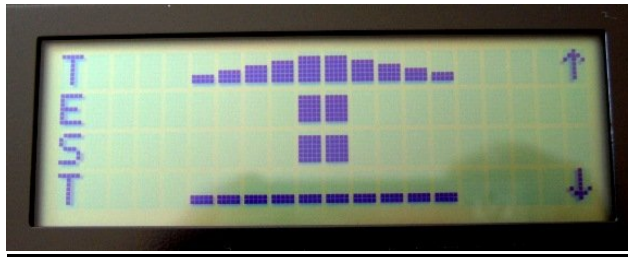


## Exemple de commande de l'afficheur LCD 4\*20



/\*\*\*\*\*

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 : Ascenseur  
Version :  
Date : 05/04/2007  
Author : Philippe Mariano  
Company : LYCEE Pierre Emile MARTIN  
Comments:

Chip type : AT90S8535  
Clock frequency : 4,000000 MHz  
Memory model : Small  
External SRAM size : 0  
Data Stack size : 128  
\*\*\*\*\*/

```
#include <90s8535.h>
```

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

```
// Adresse CGRAM de l'afficheur LCD  
// -----  
#define Add_Caract0_CGRAM 0x40  
#define Add_Caract1_CGRAM 0x48  
#define Add_Caract2_CGRAM 0x50  
#define Add_Caract3_CGRAM 0x58  
#define Add_Caract4_CGRAM 0x60  
#define Add_Caract5_CGRAM 0x68  
#define Add_Caract6_CGRAM 0x70  
#define Add_Caract7_CGRAM 0x78
```

```
// Constantes en flash  
//-----
```

```
// Caractères personnalisés placés dans la mémoire CGRAM du LCD  
const unsigned char flecheHaut[8] = {0x04, 0x0E, 0x15, 0x04, 0x04, 0x04, 0x04, 0x00}; // @0 CGRAM  
const unsigned char flecheBas[8] = {0x04, 0x04, 0x04, 0x04, 0x15, 0x0E, 0x04, 0x00}; // @1 CGRAM  
const unsigned char Symbole_Barre_Graphe[8] = {0x00, 0x0E, 0x0A, 0x0A, 0x0A, 0x0A, 0x1B, 0x00}; // @2 CGRAM  
const unsigned char Barre_graphe_niveau2[8] = {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xff, 0x00}; // @3 CGRAM  
const unsigned char Barre_graphe_niveau3[8] = {0x00, 0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0x00}; // @4 CGRAM  
const unsigned char Barre_graphe_niveau4[8] = {0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0x00, 0x00}; // @5 CGRAM  
const unsigned char Barre_graphe_niveau5[8] = {0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00}; // @6 CGRAM  
const unsigned char Barre_graphe_niveau6[8] = {0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00}; // @7 CGRAM
```

```

// Declare your global variables here

void main(void)
{
// Declare your local variables here
unsigned char i,j;

// LCD module initialization
lcd_init(20);

// Chargement de la mémoire CGRAM de l'afficheur avec les
// caractères personnalisés
for (i=0;i<8;i++)
lcd_write_byte(Add_Caract0_CGRAM + i, flecheHaut[i]);
for (i=0;i<8;i++)
lcd_write_byte(Add_Caract1_CGRAM + i, flecheBas[i]);
for (i=0;i<8;i++)
lcd_write_byte(Add_Caract2_CGRAM + i, Barre_graphe_niveau2[i]);
for (i=0;i<8;i++)
lcd_write_byte(Add_Caract3_CGRAM + i, Barre_graphe_niveau3[i]);
for (i=0;i<8;i++)
lcd_write_byte(Add_Caract4_CGRAM + i, Barre_graphe_niveau4[i]);
for (i=0;i<8;i++)
lcd_write_byte(Add_Caract5_CGRAM + i, Barre_graphe_niveau5[i]);
for (i=0;i<8;i++)
lcd_write_byte(Add_Caract6_CGRAM + i, Barre_graphe_niveau6[i]);
for (i=0;i<8;i++)
lcd_write_byte(Add_Caract7_CGRAM + i, Symbole_Barre_Graphe[i]);

lcd_gotoxy(0,0);
lcd_putchar("T");
lcd_gotoxy(0,1);
lcd_putchar("E");
lcd_gotoxy(0,2);
lcd_putchar("S");
lcd_gotoxy(0,3);
lcd_putchar("T");
lcd_gotoxy(19,0);
lcd_putchar(0);
lcd_gotoxy(19,3);
lcd_putchar(1);

j=5;
for (i=2;i<7;i++)
{
    lcd_gotoxy(j,0);
    lcd_putchar(i);
    j++;
}

j =0;
for (i=6;i>1;i--)
{
    lcd_gotoxy(j+10,0);
    lcd_putchar(i);
    j++;
};

for(i=9;i<11;i++)
{
    for (j=1;j<3;j++)
    {
        lcd_gotoxy(i,j);
        lcd_putchar(6);
    }
}

for (j=5;j<15;j++)
{
    lcd_gotoxy(j,3);
    lcd_putchar(2);
};
}

```