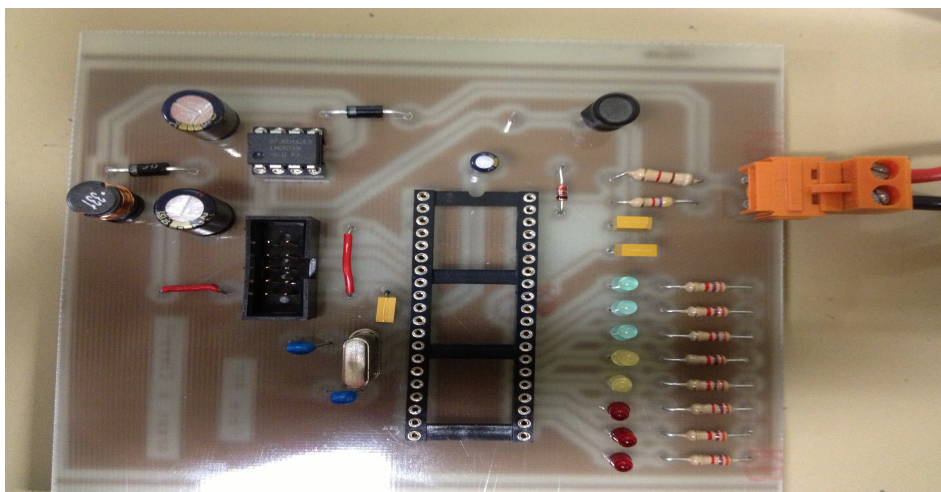


Université François-Rabelais de Tours

Institut Universitaire de Technologie de Tours

Département Génie Électrique et Informatique Industrielle



Vu-mètre à 11 LED pour une batterie 12V

Étude et Réalisation

MEDJAHED Malik
KA4
Promotion 2012/14

Enseignants :
AUGER Philippe
LEQUEU Thierry

Université François-Rabelais de Tours

Institut Universitaire de Technologie de Tours

Département Génie Électrique et Informatique Industrielle



Vu-mètre à 11 LED pour une batterie 12V

Étude et Réalisation

MEDJAHED Malik
KA4
Promotion 2012/14

Enseignants :
AUGER Philippe
LEQUEU Thierry

Table des matières

Remerciement.....	3
Introduction.....	5
1Cahier des charges.....	7
1.1Présentation du projet.....	7
2Analyse fonctionnel du vu-mètre.....	7
2.1Schéma fonctionnel général du projet.....	8
2.2Présentation des différents seuils de tension des diodes.....	8
2.3Schéma fonctionnel détaillé du projet.....	8
3Analyse technique.....	9
3.1 Alimentation à découpage.....	9
3.2 Microcontrôleur.....	10
3.2.1Utilisation des LED.....	11
3.2.2Calcul des résistances.....	11
3.3Diviseur de Tension.....	12
4Partie programmation de l'ATMEGA8535.....	13
4.1 Le programme initial.....	13
4.2La partie commune du programme.....	13
4.3La modification du programme.....	14
4.3.1 Le programme test.....	14
4.3.2 La modification du programme principal.....	15
4.3.3La Conversion Analogique Numérique (C.A.N.).....	16
4.3.4Résultat obtenu.....	18
5Partie réalisation de la nouvelle carte électronique.....	19
5.1Typon, gravure et soudure.....	20
6Planning prévisionnel.....	22
7Conclusion.....	22
8Résumé.....	23
Annexe.....	26

Remerciement

Tout d'abord, je souhaite remercier notre professeur M.LEQUEU qui m'a aidé tout au long de mon projet et à l'aide de son site qui m'a apporté les renseignements complémentaires nécessaires. Nous remercions aussi Mr. Alain le magasinier qui m'a accordé de son temps afin de me donner les composant nécessaire et enfin nous remercions M.AUGER pour avoir porté son attention à lire et à évaluer notre projet.

Introduction

Mon projet est de modifier un projet réalisé auparavant par des étudiants.

Le projet initial est un vu-mètre à 8 LED qui nous permet de visualiser la tension que nous avons sur une batterie de 12V contenu dans un KART.

Mon objectif est de modifier le programme pour contrôler un vu-mètre à 11 LED, de rajouter quelques suppléments pour faciliter son utilisation et de fabriquer une nouvelle carte moins encombrante pour pouvoir l'alimenter avec un câble de gros diamètre

.

Dans ce dossier, vous trouverez les solutions technologiques qui répondent au cahier des charges.

En premier lieu, la partie programmation qui illustre le programme initial et la modification réalisé pour commander le vu-mètre à 11 LED.

Dans un deuxième temps, la partie électronique qui définit la réalisation de la carte électronique qui répond au cahier des charges.

Pour finir, un planning de réalisation de ce projet sera mis en place.

1 Cahier des charges

Sujet : Vu-mètre à 11 LED

1.1 Présentation du projet

Je modifierai un programme informatique contenu dans un circuit intégré ATmega8535 qui commande un vu-mètre à 11 LED pour vérifier la tension de charge d'une batterie 12V. À l'aide de l'ATmega8535, chaque LED s'allumera lorsque un seuil de tension est atteint. La variation de tension varie de 10 à 15 Volt et je réaliserai une carte électronique moins encombrante pour pouvoir alimenter la carte électronique avec le câble d'alimentation préconisé par le professeur Mr LEQUEU en remplaçant l'ATMEGA 8535 par l'ATMEGA 8.

Les composants

- Un micro-contrôleur ATmega8535.
- Adaptateur allume cigare de voiture.
- Alimentation à découpage (Hacheur Série).
- Pont diviseur de tension.
- 11 LED de couleurs

Description du système

- Alimenter le micro-contrôleur.
- Traiter la tension de la batterie.
- Allumer une par une les LED avec le microcontrôleur suivant la tension de la batterie.

Les contraintes

- L'utilisation d'un circuit intégré ATmega8535.
- L'alimentation de l'ATmega8535
- Faire un programme de calcul de moyenne,
- Faire clignoter les LED lors de la mise sous tension du microcontrôleur.
- Réaliser une carte moins encombrante pour le câble d'alimentation

2 Analyse fonctionnel du vu-mètre

2.1 Schéma fonctionnel général du projet



Illustration 1: schéma fonctionnel du vu-mètre

Le vu-mètre qu'ils ont fabriqué va nous permettre de voir si la batterie de notre Kart est chargée. Les LED¹ nous serviront à afficher la tension de la batterie (indicateur de charge). Basé sur le principe d'un tensiomètre avec indicateur à LED

Lorsqu'il y a au moins une LED d'allumée la tension de la batterie sera d'au moins 10V. Le vu-mètre permet de mesurer la valeur entre 10 et 15V. Chaque LED sera allumée à un seuil de tension différent avec un pas de 0,5, lorsque la tension est inférieure à 10V toutes les LED clignotent.

2.2 Présentation des différents seuils de tension des diodes

Nombres de diodes du vu-mètre et les seuils de tension de batterie pour chaque diode.

TOUTES les LED clignotent < 10V

- LED 1 (Verte) 10V
- LED 2 (Verte) 10,5V
- LED 3 (Verte) 11V
- LED 4 (Verte) 11,5V
- LED 5 (Jaune) 12,0V
- LED 6 (Jaune) 12,5V
- LED 7 (Jaune) 13V
- LED 8 (Jaune) 13,5V
- LED 9 (Rouge) 14V
- LED 10 (Rouge) 14,5V
- LED 11 (Rouge) 15V

Chaque LED restera allumée tant que son seuil de tension ne sera pas dépassé. Donc toutes les LED seront allumées lorsque la tension sera de 15V.

¹LED (light-emitting diode), diodes électroluminescentes.

2.3 Schéma fonctionnel détaillé du projet

Ce schéma fonctionnel représente le vu-mètre. Cela regroupe les grandes fonctions du projet qui sont au nombre de trois.

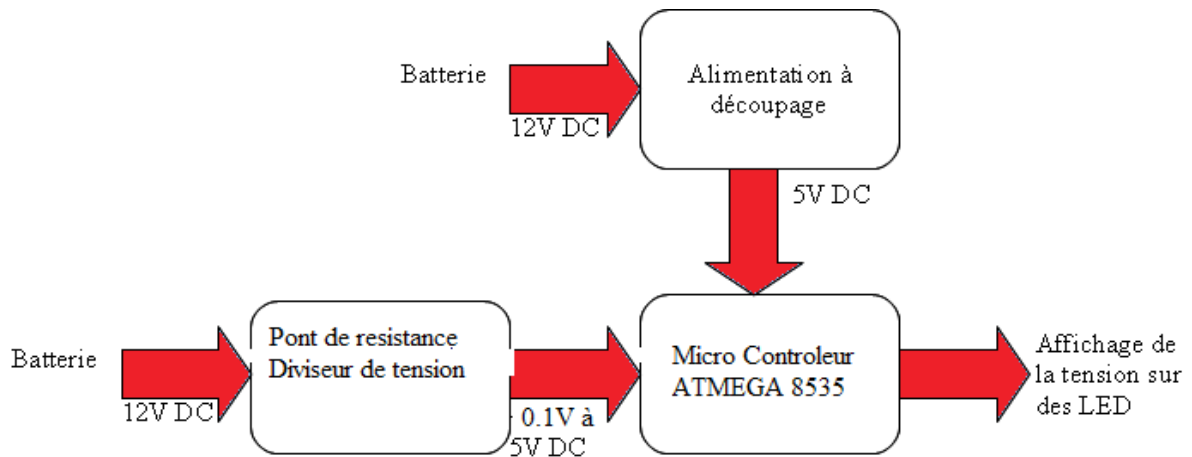


Illustration 2: Schéma fonctionnel détaillé du projet

La batterie sera d'environ 12V DC, elle peut atteindre les 15V. Sa variation de tension de 0V à 15V est due à sa charge ou à sa décharge.

Le pont diviseur de tension va permettre d'alimenter en 5V DC² le microcontrôleur pour que celui-ci puisse mesurer la tension de la batterie. sans pour autant détériorer celui-ci.

Nous pouvons voir que le microcontrôleur va permettre d'alimenter plus ou moins de LED en fonction de la tension mesurée grâce au programme informatique qu'il contient.

2 5V DC signifie 5V courant continu.

3 Analyse technique

3.1 Alimentation à découpage

Pour l'alimentation du microcontrôleur nous allons utiliser une alimentation à découpage.

Schéma fonctionnel de l'alimentation

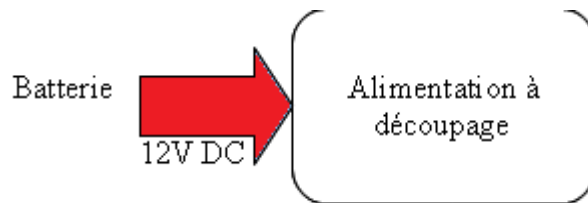


Illustration 3: Schéma fonctionnel de l'alimentation

Pour la réduction de la tension continue de 12 V à 5 V ils ont décidé de prendre un hacheur abaisseur étudié au semestre 3. ils ont seulement recalculé les valeurs des condensateurs et de l'inductance pour pouvoir l'utiliser dans leur cas.

Voici le montage d'un hacheur abaisseur de tension³ :

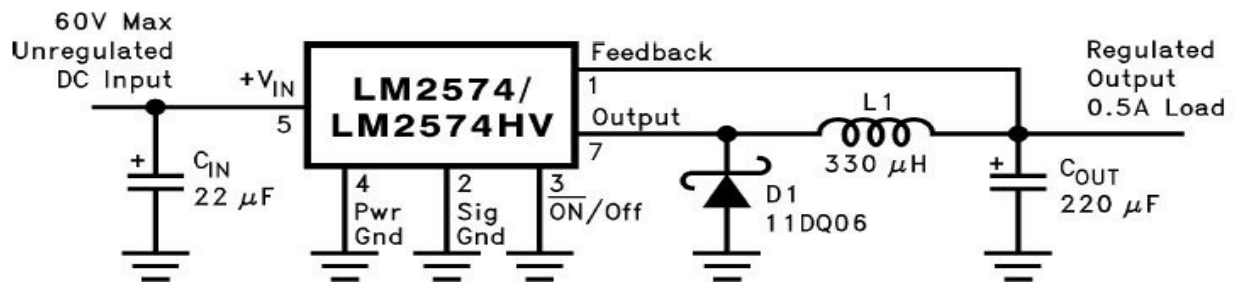


Illustration 4: Montage hacheur abaisseur

Ce montage correspond à une alimentation à découpage.

Ils ont remplacé la valeur du condensateur d'entrée $C_{IN} = 470\mu F$ et celle du condensateur ainsi que celle de la bobine afin d'obtenir un montage avec une tension d'entrée à 12V au lieu de 60V.

La bobine (L1) présente en sortie va permettre de stocker une énergie magnétique, ce qui va rendre possible le fonctionnement en continu. La diode Schottky (D1) ferme la boucle pour laisser la circulation du courant et de la tension libre. Nous utilisons une diode Schottky, car celle-ci a une

3 Le hacheur est représenté par le LM2574 : c'est un régulateur de tension (12 V à 5 V DC).

chute de tension très petite ce qui est recherché pour avoir une tension continue. De plus, le temps de réponse est très petit ce qui est souhaité, car nous sommes en 50 kHz.

3.2 Microcontrôleur

Schéma fonctionnel de niveau 1

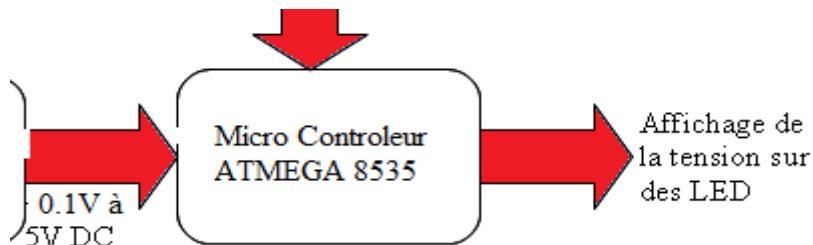


Illustration 5: Schéma fonctionnel du micro-contrôleur

Le micro-contrôleur qu'ils ont utilisés est un ATmega8535, car c'est le plus connu à l'IUT.

Le logiciel qu'ils ont utilisés est code vision AVR⁴.

Grâce au programme le microcontrôleur pourra vérifier la tension de la batterie. En

fonction de la tension mesurée pour que le microcontrôleur allume plus ou moins de LED.



Illustration 6: Logo du logiciel vision code AVR

4 C'est un logiciel de programmation utilisable à l'IUT

3.2.1 Utilisation des LED

Les LED sont alimentées par le microcontrôleur en 5V, les LED qu'ils ont utilisés doivent avoir un courant d'environ 2mA. Pour cela des résistances sont placées en série avec les LED pour limiter le courant qui les traversent. Les LED ont un fonctionnement identique à celui d'une diode normale, elles laissent passer le courant que dans un seul sens, car elles sont polarisées.

3.2.2 Calcul des résistances

Nous savons que la tension en sortie du microcontrôleur est de 5V DC, le courant dans la LED doit être de 2mA. Pour calculer la résistance nous utiliserons la loi d'ohm.

$U = 5V$ (Tension de sortie de l'ATmega8535⁵).

$I = 2mA$ (Courant dans la LED donne par le fabricant).

$U = R \times I$ Donc pour trouver R,

$$R = U / I = 5 / (2 \times 10^{-3}) = 2,5 \text{ k}\Omega$$

Nous prenons la valeur normalisée de 2,7k Ω .

5 Un ATmega8535 : c'est le composant programmable qui permet de gérer la commande de coupure de l'alimentation

3.3 Diviseur de Tension

Schéma fonctionnel de niveau 1



Illustration 7: Schéma fonctionnel du pont de résistance

Pour avoir 5V nous utiliserons un pont diviseur de tension basé sur un pont de résistance. Ce qui nous permettra de réduire la tension de 12V à 5V. Un pont diviseur de tension est constitué de deux résistances.

Comme nous pouvons le voir, la tension d'entrée V_e correspond sur notre projet à la tension de batterie et la tension de sortie V_s alimente le microcontrôleur. Pour calculer les deux résistances nous utiliserons la loi du pont diviseur de tension, $V_s = V_e \times (R_1 / (R_1 + R_2))$.

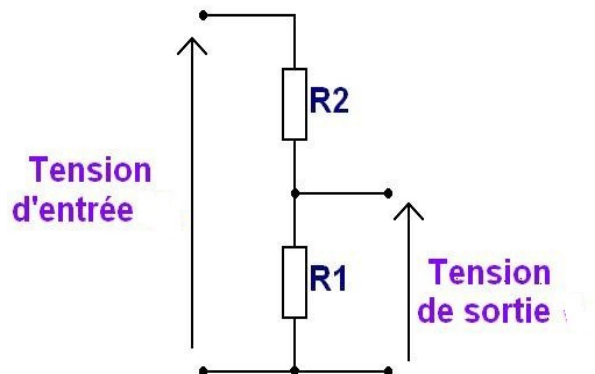


Illustration 8: Montage du diviseur de tension

4 Partie programmation de l'ATMEGA8535.

4.1 Le programme initial

Le programme initial représente le programme qui a été réalisé par les deux autres étudiants qui contrôlaient un vu-mètre à 8 LED. Ce programme est visible en annexe.

4.2 La partie commune du programme

La programmation de l'ATmega8535 a été effectuée sur le logiciel AVR Code vision.

Tout d'abord, ils ont indiqué nos ports d'entrée et de sortie, un port d'entrée est laissé à 0 en hexadécimal et on le note (0x00), le port de sortie lui est mis à 255 en hexadécimal et on le note (0xFF) comme on peut le voir ci-dessous :

```
72
73 // Input/Output Ports initialization
74 // Port A initialization
75 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
76 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
77 PORTA=0x00;
78 DDRA=0x00;
79
80 // Port B initialization
81 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
82 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
83 PORTB=0x00;
84 DDRB=0x00;
85
86 // Port C initialization
87 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
88 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
89 PORTC=0x00;
90 DDRC=0xFF;
```

Illustration 9: Partie du programme sur la configuration des ports

Ensuite, ils ont déclaré des variables sur les ports de sortie qui se fait avec des «`#define`». En dessous on retrouve les déclarations des autres variables qui sont utilisées dans le programme.

Sur le site de M.LEQUEU on y trouve des informations qui ont permis de programmer des variables sur les différents ports de sortie.

```
38 #define ADC_VREF_TYPE 0x00
39
40 #define S7 PORTC.7
41 #define S6 PORTC.6
42 #define S5 PORTC.5
43 #define S4 PORTC.4
44 #define S3 PORTC.3
45 #define S2 PORTC.2
46 #define S1 PORTC.1
47 #define S0 PORTC.0
48
49 // Read the AD conversion result
50 unsigned int read_adc(unsigned char adc_input)
51 {
52     ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
53     // Delay needed for the stabilization of the ADC input voltage
54     delay_us(10);
55     // Start the AD conversion
56     ADCSRA|=0x40;
57     // Wait for the AD conversion to complete
58     while ((ADCSRA & 0x10)==0);
59     ADCSRA|=0x10;
60     return ADCW;
61 }
62
63 // Declare your global variables here
64 unsigned int TensionBat;
65 float a=10;
66 unsigned char tampon[20];
67 float v;
```

Illustration 10: Partie du programme sur les déclarations des variables

4.3 La modification du programme

4.3.1 Le programme test

Mr. LEQUEU m'a demandé de réaliser un programme test qui sert à afficher sur un écran LCD la valeur de tension en sortie du pont de résistance de la carte du vu-mètre. Pour cela, j'ai reproduit sur plaque lab le même pont de résistance que la carte déjà réalisée pour observer la tension de sortie en fonction de la tension d'entrée, afin d'en conclure s'il y a proportionnalité. Ainsi, on obtient un courbe linéaire de fonction affine $Y = aX + b$ qu'on peut observer en annexe. Ensuite,

Monsieur LEQUEU m'a fourni le programme test qui sert à afficher la valeur hexadécimale de

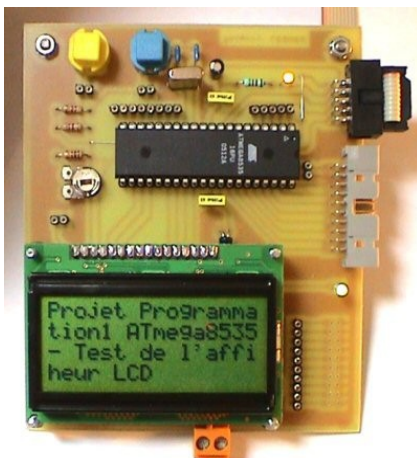


Illustration 11: Carte avec afficheur pour programme Test

```
164
165 LCD module initialization
166 lcd_init(16);
167 lcd_gotoxy(0,0);
168 lcd_putsf("Prog Vu metre 4");
169
170 while (1)
171 {
172     TensionBat=read_adc(0);
173     a=(float)(TensionBat*14.6)/1024;
174
175     sprintf(tampon,"%4d",TensionBat);
176     lcd_gotoxy(3,1);
177     lcd_puts(tampon); //on affiche la valeur de TensionBat.
178
179     v=(float)(TensionBat*5)/1024;
180     sprintf(tampon,"Vbat = %3.3f V",v);
181     lcd_gotoxy(0,2);
182
183     lcd_puts(tampon); //on affiche la valeur de TensionBat.
184
185 }
186 };
187
```

Illustration 12: Programme principal du le Test

la sortir du pont de résistance. Avec les résultats obtenus, j'ai tracé la caractéristique de la tension d'entrée en fonction de la valeur hexadécimale affichée sur le LCD (voir cours en annexe).

4.3.2 La modification du programme principal

Ensuite, j'ai réalisé un programme qui allume 11 LED. Grâce aux valeurs hexadécimales affichées sur le LCD, j'ai pu réaliser le programme principal. Le but de ce programme est d'allumer une LED lorsqu'elle atteint un seuil de tension, d'une part, sous forme hexadécimale, d'autre part, avec des valeurs sous forme de valeur numérique qui sont toutes deux disponibles en annexe. Les LED précédemment allumées de tension inférieure doivent rester allumées. Et lorsque la tension est inférieure aux bornes des tensions de seuil, toutes les LED doivent rester clignoter. C'est à dire, Les tensions de seuil varient de 10V à 15V pour une tension de batterie qui varie de 0V à 15V. Donc si la tension de la batterie est inférieure à 10V, alors toutes les LED clignotent.

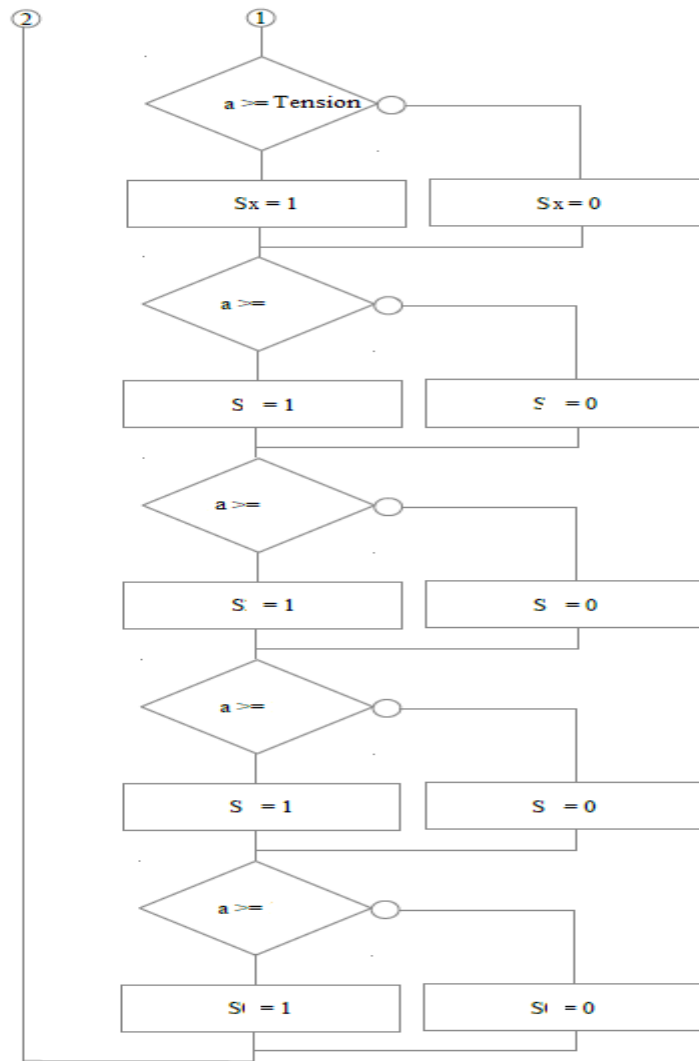


Illustration 13: Ordinogramme de la fonction principal

Voici l'ordinogramme⁶ qui réalise cette fonction :

Le mot « Tension » représente la tension de la batterie et « Sx » est le nom des déclarations des LED on remplace x par le nom de la LED qu'on souhaite allumer.

6 C'est un schéma bloc du programme informatique

4.3.3 La Conversion Analogique Numérique (C.A.N.)

Afin de faciliter la compréhension des lignes de codes du programme, j'ai réalisé une formule qui remplace les valeurs hexadécimales en valeurs numériques.

La tension que l'on affichait était une tension entière, nous voulions une tension décimale donc pour cela nous créons une variable « float A ; ». Nous créons aussi une variable « float TensionBat ; ».

$$0V \rightarrow 5V$$

$$0 \rightarrow 1024$$

La formule que j'ai trouvé pour me permettre d'afficher la tension de la batterie sur l'afficheur LCD est $A = (\text{float})((\text{TensionBat} \times 5) / 1024)$

La valeur maximale que nous visualiserons sera de 15V donc on remplace le 5 par 15.

$$V = (\text{float})((\text{TensionBat} \times 15) / 1024)$$

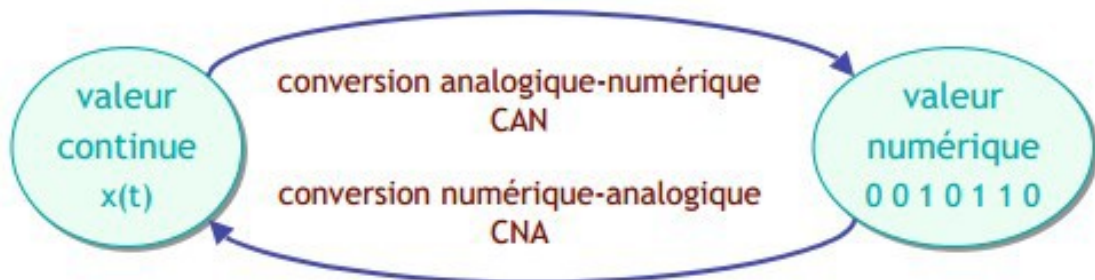


Illustration 14: Convertisseur Analogique Numérique

Enfin, mon professeur m'a demandé de faire clignoter toutes les LED lorsque la tension est inférieure à 10V (qui représentent la valeur minimale des tensions de seuil). On obtient :

```
// lcd_putsf("Prog Vumetre MLk");

while (1)
{

    if(TensionBat<647)
    {
        S12 = 1;
        delay_ms(100);
        S12 = 0;
        delay_ms(100);
    }

    TensionBat=read_adc(0); // entre 0 et 1023 !
    a=(float)(TensionBat*15.5)/1024;

    if(a>=15) //Pour 15V il consomme 130mA.
    {
        S11=1;
    }
    else
    {
        S11=0;
    }
    if(a>=14.5)
    {
        S10=1;
    }
    else
    {
        S10=0;
    }
    if(a>=14)
    {
        S9=1;
    }
    else
    {
        S9=0;
    }
    if(a>=13.5)
    {
        S8=1;
    }
    else
    {
        S8=0;
    }
}
```

Illustration 15: Ma modification apporté au programme

4.3.4 Résultat obtenu

Avant tout démarche en programmation, On m'a demandé de reproduire le pont diviseur pour en déduire les caractéristiques de la tension d'entrée en fonction de la tension de sortie et celle de la tension de la batterie en fonction des valeurs hexadécimales.

Ensuite avec les valeurs hexadécimales et numériques, j'ai modifié le programme en remplaçant avec mes valeurs, et en ajoutant trois LED supplémentaires. J'en déduit que chaque semaine les valeurs de seuil varie, donc j'en conclut que notre montage dépend de la température.

En annexe, vous trouverez les différents programmes associés à leur courbe correspondante.

5 Partie réalisation de la nouvelle carte électronique

Voici le nouveau schéma structurel de notre carte.

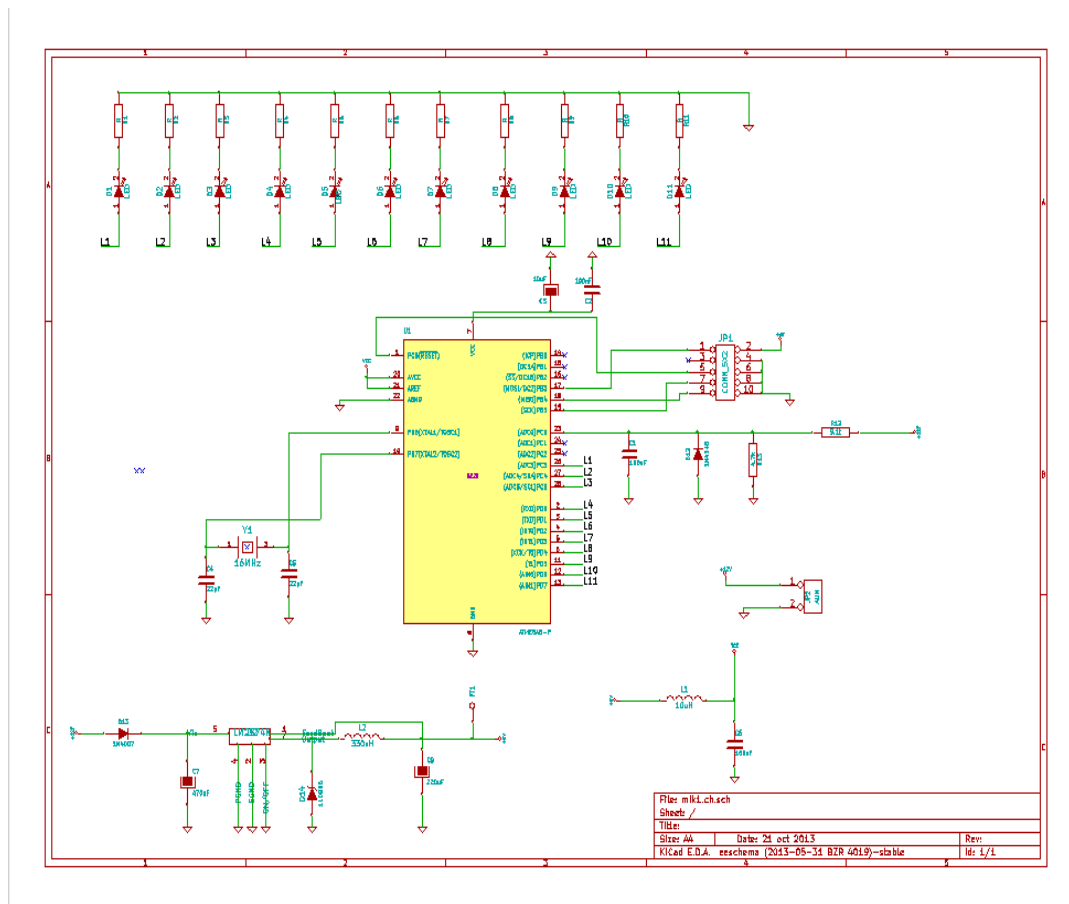


Illustration 16: Nouveau montage électrique

On peut remarquer qu'on a exactement le même schéma électrique que le schéma initial sauf que dans le nouveau schéma on a remplacé l'ATmega8535 par un ATmega8, car il est moins encombrant et cela nous permettra de pouvoir insérer le câble d'alimentation de la carte.

En annexe, vous pourrez voir le document constructeur de l'ATmega8, ce document m'a permis de réaliser sur KICAD le branchement de mon ATmega8.

5.1 Typon, gravure et soudure

Après avoir réalisé le schéma structurel sur KICAD, j'ai placé les différents composants de manière à avoir le moins de « straps » possible et de façon à libérer de la place pour le câble d'alimentation. J'ai réalisé la netliste pour pouvoir faire le typon j'ai réalisé les différentes pistes entre tous les composants de notre montage avant de réaliser le plan de masse⁷. les pistes en vert nous montrent le cuivre et les pistes en rouge montrent les pistes que nous relirons avec des fils sur la face composant (« straps »).

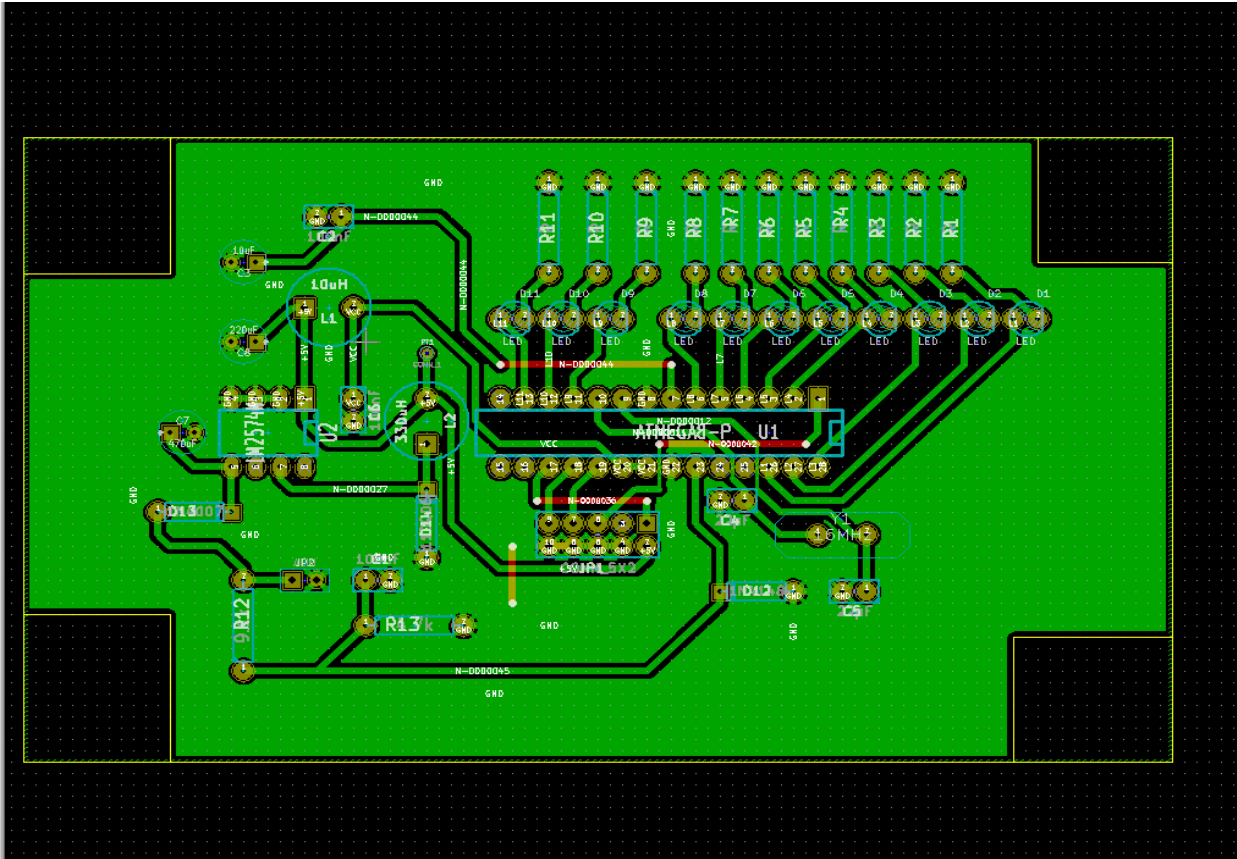


Illustration 17: Typon de ma carte

Après avoir imprimé mon typon sur un calque, j'ai insolé mon typon sur une carte de cuivre, après l'avoir passé au révélateur je peux visualiser les pistes avec une petite couche de noir.

Ensuite j'ai gravé cette carte grâce au perchlore, cette étape nous a permis de retirer le cuivre autour des pistes.

Après ces différentes étapes, j'ai effectué le perçage des différentes pastilles où j'ai soudé les composants par la suite. Les différents composants ont été soudé avec un fer à souder.

⁷ Le plan de masse signifie qu'on relie toutes les masses entre elles.

7 Conclusion

À partir du programme qui contrôle huit LED, j'ai modifié le programme pour gérer onze LED à partir des valeurs hexadécimales et des valeurs numériques que j'ai mesuré avec un programme test. Pour me faciliter dans la formule de conversion et voir s'il y avait proportionnalité j'ai tracé avec un tableau excel les courbes des tensions d'entrée en fonction des tensions de sortie.

J'ai remarqué que les valeurs de seuil se modifiaient au fur et à mesure des jours j'en ai déduit que mon montage dépend de la température d'où l'origine de plusieurs programmes avec des valeurs de tensions différentes.

Par ailleurs, le professeur m'a demandé de rajouter une option qui doit faire clignoter toutes les LED quand la tension est inférieure à 10V cela sert de témoin, la carte est bien alimentée entre 0V et 10V.

Enfin, Le câble d'alimentation issue du KART ne peut pas être inséré dans la carte déjà réalisée, donc on m'a demandé de faire une autre carte électronique moins encombrante. Je dois remplacer l'ATMega 8535 par un ATMega 8, car il est de plus petite superficie, afin de pouvoir déplacer les composants pour faire place au câble d'alimentation.

8 Résumé

Dans le cadre du cours d'étude et réalisation, on m'a imposé de modifier un projet existant qui contribue à l'association Kart.

Mon projet est un vu-mètre à onze LED. Lorsque que la tension est à 10V une LED s'allume et ainsi de suite jusqu'à 15V en laissant allumé les LED précédemment active. De plus, je dois faire clignoter les LED lorsque la tension est inférieure à 10V.

Mon cahier des charges répond aussi a faire une nouvelle carte électronique moins encombrante, afin qu'on puisse alimenter la carte avec le câble de diamètre imposant fourni par le Kart. La solution trouvée pour cette nouvelle carte est de remplacer le micro-contrôleur ATmega 8535 par un ATmega 8. En remplaçant, l'Atmega on pourra déplacer les composants encombrants

Ce dossier constitué de trois grandes parties répond au cahier des charges.

Dans un premier temps, j'ai fait l'analyse détaillé du projet.

Dans un second temps, j'ai expliqué en quoi consisté le programme et quelle modification j'ai apporté.

Dans un troisième, j'ai tenté de vous montrer par quel moyen et de quelle manière j'ai modifié la carte électronique.

Bibliographie

[1] **Thierry LEQUEU**. *La documentation de Thierry LEQUEU sur OVH*, 22 octobre 2012, [En ligne]. (Page consultée le 24/10/2012) <<http://www.thierry-lequeu.fr/>>

[2] **Nina**. *Montage pont diviseur de tension*, 05/06/2012, [En ligne]. (Page consultée le 05/10/2012) <<http://www.astuces-pratiques.fr/electronique/le-pont-diviseur-de-tension>>

[3] **ARAKELIAN et GALLIER**. « *Production personnelle* ». ARAKELIAN et GALLIER, 2012.

[4] **CNRS Orléans**. « *Conversion analogique numérique* », , [En ligne]. (Page consultée le 24/10/2012) <http://lpc2e.cnrs-orleans.fr/~ddwit/enseignement/cours_elec_num_5.pdf>

Index des illustrations

Illustration 1: schéma fonctionnel du vu-mètre.....	8
Illustration 2: Schéma fonctionnel détaillé du projet.....	9
Illustration 3: Schéma fonctionnel de l'alimentation.....	10
Illustration 4: Montage hacheur abaisseur.....	10
Illustration 5: Schéma fonctionnel du micro-contrôleur.....	11
Illustration 6: Logo du logiciel vision code AVR.....	11
Illustration 7: Schéma fonctionnel du pont de résistance.....	13
Illustration 8: Montage du diviseur de tension.....	13
Illustration 9: Partie du programme sur la configuration des ports.....	14
Illustration 10: Partie du programme sur les déclarations des variables.....	15
Illustration 11: Carte avec afficheur pour programme Test.....	16
Illustration 12: Programme principal du le Test.....	16
Illustration 13: Ordinogramme de la fonction principal.....	18
Illustration 14: Convertisseur Analogique Numérique.....	19
Illustration 15: Ma modification apporté au programme.....	20
Illustration 16: Nouveau montage électrique.....	22
Illustration 17: Typon de ma carte.....	23

Annexe