Université François Rabelais de Tours
Institut Universitaire de Technologie de Tours
Département Génie Électrique et Informatique Industrielle

Borne de contrôle de stop

Projet d'étude et Réalisation

VAUGUET Corentin
Groupe K4A
Promotion 2013-2014

Enseignants:
LEQUEU Thierry
AUGER Philippe

Université François Rabelais de Tours
Institut Universitaire de Technologie de Tours
Département Génie Électrique et Informatique Industrielle

Borne contrôle de stop

Projet d'étude et Réalisation

VAUGUET Corentin
Groupe K4A
Promotion 2013-2014

Enseignants:
LEQUEU Thierry
AUGER Philippe

Sommaire

Introduction.....5

1.Étude Préalable.....6

1.1.Cahier des charges 6
1.2.La borne de Stop, l'installation des boutons poussoirs et klaxon.....7
1.3.Planning Prévisionnel.....8

2.Programmation de l'ATMEGA8535 9

2.1.Menu Principal.....9
2.2.Application « Compte Tour ».....10
2.3.Application « 50m départ-arrêté ».....12
2.4.Application « Feu & Horloge ».....14
2.5.Problèmes rencontrés lors de la programmation.....15

Conclusion.....16

Index des mots clés 17

Index des illustrations.....18

ANNEXES.....19

Introduction

Dans le cadre du Téléthon, il m'a été demandé de réaliser durant le cours d'étude & réalisation, la programmation de deux bornes composées toutes deux d'un feu de carrefour, d'un écran LCD et d'un afficheur à LED à 7 segments.

Ces bornes doivent être capable de répondre à 3 sous-programmes différents:

- Le premier étant le « Compte Tours » , application qui sera utilisée lors du Téléthon.
- Le deuxième, une application « 50 mètres départ-arrêté », qui a déjà été réalisée par un groupe d'étudiants lors de la promotion 2011/2012.
- Et pour terminer , une application « Feu & Horloge ».
-

Dans ce rapport, j'expliquerai alors chaque partie du programme mais également les modifications effectuées sur les bornes.
Une vidéo de chaque application sera postée sur le site de Monsieur LEQUEU à l'adresse suivante :
http://www.thierry-lequeu.fr/

1. Étude Préalable

1.1. Cahier des charges

Ce projet est réalisé dans le cadre du Téléthon pour l'association E-Kart dont Monsieur LEQUEU Thierry est le président.

Le programme principal doit contenir une application « 50m Départ-arrêté » , qui avait déjà été conçue lors de la promotion 2011/2012 par Monsieur TOURNILLON Clément et BRAIN Anthony. Hors le projet doit aussi contenir les applications « Compte Tours » et « Feu & Horloge » qui n'ont pas encore été réalisées.

Nous aurons alors différentes modifications à réaliser sur le câblage de la borne :

- L'ajout de 3 boutons poussoirs permettant de choisir un programme.
- L'ajout d'un klaxon pour le programme « Feu & Horloge ».
- La modification du câblage de l'afficheur à LED d'un des afficheurs.

1.2. La borne de Stop, l'installation des boutons poussoirs et klaxon

La borne de stop est composée de 3 parties :

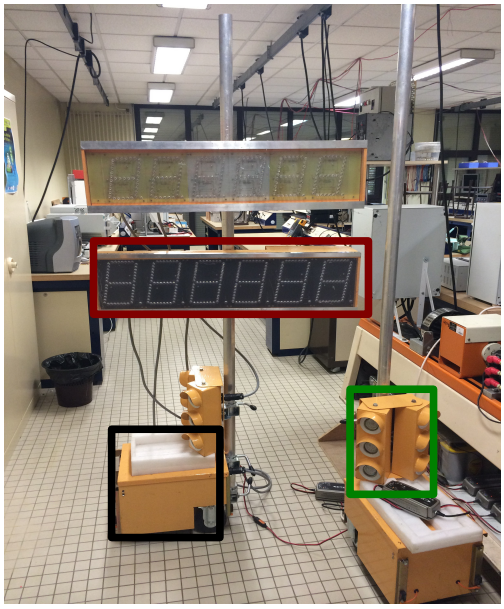

Illustration 1: Borne

- En **noir,** le bloc d'alimentation qui permet de fournir l'alimentation au niveau du bloc afficheur et des feux.

- En **rouge**, le bloc afficheur qui présente l'afficheur à LED et l'afficheur LCD.

- En **vert**, les feux de carrefour.

Pour installer les boutons poussoir et le klaxon, il fallait tout d'abord se référencer au schéma électrique de la carte d'une borne afin de trouver des entrées et sorties disponibles.

Les boutons poussoirs seront installés au dessus de l'écran LCD et le klaxon se trouvera dans le bloc d'alimentation.

1.3. Planning Prévisionnel

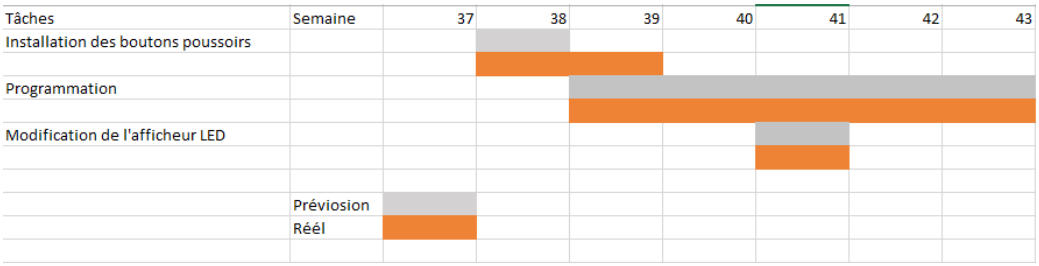

Illustration 2: Planning

Le projet se concentre exclusivement sur la programmation, il fallait néanmoins consacrer du temps à l'installation des boutons poussoirs.

De plus, l'un des afficheurs d'une borne n'était pas câblé de la même façon que l'autre borne au niveau des sorties de l'ATMEGA8535.

Il fallait alors deux programmes différents pour faire fonctionner les deux bornes.

En modifiant le câblage, il n'y a alors qu'un seul programme.

2. Programmation de l'ATMEGA8535

Pour programmer l'ATMEGA8535, le micro-contrôleur qui es utilisé ici ; nous utilisons le logiciel CodeVision AVR.

Ce logiciel permet d'écrire le programme, de le compiler et de l'enregistrer sur la mémoire de l'ATMEGA.

Dans cette partie, nous allons expliquer chaque parties du programme. Dans la suite de l'analyse , les boutons poussoirs seront nommés BPX.

2.1. Menu Principal

Lors de la mise sous tension de la borne, l'écran LCD s'allume et présente le menu suivant qui permet alors à l'utilisateur de choisir l'application à démarrer.

Illustration 3: Choix Menu

L'action sur BP1, BP2 et BP3 permet à chacun de réaliser une action différente.

Ici, l'action sur BP1 permettra de choisir «Compte Tour » ; l'action sur BP2, de choisir « 50m départ-arrêté » et l'action sur le BP3 de choisir « Feu & Horloge ». Les réglages des différents paramètres de chaque application sont établis dans le main().

2.2. Application « Compte Tour »

Pour accéder à cette application, l'utilisateur a, au préalable lors de la sélection de l'application à lancer, appuyé sur BP1. Hors, il doit aussi régler les paramètres du compte tours avant de lancer l'application.

Après avoir appuyé sur BP1, il apparaît alors les indications suivantes sur l'écran LCD :

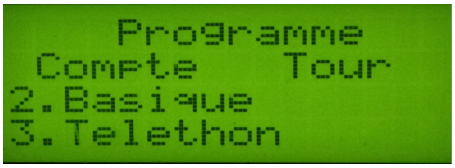

Illustration 4: Choix de l'application Compte Tour à démarrer

Appuyer sur BP2 lancera l'application Compte Tour basique, c'est à dire que le compteur s'incrémentera alors de 1 lorsque qu'un kart passera devant les capteurs de la borne.

Appuyer sur BP3 lancera l'application Compte Tours Téléthon, c'est à dire que lorsqu'un kart se présentera devant les capteurs , le compteur s'incrémentera alors de 5, ce qui correspondra à 0,50€.

Après avoir choisi ce premier paramètre, il ne reste alors qu'un seul paramètre à établir, il s'agit de la valeur du compteur.

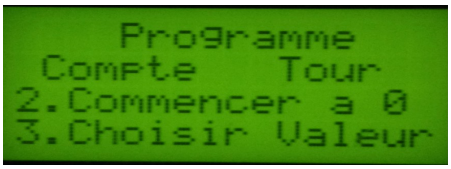

Illustration 5: Choix de la valeur du compteur

Cet affichage est commun si l'utilisateur a choisi le programme basique ou le programme Téléthon.

Si l'utilisateur appuie sur le bouton BP2, la valeur du compteur est directement mise à 0 et l'application « Compte Tours » démarre directement.

Si l'utilisateur appuie sur le BP3, il permet alors de régler la valeur du compteur avant démarrer l'application.

Illustration 6: Réglage de la valeur du compteur " Basique"

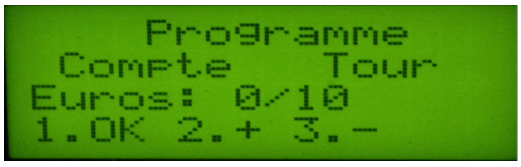

Illustration 7: Réglage de la valeur du compteur "Téléthon"

Pour régler la valeur du compteur, il faut utiliser les BP2 et BP3.

L'action sur BP2 permettra d'incrémenter la valeur du compteur de 1 si le paramètre basique à été sélectionné ou bien de 5, si le paramètre « Téléthon » a été sélectionné.

L'action sur BP3 permettra de réaliser la réciproque de l'action sur BP2.

Lorsque l'utilisateur a fini de régler la valeur de son compteur, il doit alors appuyer sur BP1 pour valider la valeur du compteur.

Tous les paramètres sont alors réglés, l'application peut alors démarrer.

Au niveau de l'afficheur , un point sera allumé pour réaliser la virgule pour l'affichage des euros.

Illustration 8: Affichage Tours

Illustration 9: Affichage des Euros

2.3. Application « 50m départ-arrêté »

Cette partie du programme avait déjà été conçue par deux étudiants, TOURNILLON Clément et BRAIN Anthony dont l'étude théorique a déjà été réalisée. Pour obtenir de plus amples informations, il est possible d'obtenir le rapport de leur projet à l'adresse suivante : http://www.thierry-lequeu.fr/data/RAP-BRAIN-TOURNILLON.pdf.

Les deux bornes communiquent ensemble à l'aide d'une liaison série RS232.

Illustration 10: Choix du type de borne

Après avoir appuyé sur BP2, l'utilisateur doit choisir s'il s'agit de la borne d'arrivée ou s'il s'agit de la borne de départ.

Si l'utilisateur appuie sur BP2, il configure alors celle-ci en borne de départ et en borne d'arrivée s'il a appuyé sur BP3.

Lorsque la borne de départ est configurée, elle envoie directement par la liaison série, un caractère permettant alors de l'initialiser.

Les bornes peuvent s'échanger différents caractères.

Source	Borne Départ
Caractère	Signification
C	Kart présent à la borne de Départ
P	Kart Présent à la borne de départ et 10s non écoulé.
A	Faux départ
O	Caractère permettant à la borne d'arrivée d'arrêté le décompte des 10s
M	Kart parti

Source	Borne d'arrivée
Caractère	Signification
D	Borne prête pour l'épreuve
X	Épreuve terminée

Dans la suite de l'analyse, la borne de départ sera nommée **Brn1** et la borne d'arrivée **Brn2**.

Au départ, les feux sont rouges, le kart se présente à la **Brn1**, le capteur le détecte et permet alors d'envoyer le caractère **'C'**.

La **Brn2** envoie alors le caractère **'D'**.

Les feux restent rouges mais en les feux oranges clignotent en même temps. La **Brn1** envoie alors **'P'** ce qui permet de lancer le décompte du temps et à la **Brn2**, de l'afficher.

Si le kart se retire, la **Brn1** envoie directement **'A'**, et on affiche le message « **FAUH** ». Si le décompte à la **Brn1** est terminé, on envoie alors **'O'** pour stopper le décompte à la **Brn2** tant que le kart n'est pas parti.

Une fois parti, la **Brn1** envoie alors **'M'** et on mesure le temps. Puis quand le kart arrive à la **Brn2**, les capteurs le détectent, ce qui met un terme à la mesure du temps.

On affiche alors le temps écoulé durant 2 secondes, puis la vitesse durant le même laps de temps et cela cinq fois de suite.

2.4. Application « Feu & Horloge »

Lorsque que l'utilisateur appuie sur BP3, nous rentrons dans la configuration des paramètres de l'heure. Nous devons alors définir les heures, allant de 0 à 24 , et les minutes de 0 à 60.

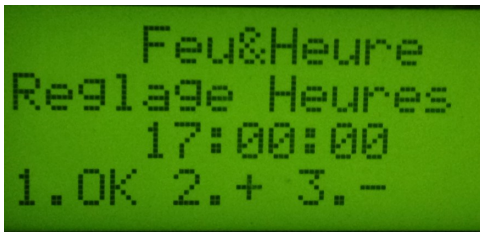

Illustration 11: Réglage de l'heure

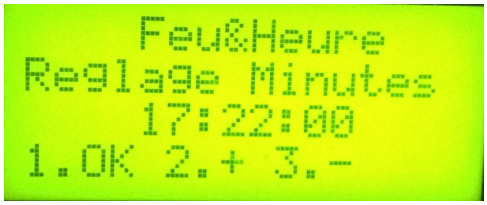

Illustration 12: Réglage des minutes

Tout d'abord, l'utilisateur doit régler les heures à l'aide de BP2 et BP3 qui permettent d'augmenter et de diminuer la valeur.

Une fois les heures réglées, nous validons la valeur avec le BP1, il faut ensuite effectuer la même procédure pour régler les minutes. Une fois la configuration de l'horloge effectuée, nous affichons l'heure sur l'afficheur à LED et le feu rouge apparaît.

Illustration 13: Affichage de l'heure

Dès qu'un kart se présentera devant la borne, l'affichage de l'heure s'arrêtera et passera à un décompte de 3 secondes à gauche ou à droite de l'afficheur, selon le côté duquel le kart se présentera.

Si l'utilisateur passe alors que le décompte n'est pas terminé, le klaxon sonne et le feu passe au orange pendant ½ seconde. Si l'utilisateur attend que le décompte se termine, le feu passe alors au vert, le kart peut alors partir. L'affichage de l'heure se remettra directement après.

2.5. Problèmes rencontrés lors de la programmation

Lors de la programmation, nous avons rencontré un problème pour afficher un nombre réel sur l'écran LCD, pour l'affichage des Euros. La solution de facilité, par manque de temps ? était d'incrémenter le compteur de 5, et d'afficher sur l'écran LCD : « Euros : 5/10 € », ce qui correspond à 50 centimes d'Euros.

Une autre solution aurait pu être mise en place, qui aurait été de toujours incrémenter la valeur du compteur de 5, de ranger cette valeur dans une chaîne et de couper cette chaîne en deux parties. Nous aurions alors converti ces deux chaînes en deux nombres entiers et écris dans une nouvelle chaîne « %d , %d €, nombre1,nombre2 ». %d correspondant à l'affichage d'un nombre entier dans une chaîne.

Conclusion

Durant ce projet, j'ai pu découvrir le logiciel CodeVisionAVR afin de programmer l'ATMEGA8535. Ce projet m'a, dans un premier temps, permit de revoir les quelques notions de langage C que nous avons pu avoir lors de notre scolarité à l'IUT, d'analyser et de corriger des erreurs qui parfois peuvent être compliquées à trouver mais dans un second temps, de comprendre le principe et le rôle d'un employé dans une entreprise lorsqu'une mission lui est confiée avec un délais à respecter. C'est un projet dans lequel je me suis beaucoup investi que cela soit durant les cours d'Étude & Réalisation ou bien chez moi, en continuant de chercher à améliorer le programme pour qu'il soit fonctionnel avant Novembre 2013.

C'est un objectif que j'ai réussi à atteindre avec beaucoup de mal et de patience mais au final, la fierté d'avoir terminé dans les temps impartis le travail qui 'a été donné.

C'est pourquoi, je souhaite remercier Monsieur LEQUEU de m'avoir proposé ce sujet et de m'avoir apporté son aide.

Index des mots clés

ATMEGA 8535
Capteurs
Entrées/Sorties
Feux de carrefour
Borne de stop
Écran LCD
Code Vision AVR
Klaxon
Boutons poussoirs(BPx)
Téléthon
50m départ-arrêté
Compte Tours
Feu&Horloge

Index des illustrations

Illustration 1: Borne....7
Illustration 2: Planning 8
Illustration 3: Choix Menu....9
Illustration 4: Choix de l'application Compte Tour à démarrer....10
Illustration 5: Choix de la valeur du compteur....10
Illustration 6: Réglage de la valeur du compteur " Basique"....10
Illustration 7: Réglage de la valeur du compteur "Téléthon"....10
Illustration 8: Affichage Tours....11
Illustration 9: Affichage des Euros....11
Illustration 10: Choix du type de borne....12
Illustration 11: Réglage de l'heure....14
Illustration 12: Réglage des minutes....14
Illustration 13: Affichage de l'heure....14

ANNEXES

File: test.c, Date: 31/10/2013, Time: 15:50:42

```
/*****************************************************
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 : Test-LCD
Version : 1
Date    : 15/02/2007
Author  : Freeware, for evaluation and non-commercial use only
Company : Thierry
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>
#include <stdio.h>
#include <delay.h>
#include <stdlib.h>
#include <math.h>

/* the LCD module is connected to PORTC */
#asm
    .equ __lcd_port=0x15
#endasm

/* now you can include the LCD Functions */
#include <lcd.h>

#define BP1    PINB.7
#define BP2    PINB.6
#define BP3    PINB.5

#define CAPT4 PIND.2
#define CAPT3 PIND.3
#define CAPT2 PIND.4
#define CAPT1 PIND.5
#define ENABLE PORTD.7


// Declare your global variables here
void ConfigATMEGA(void);
void ClrAff(void);
void EcrireLCD(int x, int y);
void Programme1(void);
void Programme2(void);
void Programme3(void);
void Programme4(void);
void Programme5(void);
void afficheur1(unsigned char adresse,unsigned char caractere, unsigned char point);
void afficheur2(unsigned char adresse,unsigned char caractere, unsigned char point);
void USART_Transmit( unsigned char data );
unsigned char USART_Receive( void ) ;
void ChoixMinutes();
void ChoixHeures();

unsigned char Texte[20];
char choix1,choix2,choix3,choix4,choix5,choix6,d,var,var1,var2,var3,d1,d2,d3,d4,d5,d6;
char carRecu='X';char car='X';
char centiemeV=0; signed char secondeV=0; char dixiemeV=0; char dsecondeV=0;
float vitesse,TempsK;
char Vsecondes;
char Vminutes;
char Vheures;
int compteur=0; int Euros=0;
char Decompte=0;
char Comptage=0;
int DecimalV,temps,UniteV,DizaineV,CentaineV;
int x=0;
char ConfigHorloge=0;

flash const unsigned char valeur_constant[] =
{0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F,0x00}; // afficheur fond noir
flash const unsigned char adresse_constant[16] =
{0,8,4,12,2,10,6,14,1,9,5,13,3,11,7,15};

void main(void)
{
```

File: test.c, Date: 31/10/2013, Time: 15:50:42

```
#asm("sei")       // Activation des fonctions d'intérruptions.

ConfigATMEGA(); // Configuration de l'ATMEGA
ClrAff(); // Eteindre tout les digits de l'afficheur
lcd_init(16); // Initialition LCD 4x16 lignes
lcd_clear();
sprintf(Texte,"Choix Programme --'");  EcrireLCD(0,0);
sprintf(Texte,"1.CompteTour--'");      EcrireLCD(0,1);
sprintf(Texte,"2.50m--'");             EcrireLCD(0,2);
sprintf(Texte,"3.Feu&Horloge--'");       EcrireLCD(0,3);

while (1)
    {
    if ((BP1==0)&(choix3==0))
        {
        ENABLE=1;
        delay_ms(500);
        lcd_clear();
        sprintf(Texte,"    Programme    Compte   Tour"); EcrireLCD(0,0);
        choix5=1;
        do
            {
            sprintf(Texte,"2.Basique");  EcrireLCD(0,2);
            sprintf(Texte,"3.Telethon"); EcrireLCD(0,3);
            if(BP2==0)
                {
                    delay_ms(500);
                    choix6=0;
                    choix5=0;
                }
            if(BP3==0)
                {
                    delay_ms(500);
                    choix6=1;
                    choix5=0;
                }
            }
        while(choix5==1);
        sprintf(Texte,"2.Commencer a 0");               EcrireLCD(0,2);
        sprintf(Texte,"3.Choisir Valeur");              EcrireLCD(0,3);
        choix1=1;
        do
            {
            if(BP2==0)                                                          // Si BP2 , compteur
passe à 0 , lancement du programme
                {
                    delay_ms(500);
                    lcd_clear();
                    sprintf(Texte,"    Programme    Compte   Tour"); EcrireLCD(0,0);
                    compteur=0;
                    choix1=0;
                }
                if(BP3==0)                                                      // Si BP3 , choisir
Valeur du compteur avant de lancer
                {
                    delay_ms(500);
                    lcd_clear();
                    choix2=1;
                    sprintf(Texte,"    Programme    Compte   Tour"); EcrireLCD(0,0);
                    do
                    {
                    if(choix6==0)
                        {
                    sprintf(Texte,"Valeur : %5d",compteur); EcrireLCD(0,2);
                    sprintf(Texte,"1.OK 2.+ 3.-");          EcrireLCD(0,3);
                        }
                    else
                        {

                        sprintf(Texte,"Euros: %d/10    ",Euros); EcrireLCD(0,2);
                        sprintf(Texte,"1.OK 2.+ 3.-");          EcrireLCD(0,3);
                        }
                    if(BP2==0)                                                  // Incrémentation
Compteur
                        {
                        delay_ms(200);
                        compteur++;
                        if(choix6==1)
                        {
                        Euros=compteur*5;
                        }
                        }
                    if (BP3==0)                                                 // Décrémentation
compteur
                        {
                        if(compteur==0)
```

```
                        {
                        compteur=0;
                        }
                    else
                        {
                        compteur--;

                        }
                    delay_ms(200);
                    if(choix6==1)
                        {
                    Euros=compteur*5;
                        }
                    }
                if(BP1==0)                                                          // Valider Valeur
                    {

                    lcd_clear();
                    sprintf(Texte,"    Programme    Compte   Tour"); EcrireLCD(0,0);
                    if(choix6==1)
                        {
                        compteur=(Euros*2);
                        }
                    choix1=0;
                    choix2=0;
                    }


                }
            while(choix2==1);
            }
        }
    while(choix1==1);
    Programme1();
    }
if((BP2==0)&(choix1==0))
    {
    ENABLE=1;
    delay_ms(500);
    lcd_clear();
    choix3=1;
    delay_ms(500);
    sprintf(Texte,"    Programme          50m");  EcrireLCD(0,0);
    sprintf(Texte,"2.Borne Depart");              EcrireLCD(0,2);
    sprintf(Texte,"3.Borne Arrivee");             EcrireLCD(0,3);

    do
        {
        if(BP2==0)
            {
            delay_ms(500);
            lcd_clear();
            sprintf(Texte,"    Programme          50m");  EcrireLCD(0,0);
            sprintf(Texte,"Borne Depart");   EcrireLCD(0,2);
            choix4=0;
            choix3=0;
            }
        if(BP3==0)
            {
            delay_ms(500);
            lcd_clear();
            sprintf(Texte,"    Programme          50m");  EcrireLCD(0,0);
            sprintf(Texte,"Borne Arrive");   EcrireLCD(0,2);
            choix4=1;
            choix3=0;
            }
        }
    while(choix3==1);
    Programme2();
    }
if((BP3==0)&(choix2==0)&(choix3==0))
    {
    ENABLE=1;
    delay_ms(500);
    lcd_clear();
    sprintf(Texte,"    Programme      Feu & Heure"); EcrireLCD(0,0);
    if(ConfigHorloge==0)
        {
        Vheures=0;
    Vminutes=0;
        Vsecondes=0;
        delay_ms(500);
        ChoixHeures();
        delay_ms(500);
        ChoixMinutes();
        }
    Programme3();

    }
```

```
    }
}
void ConfigATMEGA(void)           // CONFIGURATION DE L'ATMEGA
{
// Input/Output Ports initialization
// Port A 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
PORTA=0x00;
DDRA=0xFF;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=P State6=P State5=P State4=0 State3=0 State2=0 State1=0 State0=0
PORTB=0xE0;
DDRB=0x1F;


// 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
// Func7=Out Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=0 State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
ENABLE=1;
DDRD=0x80;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 250,000 kHz
// Mode: Normal top=FFh
// OC0 output: Toggle on compare match
TCCR0=0x08;
TCNT0=0x00;
OCR0=0xFA;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 2000,000 kHz
// Mode: CTC top=OCR1A
// OC1A output: Toggle
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: On
// Compare B Match Interrupt: Off
TCCR1A=0x40;
TCCR1B=0x0A;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x4E; // Base de temps = 2 MHz, soit 0,5 us.
OCR1AL=0x20; // Interruption quand on arrive à 20 000 (0x4E20 soit 10 ms)
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=0x10;

// Analog Comparator initialization
// Analog Comparator: Off
```

File: test.c, Date: 31/10/2013, Time: 15:50:42

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


// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x67;

}
void ClrAff()                     // Fonction permettant d'éteindre tout les Digits de l'afficheur
{
afficheur2(0,0,0);
afficheur2(1,0,0);
afficheur2(2,0,0);
afficheur2(3,0,0);
afficheur2(4,0,0);
afficheur2(5,0,0);
afficheur2(6,0,0);
}
void EcrireLCD(int x, int y)                    // Fonction permettant d'écrire le texte prédéfinie préalablement
en position X -> Colonnes de 0 à 15, Y lignes allant de  0 à 3

{
lcd_gotoxy(x,y);
lcd_puts(Texte);
}
void afficheur1(unsigned char adresse,unsigned char caractere, unsigned char point)
{
  if (point == 1)
     {

       PORTA=(valeur_constant[caractere] | 0x80); // Un caractère avec le point !      // fond noir
     }
     else
     {
     PORTA=valeur_constant[caractere]; // Un caractère
     }

  PORTB=(0b11110000 | adresse_constant[adresse & 0x0F]); // PB7 PB6 PB5 CS A0 A1 A2 A3
  PORTB.4=1;
  PORTB.4=0;  // CS = 0
  PORTB.4=1;
  PORTA=0x00;
}
void afficheur2(unsigned char adresse,unsigned char caractere, unsigned char point)
{
  if (point == 1)
     {

      PORTA=(caractere | 0x80); // Un caractère avec le point !
     }
     else
     {
     PORTA=caractere; // Un caractère
     }

  PORTB=(0b11110000 | adresse_constant[adresse & 0x0F]); // PB7 PB6 PB5 CS A0 A1 A2 A3
  PORTB.4=1;
  PORTB.4=0;  // CS = 0
  PORTB.4=1;
  PORTA=0x00;
}
void Programme1()                          // Programme CompteTours
{

    ENABLE=0;                               // Activation des sorties
    ClrAff();
    do                    // FAIRE LE TRAITEMENT TANT QUE BP2 ou BP3 n'est pas pressé
        {
        if(choix6==0)                                       // Fonctionnement Basique
            {
            sprintf(Texte,"Valeur :%5d       Tours",compteur); EcrireLCD(0,2);
            var1=compteur/10;
            d=compteur-var1*10;               //Valeur Unites
            afficheur1(0,d,0);                // Affichage digit 1
            var2=var1/10;                      //Valeur Dixaines
            d=var1-var2*10;                   // Affichage digit 2
            if (compteur>=10) {afficheur1(1,d,0);}   // Si supérieur ou égale à 10, digit 2 allumé.
```

```
            else {afficheur2(1,0,0);}                   // Sinon éteins
            var1=var2/10;                                // Valeur Centaines
            d=var2-var1*10;
            if (compteur>=100) { afficheur1(2,d,0);}
            else {afficheur2(2,0,0);}
            var2=var1/10;
            d=var1-var2*10;
            if (compteur>=1000) {afficheur1(3,d,0);}
            else {afficheur2(3,0,0);}
            var1=var2/10;
            d=var2-var1*10;
            if (compteur>=10000) {afficheur1(4,d,0);}
            else {afficheur2(4,0,0);}
            var2=var1/10;
            d=var1-var2*10;
            if (compteur>=1000000) {afficheur1(5,d,0);}
            else {afficheur2(5,0,0);}
            }
        else                                                // Téléthon
            {
            afficheur2(0,0x79,0); // Afficher E sur le digit 1
            sprintf(Texte,"Valeur :%5d/10       Euros",Euros); EcrireLCD(0,2);
            var1=Euros/10;
            d=Euros-var1*10;                //Valeur centimes d'euros
            afficheur1(1,d,0);               // Affichage digit 2
            var2=var1/10;                     //Valeur Unites d'euros
            d=var1-var2*10;
            afficheur1(2,d,1);                // Affichage digit 3 avec affichage d'un point permettant de fare la
virgule
            var1=var2/10;
            d=var2-var1*10;
            if (Euros>=100) { afficheur1(3,d,0);}
            else {afficheur2(3,0,0);}
            var2=var1/10;
            d=var1-var2*10;
            if (Euros>=1000) {afficheur1(4,d,0);}
            else {afficheur2(4,0,0);}
            var1=var2/10;
            d=var2-var1*10;
            if (Euros>=10000) {afficheur1(5,d,0);}
            else {afficheur2(5,0,0);}
            }
        if ((CAPT1==0)&&(CAPT2==0))                  // Si les deux capteurs sont coupés , la variable passe à 1
            {
            var=1;
            }
        if ((CAPT3==0)&&(CAPT4==0))                  // Si les deux capteurs sont coupés , la variable passe à 1
            {
            var3=1;
            }
        if ((CAPT1==1)&&(CAPT2==1)&&(var==1))        // Dès que les capteurs reviennent en position repos et que la
variable est à 1, le compteur s'incrémente de 1 , et les euros de 5;
            {
            Euros=Euros+5;
            compteur++;
            var=0;
            }
        if ((CAPT3==1)&&(CAPT4==1)&&(var3==1))      // Des que les capteurs reviennent en position repos et que la
variable est à 1, le compteur s'incrémente de 1 , et les euros de 5;
            {
            Euros=Euros+5;
            compteur++;
            var3=0;
            }


        }
    while((BP2&&BP3)==1);
}
void Programme2()          // Programme 50m  permet de définir selon le choix4, si on démarre le programme 4
correspondant à la borne de départ , ou programme 5 correspondant à la borne d'arrivée.
{                          // En théorie , cette fonction peut être supprimé et on pourrait lancer directement le
programme 4 ou 5 du main(); Prévoir modification.
switch(choix4)
    {
        case 0:  Programme4();        // Borne Départ
        break;
        case 1:  Programme5();         // Borne Arrivée
        break;
    }
}
void Programme3(void)     // Programme FEU&Horloge                                                              //
Rouge G = 4         // Rouge D = 32
{                                                                                                              //
Orange G = 8        // Orange D = 64
                                                                                                               // Vert
G  = 16        // Vert D = 128
```

File: test.c, Date: 31/10/2013, Time: 15:50:42

```
char g=4;                          // CODE FEU ROUGE GAUGHE                                    //
Klaxon = 2
char d=32;                          // CODE FEU ROUGE DROIT
int td,tg;
char variable=3;
char Alarme=2;
ENABLE=0;
do              // FAIRE TANT QUE BP1 et BP2 ne sont pas préssés
    {
    sprintf(Texte,"    %02d:%02d:%02d",Vheures,Vminutes,Vsecondes); EcrireLCD(0,2);


    d2=Vsecondes/10;
    d1=Vsecondes-d2*10;
    d4=Vminutes/10;
    d3=Vminutes-d4*10;
    d6=Vheures/10;                                      // AFFICHAGE DE L'HEURE
    d5=Vheures-d6*10;
    afficheur1(0,d1,0);
    afficheur1(1,d2,0);
    afficheur1(2,d3,1);
    afficheur1(3,d4,0);
    afficheur1(4,d5,1);
    afficheur1(5,d6,0);
    afficheur2(6,g+d,0);

    if (CAPT2==0)          // SI UN KART SE PRESENTE
        {
            if (CAPT1==0)                       // Si le kart grille le feu , déclenchement de l'alarme;
            {
            d=64;
            afficheur2(6,Alarme+g+d,0);
            delay_ms(1000);
            }
            ClrAff();
            do              // Faire tant que les CAPT1 et CAPT2 ne sont pas coupés
                {
                for(td=6;td>-1;td=td-1)                     //GESTION FEU DROITE
                    {
                    afficheur2(6,g+d,0);                      // FEU rouge
                    afficheur1(0,variable,0);                 // Affichage allant de 3 à 0
                    if(variable>0)
                        {
                        variable--;
                        }
                        delay_ms(1000);
                    if((td>3)&(CAPT1==1)&(d==32))       // Si la valeur est supérieur à 3 et que le capteur 1 n'est
pas coupé, feu rouge D
                        {
                        d=32;
                        }
                    if ((td>=4)&(CAPT1==0)&(d==32))         // Si la valeur est supérieur ou egale à 4 , alors feu
orange D et klaxon
                        {
                        d=64;
                        afficheur2(6,Alarme+g+d,0);
                        delay_ms(500);

                        }
                    if((td==4)&(d==32))                     // si td = 4 , feu vert D
                        {
                        d=128;
                        }
                    }
                }
                while((CAPT1==0)&(CAPT2==0));
                variable=3;                               // variable de décompte
                d=32;                                   // Retour au feu rouge
        }
    if (CAPT4==0)                                           // GESTION FEU GAUCHE?
        {                                                   // Même fonctionnement que le feu droit , seulement on
prends les valeurs pour le feu gauche
        ClrAff();
        do
            {
            for(tg=6;tg>-1;tg=tg-1)
                {
                afficheur2(6,g+d,0);
                afficheur1(5,variable,0);
                if(variable>0)
                    {
                    variable--;
                    }
                delay_ms(1000);
                if((tg>3)&(CAPT3==1)&(g==4))
                    {
                    g=4;
                    }
```

```
                if ((tg>=4)&(CAPT3==0)&(g==4))
                    {
                    afficheur2(6,Alarme+g+d,0);
                    delay_ms(500);
                    g=8;
                    }
                if((tg==4)&(g==4))
                    {
                    g=16;
                    }
                }
            }
        while((CAPT3==0)&(CAPT4==0));
        variable=3;
        g=4;
        afficheur2(5,0,0);
        afficheur2(0,0,0);

        }

    }
while((BP2&&BP1)==1);
}
void Programme4(void)           // Programme Borne Départ
{
int i;
ENABLE=0;
ClrAff();
car='X';                            // Initialise les caracteres d'envoie et réception
carRecu='X';
do          // Faire tant que BP1 et BP3 ne sont pas pressés

 {
 sprintf(Texte,"Recu:%1c Send:%1c", carRecu, car); EcrireLCD(0,3);
 if(carRecu=='X')
    {
        afficheur2(6,32+4,0);                       // Feu rouge permanent, tant qu'aucun kart ne se présente
        car='X';
        USART_Transmit(car);

    }
 if(CAPT2==0)
    {                                               // Kart présent au départ, signalement à l'autre borne
    car='C';
    USART_Transmit(car);
    carRecu=USART_Receive();
    }
 if(carRecu=='D')                           // Born d'arrivée prête
    {
        for(i=0;i<10;i++)                               // Test présence Kart pendant 10s
        {
            afficheur2(6,32+4,0);
            delay_ms(500);
            afficheur2(6,32+4+8+64,0);
            delay_ms(500);
            if((CAPT1==1)&(CAPT2==0))                               // Si il ne bouge pas , car=P
                {
                car='P';
                USART_Transmit(car);
                }
            else                                                    // Si il bouge, faux départ, car=I
                {
                car='A';
                USART_Transmit(car);
                afficheur2(3,0x71,1);
                afficheur2(2,0x77,1);
                afficheur2(1,0x3E,1);
                afficheur2(0,0x76,1);
                delay_ms(2000);
                ClrAff();

                i=10;
                }
        }
    if(car=='P')                                            // 10 secondes écoulés si le kart n'as pas bougé car=
P
        {
            x=1;
            do
                {
                    afficheur2(6,16+128,0);                 // Feu vert tant que  X=1, il passera à 0 lorsque CAPT2
reviendra en position repos seulement si le CAP1 reste à 0
                    car='O';
                    USART_Transmit(car);
                    if (CAPT1==0)
                    {
                        if(CAPT2==1)
                            {
```

```
                        x=0;
                        }
                    }
                }
            while(x==1);
            car='M';                    // Kart parti
            USART_Transmit(car);
            carRecu=USART_Receive();


        }
    if(car=='A')                                                    // Envoie du caractère si faux départ
        {
            carRecu=USART_Receive();
        }

    }
 }
while((BP3==1)&&(BP1==1));
}
void Programme5(void)        // Programme de réception
{

int i,var3,var4;
car='X';
carRecu='X';
ClrAff();
do                                  // Faire tant que BP1 et BP3 ne sont pas préssés
    {
    carRecu=USART_Receive();
    sprintf(Texte,"Recu:%1c Send:%1c", carRecu, car); EcrireLCD(0,3);
    if(carRecu=='C')                            // Kart présent à la borne de dépar
        {
            secondeV=9;                                         // Initialisation des variables pour le départ
            dixiemeV=9;
            centiemeV=9;
            car='D';
            USART_Transmit(car);
        }
    if(carRecu=='P')
        {
            x=1;
            do
                {
                    ENABLE=0;
                    Decompte=1;                             // Lancement du décompte dans le timer1
                    carRecu=USART_Receive();
                    if(carRecu=='A')
                    {
                    x=0;                                   // x=0 si le kart à fais un faux départ ou si le décompte
et terminé
                    }
                    if(carRecu=='O')
                    {
                    x=0;
                    }

                }
            while(x==1);
            Decompte=0;


        }
    if(carRecu=='A')                                        // FAUX DEPART
        {                                                   // Afficher pendant 2 sec
                    afficheur2(3,0x71,1);                      // F
                    afficheur2(2,0x77,1);                      // A
                    afficheur2(1,0x3E,1);                      // U
                    afficheur2(0,0x76,1);                      // H
                    delay_ms(2000);
                    ClrAff();
                    car='X';
                    USART_Transmit(car);
                    carRecu='X';


        }
    if(carRecu=='M')                                    // KART PARTI
        {
            car='X';                                                // Réinitialise l'autre borne
            USART_Transmit(car);
            do
            {                                                   //Le temps s'incrémente, tant que le kart ne passe
pas devant les capteurs de la borne d'arrivée
            Comptage=1;
            }
            while((CAPT1==1)&&(CAPT2==1));                                  // Kart arrivé
            Comptage=0;
```

```
            sprintf(Texte,"%1d%1d%1d%1d",dsecondeV,secondeV,dixiemeV,centiemeV);
            TempsK=atof(Texte)/3600;                                        // Conversion du
temps en Reel afin de déterminer la vitesse du kart en KM/H
            vitesse=(0,050/TempsK)*100;                                             //
Détermination de la vitesse
            vitesse=(int)vitesse;                                          //Conversion Vitesse
en Entier
            var1=vitesse/10;
            DecimalV=vitesse-var1*10;
            var2=var1/10;
            UniteV=var1-var2*10;
            var3=var2/10;
            DizaineV=var2-var3*10;
            var4=var3/10;
            CentaineV=var3-var4*10;
            for(i=0;i<5;i++)                          //Affichage de l'alternance Temps-Vitesse 5 fois
            {
            ClrAff();                                                       //avec une alternance
toutes les 2s
            afficheur1(0,CentaineV,0);
            afficheur1(1,DizaineV,0);
            afficheur1(2,UniteV,1);
            afficheur1(3,DecimalV,0);
            delay_ms(2000);
            ClrAff();
            afficheur1(0,centiemeV,0);
            afficheur1(1,dixiemeV,0);
            afficheur1(2,secondeV,1);
            afficheur1(3,dsecondeV,0);
            delay_ms(2000);

            }
        ClrAff();
        ENABLE=1;          // Désactive les sorties
        carRecu='X';


        }
    }
while((BP1==1)&(BP3==1));

}
void ChoixHeures(void)
{


    do                // Faire tant que BP1 n'est pas pressé
    {
        sprintf(Texte,"    Feu&Heure"); EcrireLCD(0,0);
        sprintf(Texte,"Reglage Heures"); EcrireLCD(0,1);
        sprintf(Texte,"    %02d:%02d:%02d ",Vheures,Vminutes,Vsecondes);  EcrireLCD(0,2);
        sprintf(Texte,"1.OK 2.+ 3.-");    EcrireLCD(0,3);
        if(BP2==0)                                  // Incrémentation Heures
                {
                delay_ms(250);
                Vheures++;
                }
        if(BP3==0)                                  // Décrémentation Heures
                {
              delay_ms(250);
                if(Vheures==0)
                    {
                    Vheures=23;
                    }
                else
                    {
                    Vheures--;
                    }
                }
        if(Vheures>23)
                {
                Vheures=0;
                }
    }
    while(BP1==1);
    delay_ms(500);

}
void ChoixMinutes(void)
{
    do     // Faire tant que BP1 n'est pas pressé
    {
        sprintf(Texte,"    Feu&Heure"); EcrireLCD(0,0);
        sprintf(Texte,"Reglage Minutes"); EcrireLCD(0,1);
        sprintf(Texte,"    %02d:%02d:%02d ",Vheures,Vminutes,Vsecondes);  EcrireLCD(0,2);
        sprintf(Texte,"1.OK 2.+ 3.-");    EcrireLCD(0,3);

        if(BP2==0)
```

```
                {
                delay_ms(250);
                Vminutes++;
                }
        if(BP3==0)
                {
                delay_ms(250);
                if(Vminutes==0)
                    {
                    Vminutes=59;
                    }
                else
                    {
                    Vminutes--;
                    }
                }
         if (Vminutes>59)
                {
                Vminutes=0;
                }
    }
    while(BP1==1);
    delay_ms(500);
    ConfigHorloge=1;            // La configuration de l'Horloge est terminée, nous ne retrerons plus dans les
fonctions ChoixMinutes, ChoixHeures
    lcd_clear();
    sprintf(Texte,"    Programme      Feu & Heure"); EcrireLCD(0,0);
}
interrupt [TIM1_COMPA] void timer1_compa_isr(void)            // TIMER 1
{
if(ConfigHorloge==1)                                            // La configuration étant faite, le temps s'écoule
    {
    temps++;
    if (temps>=100)
        {
            temps=0;
            Vsecondes++;
        if (Vsecondes>=60)
            {
            Vsecondes=0;
            Vminutes=Vminutes+1;
            }
        if (Vminutes>=60)
            {
            Vminutes=0;
            Vheures=Vheures+1;
            }
        if (Vheures>=24)
            {
            Vheures=0;
            }
        }

    }
if(Decompte==1)                 // Décompte pour Programme5 , borne arrivé de 10 à 0 s
    {

         centiemeV--;
            if(centiemeV==0)
                {
                centiemeV=9;
                dixiemeV--;
                if(dixiemeV==0)
                    {
                        dixiemeV=9;
                        secondeV--;
                        if(secondeV<0)
                            {
                            Decompte=0;
                            secondeV=0;
                            dixiemeV=0;
                            centiemeV=0;
                            x=0;
                            }

                    }

                }
     afficheur1(0,centiemeV,0);
     afficheur1(1,dixiemeV,0);
     afficheur1(2,secondeV,1);

    }
if(Comptage==1)                                             // Mesure du temps pour le 50m
    {
        centiemeV++;
            if(centiemeV==9)
                {
```

```
                dixiemeV++;
                centiemeV=0;
                    if(dixiemeV==9)
                        {
                            secondeV++;
                            dixiemeV=0;
                            if(secondeV==9)
                                {
                                    dsecondeV++;
                                    secondeV=0;
                                }
                        }
            }
    afficheur1(0,centiemeV,0);
    afficheur1(1,dixiemeV,0);
    afficheur1(2,secondeV,1);
    afficheur1(3,dsecondeV,0);
    }


}
void USART_Transmit( unsigned char data )           // Fonction de transmission de caractère RS232
{
/* Wait for empty transmit buffer */
while ( !( UCSRA & (0x20)) ) // Test de UDRE bit 5
;
/* Put data into buffer, sends the data */
UDR = data;
}
unsigned char USART_Receive( void )                             // Fonction de rececption de caractère RS232
{
/* Wait for data to be received */
while ( !(UCSRA & 0x80) ) // Test de RXC bit7
;
/* Get and return received data from buffer */
return UDR;
}
```