

Éclairage d'un kart électrique : Programmation



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

Éclairage d'un kart électrique : Programmation

TOUBLANC Kévin
THYPHONNET Franck
2^{ème} Année Groupe Q2
2007 // 2009

Enseignants :
Thierry LEQUEU
Bernard GLIKSHON

Sommaire

Introduction	4
1 Cahier des charges	5
1.1 Contraintes :.....	5
1.2 Synoptique général.....	6
2 Présentation du projet précédemment réalisé	7
2.1 Présentation.....	7
2.2 La carte électronique.....	7
2.3 Le microcontrôleur AtMega8535.....	9
2.3.1 Description des broches.....	10
2.3.2 Description des fonctions principales.....	12
2.3.2.1 La MLI.....	13
2.4 Etude et mise en place des capteurs.....	13
2.4.1 La photorésistance.....	14
2.4.2 Le potentiomètre mécanique.....	14
3 Réalisation du programme	16
3.1 Organigramme du programme.....	16
3.1.1 Description d'un organigramme.....	17
3.1.2 Explication de notre organigramme.....	19
3.2 Fonctions utilisées.....	19
3.2.1 Programmation des ports E/S.....	19
3.2.2 Programmation de la fonction MLI.....	20
3.2.3 Programmation du CAN (Convertisseur Analogique Numérique).....	20
3.2.4 Définition des symboles.....	21
3.3 Programmation des phares.....	22
3.3.1 Analyse théorique.....	22
3.3.2 Réalisation.....	23
3.4 Programmation des feux stop.....	24
3.4.1 Analyse théorique.....	24
3.4.2 Réalisation.....	24
3.5 Programmation des warnings et des clignotants.....	25
3.6 Mode Tuning.....	26
4 Finalisation du projet	27
4.1 Cartes à LEDs.....	27
4.2 Warnings et clignotants.....	27
4.3 Boitier de commande.....	27
4.4 Problème et difficultés.....	27
Planning	29
Conclusion	30
Table des illustrations	31
Index des Tableaux	31
Bibliographie	32
Annexes	32

Introduction

Au semestre 3, un groupe a réalisé une carte électronique permettant le contrôle de l'éclairage et de la signalisation d'un karting, n'ayant pas eu le temps de terminer, c'est nous qui allons finaliser ce projet.

Pour ce faire, il sera utile d'étudier la carte précédemment réalisée afin de comprendre son fonctionnement et d'y ajouter, si nécessaire, des améliorations.

Ensuite, nous devons réaliser un programme répondant au nouveau cahier des charges, que l'on implantera dans le microcontrôleur.

Puis, afin de terminer correctement ce projet, nous réaliserons entièrement le boîtier de commande de la carte.

1 Cahier des charges

Ce projet consiste en la programmation d'un microcontrôleur AtMega8535 étant implanté sur la carte d'éclairage et de signalisation d'un karting, réalisée au semestre précédent. Cette programmation permettra le contrôle de l'éclairage et de la signalisation au complet.

1.1 Contraintes :

- Microcontrôleur AtMega8535
- Etude de la carte réalisée précédemment
- Réalisation d'un circuit de commandes
 - o Interrupteur deux positions (manuel / automatique)
 - o Interrupteur trois positions (repos / feux de position / feux de croisement)
 - o Interrupteur trois positions (clignotant gauche / repos/ clignotant droit)
 - o Interrupteur deux positions (repos / warnings)
 - o Interrupteur deux positions (repos / Tuning party)
 - o Signalisation des commandes par LEDs
- Programmation du microcontrôleur
 - o Feux de position
 - o Feux de croisement
 - o Eclairage variable (mode automatique)
 - o Clignotants
 - o Warnings
 - o Feux de recul
 - o Feux stop
 - o Effet Tuning pour le Paris Tuning Show 2009
- Utilisation du logiciel Code Vision AVR

1.2 Synoptique général

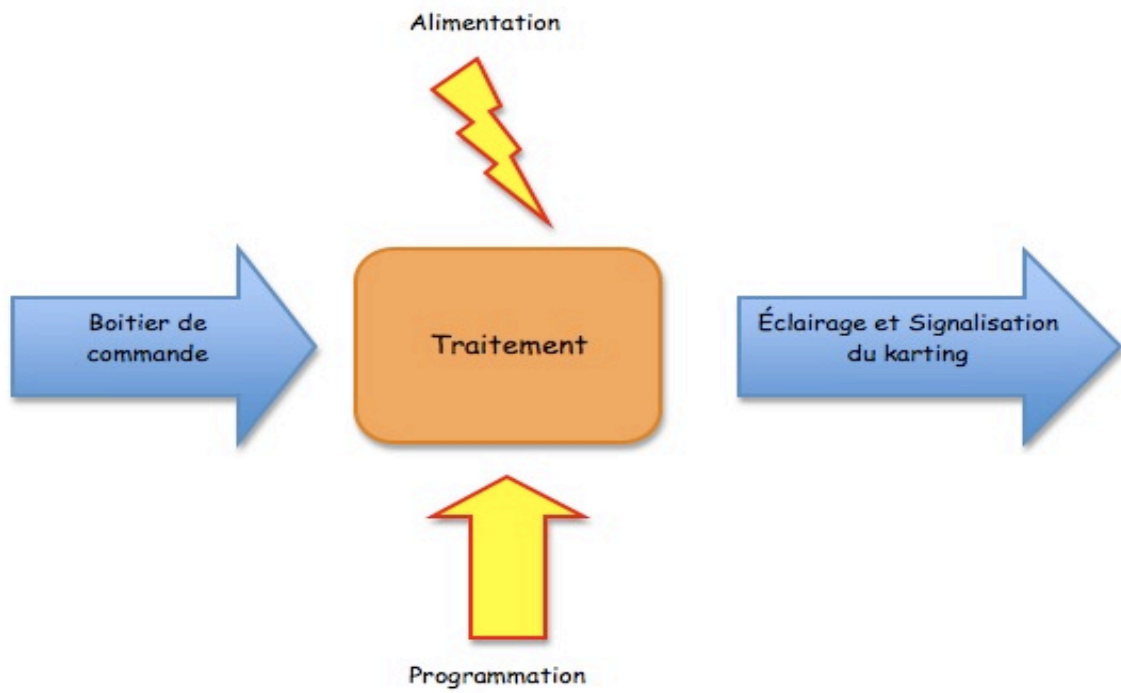


Figure 1 Schéma synoptique général du projet

2 Présentation du projet précédemment réalisé

2.1 Présentation

Le projet consistait en la réalisation de l'éclairage et de la signalisation d'un kart électrique de l'IUT GEII de Tours.

Le tableau ci-dessous énumère les différents modes recherchés.

Eclairage		
	Mode Manuel	Feux de position
		Feux de croisement
	Mode Automatique	
Eclairage Variable		
Signalisation		
	Clignotants	
	Warnings	
	Feux stop	

Pour répondre à leur cahier des charges et ainsi pouvoir obtenir un kart éclairé, ils ont réalisé une carte électronique.

2.2 La carte électronique

Dans cette sous-partie, nous allons expliquer de manière succincte le fonctionnement de la carte électronique.

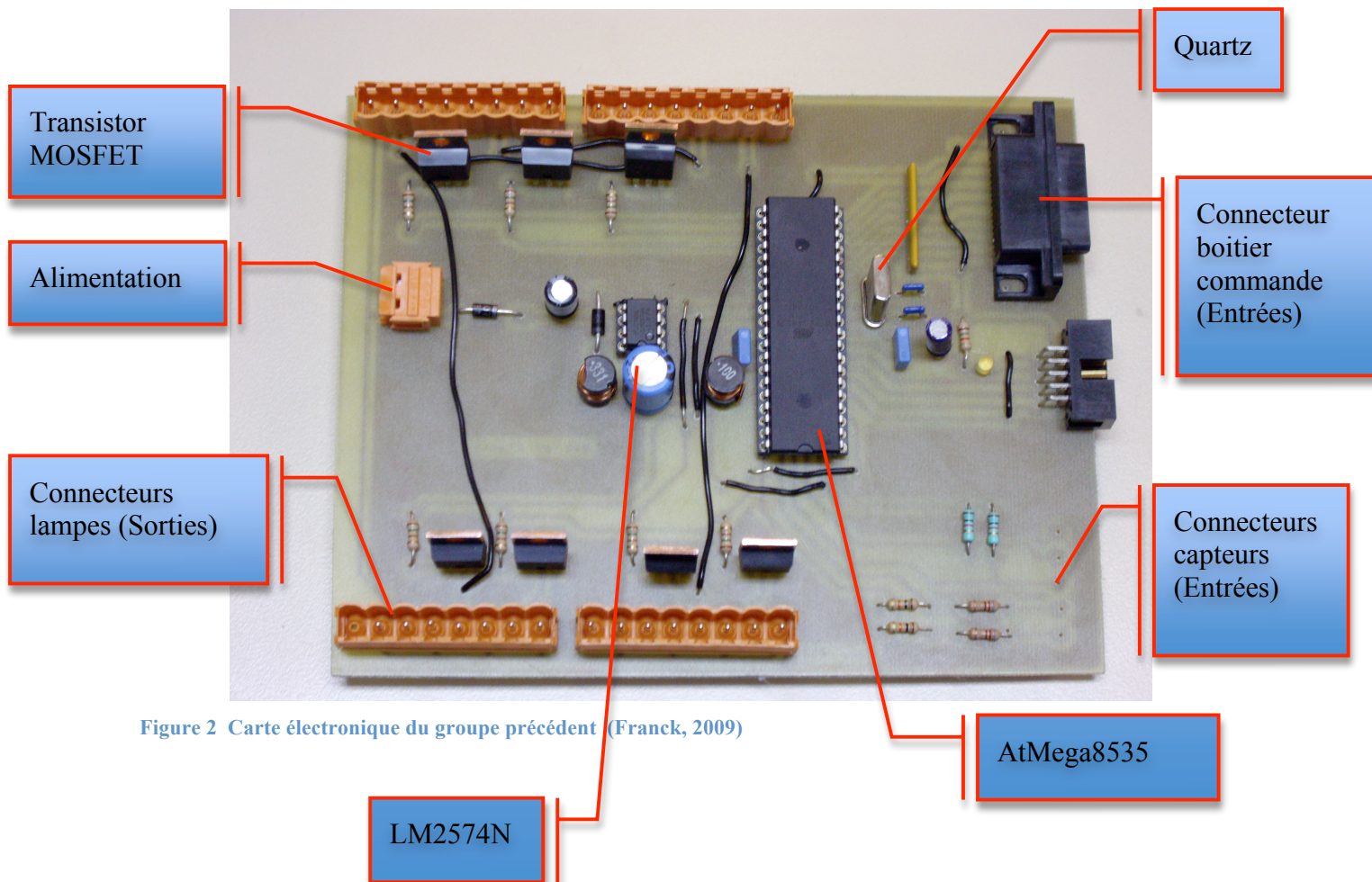


Figure 2 Carte électronique du groupe précédent (Franck, 2009)

Alimentation : Le kart possède deux batteries de 12V, c'est sur ce connecteur que nous allons récupérer une tension d'alimentation de 12V.

LM2574 : Le microcontrôleur AtMega8535 doit être alimenté en +5V, il est donc nécessaire d'abaisser la tension de 12V à 5V. Pour ce faire un montage abaisseur de tension de type buck a été réalisé grâce à l'utilisation d'un régulateur de tension à découpage, le LM2574N. Ce type de montage réduit les pertes de puissances qu'il y aurait eu si un composant linéaire tel qu'un régulateur avait été utilisé.

L'AtMega8535 : C'est un microcontrôleur 8 bits qui se présente sous la forme d'un circuit intégré de 40 broches. Il possède de nombreuses entrées/sorties ainsi que plusieurs fonctions utiles pour la suite du projet (explication plus bas). C'est à l'intérieur de celui-ci que nous implanterons le programme.

Quartz : C'est un composant qui possède la propriété d'osciller à une fréquence stable lorsqu'il est électriquement stimulé. Ce sont les propriétés piézoélectriques du minéral de quartz qui lui permettent d'obtenir une fréquence d'oscillation très précise. Le quartz utilisé est cadencé à 16 MHz, il permet de remplacer l'horloge interne de l'AtMega qui est cadencée à une fréquence moins élevée.

Transistors MOSFET : Il sert d'interrupteur de sortie, en effet il possède 3 pattes (Grille(1), Source(3), Drain(2)), la grille est reliée directement à l'AtMega et recevra des impulsions, le drain est relié à une des patte d'une lampe (l'autre pate étant reliée au +12V) et la source rejoint la masse. Lorsque la grille reçoit une impulsion, le transistor se ferme et le drain rejoint la masse, ce qui permet à l'ampoule d'avoir une tension de 12V à ses bornes (lampe : une patte au +12V et une patte à la masse). Dans le cas contraire la lampe est éteinte puisque une de ses patte est en l'air (lampe : une patte au +12V et une patte en l'air).

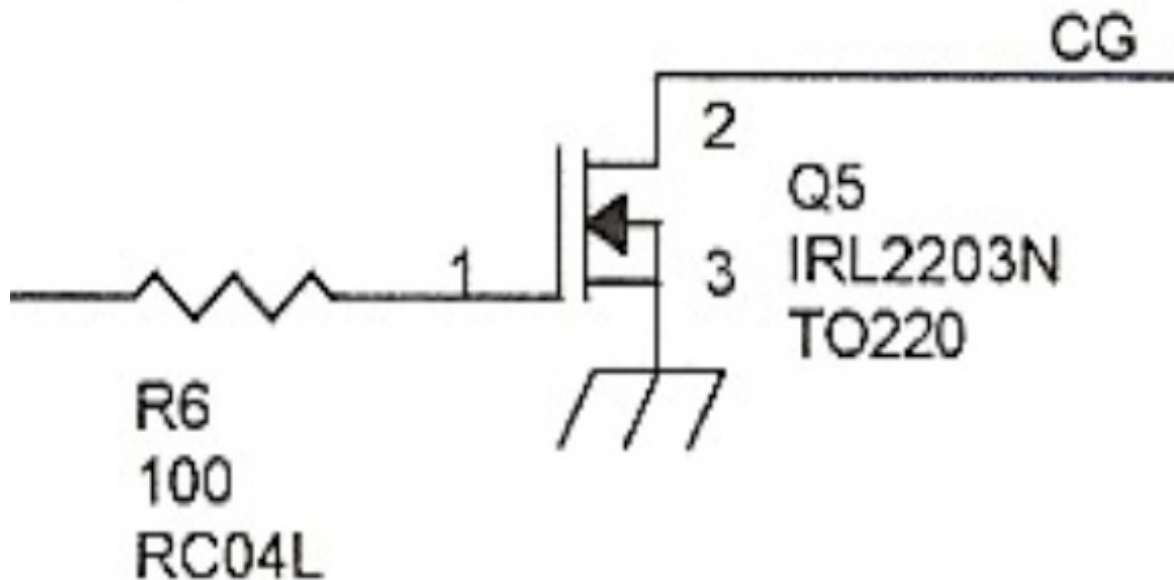


Figure 3 Schéma du transistor MOSFET

Connecteurs de sorties : Ces connecteurs seront directement branchés aux LED et permettront la liaison entre les LED et la carte.

Connecteurs d'entrées : Connecteurs branchés au boîtier de commandes et aux capteurs, ils permettront la liaison entre les interrupteurs, les capteurs et la carte.

2.3 Le microcontrôleur AtMega8535

L'AtMega8535 est le microcontrôleur 8bits qui a été choisi afin de mener à terme ce projet. Il se présente sous la forme d'un circuit intégré de 40 broches (voir ci-dessous).



Figure 4 Illustration de l'AtMéga 8535 (Chinaimportexport)

2.3.1 Description des broches

L'AtMega8535 contient quatre ports ayant chacun des fonctions particulières et contenant 8 bits directionnels numérotés de 0 à 7. Pour chacun de ces ports les pattes peuvent être configurées comme entrée ou bien comme sortie, en effet elles sont reliées à des résistances de tirages, internes au circuit intégré. En plus de ces ports, l'AtMega8535 possède plusieurs autres broches que nous allons lister et présenter ci-dessous:

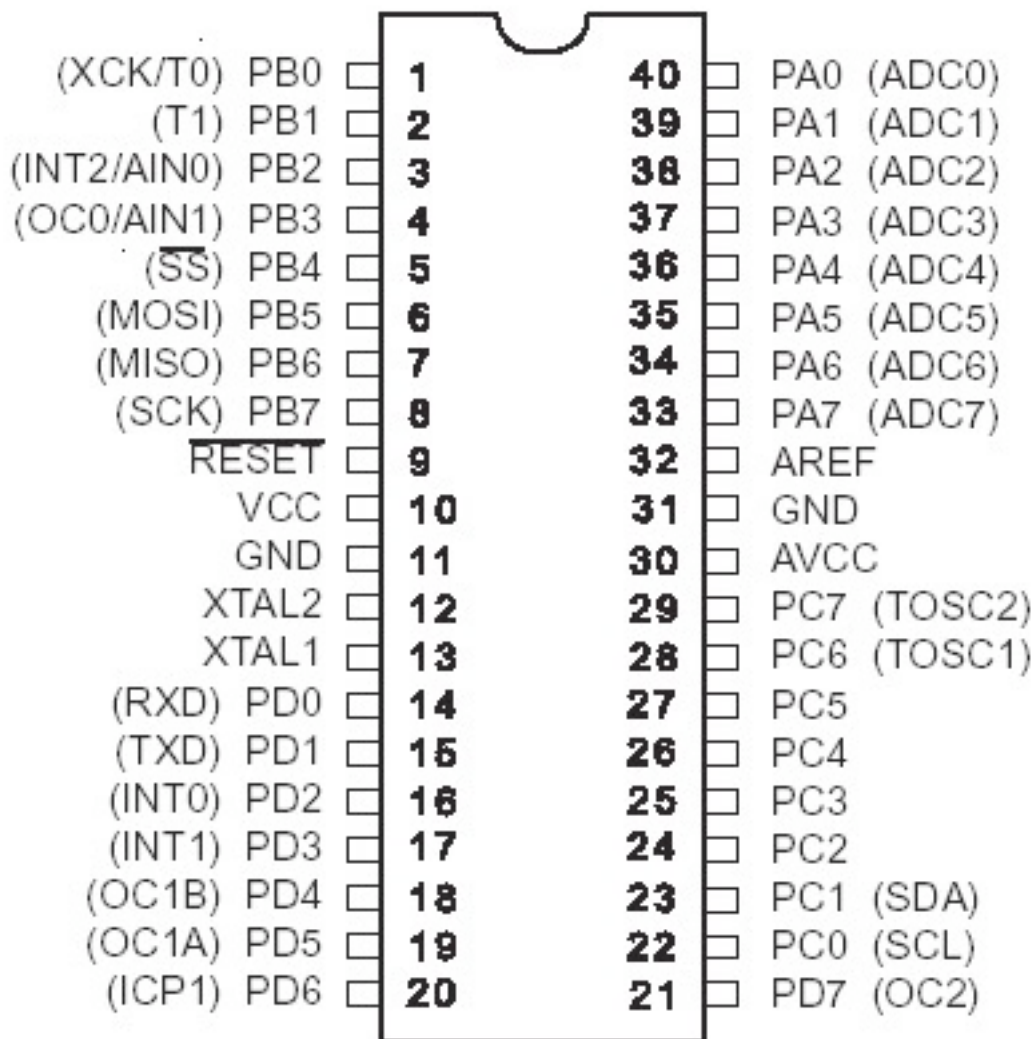


Figure 5 Brochage de l'AtMega 8535 (PADIOLLEAU)

-RESET : Elle permet la réinitialisation du microcontrôleur.

-XTAL1: Entrée de l'oscillateur externe pour l'horloge interne (quartz de 16MHz dans notre cas).

-XTAL2 : Production de l'amplificateur d'oscillateur.

-AVCC : C'est une broche d'alimentation pour le CAN1, qui doit être reliée à VCC par le biais d'un filtre passe-bas pour éviter les parasites.

-AREF : C'est l'entrée de référence analogue pour le CAN.

-AGND : C'est une masse analogique.

-VCC : C'est la broche d'alimentation du microcontrôleur (entre 3 et 5V, 5V dans notre cas).

-GND : C'est la masse de l'alimentation.

2.3.2 Description des fonctions principales

🍎 L'horloge:

Dans le cas présent, le signal d'horloge est réalisé grâce à la présence d'un quartz entre les entrées XTAL1 et XTAL2 (voir figure ci-après).

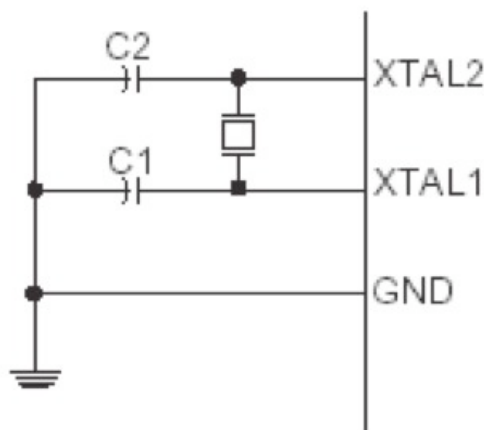


Figure 6 Schéma électrique du quartz (SAURET, 2009)

Le quartz est un composant qui possède la propriété d'osciller à une fréquence stable lorsqu'il est électriquement stimulé. Ce sont les propriétés piézoélectriques² du minéral de quartz qui permettent d'obtenir des fréquences d'oscillation très précises. Les deux condensateurs constituent des filtres sélectifs de retour et c'est cette horloge ainsi réalisée qui nous permettra d'utiliser la fonction Timer. Le quartz utilisé est cadencé à 16MHz, il permet de remplacer l'horloge interne qui est cadencée à une fréquence moins élevée.

Timer :

Le Timer est un compteur, selon son entrée, il peut compter du temps ou des événements. Il peut ainsi servir de base de temps, de compteur ou de générateur MLI. L'AtMega8535 en possède trois.

🍎 Les Convertisseurs Analogique Numérique :

Un convertisseur analogique numérique est un montage électronique dont la fonction est de générer une valeur numérique à partir d'une valeur analogique codée sur plusieurs bits étant proportionnelle à la valeur analogique d'entrée. Les convertisseurs analogique numérique de l'AtMega8535 sont situés sur le port D, ils peuvent donner un résultat numérique codé sur 10 bits par l'équation suivante :

$$\text{Résultat numérique} = (\text{Tension d'entrée} / \text{Tension de référence AREF}) \times 1024$$

🍎 Les entrées-sorties logiques :

Ces entrées ou sorties (selon la configuration du microcontrôleur) sont des interrupteurs tout-ou-rien, en effet quand elles reçoivent une valeur de tension, elles sont mises à l'état haut, de même quand elles sont réglées à un niveau haut au sein

du microcontrôleur, elles délivreront la tension VCC. Lorsqu'elles sont à l'état bas cela signifie que la tension délivrée ou reçue est de 0V.

Un état haut correspond à une valeur numérique de 1 bit (0 bit pour un état bas).

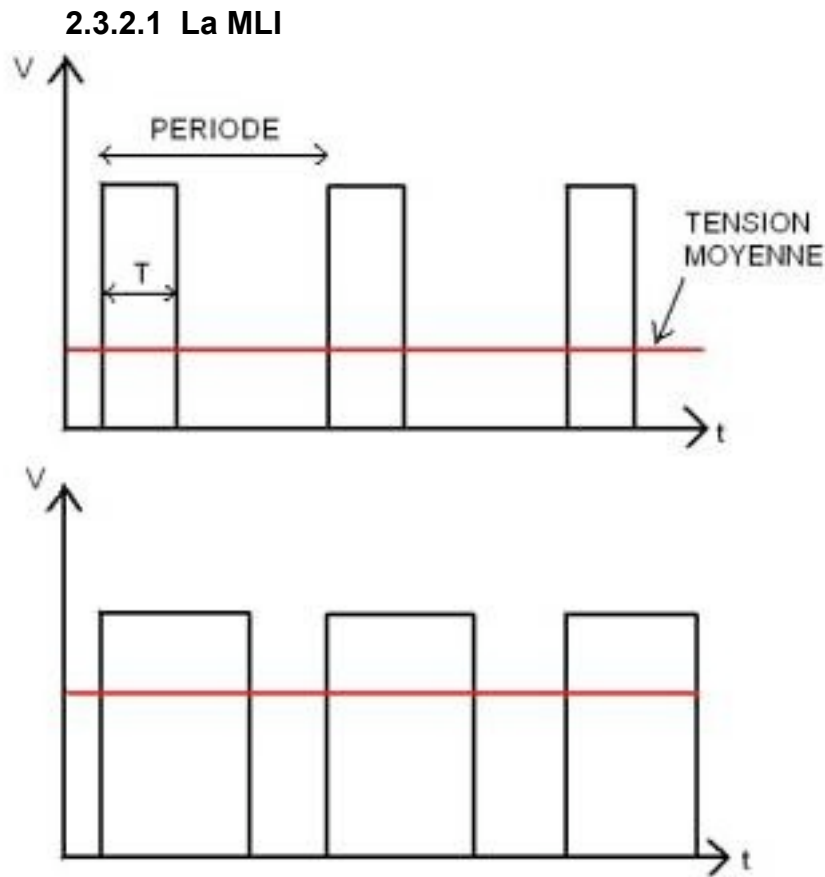


Figure 7 Illustration d'un signal MLI (coucou28, 2009)

La MLI (Modulation en Largeur d'Impulsion) ou PWM issu des initiales de Pulse Width Modulation en anglais, est tout simplement un moyen de générer une tension moyenne variable à partir d'une tension continue fixe (alimentation), en modulant le rapport cyclique d'un signal périodique.

Comme vous pouvez le voir sur le chronogramme (illustration 7), plus la durée T est importante plus la tension moyenne (en rouge) résultante est élevée.

Le calcul $T/\text{Période}$ est appelé rapport cyclique, il est souvent exprimé en pourcentage. Quand celui-ci vaut 0% alors la tension est nulle, quand il vaut 100% alors la tension moyenne est maximale.

2.4 Etude et mise en place des capteurs

Un capteur est un dispositif qui transforme l'état d'une valeur physique observable en une valeur utilisable. Il se distingue de l'instrument de mesure par le fait qu'il ne s'agit que d'une simple interface entre un processus physique et une information manipulable.

2.4.1 La photorésistance

C'est un composant électronique dont la résistivité varie en fonction de l'intensité lumineuse. Dans notre cas, sa valeur diminue avec l'accroissement de lumière (30 KOhms en pleine obscurité).



2.4.2 Le potentiomètre mécanique

C'est un composant électronique dont la résistivité varie en fonction d'un mouvement mécanique. Dans notre cas, sa valeur diminue avec l'appui sur une pédale, plus l'appui est important plus la valeur est faible (La valeur de départ est de 10 KOhms)



Figure 8 Photo du potentiomètre mécanique (Richard)

Afin de pouvoir utiliser les valeurs Ohmiques obtenues par les capteurs, nous réaliserons un pont diviseur de tension. Ce dernier nous permettra d'avoir une tension image de la résistivité de chacun des capteurs.

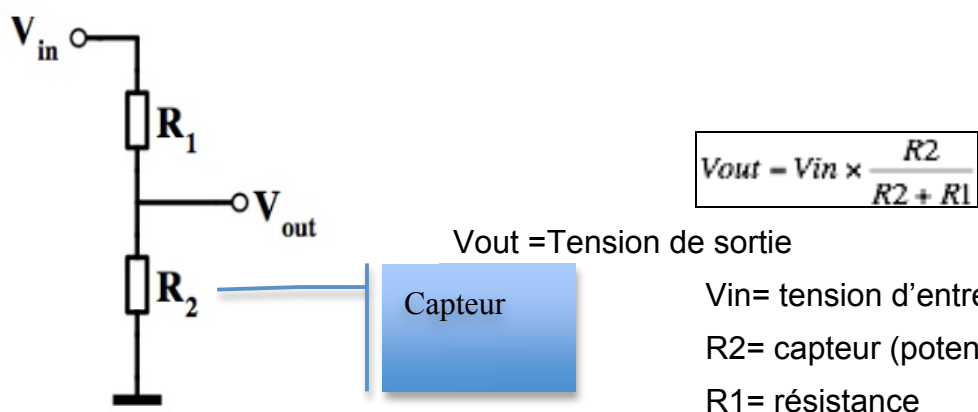


Figure 9 Schéma électrique d'un pont de résistances (Toublanc)

Nous souhaitons obtenir une tension proche de 5V (tension image) lorsque le potentiomètre à une résistance de

10 KOhms, et nous avons une tension d'entrée de 5V. Il sera donc nécessaire d'avoir une résistance R1 la plus petite possible (soit 100 Ohms) pour se rapprocher de la tension souhaitée.

Vout sera relié avec les entrées du convertisseur analogique numérique de l'AtMega8535, de cette manière nous pourrons numériquement utiliser sa valeur et donc faire varier l'éclairage des feux avant en mode automatique et des feux stop lors d'un appui sur la pédale de frein.

(Gestion des coefficients dans le programme, voir plus bas)

3 Réalisation du programme

Notre programme étant trop volumineux pour tenir dans le dossier il sera joint en annexe.

3.1 Organigramme du programme

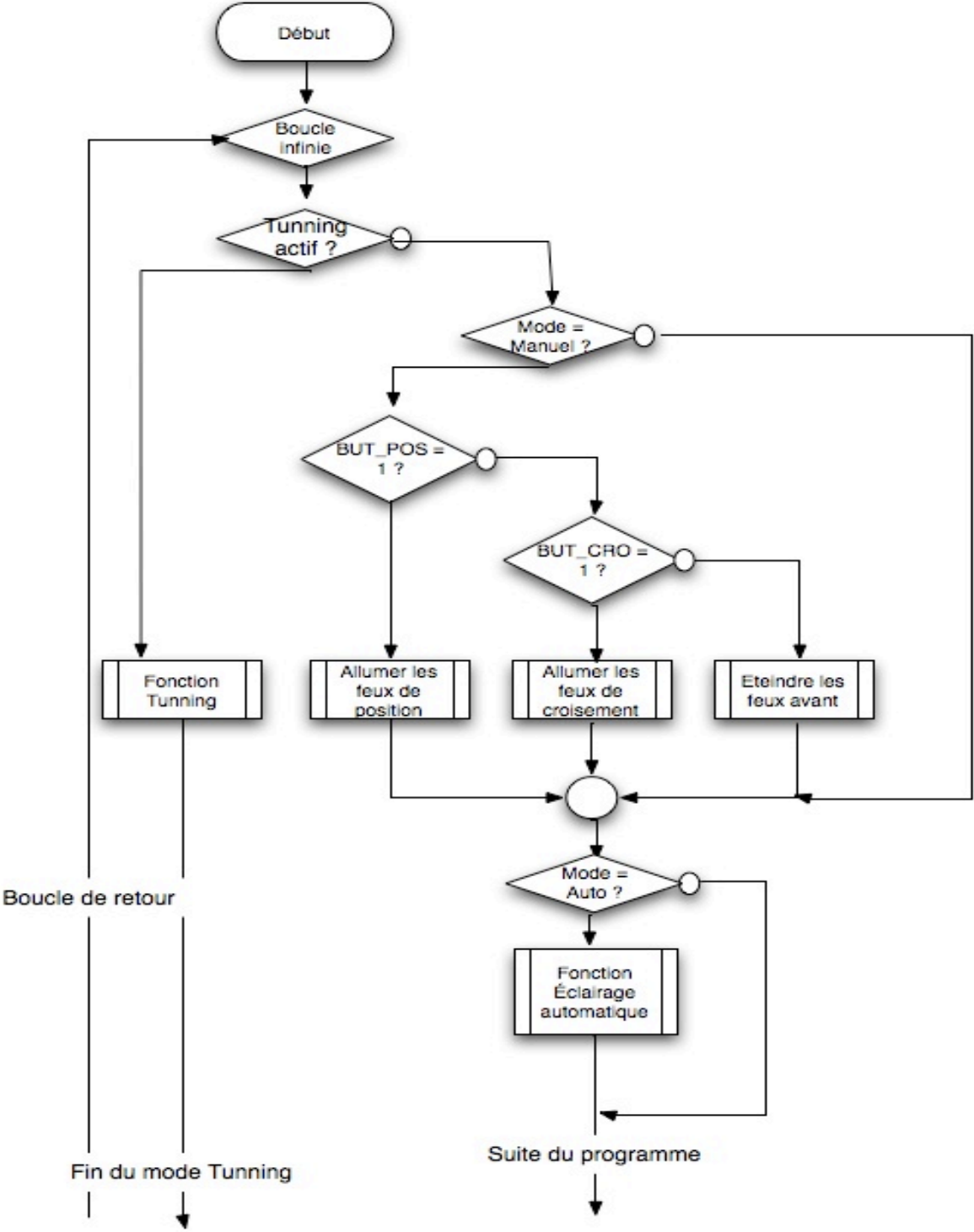


Figure 10 Organigramme Partie 1 (Toublanc)

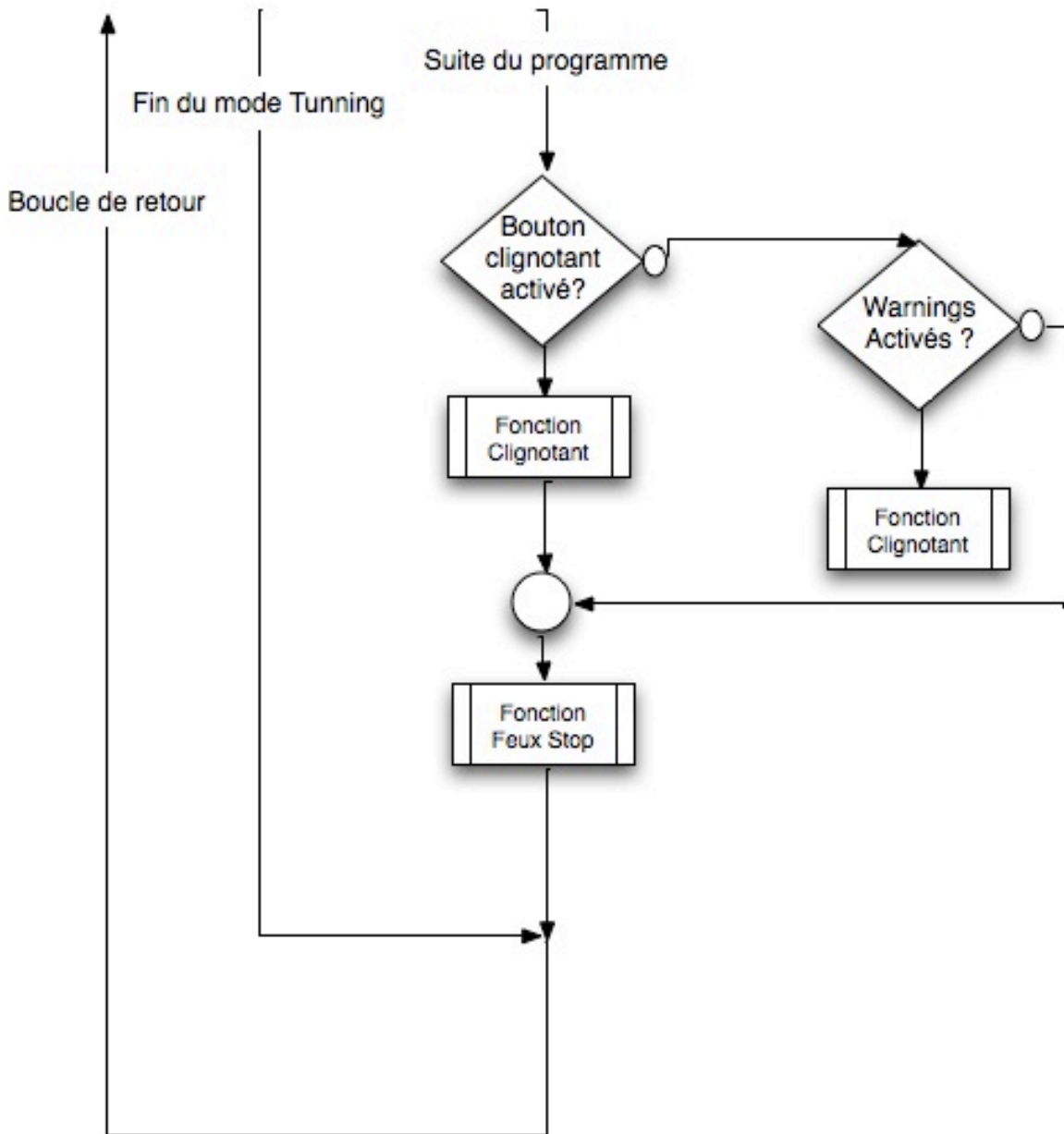


Figure 11 Organigramme Partie 2 (Toublanc)

Voici l'organigramme décrivant les actions du programme que nous avons créé.

3.1.1 Description d'un organigramme

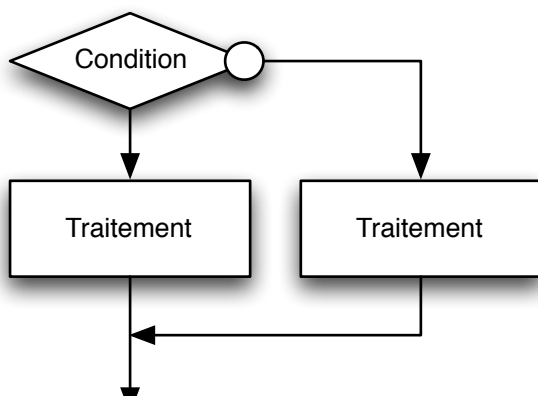


Figure 12 Illustration d'une structure condition (Toublanc)

Ceci est une alternative simple, elle permet d'effectuer un traitement seulement si la variable possède une condition.

Ainsi nous pouvons exiger L'appui d'un bouton pour allumer les phares ou les warnings.

Si la condition est fausse le second traitement s'effectuera.

Exemple de programme :

```
If (Condition)
{
  Traitement 1;
}
else
{
  Traitement 2 ;
}
```

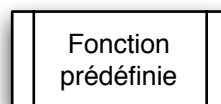


Figure 13 Illustration d'un bloc fonction (Toublanc)

Ceci est un appel de fonction ainsi nous avons pu regrouper certaines actions effectuant une tâche définie par exemple « allumer les phares » est une série de traitement.

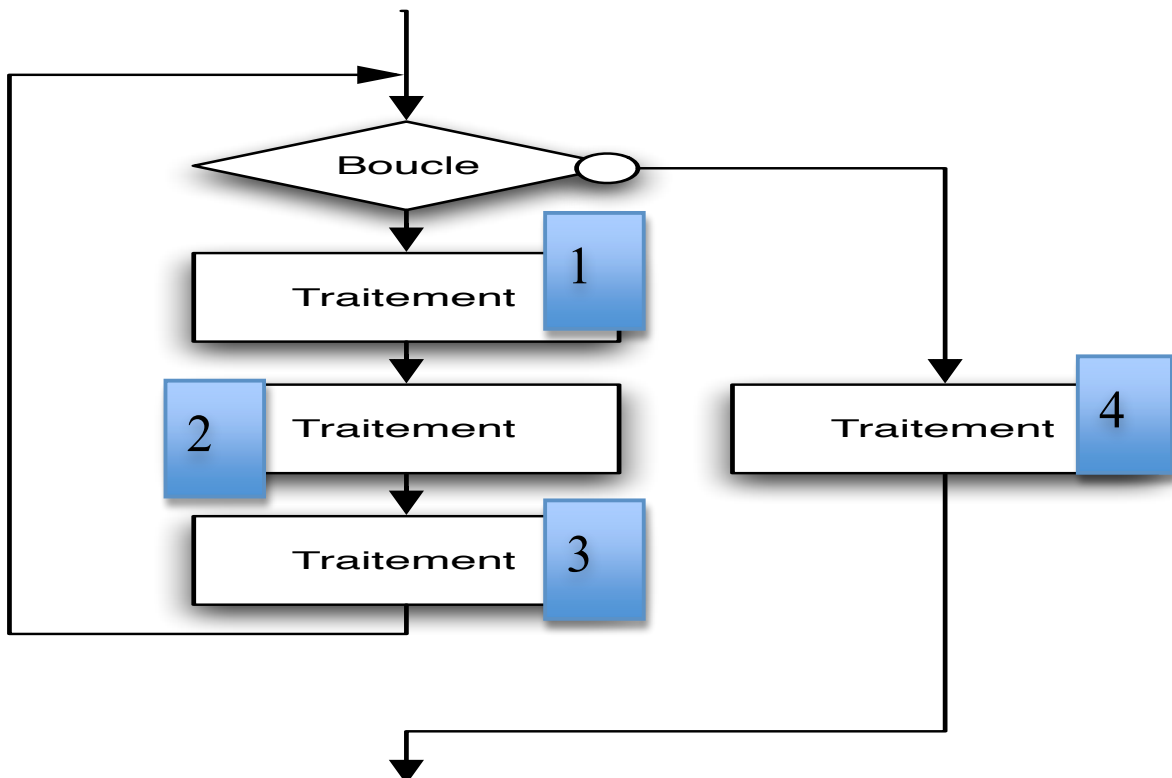


Figure 14 Illustration d'une structure Boucle (Toublanc)

Voici un exemple de boucle. Une boucle consiste à refaire une ou plusieurs actions tant que la condition de boucle est fautive. Ainsi nous avons utilisé une boucle pour permettre à notre programme de ne jamais s'arrêter (Boucle Infinie).

```
Exemple de code :  
While (Condition)  
{  
    Traitement1;  
    Traitement2 ;  
    Traitement3 ;  
}  
Traitement4 ;
```

3.1.2 Explication de notre organigramme

L'organigramme présent ci dessus nous explique que le programme sera géré par une boucle infinie. Puisque le cahier des charges nous impose d'assurer le fonctionnement de l'éclairage sans interruption.

La seconde chose que nous pouvons apprendre de cet organigramme est que pour avoir un fonctionnement classique il faut que le bouton « Tuning » ne soit pas activé. Alors nous pourrons choisir entre le mode Manuel et le mode Automatique grâce à un interrupteur.

En mode manuel les phares pourront être allumés selon le bon vouloir de l'utilisateur contrairement au mode automatique qui empêchera l'activation manuelle des phares : ils seront allumés automatiquement en fonction de la luminosité extérieure grâce à la photorésistance.

3.2 Fonctions utilisées

3.2.1 Programmation des ports E/S

```
// 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=Out Func2=In Func1=In Func0=In  
// State7=T State6=T State5=T State4=T State3=0 State2=T State1=T State0=T  
PORTB=0x00;  
DDRB=0x08;  
  
// 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;
```

```
DDRD=0x80;
```

Grâce à ces lignes de code en C nous avons pu paramétrer les ports du Microcontrôleur. Ainsi le port A est paramétré en entrée (IN), le port D l'est aussi sauf la broche 7. Les broches 0, 1, 2, 4, 5, 6, 7, du port B sont en entrées et la broche 3 est en sortie. Finalement le port C sera paramétré en sortie.

3.2.2 Programmation de la fonction MLI

Afin de pouvoir utiliser un éclairage variable pour les phares du karting nous avons utilisé la fonction MLI du microcontrôleur. Cette fonction sera aussi utilisée pour l'éclairage variable des feux de stop.

```
// Timer/Counter 0 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: 16000,000 kHz
```

```
// Mode: Phase correct PWM top=FF
```

```
// OC0 output: Non-Inverted PWM
```

Les lignes de codes ci-dessus sont générées par le logiciel CodeVision AVR. Elles nous indiquent que le timer 0 est programmé par rapport à l'horloge interne du microcontrôleur qui est cadencée à 16 MHz. Ces lignes nous indiquent aussi que le timer sera en mode PWM (Pulse Width Modulation) qui n'est autre que le mode MLI (Modulation de largeur d'impulsions).

```
TCCR0=Y;
```

```
TCNT0=0x00;
```

```
OCR0=0x00;
```

Ces lignes sont aussi générées automatiquement, pour pouvoir faire varier le rapport cyclique du signal de sortie il nous suffira de changer la valeur du registre OCR0 :

En effet le registre OCR0 est le registre qui fera varier la valeur du rapport cyclique. Il est codé sur 8 bits (1 Octet) ainsi il peut varier de 0 (0V) à 255 (5V).

Le mode MLI ne suffit pas à faire varier les feux en fonction des capteurs. Il nous faut en un premier temps acquérir les signaux des capteurs. Nous avons donc utilisé le mode CAN du microcontrôleur.

3.2.3 Programmation du CAN (Convertisseur Analogique Numérique)

Les signaux issus des capteurs sont des signaux analogiques nous ne pouvons donc pas les faire entrer sur une broche d'entrée/sortie numérique de l'AtMega 8535. Il nous avons donc utilisé le mode CAN (ADC en anglais pour Analogic Digital Converter) du composant programmable.

```

// ADC initialization
// ADC Clock frequency: 125,000 kHz
// ADC Voltage Reference: AREF pin
// ADC High Speed Mode: On
// ADC Auto Trigger Source: None
// Only the 8 most significant bits of
// the AD conversion result are used
ADMUX=ADC_VREF_TYPE;
ADCSRA=0x87;
SFIOR&=0xEF;
SFIOR|=0x10;

```

Voici les lignes de code générées par le logiciel, elles servent à initialiser la fonction convertisseur et nous indiquent que le convertisseur est cadencé à 125 KHz, que le CAN prend sa référence de tension sur la broche « AREF » (broche 32) et que le convertisseur est codé sur 8 bits ou 1 Octet.

Pour utiliser le convertisseur nous avons utilisé une fonction de l'AtMega : read_adc(Numéro de broche). La photorésistance (pour l'éclairage avant est brancher sur l'entrée 2 des convertisseurs et le potentiomètre mécanique est connecté sur le convertisseur numéro 3. Ainsi pour obtenir les informations des capteurs il suffit d'entrer ces lignes de code :

```

read_adc(2) ; // Photorésistance
read_adc(3) ; // Potentiomètre mécanique

```

3.2.4 Définition des symboles

Nous utilisons une référence du langage C, le #define. Celui-ci nous permet d'attribuer un mnémonique à une entrée ou à une sortie afin de rendre le programme plus lisible.

Tableau 1 Liste des symboles d'entrée (Franck, 2009)

Variables	Pâtes	Mnémoniques	Désignations
Entrées	PD6	TUNING	Interrupteur mode Eclairage/Tuning
	PD0	BUT_MOD	Interrupteur mode Manuel/Automatique
	PD5	BUT_POS	Interrupteur feux de position
	PD4	BUT_CRO	Interrupteur feux de croisement
	PD1	BUT_WAR	Interrupteur Warning
	PD3	BUT_CLG	Interrupteur clignotant gauche
	PD2	BUT_CLD	Interrupteur clignotant droit
	PA0		Tension indiquant la marche avant
	PA1		Tension indiquant la marche arrière

	PA2	EVAR	Tension récupérée de la photorésistance
	PA3	FREIN	Tension récupérée du potentiomètre mécanique (frein)

Tableau 2 Liste des symboles de sortie (Franck, 2009)

Sorties	PC0	FR	Sortie feux de recul
	PC1	EA	Sortie éclairage arrière
	PC6	CG	Sortie pour les clignotants gauches
	PC7	CD	Sortie pour les clignotants droits
	PB3 (Timer OC0)	AV	Sortie pour l'éclairage avant
	PD7 (Timer OC2)	FS	Sortie pour les feux stops
	PA4		Sortie LEDs bleues
	PA5		Sortie LEDs vertes

Les tensions indiquant la marche avant et la marche arrière ne nous sont pas fournies, par conséquent nous n'avons pas donné de mnémonique à ces entrées.

Les LEDs bleues et vertes ont été rajoutées lors de la mise en œuvre de la carte sur le kart électrique pour Paris Tuning Show, ainsi nous n'avons pas eu le temps de leurs attribuer des symboles.

Les symboles sont attribués, avant le programme principal et après la définition des librairies, par la commande :

```
#define TUNNING PIND.6
#define FR PORTC.0
```

Acronyme PIND.6 correspond à la broche 7 (Bit numéro 6) du port D paramétrée en entrée.

Acronyme PORTC.0 correspond à la broche 1 (Bit numéro 0) du port A paramétrée en sortie.

3.3 Programmation des phares

3.3.1 Analyse théorique

Pour obtenir différent pallier de tension à la sortie du montage il nous est nécessaire d'utiliser le mode MLI du microcontrôleur :

En mode manuel nous voulons deux pallier de tension afin de d'obtenir les feux de position et les feux de croisement. En mode automatique nous avons décidé de réaliser trois paliers de tensions car cela simule les trois modes de luminosité des phares d'une voiture.

Afin d'effectuer ces paliers la première étape consiste à acquérir la tension aux bornes de la photorésistance. Nous avons donc utilisé le CAN du composant programmable. Nous obtenons alors un nombre image de la tension codée sur 8 bits (0 (0V) à 255 (5V)).

L'AtMega 8535 est capable de fournir une tension de sortie au maximum de 5V. Afin d'élever cette tension nous avons utilisé des montages Tout-Ou-Rien. (Cf 2.2 Carte électronique Transistor MOSFET). Le groupe précédent a fait des mesures sur les lampes et a déterminé que pour les paliers il fallait obtenir 8V, 10V, et 12V sur les phares. Si nous rapportons ces valeurs aux tensions présentes avant le montage transistor ces paliers sont de : 3.4V, 4.2V, 5V. Par le rapport de proportionnalité entre les valeurs analogiques et les valeurs numériques on obtient les paliers numériques suivant : 173, 214 et 255. Une fois ces paliers obtenus nous pouvons les utiliser dans la programmation.

Pour le mode automatique il a fallu détecter la luminosité de la salle nous avons donc effectué des tests avec la photorésistance en changeant les valeurs numériques (après le CAN) dans le programme nous avons donc pu déterminer les plages de luminosité :

0<N<125 → les phares sont éteints
125<N<173 → les feux de positions sont allumés
173<N<220 → les feux de croisement sont allumés
N>220 → les feux de route sont allumés

3.3.2 Réalisation

Extrait du programme :

```
Ne=read_adc(EVAR) ; //Obtention de la Luminosité (Ne)
if (Ne>125) // 125 → luminosité minimale pour feux de position
{
LED PORTC.2=0; //PORTC.2 et PORTC.4 = broche assigné pour les plaques à
PORTC.4=0;
OCR0=173; // Palier numéro 1 (feux de position)
EA=1; // Eclairage Arrière
}
if( Ne>173) // 173 → luminosité minimale pour feux de croisement
{
PORTC.2=1;
PORTC.4=0;
OCR0=214; //Palier numéro 2 (feux de croisement)
EA=1;
}
if (Ne>220)// 220 → luminosité minimale pour feux de route
```

```

{
PORTC.2=1;
PORTC.4=1;
OCR0=255; //palier numéro 3 (feux de route)
EA=1;
}

else // sinon éteindre tout les feux
{
PORTC.2=0;
PORTC.4=0;
OCR0=0;
EA=0;}

```

3.4 Programmation des feux stop

3.4.1 Analyse théorique

Pour programmer les feux stop nous avons utilisé le même principe que pour les phares à la seule différence que nous n'utiliserons pas des paliers mais un rapport de proportionnalité pour une variation continue de la luminosité.

Les ampoules utilisées pour les feux stop sont constituées de LEDs il est donc nécessaire de prendre en compte la tension de seuil de 0,6V. Ainsi nous voulons que lorsque la tension d'entrée est nulle, une valeur numérique non nulle lui corresponde. Nous avons besoin de 0,6V minimum sur les lampes ce qui correspond à 0,25V avant le transistor MOSFET et 12,75 en valeur numérique.

$$N = a \cdot x + b \text{ avec } b = 12,75 \text{ et } a = (255 - 12,75) / (5 - 0) = 48,5$$

Grâce à cette formule nous avons compensé la tension de seuil des LEDs nous pouvons donc programmer les feux stop.

3.4.2 Réalisation

Extrait du programme :

```

float Nf ;
Nf=read_adc (FREIN); // Acquisition de la tension (potentiomètre)
Nf=Nf/51.0 ; //Conversion de Nf
OCR2=(48,5*Nf+12,75); //Allumage des frein en fonction du potentiomètre

```

La conversion de Nf consiste à ramener la valeur de Nf obtenue par le CAN (variant de 0 à 255) à une valeur comprise entre 0 et 5V.

3.5 Programmation des warnings et des clignotants

Pour programmer les warnings et les clignotant nous avons utilisé une suite de condition (if) et nous avons utiliser la fonction delay_ms(t) sui permet de faire attendre le programme « t » millisecondes.

Extrait du programme :

```
if(BUT_WAR==0) //Si le bouton warnings est activé
{
  CD=1;CG=1;// CD= Clignotant Droit CG= Clignotant gauche
  delay_ms(400);
  CD=0;CG=0;
  delay_ms(400);
}
else //sinon
{
  if(BUT_CLD==0)//Clignotant droit ?
  {
    CD=1;
    delay_ms(400);
    CD=0;
    delay_ms(400);
  }

  if(BUT_CLG==0) //Clignotant gauche ?
  {
    CG=1;
    delay_ms(400);
    CG=0;
    delay_ms(400);
  }
}
```

Comme demandé dans le cahier des charges nous avons programmé de tel sorte à ce que les warnings soient prioritaires sur les clignotants : nous testons en premier le bouton warning et s'il n'est pas activé alors nous testons les clignotants.

3.6 Mode Tuning

Il nous a été demandé d'insérer un « mode tuning » au projet afin que le kart puisse participer au PTS09 (salon du tuning à Paris).

Ce mode est activé par l'appuie sur un interrupteur, il est prioritaire sur tous les autres, en effet, une fois enclenché, toutes les fonctions de l'éclairage et de la signalisation sont désactivées



Figure 15 Photo du kart (Toublanc)

Pour rendre ce mode plus sympathique nous nous sommes servis des interrupteurs précédemment utilisés (éclairage et signalisation en mode manuel ou automatique) ; ces derniers permettrons de faire le choix entre différentes séries d'allumages des feux, en effet nous avons conçu plusieurs fonctions de type « tuning », ces fonctions sont référencées dans le tableau ci-dessous.

Tableau 3 Liste des fonctions tuning (Franck, 2009)

Interrupteurs	Fonctions	Effets
Bouton Croisement	K2000	Chenillard sur les feux arrières
Bouton Position	Speedy	Tourbillonnement en crescendo des feux arrières
Clignotant droit	Infernito	Allumage des feux arrières en diagonale

4 Finalisation du projet

4.1 Cartes à LEDs

Deux cartes contenant chacune un nombre important de LEDs ont été rajoutées sur le kart. Afin de pouvoir les faire fonctionner, nous avons dû modifier la carte électronique du groupe précédent en rajoutant deux transistors MOSFET et en utilisant deux nouvelles sorties sur l'AtMega8535. Précisons que la gestion de l'éclairage de ces LEDs ne se fera pas de manière indépendante pour chacune des cartes, en effet chaque carte contient deux séries de LEDs, un contour de LEDs bleues et un intérieur de LEDs vertes. Chaque transistor permettra l'éclairage d'une couleur sur les deux cartes (droite et gauche). Afin de se servir de ces cartes, nous avons réalisé un programme les prenant en compte, celui ci permet l'alimentation des LEDs de contours en feux de positions et de toutes les LEDs en feux de croisements.

4.2 Warnings et clignotants

Lors de l'utilisation pratique de l'éclairage, nous nous sommes aperçus que les clignotants et les warnings ralentissaient le fonctionnement, en effet la fonction utilisée fait attendre le programme. Pour éviter de bloquer ce dernier pendant 400 millisecondes il serait nécessaire d'utiliser des interruptions systèmes en tant que compteur. En effet les interruptions s'effectuent en parallèle avec l'exécution du programme principale.

4.3 Boitier de commande

Afin de finaliser ce projet, nous nous sommes intéressés au boitier de commande. Le boitier fournis était terne et sans grand intérêt, nous avons donc décidé de l'améliorer.

Pour ce faire nous avons commandé des interrupteurs dignes de ce nom d'une part, puis nous avons repensé totalement le boitier d'autre part. L'idée était d'ajouter des indications retro éclairées de deux couleurs différentes, l'une pour l'éclairage et la signalisation et l'autre pour le mode tuning afin que l'utilisateur s'y retrouve sans mal. Puis, nous aurions ajoutés des LEDs vertes sur le boitier afin d'indiquer au conducteur quel clignotant est allumé ou alors si il s'est mit en warning (tel le panneau de bord d'une voiture). Ces améliorations entraînent le montage d'une nouvelle carte électronique avec la gestion d'un nombre plus important d'entrées-sorties nécessaire pour l'éclairage du boitier.

Suite aux grèves cette partie n'a pu être achevé.

4.4 Problème et difficultés

Lors de ce projet nous avons rencontré plusieurs problèmes que nous allons énuméré ci-dessous :

Le boitier de commande étant mal câblé par rapport a ce qui avait été prévu par le groupe précédent (entrées du connecteur et de l'AtMega8535), nous avons dû réétudier la carte électronique. (Remercions malgré cela l'étudiant qui s'est dévoué pour monter tous les interrupteurs).

Contraintes de temps pour terminer le projet avant le PTS09.

Avec le rajout des cartes à LEDs nous avons eu des difficultés pour modifier la carte électronique dans les temps.

Problème avec la programmation, en effet à chaque fois que le montage n'est plus alimenté, nous perdons la gestion des MLI par le microcontrôleur, ce qui reste un grand mystère.

Planning

Taches / Semaines	4	5	6	7	8	9	10	11	12	13	14	15
Etude du projet précédent	X								O	O		
Recherches des composants	X	X							O	O		
Etude du logiciel Code Vision		X	X						O	O		
Réalisation du boitier de commande et adaptation								X	O	O		
Programmation			X	X	X				O	O		
Tests et Validations					X				O	O		
Rédaction du rapport									O	O	X	X
Préparation de la présentation									O	O	X	
Implantation sur le Karting					X				O	O		
			Vacances						O	Grève		
			Planning prévisionnel						X	Planning Réel		

Conclusion

La problématique était de finaliser le projet : éclairage d'un kart électrique de l'IUT GEII de Tours, nous avons donc établi un cahier des charges et tenté de mener à terme ce projet essentiellement informatique.

Ce projet nous a permis de découvrir le logiciel CodeVision AVR ainsi que toutes les fonctionnalités courantes disponibles sur le microcontrôleur AtMega8535.

La finalisation de ce projet a presque été menée à son terme, en effet toutes les parties éclairage, signalisation et tuning ont été testées et utilisées lors du salon du tuning à Paris (PTS09). Seul reste à terminer le montage du boîtier de commande et quelques améliorations peuvent être apportées.

Lors de mise en œuvre d'un tel projet, plusieurs compétences sont nécessaires : le respect d'un cahier des charges, la recherche et l'apprentissage de nouvelles interfaces, le travail en équipe et la contrainte de durée, ce sont toutes ces contraintes qui font de ce travail pratique un entraînement primordial pour le stage.

Table des illustrations

Figure 1	Schéma synoptique général du projet	6
Figure 2	Carte électronique du groupe précédent (Franck, 2009)	8
Figure 3	Schéma du transistor MOSFET	9
Figure 4	Illustration de l'AtMega 8535 (Chinainportexport)	10
Figure 5	Brochage de l'AtMega 8535 (PADIOLLEAU)	11
Figure 6	Schéma électrique du quartz (SAURET, 2009).....	12
Figure 7	Illustration d'un signal MLI (coucou28, 2009).....	13
Figure 8	Photo du potentiomètre mécanique (Richard).....	14
Figure 9	Schéma électrique d'un pont de résistances (Toublanc).....	14
Figure 10	Organigramme Partie 1 (Toublanc).....	16
Figure 11	Organigramme Partie 2 (Toublanc).....	17
Figure 12	Illustration d'une structure condition (Toublanc)	17
Figure 13	Illustration d'un bloc fonction (Toublanc).....	18
Figure 14	Illustration d'une structure Boucle (Toublanc)	18
Figure 15	Photo du kart (Toublanc).....	26

Index des Tableaux

Tableau 1	Liste des symboles d'entrée	21
Tableau 2	Liste des symboles de sortie.....	22
Tableau 3	Liste des fonctions tuning	26

Bibliographie

- [1]Chinainportexport. (s.d.). Récupéré sur <http://www.chinainportexport.wikispaces.com/coucou28>. (2009, 04 03). Récupéré sur <http://coucou.28.free.fr/commandecc.htm>
- [2]Franck, THYPHONNET
- [3]PADIOLLEAU, J.-L. (s.d.). Consulté le 04 03, 2009, sur http://sti.ac-orleans-tours.fr/spip/article.php3?id_article=98
- [4]Richard, Clément TOURS, 37, FRANCE.
- [5]SAURET, F. (2009, 04 03). Récupéré sur <http://anyedit.free.fr/telechargement/atmega8535-francais.pdf>
- [6]Toublanc, Kévin TOURS, 37, FRANCE.

Annexes