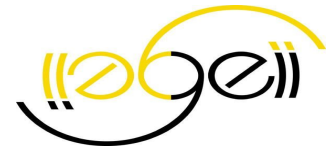


Gestion des feux de signalisation pour un kart électrique





Gestion des feux de signalisation pour un kart électrique

Sommaire

| | |
|---|----|
| Introduction..... | 4 |
| 1.Présentation du projet..... | 5 |
| 1.1.Cahier des charges..... | 5 |
| 1.2.Schéma fonctionnel..... | 6 |
| 1.3. Planning prévisionnel..... | 8 |
| 2.Étude du projet précédent..... | 8 |
| 2.1.Présentation..... | 8 |
| 2.2.Étude théorique de la carte..... | 9 |
| 2.3.Étude des capteurs..... | 13 |
| 3.Programmation informatique..... | 14 |
| 3.1.Organigramme principal..... | 14 |
| 3.2.Gestion des clignotants et des warnings | 16 |
| 3.4.Gestion du mode automatique..... | 17 |
| 4.Finalisation du projet..... | 19 |
| 4.1.Assignation des pins..... | 19 |
| 4.2.Boîtier de commande..... | 19 |
| 4.3.Carte pour feux avant..... | 20 |
| Conclusion..... | 22 |
| Résumé | 23 |
| Index des illustrations..... | 24 |
| Index des tables..... | 25 |
| Bibliographie..... | 26 |
| Annexe..... | 27 |

Introduction

Au cours de notre 4^{ème} semestre de formation, nous avons réalisé un projet dans le cadre du cours d'Étude et Réalisation.

Notre projet a pour objectif d'améliorer les études faites par les étudiants des années précédentes. Nous allons donc optimiser au maximum le programme informatique permettant la gestion des feux de signalisation du kart. Nous prendrons soin de réparer et d'améliorer les éléments du kart ne fonctionnant pas correctement, comme notamment le câblage ou les cartes électroniques assurant l'éclairage du véhicule.

Pour cela nous ferons dans un premier temps une présentation du projet. Ensuite nous ferons une étude approfondie du projet fait précédemment par les autres étudiants. Dans une troisième partie nous effectuerons la programmation informatique gérant les feux de signalisation. Finalement, nous présenterons la partie électronique que nous avons optimisé.

1. Présentation du projet

Le sujet consiste en la gestion du fonctionnement de feux de signalisation avant et arrière pour kart électrique.

1.1. Cahier des charges

Le sujet consiste en la gestion du fonctionnement de feux de signalisation avant et arrière pour kart électrique.

Pour cela nous devons respecter diverses contraintes :

- L'utilisation d'un micro contrôleur ATmega8535
- Étude approfondie de la carte déjà élaborée. Étudier son fonctionnement, les parties du montage à améliorer.
- Étude du circuit de commande :
 - Interrupteur deux positions (fonctionnement manuel/automatique)
 - Interrupteur trois positions (clignotant gauche/arrêt/droit)
 - Interrupteur trois positions (feux position/arrêt/feux de route)
 - Interrupteur deux positions (warning/arrêt)
- Vérification de la programmation du micro contrôleur :
 - Feux de position/croisement
 - Éclairage variable en mode automatique
 - Clignotants
 - Warnings
 - Feux stop
 - Éventuels feux de recul

Nous utiliserons le logiciel Code Vision AVR pour implanter notre programme dans l'ATmega8535.

Nous essayerons de câbler les feux sur le kart.

1.2. Schéma fonctionnel

Dans cette partie nous expliquerons le fonctionnement de la carte et son rôle vis-à-vis du kart.

1.2.1. Schéma fonctionnelle niveau 1

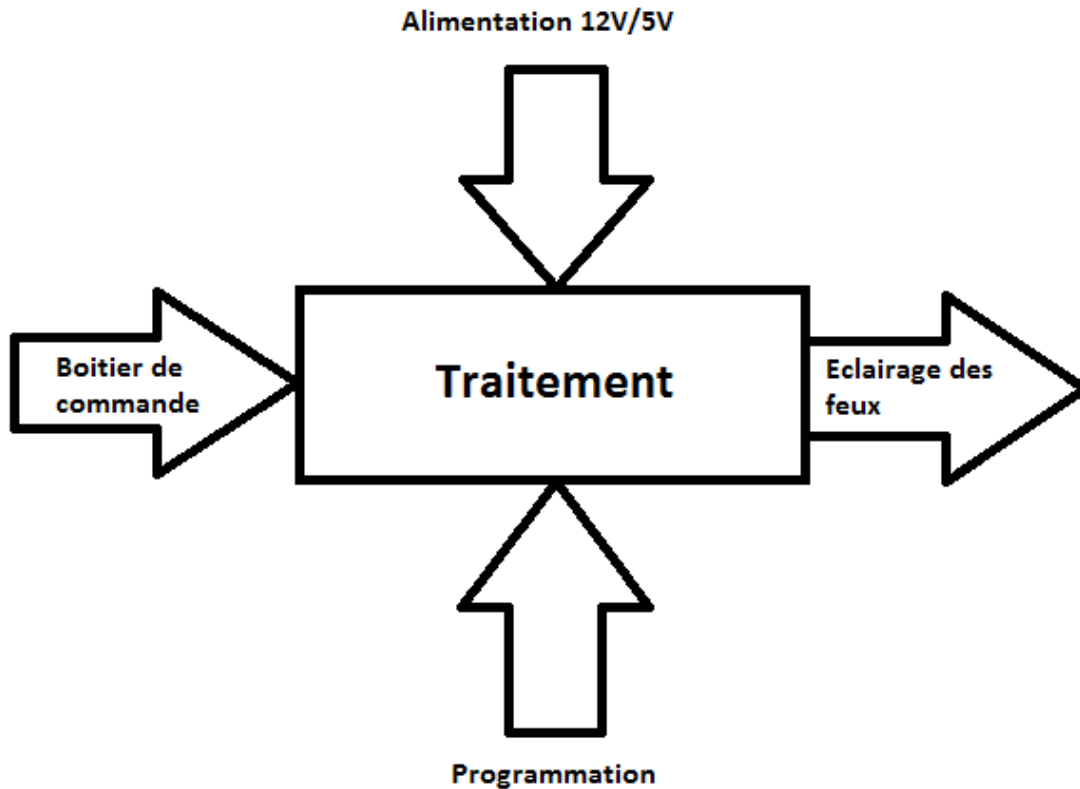


Illustration 1: Schéma fonctionnel de niveau 1

On retrouve ici la fonction globale de notre projet:

Le boîtier de commande transmet les différentes informations aux feux.

Traitement : les informations transmises par le boîtier de commande sont ensuite traitées par la carte électronique contrôlant l'éclairage et la signalisation du kart électrique.

Alimentation : cette carte est quant à elle alimentée en 12V pour ce qui concerne les différents composants de la carte sauf pour l'ATMega8535 qui doit être alimenté en 5V.

L'éclairage et le **traitement** des informations transmises par le boîtier de commande est géré via la programmation de l'**ATMega8535**.

1.2.2. Schéma fonctionnel niveau 2

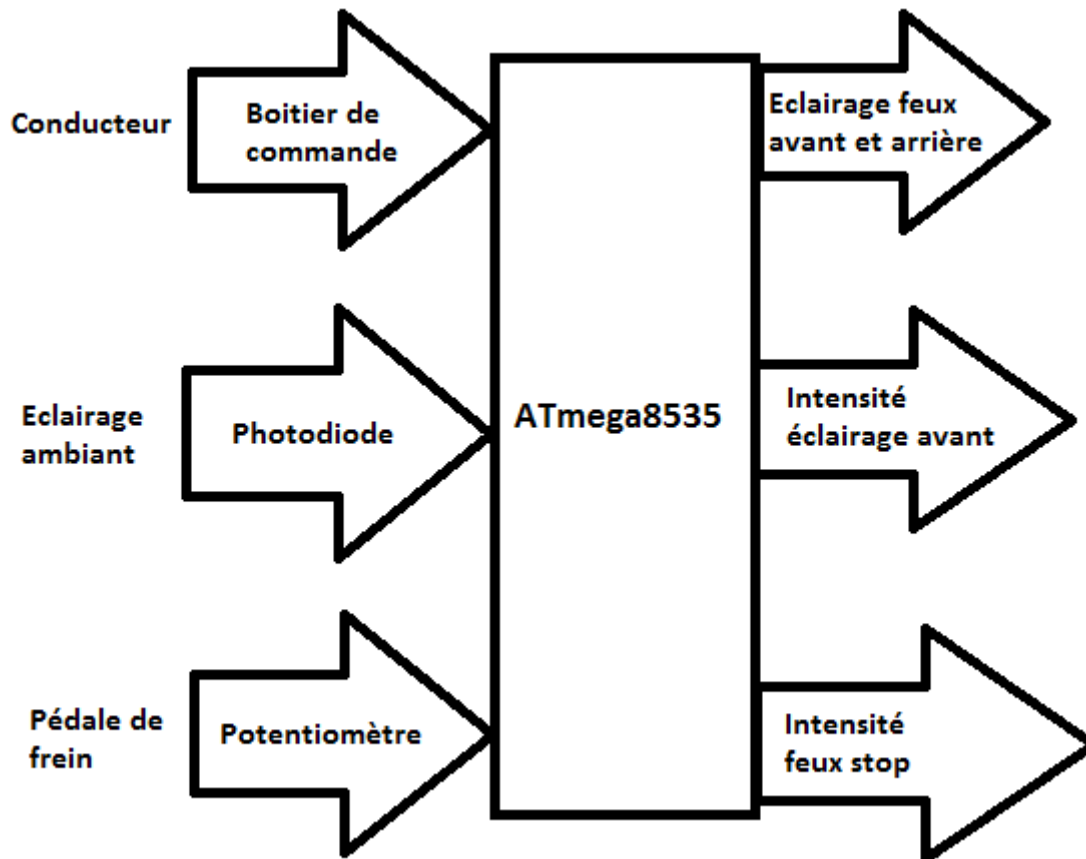


Illustration 2: Schéma fonctionnel de niveau 2

On retrouve ici la fonction d'usage de notre projet:

Alimentation 12V/5V : notre système ayant comme première application d'être implanté dans un kart électrique comportant une batterie de 12V, l'ensemble des composants de la carte sont alimentés en 12V continu sauf l'AtMega8535 qui doit être alimenté en 5V.

Le boîtier de commande : transmet les différentes informations permettant l'éclairage des feux avant et arrière.

La photo-résistance : permet l'éclairage progressif des feux avant en fonction de l'éclairage ambiant.

Le potentiomètre : permet de contrôler l'éclairage progressif des feux stop en fonction de l'appui progressif sur la pédale de frein.

ATmega8535 : permet la programmation des différents éclairages du kart électrique.

1.3. Planning prévisionnel

Afin d'organiser correctement notre projet, nous avons réalisé un planning prévisionnel de la répartition des tâches. Cela nous permet de retracer l'évolution du projet.

| PLANNING PREVISIONNEL | | | | | | | | | |
|--|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Taches/Semaines | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 |
| Choix du sujet | Prévisionnel | | | | | | | | |
| Recherche d'information | Réel | Prévisionnel | | | | | | | |
| Élaboration du cahier des charges et du planning | Réel | | | | | | | | |
| Formation Orcad | | Prévisionnel | | | | | | | |
| Recherche de solution | Prévisionnel | Réel | Prévisionnel | Prévisionnel | Prévisionnel | | | | |
| Réalisation d'un éventuel typhon | | | Prévisionnel | Prévisionnel | | Réel | | Réel | |
| Test et vérification | | | Prévisionnel | Prévisionnel | Prévisionnel | Prévisionnel | | | |
| Rédaction du rapport | | | | Prévisionnel | Prévisionnel | Prévisionnel | Prévisionnel | Réel | |
| Essai du prototype | | | | | | | Prévisionnel | Prévisionnel | Réel |
| Remise des dossiers | | | | | | | | Prévisionnel | Réel |
| Soutenance orale | | | | | | | | | Prévisionnel |

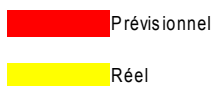


Tableau 1: Planning prévisionnel

2. Étude du projet précédent

2.1. Présentation

Le projet précédent consistait en la réalisation d'une carte électronique qui contrôle l'éclairage et la signalisation d'un kart électrique.

Cette carte gère deux modes de fonctionnement :

- Mode manuel : gestion manuelle des feux.
- Mode automatique : éclairage variable en fonction de la luminosité et de l'appui sur la pédale de frein.

2.2. Étude théorique de la carte

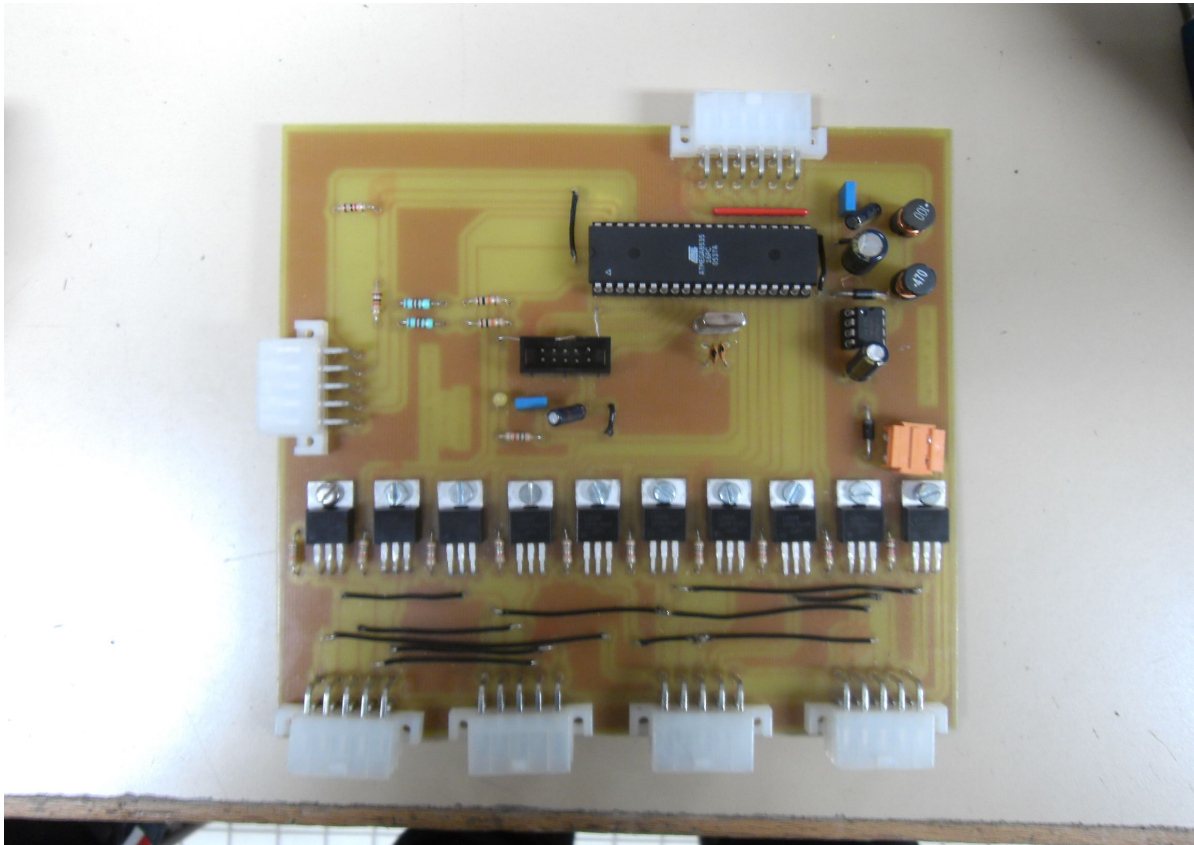


Illustration 3: carte électronique : vue de haut

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

2.2.1. Le fonctionnement de la carte

- **LM2574 :**

Le micro-contrôleur 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 puissance qu'il y aurait eu si un composant linéaire tel qu'un régulateur linéaire avait été utilisé.

- **L'ATMega8535 :**

C'est un micro-contrôleur 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 (cf partie 2.2.2). C'est par l'intermédiaire de ce composant que nous implanterons le programme.

- Quartz :

C'est un composant qui possède la propriété d'osciller à une fréquence stable et précise lorsqu'il est alimenté. Le quartz utilisé est cadencé à 16MHz. Il permet de remplacer l'horloge interne de l'ATMega8535 qui est cadencé à une fréquence moins élevée.

- **Transistors MOSFET :**

Ils servent d'interrupteur de sortie. En effet ils possèdent 3 pattes (grille(1), drain(2), source(3)). La grille est reliée directement à l'ATMega8535 et recevra des impulsions, le drain est relié à une patte d'une des lampes (l'autre patte 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 pattes n'est plus connecté (lampe : une patte au +12Vest une patte non connectée).

- **Connecteurs de sorties :**

Ces connecteurs seront directement branchés aux LED et permettront la liaison entre les LED et la carte.

- **Connecteur de commande :**

Connecteur branché au boîtier de commande et aux capteurs, il permet la liaison entre les interrupteurs, les capteurs et la carte. [1]

- **Lampes :**

Dans les précédents projets des lampes à LED ont été utilisées afin de permettre l'éclairage du kart. Ces lampes à LED ont une tension d'alimentation de 12V, ce qui correspond à la tension d'alimentation de la carte électronique (par le kart), et une puissance de 1W. Les lampes à LED nous permettent aussi de varier les couleurs afin d'avoir des feux de la couleur correspondant à la demande. De plus les lampes à LED ont une faible consommation d'énergie ce qui permet de préserver l'autonomie de la batterie. Ces lampes seront commandées via les transistors MOSFET. [2]

2.2.2. Le micro-contrôleur ATmega8535

L'ATmega8535 est un micro-contrôleur 8bits fabriqué par la société Atmel, qui se présente sous la forme d'un circuit intégré 40 pattes.

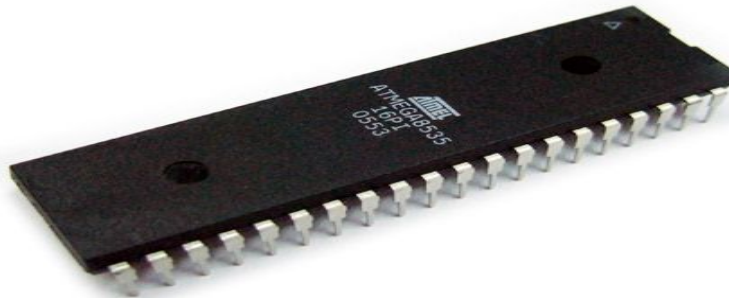


Illustration 4: Atmega8535 [3]

L'ATmega8535 possède 4 ports A, B, C, D, ayant chacun des fonctions particulières et contenant 8 bits numérotés de 0 à 7. Pour chacun de ces ports, les pattes peuvent être configurées en entrées ou sorties. [Illustration 5]

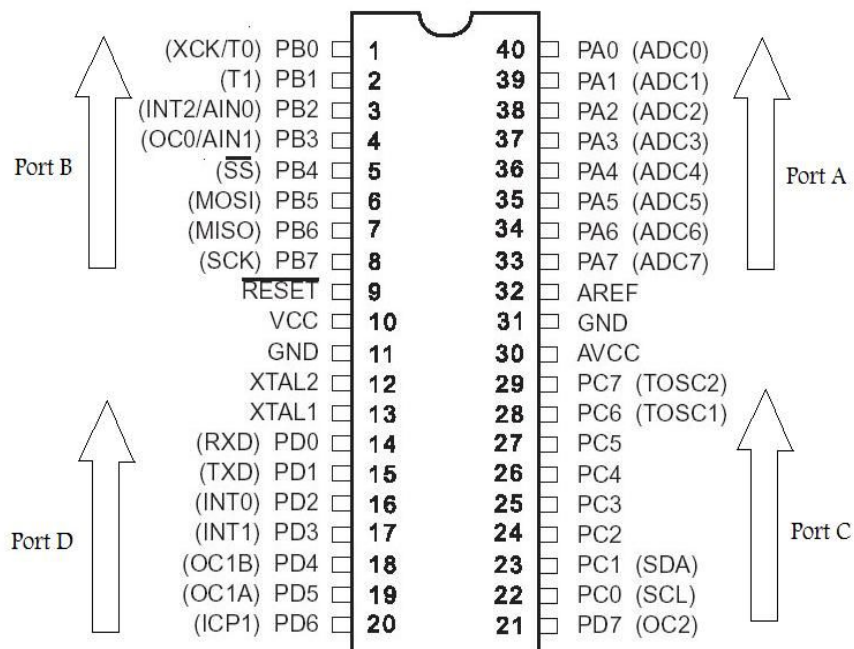


Illustration 5: Brochage Atmega8535 [4]

- **RESET** : cette patte permet de réinitialiser le micro-contrôleur, il faut appliquer une tension de VCC pour inhiber le reset.
- **AVCC** : c'est une patte d'alimentation pour le CAN1, elle doit être reliée au VCC via un filtre passe-bas qui élimine les parasites.
- **AREF** : c'est l'entrée de référence analogique du CAN.
- **AGND** : c'est la masse analogique.
- **MISO, MOSI, SCK** : pattes de programmation du micro-contrôleur.
- **INT0, INT1** : Interruptions externes.
- **VCC** : c'est la broche d'alimentation du micro-contrôleur.
- **GND** : c'est la masse de alimentation. [1]

2.2.2.1. Le CAN¹

Le CAN ou convertisseur analogique numérique, permet d'obtenir l'image d'une tension sur un certain nombre de bits.

Nous allons utiliser le CAN pour l'acquisition de la tension retournée par les deux capteurs.

En effet la photo-résistance nous renvoie une tension proportionnelle à l'éclairage ambiant, et potentiomètre mécanique nous renvoie une tension proportionnelle à la position de la pédale. [1]

2.2.2.2 La MLI²

La MLI ou modulation en largeur d'impulsion est un moyen d'obtenir une tension moyenne variable à partir d'une alimentation continue. Elle repose sur la modulation du rapport cyclique α .

Le rapport cyclique est calculé par la formule suivante : $\alpha = T / \text{Période}$.

Ainsi, à 100% la tension moyenne est maximale tandis qu'à 0% la tension est nulle. Nous allons utiliser la MLI pour moduler l'éclairage des feux stop en fonction de la position de la pédale de frein. [1]

1 Convertisseur analogique numérique

2 Modulation de largeur d'impulsion

2.3. Étude des capteurs

2.3.1. Photo-résistance

C'est un composant électronique ayant la propriété d'avoir une résistance qui varie en fonction de la luminosité. Nous allons l'utiliser pour faire varier la luminosité des feux en fonction de la lumière ambiante. [1]

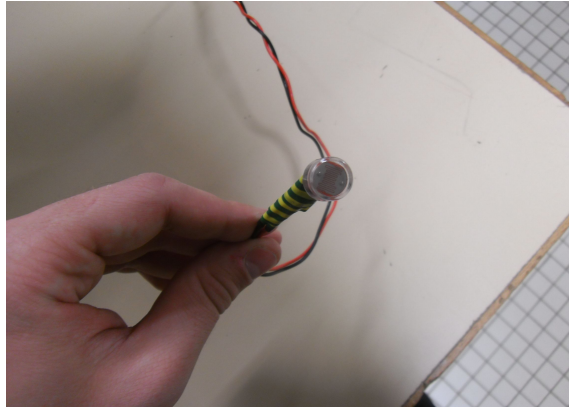


Illustration 6: Photo-résistance

2.3.2. Le potentiomètre mécanique

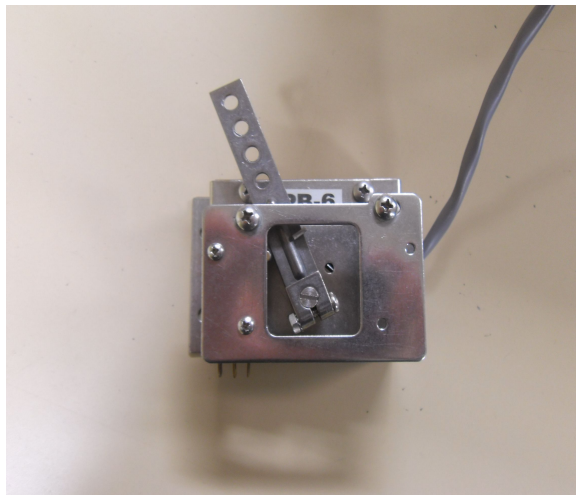


Illustration 7: Potentiomètre mécanique

C'est un composant mécanique ayant la propriété d'avoir une résistance qui varie en fonction de la position de la pédale. Nous allons l'utiliser pour faire varier la luminosité des feux stop en fonction de la position de la pédale de frein. [1]

3. Programmation informatique

Dans cette partie nous décrivons toute la partie informatique du projet.

3.1. Organigramme principal

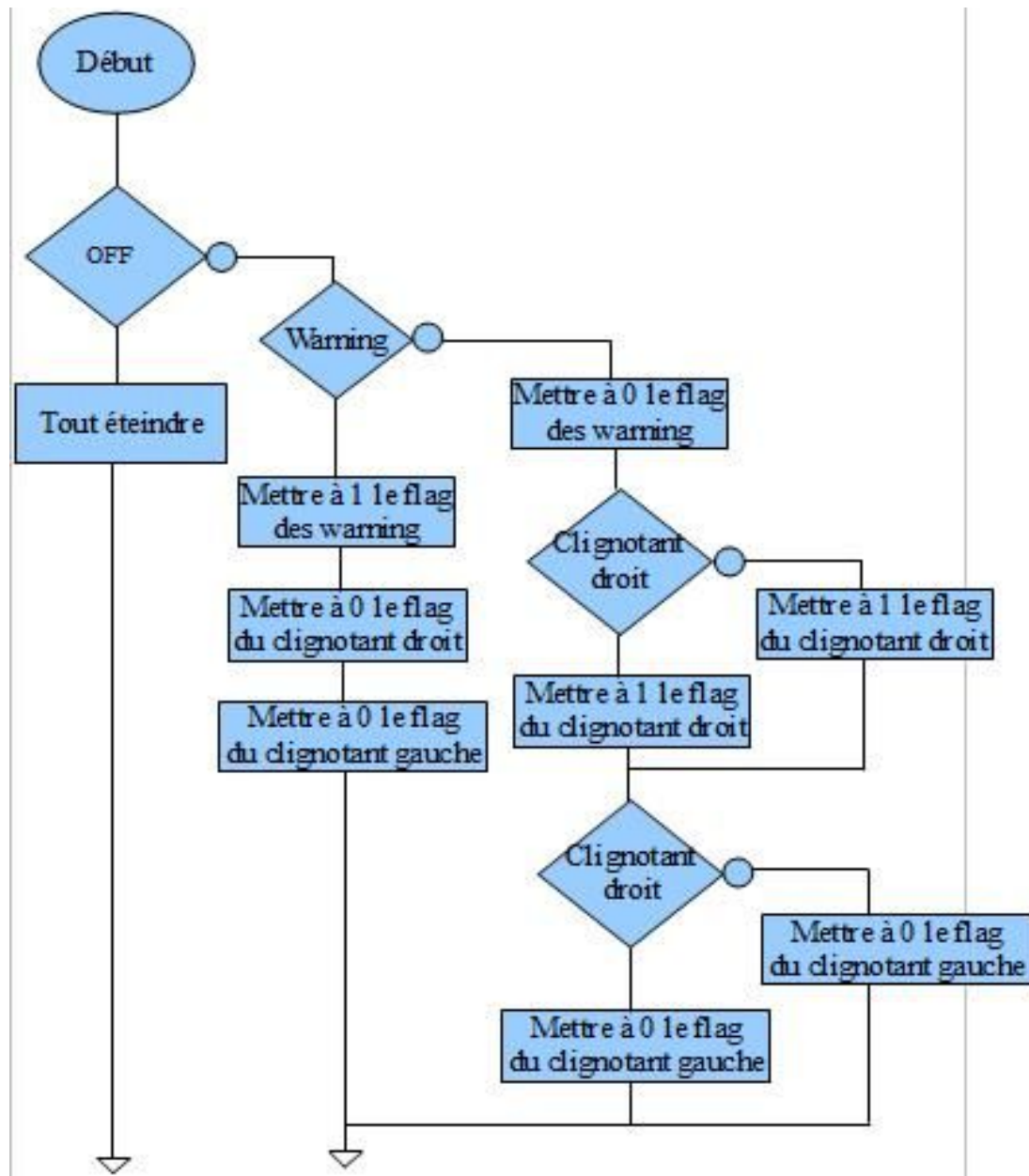


Illustration 8: Ordinogramme principal début

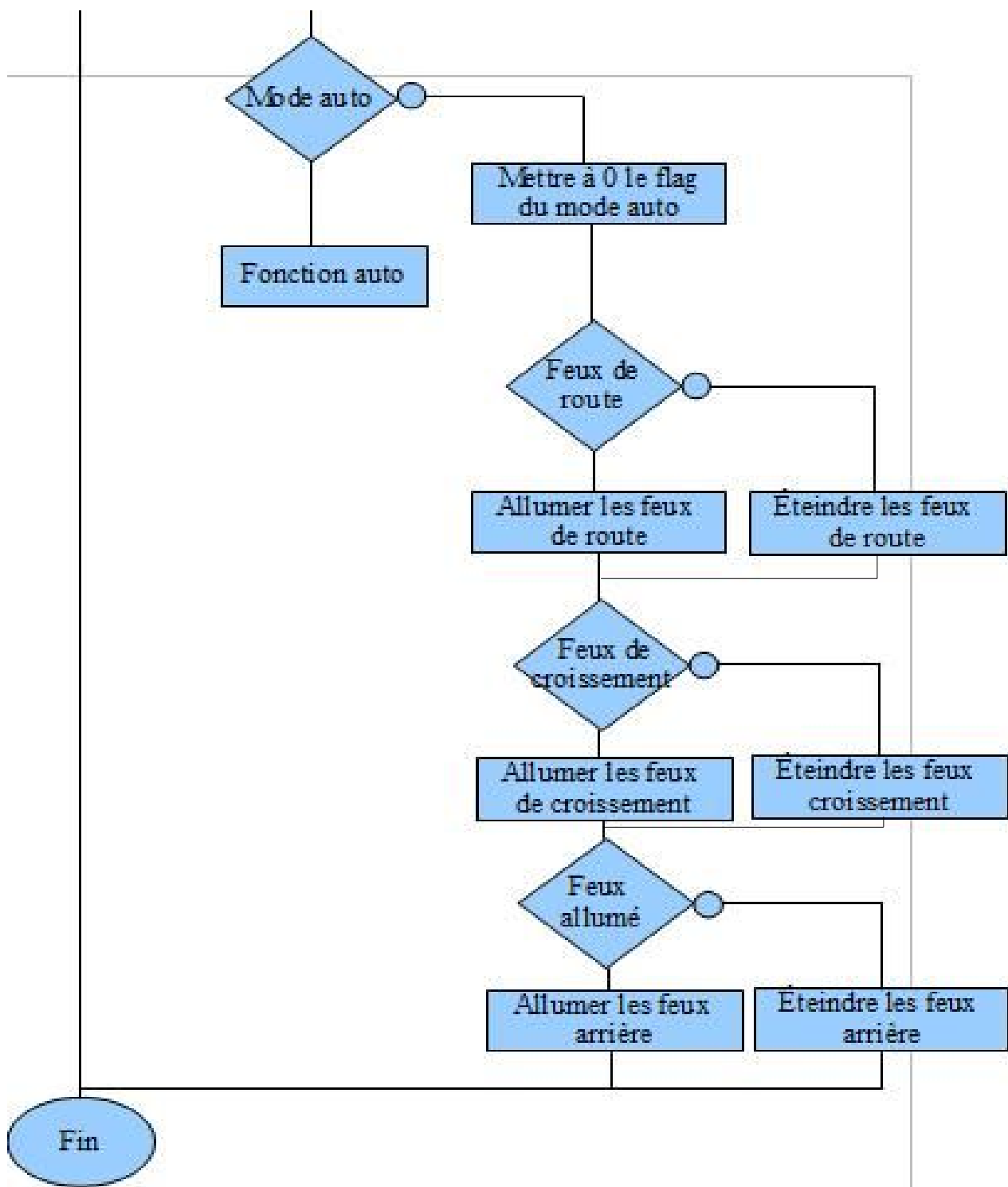


Illustration 9: Ordinogramme principal fin

Le programme est implanté dans une boucle infinie. Les flags des warnings et des clignotants sont utilisés dans une fonction d'interruption interne qui permet de les faire clignoter.

3.2. Gestion des clignotants et des warnings

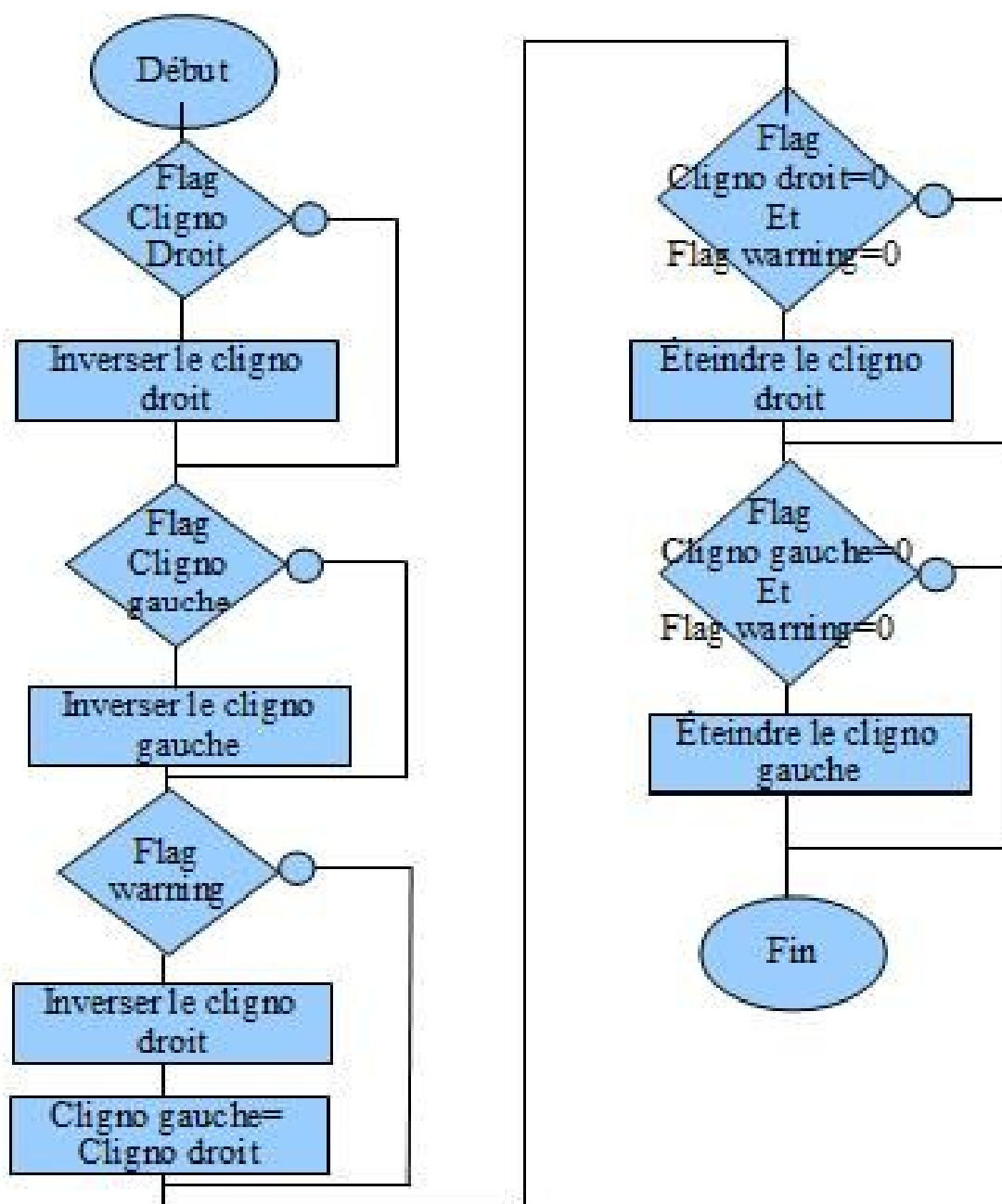


Illustration 10: Ordinogramme de la fonction d'interruption

Une fonction d'interruption a été utilisée pour gérer les clignotants les warnings afin de les rendre indépendants des autres feux du kart. Cette fonction d'interruption s'exécute toutes les 0,5 secondes.

3.3. Gestion du mode automatique

Ce mode a pour objectif, lorsqu'il est activé, d'adapter l'intensité lumineuse des feux en fonction de la luminosité de l'environnement ainsi que de gérer l'éclairage progressif des feux stop en fonction de l'appui sur la pédale de frein. Dans ce mode les interrupteurs permettant de gérer l'éclairage des feux sont désactivés par le biais d'une fonction d'interruption. Cette fonction recevra un numéro de voie en paramètre et devra lire la tension sur cette voie et la convertir en valeur numérique :

Pour configurer le CAN :

```
// ADC interrupt service routine
// Read the 8 most significant bits
// of the AD conversion result
unsigned char read_adc(unsigned char adc_input)
{
    ADMUX=adc_input|ADC_VREF_TYPE;
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCH;
}
```

Un filtrage numérique a été réalisé afin d'éviter tout parasites provenant de la pédale de frein.

```
Moyenne=Moyenne+read_adc(3);
N++;
    if(N>=100)
    {
        OCR0=(Moyenne/100)/2.15;
```

```
Moyenne=0;
```

```
N=0;
```

```
}
```

La modification du registre OCR2 permet de modifier la valeur du rapport cyclique du timer et ainsi faire varier la valeur moyenne de la tension de sortie. La valeur a été corrigée en la divisant par 2,5 pour adapter la caractéristique des diodes. Cela a permis de dilater la zone de variation de la luminosité en fonction de l'appui sur la pédale de frein et donc d'obtenir une variation plus douce sur la plage de mouvement de la pédale.

Pour la variation de l'intensité lumineuse des feux avant en fonction de la lumière environnante la même technique a été utilisée afin d'obtenir un éclairage maximal de nuit et minimal en plein jour :

```
J++;
```

```
if(J>=100)
```

```
{
```

```
luminosite=(luminosite/100);
```

```
OCR2=(((luminosite-70)/2) / (10*exp(-(luminosite-70)/4)+1));
```

```
luminosite=0;
```

```
J=0;
```

```
}
```

[1]

4. Finalisation du projet

4.1. Assignment des pins

La table de routage des feux précédente n'était pas conforme nous en avons donc fait une nouvelle :

| Feux arrière gauche | Feux arrière droit | Feux avant gauche | Feux avant droit |
|--|--------------------|-------------------|------------------|
| Cg . Fa Fs Fs | Cd . Fa Fs Fs | F Cg Fa Fs Fs | F Cd Fa Fs Fs |
| F = Feux de route, Cd/Cg = Clignotants gauche/droit, Av = Feux avant, C = Feux de croisement, Fa = Feux arrière, Fs = Feux de stop | | | |

Tableau 2: Table de routage des feux

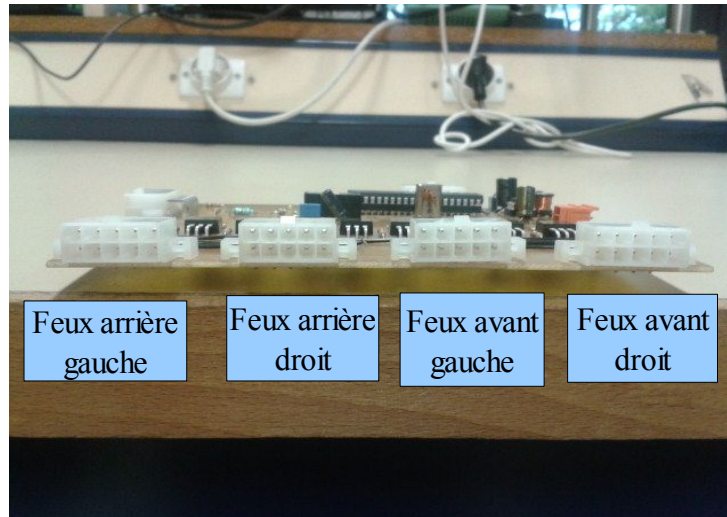


Illustration 11: carte électronique vue de face

4.2. Boîtier de commande

La maquette est commandée par un boîtier de commande conçu par un groupe d'étudiants en 2009. ce boîtier contient les interrupteurs correspondants aux différents feux de signalisation du kart électrique.



Illustration 12: boîtier de commande

Le câblage à l'intérieur a été réalisé avec des fils rigides soudés sur un connecteur DB15 afin de communiquer avec la carte électronique.

Le bouton ON/OFF du boîtier de commande ne fonctionnait pas correctement, lorsque l'on appuyait sur OFF les lumières restaient tout de même allumées. Cela était dû à un problème de programmation que nous avons résolu. [1]

4.3. Carte pour feux avant

Lors des essais sur le kart nous nous sommes aperçus que les LED des cartes pour feux avant ne s'éclairaient pas toutes à la même intensité. Nous avons testé ces cartes pour savoir d'où venait le problème. Ces variations de luminosité étaient dû au fait que le courant, dans chaque branche contenant les LED, n'était pas constant.

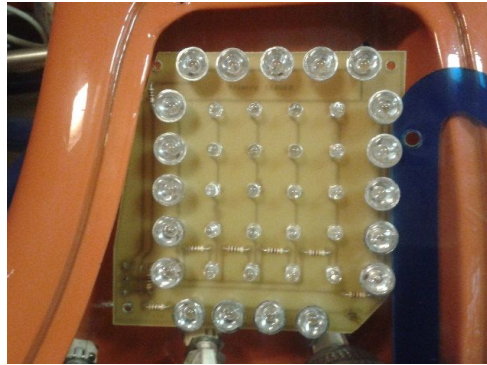


Illustration 13: Carte pour feux avant

Nous avons donc dû créer un nouveau montage dans lequel le courant serait constamment égal à 20mA. Pour cela il a fallu créer un générateur de courant à l'aide d'un transistor et d'une diode Zener. La carte, devant être alimentée en 12V, nous avons dû réduire le nombre de LED à trois car celles-ci peuvent consommer jusqu'à 3,5V chacune (pour quatre LED les 12V ne seraient donc pas suffisants).

-Les LED utilisées dans le montage ont une plage de tension qui varie entre 2,9-3,5V pour un courant de 20mA, donc pour 3 LED $V_{D_{totale}}=10,5V$ au maximum d'où:

- $R1=VR1/I$ avec $VR1=10,5V$ et $I=20mA$.
 $R1=5,2Kohm$ on prendra $R1=4,7Kohm$ en valeur normalisée.

- $VR2=V_z-V_{be}=0,7V$ d'où $R2=VR2/I$ avec $I=20mA$ donc $R2=35ohm$ on prendra $R2=33ohm$ en valeur normalisée.

-Pour la diode zener l'on prendra une diode 1V4, qui permet un fonctionnement normal pour une tension de 1,4V.

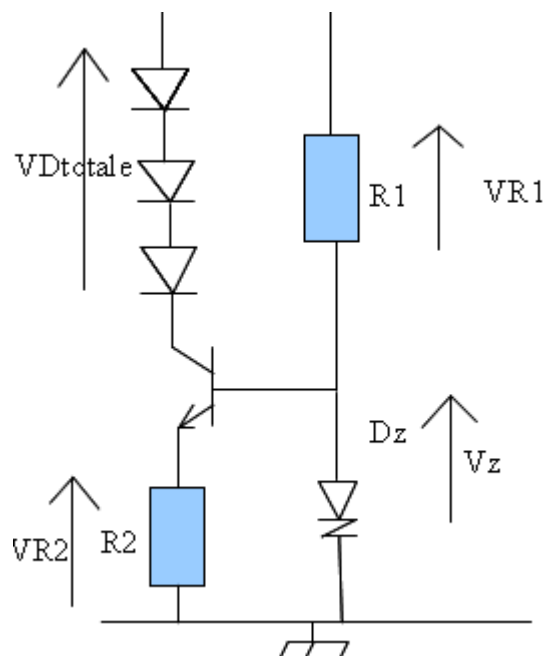


Illustration 14: Schéma du montage pour carte feux avant

4.3.1. Typon

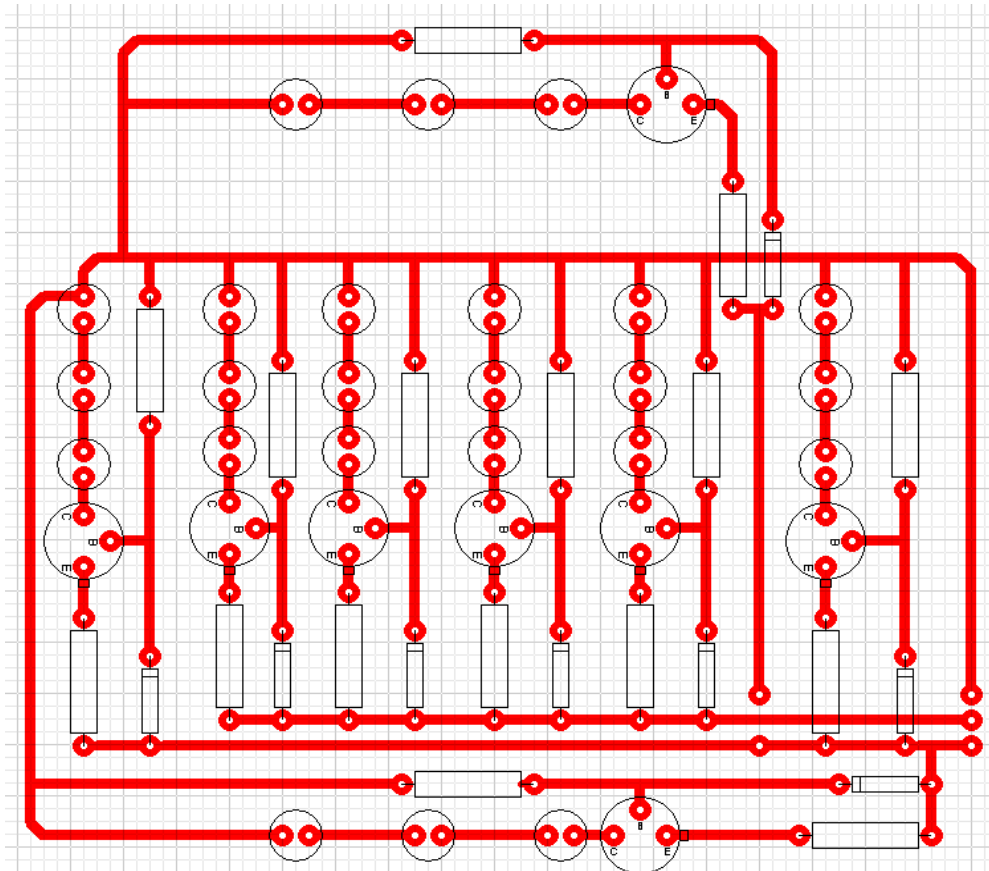


Illustration 15: Typon de la carte pour feux avant

Conclusion

Le but de ce projet était de mener à terme l'éclairage d'un kart électrique de l'IUT GEII de Tours. Pour cela nous avons établi un cahier des charges pour tenter de mener à bien ce projet.

Nous avons cherché à améliorer le programme déjà existant. Nous avons installé la carte électronique sur le kart permettant d'obtenir l'éclairage souhaité. Il a fallu régler divers problèmes de câblage et autres soucis qui rendaient l'esthétique de l'éclairage désagréable.

Dans l'ensemble nous avons accompli les objectifs que nous nous étions fixés, le programme fonctionne comme nous le souhaitions, les cartes à LED pour l'éclairage du kart sont fonctionnelles.

Ce projet nous a permis de découvrir le logiciel CodeVision AVR et d'apprendre à implanter le programme dans le microcontrôleur AtMega8535. Ce travail nous a permis de revoir nos compétences en électronique, et d'être efficace en peu de temps.

Résumé

Pour ce projet réalisé dans le cadre du semestre 4 nous avons dû réaliser notre propre cahier des charges ainsi que notre planning prévisionnel.

Dans un premier temps nous avons étudié les projets réalisés précédemment grâce à divers dossiers d'étudiants . Nous avons repris le projet là où les derniers étudiants l'avaient laissé et avons étudié les choses à améliorer.

Ensuite nous avons implanté le programme des derniers étudiants ayant effectué le projet dans l'AtMega8535 de la carte gérant les feux de signalisation du kart. Puis nous avons fait des tests sur une maquette tenant un jeu de lampes. Nous avons modifié les parties du programme ne fonctionnant pas correctement.

Par la suite, nous avons installé la carte gérant les feux de signalisation du kart sur celui-ci. Nous avons passé quelques séances à améliorer le câblage reliant la carte aux feux.

Finalement, nous avons décidé d'améliorer les feux avant en créant une nouvelle carte à LED plus performante.

Index des illustrations

| | |
|--|----|
| Illustration 1: Schéma fonctionnel de niveau 1..... | 6 |
| Illustration 2: Schéma fonctionnel de niveau 2..... | 7 |
| Illustration 3: carte électronique : vue de haut..... | 9 |
| Illustration 4: Atmega8535 [3]..... | 11 |
| Illustration 5: Brochage Atmega8535 [4]..... | 11 |
| Illustration 6: Photo-résistance..... | 13 |
| Illustration 7: Potentiomètre mécanique..... | 13 |
| Illustration 8: Ordinogramme principal début..... | 14 |
| Illustration 9: Ordinogramme principal fin..... | 15 |
| Illustration 10: Ordinogramme de la fonction d'interruption..... | 16 |
| Illustration 11: carte électronique vue de face..... | 19 |
| Illustration 12: boîtier de commande..... | 19 |
| Illustration 13: Carte pour feux avant..... | 20 |
| Illustration 14: Schéma du montage pour carte feux avant..... | 20 |
| Illustration 15: Typon de la carte pour feux avant..... | 21 |

Index des tables

| | |
|---|----|
| Tableau 1: Planning prévisionnel..... | 8 |
| Tableau 2: Table de routage des feux..... | 19 |

Bibliographie

[1] : P.VIEYRA, P. TALLET-PINET, Éclairage 2010, (page consulté le 28 octobre 2011), <<http://www.thierry-lequeu.fr/data/RAP-VIEYRA-TALLET-PINET.pdf>>

[2] : F.LAMBERT, M.N. BIN ROSLI, Éclairage 2010, (page consulté le 28 octobre 2011), <<http://www.thierry-lequeu.fr/data/RAP-LAMBERT-BIN-ROSLI.pdf>>

[3] : ELECTRICLY, Interrupt Vectors in ATmega8535, (page consulté le 4 novembre 2011), <<http://electricly.com/interrupt-vectors-in-atmega8535/>>

[4] : ELECTRICLY, All About ATmega8535, (page consulté le 4 novembre 2011), <<http://electricly.com/all-about-atmega8535/>>

Annexe

Programme principal

/*****

This program was produced by the

CodeWizardAVR V1.24.2c Professional

Automatic Program Generator

© Copyright 1998-2004 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.ro>

e-mail:office@hpinfotech.ro

Project :

Version :

Date : 22/09/2011

Author : F4CG

Company : F4CG

Comments:

Chip type : ATmega8535

Program type : Application

Clock frequency : 8,000000 MHz

Memory model : Small

External SRAM size : 0

Data Stack size : 128

*****/

#include <mega8535.h>

```

#include <stdio.h>
#include<delay.h>
#include <math.h>
//variables globales
//Déclaration des Sorties

#define feux_recul PORTD.0
#define feux_arriere PORTD.1
#define feux_avant PORTD.2
#define cligno_g PORTD.3
#define cligno_d PORTD.4
#define feux_stop PORTB.3
#define feux_route PORTD.7
#define croisement PORTD.6
//Déclaration des Entrées
#define BP_cligno_d PINC.3 //INVERSE
#define BP_cligno_g PINC.2 //INVERSE
#define BP_warning PINC.1 //INVERSE
#define BP_mode PINC.0 // AUTO/MANU
#define BP_feux_route PINC.4 //INVERSE
#define BP_feux_crois PINC.5 //INVERSE
#define BP_eclairage PINC.6 // ON/OFF

#define ADC_VREF_TYPE 0x20
unsigned char Cligno_D;
unsigned char Cligno_G;
unsigned char Warning;

```

```

float Moyenne=0;
float luminosite=0;
int N=0, J=0;
int FlagTunnig1;
int FlagTunnig2;

// ADC interrupt service routine
// Read the 8 most significant bits
// of the AD conversion result
unsigned char read_adc(unsigned char adc_input)
{
    ADMUX=adc_input|ADC_VREF_TYPE;
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCH;
}

interrupt [TIM1_COMPA] void timer1_compa_isr(void)
{
    // Place your code here

    if(Cligno_D)
    {
        cligno_d=!cligno_d;
    }
}

```

```
}  
if(Cligno_G)  
{  
    cligno_g=!cligno_g;  
}  
  
if(Warning)  
{  
    cligno_d=!cligno_d;  
    cligno_g=cligno_d;  
}  
if((Warning==0)&&(Cligno_D==0))  
{  
    cligno_d=0;  
}  
if((Warning==0)&&(Cligno_G==0))  
{  
    cligno_g=0;  
}  
if(FlagTunnig1==1)  
{  
  
    if(OCR2 == 255)  
    {
```

```
        OCR2=0;
    }
    if(OCR0 == 255)
    {
        OCR0=0;
        OCR2=255;
    }
    if(croisement == 1)
    {
        croisement = 0;
        OCR0=255;
    }
    if(feux_avant == 1)
    {
        feux_avant = 0;
        croisement = 1;
    }
    if(feux_arriere == 1)
    {
        feux_arriere = 0;
        feux_avant = 1;
    }
    if(feux_recul == 1)
    {
        feux_recul = 0;
        feux_arriere = 1;
    }
}
```

```

        if((feux_recul == 0)&&(feux_arriere == 0)&&(feux_avant == 0)&&(croisement
== 0)&&(OCR0 == 0)&&(OCR0 == 0))
        {
            feux_recul = 1;

        }

    }
    if(FlagTunnig2==1)
    {
        if(OCR2 == 0)
        {
            OCR2=255;
        }
        if(OCR0 == 0)
        {
            OCR0=255;
            OCR2=0;
        }
        if(croisement == 0)
        {
            croisement = 1;
            OCR0=0;
        }
        if(feux_avant == 0)
        {
            feux_avant = 1;
            croisement = 0;
        }
    }

```



```

    }
    if(feux_arriere == 0)
    {
        feux_arriere = 1;
        feux_avant = 0;
    }
    if(feux_recul == 0)
    {
        feux_recul = 1;
        feux_arriere = 0;
    }
    if((feux_recul == 1)&&(feux_arriere == 1)&&(feux_avant == 1)&&(croisement
== 1)&&(OCR0 == 255)&&(OCR0 == 255))
    {
        feux_recul = 0;
    }
}
}
}

```

//variables globales

//Déclaration des Sorties

//Programme principale

void main(void)

{

```
//Initialisation du port A
PORTA=0x00;
DDRA=0x00;
//Initialisation du port B
PORTB=0x00;
DDRB=0x08;
//Initialisation du port C
PORTC=0x00;
DDRC=0x00;
//Initialisation du port D
PORTD=0x00;
DDRD=0xFF;
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 16000,000 kHz
// Mode: Phase correct PWM top=FFh
// OC0 output: Non-Inverted PWM
TCCR0=0x61;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 3,906 kHz
// Mode: CTC top=OCR1A
// OC1A output: Discon.
// OC1B output: Discon.
```

```
// Noise Canceler: Off

// Input Capture on Falling Edge
TCCR1A=0x00;
TCCR1B=0x0D;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x0f;
OCR1AL=0x3c;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: 16000,000 kHz
// Mode: Phase correct PWM top=FFh
// OC2 output: Non-Inverted PWM
ASSR=0x00;
TCCR2=0x61;
TCNT2=0x00;
OCR2=0x00;

MCUCR=0x00;
MCUCSR=0x00;

ACSR=0x80;
```

```

SFIOR=0x00;

// ADC initialization
// ADC Clock frequency: 125,000 kHz
// ADC Voltage Reference: AREF pin
// ADC High Speed Mode: Off
// ADC Auto Trigger Source: None
ADMUX=ADC_VREF_TYPE;
ADCSRA=0x87;
SFIOR&=0xEF;

// Global enable interrupts
#asm("sei")

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x10;
#asm("sei")
while (1)
{

    if(BP_eclairage==0) //Mettre ON
    {

        if(BP_warning==0)
        {

```

```
Cligno_G=0;
    Cligno_D=0;
    Warning=1;

}
else
{
    Warning=0;

    if(BP_cligno_g==0)
    {
        Cligno_G=1;
    }
    else
    {
        Cligno_G=0;
    }
    if(BP_cligno_d==0)
    {
        Cligno_D=1;
    }
    else
    {
        Cligno_D=0;
    }
}

if(BP_mode) //MODE MANUEL
```

```
{  
    if(BP_feux_route==0)  
    {  
        OCR2=255;  
    }  
    else  
    {  
        OCR2=0;  
    }  
  
    if(BP_feux_crois==0)  
    {  
        croisement =1;  
    }  
    else  
    {  
        croisement =0;  
    }  
  
    if((BP_feux_route==0)|| (BP_feux_crois==0))  
    {  
        feux_arriere =1;  
    }  
  
    if((BP_feux_route==1)&&(BP_feux_crois==1))  
    {
```

```

        feux_arriere =0;
    }
    FlagTunnig1=0;
    FlagTunnig2=0;
}
else
{
    if(BP_feux_route==0)
    {

        FlagTunnig1=0;
    }
    if(BP_feux_crois==0)
    {

        FlagTunnig2=0;
    }
    if((BP_feux_route==1)&&(BP_feux_crois==1))
    {

        FlagTunnig1=0;
        FlagTunnig2=0;
        Moyenne=Moyenne+read_adc(3);
        N++;
        if(N>=100)
        {
            OCR0=(Moyenne/100)/2.15;
            Moyenne=0;

```

```

        N=0;
    }
    feux_recul =0;
    feux_arriere =0;
    feux_avant =0;
    croisement =0;
    luminosite=luminosite+read_adc(2);
    J++;
    if(J>=100)
    {
        luminosite=(luminosite/100);
        OCR2=(((luminosite-70)/2) / (10*exp(-(luminosite-70)/4)+1));
        luminosite=0;
        J=0;
    }
}

}
else
{

    feux_recul =0;
    feux_arriere =0;
    feux_avant =0;
    cligno_g =0;

```



```
cligno_d =0;  
OCR0=0;  
OCR2=0;  
croisement =0;
```

```
}
```

```
}
```

```
}
```