



Projet Tutoré de 2^e année

Bornes 50metres départ / arrêter



Projet Tutoré de 2^e année

Bornes 50 mètres départ / arrêté

<ROULOT-SALMON>

<2010 - K3B>

<2008/2011>

Enseignants

<THIERRY LEQUEU>

<MME LAURANCEAU>

Sommaire

Introduction.....	4
1. Présentation du projet.....	5
1.1. Cahier des charges.....	5
2. Études.....	10
2.1. Partie mécanique.....	10
2.2. Partie électrique.....	11
2.3. Partie informatique.....	14
3. Le challenge e-kart à Vierzon.....	25
Conclusion.....	27
Résumé.....	28
.....	28
Index des illustrations.....	29
Bibliographie.....	30

Introduction

Nous avons choisi, au cours de ce semestre 3 décalé, de continuer la mise au point d'un projet âgé de deux ans et jusqu'ici inachevé. Lors des heures dédiées à la création d'un projet en rapport direct avec notre enseignement technique, nous avons choisi de nous consacrer à l'étude des « bornes pour les 50 mètres départ arrêté ». Ce projet, en relation directe avec la compétition e-kart de Vierzon, sera utilisé dans le but d'acquérir le temps que réalisent les kartings électriques pour parcourir ces 50mètres. Pour mener à bien ce projet, nous aurons besoin d'un solide plan auquel nous devons nous tenir avec rigueur ainsi qu'un planning prévisionnel essentiel afin de ne pas perdre de temps. Nous énoncerons dans un premier temps le cahier des charges et le planning associé, afin de savoir dans quelle direction s'orienter. Puis nous aborderons les différentes études menées afin de répondre au mieux aux attentes. Enfin nous traiterons de l'apport du challenge e-kart de Vierzon dans la validation de notre travail.

1. Présentation du projet

Une compétition regroupant de nombreux e-karts conçus et réalisés par des étudiants en collaboration directe avec leurs enseignants se déroule chaque année en France. Ce challenge confronte les e-karts sur différents domaines, tels que l'endurance, où encore la vitesse, c'est sur ce dernier point que notre projet prend tout son intérêt.

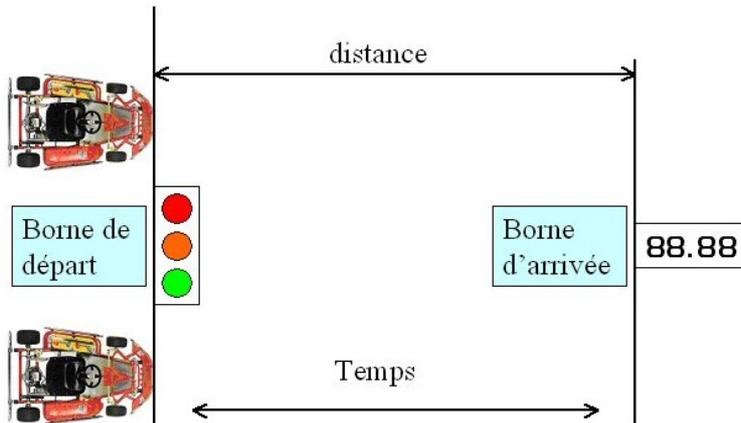


Illustration 1: schéma de principe de la mesure du temps de passage des karts[2]

Le principe est simple, les e-karts s'affrontent lors d'une course de 50 mètres sur « départ arrêté », le temps pour effectuer ce parcours est mesuré par deux bornes, placées au départ et à l'arrivée. Enfin, ces bornes réalisent l'affichage du temps.

Le projet consiste à concevoir et réaliser ces deux bornes en nous basant sur les études effectuées depuis deux ans.

Au cours de l'étude, nous devons trouver des solutions aux problèmes rencontrés les années précédentes et mettre en place des adaptations technologiques afin de répondre au mieux au cahier des charges imposé.

1.1. Cahier des charges

1.1.1. Origine du projet

Ce projet nous a été proposé par M. Thierry LEQUEU dans le cadre des travaux d'études et réalisations au programme du semestre 3 décalé de notre D.U.T. de Génie électrique et informatique industrielle. Le sujet nous a de suite charmés de par l'environnement qu'il vise, à savoir le karting électrique.

1.1.2. Description des fonctions principales

Les deux bornes devront être capables de détecter le passage des karts et pouvoir communiquer entre elles par le biais d'une liaison FM¹. Le dispositif devra également afficher le temps effectué durant les 50 mètres. De par leur utilisation elles seront amenées à être déplacées, elles devront donc être facilement transportables, voir démontables, être résistantes aux chocs, autonomes (alimentation par batteries 12V continu), pouvoir communiquer sans fil, être au maximum étanche, et économique en énergie.

¹ FM:frequency modulation

Les afficheurs pour leur part devront être visibles de loin. Dans un souci d'esthétique et de fonctionnalité, les deux bornes devront également être le plus identique possible (en fonction du matériel mis à disposition).

1.1.3. Schéma synoptique de niveau 1

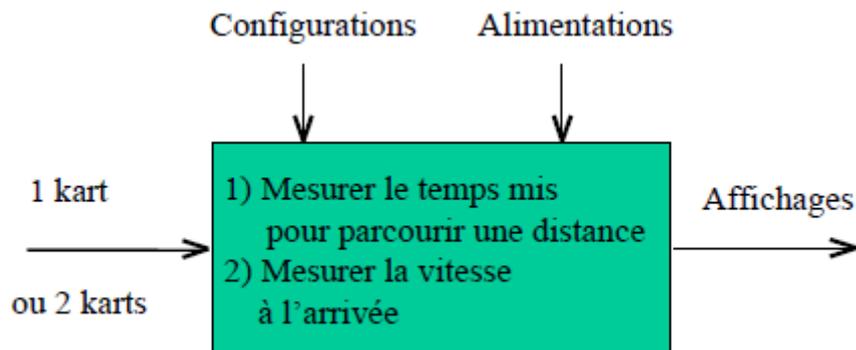


Illustration 2: schéma synoptique de niveau 1 [2]

L'alimentation afin de répondre au mieux au cahier des charges, notamment aux contraintes de transport et d'autonomie sera effectuée à l'aide d'une batterie de voiture 12V (pouvant varier d'une tension de 10volt à 14volt). L'utilisation du microcontrôleur Atmega 8535 est aussi imposée.

1.1.4. Contraintes

1.1.4.1 Mécaniques

Comme lors de tout projet, nous avons des contraintes à respecter. Celles exigées au niveau des bornes en elles-mêmes sont diverses et variées, ainsi, celles-ci doivent être identiques au maximum. Elles sont désirées simples d'utilisation et ne doivent pas être reliées par de quelconques câbles, d'où la mise en place d'une liaison FM dont nous reparlerons. De plus, le rangement des bornes arrivée et départ se veut efficace et facile. Ces bornes doivent impérativement résister aux chocs divers, ne pas être perméables. On souhaite enfin qu'elle ne dégage pas de chaleur (dans un souci d'économie d'énergie elles doivent pouvoir fonctionner de -10°C jusqu'à $+ 50^{\circ}\text{C}$) et qu'elles soient visibles de loin par les pilotes.

1.1.4.2 Alimentation

Elles doivent être autonomes en énergie et fonctionner sur une batterie OPTIMA 12V 48AH JAUNE.

1.1.4.3 Matériel

L'utilisation d'un microcontrôleur ATmega 8535 de chez ATMEL est imposée. Thierry LEQUEU dispose des outils de développement nécessaires à la programmation de ce microcontrôleur. La communication entre balises se fera de préférence par une liaison RS232 et de transmetteur HF à 433 MHz. Voir en annexe 1 la nomenclature des composants.

1.1.5. Planning Prévisionnel

Semaine	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Prise en connaissance du sujet	X																
Formation Orcad			X														
Réalisation du cahier des charges et planning prévisionnel	X	X															
Réflexion sur le système de détection		X															
Réalisation de la deuxième carte alimentation 24V			X														
Réalisation de la carte distribution				X													
Câblage de la carte distribution					X												
Câblage de la carte alimentation 24V				X													
Câblage des capteurs					X												
Câblage des afficheurs et des microcontrôleurs						X			X								
Tests de la programmation et résolution des erreurs										X	X	X	X				
Tests en intérieur													X				
Rédaction rapport									X	X	X	X	X	X	X		
Soutenance orale																	X
Programmation du microcontrôleur									X	X	X	X	X				
Travail sur l'ergonomie des bornes													X	X	X	X	
Concours e-kart à Vierzon _Test en grandeur réelle														X			

1.1.6. Planning réel

Semaines	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Prise en connaissance du sujet	X																
Formation Orcad			X														
Mise en place d'un interrupteur sur les bornes									X								
Câblage des LEDS sur les bornes				X	X												
Fixation des afficheurs et des modules d'alimentation 15V										X							
Réalisation de la deuxième carte d'alimentation 24V						X											
Fixation des capteurs reflex						X											
Fixation des réflecteurs						X											
Test en extérieur avec ensoleillement						X											
Fixation de la carte 24V sur la première borne					X												
Fixation de la carte 24V sur la deuxième borne									X								
Fixation des LEDS					X												
Câblage et test des capteurs reflex					X												
Réalisation du typon de la carte alimentation 24V				X													
Réalisation des cartes distribution du 24V				X													

Fixation des cartes de distribution du 24V					X												
Réalisation du cahier des charges	X																
Challenge e-kart à Vierzon												X					
Programmation des microcontrôleurs								X	X	X	X						
Tests de la programmation									X	X	X						
Travail sur l'ergonomie de la borne											X						
Soutenance orale																	X
Projet tutoré									X	X	X	X	X				

2. Études

2.1. Partie mécanique

Afin de répondre au cahier des charges, nous avons dû revoir la structure des bornes d'arrivés et de départ, la structure d'origine étant trop instable et donc risquait de se renverser au moindre coup de vent. Pour pallier à ce problème, nous avons ajouté un second axe de fixation relié par une plaque de bois au premier. Nous avons donc rigidifié la structure avec ce deuxième axe rendant ainsi l'ensemble moins sensible au vent ou autre vibration créée par le passage à grande vitesse des karts ou simplement à la condition atmosphérique. Le poids des afficheurs est mieux repartie, la borne est plus stable, et cependant reste toujours facilement démontable et transportable, la borne répond donc au cahier des charges.

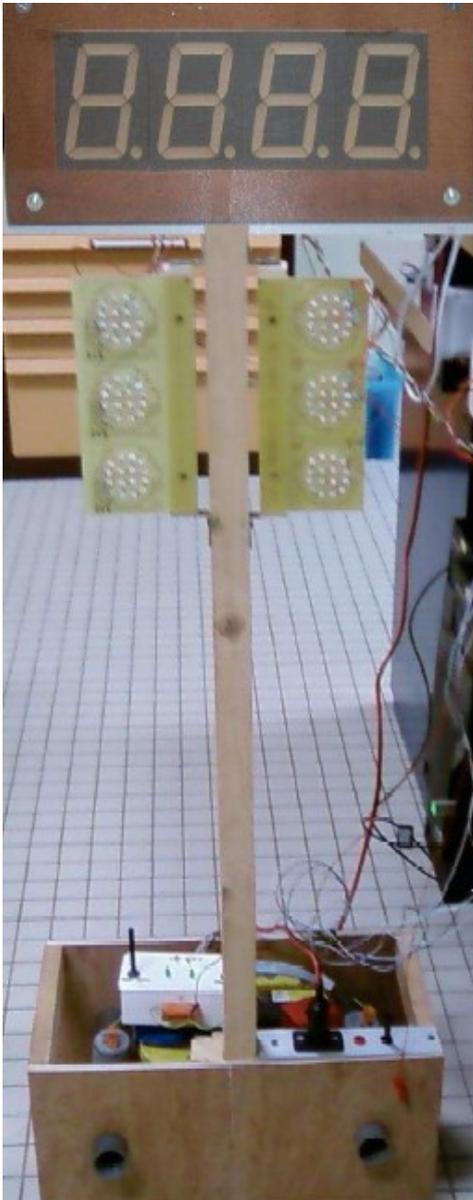
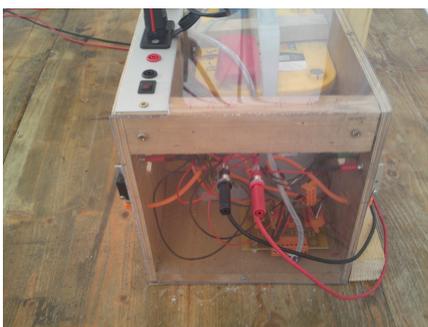


Illustration 3: état de départ de la borne[1]



Illustration 4: état actuel de la borne



Afin de protéger les cartes d'alimentation et de distribution (zone sensible des bornes), nous les avons couvertes avec du plexiglas transparent. Le plexiglas permettant de visualiser les zones sensibles tout en les protégeant des projections de gravier et de pluie.

Nous avons également l'ancien système de signalisation du départ par une implantation de feu à LED dans la plaque de

liaison des axes.

Nous n'avons malheureusement pas eu le temps de câbler ces feux, mais nous avons quand même prévu leurs implantations.

2.2. Partie électrique

Cette partie de l'étude étant très largement traitée dans le rapport d'Hertu\Hitey, nous avons choisi, de par les contraintes de temps, de ne rappeler et de ne traiter que les grandes lignes de l'étude de la carte d'alimentation 24V et de la carte distribution. Cette dernière ayant été partiellement refaite suite au perfectionnement de la détection des passages des karts et de la signalisation de ces passages à l'aide des diodes (rouge et verte) implantées dans la structure des bornes.

2.2.1. Carte d'alimentation

Cette carte doit pouvoir fournir une tension de 24V, nécessaire au fonctionnement du système, et délivrer un courant maximum de 120mA. En effet, une étude réalisée les années précédentes à montré que les bornes consommaient au maximum 30mA chacune. L'alimentation fonctionne grâce à une batterie de voiture pouvant fournir une tension de 12V. Afin de répondre à ces contraintes, il a été retenu le montage hacheur boost élévateur de tension suivant :

La documentation constructeur nous indique que ce montage comporte quelques contraintes :

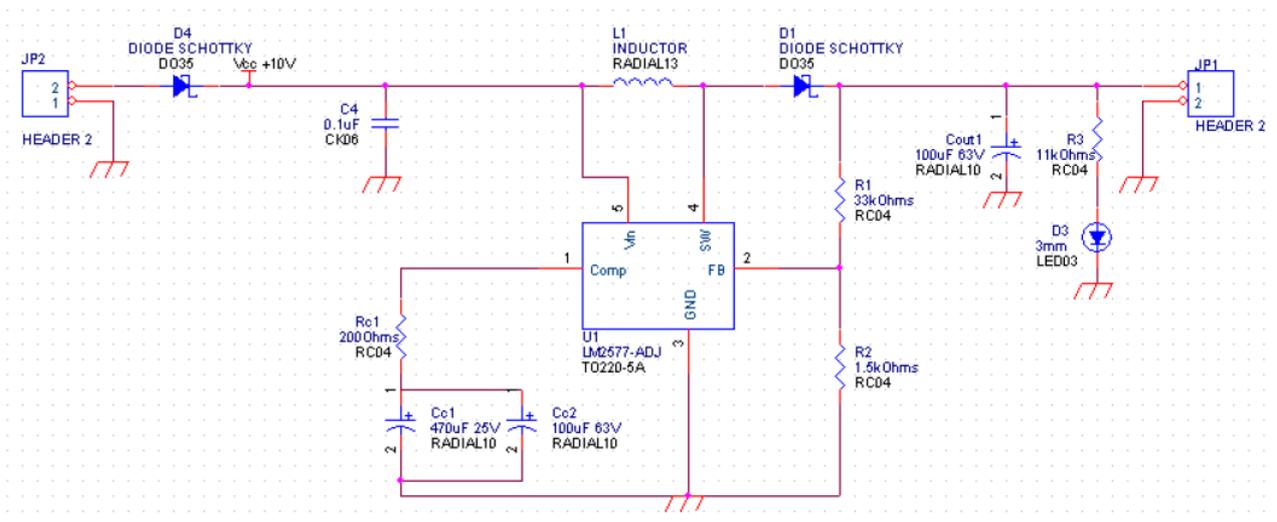
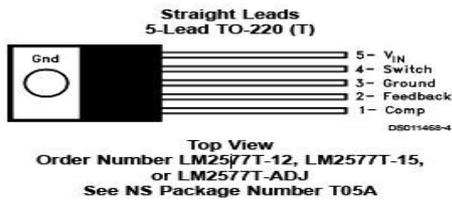


Illustration 5: schéma carte d'alimentation

Vout max doit être inférieure à 60V et supérieure à la tension minimum d'entrée du montage (donc supérieur à 10V), ces deux contraintes permettent malgré tout au montage de remplir la fonction attendue, à savoir l'alimentation des capteurs en 24V.



Illustration 6: Photo de la carte d'alimentation[1]



Après dimensionnement de tous les composants, nous obtenons les valeurs présentes ci-dessus avec une inductance minimum fixée à $L_{min}=1500\mu H$.

Illustration 7: schéma d'implantation du LM2577T[3]

2.2.2. Carte de distribution

Nous avons dû remplacer les borniers à trois entrées par des borniers quatre entrées afin de pouvoir câbler les LEDs rouge et verte permettant de visualiser la coupure du faisceau (permettant ainsi de positionner les réflecteurs plus rapidement en face des capteurs.)

Nous commençons tout d'abord par étudier le câblage des capteurs :

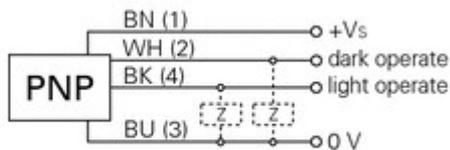


Illustration 9: schéma d'entrée/sortie des capteurs

L'entrée « dark operate » présente une tension de 5V lorsque le faisceau lumineux des capteurs est coupé (détection du passage d'un kart) et la « light operate » renvoyant 5V lorsque les capteurs ne détectent rien.

Les capteurs devront être alimentés en $+V_s = 24V$ à l'aide de la carte d'alimentation, et les informations des capteurs devront également être envoyés jusqu'au microcontrôleur.

Nous obtenons donc toutes les contraintes auxquelles devra répondre la carte de distribution, ce qui permet d'établir le schéma suivant et par la suite de générer le typon de la carte :



Illustration 8: Photo de l'implantation des diodes



Illustration 10: Photo d'un capteur

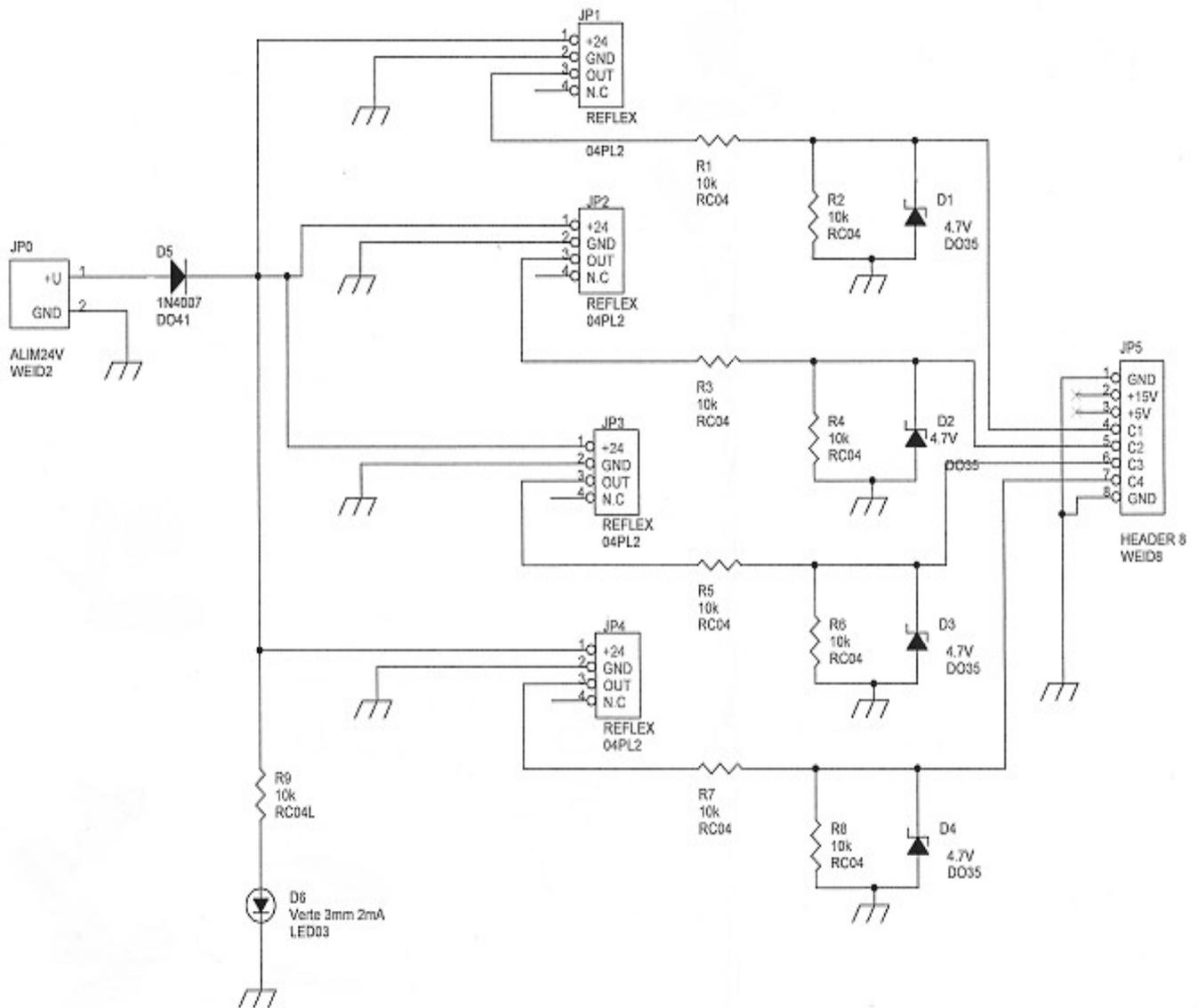


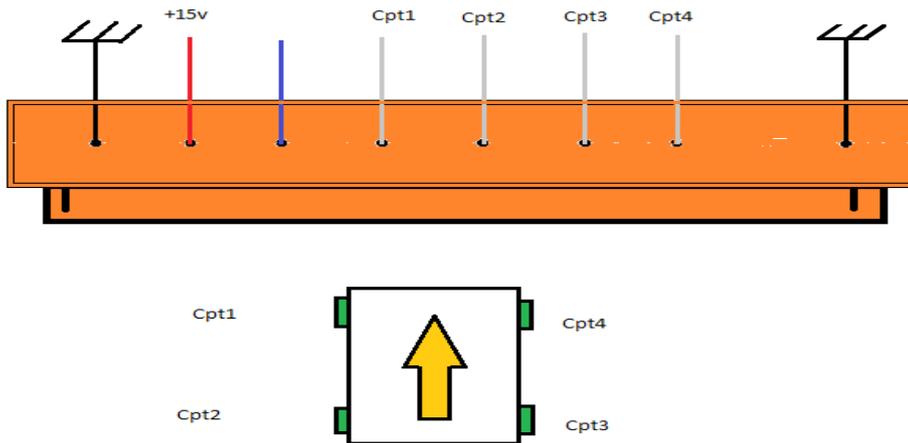
Illustration 11: Schéma de la carte de distribution

L'entrée OUT étant câblée sur la sortie « dark operate » des capteurs avec les diodes rouges, et les sorties « light operate » sur l'entrée N.C avec les diodes vertes (les diodes étant aussi reliées à la masse).

2.2.2.1 Liaison de la carte distribution vers la carte microcontrôleur

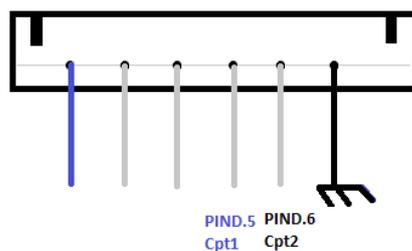
Nous choisissons d'envoyer les informations captées par la barrière reflex située sur l'avant de la borne de départ (capteur prévu pour lancer le chronométrage de la course) à l'entrée PIND.6 du microcontrôleur. La détection de l'arrivée se fera également sur le PIND.6 de la carte microcontrôleur de la borne d'arrivée.

Le câble de liaison entre les capteurs placés sur les bornes et leurs cartes microcontrôleur se présentera de façon identique à ceci près que les deux borniers ne sont pas identiques (bornier couder/bornier plat), tout le câblage de la nappe sera inversé au niveau du bornier.



2.2.2.2 Câblage de la borne d'arrivée

Le connecteur côté carte microcontrôleur sera un connecteur de type KK câblé de la façon suivante pour les deux bornes :



La programmation se fera donc à partir de la détection du passage à l'état 0 (0V) des PIND.6 et PIND.5.

La carte microcontrôleur mettant à disposition un connecteur KK reliant les entrées allant de PIND.6 à PIND.2, un bouton relié à l'entrée PIND.6 est également mis à notre disposition.

2.3. Partie informatique

Cette dernière partie de notre projet est celle qui va « lier » nos deux bornes, en effet sans communication entre celles-ci, rien n'est possible. Cette programmation se fera sur les micros-contrôleurs AT mega 8535, grâce au logiciel Code vision.

Nous avons mis beaucoup de temps à entrer dans le vif du sujet, n'ayant d'une part aucune connaissance en programmation d'ATmega, d'autre part, le précédent programme était peu commenté, nous avons donc décidé de repartir de zéro.

2.3.1. Présentation de l'ATMega 8535

Le composant ATMega 8535 nous est imposé pour la partie programmation.

L'illustration suivante nous montre les deux faces de la carte microcontrôleur. Les boutons poussoirs nous ont servi pour simuler les capteurs lors des phases de test.

Nous en avons donc défini deux, l'un pour simuler le start et l'autre pour le stop. Dans le programme final, il n'en restera qu'un servant à la remise à zéro.

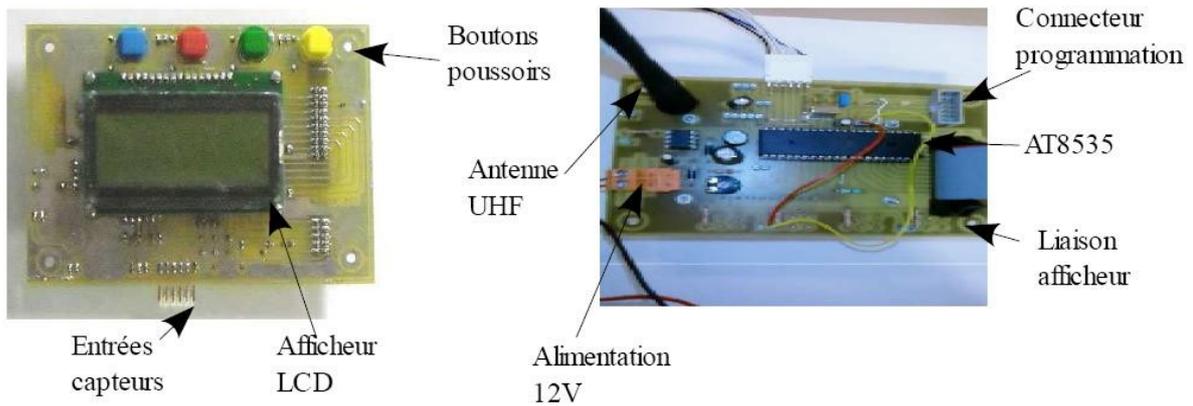


Illustration 12: Présentation d'un ATMEGA 8535

Le microcontrôleur AT8535 est un circuit programmable capable d'exécuter un programme et qui possède des circuits d'interface intégrés.

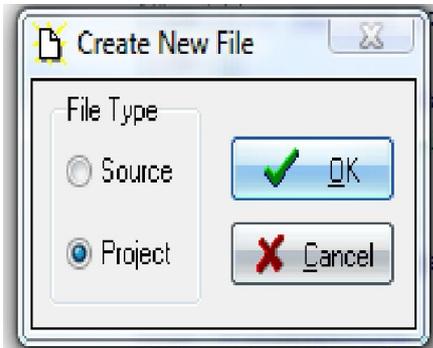
Il comporte différentes fonctions :

- CPU 8 Bits capable d'exécuter une instruction par cycle d'horloge,
- 8 Ko de mémoire programme EEPROM FLASH programmable,
- 512 Octets d'EEPROM (Stockage de données non volatiles),
- 512 Octets de RAM statique,
- convertisseur Analogique Numérique 10 bits à 8 entrées multiplexées,
- liaisons séries synchrone (SPI) et asynchrone (SCI),
- 2 TIMERS 8 bits (dont 1 utilisable en RTC a l'aide d'un oscillateur externe),
- 1 TIMER 16 bits,
- 1 Comparateur de tensions analogiques,
- 2 entrées d'interruptions externes et une entrée de RESET,
- 4 Ports d'entrées/sorties 8 bits.

De plus, il possède 4 ports d'E/S numériques de 8 bits chacun. Chaque bit ayant la possibilité d'être configuré en entrée ou en sortie. Nous les utiliserons pour les liaisons avec les capteurs de nos bornes.- 4 Ports d'entrées/sorties 8 bits. [4]

2.3.1. Génération d'un nouveau projet

À l'aide de Code vision, nous avons donc généré un nouveau projet, configurant les paramètres comme suivant :



Une fois le projet généré et sauvegardé avec le menu « generate and save » accessible depuis le menu « edit » .

Le logiciel Code vision édite lui même la partie suivante :

Illustration 13: Génération d'un nouveau projet

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

```
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 des afficheurs borne 50 m //Nom du projet que l'on a entré
```

```
Version : 3
```

```
Date : Mai 2010 //Date définie par l'utilisateur
```

```
Author : Thierry LEQUEU
```

```
Company :
```

```
Comments:
```

```
Chip type : Atmega8535 //Paramètre entré
```

```
Program type : Application // lors de la génération
```

```
Clock frequency : 16,000000 MHz // du nouveau projet
```

```
Memory model : Small
```

```
External SRAM size : 0
```

```
Data Stack size : 128
```

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

2.3.2. Les timers

Pour intégrer un « timer », il suffit d'en générer un et de définir tous les paramètres :



Pour générer une minuterie, il suffit d'entrer dans le menu ci-contre. Un "timer" est composé principalement d'un compteur qui évolue systématiquement lorsque le "timer" est actif.

Le comptage peut se faire à l'aide de différentes fréquences, fournies grâce d'une horloge interne ou externe. Le "timer" possède un ou plusieurs registres de comparaison et une ou plusieurs broches d'entrée/sortie configurable.

Illustration 14: Menu "generate and save" de Code Vision

Les "timers" 8 bits peuvent compter de 0 à 255 alors que le "timer" 16 bits compte de 0 à 65 535. Un registre de comparaison permet de comparer en permanence la valeur du compteur à celle que l'on a chargée durant la configuration. Lorsqu'il y a égalité entre les 2 valeurs, alors l'interruption du "timer" s'exécute si celle-ci est autorisée. On entre ensuite les paramètres du "timer":

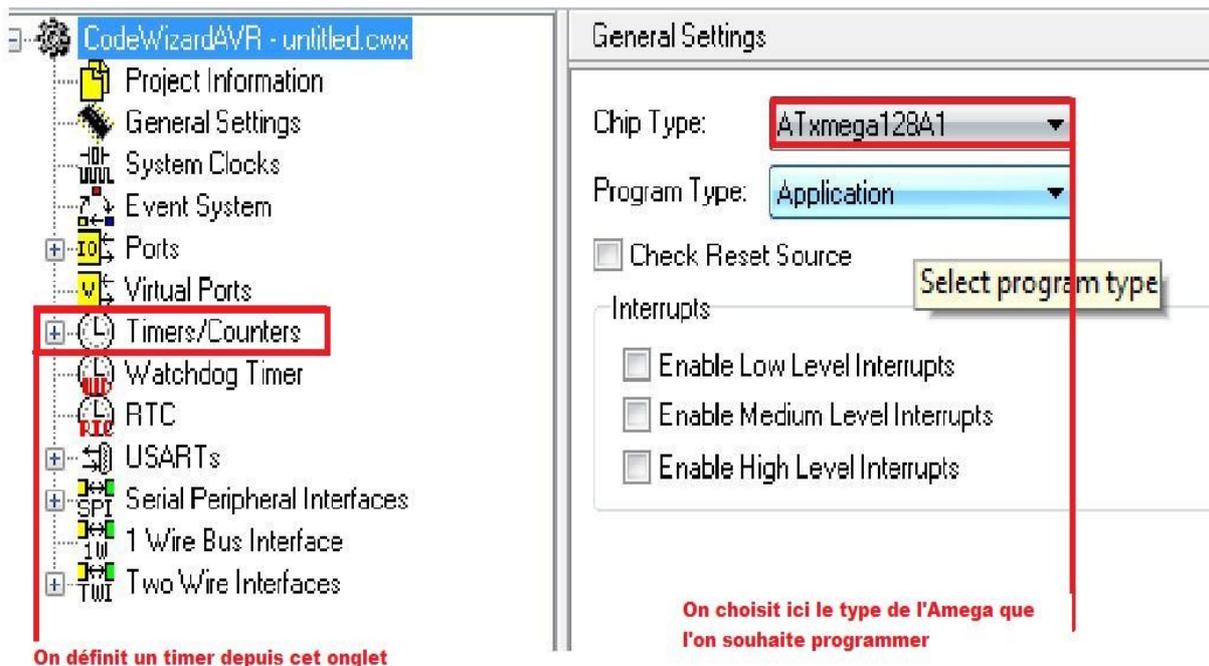


Illustration 15: Paramétrage des timers

```

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

```

2.3.3. Mise en place de la fonction LCD

L'affichage LCD² nous est préconisé pour faciliter la gestion des erreurs informatiques dans le programme, nous verrons pleinement son utilité lors des phases de test de la programmation.

On doit donc intégrer le module LCD dans notre programme, encore une fois, Code vision va gérer cela pour nous, toujours depuis le même menu.

```

#include <lcd.h>
// Bibliothèque LCD responsable de l'affichage de
// nos variables et autres.
lcd_init(16);
// LCD module initialization

```

Le logiciel génère alors automatiquement la bibliothèque ainsi que les lignes de codes gérant l'initialisation de notre module LCD .

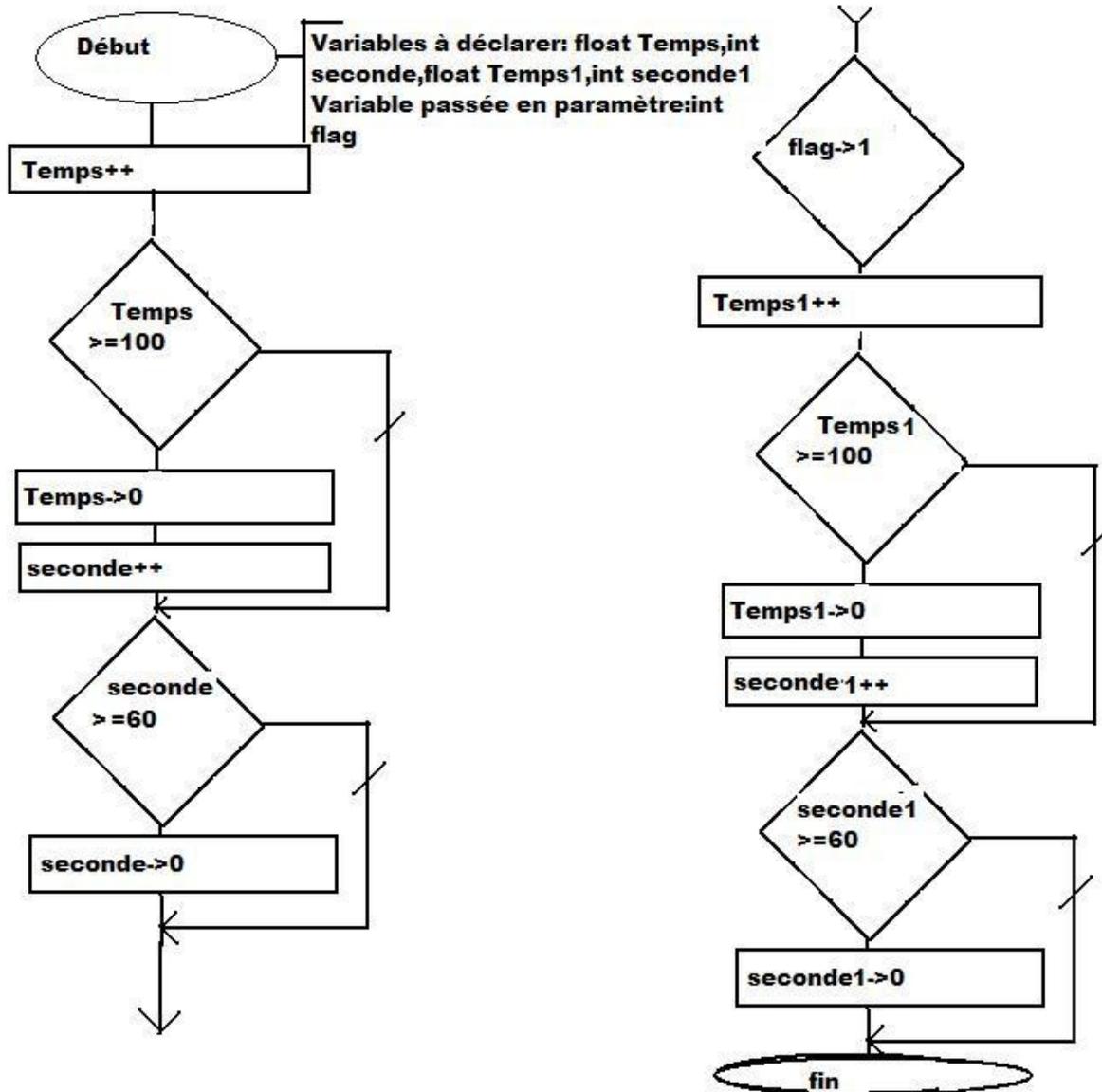
² LCD:Liquid Crystal Display

2.3.4. Réalisation du programme

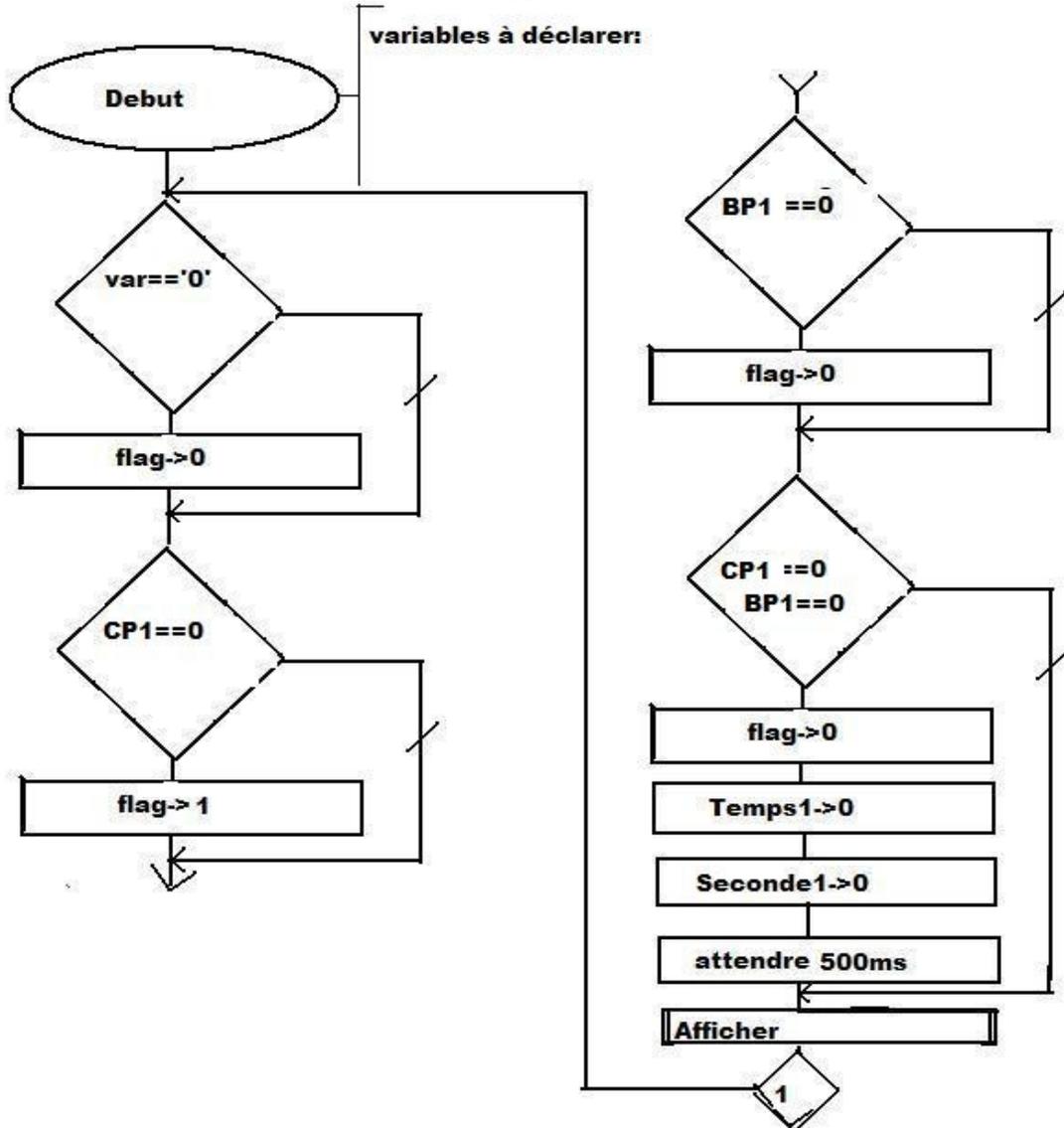
Nous avons désormais tous les outils nécessaires à la réalisation du programme, ou plutôt du programmes, en effet, nous n'aurons pas un, mais deux programmes destinés aux bornes de départ et d'arrivée gérant respectivement la réception et l'émission.

2.3.4.1 Algorithme du programme de la borne de départ

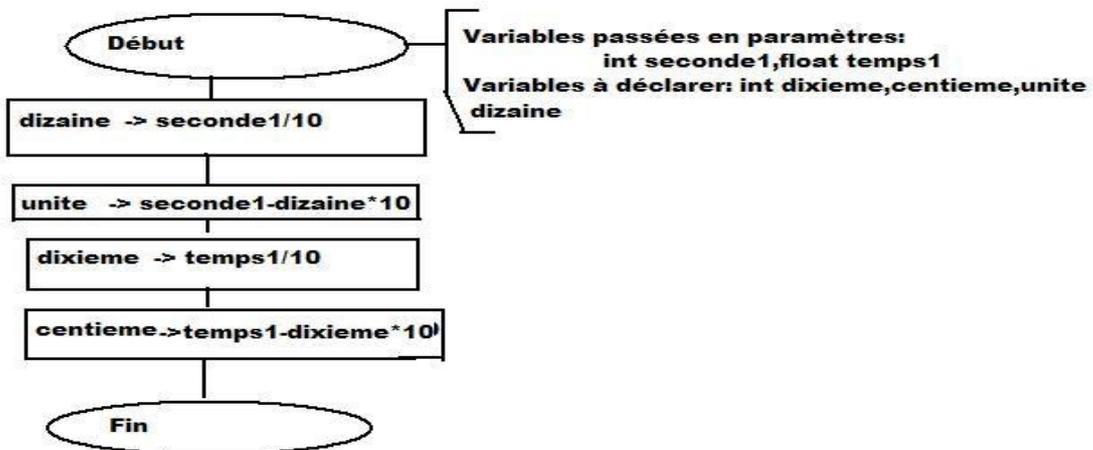
Afin de programmer la borne de départ, nous devons d'abord réfléchir à l'algorithme de la fonction "timer" :



Puis, nous devons concevoir la fonction principale,



La fonction affichage quand à elle est pensée comme ceci:



2.3.4.2 Programme de la borne de départ

Une fois le premier timer généré, on entre les lignes de code de la tempo qui ne va cesser de compter

```
temps++; //Le premier timer s'incrémente
if (temps>=100) //Si le temps dépasse la valeur 100 alors le temps se réinitialise
{
    //c'est en fait ce qui va gérer les centièmes
    temps=0;
    seconde++; //On incrémente alors les secondes
    if (seconde>=60) //Lorsque les secondes dépasse la valeur 60,alors on remet à zéro,parce
    qu'un karting met en moyenne 5 secondes pour effectuer les 50 mètres, on a donc de la marge
    avec nos 60 secondes
    {
        seconde=0;
    }
}
```

```
if (flag == 1) //Si l'état du flag est a un,on autorise le second timer à compter
{
    temps1++; //temps1 s'incrémente
    if (temps1>=100) //On retrouve le même principe que précédemment
    {
        temps1=0;
        seconde1++;
        if (seconde1>=60)
        {
            seconde1=0;
        }
    }
}
```

Le deuxième timer ci-dessus, ne va compter que lorsque la condition va s'avérer exacte.

C'est cette tempo qui sera affichée sur l'afficheur 7 segments de la borne de départ. La première tempo sera affichée sur notre module LCD, nous pourrons ainsi vérifier la concordance entre les deux lorsque les deux compteront, et voir si la tempo est précise.

Le but de créer deux tempo comme ici, est de pouvoir se servir d'une routine d'interruption qui comparera les deux à fréquence régulière.

Une fois les timers programmés, nous codons la fonction principale du programme :

```

while (1) //On place le code dans une boucle infinie
{
    var=USART_Receive() ; //Nous plaçons ce que nous recevons dans une variable
appelée var
    if(var=='0') //Si cette variable est égale a 0,alors on place le flag à zéro
    {
        flag=0;
    }
    if(CAP1==0) //Si le capteur 1 est a l'état zéro,on place le flag à un
    {
        flag=1;
    }
    if(BP1==0) //En revanche,si le Bouton poussoir un est a zéro,on place le flag à
zéro
    {
        flag=0;
    }
    if((BP1==0)&&(CP1==0))
    {
        temps1=0; //Si on met l'état du capteur1 et le bouton poussoir à zéro,alors
on réinitialise le temps du second timer
        seconde1=0; //de même,on réinitialise les seconde du second timer
        flag=0; //On remet alors le flag à zéro durant 500 ms pour avoir le temps
d'enlever les doigts du bouton poussoir et le pied de devant le capteur
        delay_ms(500);
    }
}

```

Remarque : Nous passons par l'utilisation de la variable flag pour éviter le retour à l'état un du capteur une fois le karting passé.

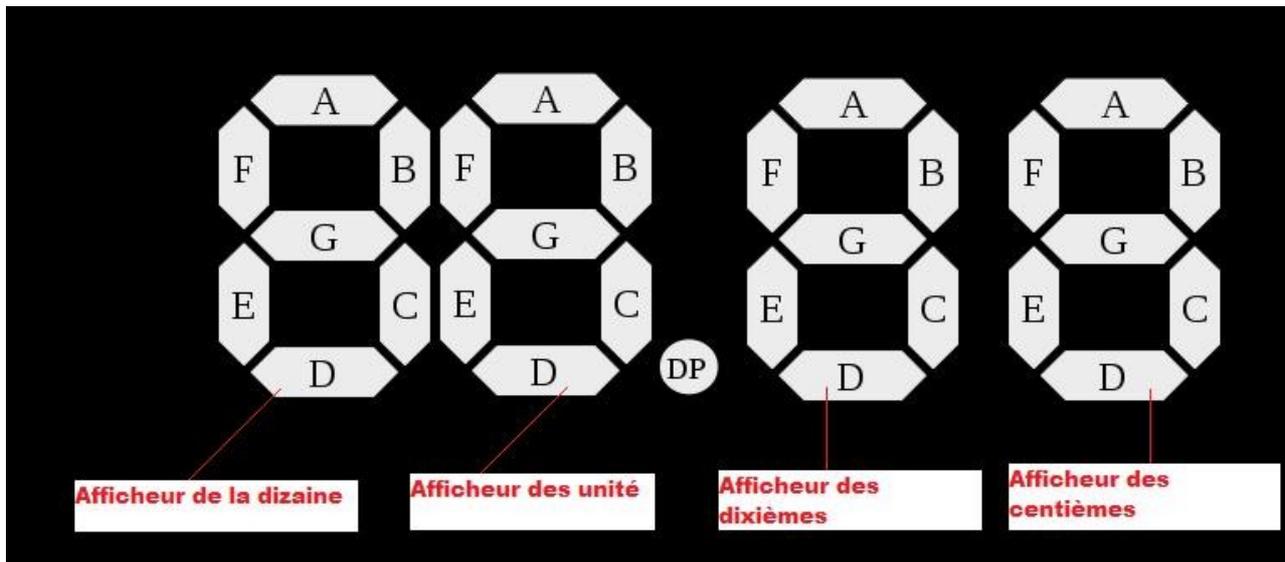
Nous n'avons plus qu'à gérer la partie affichage du programme :

```

dizaine = seconde1/10;
unite = seconde1-dizaine*10;
dixieme = temps1/10;
centieme = temps1-dixieme*10;
// Afficheur d'adresse 0
if (dizaine == 0) { //Si la valeur de dizaine est zéro
    afficheur1(0,13,0); //Alors on « allume » le segment D
}
else { //Sinon on y place la valeur de la dizaine
    afficheur1(0,dizaine,0); // Caractère des dizaines.}

```

Expliquons ce programme avec un exemple, imaginons que le temps à afficher soit de 55 secondes et 63 centièmes. Notre afficheur 7 segments se présente comme suivant:



Dans notre exemple le temps1 est de 63, et la variable seconde1 est égale à 55. Le programme va donc effectuer successivement :

$$55/10=5,5$$

Le microcontrôleur va donc placer le résultat 5 dans la variable dizaine. Remarquez que la valeur 0,5 ne sera pas retenue, car dizaine est une variable déclarée entière.

L'opération suivante est $55 - (5*10)=5$.

La valeur 5 va alors être stockée dans la variable unité qui est aussi une variable entière.

Puis vient le calcul dont le résultat sera placé dans la variable dixième soit :

$$63/10=6,3 \text{ ,encore une fois, la valeur retenue sera 6 et non 6,3.}$$

Enfin, le programme effectue le calcul pour déterminer la valeur des centièmes :

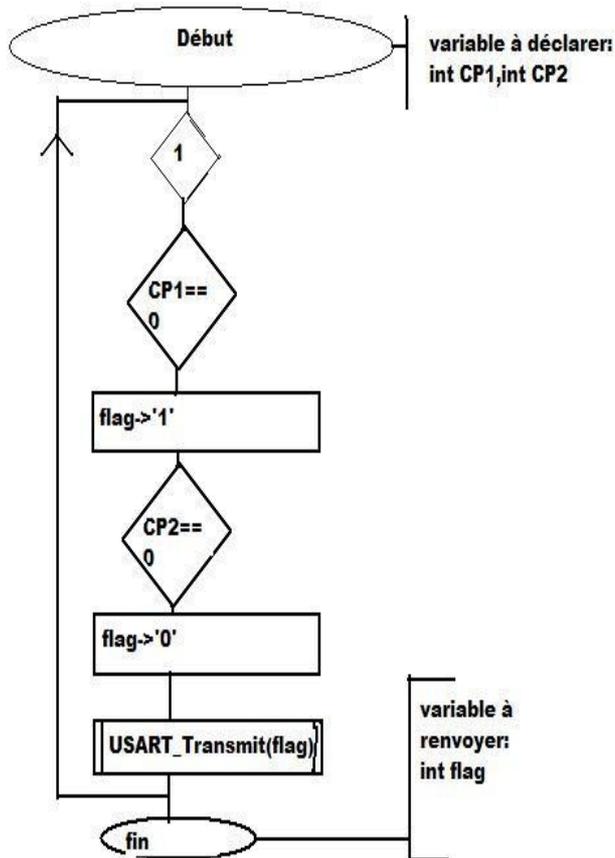
$$63-(6*10)=3 \text{ , la valeur 3 sera mise dans l'afficheur 7 segments des centièmes.}$$

On aura bien la valeur 55.63 affichée au final.

2.3.4.3 Algorithme du programme de la borne d'arrivée

Ce deuxième et ultime programme est le point clé du bon fonctionnement de nos bornes, c'est ici que nous allons finir la communication inter-bornes. Le premier programme étant codé, la borne des départs est maintenant susceptible de recevoir les informations nécessaires à l'arrêt du chronomètre, pour cela, il suffit juste que la deuxième borne lui fasse parvenir l'information du passage du karting.

C'est ce dont nous allons nous occuper dans ce programme, nous commençons par mettre nos idées dans un algorithme avant de nous lancer dans les lignes de codage.



Remarquez que le programme de la borne d'arrivée est très sommaire, tout le code est ancré dans la fonction principale qui a pour seul rôle de renvoyer à la borne de départ l'état de la variable définie par les capteurs reflex 1 et 2. On peut aussi noter l'utilisation de la fonction USART, mais cette fois-ci dans le cadre d'une émission (« transmit »).

Cette fonction est, à l'image des fonctions LCD ou Timer, générée automatiquement par le logiciel Code Vision.

Une fois le « plan » édifier, nous pouvons coder la fonction principale:

```

While (1) //Boucle infinie
{
    if(CP1==0) //Si CP1 est égale à 0,c'est à dire si le
    capteur 1 de la borne d'arrivée détecte le passage du kart
    {
        flag='1'; //Alors on met flag à l'état 1
    }
    if(CP2==0) ///Si CP2 est égale à 0,c'est à dire si le
    capteur 1 de la borne d'arrivée détecte le passage du kart
    {
        flag='0';//Alors on met flag à l'état 0
    }
    USART_Transmit(flag);//On envoie l'état de flag à la borne de
    départ
}
  
```

Notez que l'on envoie pas la valeur 1 ou 0, mais bien le code ASCII de 1 ou 0.

On remet le flag à zéro par le biais du capteur 2, c'est à dire, au moment où le karting a fini sa course. Cette opération est essentielle pour la remise à zéro du temps au niveau de la borne une, qui s'effectue en activant le bouton-poussoir 1 et le capteur 1 simultanément. Mais cette opération n'est possible que si flag est égal à zéro à ce moment.

2.3.5. Problèmes rencontrés et solutions apportées

Comme nul n'est sensé l'ignorer, une programmation marche rarement du premier coup,c'est pour cela que l'on a inventé le débogage ! Afin de faciliter celui-ci,nous avons fixé l'écran LCD sur la carte microcontrôleur.Celui ci nous indiquait alors de manière générale :

```
lcd_gotoxy(0,0); //A la ligne une de la première colonne:
lcd_putsf("Borne arrivee"); //La borne dont il est question
lcd_gotoxy(0,1); // //A la ligne deux de la première colonne:
lcd_putsf("Epreuve 50 m DA"); //Affichage du nom du projet
lcd_gotoxy(0,2); //A la ligne trois de la première colonne:
sprintf(tampon,"Tempo = %3d",tempo); //Affichage de la tempo
lcd_gotoxy(0,3); //A la ligne quatre de la première colonne:
sprintf(tampon,"flag = %1c ,%d",flag,CP1); //Affichage de l'état de flag et des
capteurs
```

Ces informations nous ont été précieuses afin d'analyser et de comprendre d'où venait la source des problèmes. Ainsi, nous avons pu voir que le flag ne se mettait pas correctement à l'état un. Ne permettant ainsi pas à la temps1 de compter. Comme lors des tests nous simulions les capteurs par des boutons poussoirs, nous avons tout de suite compris qu'il était le problème. En effet, il s'est révélé que l'assignation n'était pas correcte et que, de plus, le bouton poussoir était mal câblé et possédait un faux contact. Puis nous avons vu que le flag ne se mettait pas correctement à zéro non plus, ne provoquant ainsi pas l'arrêt du chronomètre, le chronomètre s'arrêtait, mais lorsque l'on enlevait le doigt du bouton poussoir, il redémarrait de sitôt. C'était en réalité une erreur de condition dans le programme. Autre problème rencontré, l'afficheur indiquait le temps que si nous restions le doigt appuyé au bouton-poussoir, dès le retrait du doigt le chronomètre comptait toujours, mais n'affichait plus. C'était en fait dû à un nouveau code dans les Atmega, il s'agissait de la fonction « enable »,qui autorise en fait l'affichage des 7 segments,la notre n'était pas correctement défini.

Après réalisation des programmes et du débogage, tout marche à merveille et le chronomètre s'avère très précis, nous l'avons comparé avec un chronomètre manuel.

Nous avons placé en annexe 2 un schéma réalisé par nos soins qui résume très bien la programmation.

3. Le challenge e-kart à Vierzon

Nous avons, lors du challenge robotique et e-kart, eu l'opportunité de tester nos bornes en grandeur réelle. Ce fut pour nous le premier test déterminant. Mais les conditions là bas s'avèrent cauchemardesques de par l'émission de beaucoup d'ondes parasites (dû aux émissions wi-fi, aux micros sans fil, aux cartes RFID³ présentes sur les karts pour comptabiliser leurs passages par la ligne d'arrivée de la piste) présentes.

Dès notre arrivée nous avons testé nos bornes sur une très courte distance (de l'ordre de quelques mètres) et tout de suite nous avons remarqué un léger dysfonctionnement lors du comptage, en effet le compteur s'arrêtait quelquefois avant que le faisceau du capteur de la borne d'arrivée soit coupé.

Au bout de plusieurs essais, nous remarquons que cette anomalie ne se produit que lorsque les kartings courent sur la piste située à quelques mètres de notre lieu de test. L'anomalie étant minime nous décidâmes tout de même de tester lors de l'épreuve départ arrêtée, l'échange des informations entre les bornes espacées de 50m.

À cette distance les bornes ne communiquaient plus du tout, c'est-à-dire que chacune des bornes fonctionnait bien individuellement, la borne de départ lançait le chrono lorsque le kart passait devant, et la borne d'arrivée changer l'état du flag (visible depuis l'écran LCD) lors du passage des karts devant les capteurs 1 et 2. Ceci peut être dû à plusieurs choses :

- Le mauvais dimensionnement de la liaison FM: cette liaison est peut-être trop faible, pour résoudre ce problème nous devons donc amplifier les informations.
- Les parasites étaient peut-être trop important pour que les bornes puissent capter les informations : nous pourrions donc, afin de résoudre ceci, amplifier également la liaison FM et créer un protocole permettant l'isolation des informations transmises ou transmettre l'information par câble, cependant cette dernière solution ne répondait pas au cahier des charges.
En plus de ce problème nous avons dû résoudre rapidement le problème de l'alimentation des capteurs, autrement dit du problème de notre carte d'alimentation.
- N'ayant toujours pas accès au bon régulateur nous avons choisi de mettre en place deux fiches sur la structure permettant une alimentation directe de notre carte de distribution et donc de nos capteurs.

³ Radio frequency identification: méthode pour mémoriser et récupérer des données à distance

Conclusion

Lors de notre semestre 3 décalé, la reprise du projet « 50 mètres départ arrêté » déjà existant nous a permis de traiter toute unité d'enseignement issue de notre DUT GEII. En effet, nous avons effectué de l'électronique, de l'informatique, et même de la mécanique.

Ce projet a vraiment suscité en nous beaucoup d'intérêt puisqu'il est très varié et très « concret ». De plus, nous avons particulièrement aimé le secteur visé, à savoir le karting. Ajouté à cela l'immense privilège d'assister au concours e-kart et robotique à Vierzon, ce en tant qu'« invité » et non-spectateur. Nous avons mis à bien tout notre savoir pour mettre au point les bornes de mesures, mais cela nous en a apporté aussi, nous plongeant dans un environnement similaire à celui que doit être l'entreprise. Nous avons réellement mis un point d'honneur à faire fonctionner ces bornes, quitte à sacrifier bon nombre de nos heures libres, ne considérant pas cela comme une contrainte.

À l'image de bon professionnel, nous avons commencé par nous imprégner du sujet afin d'en comprendre parfaitement le cahier des charges et ce que les enseignants attendaient de nous. Nous sommes désormais certains d'avoir répondu à leurs attentes, en terme de sérieux et d'investissement, investissement qui n'a pas été mis de côté, autant pour ce qui est du côté pratique que pour ce qui est de la trace écrite. On y voit ici encore une facette de l'enseignement Geii puisque les qualités de rédaction et de synthèse qui nous ont été enseignées en cours de culture et communication ont été mises à profit. Nous avons passés la moitié du temps disponible à la partie mécanique, pour laquelle nous avons dû faire un travail d'ergonomie, voulant rendre la borne aussi « parfaite » que possible. Le côté électronique a été repris en majeure partie puisque la théorie avait été faite par nos prédécesseurs, nous n'avons eu qu'à apporter quelques modifications sur l'une des cartes et à les réaliser. Pour ce qui est de l'informatique, le travail a été long puisque nous sommes partis quasiment de zéro. Nous avons, grâce à ce projet, fait nos premiers en programmation Atmega. Pour finir, nous pouvons ajouter que ce projet nous a directement confronté aux problèmes auxquels nous sommes en droit de nous attendre en tant que « concepteurs », et surtout à leurs résolutions, ce en autonomie. C'est à nous de chercher des solutions, d'effectuer les tests nécessaires, de passer les commandes, de lire les datasheet, de revoir la théorie de certains ensembles ou sous-ensemble. Nous devons être à même d'identifier la source de l'erreur, et bien sur de faire en sorte que celle-ci cesse.

Résumé

Au cours de ce semestre trois nous avons mené à bien notre projet, projet qui consistait à concevoir et réaliser, à l'aide des études précédentes, deux bornes visant à mesurer le temps de passage de karting entre deux-points distants de 50 mètres. Le cahier des charges nous imposait des bornes ergonomiques au possible, devant être étanches démontables, solides, simples.

Pour les réaliser nous sommes passé par une étude mécanique durant laquelle nous avons eu à revoir le système d'acquisition, la mise en place de feux et la stabilité des structures. Le changement de système d'acquisition nous obligeait à repenser le câblage interne de la borne et à adapter les cartes capteurs (en nous basant toujours sur les études précédentes).

Enfin, il nous a fallu configurer la communication entre les deux bornes, gérer l'acquisition des informations venant des capteurs, et les utiliser afin de pouvoir afficher le temps de passage. Pour ceci nous avions à notre disposition une carte Atmega8535 (dont l'utilisation était imposée) et le logiciel code vision.

Nous avons eu la chance de participer au challenge e-kart, au cours duquel nous avons pu effectuer des tests en grandeur réelle dans un milieu où transitait de nombreuses ondes parasites (micro sans fil, carte RFID...). Nous avons donc pu répertorier les problèmes afin de pouvoir les résoudre dans les dernières heures de notre semestre.

<226> mots

Index des illustrations

Illustration 1: schéma de principe de la mesure du temps de passage des karts[2].....	5
Illustration 2: schéma synoptique de niveau 1 [2].....	6
Illustration 3: état de départ de la borne[1].....	10
Illustration 4: état actuel de la borne.....	10
Illustration 5: schéma carte d'alimentation.....	11
Illustration 6: Photo de la carte d'alimentation[1].....	11
Illustration 7: schéma d'implantation du LM2577T[3].....	12
Illustration 8: Photo de l'implantation des diodes.....	12
Illustration 9: schéma d'entrée/sortie des capteurs.....	12
Illustration 10: Photo d'un capteur.....	12
Illustration 11: Schéma de la carte de distribution.....	13
Illustration 12: Présentation d'un ATMEga 8535.....	15
Illustration 13: Génération d'un nouveau projet.....	16
Illustration 14: Menu "generate and save" de Code Vision.....	17
Illustration 15: Paramétrage des timers.....	17

Bibliographie

[1] Hitey/Hetru, "*Rapport de projet*", 2009.

[2] Thierry LEQUEU, « Cahier des charges », <http://www.thierry-lequeu.fr/data/Projet-50m-DA.pdf>, (page consulté le 2010).

[3] Datasheet LM2577T, <http://www.futurlec.com/Linear/LM2577T-ADJ.shtml>, 2010.

[4] Datasheet Atmega8535, http://www.atmel.com/dyn/ressources/prod_documents/2502.pdf, 2010.

Composant	Quantité	Référence	Prix unitaire (€)
Barrière réflex sans réglage de la sensibilité	8	<i>FDPK 10P5130/S35A</i>	55
Câble+Connecteur	8	<i>ESW 31AP</i>	3,5
Réflecteur rectangulaire 54x75-2	8	<i>101 4561 – FTDR 047A048</i>	11,56
Régulateur de tension d'alimentation à découpage (boîtier TO220-5)	2	<i>LM 25 77T – ADJ – NOPB</i>	9,2
Inductance 1500 µH	2	<i>ELC09D152F – 540 8796P</i>	0,14
Condensateur FC 100µF 63V électrochimique polarisé	4	<i>EEUFC1J101 – 315 0962</i>	2,29
Condensateur FC 470µF 25V électrochimique polarisé	2	<i>EEUFC1E471 – 628 3790</i>	2,64
Diode Schottky	4	<i>1N5822RLG – 6547139</i>	2
LED faible consommation	2	<i>11DQ04 – 395 2459</i>	3,42
Condensateur 100nF	2		0,5
Résistance 33kΩ	2	<i>RCC025 33K J – 386 632</i>	3,75
Résistance 1,8kΩ	2		0,01
Résistance 200Ω	2		0,01
Résistance 11kΩ	2		0,01
Bornier 2 connections	8	NC	NC
Bornier 4 connections	8	NC	NC
Bornier 8 connections	2	NC	NC
TOTAL	617€		

Annexe 2:Schéma en rapport au principe de la programmation

