



# Rapport de projet tutoré 2<sup>ème</sup> année

## Modification du Kart électrique

*Commande de l'éclairage et modification du matériel*

MEUKENS Pierre  
TUILARD Benjamin  
Groupe Q2 – Promo 2009/2012

**Professeurs référents:**  
- AUGER Philippe  
- LEQUEU Thierry

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



# Rapport de projet tutoré 2<sup>ème</sup> année

## Modification du Kart électrique

*Commande de l'éclairage et modification du matériel*

MEUKENS Pierre  
TUILARD Benjamin  
Groupe Q2 – Promo 2009/2012

**Professeurs référents:**  
- AUGER Philippe  
- LEQUEU Thierry

# Table des matières

Remerciements.....	4
Introduction.....	5
1.Présentation du sujet.....	6
2.Analyse du projet.....	7
2.1.Étude de la maquette et problèmes rencontrés.....	7
2.2.Solutions envisagées et recherches effectuées.....	9
3.L'éclairage du Kart.....	17
3.1.La programmation.....	17
3.2.Le mode manuel.....	19
3.3.Le mode automatique.....	22
4.Optimisation matériel.....	30
4.1.Modification du Kart.....	30
4.2.Phrase de position avant.....	32
Conclusion.....	35
Résumé.....	36
Index des illustrations.....	37
Bibliographie.....	38
Annexes.....	39

## Remerciements

Nous souhaitons tout d'abord, avant de commencer notre rapport de projet tutoré de 2e année, à adresser des remerciements à certaines personnes qui nous ont aidé fortement durant ce projet.

Dans un premier temps nous remercions l'IUT GEII de nous offrir cette possibilité de pouvoir travailler sur un projet dans le cadre de notre formation. Celui-ci nous permet ainsi de mettre en œuvre quelque chose de concret et de pouvoir appliquer directement notre réflexion et nos connaissances sur un problème pour en trouver la solution.

Nous remercions ensuite Mr LEQUEU Thierry sans qui nous n'aurions pas pu travailler sur ce projet. Il a toujours su répondre à nos demandes.

Pour finir nous remercions Mr VAUTIER Richard qui nous a toujours accueilli dans le magasin. Il nous a grandement conseillé et aidé durant ce projet.

# Introduction

Dans le cadre du cours d'études et réalisation, nous avons réalisé un projet proposé par Mr Thierry Lequeu. Celui-ci portait sur l'éclairage du kart électrique de l'IUT.

Ce qu'il nous était demandé dans ce projet c'était de gérer les différents modes d'éclairage à savoir, un mode manuel basique qui va permettre d'allumer tous les feux du kart, et un mode automatique qui va, avec des capteurs, faire un éclairage progressif.

Nous allons donc voir dans ce dossier quelles améliorations nous avons pu apporter. Pour cela nous verrons dans un premier temps une présentation du sujet, des recherches que l'on a effectué ainsi que le cahier des charges et le planning lié à notre projet

Cette partie sera suivie de la partie programmation, comment nous avons résolu la gestion des différents modes d'éclairage.

Nous finirons par étudier les améliorations matérielles ajoutées au kart tout au long de notre projet avant de conclure sur celui-ci.

# 1. Présentation du sujet

Comme dit dans l'introduction, le but du projet est de travailler sur la partie informatique contrôlant l'éclairage du kart, le programme ainsi réalisé devra respecter les conditions suivantes:

- Récupérer les informations des commandes, et des capteurs
- Commander les différentes lampes à LED<sup>1</sup> correspondantes à l'action voulue
- Gérer la fréquence de clignotement pour les clignotants et feux de détresse

Cependant, même si ce n'était pas le but premier de ce projet nous avons effectué d'autres tâches sur le kart. A commencer par les commandes au volants qui doivent permettre d'envoyer les informations à la carte de gestions des feux. Tout en respectant quelques contraintes:

- Les commandes doivent être lisibles et rapides à retrouver
- Les câbles de commandes ne doivent pas gêner ni le pilote, ni les rotations du volant

Ensuite nous avons réalisé de nouvelles cartes pour remplacer les anciens feux de position, ces cartes doivent écrire IUT pour l'une et GEII pour l'autre tout en respectant les contraintes suivantes:

- La taille des cartes limitées à 10cm par 11 cm
- L'éclairage qui ne doit pas être trop puissant
- La carte doit être alimenté à 0/+12 V

Enfin on a aussi réalisé l'installation physique d'allume-cigare permettant de recharger le kart, ceux-ci doivent rester accessible, et être placés sur le « tableau de bord » du kart.

---

1 LED: Ligth Emmiting Diode ou Diode ÉlectroLuminescente (DEL)

## 2. Analyse du projet

### 2.1. Étude de la maquette et problèmes rencontrés

#### 2.1.1. Présentation de la maquette

Pour appréhender notre projet, nous avons à notre disposition une maquette disposant des différents phares montés sur le kart électrique.



*Illustration 1: Maquette d'étude de l'éclairage du kart, photo prise par les auteurs*



*Illustration 2: Boîtier de commandes des feux*

Cette maquette peut être décomposée en plusieurs parties :

- L'ensemble des phares avant du kart soit: les clignotants gauche et droite, feux de position, feux de croisement et feux de route.
- L'ensemble des phares arrière du kart soit: les clignotants gauche et droite, feux de stop, feux de témoin.
- Une carte électronique comportant l'ATmega8535
- Un boîtier de commandes des différents feux.

## 2.1.2. L'ATmega8535

L'ATmega8535 est un composant électronique programmable dit « micro-contrôleur ».

On remarque qu'il est composé de 32 ports d'entrées/sorties regroupés en 4 groupes comportant chacun 8 broches : PA, PB, PC et PD.

Le micro-contrôleur possède de nombreuses fonctions telles que un comparateur, un convertisseur, des horloges internes, des interruptions et autres fonctions, qui seront plus ou moins exploitées dans notre projet lors de la programmation des commandes des feux.

Grâce au conseil de Mr Lequeu nous avons utilisé l'outil « Code Wizard AVR » pour la programmation de nos commandes de feux, ce qui nous a permis de comprendre le fonctionnement du programme tout en continuant de réaliser le programme principal.

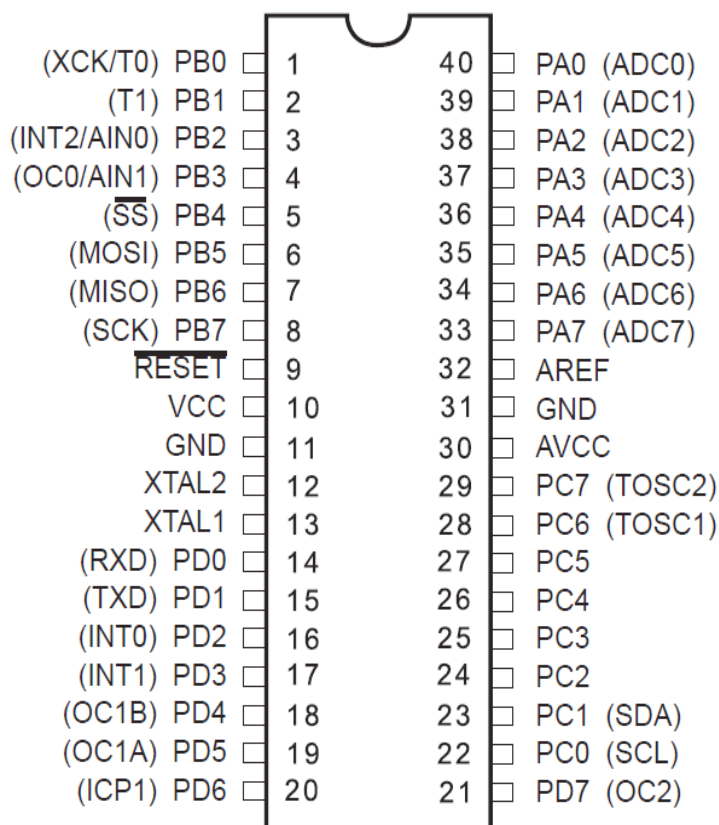


Illustration 3: Schéma de l'ATmega8535 issu de sa datasheet[1]



### **2.1.3. Premier test de la maquette**

Dans un premier temps afin d'appréhender la maquette, nous avons effectué un premier test de la maquette avec un programme déjà implanté dans l'ATmega8535.

Celui-ci nous a permis de mettre en évidence les premiers problèmes rencontrés :

- La luminosité des phares n'était pas maximale lorsqu'on était sur une commande progressive des feux,
- Le début de l'éclairage des feux Stop en progressif venait beaucoup trop tard,
- L'éclairage progressif n'était absolument pas linéaire.

### **2.2. Solutions envisagées et recherches effectuées**

La solution la plus évidente et la plus simple pour réaliser notre projet était de nous baser sur le travail déjà effectué sur la maquette. On a ainsi gardé la carte électronique avec l'ATmega8535 qui sera branchée sur le kart.

Nous avons ainsi listé les choses à faire :

- Faire l'étude complète de la maquette et de sa carte électronique,
- Étudier la documentation des composants tel que les capteurs et l'ATmega8535,
- Programmer la gestion des feux,
- Tester notre programme.

De cette liste nous avons alors créé un planning prévisionnel

### 2.2.1. Le planning

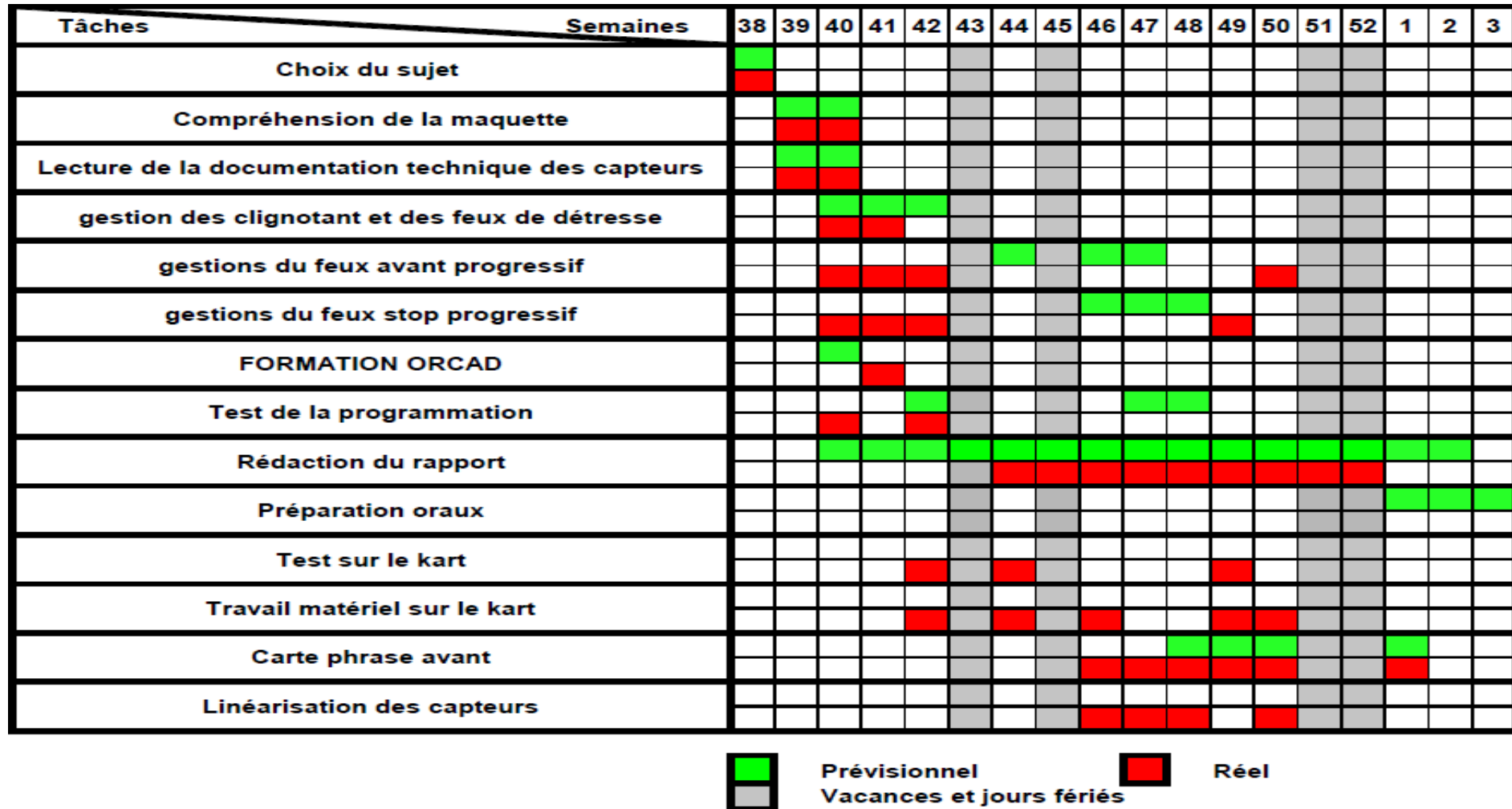


Illustration 4: Planning du projet réalisé par les auteurs

Comme on peut le voir sur ce planning nous avons prévu d'étaler la programmation sur la quasi totalité de la durée du projet, car nous jugions que c'était le temps qu'il nous fallait afin d'appréhender le fonctionnement de la programmation de l'ATmega 8535 et de réaliser le programme principal.

De plus, nous avons prévu de faire chaque parties sur différentes séances afin de pouvoir les tester indépendamment les unes des autres, mais au vu de la programmation très rapide et simple à comprendre on a pu tester chacune des fonctions en même temps.

Les tests se faisaient essentiellement sur une maquette où l'on retrouvait les différentes lampes à LED présentes sur le kart, mais par la suite nous avons pu réaliser les tests directement sur les lampes du kart.

Ainsi nous avons réalisé différents problèmes, tels qu'une mauvaise disposition des ampoules, des couleurs identiques pour les feux de croisement et les pleins phares, des LED grillées sur la carte des feux de positions, mais aussi que l'éclairage des feux progressifs n'étaient absolument pas linéaires.

Ces problèmes ont entraîné un remaniement du planning ajoutant des travaux annexes à notre projet.

Nous devons alors effectuer une linéarisation des capteurs afin que l'éclairage des feux progressifs soit complètement linéaire, changer les feux de positions soit créer de nouvelles cartes électroniques pour les feux de positions et pour finir nous avons décidé de mettre les commandes sur le volant du kart car nous pensions que cela serait plus simple que d'installer le boîtier de commandes sur le kart.

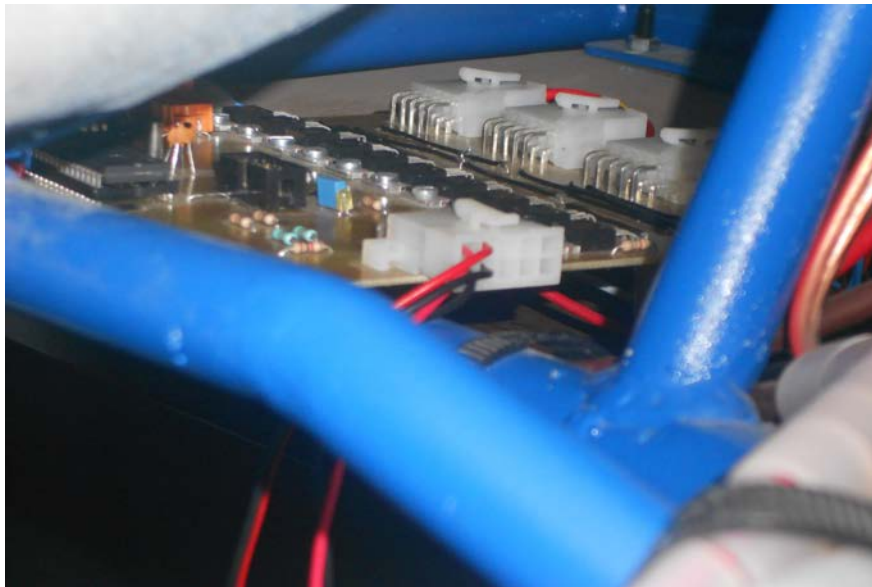
Notre premier travail a été d'étudier la carte électronique qui nous sert de support tout le long de notre projet.

### 2.2.2. Étude de la carte électronique

Comme nous l'avons dit précédemment la maquette comporte une carte électronique sur laquelle est implantée l'Atmega8535,

On peut voir différentes choses principales sur cette carte en dehors de l'Atmega8535 :

- **Un régulateur de tension à découpage (LM2574N)**: L'ATmega8535 fonctionnant en 5V, il est nécessaire d'abaisser alors la tension d'alimentation de la carte de 12Volts grâce à un montage abaisseur de tension.
- **Un Quartz** : composant oscillant, à une fréquence précise lors d'une mise sous tension, il est utilisé pour osciller à 16Mhz et ainsi remplacer l'horloge interne de 8Mhz de l'Atmega8535.
- **Des transistors MOSFET<sup>2</sup>** : ils servent d'interrupteur commandés par les sorties. Lors d'une impulsion venant de l'atmega8535, le transistor se ferme alimentant ainsi les lampes associées sous une tension de 12V.
- **Des Connecteurs d'entrées et de sorties** : ils font le lien entre la carte et les LED en sorties et les capteurs en entrées.

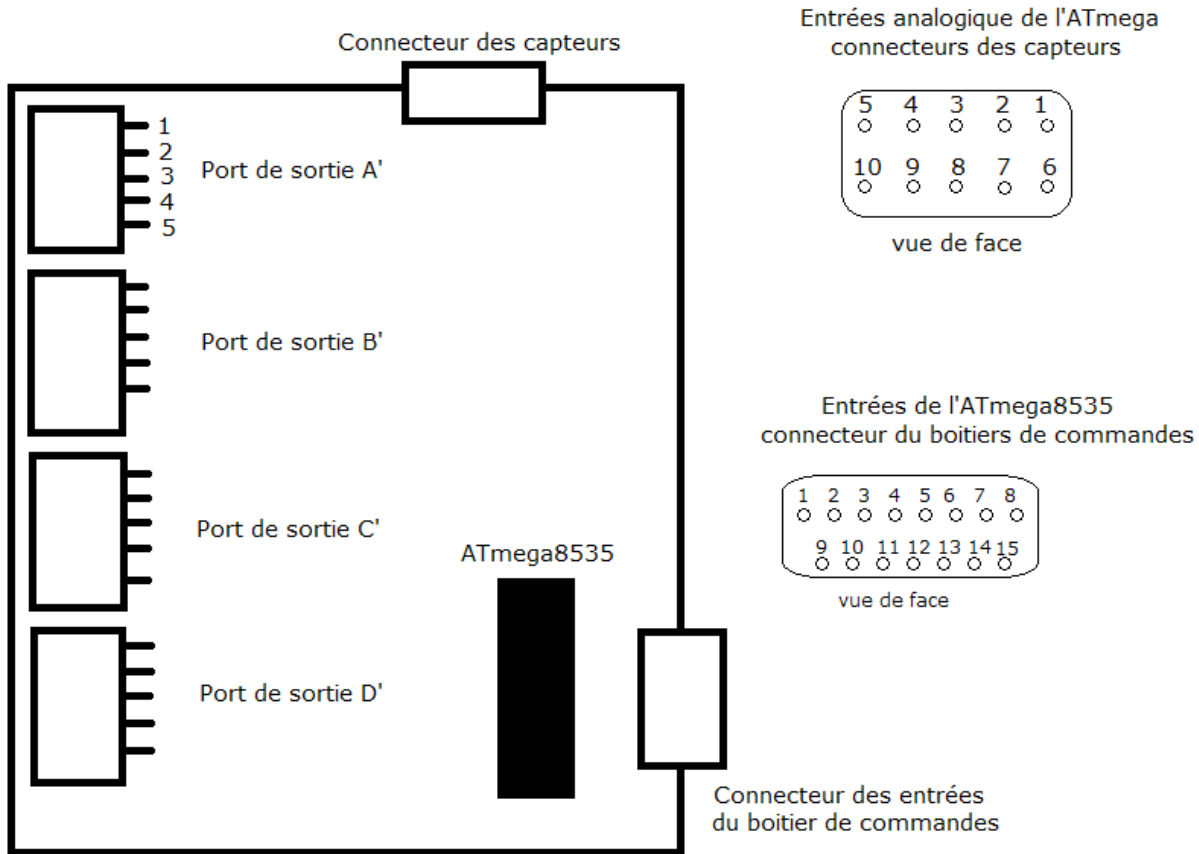


*Illustration 5: Carte électronique permettant la gestion des feux installé directement sous le Kart, photographie faite par les auteurs*

---

2 MOSFET : Metal Oxide Semi Conducteur Field Effect Transistor

Nous avons dans un premier temps récupéré les informations des entrées/sorties sur la carte électronique en nous basant sur un schéma simplifié ci-dessous.



*Illustration 6: Schéma simplifié de la carte électronique de la maquette réalisé par les auteurs*

Avant de pouvoir tester chaque LED[2], nous avons récupéré les informations des ports d'entrées/sorties de la carte Atmega8535, de la carte et du boîtier de commandes. Nous avons relevé sur quel port était connecté chaque entrées et ainsi nous avons pu dresser la liste des entrées/sorties de l'ATmega basé sur le schéma simplifié de la carte ci-dessus.

Cette liste a pour but de nous faciliter la tâche dans la désignation des feux et permettre un repérage rapide des affectations des feux sur les ports de l'ATmega8535.

## 2.2.3. Récupération des informations d'entrées/sorties

### 2.2.3.1. Connecteur des capteurs

BROCHES	DÉSIGNATION	AFFECTATION SUR L'ATmega
1	Masse du Capteur Frein	-
2	Masse du Capteur Luminosité	-
3 à 5	Masse	-
6	Capteur Luminosité	PA3
7	Capteur Frein	PA2
8 et 9	-	-

Illustration 7: Listage des broches du connecteurs des capteurs fait par les auteurs

### 2.2.3.2. Connecteur du boîtier

BROCHES	DÉSIGNATION	COULEUR FIL DANS LE CONNECTEUR	COULEUR FIL DANS LE BOITIER	AFFECTATION
1	-	marron	-	PC7
2	ON/OFF	Jaune	Gris	PC6
3	Plein	Violet	Vert	PC4
4	Code	Blanc	Bleu	PC5
5	Clignotant Droit	Orange terne	Marron	PC2
6	Clignotant Gauche	Noir	Violet	PC3
7	Warning	Orange vif	Jaune	PC1
8	Manu/Auto	Bleu	Blanc	PC0
9	Led Témoin	Rouge	-	Condensateur
10 à 15	Masse	Noir	Noir	-

Illustration 8: Listage des broches du connecteurs du boîtier fait par les auteurs

### 2.2.3.3. Sortie de l'ATmega8535

PORT DE LA CARTE	BROCHE	DÉSIGNATION	AFFECTATION
A'	1	Cligno Gauche Arrière	PD4
	2	Recul	PD0
	3	Témoin Arrière	PD1
	4	Stop Arrière	PB3
	5	Stop Arrière	PB3
B'	1	Cligno Droit Arrière	PD3
	2	Recul	PD0
	3	Témoin Arrière	PD1
	4	Stop Arrière	PB3
	5	Stop Arrière	PB3
C'	1	Route Avant	PD7
	2	Cligno Gauche Avant	PD4
	3	Croisement Avant	PD2
	4	Position Avant	PD6
	5	Route Avant	PD7
D'	1	Route Avant	PD7
	2	Cligno Droit Avant	PD3
	3	Croisement Avant	PD2
	4	Position Avant	PD6
	5	Route Avant	PD7

Illustration 9: Listage des sortie de l'ATmega 8535 fait par les auteurs

## 2.2.4. Recherches préliminaires de la programmations

Dans cette partie, nous allons faire une étude fonctionnelle de la programmation.

### 2.2.4.1. Schéma fonctionnel de niveau 1

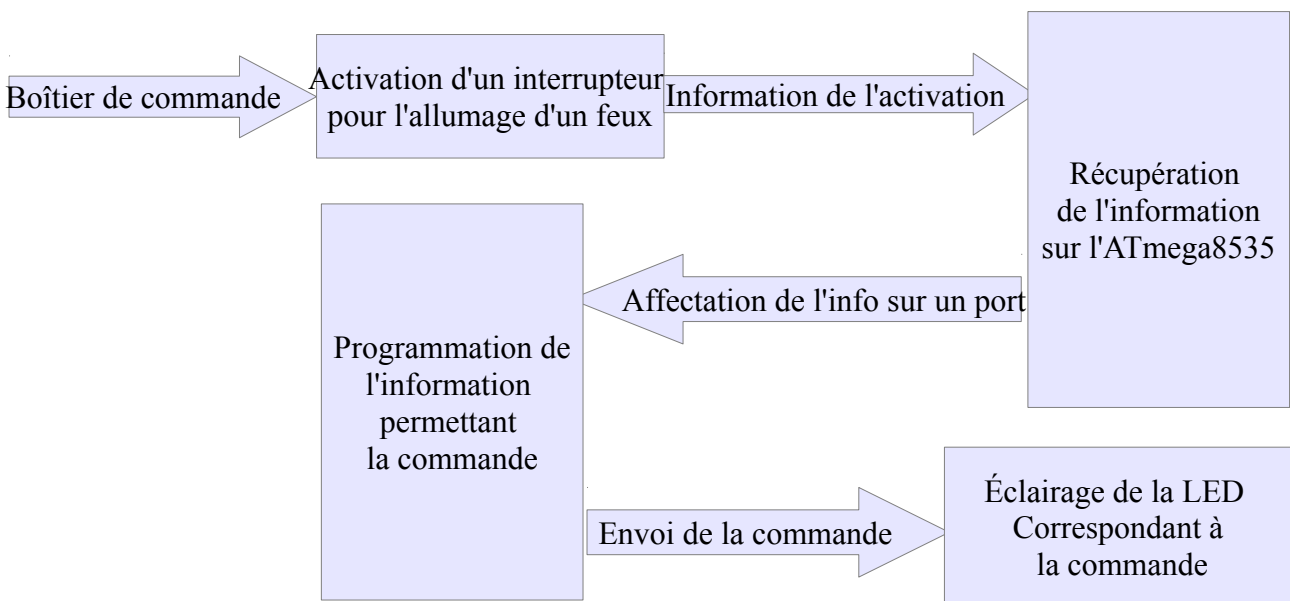


*Illustration 10: Schéma fonctionnel de niveau 1 réalisé par les auteurs*

Comme on peut le voir, on attend les informations qui proviennent des commandes, puis on les traite afin de commander les feux correspondants.

### 2.2.4.2. Schéma fonctionnel de niveau 2

Après avoir établi le niveau 1 du schéma fonctionnel, nous allons à présent aller un peu plus loin dans l'analyse.



*Illustration 11: Schéma fonctionnel de niveau 2 réalisé par les auteurs*

L'analyse fonctionnelle terminée et toutes les recherches effectuées, nous pouvons alors maintenant commencer à rentrer dans notre projet.



## 3. L'éclairage du Kart

### 3.1. La programmation

#### 3.1.1. Présentation

Dans cette partie, l'étude va s'axer autour du programme réalisé sur l'ATmega8535 qui va permettre de gérer tout l'éclairage du kart et les différents modes possibles. Une représentation simple du fonctionnement est la suivante :

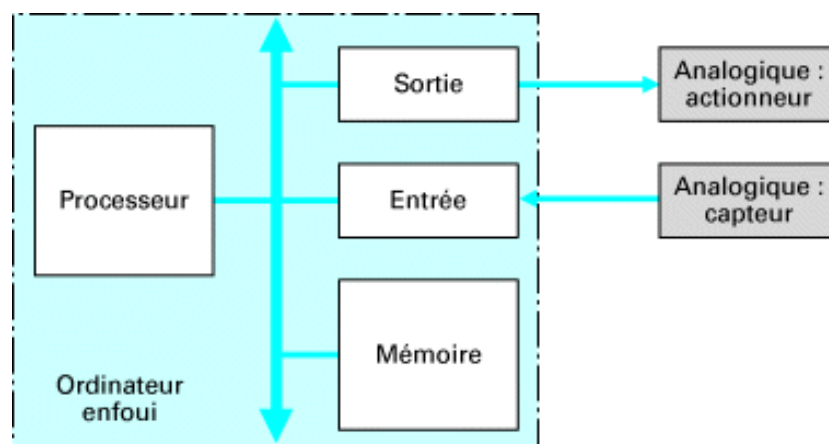


Illustration 12: Schéma de fonctionnement[0]

L'entrée peut aussi se faire par des interrupteurs, on utilisera que deux capteurs dans notre application.

Pour voir l'ensemble du travail on peut séparer le tout en différentes parties: on va donc voir le mode manuel qui est la base de l'éclairage et que l'on retrouve sur n'importe quel véhicule, on enclenche un interrupteur et la lampe associée s'allume.

Ensuite on pourra voir le mode automatique qui va lui se rapprocher des modèles de voitures récentes en proposant un éclairage progressif en fonction de la luminosité ambiante. Et nous verrons par la suite le mode « expo » prévu pour faire de la lumière dans le cadre d'une présentation du Kart, par exemple au challenge pédagogique E-Kart qui se déroule durant le festival du Car Tec-inno de Vierzon.

Dans cette partie afin de faciliter la lecture et d'illustrer le programme nous mettrons en concordance les ordinogrammes et le programme.

### 3.1.2. Les directives de pré-processeurs

Sous ce nom barbare se cachent quelques fichiers à inclure. Ces fichiers contenant la plupart des fonctions utilisées dans le programme afin de ne pas les déclarer à chaque fois. On retrouve aussi des définitions de nom de variable. On a donc « renommer » les ports de l'ATmega8535 afin qu'ils soient plus faciles à mémoriser une fois que l'on aura la programmation.

```
1 #include <stdio.h>
2 #include <mega8535.h>
3 #include <delay.h>
4
5 #define feux_recul PORTD.0
6 #define feux_temoin PORTD.1
7 #define croisement PORTD.2
8 #define CD PORTD.4
9 #define CG PORTD.3
10 #define stop PORTB.3
11 #define position PORTD.6
12 #define route PORTD.7
13
14 #define ON_OFF PINC.6
15 #define code PINC.5
16 #define plein PINC.4
17 #define ICG PINC.3
18 #define ICD PINC.2
19 #define Warning PINC.1
20 #define mode PINC.0
21 #define posi PINC.7
```

Fichiers contenant les fonctions

Renommage des sorties

Renommage des entrées

Numero de ligne

Illustration 13: Directives de pré-processeur réalisées par les auteurs

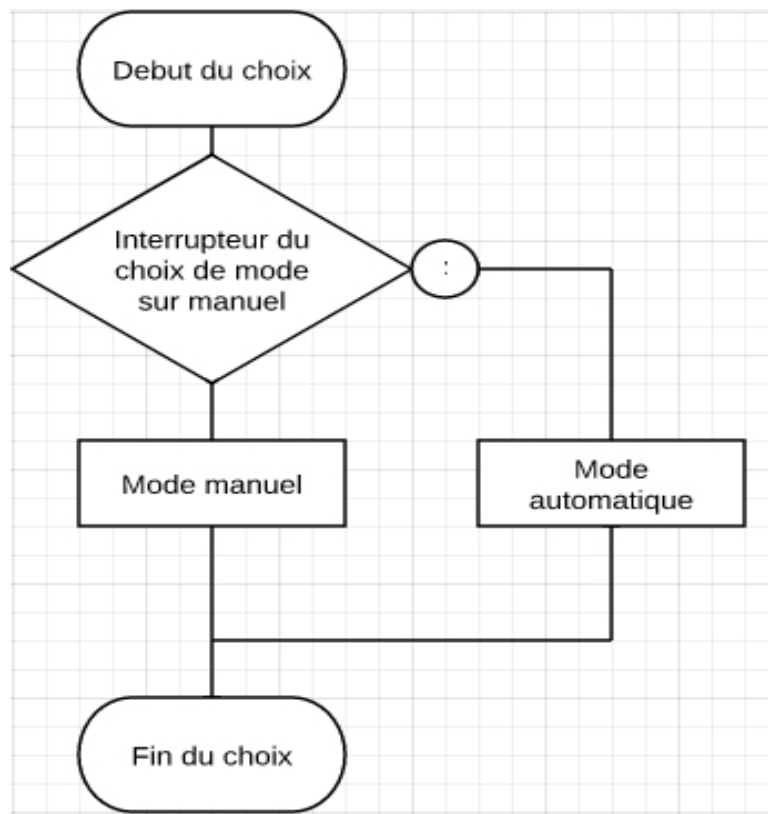
Comme on peut le voir on a ajouté les fichiers suivants :

- **stdio.h** : Contient les fonctions standard du langage C.
- **mega8535.h** : Contient toutes les fonctions relatives au micro-contrôleur utilisé, ainsi que les déclarations des entrées/sorties de celui-ci.
- **delay.h** : Contient les fonctions permettant d'attendre un laps de temps donné.

Ensuite, on a les sorties renommées avec le nom des feux afin de les repérer rapidement et ainsi accélérer la programmation. De même pour les entrées avec les commandes existantes.

### 3.2. Le mode manuel

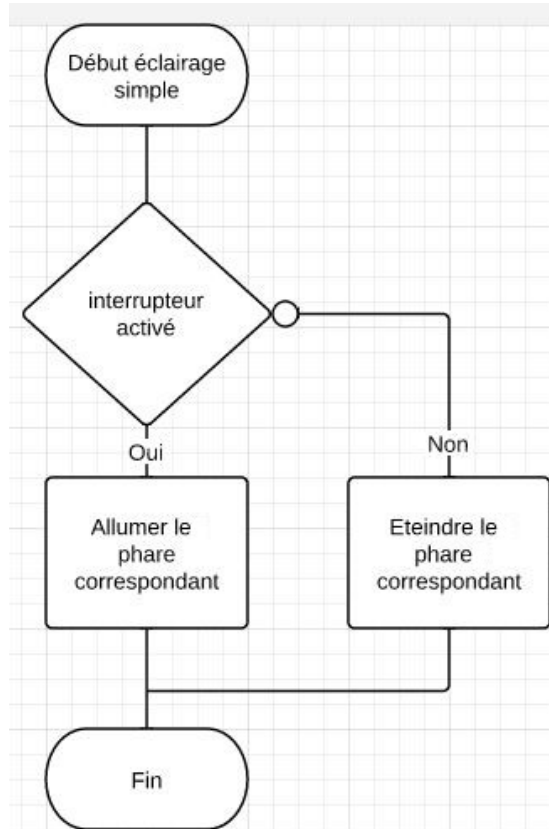
Comme dit dans la présentation ce mode est le plus simple à mettre en place, car c'est un fonctionnement de type interrupteur. Par contre il a fallu mettre en place un programme permettant de sélectionner le mode de fonctionnement.



*Illustration 14: Ordinogramme de sélection du mode réalisé par les auteurs*

### 3.2.1. Les lampes « simples »

On va donc étudier l'éclairage lié à chaque commandes. Le premier programme qui a été fait est celui gérant les feux de croisement et les pleins phares.



*Illustration 15: Ordinogramme d'allumage des phares réalisé par les auteurs*

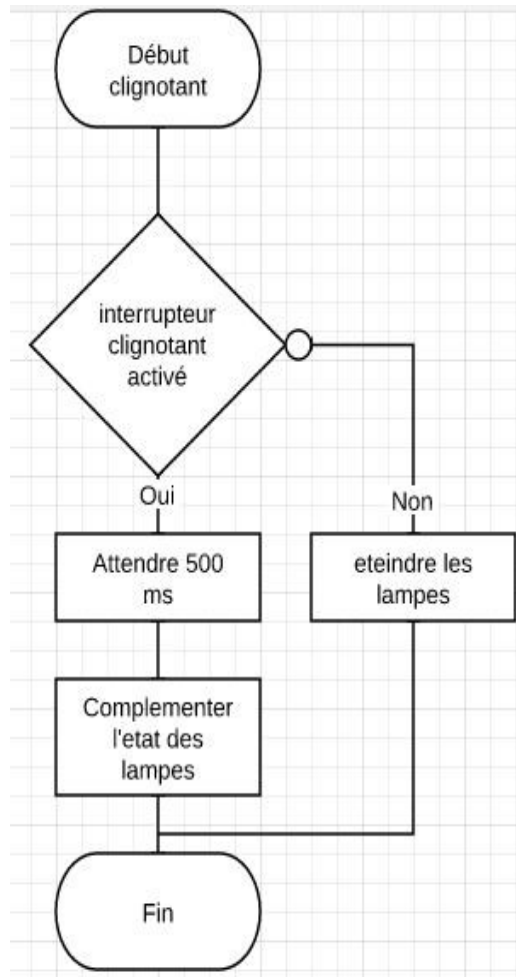
Pour l'allumage des phares, on utilise le code le plus simple. Cependant, il a fallu adapter le code original, sans oublier que les pleins phares se trouvaient sur les sorties des MLI<sup>3</sup>. De plus le code n'étant pas binaire, il a été nécessaire de le mettre en hexadécimal pour avoir une luminosité maximum.

De plus lors des tests sur le Kart on a pu voir que les feux de croisement et les pleins phares étaient de même couleur (Blanc), la différence d'éclairage était alors inexistante. Afin de palier à cet aléa, nous avons décidé de mettre en place uniquement les pleins phares, pour ainsi faire s'allumer les deux ampoules blanches. En ce qui concernent les interrupteurs nous avons donc effectué plusieurs tests pour l'éclairage des feux de croisement. On vérifie si l'interrupteur « croisement » ou « plein » est activé, pour ainsi allumer ou éteindre les phares.

<sup>3</sup> MLI: Modulation par largeur d'impulsion, utilisée ici pour réaliser l'éclairage variable (CF partie 3.3)

### 3.2.2. Les clignotants

Pour ce bout de code qui va gérer les clignotants et qui sera commun au mode manuel et automatique, on retrouve le même schéma que précédemment, c'est-à-dire que l'on fait un test sur un interrupteur. Et on allume ou on éteint la lampe correspondante, à la nuance près qu'ici on aura un temps d'attente à ajouter. Comme on peut le voir sur le schéma suivant:



*Illustration 16: Ordinogramme de fonctionnement des clignotants réalisé par les auteurs*

Comme on peut le voir, après l'activation de l'interrupteur des clignotants ou des « Warnings », on attend 500ms et on inverse l'état de la lampe (l'allumer si elle est éteinte ou l'éteindre si elle est allumée). Sinon on éteindra les lampes encore allumées. Le programme principal, contenant tous les tests tourne en boucle grâce à une boucle infinie (While(1)) donc le test se faisant à chaque tour de boucle (très rapide, à peine quelques millisecondes) on aura une inversion des lampes, tant que l'interrupteur est activé, ce qui provoquera le clignotement.

La valeur de temps choisie a été obtenue par essais successifs afin de trouver une fréquence de clignotement qui s'est révélée être la plus proche du comportement réel d'un clignotant.

Maintenant que la partie la plus simple à mettre en place a été réalisée on va pouvoir s'intéresser à la partie qui nous a demandé le plus de temps et de réflexion.

### **3.3. Le mode automatique**

Le mode automatique, s'il est sélectionné, va récupérer l'information d'un capteur de luminosité (dont la référence nous est inconnue) et d'un capteur de freinage (PB6 de chez Curtis). Grâce aux informations renvoyées par les capteurs, on a pu réaliser très sommairement un programme qui récupérait la valeur et la redistribuait sur les lampes, avec une formule afin d'avoir une valeur comprise entre 0 et 255, la plage de variation de la MLI de l'Atmega8535.

La formule étant la suivante, en prenant pour valeur maximum des capteurs 5 Volt.

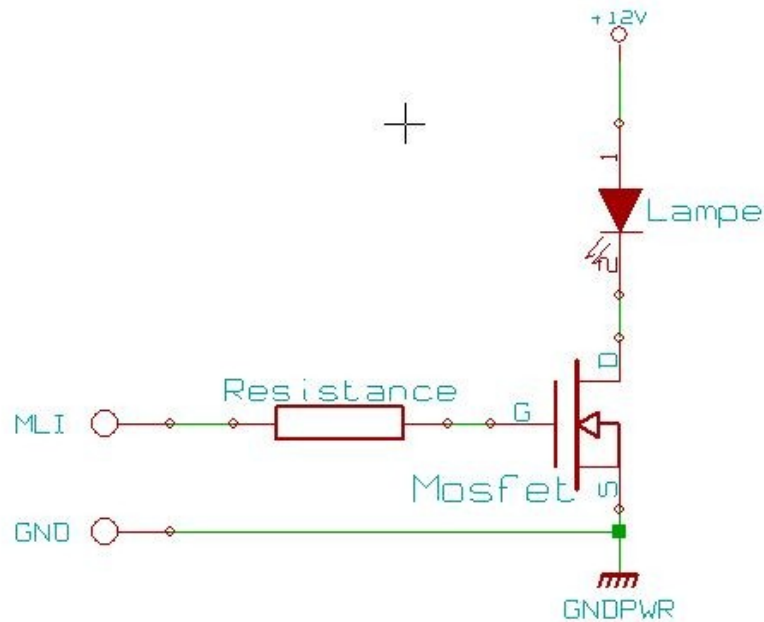
$$\text{Valeur de la MLI} = \frac{\text{Valeur du capteur} \times 255}{5}$$

Une fois ce programme réalisé et implémenté on s'est retrouvé, malgré une variation désirée, avec un éclairage non linéaire. Cela étant dû au fait que les lampes à LED n'ont pas un éclairage linéaire. Nous avons donc du effectuer des mesures sur celles-ci afin de les linéariser, afin d'éviter de programmer des paliers d'éclairage et rendre les variations d'éclairage plus fluide.

### 3.3.1. La linéarisation des lampes

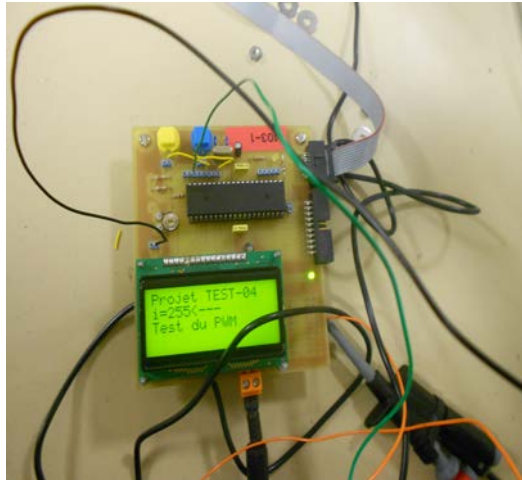
#### 3.3.1.1. Explications des mesures

Afin d'effectuer les tests sur chaque lampes indépendamment les unes des autres on a réalisé le montage que l'on avait sur la maquette mais pour une seule lampe.



*Illustration 17: Schéma du montage réalisé mesurant la valeur des lampes réalisé par les auteurs*

Donc comme on peut le voir on a reproduit le câblage de la maquette en adaptant le montage à une seule ampoule. Si l'on câble ce montage en faisant bien attention au branchement du Mosfet on obtient une luminosité qui dépend de la tension d'entrée sur la broche ici notée MLI. Afin de pouvoir tester rapidement la valeur de chaque lampes, nous avons récupéré un programme sur le site internet de Mr Lequeu. Implémenté sur une maquette présente à l'IUT, nous y avons branché notre fil MLI et notre fil GND (respectivement le fil vert et le fil noir sur l'image suivante).



*Illustration 18: Maquette de test, photo réalisée par les auteurs*

Une fois le montage réalisé il nous fallait pouvoir mesurer l'éclairage d'une lampe sans que la mesure ne soit perturbée par la luminosité ambiante. Nous avons donc pris une boîte en carton qui nous a servit de « boîte noire » dans laquelle nous avons placé l'ampoule câblée. La maquette de tests est laissée à l'extérieur afin de pouvoir faire varier la valeur de la MLI. En face de cette lampe, nous avons installé un luxmètre, qui a pour but de mesurer le flux lumineux.



*Illustration 19: Montage de test de linéarisation photo réalisée par les auteurs*



Afin d'étalonner l'appareil de mesure, on l'allume avec le cache afin d'obtenir la valeur pour une luminosité nulle. Ensuite, on retire le cache pour pouvoir commencer les mesures.

Nous avons dès lors effectué les mesures pour chaque couleur de lampe, afin de réaliser une étude complète de l'éclairage du Kart. Les seules qui nous intéressent dans cette partie sont les lampes de couleur rouges et blanches respectivement pour le freinage et l'éclairage avant du Kart. Nous verrons donc dans cette partie les courbes correspondantes à ces deux couleurs.

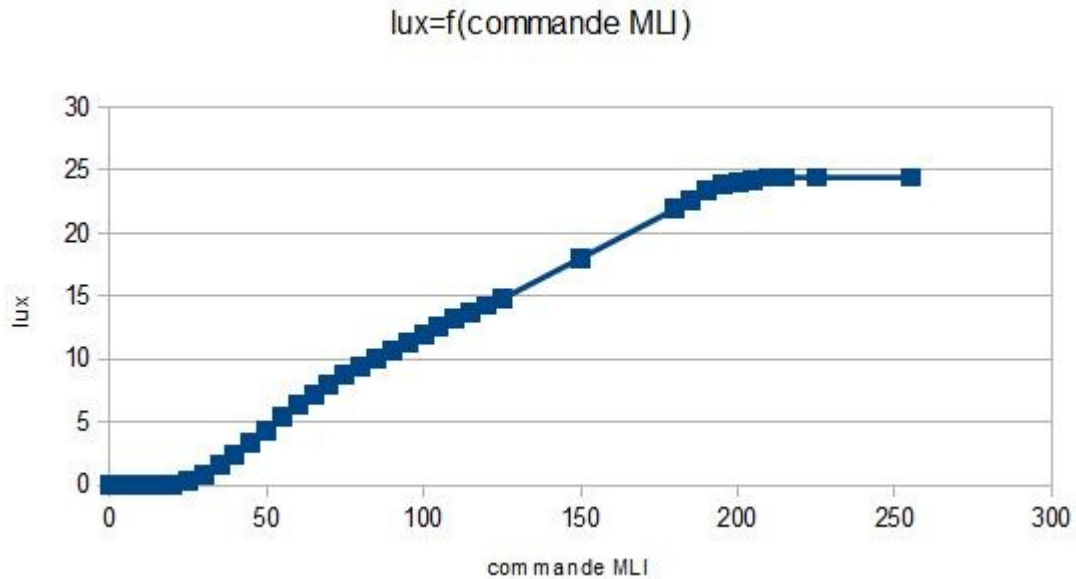
La démarche expérimentale est la suivante:

- On place la lampe et le luxmètre dans la boîte noire.
- On fait varier la valeur de la MLI.
- On entre les valeurs mesurées dans un tableur de type Excel.
- On trace ensuite la courbe  $\text{Lux} = f(\text{MLI})$  complète.
- On regarde les points entre lesquels la courbe est linéaire.
- On isole la partie de la courbe linéaire et on en extrait son équation.
- On mesure la plage de variation du capteur.
- On tire les équations à entrer dans le programme afin de linéariser les lampes.

On va voir à travers les deux exemples des deux couleurs cette démarche et surtout les formules à entrer dans le programme.

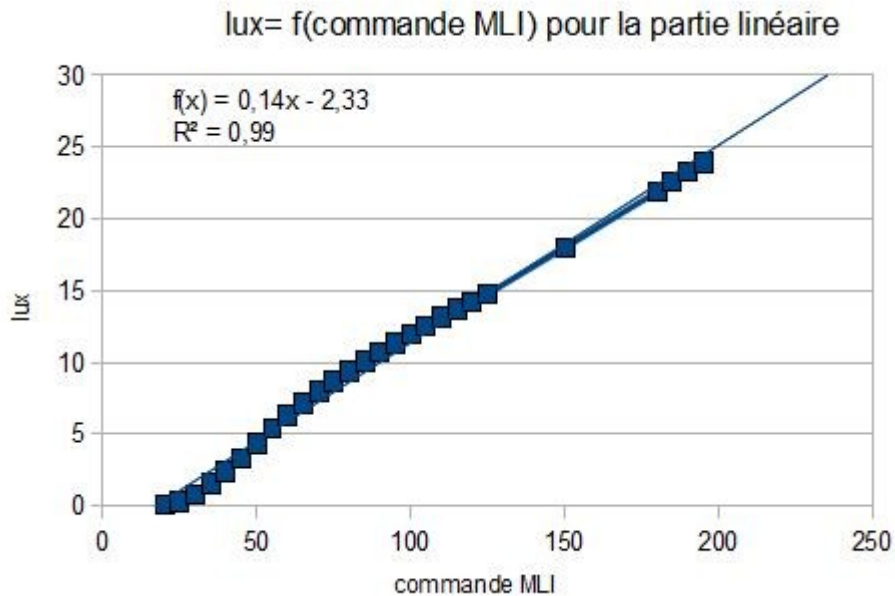
### 3.3.1.2. La couleur rouge

Donc comme nous l'avons vu précédemment, nous avons effectué le test de linéarisation avec une ampoule rouge, le résultat est le suivant :



*Illustration 20: Courbe pour les lampes de couleur rouge réalisée par les auteurs*

On peut voir que la partie linéaire se situe entre les valeurs de MLI de 25 à 195, afin de valider ce choix on fait un zoom sur cette partie



*Illustration 21: Zoom de la courbe précédente réalisée par les auteurs*

On peut voir que cette partie est bien linéaire, on peut donc récupérer les valeurs précédente afin de retrouver les formules. La première chose à faire est de mesurer la plage de variation du capteur de frein qui varie de 0V à 4,17V. Le capteur étant alimenté en 5V on peut effectuer la conversion suivante :  $\frac{4,17 \times 255}{5} = 212,67 \approx 213$  On a donc la valeur maximale du capteur. Cette valeur va nous permettre de mettre en œuvre la formule principale de linéarisation des lampes.

$\frac{\text{Tension instantanée du capteur} \times (195 - 25)}{213} + 25$  La plage de variations de la fonction ainsi créée sera uniquement présente dans cette plage de variations. En effet, si la tension mesurée est de 0 en hexadécimal (0 Volt réels) alors on a 25 pour la commande de MLI et si on a 213 en hexadécimal (4,17 Volt réels) on a 195, ces deux valeurs étant les extrêmes de la partie linéaire pour cette lampe.

### 3.3.1.3. La couleur blanche

De même que précédemment, on a réalisé la courbe des lampes à LED de couleur blanche et on a obtenu la courbe suivante:

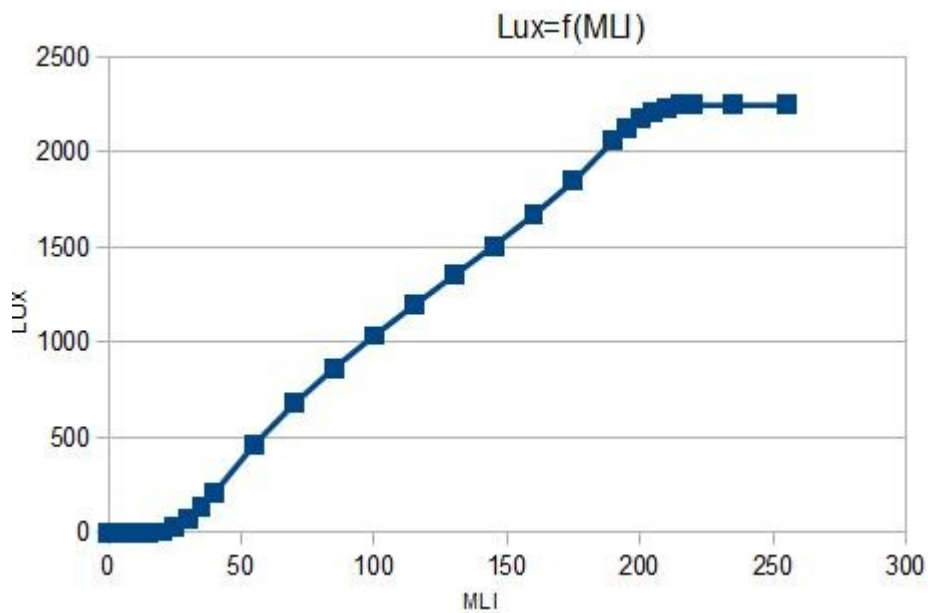


Illustration 22: Courbe de variation des LED blanches réalisée par les auteurs

Dans ce cas-ci les extremums de la partie linéaire sont 25 et 210, le zoom effectué dans cet intervalle confirme cette tendance:

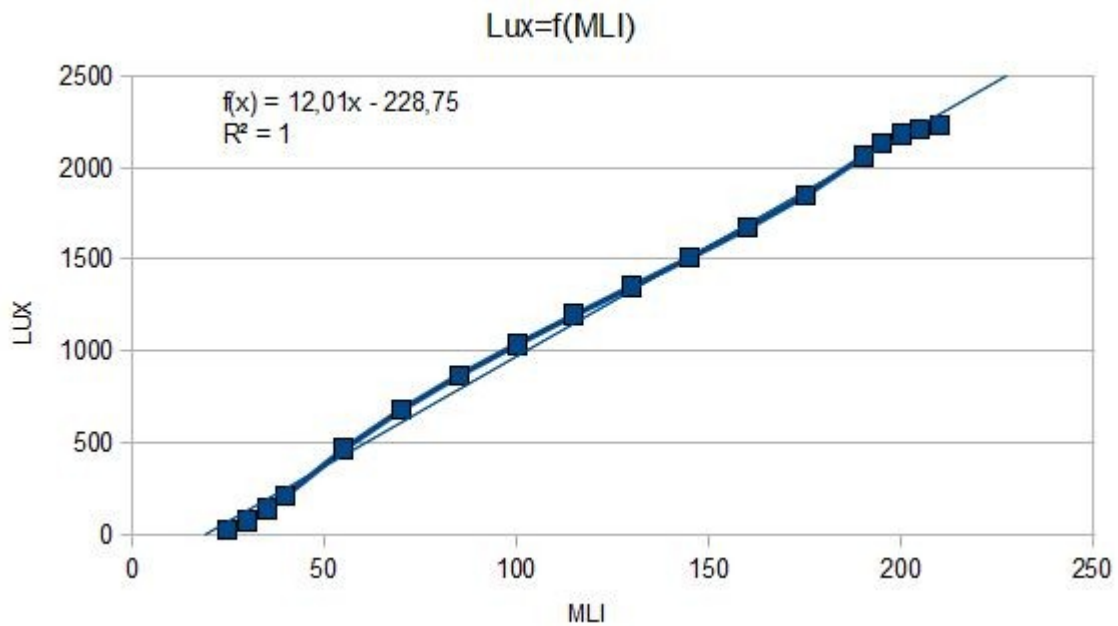


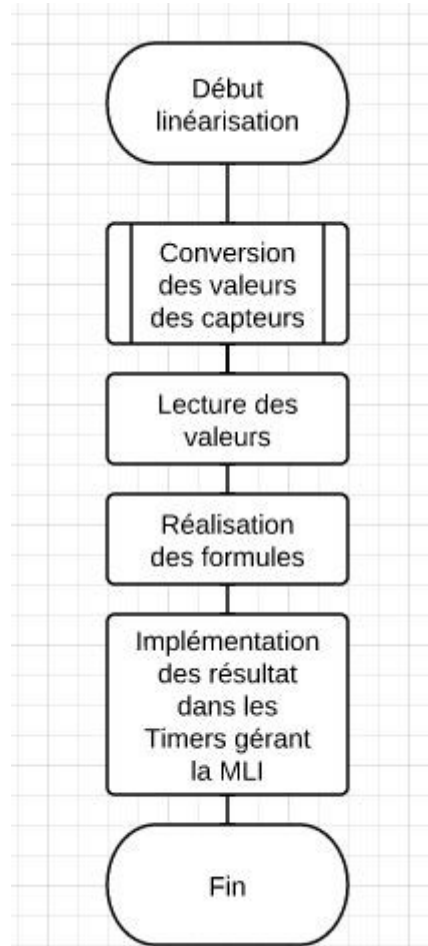
Illustration 23: Zoom de la courbe précédente réalisée par les auteurs

On refait une mesure de tension mais cette fois-ci sur le capteur de luminosité, celui-ci variant de 0V à 5V, la valeur maximale qui pourra être envoyée par le capteur en hexadécimal sera de 255. On peut donc trouver la formule de linéarisation

$$\frac{\text{Valeur instantanée du capteur} \times (210 - 25)}{255} + 25$$

### 3.3.2. La programmation

Maintenant que l'on a les deux formules, il suffit de réaliser le programme afin d'implémenter ces valeurs au programme.



*Illustration 24: Ordinogramme de la fonction de linéarisation réalisé par les auteurs*

Comme on peut le voir encore une fois le programme est assez simple. Cependant, toute la recherche en amont nous a permis de faire une étude de linéarité sur les lampes à LED des différentes couleurs utilisées dans ce projet. De plus, cette étude permet d'en apprendre plus sur les capteurs, sachant que les documentions techniques sont très incomplètes ou impossibles à trouver dans le cas du capteur de luminosité.

## 4. Optimisation matériel

### 4.1. Modification du Kart

#### 4.1.1. Le volant

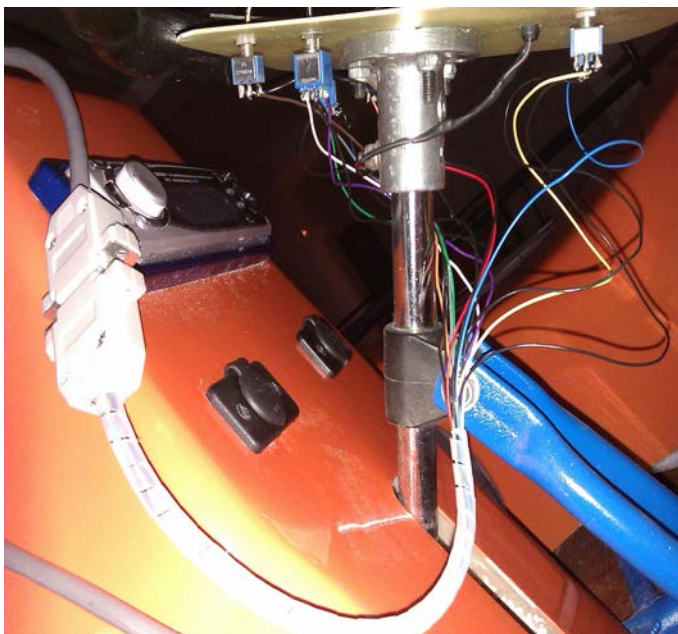
Dans cette partie, nous voulons mettre toutes les commandes du kart à la portée du conducteur. Les commandes se trouvaient sur un boîtier. Nous devons alors placer ce boîtier de telle sorte à ce que le conducteur puisse y avoir accès facilement. Or pour cela il fallait découper la carrosserie du kart pour insérer le boîtier, ce qui n'était absolument pas envisageable. Il a donc fallu imaginer un autre système. Nous avons alors pensé à mettre les commandes sur le volant même du kart, comme les voitures de type F1.

Pour cela, nous avons utilisé une plaque de circuit imprimé vierge, c'est-à-dire sans cuivre ni résine photosensible. Cette plaque a été ensuite dimensionnée pour pouvoir être installée sur le volant de telle sorte à ne pas gêner la conduite du kart.



*Illustration 25: Commandes au volant, photographie faite par les auteurs*

Après avoir dessiné sur la plaque les contours des broches du volant, on a pu placer les positions désirées des interrupteurs. Puis nous avons percé la plaque et mis les interrupteurs.



*Illustration 26: Connecteur des commandes au volant, photographie faite par les auteurs*

La plaque finie, il ne restait plus qu'à câbler les interrupteurs sur un connecteur 15 broches. Nous avons donc utilisé le tableau d'affectation des broches de sortie de l'ATmega8535 sur le connecteur du boîtier décrit précédemment.

Une fois le câblage effectué, nous avons alors regroupé les câbles dans une gaine spirale afin de les protéger.

#### 4.1.2. Antenne et allume-cigare

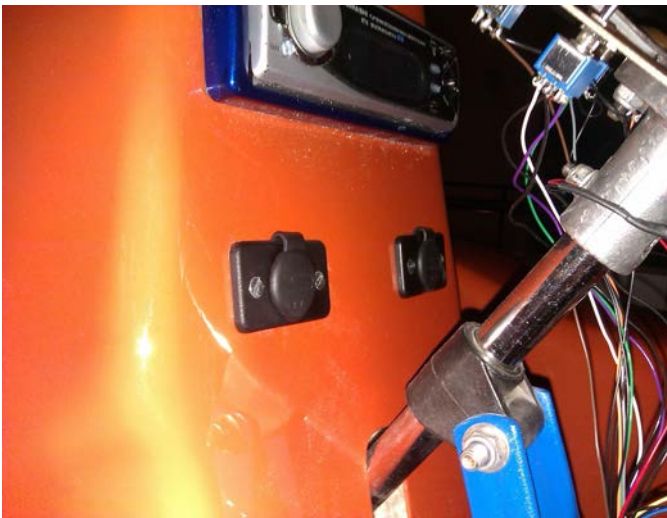
Dans le cadre d'optimisation du kart électrique, nous avons eu à installer deux allume-cigares et une antenne.

Les allume-cigares ont pour but de pouvoir recharger les batteries et d'alimenter un gyrophare fait par les étudiants des années précédentes.

L'antenne permettra seulement de réceptionner les fréquences radio sur l'autoradio installé sur le kart.

Pour les installer, nous avons dû percer la carrosserie du kart. Tout d'abord nous avons percé deux trous sur le nez du kart à un diamètre correspondant au diamètre de l'allume cigare. Nous y avons ensuite serti des fils sur des cosses pour pouvoir directement les brancher sur les allume-cigares. Ces fils iront ainsi sur les câbles reliés à la batterie du Kart.

Pour l'antenne, nous avons aussi directement percé la carrosserie à l'arrière du Kart. Nous y avons soudé un fil standard, non protégé des parasites, car l'IUT ne possède pas de bobine de fil protégé. Ce fil passe alors sous le kart pour arriver dans le nez de celui-ci et ainsi être branché sur l'emplacement prévu de l'antenne sur l'autoradio.



*Illustration 28: allume-cigares installés sur le kart, photo faite par les auteurs*



*Illustration 27: Antenne de la radio installée sur le kart, photo faite par les auteurs*



## 4.2. Phrase de position avant

### 4.2.1. Étude et recherches préliminaires

Lors de notre première étude sur le kart, nous avons pu voir les phares de position avant installés mais ne fonctionnant pas de façon optimale. Or un autre groupe d'étudiants travaillaient en parallèle sur le kart et l'un de leurs objectifs fut de changer ces phares.

Après une étude rapide de la carte des phares, ils se sont aperçus que l'éclairage de l'ensemble des LED n'était pas homogène et que cela était dû au fait que l'intensité qui traversait les LED n'était pas constante.

Ils ont alors proposé une solution à ce problème. Ils ont créé un montage source de courant à base de transistor et diode Zener schématisé ci dessous.

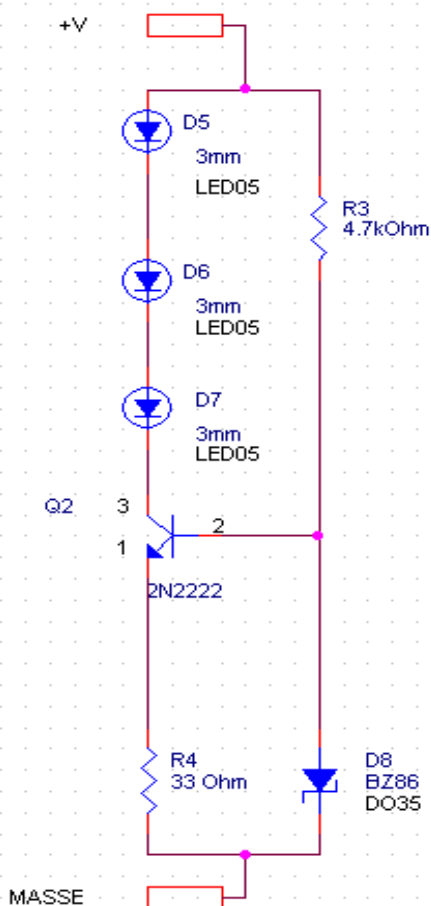


Illustration 29: Schéma électrique du montage source de courant, fait par les auteurs sous Orcard

Ils ont créé un groupement de 3 LED haute luminosité\*. Celle-ci consommant 3,5 Volts au maximum pour un courant de 20 milliampères, représentant une tension maximale sur le groupement de 10,5Volts.

La carte étant alimentée au maximum en 12 Volts, l'ajout d'une LED sur le groupement sera donc impossible.

Nous avons alors optimisé la carte de l'autre groupe en la rendant plus ludique tout en nous basant sur le montage ci-contre qui fonctionnait parfaitement.

Notre but était de créer 2 nouvelles cartes des phares de positions avant mais en positionnant les LED de telle sorte qu'elles forment sur la première carte « IUT » et sur la deuxième « GEII ».

\* [3]



Nous avons utilisé le logiciel ORCAD pour réaliser les différents typons.

Tout d'abord, il fallu schématiser le montage à transistor (voir précédemment, illustration 16). Or il a fallu plusieurs groupements de LED, ce qui rendait la schématisation plus dure. On a alors hiérarchisé le montage de base en créant des blocs pour ainsi les répéter autant de fois que souhaitée, ne restait plus qu'à mettre l'alimentation (voir en annexe, illustration 32).

Les schémas finis, il ne restait plus qu'à faire les typons. Nous avons alors une contrainte pour ceux-ci. Le dimensionnement des cartes ne pouvait excéder 10cm par 11cm pour une raison mécanique, celles-ci ne pouvant pas être installées si les dimensions étaient plus grandes.

Une fois les typons réalisés (voir annexe, illustration 33 et 34), nous avons pu graver les cartes, les percer et souder les composants.

Lors de la première carte « IUT », aucun problème n'est survenu lors de sa réalisation. Mais pour la carte « GEII », cette dernière comportant plus de groupements de LED, on a dû faire face à un problème d'encombrement des composants par rapport à la taille de la carte. Pour résoudre le problème nous avons dû réduire lors de la réalisation du typon le pas des résistances et des diodes Zener, nous obligeant à les souder de façon axial.



*Illustration 30: Carte phare avant GEII, photographie fait par les auteurs*

Après avoir testé préalablement les deux cartes en les alimentant par un générateur, nous les avons alors installées sur le Kart.

Lors de l'installation, nous avons été confronté à un léger problème. Sur le kart, il y a 2 fils de 12Volts et 1 fil de masse partant de la carte de l'ATmega et arrivant sur l'emplacement des phares. Or sur la carte des phares, il n'y a qu'un emplacement prévu pour le 12Volts et deux pour les masses.

Nous avons alors mis une gaine thermique sur l'un des fils de 12 Volts, et soudé normalement les deux autres fils sur la carte.



*Illustration 31: Phare de position avant IUT en fonctionnement sur le kart, photo faite par les auteurs*

## Conclusion

Dans ce dossier nous avons pu voir le travail que nous avons réalisé durant les séances d'étude et réalisation. Durant ce projet, nous avons donc réalisé un programme complet pour l'éclairage. Mais le plus important dans la partie de programmation a été l'étude de linéarisation des lampes, permettant d'en savoir plus sur celles-ci et de faire l'éclairage progressif.

Ensuite, vu que la partie de programmation a été faite plus rapidement que prévu nous avons pu travailler sur la partie matérielle du kart. Nous avons travaillé sur les feux de position avant afin d'écrire « IUT » et « GEII ». Nous avons ensuite pu installer des allumes cigares, afin de pouvoir recharger les batteries et alimenter un gyrophare (réalisé par d'ancien étudiant de l'IUT), ainsi qu'une antenne servant à réceptionner la radio.

Durant ce projet nous avons pu mettre en pratique des compétences vues à l'IUT. Par exemple, le module complémentaire de micro-contrôleur nous a permis de comprendre plus facilement comment faire fonctionner la MLI.

Le projet de base et les améliorations étant finies cela laisse donc de nombreuses possibilités pour faire évoluer le kart comme l'installation d'un moteur ou pour permettre, grâce à l'antenne, la communication entre le pilote et son équipe.

Pour finir, nous pouvons dire que ce projet nous a fortement intéressé autant pour sa diversité que pour le fait de travailler sur le kart qui pourra représenter l'IUT GEII de Tours lors de différents événements.

## Résumé

Le projet portait initialement sur la gestion des feux progressifs ainsi que des clignotants du Kart électrique de l'IUT.

Une maquette regroupant le fonctionnement des phares était à notre disposition, et nous avons pu appréhender notre projet à travers cette dernière. Nous avons tout d'abord étudié son fonctionnement avec un programme implanté pour se rendre compte du réel travail et des recherches à effectuer ainsi que des problèmes existant.

Une fois l'étude et les recherches faites, nous avons pu nous concentrer sur le cœur de notre projet : la programmation, celle-ci se faisant sur le micro-contrôleur Atmega8535. Les cours de module complémentaire microcontrôleur, que l'on suivait parallèlement, traitaient justement ce composant et cela nous a grandement facilité pour notre projet et que l'on a pu programmer sans aucune difficulté.

Durant notre programmation, nous avons fait face à une principale difficulté : de rendre l'éclairage des feux progressifs linéaire par rapport aux capteurs. Pour cela nous avons effectué des tests de linéarisation avec la MLI pour rendre l'éclairage le plus linéaire possible et de choisir par la suite la couleur des ampoules la plus appropriée.

Ce n'est qu'après ces tests que nous avons programmé l'ensemble de l'éclairage du kart à savoir 3 modes: manuel, automatique, correspondant à un éclairage progressif, et un mode exposition où l'ensemble des phares ne sont plus commandés mais s'éclaire de façon autonome.

En parallèle à ce projet, nous avons eu des petits travaux annexes rentrant dans le cadre de notre projet. Tout d'abord nous avons installé les commandes au volant, des allure-cigares et une antenne permettant respectivement un accès aux commandes des phares facile pour le pilote, un branchement des batteries et d'un gyrophare puis une réception de la radio.

Pour finir nous avons créé de nouveaux phares de position avant. Ceux-ci ne fonctionnant pas de façon optimal, un autre groupe d'étudiant a apporté une solution à ce problème. Nous avons alors repris cette solution en l'optimisant et ainsi créé deux cartes électronique pour les phares de position avant s'éclairant de manière à former « IUT » et « GEII ».

Nombre de mots : 345

## Index des illustrations

Illustration 1: Maquette d'étude de l'éclairage du kart, photo prise par les auteurs.....	7
Illustration 2: Boîtier de commandes des feux.....	7
Illustration 3: Schéma de l'ATmega8535 issu de sa datasheet[1].....	8
Illustration 4: Planning du projet réalisé par les auteurs.....	10
Illustration 5: Carte électronique permettant la gestion des feux installé directement sous le Kart, photographie fait par les auterus.....	12
Illustration 6: Schéma simplifié de la carte électronique de la maquette réalisé par les auteurs.....	13
Illustration 7: Listage des broches du connecteurs des capteurs fait par les auteurs.....	14
Illustration 8: Listage des broches du connecteurs du boîtier fait par les auteurs.....	14
Illustration 9: Listage des sortie de l'ATmega 8535 fait par les auteurs.....	15
Illustration 10: Schéma fonctionnel de niveau 1 réalisé par les auteurs.....	16
Illustration 11: Schéma fonctionnel de niveau 2 réalisé par les auteurs.....	16
Illustration 12: Schéma de fonctionnement[0].....	17
Illustration 13: Directives de pré-processeur réalisées par les auteurs.....	18
Illustration 14: Ordinogramme de sélection du mode réalisé par les auteurs.....	19
Illustration 15: Ordinogramme d'allumage des phares réalisé par les auteurs.....	20
Illustration 16: Ordinogramme de fonctionnement des clignotants réalisé par les auteurs.....	21
Illustration 17: Schéma du montage réalisé mesurant la valeur des lampes réalisé par les auteurs...	23
Illustration 18: Maquette de test, photo réalisée par les auteurs.....	24
Illustration 19: Montage de test de linéarisation photo réalisée par les auteurs.....	24
Illustration 20: Courbe pour les lampes de couleur rouge réalisée par les auteurs.....	26
Illustration 21: Zoom de la courbe précédente réalisée par les auteurs.....	26
Illustration 22: Courbe de variation des LED blanches réalisée par les auteurs.....	27
Illustration 23: Zoom de la courbe précédente réalisée par les auteurs.....	28
Illustration 24: Ordinogramme de la fonction de linéarisation réalisé par les auteurs.....	29
Illustration 25: Commandes au volant, photographie faite par les auteurs.....	30
Illustration 26: Connecteur des commandes au volant, photographie faite par les auteurs.....	30
Illustration 27: Antenne de la radio installée sur le kart, photo faite par les auteurs.....	31
Illustration 28: allumes cigares installés sur le kart, photo faite par les auteurs.....	31
Illustration 29: Schéma électrique du montage source de courant, fait par les auteurs sous Orcard.	32
Illustration 30: Carte phare avant GEII, photographie fait par les auteurs.....	33
Illustration 31: Phare de position avant IUT en fonctionnement sur le kart, photo faite par les auteurs.....	34
Illustration 32: Schéma électrique par bloc du phare dit "GEII", fait par les auteurs sous Orcad....	40
Illustration 33: Typon du phare de position avant "IUT", fait par les auteurs.....	41
Illustration 34: Typon du phare de position avant "GEII", fait par les auteurs.....	41

## Bibliographie

- [1] **Atmel**. *datasheet Atmega8535*, , [En ligne]. (Page consultée le 18/10/2011) <[http://www.atmel.com/dyn/resources/prod\\_documents/doc2502.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2502.pdf)>
- [2] **Le figao**. *Le marché de l'éclairage poursuit sa mue*, , [En ligne]. (Page consultée le 18/10/2011) <<http://www.lefigaro.fr/conso/2011/03/28/05007-20110328ARTFIG00692-le-marche-de-l-eclairage-poursuit-sa-mue.php>>
- [0] **Technique de l'ingénieur**. *Introduction aux systèmes embarquées*, , [En ligne]. (Page consultée le 18/10/2011) <<http://0-www.techniques-ingenieur.fr/portail.scd.univ-tours.fr/base-documentaire/mesures-analyses-th1/systemes-embarques-ti170/introduction-aux-systemes-embarques-h8000/>>
- [3] **ledux**. *Les autres avantages de la LED*, , [En ligne]. (Page consultée le 18/10/2011) <<http://www.ledux.fr/>>

# Annexes

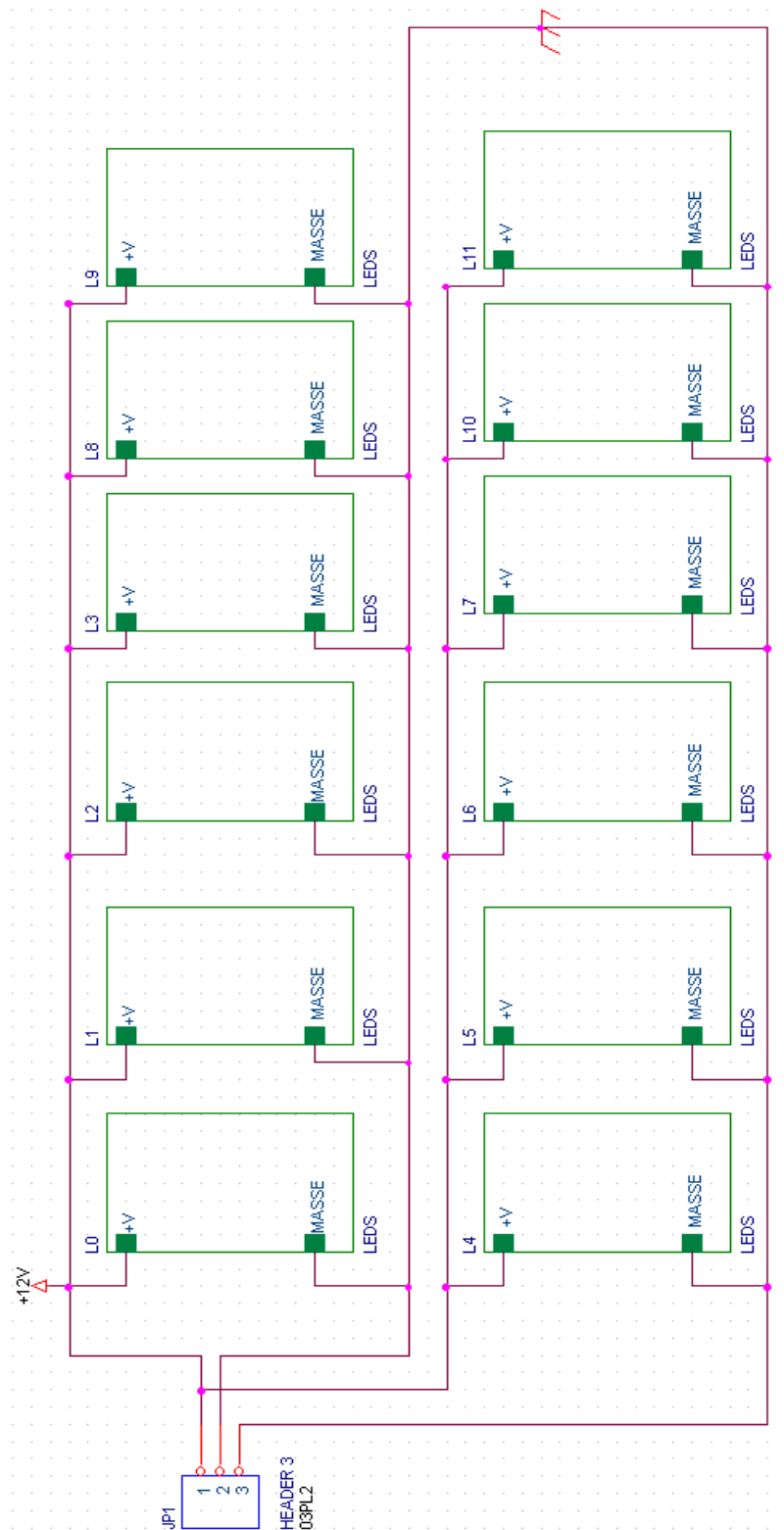
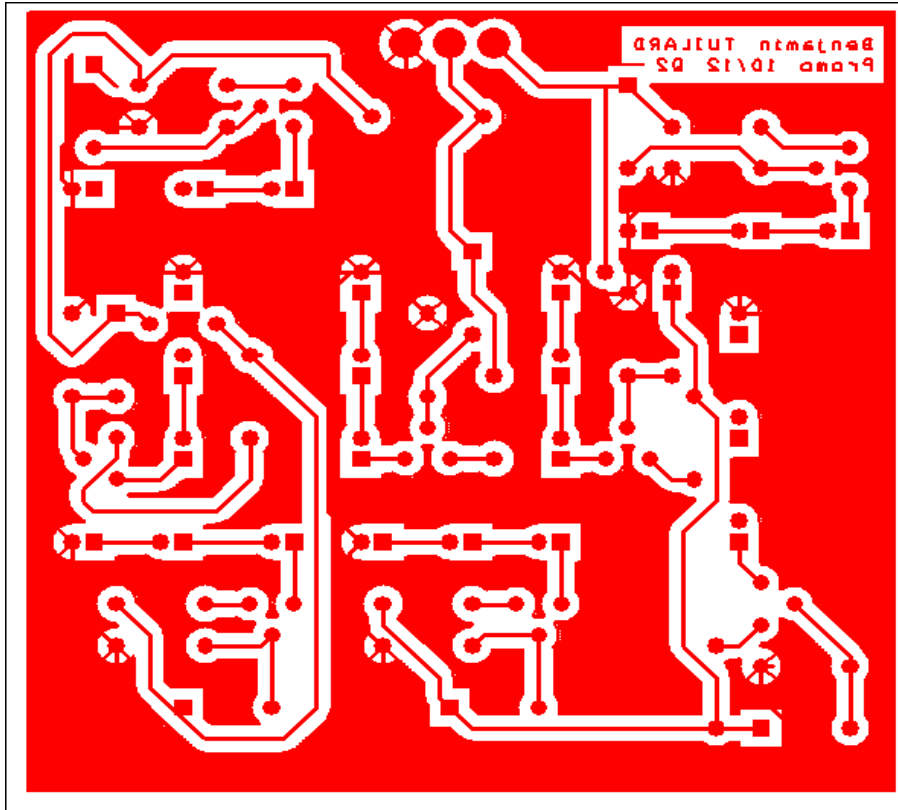
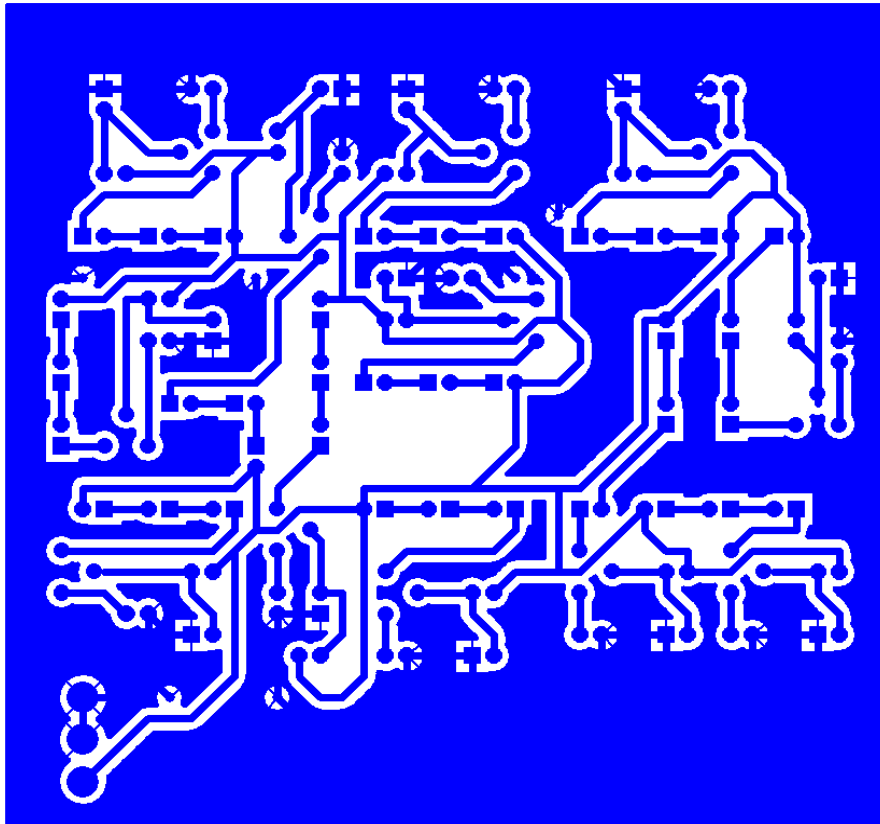


Illustration 32: Schéma électrique par bloc du phare dit "GEII", fait par les auteurs sous Orcad





*Illustration 33: Typon du phare de position avant "IUT", fait par les auteurs*



*Illustration 34: Typon du phare de position avant "GEII", fait par les auteurs*

## Nomenclature des composants des cartes phares avants

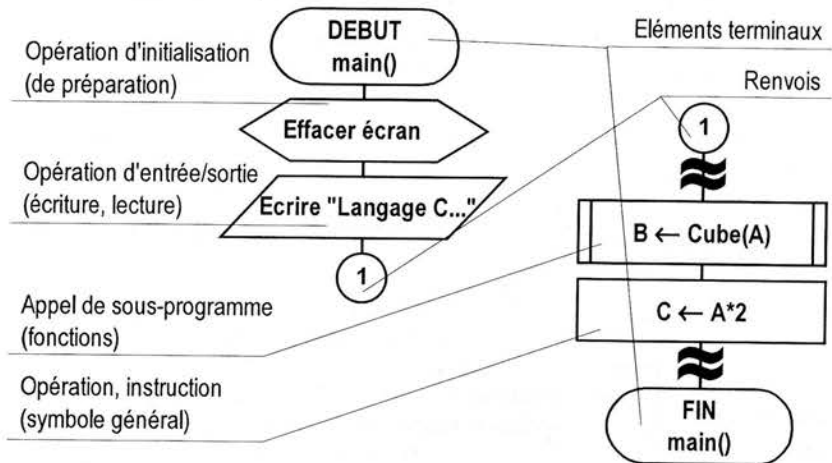
(cf illustration 26 page 31)

Désignation sur le schéma	Désignation	Valeur / Affectation	Nbre	Technologie	
D5,D6,D7	LED	3mm	60		
Q2	Transistor	2N2222	20	NPN TO18	
R3	Résistance	4,7 kOhm	20	5% 1/4W	
R4	Résistance	33 Ohm	20	5% 1/4W	
D8	Diode Zener	BZ86	20	1v4	

# ORGANIGRAMMES DE PROGRAMMATION ET LANGAGE C

D'après NF Z67-010 Août 1975

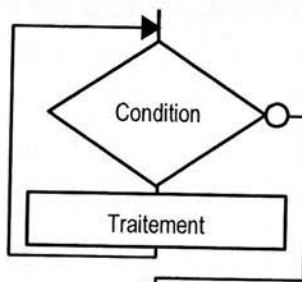
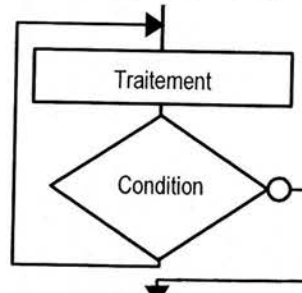
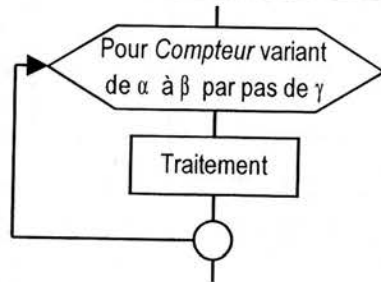
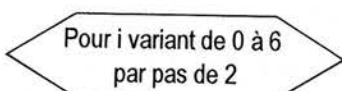
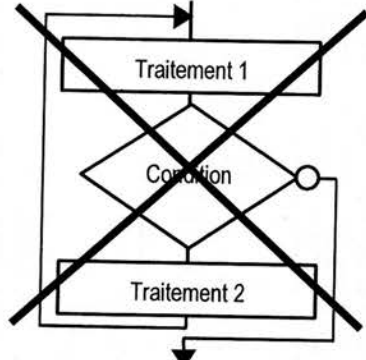
## Éléments terminaux et opérations



## Structures alternatives

Alternative simple		<pre>if ( Condition ) {     // Traitement1; }</pre>
Alternative double		<pre>if ( Condition ) {     // Traitement1; } else {     // Traitement2; }</pre>
Sélecteur de choix		<pre>switch (Sélecteur) {     case Val1:         // Traitement1;         break;     case Val2:         // Traitement2;         break;     case Val3:         // Traitement3;         break;     /* ... .. */     default:         // TraitementA; }</pre>

## Structures répétitives

<p>Tant que... (itération à condition d'entrée)</p>		<pre>while ( Condition ) {     // Traitement ; }</pre>
<p>Faire... tant que (itération à condition de sortie)</p>		<pre>do {     // Traitement ; } while ( Condition ) ;</pre>
<p>Pour... (itération avec compteur)</p>	 <p><b>Exemple :</b></p> 	<pre>for ( Init. ; Cond. ; Pas ) {     // Traitement ; }</pre> <p><b>Exemple :</b></p> <pre>for ( i=0; i&lt;=6; i=i+2 ) {     // Traitement ; }</pre>
<p>Structure non programmable (forme générale d'une structure d'itération)</p>		<p>Cette structure n'existe ni en C, ni en Pascal !</p>

## Remarques

Toutes les extrémités d'un symbole doivent posséder un chemin.

**Chaque structure de base n'a qu'une seule entrée et qu'une seule sortie.**

Les structures peuvent être imbriquées les unes dans les autres (une opération de traitement peut être composée d'une ou de plusieurs structures).

Il n'existe qu'un seul point d'entrée et qu'un seul point de sortie pour l'ordinogramme. Pour le programme principale, il peut ne pas exister de point de sortie (boucle infinie).

## Programme des feux

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

#define feux_recul PORTD.0
#define feux_temoin PORTD.1
#define croisement PORTD.2
#define CD PORTD.4
#define CG PORTD.3
#define stop PORTB.3
#define position PORTD.6
#define route PORTD.7

#define ON_OFF PINC.6
#define code PINC.5
#define plein PINC.4
#define ICG PINC.3
#define ICD PINC.2
#define Warning PINC.1
#define mode PINC.0
#define posi PINC.7

#define ADC_VREF_TYPE 0x20

// Lire le resultat de la convesion analogique
```

```

unsigned int read_adc(unsigned char adc_input)
{
ADMUX=adc_input|ADC_VREF_TYPE;
// Demarrer la conversion analogique
ADCSRA|=0x40;
// Attendre la fin de la conversion analogique
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCH;
}

```

```

void main(void)

```

```

{

```

```

    int tensionf;

```

```

    int tensionl;

```

```

    int i;

```

```

// Initialisation du Timer/Counter 0

```

```

// Source de l'horloge: horloge système

```

```

// Valeur de l'horloge: 16000,000 kHz

```

```

// Mode: Generateur de MLI

```

```

// Sortie OC0 : MLI non-inversée

```

```

TCCR0=0x61;

```

```

TCNT0=0x00;

```

```

OCR0=0x00;

```

```

// Initialisation Timer/Counter 2

// Source de l'horloge: horloge système

// Valeur de l'horloge: 16000,000 kHz // Mode: Generateur de MLI

// Sortie OC2 : MLI non inversée

ASSR=0x00;

TCCR2=0x61;

TCNT2=0x00;

OCR2=0x00;

// Initialisation des interruptions externes

// INT0: Off

// INT1: Off

// INT2: Off

MCUCR=0x00;

MCUCSR=0x00;

// Initialisation des interruption de(s) Timer(s)/Counter(s)

TIMSK=0x10;

// Initialisation du comparateur analogique // Comparateur analogique: Off

// Capture de l'entrée du comparateur analogique par le Timer/Counter 1: Off

ACSR=0x80;

SFIOR=0x00;

// Autorisation globale des interruptions

#asm("sei")

//Declaration et initialisation des ports d'entrée\sortie

```

```

PORTA=0x00;
DDRA=0x00;
PORTB=0x00;
DDRB=0x08;
PORTC=0x00;
DDRC=0x00;
PORTD=0x00;
DDRD=0xFF;
ADMUX=ADC_VREF_TYPE;
ADCSRA=0x87;
SFIOR&=0xEF;

while(1)
{
    //Verification de l'allumage des commandes
    if (ON_OFF==0)
    {
        //Verification du mode automatique
        if(mode==1)
        {
            //Lecture des entrées variables
            tensionf=read_adc(3);
            tensionl=read_adc(2);
            OCR0=((tensionf*175)/213)+20;
            OCR2=((tensionl*185)/255);
            if (OCR2>=50)
            {
                feux_temoin=1;
            }
        }
    }
}

```



```

    }
    else
    {
        feux_temoin=0;
    }

//Mode présentation
if (plein==0)
{
    feux_temoin=0;
    position=1;
    for (i=0; i<=255; i++)
    {
        OCR0=((i*175)/213)+20;
        OCR2=((i*185)/255);
        delay_ms(20);
    }
    feux_recul=1;
    delay_ms(500);
    feux_temoin=1;
    delay_ms(500);
    croisement=1;
    delay_ms(500);
    CD=1;
    CG=1;
    delay_ms(1000);
    for (i=0; i<=9; i++)
    {

```

```

        feux_recul=!feux_recul;

        feux_temoin=!feux_temoin;

        croisement=!croisement;

        CD=!CD;

        CG=!CG;

        delay_ms(100);
    }

    feux_recul=0;

    delay_ms(500);

    feux_temoin=0;

    delay_ms(500);

    croisement=0;

    delay_ms(500);

    CD=0;

    CG=0;

    delay_ms(500);
}

else
{
    position=0;

    feux_recul=0;

    feux_temoin=0;

    croisement=0;

    CD=0;

    CG=0;
}
}

//Mode manuel

```

```

else
{
    //Allumage des plein phare
if (plein==0)
    {
        OCR2=255;
        feux_temoin=1;
    }
//Extinction des plein phare
    else
    {
        OCR2=0;
    }
//Allumage des feux de croisement
    if ((code==0)|(plein==0))
    {
        croisement=1;
        feux_temoin=1;
    }
//Extinction des feux de croisement
    else
    {
        croisement=0;
    }
//Allumage des feux de position
if (posi==0)
{
    position=1;
}

```

```

        feux_temoin=1;
    }
    //Extinction des feux de position
    else
    {
        position=0;
    }
    //Extinction des phares arrières
        if ((code==1)&&(plein==1)&&(posi==1))
        {
            feux_temoin=0;
        }
    }
    //Allumage du clignotant droit
    if ((ICD==0)&&(Warning==1))
    {
        CD=!CD;
        delay_ms(500);
    }
    //Allumage du clignotant gauche
    if ((ICG==0)&&(Warning==1))
    {
        CG=!CG;
        delay_ms(500);
    }
    //Allumage des warnings
    if (Warning==0)
    {

```

```

        CD=!CD;

        CG=!CG;

        delay_ms(500);
    }

    //Extinction de tout les phares clignotants
    if ((Warning==1)&&(ICD==1)&&(ICG==1))
    {
        CD=0;

        CG=0;
    }
}

//Commande non allumée/ extinction de tout l'éclairage
else
{
    feux_recul=0;

    feux_temoin=0;

    croisement=0;

    CG=0;

    CD=0;

    OCR0=0;

    OCR2=0;

    position=0;

    OCR0=0;

    CD=0;

    CG=0;
}
}
}

```