

Université François Rabelais de Tours

Institut Universitaire de Technologie de Tours

Département Génie Électrique et Informatique Industrielle

Projet d'étude et réalisation



Wattmètre pour chargeur de batterie 48V 50A

Arthur CLEACH
Pierre-Alexandre DIARD
2^{ème} année-S2
Promotion : 2006/2008

Enseignant:
M Thierry Lequeu
M Jacky Brun

Université François Rabelais de Tours

Institut Universitaire de Technologie de Tours

Département Génie Électrique et Informatique Industrielle

Wattmètre pour chargeur de batterie 48V 50A

Arthur CLEACH
Pierre-Alexandre DIARD
2^{ème} Année-S2
Promotion : 2006/2008

Enseignant:
M Thierry Lequeu
M Jacky Brun

Sommaire

Introduction.....	4
1.Le projet :.....	5
1.1.Cahier des charges.....	5
1.2.Planning.....	5
1.3.Synoptique.....	6
1.4.Définition de la carte.....	7
2.Mesures :.....	9
2.1.Principe du convertisseur analogique numérique.....	9
2.2.Recherche de coefficient :.....	9
2.3.Résultats.....	10
2.4.Vérification des coefficients et problème rencontré.....	11
3.Programme :.....	12
3.1. Tests des différentes fonctions.....	12
3.2.Explication des fonctions du programme.....	13

Introduction

Dans le cadre de notre projet d'étude et réalisation, nous avons élaboré un système qui doit permettre de mesurer le courant de charge d'un kart électrique de 0 à 50 A, de mesurer la tension des batteries de 48 V (de 40 V à 64 V) puis d'afficher la puissance et enfin la température ambiante du boîtier et l'énergie.

Le wattmètre est un appareil destiné à mesurer une puissance. Pour cela, il doit indiquer la valeur moyenne du produit (en valeur instantanée) d'une tension et d'un courant. Un wattmètre comporte toujours deux circuits : un circuit de courant et un circuit de tension.

Tout d'abord il s'agira de présenter le projet dans son ensemble, puis d'étudier les mesures réalisées pour enfin expliquer la partie programmation.

1. Le projet :

1.1. Cahier des charges



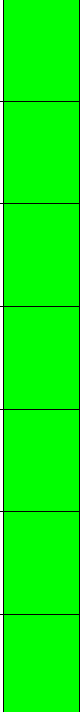


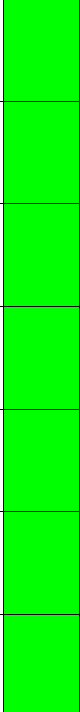


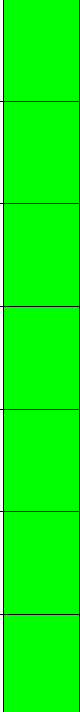



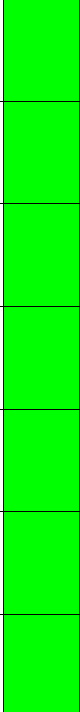

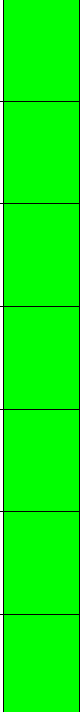









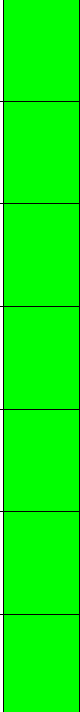





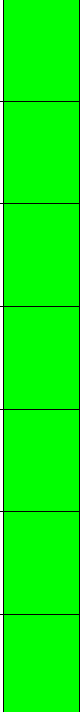




Notre projet consiste à la visualisation sur un afficheur LCD, du courant de charge de 0 à 50 A, de la tension des batteries de 48 V, de la puissance, de l'énergie et de la température lors de la charge électrique des batteries d'un kart électrique.

Pour cela nos composants principaux seront un microcontrôleur ATMEL Atmega 8535 munis d'un CAN, un capteur de courant LEM HAS 200-S, et un afficheur LCD 16 segments.

La carte et le typon nous étant fournis, nous nous sommes occupés que de la partie programmation.

Les contraintes sont tout d'abord de garder le module d'alimentation qui, à partir de 48 V donne du +15V/-15V, de plus la taille du boîtier de protection est imposé : 180*130*75. La carte devra être sur-élevée pour que le câble de gros calibre passe en dessous de la carte et que l'on puisse lire l'afficheur à travers la plaque transparente du boîtier. La carte sera implantée dans un boîtier de protection.

1.2. Planning

Numéros de semaines	4	5	6	7	8	9	10	11	12	13	14
Cahier des charges,planning	 										
Documentation	 										
Définition de la carte		 									
Mise sous tension			 								
Programmation:						 	 	 	 		
1.Mesures						 	 				
2.Calculs							 	 			

3.Base de temps												
4.Énergie												
Dossier à rendre												
Oral												

1.3. Synoptique

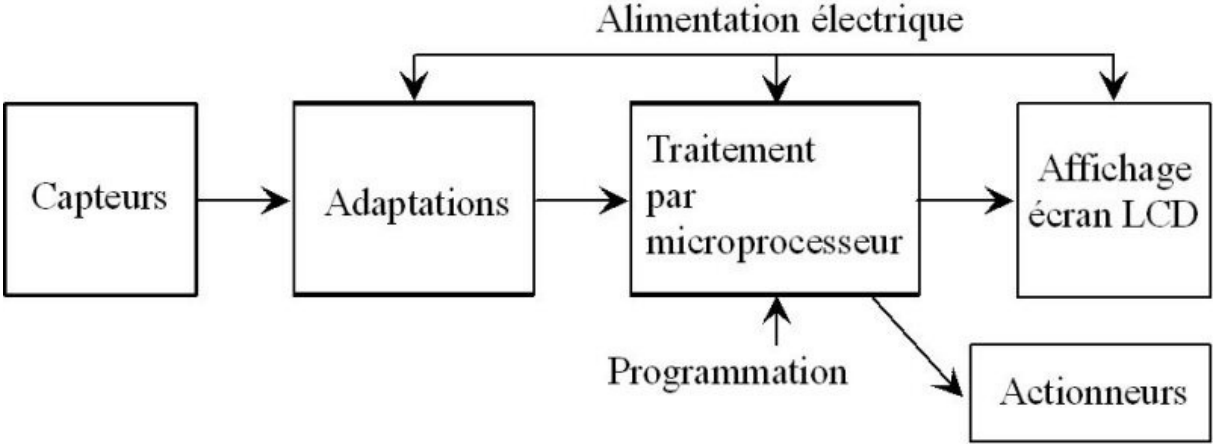


Illustration 1: synoptique d'une carte à microcontrôleur

Dans ce synoptique, il y a tout d'abord les capteurs ici, il s'agit d'un capteur de courant et d'un capteur de température :

- Le capteur de courant va nous être utile pour réaliser différentes mesures en fonction du courant, plus particulièrement pour rechercher le coefficient qui nous servira par la suite dans la bonne utilisation du programme.

- Le capteur de température va permettre après avoir réalisé le programme de surveiller la température ambiante du boîtier de la carte.

Ensuite, il y a l'adaptation, c'est la partie mesure de notre projet, c'est à dire la partie recherche de coefficient, qui sera explicitée dans la suite du rapport.

Enfin, il y a le traitement par microprocesseur qui représente la partie la plus importante de notre projet, c'est-à-dire la partie programmation sous le logiciel AVRCodeVision.

L'affichage sur l'écran LCD est compris dans la programmation.

1.4. Définition de la carte

1. Récupération du courant

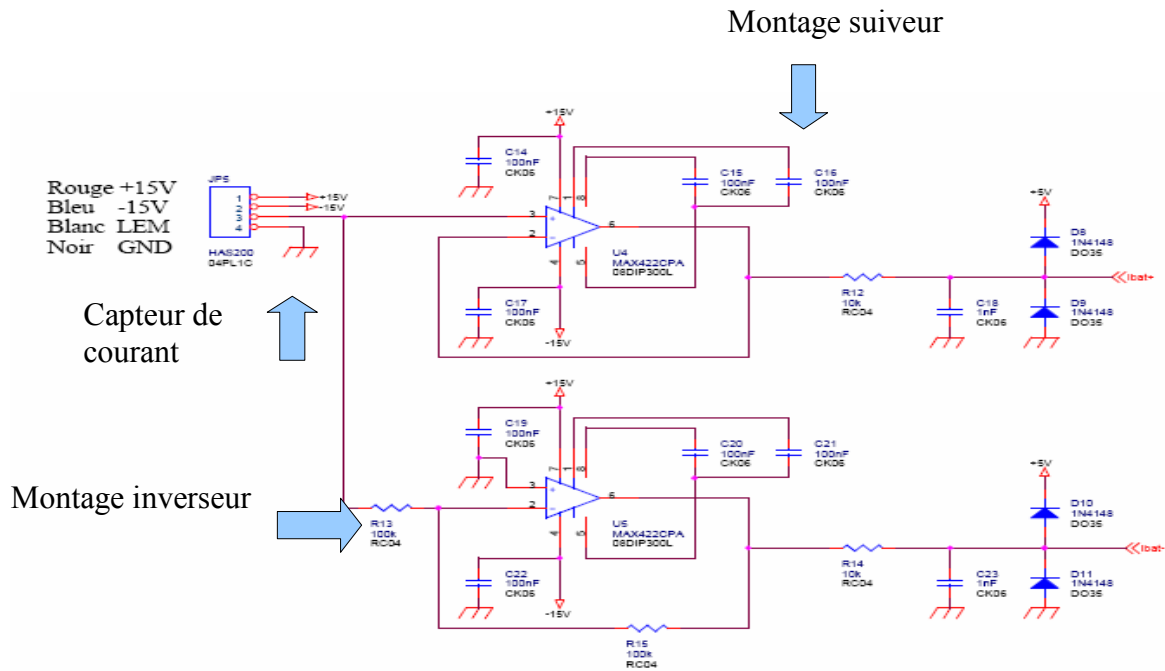


Illustration 2: Montage en sortie du capteur de courant

Le capteur de courant récupère les courants pour les envoyer sur l'un des deux montages, suivant son signe.

Le montage inverseur inverse les courants négatifs qui vont sur une entrée du microcontrôleur.

Le montage suiveur envoie directement l'image du courant positif sur une patte du microcontrôleur.

2. Récupération de la tension

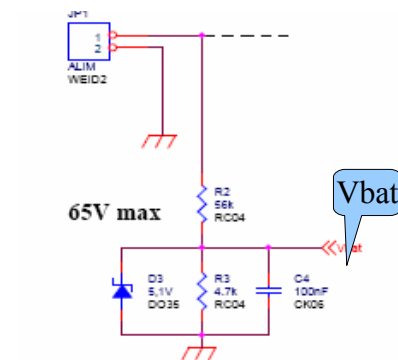


Illustration 3: pont diviseur

L'alimentation de 48 Volts est reliée au pont diviseur. Le pont diviseur permet d'envoyer au convertisseur analogique numérique une tension entre 0 et 5 Volts. On peut remarquer la sortie Vbat qui est relié à l'entrée du CAN.

3. Microcontrôleur et afficheur LCD

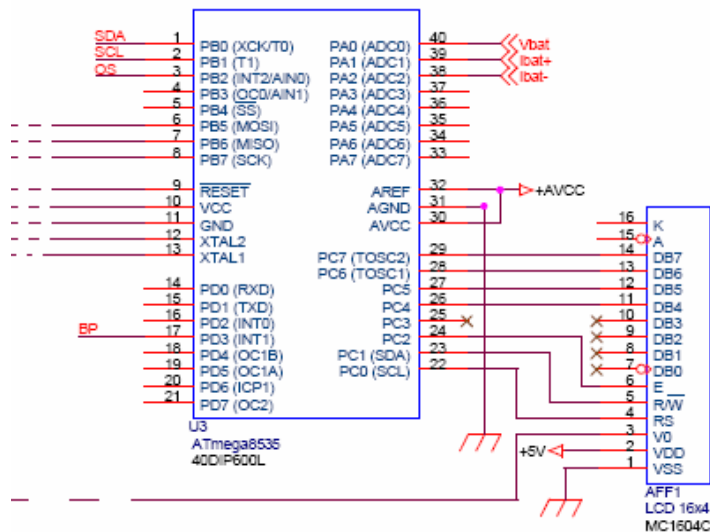


Illustration 4: Microcontrôleur

Pour notre projet, nous nous sommes servi de l'entrée INT1 pour la remise à zéro de l'énergie à l'aide du bouton poussoir. La sortie ADC0 nous permet de récupérer la valeur de la tension. Les sorties ADC1 et ADC2 nous ont servi à récupérer l'image du courant négatif et positif.

2. Mesures :

2.1. Principe du convertisseur analogique numérique

Valeur d'entrée et de sortie du microcontrôleur :

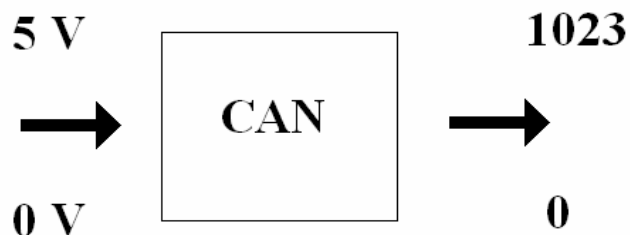


Illustration 5: Valeur d'entrée et de sortie du CAN

Le principe du convertisseur analogique numérique est simple, il est alimenté entre 0 et 5 Volts, et renvoi en sortie une valeur entre 0 et 1023. Pour le calcul des coefficients nous avons dû établir un produit en croix qui est $1023/5$.

2.2. Recherche de coefficient :

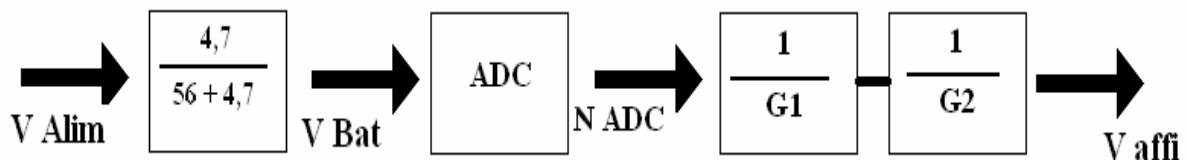


Illustration 6: Schéma bloc représentatif du circuit électrique

Le schéma bloc nous à permis de comprendre le fonctionnement du circuit, pour ensuite chercher une technique servant à trouver les coefficients dont nous avons besoin dans la réalisation de notre programme.

Le premier bloc représente le calcul du pont diviseur.

Le bloc ADC représente le CAN¹, c'est un CAN sur dix bits, ce qui correspond bien à la sortie compris entre 0 et 1023 car $(2^{10}) - 1 = 1023$.

Les Deux blocs suivant correspondent aux coefficients que l'on ajoutera au programme pour envoyer sur l'afficheur LCD la même valeur d'alimentation pour la tension.

Principe utilisé pour calculer le coefficient de la tension :

Il faut établir une relation de récurrence entre la valeur d'alimentation et celle de sortie du CAN. Il faut donc relever plusieurs valeurs pour pouvoir tracer une courbe de régression. Cette courbe de régression permet d'afficher le coefficient. Plus précisément, nous avons créé un programme qui permet d'afficher la valeur de la tension sortie du CAN, c'est à dire entre 0 et 1023. Après plusieurs relevés, nous avons rentré les valeurs obtenues dans un tableau sous excel, ce qui nous à permis d'obtenir la courbe 1.

Principe utilisé pour calculer le coefficient de le courant :

C'est le même principe que pour la tension, sauf que l'on place le capteur de courant en court circuit. Après avoir imposé différentes valeurs de courant, Les valeurs en sortie du CAN sont affichable. Ces valeurs nous permette de tracer la courbe 2.

2.3. Résultats

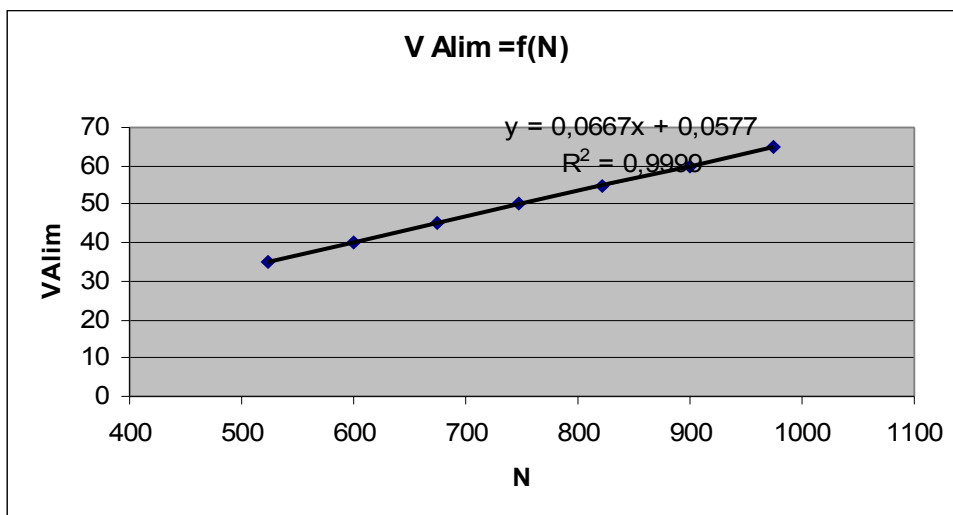


Illustration 7: Courbe 1 : Valeur de la tension en fonction de la valeur de sortie du CAN

1 Convertisseur Analogique Numérique

Il est possible d'observer que la courbe est linéaire et que la courbe de régression est très précise ($R^2=0,9999$).

Grâce à la courbe de régression, il est possible d'afficher le coefficient de la courbe, ce coefficient sera celui qui servira lors de la programmation.

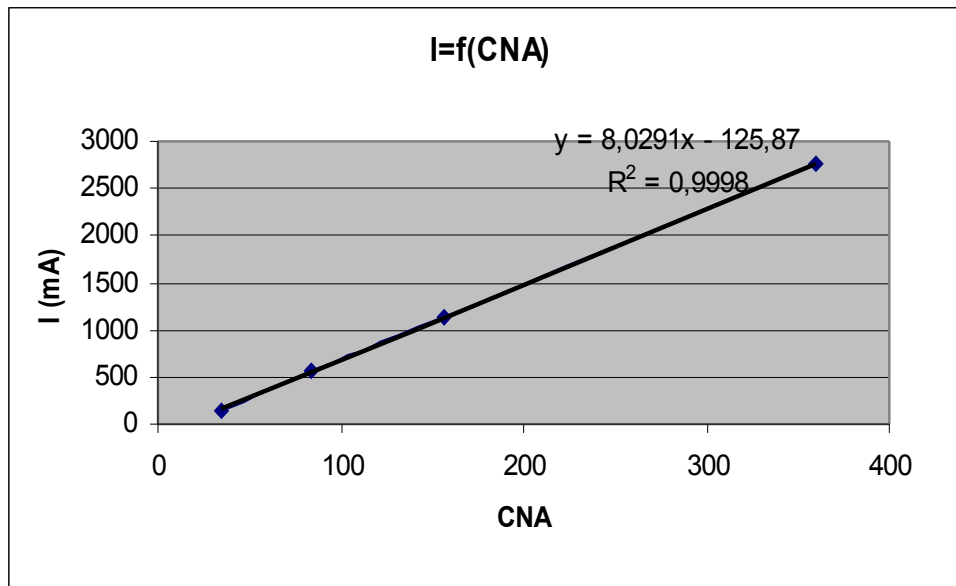


Illustration 8: Courbe 2 : Valeur du courant en fonction de la valeur de sortie du CAN

Il faut observer que la courbe est linéaire et que la précision est bonne, le coefficient servira dans la fonction pour afficher le courant.

2.4. Vérification des coefficients et problème rencontré

Pour vérifier nos résultats il nous a suffi de rentrer les différentes fonctions de notre programme.

Le problème que nous avons rencontré est la différence observée entre la valeur d'alimentation et la valeur affichée sur l'écran LCD. Cette différence était de 2 Volts au lieu d'être nulle.

Après réflexion, nous avons supposé que s'était la diode zener du pont diviseur qui créait probablement une perturbation. Notre supposition a été vérifiée puisqu'en enlevant cette diode zener, nous avons obtenu des valeurs identiques entre l'alimentation et la valeur affichée.

Après avoir trouvé les coefficients, nous sommes passés à la programmation

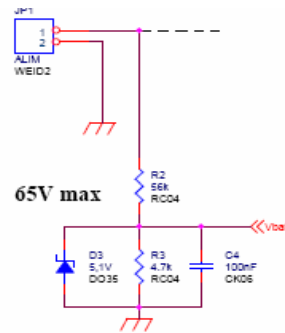


Illustration 9: Pont diviseur

3. Programme :

3.1. Tests des différentes fonctions

```
sprintf(tampon,"%2d",i);  
lcd_gotoxy(11,0);  
lcd_puts(tampon);
```

- i correspond à la variable de sortie du CAN.

- tampon est la chaîne de caractère car pour envoyer sur l'afficheur il faut que ce soit une chaîne de caractère.

- %2d signifie qu'il aura deux chiffres d' affichés sur l'afficheur.

Disposition de l'afficheur LCD :

L'afficheur LCD contient 16 colonnes et 4 lignes.

I	=				A				E	=				J
U	=			V										
T	=			°	C									
P	=					W								

Illustration 10: Représentation de l'afficheur LCD

La fonction « `sprintf(tampon,"%2d",i);` » permet de placer la variable `i` dans la chaîne de caractère.

La fonction « `lcd_gotoxy(11,0);` » définit l'emplacement sur l'afficheur. Ici cela signifie que l'on affiche les données sur la 11^{ème} colonne et sur la première ligne.

La fonction « `lcd_puts(tampon);` » a pour but d'envoyer la chaîne de caractère sur l'afficheur LCD.

3.2. Explication des fonctions du programme

1. La température

```
temp=lm75_temperature_10(CPT1);
signe='+';
if (temp<0)
{
signe='-';
temp=-temp;
};
sprintf(tampon,"Temp = %c%i.%u\xdfC",signe,temp/10,temp%10);

lcd_gotoxy(0,2);
lcd_puts(tampon); //on affiche
```

Grâce au capteur de température, on relève la valeur du CAN de la sortie CPT1.

La variable `temp` a pour but de prendre la valeur CPT1. La boucle condition « `if` » permet de récupérer la température positive.

2. Le Courant

```

j=read_adc(2);           // Courant négatif
j=j*8.0291;             // coefficient de 8.0291
if(j==0)                // 0 est la valeur par défaut
{
    j=read_adc(1);      // Courant positif
    j=j*8.0291;
    sprintf(tampon,"%4d",j); // 4 chiffres
    lcd_gotoxy(2,0);      //colonnes N° 2 et 1er ligne
    lcd_puts(tampon);
}
else
{
    sprintf(tampon,"%4d",j);
    lcd_gotoxy(2,0);
    lcd_puts(tampon);
}

```

Adc2 correspond au courant négatif : ibat -
Adc1 correspond au courant positif : ibat +

La commande « read_adc(2); » permet de récupérer sur la sortie adc2 du microcontrôleur, l'image du courant négatif. On multiplie cette valeur par le coefficient trouvé précédemment. Après comparaison avec la valeur par défaut, (0 signifie qu'il n'y a aucune donnée sur la sortie concernée adc2) si la valeur est égale à 0, le programme affiche la valeur lue sur la patte adc1. Sinon la valeur affichée est celle de la patte adc2.

3. La tension

```

i=read_adc(0);
i=i*0.0667;           // coefficient de 0,0667
sprintf(tampon,"%2d",i);
lcd_gotoxy(11,0);
lcd_puts(tampon);

delay_ms(100);       // Attente de 0,1s

```

On lit sur la patte adc0 la valeur de la tension entre 0 et 1023 mais après multiplication du coefficient de la tension, on obtient la valeur d'alimentation. Le « delay » produit une attente de 0,1s.

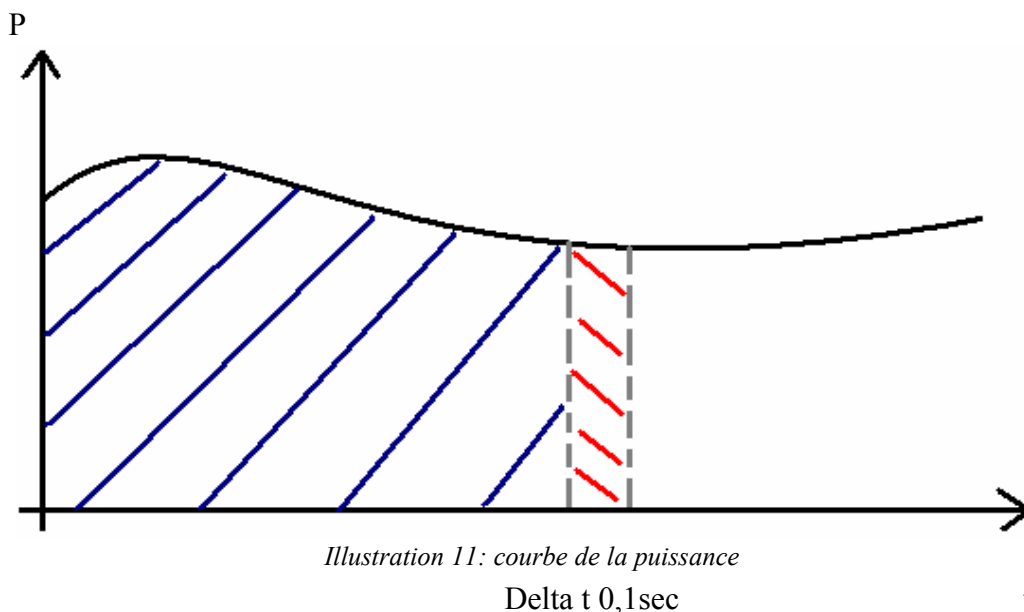
4. La puissance et l'énergie

```
p=i*(j/1000); // i : tension et j : courant
sprintf(tampon,"%4d",p);
lcd_gotoxy(2,3);
lcd_puts(tampon);

E=E+(p*0.1);
if(PIND.3==0) // adresse bouton poussoir
{
    E=0;
}
sprintf(tampon,"%6d",E);
lcd_gotoxy(2,1);
lcd_puts(tampon);
```

Après avoir récupéré le courant et la tension. On peut en déduire la puissance et l'énergie de la batterie.

L'énergie étant l'intégrale de la puissance, on incrémente l'énergie E avec la puissance multipliée par 0,1sec.



Sur la représentation de la puissance P, E est représenté par l'aire située sous la courbe. Donc pour récupérer l'énergie on lui rajoute toutes les 100ms l'air suivante, soit $0,1 * P$. Le bouton poussoir PIND3 permet de remettre l'énergie à zéro.

Conclusion

Notre projet a consisté en une visualisation rapide des valeurs utilisées pour la charge d'une batterie. La résolution de la problématique nous a amené à nous interroger sur les améliorations possibles comme la mise en place d'un timer au sein du programme afin de gérer le temps de boucle.

Index des illustrations

Illustration 1: Synoptique d'une carte à microcontrôleur.....	6
Illustration 2: Montage en sortie du capteur de courant.....	7
Illustration 3: Pont diviseur.....	8
Illustration 4: Microcontrôleur.....	8
Illustration 5 : Valeur d'entrée et de sortie du CAN.....	9
Illustration 6 : Schéma bloc représentatif du circuit électrique.....	9
Illustration 7 : Courbe 1 : Valeur de la tension en fonction de la valeur de sortie du CAN.....	10
Illustration 8 : Courbe 2 : Valeur du courant en fonction de la valeur de sortie du CAN.....	11
Illustration 9 : Pont diviseur.....	12
Illustration 10 : Représentation de l'afficheur LCD.....	12
Illustration 11 : Courbe de la puissance.....	15

Bibliographie:

Site de M Thierry Lequeu

