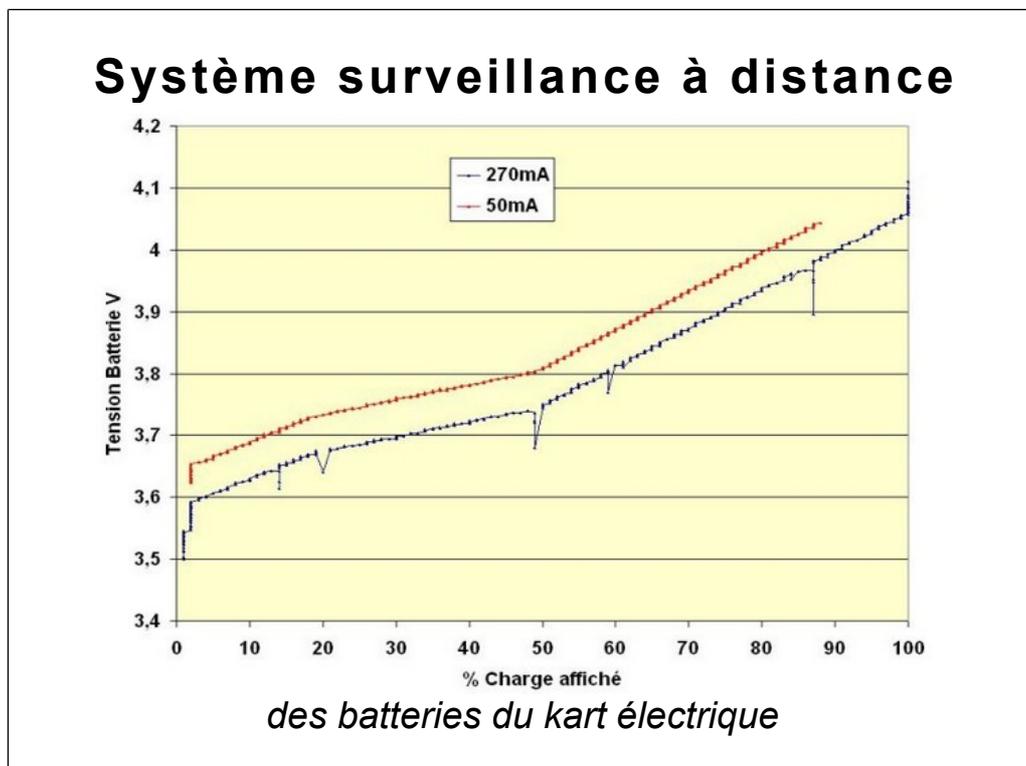


Rapport du projet  
Études & Réalisations  
Semestre 3



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



# **Systeme surveillance à distance**

*des batteries du kart électrique*

IBRAHIM Mohd Khairul Azlan  
2è année – Q1  
2012-2013

Enseignants  
GLIKSOHN Charles  
LEQUE Thierry

# Sommaire

Introduction.....	4
1. Le kart électrique et le système surveillance.....	5
1.1. Le kart électrique.....	5
1.1.1. Les batteries d'un kart électrique.....	6
1.1.2. Le autres parts.....	6
1.2. Le système surveillance .....	7
2. Cahier des charges .....	8
2.1. Introduction.....	8
2.2. Principe.....	8
2.3. Solution.....	8
2.4. Plan Prévisionnel.....	9
3. Réalisation du projet.....	10
3.1. La carte électronique.....	11
3.1.1. Alimentation.....	13
3.1.2. Mesures des tensions.....	15
3.1.3. Transmission des données.....	18
3.2. Le programmation.....	18
3.2.1. Le programmation de l'ATmega8535L.....	19
3.2.2. Logiciel.....	19
Conclusion.....	20
Résumé.....	21
Index des illustrations.....	22
Bibliographie.....	23
ANNEXES.....	24

## Introduction

Dans le cadre ma formation en 2<sup>e</sup> année, un projet d'études doit être réalisé. J'ai choisi un projet sur le kart électrique dont le titre est « Système surveillance à distance ».

Ce projet a été proposé par M. LEQUE Thierry pour répondre à un besoin de son ami dans le circuit du kart électrique. Le projet a été commencé par trois anciens étudiants de GEII. Puisque je suis intéressé par le véhicule électrique, ce projet sera un bon pas pour comprendre son fonctionnement.

L'objectif de ce projet est d'être capable de récupérer des données numériques d'un kart électrique sans celui ci s'arrête. Dans un premier temps, seul la valeur de la tension des batteries du kart sera étudiée. Ces valeurs seront tout d'abord mesurées à l'aide d'un micro-contrôleur et ensuite seront affichées sur ordinateur à l'aide à une transmission par Wi-Fi.

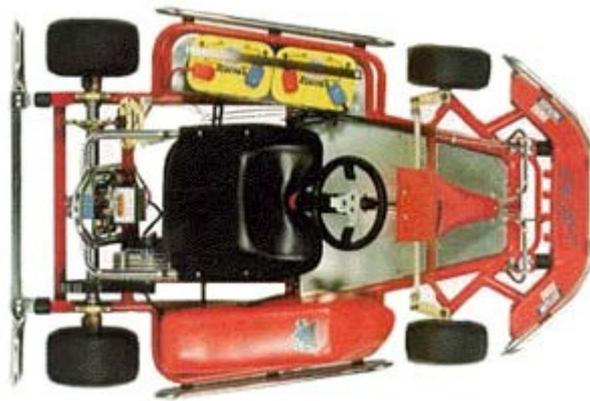
Premièrement, vous trouverez dans ce rapport l'introduction général du kart électrique et le système surveillance pour bien comprendre le but de mon projet. Ensuite, la présentation du cahier des charges. Enfin, je vous expliquerai comment mon projet est fait en faisant apparaître deux grandes parties : la création de la carte électronique et le programmation.

# 1. Le kart électrique et le système surveillance

Avant d'aller plus loin, il sera mieux de se cultiver un peu sur les technologies qui seront utilisés dans le projet.

## 1.1. Le kart électrique

Partout dans le monde, le kart électrique ou go-kart est un sport vraiment connu. Il est moins célèbre que le sport Formule 1<sup>1</sup>, mais le concept reste le même.



*Illustration 1: Image d'un kart électrique.[1]*



*Illustration 2: Le piste à Joué-les-Tours (37).[2]*

Depuis 2003, le département Génie Électrique et Informatique Industrielle, de l'Institut Universitaire de Technologie de Tours ajoute dans son cursus scolaire une étude sur le kart électrique[3]. Ceci est dans le but de réaliser un convertisseur d'énergie pour les étudiants de 2<sup>ème</sup> année dans le cadre des projets Études et Réalisations, l'option Électrotechnique et Électronique de Puissance (EEP).

---

<sup>1</sup> La Formule 1, communément abrégée en F1, est une discipline de sport automobile.

### 1.1.1. Les batteries d'un kart électrique

Un kart électrique utilise en général un moteur à courant continu, qui est alimenté par une source d'énergie électrique. Il possède dans la plupart des cas 4 batteries de 12 volts.

Modèle	660 A	765 A	765 A - US	975 A
Tension nominale	12 V	12 V	12 V	12 V
Démarrage à -18°C	660 A	765 A	765 A	975 A
Capacité de réserve à 25 A	98 mn	124 mn	124 mn	155 mn
Longueur (mm)	228	254	254	324
Largeur (mm)	170	172	172	165
Hauteur (mm)	195	199	199	238
Poids (kg)	16,7	19,5	19,9	27,1
Référence	233157	227646	227647	227653

Randoequipement.com



Illustration 3: BATTERIE OPTIMA JAUNE EN 12V - 75 Ah YTS 5.5 [0]

### 1.1.2. Le autres parts

Le karting ne cesse d'évoluer. Il existe de plus en plus d'éléments afin de créer un kart plus puissante et donc plus efficace. Certains chercheurs ajoutent les composants informatiques pour surveiller les états de chaque fonction.

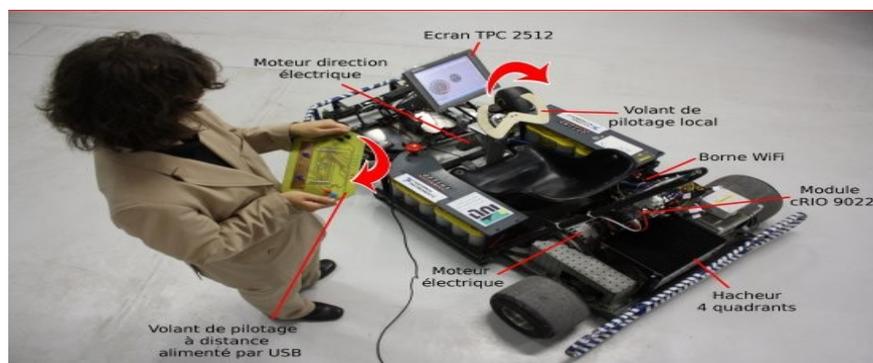


Illustration 4: Les différents divisons d'un kart électrique. [0]

## 1.2. Le système surveillance

En général, un système surveillance est conçu pour la raison de la sécurité. En France, notamment à Paris, les policiers ont implanté beaucoup de systèmes surveillance cameras pour surveiller les publics afin de réduire des nombres de crimes[4].

Dans un système embarqué<sup>2</sup>, il permet à utilisateur de suivre les états d'un élément souhaité.



*Illustration 5: Une équipe de vente DOMOBEC est en train de tester son CCTV. [0]*

---

2 Un système électronique et informatique autonome, souvent temps réel, spécialisé dans une tâche bien précise.

## **2. Cahier des charges**

Au cours de la première séance, et après avoir pris connaissance le sujet, j'ai commencé à rédiger le cahier des charges pour le projet. La connaissance sur le cahier des charges permettra à bien comprendre le problème posé.

### **2.1. Introduction**

Le problème global pour une équipe d'un kart électrique est de connaître l'état des quatre batteries qui alimentent le kart. De plus, l'équipe souhaitera décider pendant que le kart fait son cours, si les batteries doivent être remplacées ou non. Pour eux, la décision prise évitera la vérification inutile sur les batteries et aidera à réduire le temps pour le contrôle technique, qui ensuite permettra au conducteur de rester longtemps dans le circuit.

### **2.2. Principe**

Dans le but de répondre à la problématique, le système doit assurer les fonctionnalités suivantes :

- Mesures de la tension des batteries sur le kart
- Afficher les tensions des batteries sur l'ordinateur

### **2.3. Solution**

Un micro-contrôleur est un circuit intégré qui est capable de faire plein de travaux compliqués comme le calcul et le transmission des données. Il existe plusieurs type de micro-contrôleur, alors il faut bien le choisir afin de ne pas en acheter un qui soit trop cher et puissance, alors que l'on n' utilisera que des fonctionnalités de bases.

Je vais créer donc une carte électronique qui possède un micro-contrôleur. Il se comportera comme le cerveau du mon projet.

Puisque l'intérêt du projet est de créer un système surveillance à distance, je décide d'utiliser le transmission sans fil. Je pourrais soit utiliser une transmission FM qui est classique, soit une transmission par Wi-fi. Sachant qu'un ordinateur portable est équipé déjà avec une connexion Wi-fi, ce sera mieux d'en profiter.

Sur un ordinateur, il faudra que je crée un logiciel qui permette de recevoir les données numériques envoyer par le micro-contrôleur.

Ma connaissance sur l'électronique et l'informatique industrielle (programmation) me permettra d'atteindre le but et de valider la solution proposé.

## 2.4. Plan Prévisionnel

Ce plan a été défini lors de la première séance. Il ne doit pas forcément être respecté, mais cela est un bon moyen pour bien évaluer les démarches et les étapes pour atteindre le but du projet

Séance	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Semaine	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	1	2	3
Choix et découverte du sujet	■	■							■								■	■		
Rédaction du cahier des charges	■	■							■								■	■		
Étude théorique	■		■	■	■				■								■	■		
Réalisation des cartes	■					■	■	■	■	■							■	■		
Programmation	■								■		■	■	■				■	■		
Assemblage et Test	■								■						■	■		■	■	
Test-Prototype	■								■								■	■		
Remise du cahier de projet	■								■								■	■		

■ Planning Prévisionnel  
■ Planning Réel qui sera rempli plus tard

Illustration 6: Le planning prévisionnel qui a été fait en 1ère séance

### 3. Réalisation du projet

Après avoir bien compris le cahier des charges défini dans la première séance, je conclus que le projet n'a besoin qu'une seule carte électronique. Cette carte sera intégrée dans un kart électrique pour mesurer et récupérer les tensions des batteries.

Le micro-contrôleur sera ATmega8535L (*voir annexe 3*) et le module Wi-fi sera un EZL80C (*voir annexe 4*).

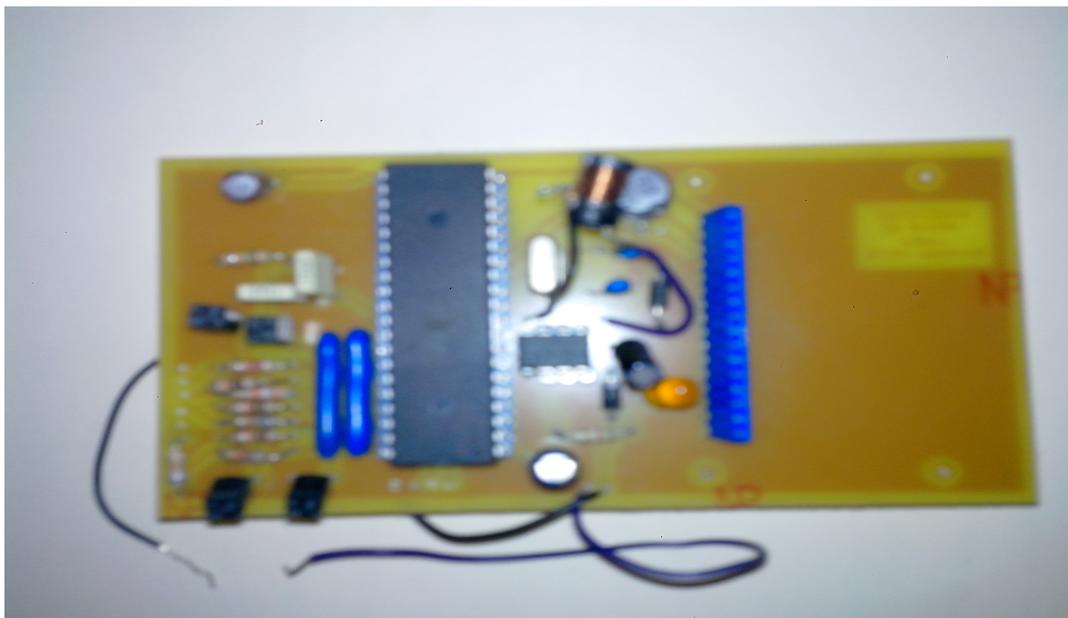
Leur valeur numérique sera ensuite transmis à l'ordinateur en Wi-Fi. Il faudra donc créer un logiciel qui sera capable de traiter cette information.

Dans ce chapitre, vous trouverez donc deux grandes sous-parties : la carte électronique et le programmation.



*Illustration 7: Structure générale du projet réalisé.[0]*

### 3.1. La carte électronique



*Illustration 8: Carte réalisée au cours du projet*

La carte électronique est divisée en trois éléments essentiels suivants : l'alimentation, les mesures des tensions et la transmission des données.

(un bloc de fonctionnement)

La réalisation du typon de la carte sera fait par le logiciel Kicad<sup>3</sup> (**voir annexe 5**). Il existe beaucoup d'autre logiciels qui sont meilleurs, mais à l'IUT on se limite à ce logiciel gratuit. Vous trouvez ci-après le typon<sup>4</sup> que j'ai crée avec le logiciel Kicad (**voir *Illustration 9***).

---

3 Le logiciel Kicad est un suite open source pour la réalisation de schémas électroniques et de circuits imprimés sous licence GPL.

4 Typon ou PcB

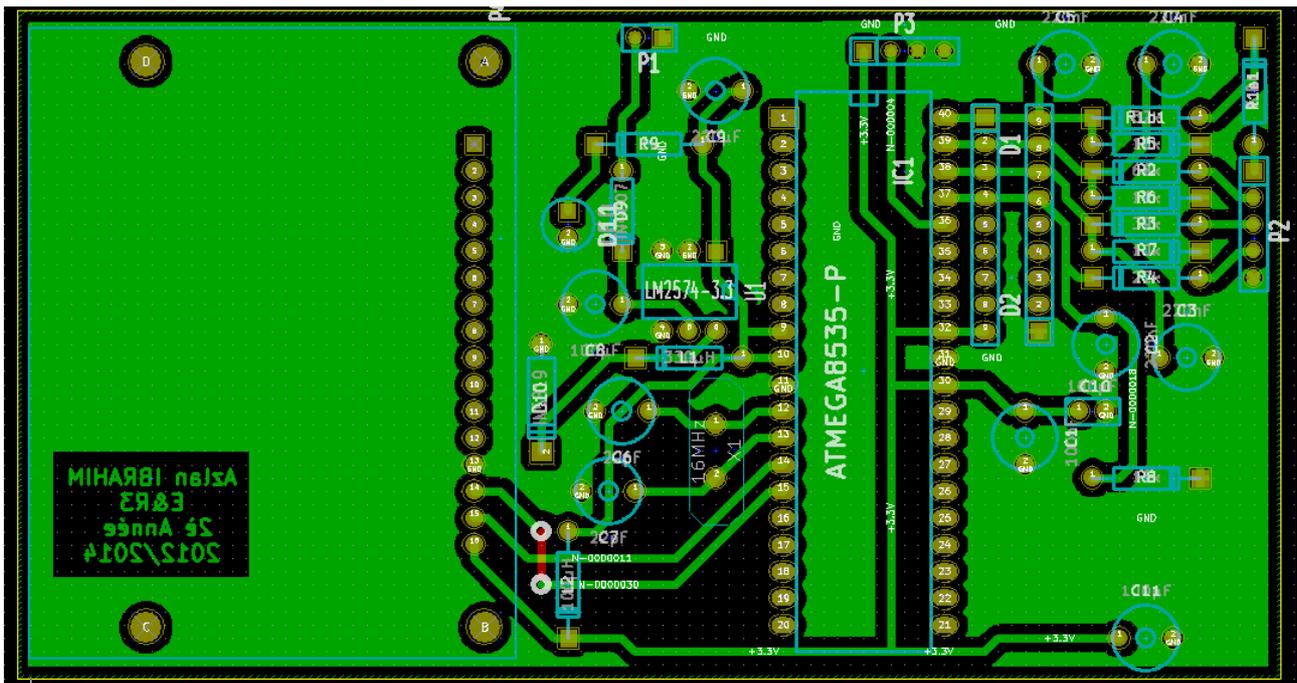


Illustration 9: Le typon de la carte réalisé en utilisant le Kicad

La source d' alimentation sera fournie par une alimentation à découpage<sup>5</sup>. L'explication plus détaillé sur l'alimentation sera trouvé plus tard dans la sous-partie intitulé « alimentation ».

On se rappelle que l'objectif de ce projet est de pouvoir mesurer les tensions aux bornes des batteries du kart. Je vais créer donc une sous-partie appelée « les mesures des tensions » pour expliquer comment elle travaille.

Les tensions mesurées seront ensuite transmis pour être reçues par un afficheur, dans mon cas j'utiliserai l'ordinateur parce qu'il possède déjà un connecteur Wi-Fi.

Dans cette partie, il y a donc trois sous-parties : l'alimentation, les mesures des tensions et la transmission des données.

<sup>5</sup> Une alimentation électrique dont la régulation est assurée par des composants électrotechnique utilisés en commutation (généralement des transistors).

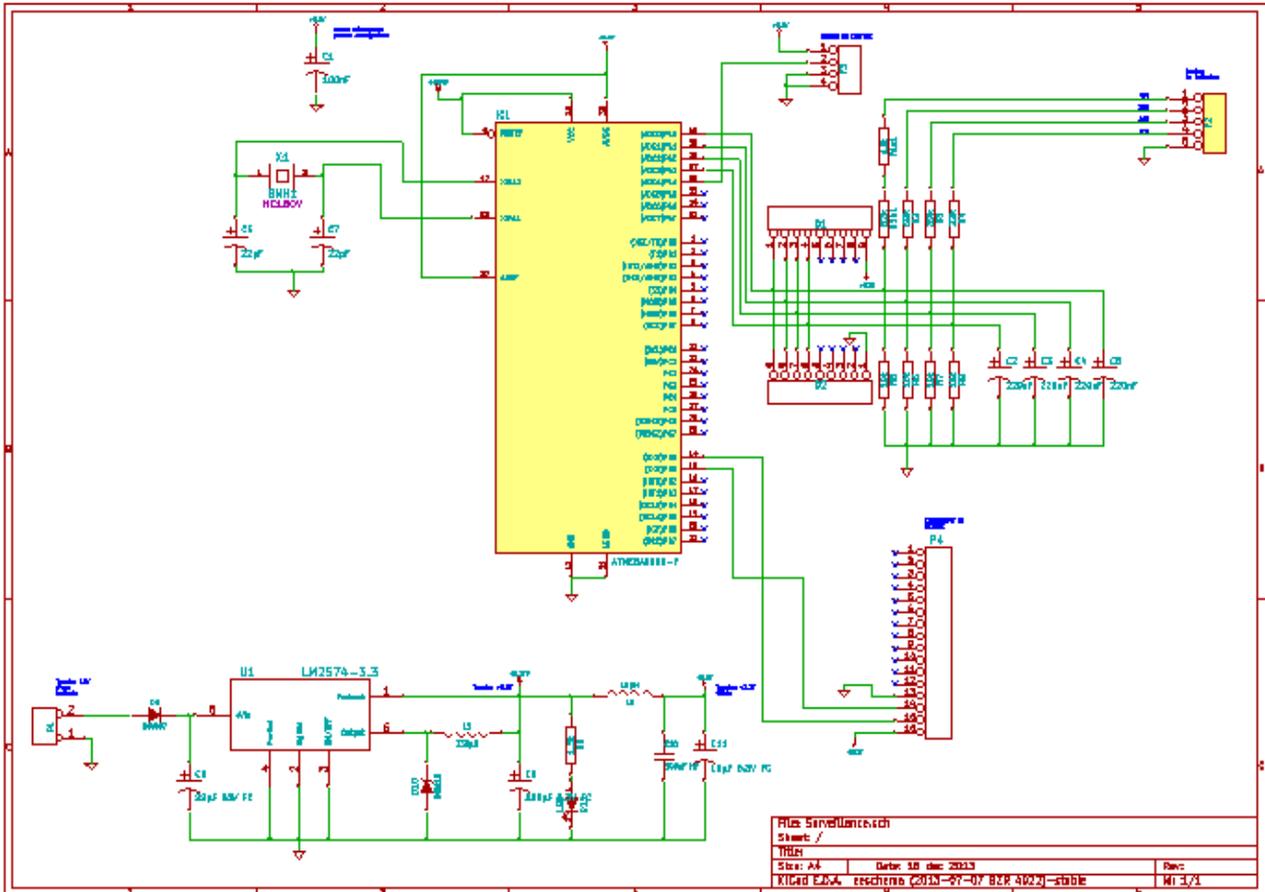


Illustration 10: Schéma complet du montage réalisé

### 3.1.1. Alimentation

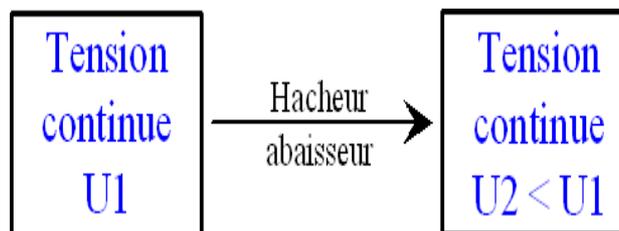


Illustration 11: Hacheur série de type BUCK.

Puisque la carte sera intégrée sur le kart, la seule source d'alimentation disponible sera une des quatre batteries de 12V. La carte électronique comprend d'un micro-contrôleur d'ATmega8535L et d'un module Wi-Fi EZL80C qui ne se mettent à conduire<sup>6</sup> qu'avec une tension de 3,3V. Il est impératif d'avoir la tension de 3,3V pour que la carte fonctionne correctement. En partant de 12V, l'alimentation découpage de type Buck<sup>7</sup>

<sup>6</sup> Se mettre à conduire est un terme électronique pour dire que le composant fonctionne avec une condition

<sup>7</sup> Appelé aussi un hacheur série ou convertisseur continu-continu.

abaisse cette tension à la valeur 3,3V souhaitée. Ici, la tension  $U_1$  est 12 V qui sera abaissé à  $U_2=3,3V$  (**voir Illustration 11**) .

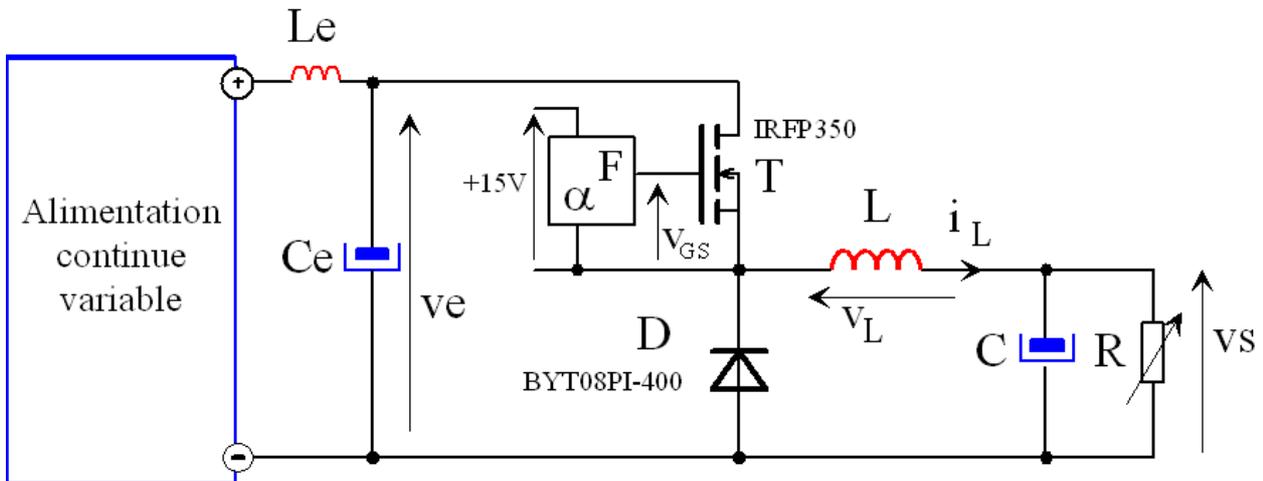


Illustration 12: Structure générale du hacheur série de type BUCK.

Le hacheur série ou Buck (**voir Illustration 12**) est constitué d'un LM2574 -3,3V (**voir Illustration 13 et annexe 6**) qui est capable de convertir une tension d'entrée maximum 60V à 3,3V.

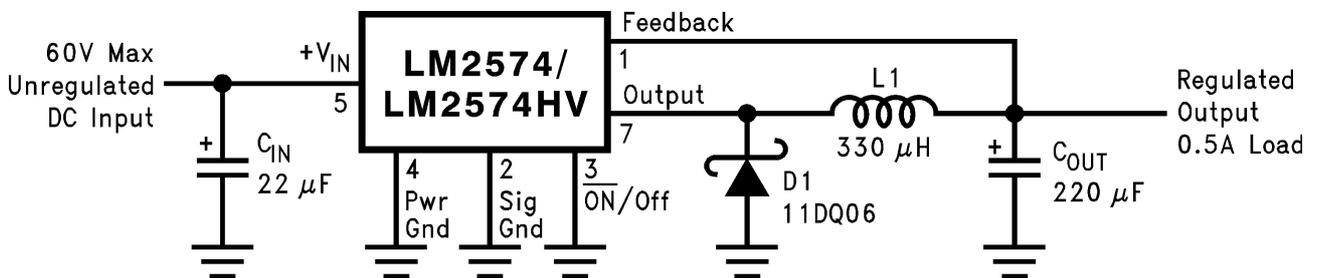


Illustration 13: Structure général du LM2574

Le schéma final du montage pour la partie alimentation est :

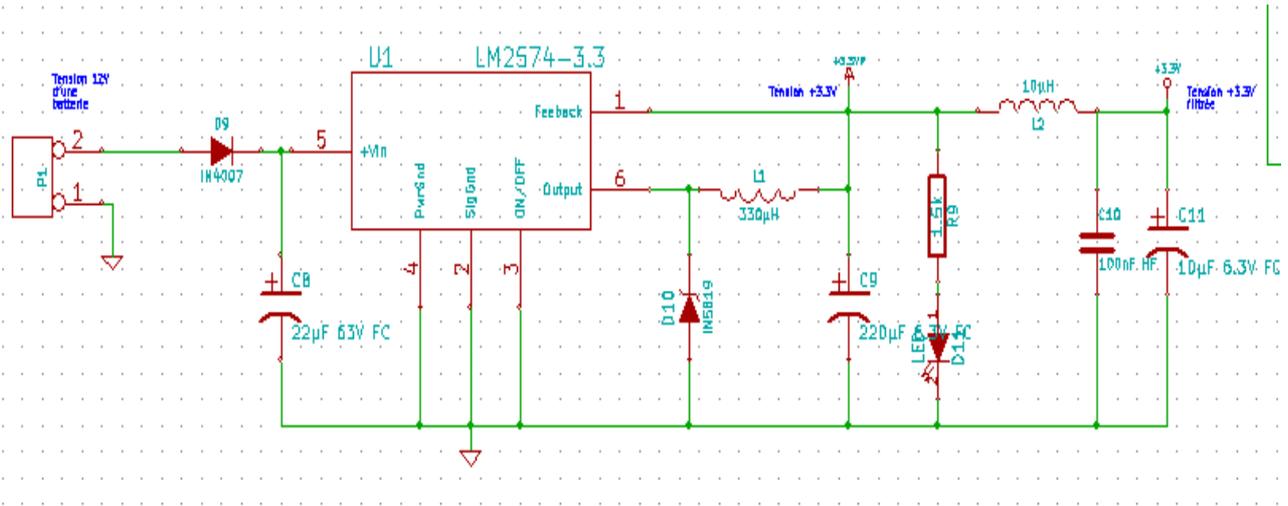


Illustration 14: Schéma d'alimentation 3,3V de l'ATmega8535L.

### 3.1.2. Mesures des tensions

Les batteries sont branchées en série ce qui ne permet pas donc de mesurer directement la tension au borne de chacune d'entre elle. Un kart possède quatre batterie, il faudra alors utiliser une opération mathématique pour déduire leur tension. Cette opération sera effectué par l'ATmega8535L que j'expliquerai plus tard.

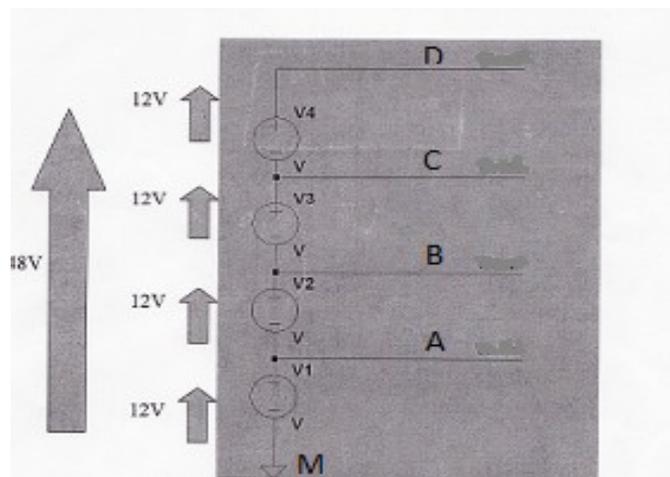


Illustration 15: Schéma des batteries en série

Puisqu'elles sont en série, la tension des différents points est la somme de chaque tension entre le point souhaité et la référence, ici ce sera la masse M. Par exemple, pour le point A, la tension  $V_{AM}$  vaudra la tension de la première batterie :

$$V_{AM} = V_{bat1} = 12V \text{ maxi}$$

Ensuite, les tensions  $V_{BM}$ ,  $V_{CM}$  et  $V_{DM}$  seront :

$$V_{BM} = V_{bat1} + V_{bat2} = 24V \text{ maxi}$$

$$V_{CM} = V_{bat1} + V_{bat2} + V_{bat3} = 36V \text{ maxi}$$

$$V_{DM} = V_{bat1} + V_{bat2} + V_{bat3} + V_{bat4} = 48V \text{ maxi}$$

Pour chaque de la tension des batteries, la variation de leur valeur sera traitée par l'ATmega8535L. Ses entrées ne peuvent dépasser 5 volts, ce qui veut dire qu'il traite une variation de la tension entre 0 jusqu'à 5 volts. Il faut donc effectuer un pont de diviseur de tension<sup>8</sup> après les mesures des tensions des quatre batteries (**voir illustration 16**).

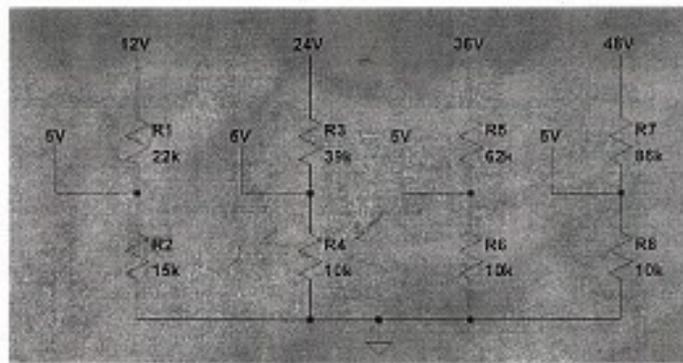


Illustration 16: Pont diviseur de tension

$$5V = \frac{R2 * 12V}{R1 + R2} \quad \text{Donc} \quad R1 = \frac{R2(12V - 5V)}{5V} \quad \text{A.N :} \quad R1 = \frac{15 \cdot 10^3 (12V - 5V)}{5V} = 21k\Omega$$

$$5V = \frac{R4 * 24V}{R3 + R4} \quad \text{Donc} \quad R3 = \frac{R4(24V - 5V)}{5V} \quad \text{A.N :} \quad R3 = \frac{10 \cdot 10^3 (24V - 5V)}{5V} = 38k\Omega$$

$$5V = \frac{R6 * 36V}{R5 + R6} \quad \text{Donc} \quad R5 = \frac{R6(36V - 5V)}{5V} \quad \text{A.N :} \quad R5 = \frac{10 \cdot 10^3 (36V - 5V)}{5V} = 62k\Omega$$

$$5V = \frac{R8 * 48V}{R7 + R8} \quad \text{Donc} \quad R7 = \frac{R8(48V - 5V)}{5V} \quad \text{A.N :} \quad R7 = \frac{10 \cdot 10^3 (48V - 5V)}{5V} = 86k\Omega$$

Les constructeurs des résistances ne fournissent que certaines valeurs. Il faut donc normaliser les valeurs calculées afin de pouvoir les utiliser. En général, la valeur normalisée est un peu au dessus de la valeur calculée.

<sup>8</sup> Le diviseur de tension est un montage électronique simple qui permet de diviser une tension d'entrée

R1 → 22kΩ

R3 → 39kΩ

R5 → 68kΩ

R7 → (82kΩ + 6,8kΩ) en série

En faisant ce calcul, on a alors une tension d'entrée du micro-contrôleur ATmega8535L comprise entre 0V et 5V.

Après le diviseur des tensions, je mets un pont de diodes pour protéger le micro-contrôleur au cas où la tension d'entrée qui arrive dépasse 5V. Il existe également des condensateurs pour la même raison.

Le schéma ci-après permet de bien voir la partie mesures de tension des quatre batteries.

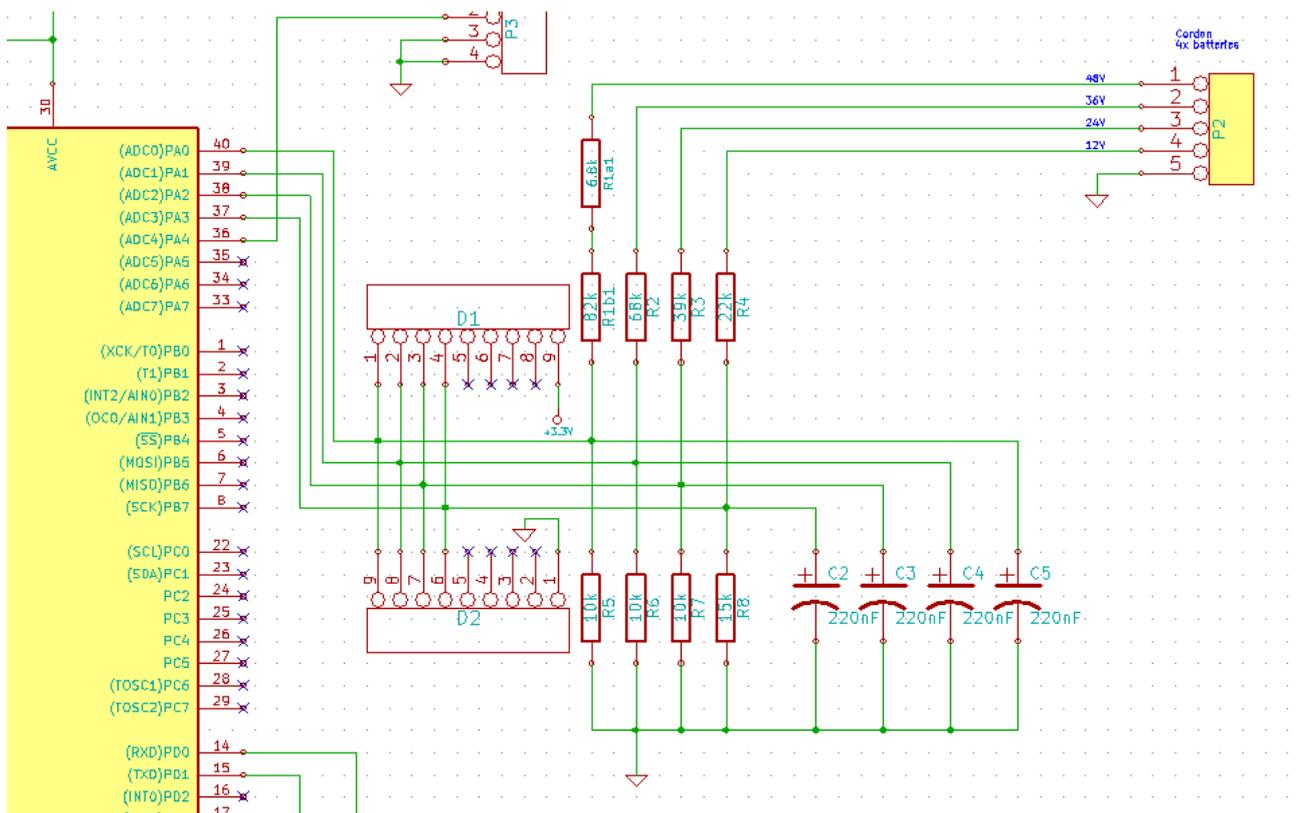


Illustration 17: Schéma des mesures des tension de 4 batteries

### 3.1.3. Transmission des données

Le modelé Wi-fi utilisé sera l'EZL80C. Ce modèle utilisé n'est plus produit par le constructeur. M LEQUEU se rappelle qu'il en gardait quelques uns. Pour pouvoir transmettre les trames d'information, il faudra programmer l'Atmega8535L et le port d'entrée et de sortie de l'EZL80C qui sera fait dans la partie « Programmation ».

Sur l'EZL80C, quatre bornes seront utilisées : l'entrée d'alimentation de 3,3V, la borne du réception RxD, la borne du transmission TxD et la masse (GND).

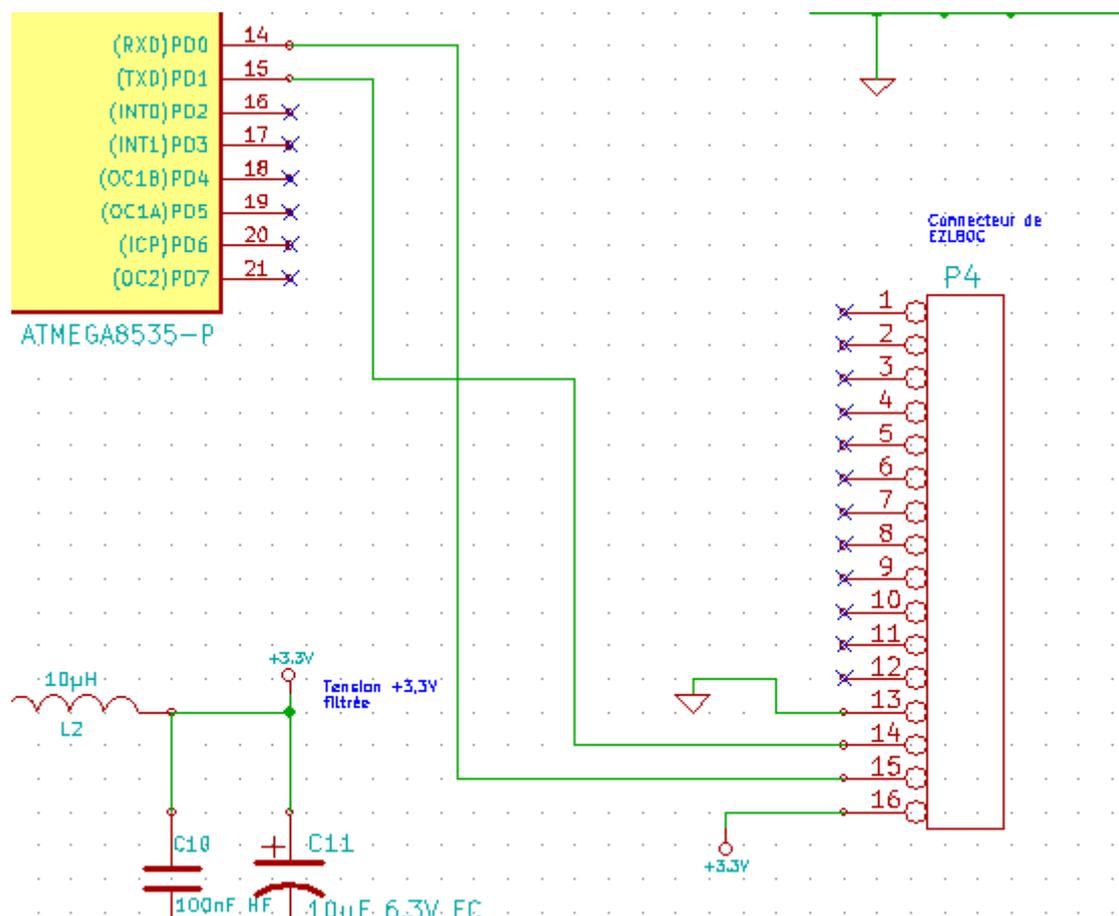


Illustration 18: Schéma de transmission par EZL80C

## 3.2. Le programmation

Au moment où ce rapport a été écrit, je n'avais pas encore commencé cette partie. Pourtant, vous trouverez les informations essentielles.

### **3.2.1. Le programmation de l'ATmega8535L**

Il y a deux programmes qu'il faudra écrire dans l'ATmega8535L pour pouvoir le faire fonctionner correctement. Dans un premier temps, ce sera la partie mesure des tensions, qui est le C.A.N<sup>9</sup>. En effet, les batteries fournissent des valeurs analogiques qui ne peuvent être traité par ordinateur. Ensuite, il faudra coder la partie transmission l'USART<sup>10</sup> pour faire le lien avec l'EZL80C.

Le logiciel utilisé est CodeVisonAVR qui pourra être écrit en langage C. Les deux codes sont déjà faite par M. LEQUE. Il faudra juste changer quelques variables et supprimer quelques lignes inutiles.

Vous trouverez dans *l'annexe 1* le programmation de test de C.A.N et dans *l'annexe 2* le programmation de test de l'USART.

### **3.2.2. Logiciel**

À l'aide de logiciel C++ Builder qui est disponible à l'IUT GEII, un logiciel qui est capable de traiter les informations envoyées et d'afficher les valeurs des tensions en fonction de temps. Cette partie n'a pas été vraiment commencée car en ce moment, je viens juste d'apprendre à l'exploiter.

---

9 Convertisseur analogique numerique

10 Universal Synchronous/Asynchronous Receiver/Transmitter

## **Conclusion**

L'étude sur la solution proposé peut laisser supposera que le système fonctionnera. Puisque je n'ai pas vraiment de connaissance sur la fonctionnalité du micro-contrôleur, j'ai dépense énormément temps sur lui.

Je souhaite toujours pouvoir tester mon système après l'écriture de ce rapport. Le logiciel (programme) sur l'ordinateur sera fait pendant les vacances de Noël.

## Résumé

Au cours de la réalisation du projet, j'ai rencontré beaucoup de difficulté notamment dans la partie compréhension du catalogue du constructeur. La plupart utilisent des symboles bien différent qu'en Europe que je dois se méfier.

De plus, pour trouver un composant qui correspond bien à mon souhaite, il faudra faire une recherche sur le site en ligne qui quelques fois n'est pas facile.

63 mots

## Index des illustrations

Illustration 1: Image d'un kart électrique.[1].....	5
Illustration 2: Le piste à Joué-les-Tours (37).[2].....	5
Illustration 3: BATTERIE OPTIMA JAUNE EN 12V - 75 Ah YTS 5.5 [0].....	6
Illustration 4: Les différents divisons d'un kart électrique. [0] .....	6
Illustration 5: Une équipe de vente DOMOBEC est en train de tester son CCTV. [0].....	7
Illustration 6: Le planning prévisionnel qui a été fait en 1ère séance.....	9
Illustration 7: Structure général le projet réalisé.[0].....	10
Illustration 8: Carte réalisée au cours du projet.....	11
Illustration 9: Le typon de la carte réalisé en utilisant le Kicad.....	12
Illustration 10: Schéma complet du montage réalisé.....	13
Illustration 11: Hacheur série de type BUCK.....	13
Illustration 12: Structure générale du hacheur série de type BUCK.....	14
Illustration 13: Structure général du LM2574.....	14
Illustration 14: Schéma d'alimentation 3,3V de l'ATmega8535L.....	15
Illustration 15: Schéma des batteries en série.....	15
Illustration 16: Pont diviseur de tension.....	16
Illustration 17: Schéma des mesures des tension de 4 batteries.....	17
Illustration 18: Schéma de transmission par EZL80C.....	18

## Bibliographie

- [1] , "", [En ligne]. <[http://www.kartelec.com/f/fr\\_tech.htm](http://www.kartelec.com/f/fr_tech.htm)> (Page consultée le ).
- [2] , "", [En ligne]. <<http://www.pole-karting.com/fra/visite.htm>> (Page consultée le ).
- [3] **Jean Luthier**, "*Historique du kart électrique au sein de l'I.U.T GEII de Tours*", [En ligne]. <[http://www.e-kart.fr/2006/4\\_presentation/presentation\\_historique.html](http://www.e-kart.fr/2006/4_presentation/presentation_historique.html)> (Page consultée le ).
- [0] , "", [En ligne]. <<http://www.randoequipement.com/accessoires/batterie-optima-jaune-en-12v-75-ah-yts-5.5-demarrage-4x4-camping-car-bateaux.htm>> (Page consultée le ).
- [0] , "", [En ligne]. <<http://sine.ni.com/cs/app/doc/p/id/cs-13193>> (Page consultée le ).
- [4] , "*Debate swirls as Paris embraces video surveillance*", [En ligne]. <<http://www.france24.com/en/20120117-debate-swirls-around-paris-new-high-surveillance-system-cameras-cctv-police/>> (Page consultée le ).
- [0] , "*Système de surveillance vidéo - CCTV et IP*", [En ligne]. <<http://domobec.com/category/services/video-surveillance-systems-cctv-and-ip>> (Page consultée le ).
- [0] , "*PCI Card turns PC into surveillance monitoring system.*", [En ligne]. <<http://news.thomasnet.com/news/computer-hardware-peripherals/i-o-interface-cards-adapters/pmc-interface-cards/120>> (Page consultée le ).

## **ANNEXES**

# Annexe 1 : Programmation de test du C.A.N.

```
/******  
This program was produced by the  
CodeWizardAVR V1.25.3 Evaluation  
Automatic Program Generator  
© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.  
http://www.hpinfotech.com
```

```
Project : Programme de test du C.A.N  
Version :  
Date    : 08/01/2008  
Author  : Thierry LEQUEU  
Company :  
Comments:
```

```
Chip type       : ATmega8535  
Program type    : Application  
Clock frequency : 16,000000 MHz  
Memory model    : Small  
External SRAM size : 0  
Data Stack size : 128  
*****/
```

```
#include <mega8535.h>  
  
// Alphanumeric LCD Module functions  
#asm  
    .equ __lcd_port=0x15 ;PORTC  
#endasm  
#include <lcd.h>  
  
#define ADC_VREF_TYPE 0x00  
  
// Read the AD conversion result  
unsigned int read_adc(unsigned char adc_input)  
{  
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);  
    // Start the AD conversion  
    ADCSRA|=0x40;  
    // Wait for the AD conversion to complete  
    while ((ADCSRA & 0x10)==0);  
    ADCSRA|=0x10;  
    return ADCW;  
}  
  
#include<stdio.h>  
#include<delay.h>  
  
// Declare your global variables here  
  
void main(void)  
{  
    // Declare your local variables here  
    unsigned int i,j;  
    unsigned char tampon[20];  
  
    // 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=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x00;

// Port C 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
PORTC=0x00;
DDRC=0x00;

// Port D 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
PORTD=0x00;
DDRD=0x00;

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

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;

```

```

TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

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

// ADC initialization
// ADC Clock frequency: 1000,000 kHz
// ADC Voltage Reference: AREF pin
// ADC High Speed Mode: Off
// ADC Auto Trigger Source: None
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;
SFIOR&=0xEF;

// LCD module initialization
lcd_init(16);
lcd_gotoxy(0,0);
lcd_putsf("Projet TEST-03");
lcd_gotoxy(0,1);
lcd_putsf("Test du C.A.N.");
lcd_gotoxy(0,2);
lcd_putsf("Thierry LEQUEU");
lcd_gotoxy(0,3);
lcd_putsf("8 janvier 2008");

delay_ms(1000);
lcd_gotoxy(0,0);
lcd_putsf("V0----- V4-----");
lcd_gotoxy(0,1);
lcd_putsf("V1----- V5-----");
lcd_gotoxy(0,2);
lcd_putsf("V2----- V6-----");
lcd_gotoxy(0,3);
lcd_putsf("V3----- V7-----");

while (1)
{
    for (j=0;j<=3;j++) {
        i=read_adc(j);
        sprintf(tampon,"%4d",i);
        lcd_gotoxy(3,j);
        lcd_puts(tampon); //on affiche la valeur de i.

        i=read_adc(j+4);
        sprintf(tampon,"%4d",i);
    }
}

```

```
    lcd_gotoxy(11,j);  
    lcd_puts(tampon); //on affiche la valeur de i.  
};  
delay_ms(10);  
};  
}
```

## Annexe 2 : Programmation de test de l'USART.

```
/******  
This program was produced by the  
CodeWizardAVR V1.25.3 Evaluation  
Automatic Program Generator  
© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.  
http://www.hpinfotech.com  
  
Project : Test des afficheurs borne 50 m  
Version : 3  
Date    : 3 juillet 2007  
Author  : Thierry LEQUEU  
Company :  
Comments:  
  
Chip type      : ATmega8535  
Program type   : Application  
Clock frequency : 16,000000 MHz  
Memory model   : Small  
External SRAM size : 0  
Data Stack size : 128  
*****/  
  
#include <mega8535.h>  
#include <stdio.h>  
  
// Alphanumeric LCD Module functions  
#asm  
    .equ __lcd_port=0x15 ;PORTC  
#endasm  
#include <lcd.h>  
#include <delay.h>  
  
#define BP1 PIND.7  
#define BP2 PIND.6  
#define ENABLE PORTD.4  
  
// Declare your global variables here  
flash const unsigned char adresse_constant[16] =  
{0,8,4,12,2,10,6,14,1,9,5,13,3,11,7,15};  
  
#define DIGIT_A 64; { bit b6 }  
#define DIGIT_B 128; { bit b7 }  
#define DIGIT_C 2; { bit b1 }  
#define DIGIT_D 4; { bit b2 }  
#define DIGIT_E 8; { bit b3 }  
#define DIGIT_F 32; { bit b5 }  
#define DIGIT_G 16; { bit b4 }  
#define DIGIT_P 1; { bit b0 }  
  
flash const unsigned char valeur_constant[] =  
{0xEE,0x82,0xDC,0xD6,0xB2,0x76,0x7E,0xC2,0xFE,0xF6,  
0,1,2,4,8,16,32,64,128};  
  
unsigned char  
tampon[20],seconde,secondel1,temps,temps1,dizaine,unite,dixieme,centieme,test,var  
;
```

```

// Declare your global function here

unsigned char USART_Receive( void )
{
/* Wait for data to be received */
while ( !(UCSRA & 0x80) ) // Test de RXC bit7
;
/* Get and return received data from buffer */
return UDR;
}

// Timer 1 output compare A interrupt service routine
interrupt [TIM1_COMPA] void timer1_compa_isr(void)
{
temps++;
if (temps>=100)
{
temps=0;
seconde++;
if (seconde>=60)
{
seconde=0;
};
};
if (test == 1) {
temps1++;
if (temps1>=100)
{
temps1=0;
secondel++;
if (secondel>=60)
{
secondel=0;
};
};
}
}

// Affiche une valeur sur 8 bits à l'adresse de l'afficheur :
void afficheur1(unsigned char adresse,unsigned char caractere, unsigned char
point)
{
if (point == 1) {
PORTA=(valeur_constant[caractere] | 0x01); // Un caractère avec le point !
}
else {
PORTA=valeur_constant[caractere]; // Un caractère
}

PORTB=(0b00010000 | adresse_constant[adresse & 0x0F]); // PB7 PB6 PB5 CS A0 A1
A2 A3
PORTB.4=1;
PORTB.4=0; // CS = 0
PORTB.4=1;
PORTA=0x00;
}

void main(void)
{
// Declare your local variables here

```

```

// 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; en entrée
DDRA=0xFF; // en sortie

// Port B 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
PORTB=0xFF;
// DDRB=0x00; en entrée
DDRB=0xFF; // en sortie

// Port C 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
PORTC=0x00;
DDRC=0x00;

// Port D 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
PORTD=0x10;
ENABLE=1;
// DDRD=0x00; en entrée
DDRD=0x38; // BP1 BP2 OC1A IN INT1 IN IN IN
           // IN7 IN6 OUT5 OUT4 OUT3 IN2 IN1 IN0

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

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 2000,000 kHz
// Mode: CTC top=OCR1A
// OC1A output: Toggle
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: On
// Compare B Match Interrupt: Off
TCCR1A=0x40;
TCCR1B=0x0A;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x4E; // Base de temps = 2 MHz, soit 0,5 us.
OCR1AL=0x20; // Interruption quand on arrive à 20 000 (0x4E20 soit 10 ms)
OCR1BH=0x00;

```

```

OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x10;

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

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: Off
// USART Mode: Asynchronous
// USART Baud rate: 9600
UCSRA=0x00;
UCSRB=0x10;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x67;

// LCD module initialization
lcd_init(16);

/* switch to writing in Display RAM */
lcd_gotoxy(0,0);
lcd_putsf("Projet TEST-51");
lcd_gotoxy(0,1);
lcd_putsf("Borne 50 m DA");

PORTB=0xFF;
seconde=0;
temps=0;
seconde1=0;
temps1=0;
test=0;
var='0';
ENABLE=0;
afficheurl(0,11,0); // Un caractère
afficheurl(1,11,0); // Un caractère
afficheurl(2,11,0); // Un caractère
afficheurl(3,11,0); // Un caractère

```

```

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

while (1)
{
// Place your code here
PORTD.3=1;

// Reception sur la liaison série :
var=USART_Receive();
if (var == 'G') {
test=1;
}
if (var == 'S') {
test=0;
}

// Test des boutons poussoirs :
if (BP1 == 0) { // BP1 : GO !
delay_ms(20);
if (BP1 == 0) {
test=1;
}
}

if (BP2 == 0) { // BP2 : STOP !
delay_ms(20);
if (BP2 == 0) {
test=0;
}
}
sprintf(tampon,"Temps = %2d:%2d%3d",seconde,temps,test);
lcd_gotoxy(0,2);
lcd_puts(tampon);

sprintf(tampon,"Var = %1c",var);
lcd_gotoxy(0,3);
lcd_puts(tampon);

dizaine = secondel/10;
unite = secondel-dizaine*10;
dixieme = tempsl/10;
centieme = tempsl-dixieme*10;
// Afficheur d'adresse 0
if (dizaine == 0) {
afficheur1(0,13,0);
}
else {
afficheur1(0,dizaine,0); // Caractère des dizaines.
}
// Afficheur d'adresse 1
afficheur1(1,unite,1); // Caractère des unités.
// Afficheur d'adresse 2
afficheur1(2,dixieme,0); // Caractère des dixièmes.
// Afficheur d'adresse 3
afficheur1(3,centieme,0); // Caractère des centièmes.

PORTD.3=0;};}

```

