

# 1. Université François-Rabelais de Tours

Institut Universitaire de Technologie de Tours

Département Génie Électrique et Informatique Industrielle

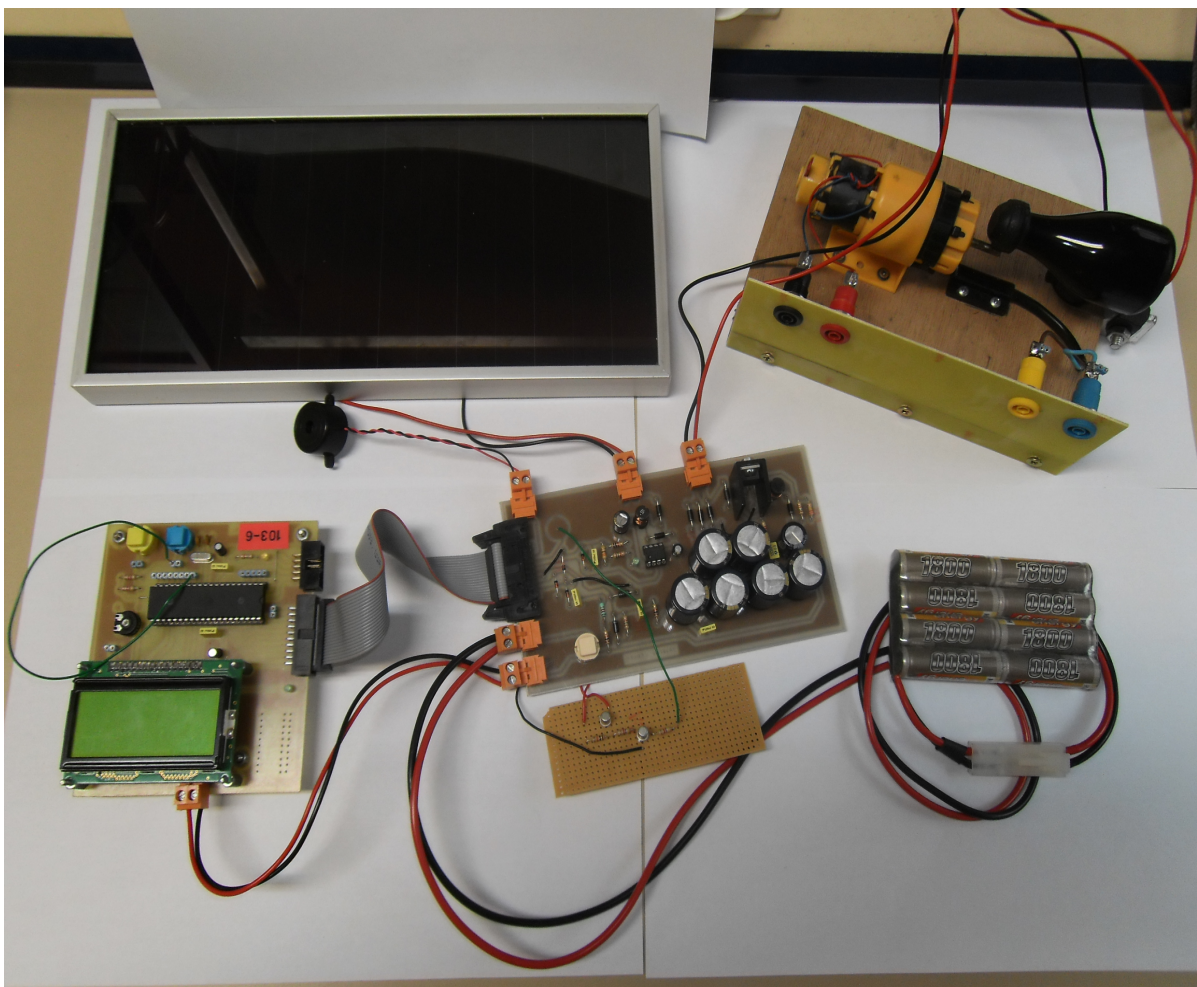
UNIVERSITE FRANCOIS-RABELAIS  
TOURS



Institut Universitaire de Technologie

Département  
GENIE ELECTRIQUE ET  
INFORMATIQUE INDUSTRIELLE

## *Éclairage à LED pour vélo avec autonomie*



DELAROCHE Thibault

MOLLINIER Vincent

Groupe K4A

Promotion 2009/2012

Enseignants

LEQUEU Thierry

RODIER Sofi

# 1. Université François-Rabelais de Tours

Institut Universitaire de Technologie de Tours

Département Génie Électrique et Informatique Industrielle

UNIVERSITE FRANCOIS-RABELAIS  
TOURS



Institut Universitaire de Technologie

Département  
GENIE ELECTRIQUE ET  
INFORMATIQUE INDUSTRIELLE

## *Éclairage à LED pour vélo avec autonomie*

DELAROCHE Thibault

MOLLINIER Vincent

Groupe K4A

Promotion 2009/2012

Enseignants

LEQUEU Thierry

RODIER Sofi

# Sommaire

1.Université François-Rabelais de Tours.....	1
Introduction.....	4
1.Présentation du projet.....	5
1.1.Cahier des charges.....	5
1.2.Définition des différentes fonctions.....	6
1.3.Planning.....	7
2.Pôle électronique du projet.....	8
2.1.Étude des sources d'énergies.....	8
2.2.Réalisation de la carte électronique.....	15
3.Pôle informatique du projet.....	25
3.1.Présentation de l' ATmega8535.....	25
3.2.Explications du programme.....	26
3.3.Remarques, problèmes rencontrés et utilisation de CodeVision.....	32
Conclusion.....	42
Résumé.....	43
Annexes.....	46

## Introduction

Au cours de ce quatrième semestre nous devons réaliser un projet dans le cadre du cours d'étude et réalisation. Nous avons choisi parmi plusieurs sujets proposés et nous nous sommes orientés vers un projet concernant l'éclairage d'un vélo.

Dans un souci d'économie d'énergie, ce projet propose d'utiliser différentes sources d'énergie pour alimenter l'éclairage à LED du vélo. Telles qu'une batterie, une dynamo et enfin un panneau solaire. Nous allons également utiliser un microcontrôleur de type ATmega8535 qui permettra à l'ensemble de ce système de fonctionner en totale autonomie.

Dans un premier temps, nous allons approfondir la présentation du projet, puis nous allons aborder l'aspect électronique du projet avec l'étude des différentes sources d'énergies ainsi que l'étude et la réalisation de la carte électronique. Dans un second temps, nous allons voir l'aspect informatique du projet avec l'étude et la programmation de l'ATmega8535. Pour terminer, nous allons aborder les phases de tests et de mesures de la carte et du programme.

# 1. Présentation du projet

Ce projet consiste à réaliser l'alimentation de l'éclairage à LED en autonomie d'un vélo. Pour réaliser ce projet nous devons mettre en œuvre plusieurs sources d'énergies ainsi qu'un microcontrôleur. La nature des sources d'énergies telle que le panneau solaire ou encore la dynamo permettront au système d'être alimenté en permanence pendant l'utilisation du vélo. De plus la mise en œuvre d'un microcontrôleur permettra de choisir qu'elle type de source utiliser pour recharger la batterie mais également de connaître le niveau de tension fournie par chacune des sources.

## 1.1. Cahier des charges

Nous reprenons deux projets existants : celui de Louise Prodhomme et celui de Jean Todesco et Trystan Rouze. Nous devons réaliser un chargeur de batterie pour l'éclairage d'un vélo par panneau solaire afin de rendre ce système autonome en énergie. Il comporte déjà une batterie 9V, un ensemble de super-condensateurs et une dynamo. Nous utiliserons un ATmega8535 pour gérer les sources d'énergies citées précédemment. On utilise le panneau solaire pour recharger la batterie la journée. Pour prolonger l'autonomie de la batterie, on utilisera aussi une dynamo. Lorsque le vélo sera arrêté (à un croisement par exemple), les condensateurs se chargeront de maintenir l'éclairage. Les condensateurs seront branchés à la batterie. Ce système doit fonctionner par tout les temps (pluie, soleil...) Il doit prendre le moins de place possible car ce système doit être transporté sur un vélo. Nous devons mettre en œuvre toutes les sources d'énergies mises à disposition. Le microcontrôleur ATmega8535 permettra la gestion de ce système.

### Objectif à atteindre

- Rendre autonome l'éclairage d'un vélo
- Doit fonctionner par tous les temps

### Contraintes

- Doit prendre le moins de place possible (encombrement minimum)
- Faire attention à la consommation
- Éviter des dépenses élevées pour obtenir un budget minimum
- Utilisation d'un panneau solaire
- Utilisation d'un ATmega8535

## 1.2. Définition des différentes fonctions

Pour faciliter la compréhension de notre projet, nous allons utiliser, pour définir les fonctions à réaliser, des schémas fonctionnels.

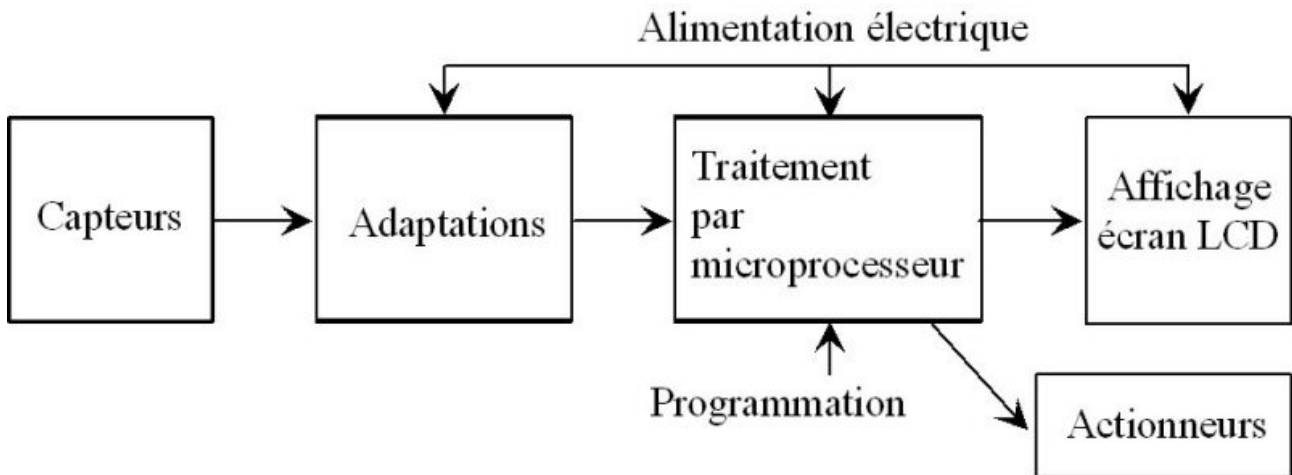


Illustration 1: Schéma fonctionnel général

Voici le schéma fonctionnel que doit réaliser notre projet

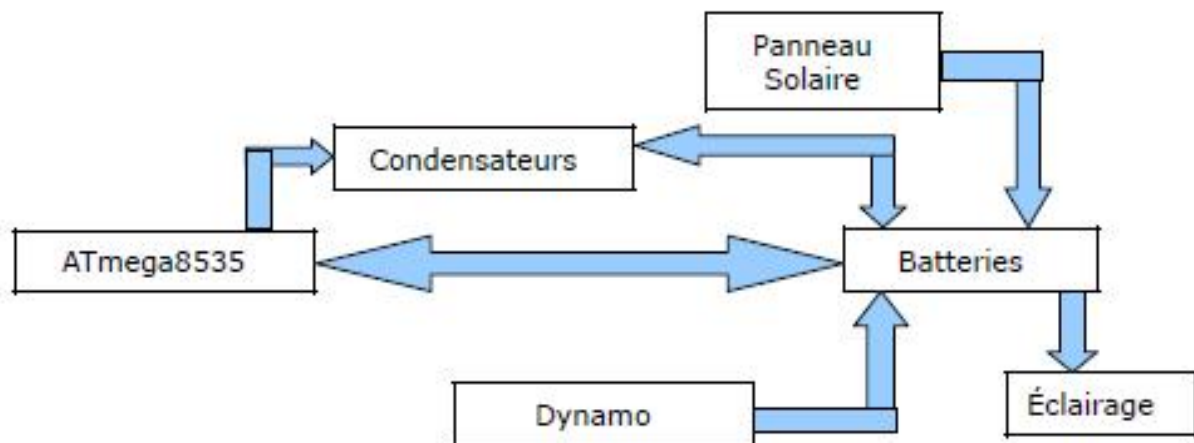


Illustration 2: Schéma fonctionnel du projet

Ci-dessus, nous pouvons voir que les différentes sources d'énergies fournissent de l'énergie à la batterie de 9V. Puis notre batterie alimente l'éclairage du vélo. De plus, nous pouvons remarquer sur ce schéma que c'est bien l'ATmega8535 qui gère l'utilisation des différentes sources d'énergies.



## 2. Pôle électronique du projet

Dans cette partie nous allons aborder l'ensemble des sources d'énergies utilisées, ainsi que l'étude et la réalisation de la carte électronique.

### 2.1. Étude des sources d'énergies

#### 2.1.1. Le panneau solaire

Afin de rendre autonome l'éclairage du vélo nous ajoutons un panneau solaire qui pourra fournir de l'énergie à volonté lors d'une exposition au soleil du dispositif.

Pour connaître les différentes caractéristiques du panneau nous avons effectué plusieurs tests en extérieur afin de savoir ce que pouvait fournir le panneau en tension et en courant en fonction de la charge et de l'ensoleillement. Pour procéder au test, nous avons mis en place le panneau solaire au soleil puis nous avons mis en parallèle un voltmètre, en série un ampèremètre et pour simuler la charge nous avons branchés aux bornes du panneau un rhéostat d'une valeur de  $250\Omega$ . Sachant que le rendement des cellules photovoltaïques se dégrade lorsqu'elles chauffent, nous avons fait des mesures en fonction du temps d'exposition au soleil.



*Illustration 4: Réalisation du test pour le panneau solaire*



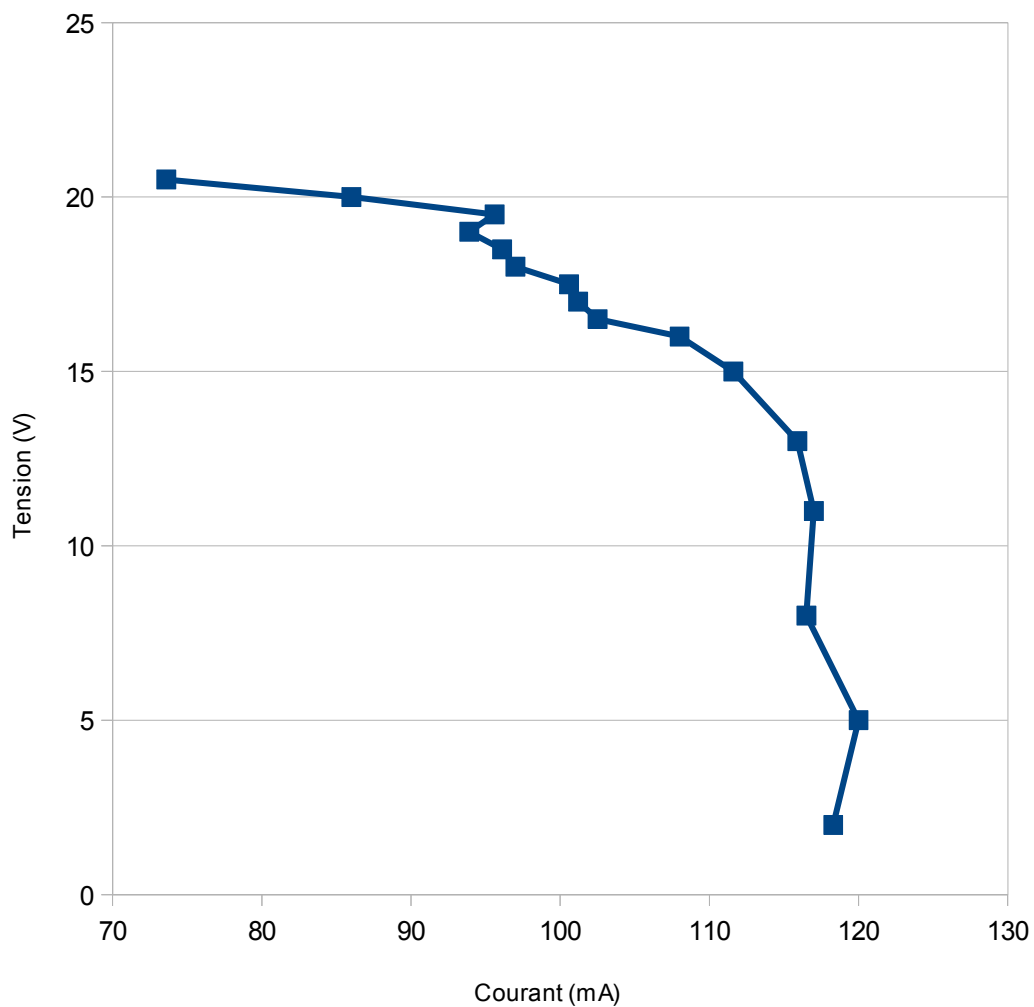
Voici les résultats obtenus

Tension (V)	20.5	20	19.5	19	18.3	18	17.5	17	16.5	16	15	13	11	8	5	2
Courant (mA)	73.6	86	95.6	93.9	96.1	97	100.6	101.2	102.5	108	111.6	115.9	117	116.5	120	118.3
Puissance (W)	1.51	1.72	1.86	1.78	1.75	1.74	1.76	1.72	1.69	1.73	1.67	1.5	1.29	0.93	0.6	0.24

*Illustration 5: Mesures lors du test du panneau solaire*

Pour mieux comprendre les résultats nous avons décidés de les exprimer sous forme d'une courbe qui est présente ci-dessous (Illustration 6).

### Caractéristique U/I du panneau solaire



*Illustration 6: Caractéristique tension/courant du panneau solaire*

Sur cette caractéristique tension courant nous pouvons constater que plus le courant est élevé dans le panneau solaire, plus la tension fournie par celui-ci est faible. Nous savons également que le rendement du panneau solaire est assez faible. Mais cela reste une source d'énergie infinie donc, une faible quantité de quelque chose d'infini est déjà mieux que rien ! Pendant la durée des tests nous avons constaté une baisse de tension dû à une exposition trop longue au soleil. Les cellules photovoltaïques perdent de leurs efficacité lors qu'elles chauffent.

Nous avons également réalisé un test en ombrageant une petite partie du panneau solaire ce qui a eu comme conséquence de faire chuter (à un niveau faible) la tension aux bornes du panneau solaire.

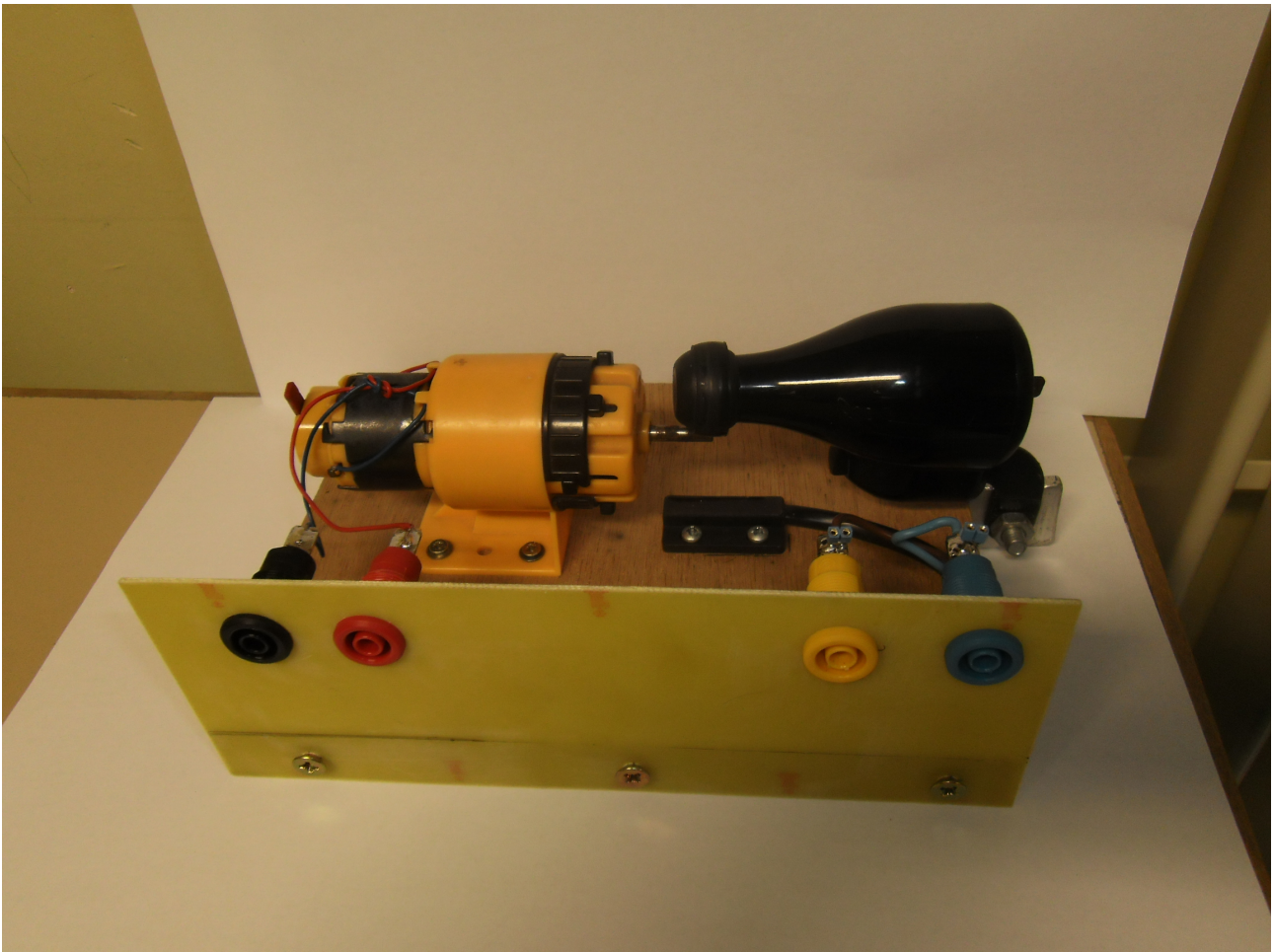
Voici une photo du panneau solaire mis en œuvre



*Illustration 7: Panneau solaire ASI-F 2/12  
utilisé pour le projet*

### 2.1.2. Étude de la dynamo

Dans notre projet nous devons utiliser une dynamo pour produire de l'énergie. Celle-ci est destinée à la batterie 9V. Pour tester cette dynamo, nous avons réalisé un banc de test utilisant un moteur à courant continu qui va simuler la roue de vélo.



*Illustration 8: banc de test pour la dynamo*

A l'aide d'un oscilloscope, nous avons réalisé une mesure de la tension aux bornes de la dynamo.

Nous avons pu constater que cette tension est quasiment sinusoïdale. Elle possède une fréquence de  $f=34\text{Hz}$ , une amplitude de 6V et une valeur efficace de 1,7V. Ces informations seront importantes pour la réalisation de la carte électronique.

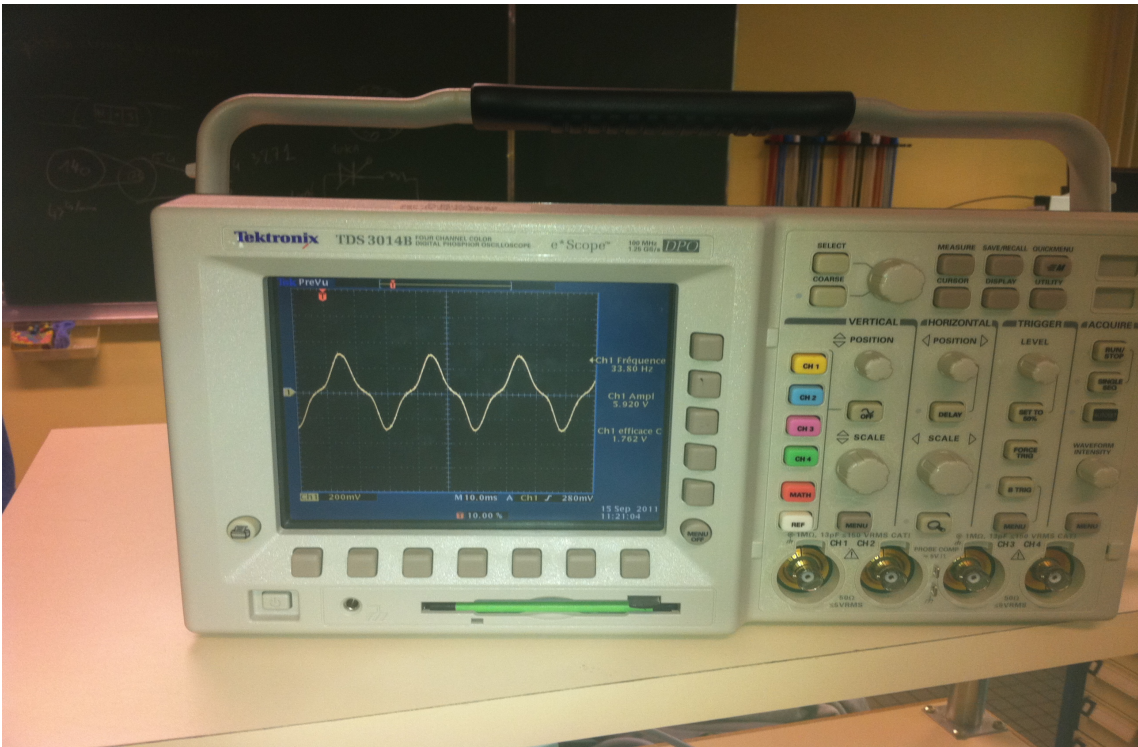


Illustration 9: Mesure de la tension au bornes de la dynamo

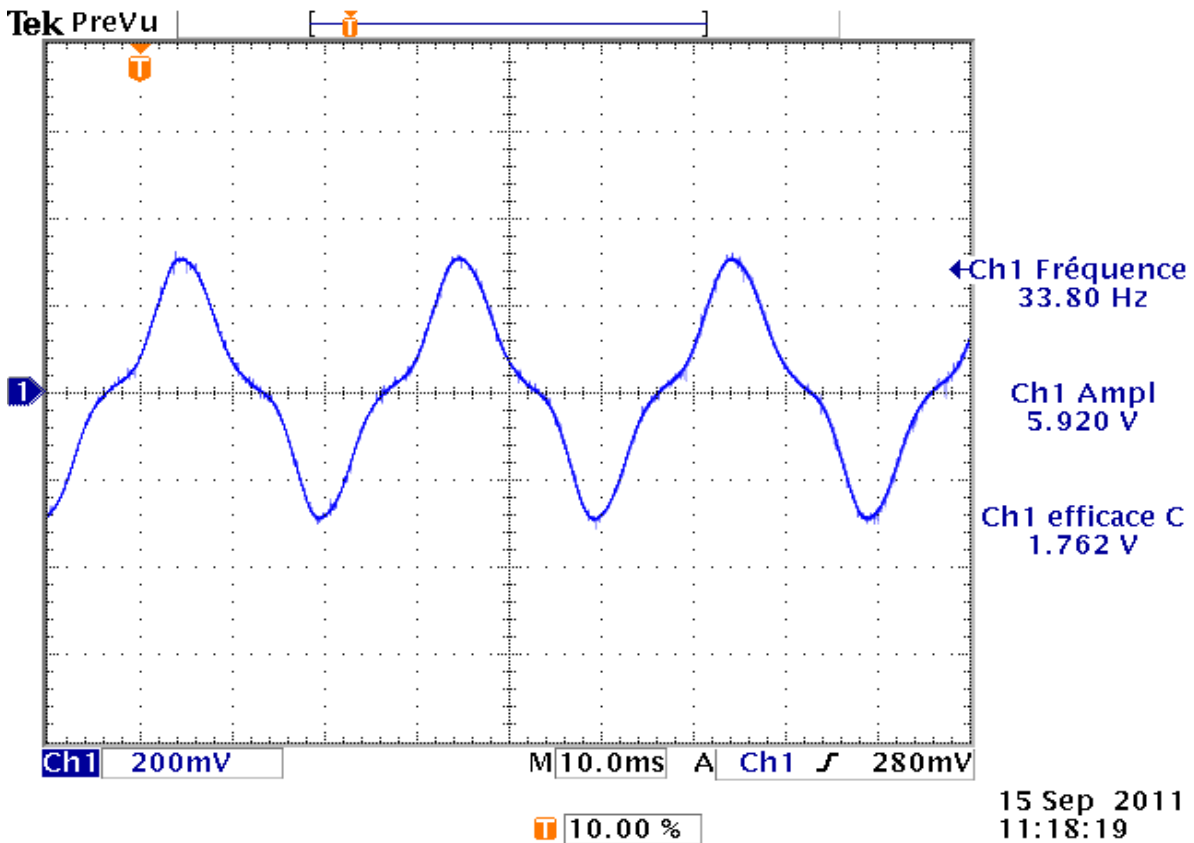


Illustration 10: Copie d'écran de l'oscilloscope

### 2.1.3. Étude de la batterie de 9V

Pour l'alimentation de l'ATmega8535 ainsi que celle de l'éclairage du vélo, nous avons choisi d'utiliser une batterie de 9V et de 1800mAH.



*Illustration 11: Batterie utilisée pour ce projet*

Pour connaître la capacité de la batterie à fournir de la tension et l'estimation du temps de décharge de celle-ci nous avons effectué différents tests. Dans un premier temps, nous avons étudié la datasheet<sup>1</sup> de cette batterie pour connaître le seuil minimum qu'elle pouvait accepter pour pouvoir la recharger correctement. Nous avons déterminé que ce seuil se situe vers 8,4V.

Ensuite nous avons réalisé un test qui consiste à mesurer le temps que met la batterie à atteindre le seuil de décharge. Pour ce faire nous avons mis aux bornes de la batterie une charge pour la décharger, puis mis en série un ampèremètre et enfin un voltmètre en parallèle.

---

<sup>1</sup> Données fournies par le constructeur

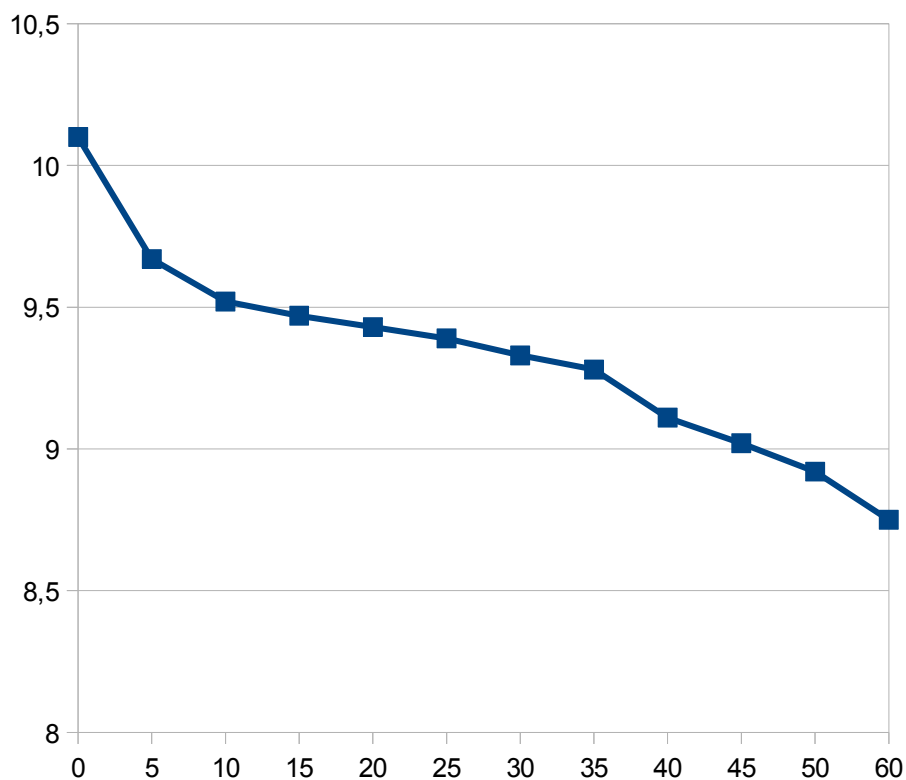
Voici les résultats obtenus:

temps(min)	0	5	10	15	20	25	30	35	40	45	50	60
U(V)	10.1	9.67	9.52	9.47	9.43	9.39	9.33	9.28	9.11	9.02	8.92	8.75
I(A)	1.08	1.04	1.02	1.02	1.01	1.01	1	1	0.98	0.97	0.96	0.94

*Illustration 12: Mesures du test pour la batterie*

Nous avons ensuite mis ces résultats sous formes de tableau

Tension U (V) en fonction du Temps (min)



*Illustration 13: Courbe exprimant la tension en fonction du temps dans la batterie*

Nous pouvons constater sur cette courbe que la chute de tension dans la batterie est plus importante les dix premières minutes alors que par la suite la chute de tension devient constante. Nous avons estimé que la batterie met 1h15 à atteindre son seuil de décharge. Nous avons également constatés que la tension de la batterie varie entre 10,1V et 8.4V. Ces informations vont nous aider à réaliser la carte électronique.

## 2.2. Réalisation de la carte électronique

Après avoir étudié les différentes sources d'énergies nous allons maintenant définir les différents modules que doit comporter la carte électronique. Chaque source d'énergie possède des caractéristiques différentes. De plus chacune d'entre elle doit remplir la même fonction : elles doivent recharger la batterie.

### 2.2.1. Définition des besoins de la carte électronique

Pour réaliser toutes cartes électronique, nous devons savoir, dans un premier temps, ce que doit réaliser cette carte. Dans notre projet, la programmation de l'ATmega8535 va être réaliser sur une carte indépendante qui est fournie par l'Université. Notre carte électronique va donc servir d'intermédiaire entre le microcontrôleur et les différentes sources de tension. Notre carte comportera un interrupteur de marche/arrêt en fonction du niveau de tension dans la batterie, un buzzer qui servira d'alarme pour prévenir l'utilisateur que la tension de la batterie va devenir bientôt insuffisante. Un port de communication pour pouvoir communiquer avec l'ATmega. Elle comportera également différents hacheurs de types Buck et Boost nous allons voir cela dans les paragraphes suivants.

Pour chaque source, nous devons réaliser une adaptation en tension pour qu'elle puisse être acceptée par la batterie dans un premier temps, puis par la carte de programmation dans un second temps. Ce qui veut dire que nous devons réaliser des hacheurs abaisseurs ou éleveurs. Mais cela dépend de la tension fournit par les sources d'énergies.

Dans le cas de la dynamo, nous devons d'abord réaliser un pont redresseur pour transformer la tension alternative en tension continue. Nous devons également réaliser un hacheur éleveur pour élever la tension à 10V afin de l'envoyer sur la batterie.

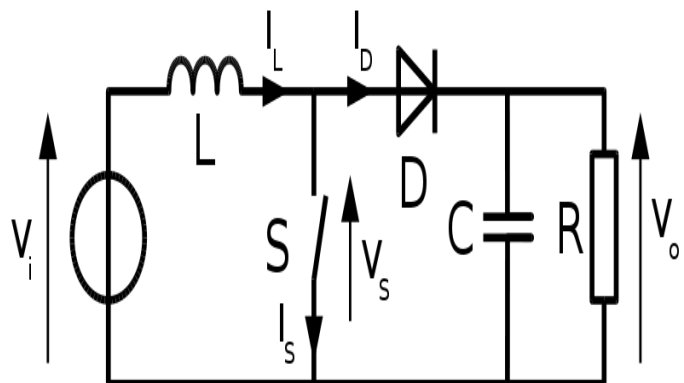


Illustration 14: Hacheur de type Boost (éleveur)

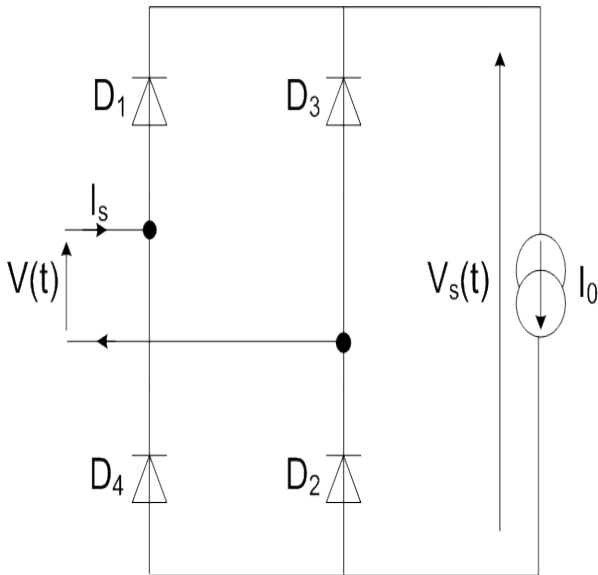


Illustration 15: Pont redresseur double

alternance

Pour le panneau solaire, nous devons réaliser un hacheur de type Buck (hacheur abaisseur) pour abaisser la tension de sortie du panneau qui peut monter jusqu'à 20V. Nous devons abaisser cette tension jusqu'à 10V.

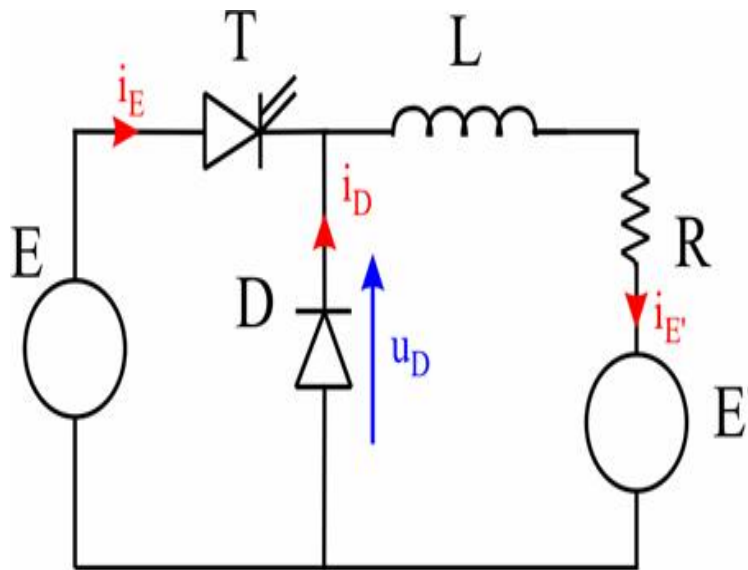
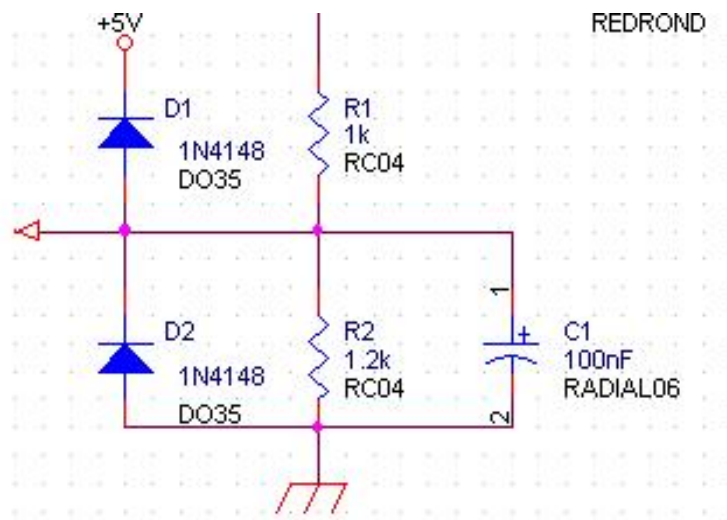


Illustration 16: Hacheur Buck (abaisseur)

Pour gérer les différentes sources de tension nous utilisons donc un ATmega8535 pour ce faire nous devons réaliser des mesures de tension et réaliser des autorisations. Nous devons réduire les tensions qui proviennent des sources d'énergies à 5V car l'ATmega ne supporte aux bornes de ses entrées analogiques que 5V. Pour y parvenir nous devons donc réaliser des ponts diviseurs de tension comme ci-dessous (Illustration 17).





*Illustration 17: Pont diviseur de tension*

Ici les diodes D1 et D2 servent à limiter la tension vers l'ATmega8535, car si la tension dépasse 5V, elle sera redirigée dans les diodes.

# Schéma explicatif des fonctions de la carte électronique

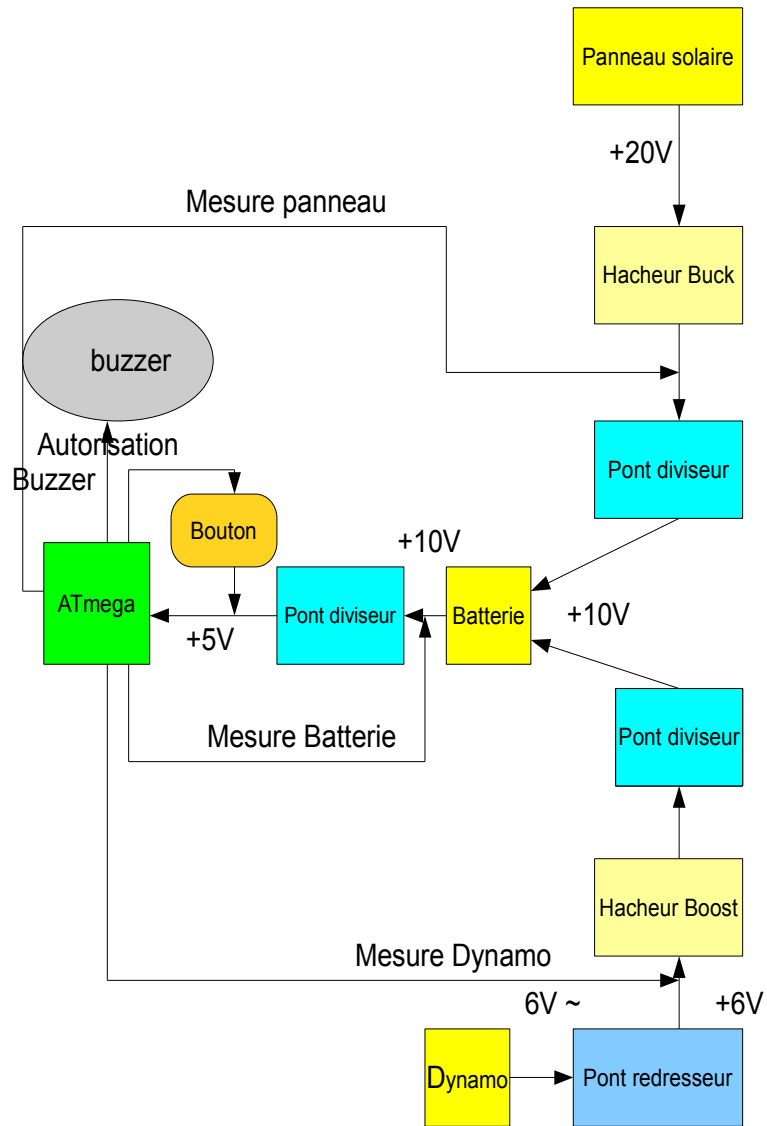


Illustration 18: Schéma explicatif du montage à réaliser

## 2.2.2. Réalisation du schéma électrique sous Orcad Capture

Une fois les tests préliminaires terminés et avoir vu les tâches que doit effectuer la carte électronique, nous allons aborder la réalisation du schéma électrique sous Orcad Capture. Pour réaliser cette carte, nous avons suivi un cours sur l'utilisation de Orcad Capture et Orcad Layout Plus. Ce cours a été fait par Thierry LEQUEU.

Partie alimentation :

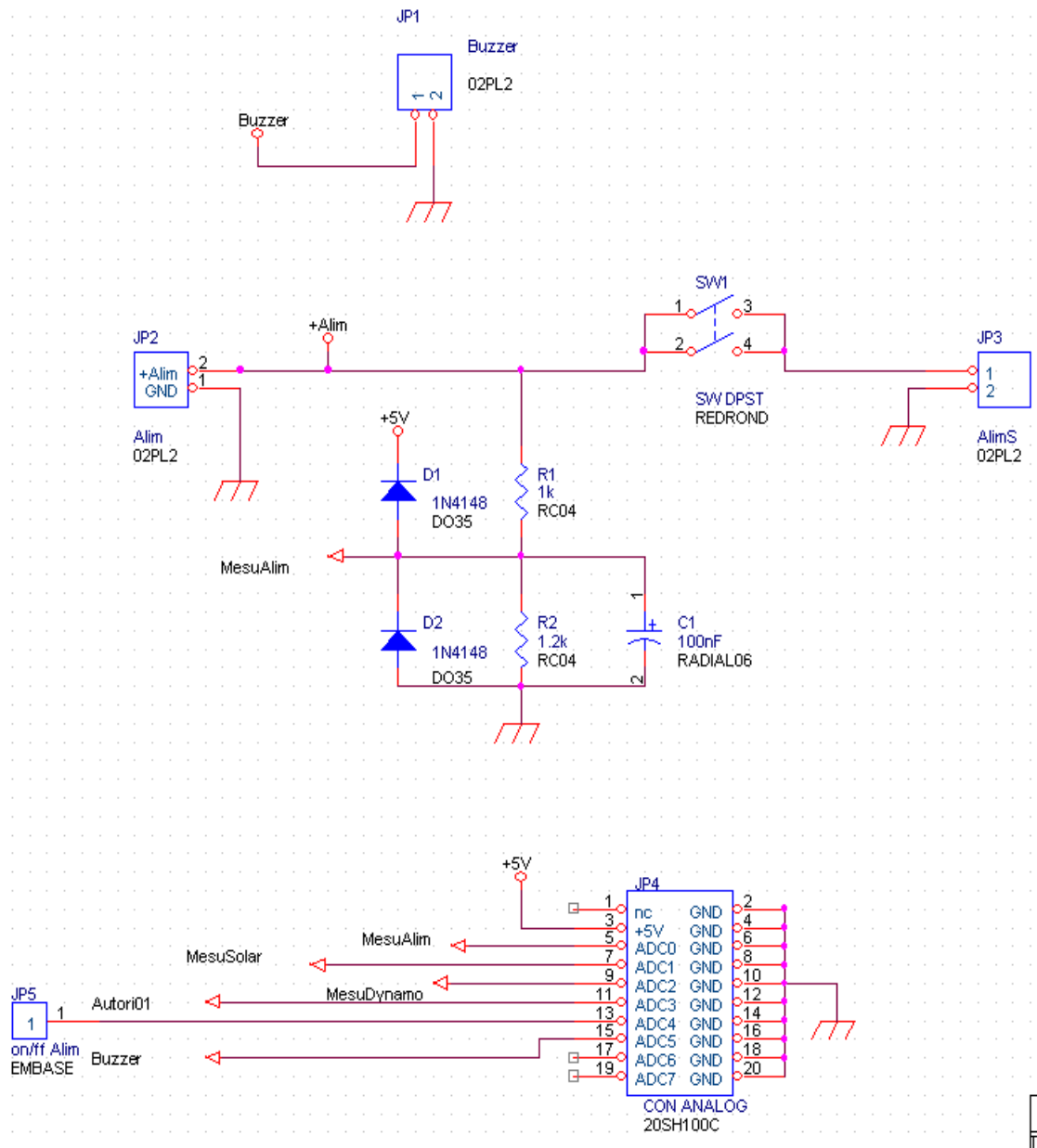


Illustration 19: Partie alimentation de la carte

Ci-dessus (Illustration 19), nous pouvons voir en bas le connecteur de communication avec l'ATmega8535. D1 et D2 servent à limiter la tension aux bornes de ADC0 pour pas qu'elle dépasse 5V. R1 et R2 ainsi que C1 forment le diviseur de tension de la tension de la batterie. SW1 est le bouton poussoir servant à réaliser l'auto-maintien sur l'ATmega. JP1, JP2, JP3 sont des connecteurs.

Calcul pour les résistances R1 et R2

$$(U_2 \cdot R_1) + (U_2 \cdot R_2) = U_1 R_2$$

$$(U_2 \cdot R_1) = (U_1 \cdot R_2) - (U_2 \cdot R_2)$$

$$(U_2 \cdot R_1) = (U_1 - U_2) \cdot R_2$$

$$\frac{(U_2 \cdot R_1)}{(U_1 - U_2)} = R_2 \quad \text{avec } R_1 = 1\text{k}\Omega \text{ et } U_2 = 5\text{V} \text{ et } U_1 = 9\text{V}$$

$$R_2 = \frac{5 \cdot 10^3}{(9 - 5)}$$

$$R_2 = 1,2\text{k}\Omega$$

Pour la dynamo (voir page suivante)

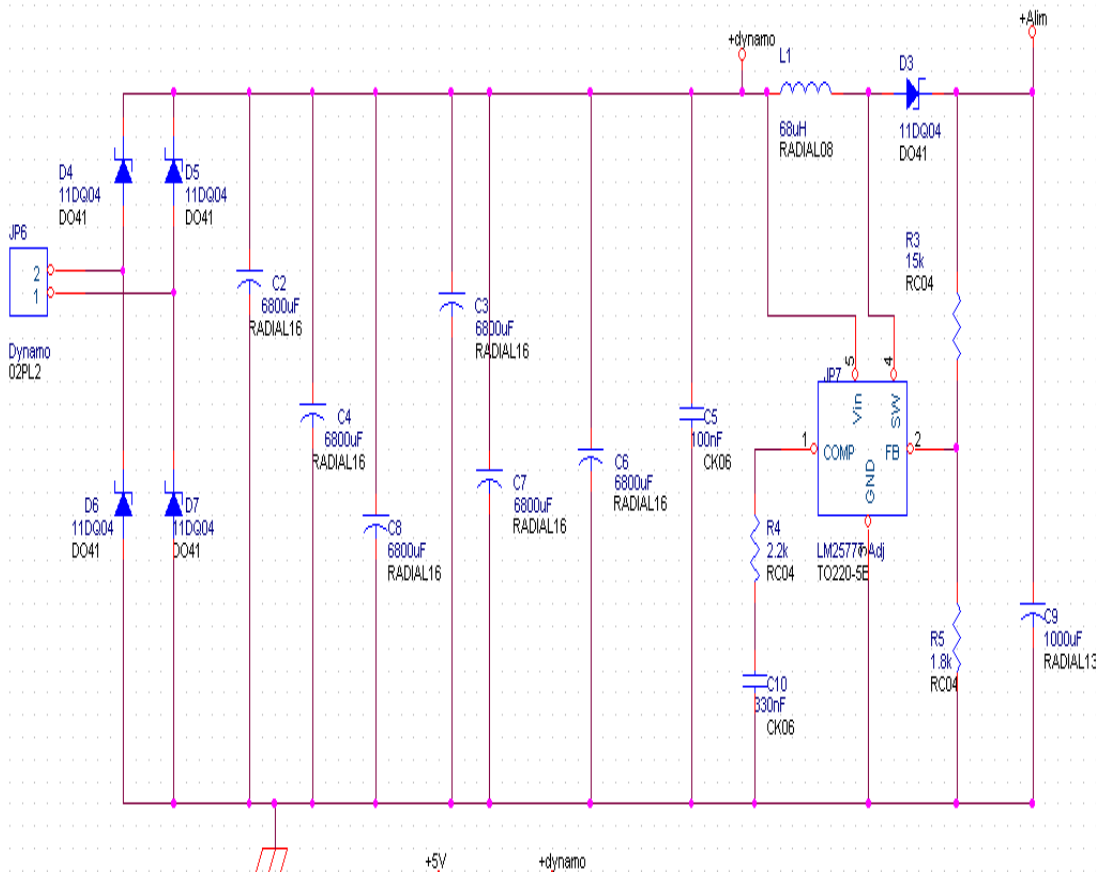


Illustration 20: Partie 1 de la dynamo

Ci-dessus (Illustration 20), nous pouvons distinguer les quatre diodes de redressement c'est-à-dire celles qui forment le pont diviseur (D4, D5, D6, D7). Les six condensateurs de 6800µF (C2, C3, C4, C5, C6, C7) sont les condensateurs de filtrage pour la tension sinusoïdale de la dynamo, ils « lissent » la tension, ils ont été dimensionnés par Louise Prodhomme lorsqu'elle travaillait sur le

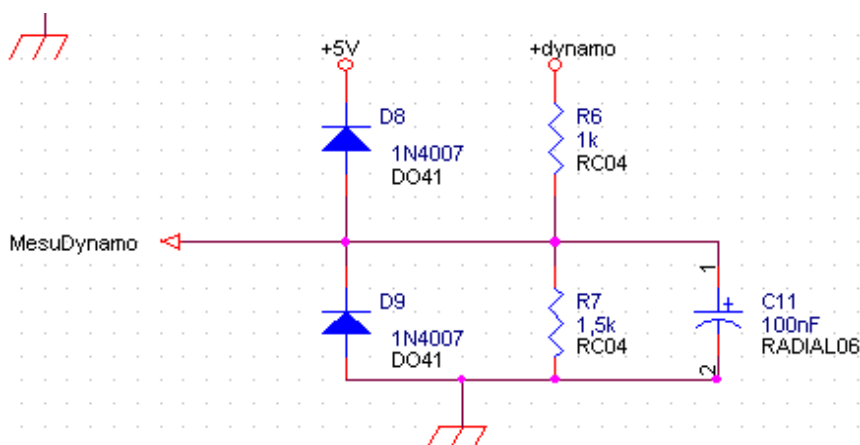


Illustration 21: Partie 2 de la dynamo

projet. Les autres composants sont les coeux qui forment le hacheur Boost (élevateur).

Sur l'image précédente (Illustration 21), on montre le pont diviseur que nous avons dû mettre en place pour mesurer la tension aux bornes de la dynamo. Il est constitué des résistances R6 et R7.

Calcul de R6 et de R7

$$(U_2 \cdot R_6) + (U_2 \cdot R_7) = U_1 R_7$$

$$(U_2 \cdot R_6) = (U_1 \cdot R_7) - (U_2 \cdot R_7)$$

$$(U_2 \cdot R_6) = (U_1 - U_2) \cdot R_7$$

$$\frac{(U_2 \cdot R_6)}{(U_1 - U_2)} = R_7$$

avec  $R_6 = 1\text{k}\Omega$  et  $U_2 = 5\text{V}$  et  $U_1 = 8,5\text{V}$  car on a  $6 \cdot \sqrt{2}$  aux bornes de la dynamo.

$$R_2 = \frac{5 \cdot 10^3}{(8,5 - 5)}$$

Donc  $R_2 = 1428,5\Omega = 1,5\text{k}\Omega$

Pour le panneau solaire (voir page suivante)

LM2574N-5.0      11DQ06  
 Vin = 7-40 V      SCHOTTKY  
 Vout = 5.0V      VRRM = 60 V  
 Iout = 0.5 A      IF(AV) = 1 A

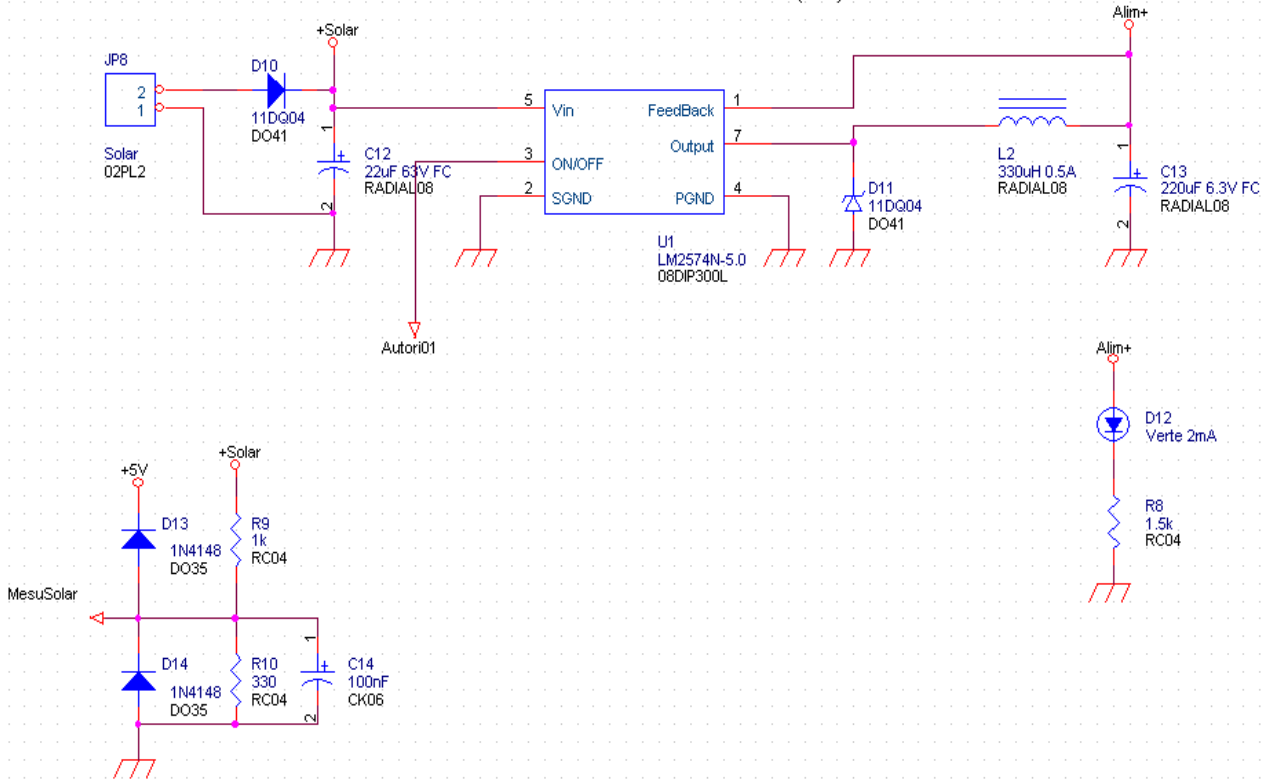


Illustration 22: partie du panneau solaire

En haut de l'illustration 22, nous pouvons voir l'adaptation en tension du panneau solaire. Les composants qui figurent sur ce montage sont les éléments qui composent le hacheur Buck (abaisseur). Le LM2574N est un régulateur en tension, il régule la tension jusqu'à 5V. R8 a été calculé pour appeler le courant nécessaire au fonctionnement de la diode D12.

Calcul de R8:

$$U = R \cdot I$$

$$I = \frac{U}{R} \quad \text{donc} \quad R = \frac{U}{I} \quad \text{avec } U=5V \text{ et } I=2mA$$

$$R = \frac{5}{2 \cdot 10^{-3}} \quad R8 = 2,5k\Omega$$

En bas à gauche de l'image, se situe le pont diviseur mis en œuvre pour le panneau solaire.

Calcul de R9 et R10

$$(U2 \cdot R9) + (U2 \cdot R10) = U1R10$$

$$(U2 \cdot R9) = (U1 \cdot R10) - (U2 \cdot R10)$$

$$(U2 \cdot R9) = (U1 - U2) \cdot R10$$

$$\frac{(U2 \cdot R9)}{(U1 - U2)} = R10 \quad \text{avec } R9=1\text{k}\Omega \text{ et } U2=5\text{V} \text{ et } U1=20\text{V}$$

$$R10 = \frac{5 \cdot 10^3}{(20 - 5)}$$

$$R10 = 333,3\Omega$$

### ***2.2.3. Réalisation du typon de la carte électronique sous Orcad layout plus***

Après avoir fait le schéma électrique sous Orcad Capture nous allons réaliser le typon sous Orcad Layout plus. Pour se faire, nous devons savoir quels composants nous allons utiliser car ce logiciel utilise les empreintes physiques pour réaliser le typon. De plus, nous avons dû rechercher les bibliothèques de composants créés par Thierry LEQUEU pour mener à bien ce projet.

Pour réaliser ce typon nous avons dû, dans un premier temps, affecter à tous les composants une empreinte physique puis créer une netlist. Il a ensuite fallu charger le schéma électrique sous Layout Plus puis réaliser le routage.

Sur la page suivante, la netlist réalisée pour ce projet, elle montre le nom des composants, leur valeurs, leur quantité, leur empreinte physique. Mais nous pouvons également réaliser une colonne prix pour estimer le coût de notre carte électronique.



Adaptation en tension du panneau solaire 20V->5V Revised: Thursday, October 13, 2011					
Auteur: Thibault Delaroche et Vincent Mollinier		Revision:			
Bill Of Materials		Page1			
N°	Quantité	Référence	Désignation	Empreinte	Prix en euros
1	2	C11,C1	100nF	RADIAL06	0.85
2	6	C2,C3,C4,C6,C7,C8	6800uF	RADIAL16	9.8
3	2	C14,C5	100nF	CK06	2.07
4	1	C9	1000uF	RADIAL13	0.89
5	1	C10	330nF	CK06	1.03
6	1	C12	22uF 63V FC	RADIAL08	0.03
7	1	C13	220uF 6.3V FC	RADIAL08	0.03
8	4	D1,D2,D13,D14	1N4148	DO35	0.43
9	7	D3,D4,D5,D6,D7,D10,D11	11DQ04	DO41	3.04
10	2	D9,D8	1N4007	DO41	0.09
11	1	D12	Verte 2mA	LED03	0.17
12	1	JP1	Buzzer	02PL2	5
13	1	JP2	Alim	02PL2	0.3
14	1	JP3	AlimS	02PL2	0.3
15	1	JP4	CON ANALOG	20SH100C	0.3
16	1	JP5	on/ff Alim	EMBASE	0.3
17	1	JP6	Dynamo	02PL2	0.3
18	1	JP7	LM2577T Adj	TO220-5B	0.3
19	1	JP8	Solar	02PL2	0.3
20	1	L1	68uH	RADIAL08	1.7
21	1	L2	330uH 0.5A	RADIAL08	2.4
22	3	R1,R6,R9	1k	RC04	0.18
23	1	R2	1.2k	RC04	0.06
24	1	R3	15k	RC04	0.06
25	1	R4	2.2k	RC04	0.06
26	1	R5	1.8k	RC04	0.06
27	1	R7	1,5k	RC04	0.06
28	1	R8	1.5k	RC04	0.06
29	1	R10		330 RC04	0.06
30	1	SW1	SW DPST	REDROND	
31	1	U1	LM2574N-5.0	08DIP300L	2.42
Total					32.66

Illustration 23: Liste des composants, plus prix

Après avoir réalisé le routage du typon nous obtenons l'illustration 24

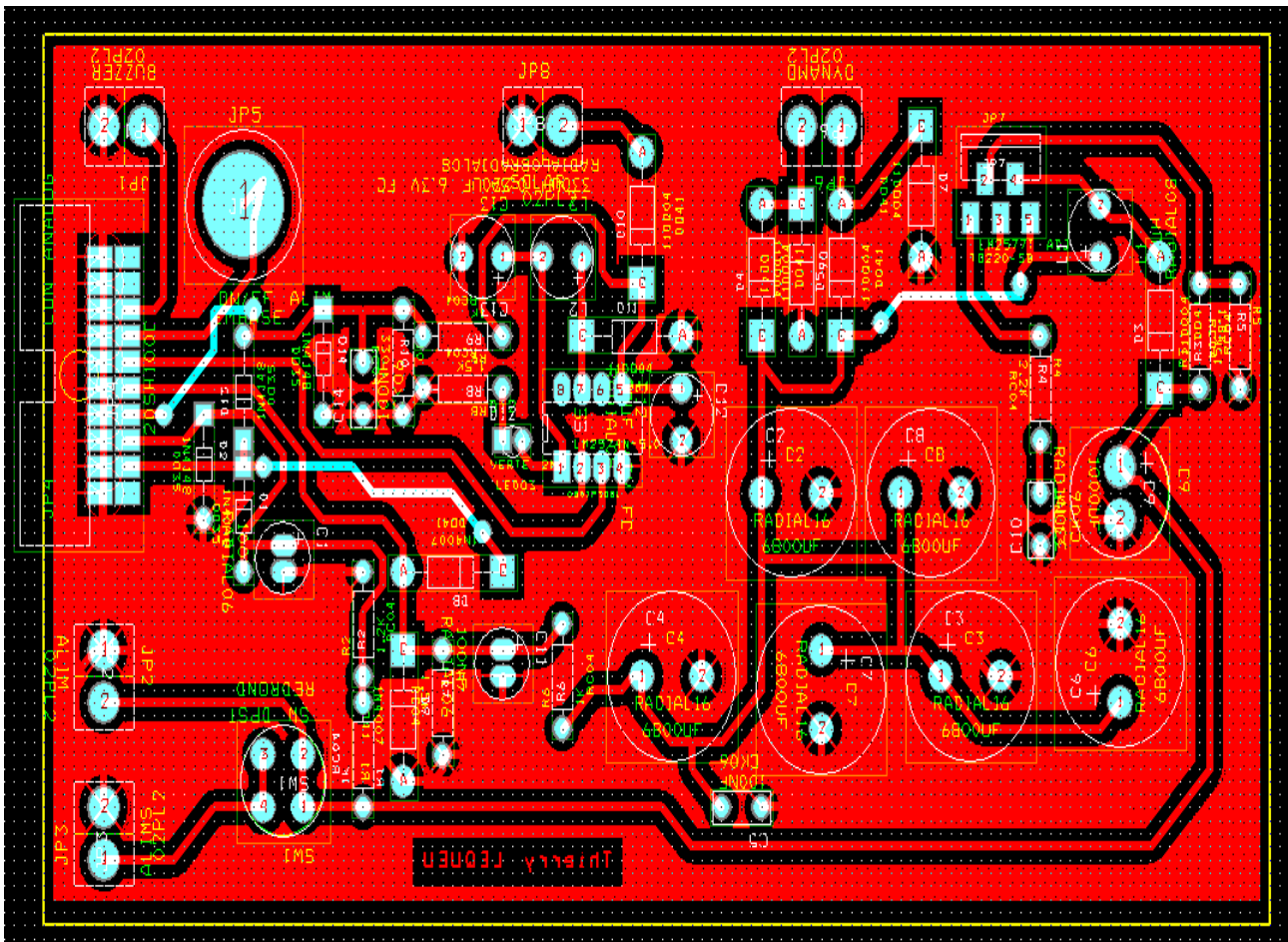


Illustration 24: Typon de la carte

Après avoir créé cette carte, nous avons procédé à une série de tests qui consiste à savoir si il n'y avait pas de court-circuit ou autre défaillance. Sur cette carte, nous avons rencontré un problème majeur car le bouton poussoir ne fonctionnait pas correctement. C'est pour cela que nous avons dû rajouter deux transistors ainsi que deux résistances, pour pouvoir réaliser cette fonction. Après modification l'ensemble du système fonctionne parfaitement.

Maintenant que nous avons abordé le pôle électronique du projet nous allons aborder la partie informatique.

### 3. Pôle informatique du projet

#### 3.1. Présentation de l'ATmega8535

L'ATmega est un micro contrôleur développé par Atmel. Il dispose de 40 broches dont 8 sont déjà assignées comme l'alimentation (broche 10), la masse (broche 11 et 31) ou encore le reset (broche 9). Les 32 autres pattes du composant sont disposées sur 4 ports A, B, C et D. On a donc 8 pattes par port, chacune pouvant être configurée en entrée ou en sortie.

Chaque port a ses spécificités. On peut utiliser le port A comme convertisseur analogique numérique (ADC en anglais) par exemple.

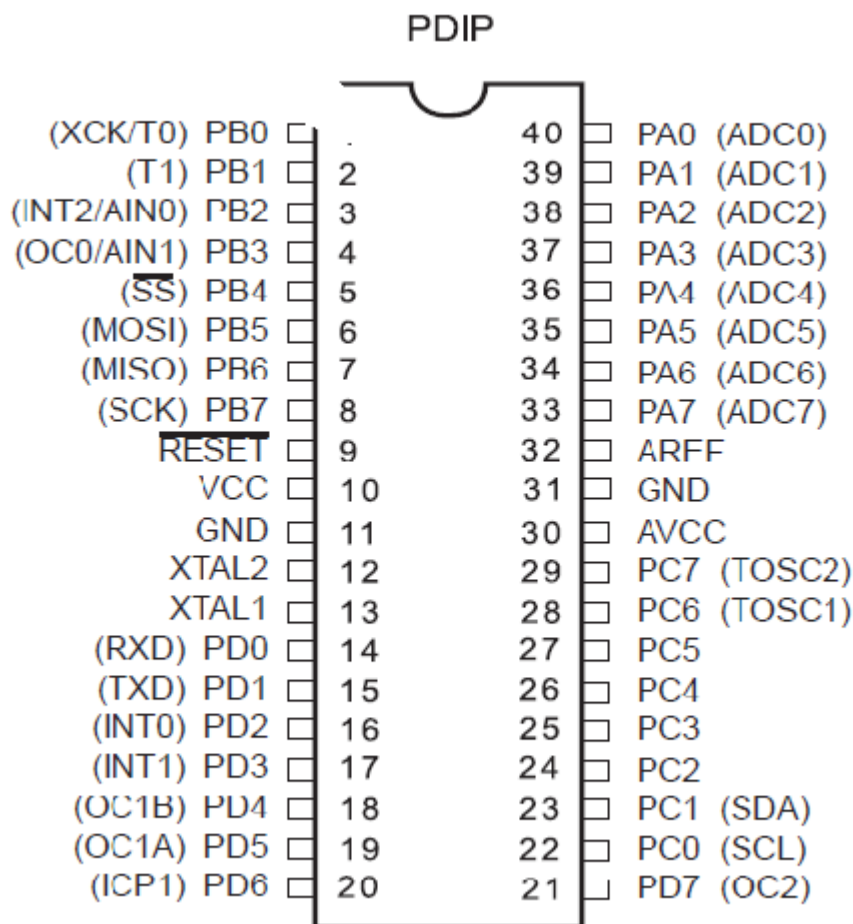


Illustration 25: Schéma global de l'ATmega 8535

Pour notre projet, nous avons besoin d'un écran LCD et de la fonction ADC<sup>2</sup>. Nous avons donc configuré le port A de façon à obtenir cette conversion et comme nous utilisons la carte de

<sup>2</sup> Analog Digital Converter : permet de convertir une tension analogique en une tension numérique

programmation de l'IUT, l'écran LCD est déjà relié au port C du microcontrôleur (voir annexe page 69).

### 3.2. Explications du programme

L'ATmega8535 doit réaliser différentes fonctions pour permettre le bon fonctionnement du projet.

Il doit autoriser la charge de la batterie lorsque sa tension est inférieure à 8,4V. De plus, il devra avertir l'utilisateur que la tension devient faible via un buzzer (haut parleur).

Lorsque la tension de la batterie est inférieure à 8V, le micro-contrôleur doit couper son alimentation et celle de l'écran LCD pour ne plus consommer de courant et permettre à la batterie de se recharger. Le programme doit donc commencer par mesurer la tension aux bornes de la batterie. Si celle-ci est trop faible, il se coupe instantanément. Sinon, le programme démarre normalement et mesure ensuite les deux autres tensions.

À chaque instant, l'utilisateur du système doit pouvoir consulter la tension de la batterie, de la dynamo et du panneau solaire grâce à un bouton poussoir. Au premier appui, il pourra lire la tension aux bornes de la batterie, au deuxième celle de la dynamo et au troisième celle du panneau solaire.

#### 3.2.1. Les entrées et sorties

```
// Entrées analogiques :
#define ADC_VREF_TYPE 0x00

#define MesBat 0      // Normalement entrée N°0
#define MesSol 1      // Normalement entrée N°1
#define MesDyn 2      // Normalement entrée N°2

// Entrées numériques :
#define BP1         PIND.0

// Sorties numériques :
#define PanToBat    PORTA.3
#define Alim        PORTA.4
#define Buzzer      PORTA.5
```

*Illustration 26: Définition des entrées et des sorties*

On utilise trois entrées analogiques qui correspondent à la conversion analogique numérique pour le panneau solaire, la batterie et la dynamo et une entrée numérique pour le bouton poussoir.

Nous avons également trois sorties permettant l'autorisation de la charge de la batterie par le panneau solaire, la coupure de l'alimentation et le buzzer.

### 3.2.2. Les variables

```
    }  
  
    // Declare your global variables here  
  
    void main(void)  
    {  
        // Declare your local variables here  
        int j=0;  
        float kbat, kpan, kdyn;  
        unsigned int ibat, ipan, idyn;  
        unsigned char TensionBat[20], TensionPan[20], TensionDyn[20];  
    }
```

Illustration 27: Déclaration des variables

Les variables, `kbat`, `kpan` et `kdyn` sont des variables images de la tension mesurée aux bornes de la batterie, du panneau solaire et de la dynamo. Ce sont les valeurs de ces variables qui sont affichés sur l'écran LCD. On doit afficher une valeur décimale, on utilise donc un **float**.

Les variables `ibat`, `ipan` et `idyn` sont les tensions images comprises entre 0 et 5 volts mesurées sur les entrées analogiques et ensuite converties en valeurs numériques afin de pouvoir les traiter dans le programme. On définit leur type en se reportant au prototype de la fonction qui permet de convertir la tension analogique en valeur numérique. On utilise donc des **unsigned int** bien que l'on puisse aussi utiliser un **float**. Les variables, `TensionDyn` et `TensionPan` sont des tableaux de chaînes de caractères dans lesquels on placera la tension à afficher. La variable « j » permet de gérer les appuis sur le bouton poussoir. A chaque appuie, on incrémente « j » à chaque appuie de façon à savoir s'il faut afficher la tension de la batterie, du panneau solaire ou de la dynamo.

### 3.2.3. Initialisation des ports

Après avoir configuré les entrées et sorties de l'ATmega8535, le programme initialise automatiquement les ports.

Comme on a 8 bits par port, Code Vision les convertis en valeur hexadécimale. Le bit est configuré en entrée s'il a la valeur 0 et en sortie s'il a la valeur 1.

```
// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=Out Func5=Out Func4=Out Func3=Out Func2=In Func1=In Func0=In
// State7=T State6=0 State5=0 State4=0 State3=0 State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x78;

// 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;
```

*Illustration 28: Initialisation de tout les ports*

### 3.2.4. Initialisation ADC et LCD

Le programme initialise automatiquement le convertisseur analogique numérique au démarrage du micro-contrôleur ainsi que l'écran LCD.

De plus, lorsque le programme démarre, il doit vérifier immédiatement si la tension de la batterie est suffisamment importante pour rester en route. On lit alors la valeur de la tension aux bornes de la batterie (ramené à une tension comprise entre 0 et 5 volts) puis on calcule la tension réelle.

```

// ADC initialization
// ADC Clock frequency: 125,000 kHz
// ADC Voltage Reference: AREF pin
// ADC High Speed Mode: Off
// ADC Auto Trigger Source: Free Running
ADMUX=ADC_VREF_TYPE;
ADCSRA=0x87;
SFIOR&=0x0F;

// LCD module initialization
lcd_init(16);
lcd_putsf("Bonjour");
lcd_gotoxy(0,0); //x,y

//Mesurer la tension de la batterie au démarrage
ibat = read_adc(MesBat);
kbat = (float)ibat*2*5/1024;
delay_ms(2000);

```

*Illustration 29: Initialisation ADC et de l'écran LCD*

### 3.2.5. Début de la boucle

La fonction « while(1) » permet de faire une boucle dans le programme pour que celui-ci ne s'arrête jamais. On commence la boucle en mesurant la tension présente aux bornes de la batterie, de la dynamo et du panneau solaire grâce à la fonction « read\_adc ».

```

while (1)
{
//TEST AFFICHAGE INCREMENTATION j
/* sprintf(TensionBat,"Bonjour - j = %d",j);
lcd_gotoxy(0,0); //x,y
lcd_puts(TensionBat); */

ibat = read_adc(MesBat); //mesure de la tension batterie
idyn = read_adc(MesDyn); //mesure tension dynamo
ipan = read_adc(MesSol); //mesure tension panneau solaire

```

*Illustration 30: Explication du programme*

### 3.2.6. Incrémentation de la variable « j »

Lors d'un appui sur le bouton poussoir BP1, on incrémente de 1 la variable « j » et on attend 200 milliseconde, le temps que le bouton soit relâché et que la variable ne change pas plusieurs fois de valeurs lors d'un seul appui. On ne désire mesurer que 3 tensions différentes, il faut donc que « j » varie de 1 à 3. Lorsque « j » est supérieur ou égal à 4, on décide donc de lui redonner la valeur 1.

```
//Incrémentation de j lors d'un appuie sur BP
if(BP1==0)
{
    j++;                // j = j++ NE FAIT RIEN!!!!!!! ATTENTION!!!!!!
    delay_ms(200);
}
if(j>=4)
    j=1;
```

*Illustration 31: Incrémentation de « j »*

### 3.2.7. traitement de la conversion en tension réelle et affichage

Lorsque « j » vaut 1, on décide de traiter la tension aux bornes de la batterie. On commence par effacer tout ce qui pouvait être affiché sur l'écran LCD puis on écrit le mot « batterie » pour savoir quelle tension nous allons afficher. On traite ensuite la mesure faite au début de la boucle: on calcule la tension réelle aux bornes de la batterie que l'on va stocker dans le tableau « TensionBat ». On limitera la mesure aux trois premières décimales, ce qui est suffisant dans notre cas. On se place ensuite sur la deuxième ligne de l'écran LCD et on affiche la valeur de « TensionBat » toutes les 50ms afin d'éviter que les derniers chiffres de la mesure ne changent constamment.

On réitère la même opération lorsque « j » vaut 2 pour afficher la tension aux bornes du panneau et lorsque « j » vaut 3 pour afficher la tension au bornes de la dynamo.



```

//MESURE TENSION BATTERIE
if(j==1)
{
    lcd_clear();
    lcd_gotoxy(0,0);
    lcd_putsf("batterie");

    kbat = (float)ibat*2*5/1024; //valeur de la tension réelle, calcul forcé en float
    sprintf(TensionBat,"%4.3f",kbat);
    lcd_gotoxy(0,1);
    lcd_puts(TensionBat); //on affiche la valeur de k
    delay_ms(50);
}

//MESURE TENSION PANNEAU
if(j==2)
{
    lcd_clear();
    lcd_gotoxy(0,0);
    lcd_putsf("panneau ");

    kpan = (float)ipan*24/1024; //valeur de la tension réelle
    sprintf(TensionPan,"%4.3f",kpan);
    lcd_gotoxy(0,1);
    lcd_puts(TensionPan); //on affiche la valeur de k
    delay_ms(50);
}

//MESURE TENSION DYNAMO
if(j==3)
{
    lcd_clear();
    lcd_gotoxy(0,0);
    lcd_putsf("dynamo ");

    kdyn = (float)idyn*2*5/1024; //valeur de la tension réelle
    sprintf(TensionDyn,"%4.3f",kdyn);
    lcd_gotoxy(0,1);
    lcd_puts(TensionDyn); //on affiche la valeur de k
    delay_ms(50);
}

```

*Illustration 32: traitement des tensions*

### **3.2.8. Traitement des données**

On traite ensuite les données mesurées.

Si la tension aux bornes de la batterie est inférieure ou égale à 8,4 volts, on autorise le panneau solaire à recharger la batterie et on active un buzzer afin de prévenir l'utilisateur que l'autonomie du système est réduite. Si la tension aux bornes de la batterie est supérieure à 8,4 volts alors on ne la recharge pas et le buzzer ne s'enclenche pas.

Si la tension de batterie devient inférieure à 8 volts alors on coupe l'alimentation de l'ATmega et il faut alors recharger la batterie. Autrement, on maintient l'alimentation et le système fonctionne normalement.

```

if(kbat<=8.4)
{
    PanToBat = 1;           //recharger batterie
    Buzzer = 1;           //buzzer en marche pendant 2s
    delay_ms(500);
    Buzzer = 0;           // éteindre buzzer
}

else                       //sinon ne rien faire
{
    PanToBat = 0;
Buzzer = 0;
}

if(kbat<=8)
    Alim = 0;
else
    Alim = 1;

```

*Illustration 33: Traitement des données*

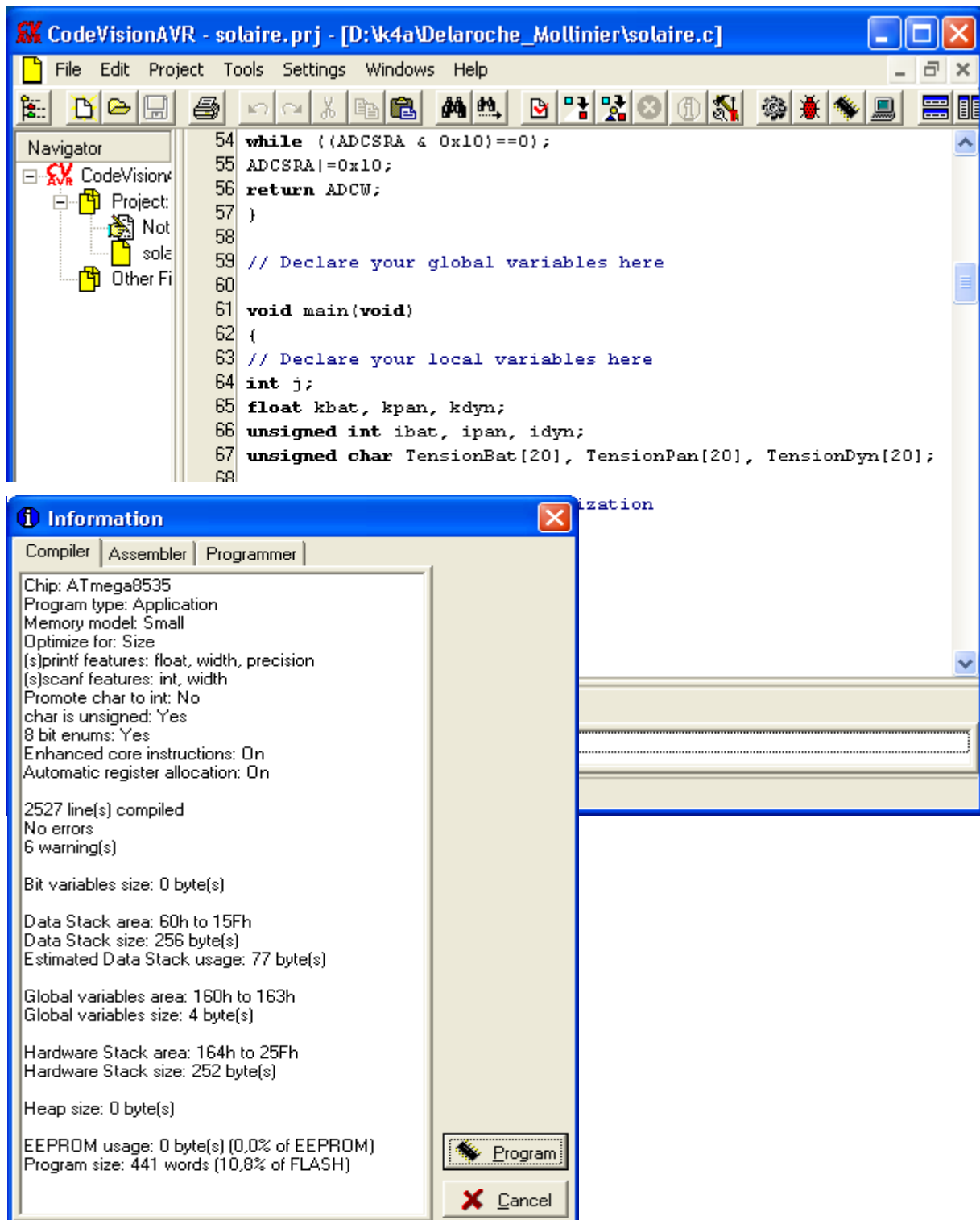
### **3.3. Remarques, problèmes rencontrés et utilisation de CodeVision**

#### **3.3.1. Différence de taille selon le type de variables**

Selon le type de variable utilisée, la mémoire utilisé dans le micro-contrôleur n'est pas la même. Ainsi un **float** nécessite plus de place qu'un **unsigned int**. Afin de gagner de la place dans la mémoire, on essaiera plutôt d'utiliser des variables qui nécessitent moins de mémoire.

Type	Size (Bits)	Range
bit	1	0 , 1
char	8	-128 to 127
unsigned char	8	0 to 255
signed char	8	-128 to 127
int	16	-32768 to 32767
short int	16	-32768 to 32767
unsigned int	16	0 to 65535
signed int	16	-32768 to 32767
long int	32	-2147483648 to 2147483647
unsigned long int	32	0 to 4294967295
signed long int	32	-2147483648 to 2147483647
float	32	$\pm 1.175e-38$ to $\pm 3.402e38$
double	32	$\pm 1.175e-38$ to $\pm 3.402e38$

*Illustration 34: Taille des variables*



*Illustration 35: Taille des variables*

En utilisant 3 **unsigned int** plutôt que des **float**, on utilise 77 bytes.

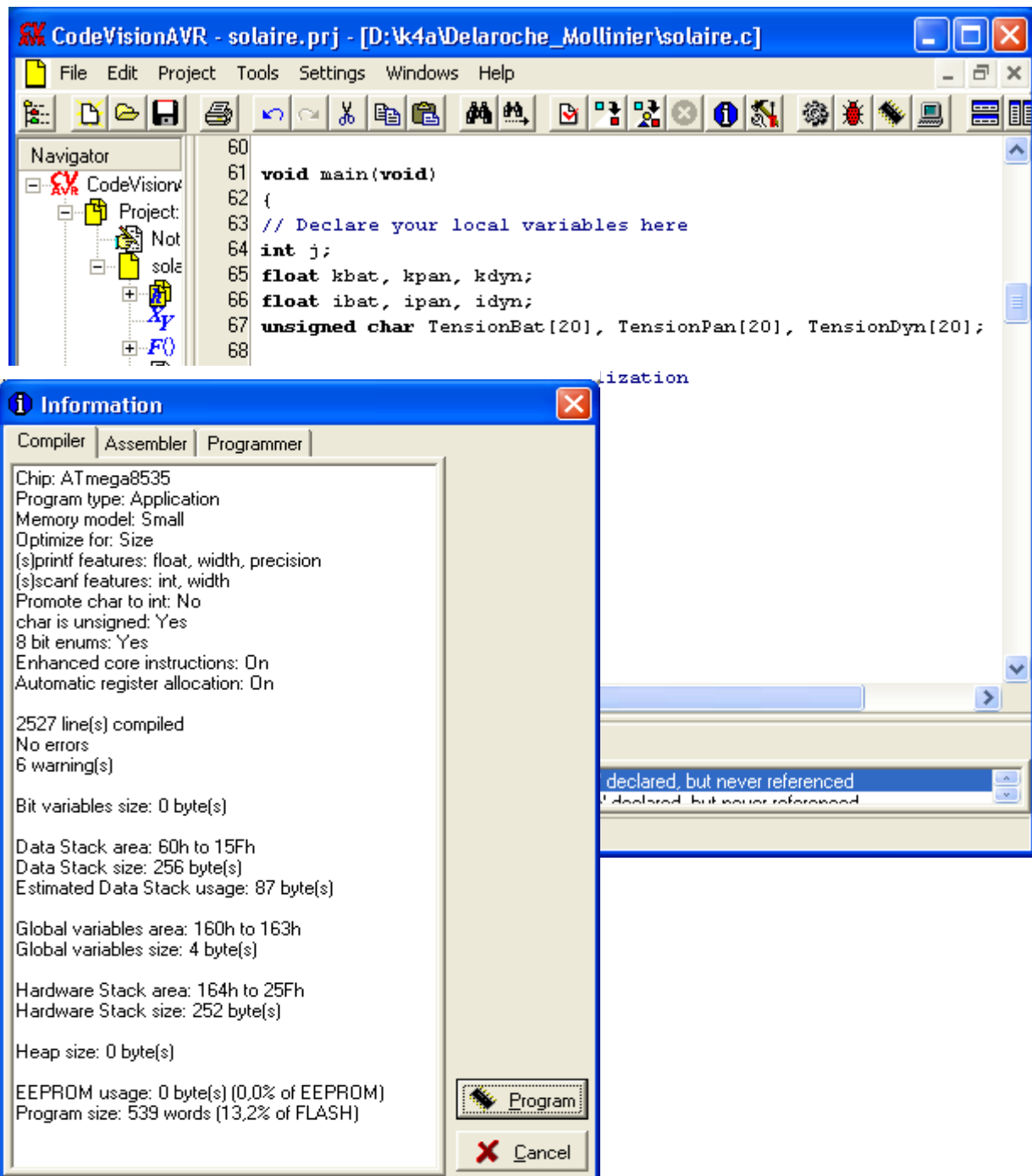


Illustration 36: Taille des variables bis

Le fait d'utiliser 3 **float** alors que l'on peut utiliser des **unsigned int**, on passe à 87 bytes utilisés. On perd ainsi 10 bytes.

A noter que le test a été fait avant que le programme ne soit terminé. La mémoire totale nécessaire n'est donc pas celle de l'image

### 3.3.2. Problèmes rencontrés

```
if(BP1==0)
{
    j++;           // j = j++ NE FAIT RIEN!!!!!! ATTENTION!!!!!!
    delay_ms(200);
}
if(j>=4)
    j=1;
```

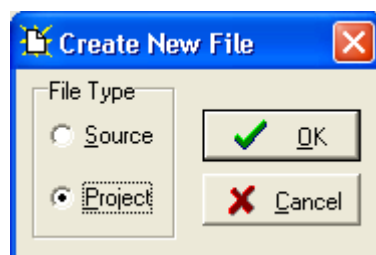
*Illustration 37: suite des problèmes*

Nous avons rencontré une erreur dans le programme à cause d'une faute d'inattention en voulant incrémenter « j ». Le plus logique est d'écrire « **j = j+1** » mais il existe des formes contractées pour écrire plus vite. Ainsi « **j = j+1** » peut aussi s'écrire « **j++** ». Par inadvertance, nous avons écrit « **j = j++** », ce qui ne veut plus rien dire.

Nous avons rencontré un autre problème pour le fonctionnement du convertisseur analogique numérique. Les valeurs affichées sur l'écran LCD étaient totalement fausses. Au départ, nous avons écrit les « #define » de ces entrées comme si elles étaient numériques. Comme ce sont des entrées numériques, elles sont forcément sur le port A, il suffit donc de noter le numéro de la patte compris entre 0 et 7.

### 3.3.3. Démarrage de code vision et configuration de l'ATmega8535

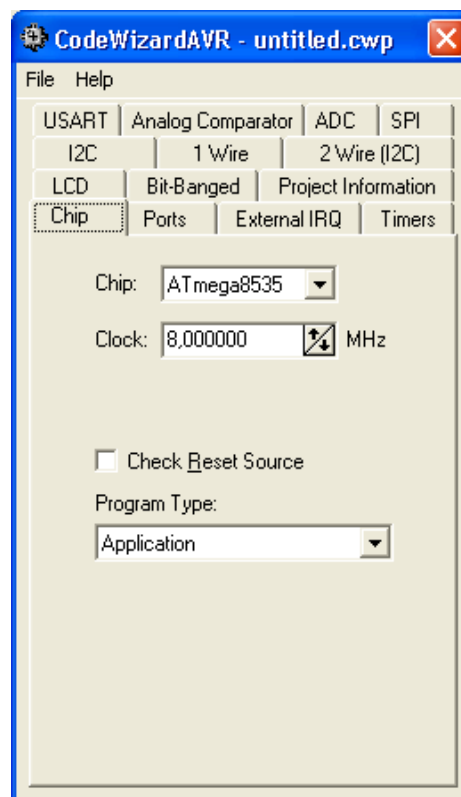
On commence par démarrer le logiciel CodeVision et on crée un nouveau projet afin de débiter la programmation.



*Illustration 38:*

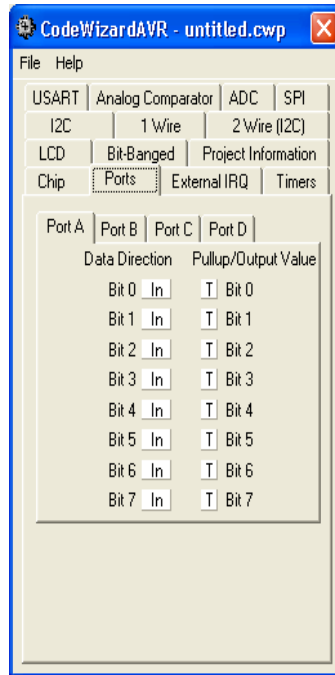
*Démarrage de CodeVision*

Il apparait alors une nouvelle fenêtre « CodeWizard AVR »



*Illustration 39: Fenêtre CodeWizrd*

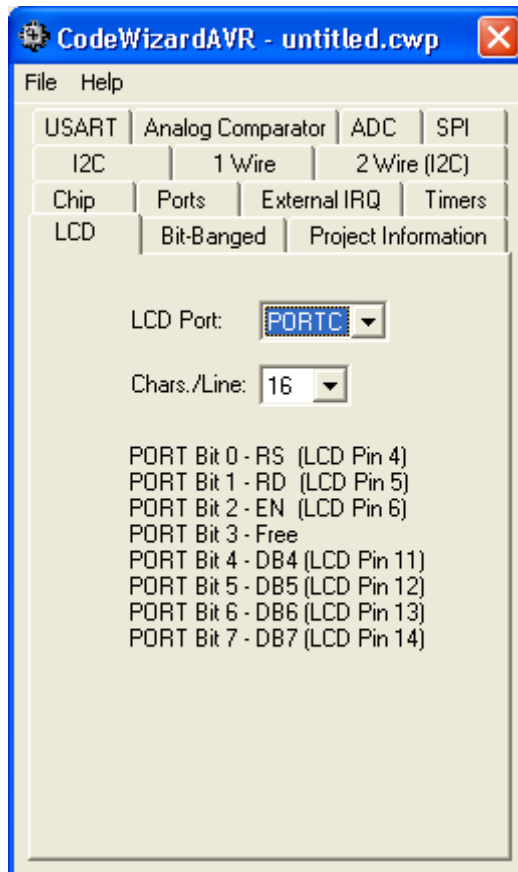
Dans « Chip », On règle la fréquence d'horloge de l'ATmega8535.



*Illustration 40: suite  
de l'explication du  
logiciel*

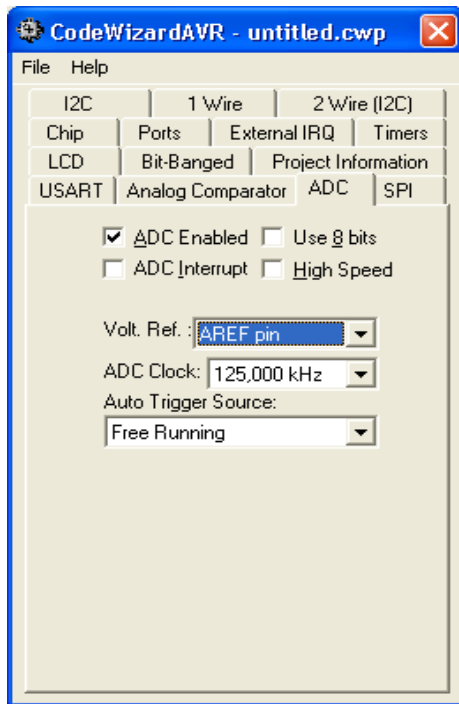
« Ports » permet de régler les broches de chaque port en entrée ou en sortie.



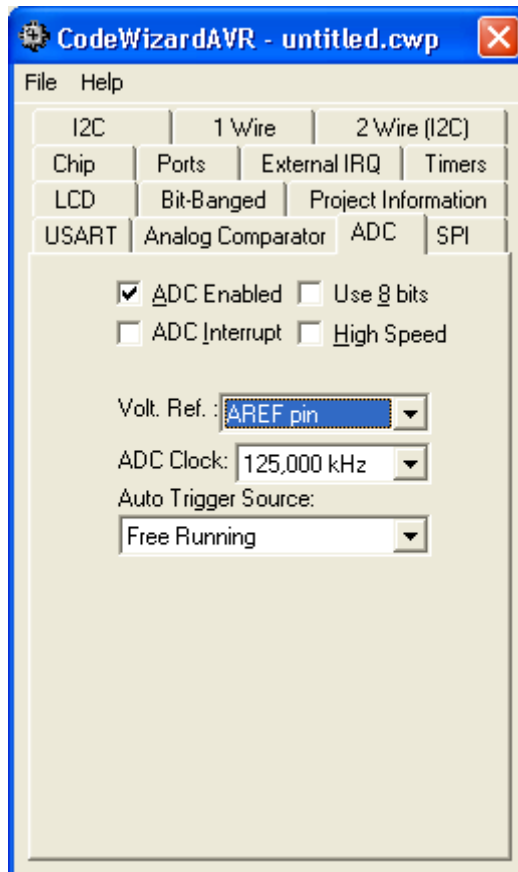


*Illustration 41: Onglet LCD*

L'onglet « LCD » permet de sélectionner sur quel port nous allons relier l'écran LCD. Dans notre cas, c'est le port C. A noter que ce sont des sorties pour l'ATmega8535 mais qu'il n'y a pas besoin de les configurer dans « Ports » une fois que l'on a réglé les paramètres dans « LCD ».



*Illustration 42: configuration de l'ATmega*



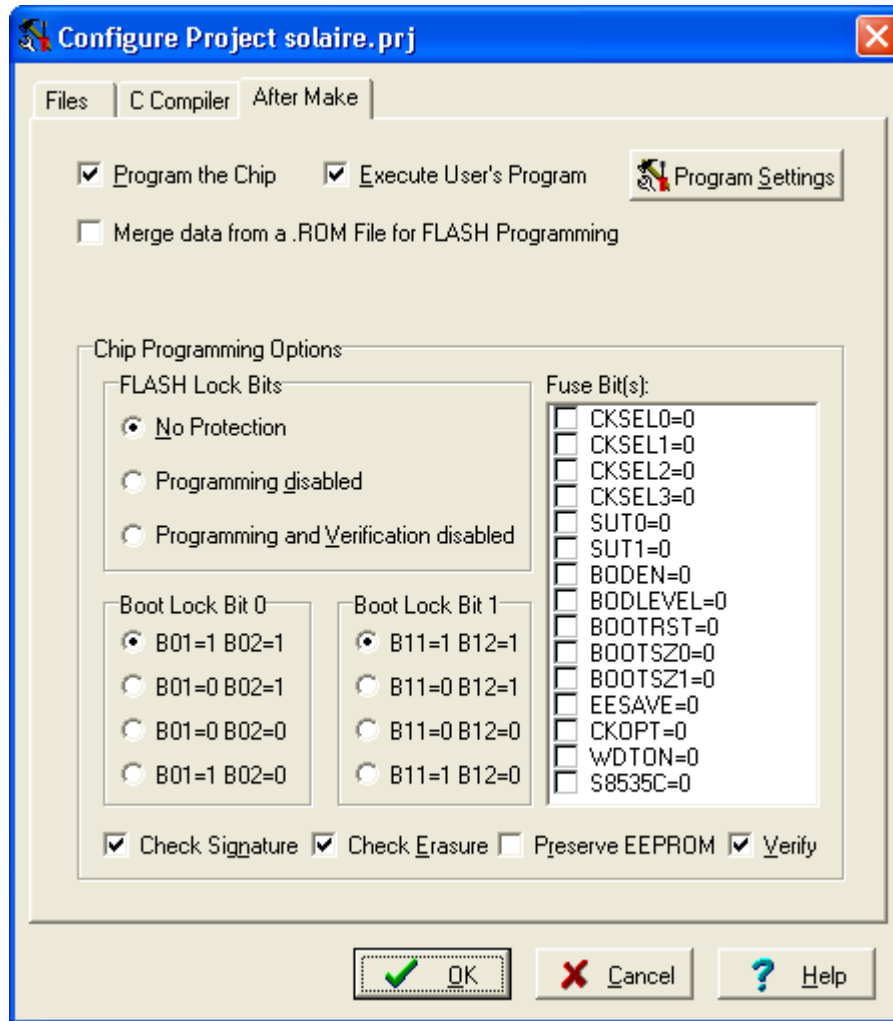
*Illustration 43: Onglet ADC*

On règle ensuite le convertisseur analogique numérique via « ADC ». On coche la case « ADC enabled » et on le met en « free running » afin qu'il convertisse dès qu'on lui demande.

Dans notre cas, il n'y a pas besoin de plus de réglages ; les autres fonctions ne nous sont pas utiles. On clique alors sur « File » puis « Generate, save and exit » pour démarrer le projet. Il est important de donner le même nom à chaque fichier à enregistrer et de les placer dans le même dossier.

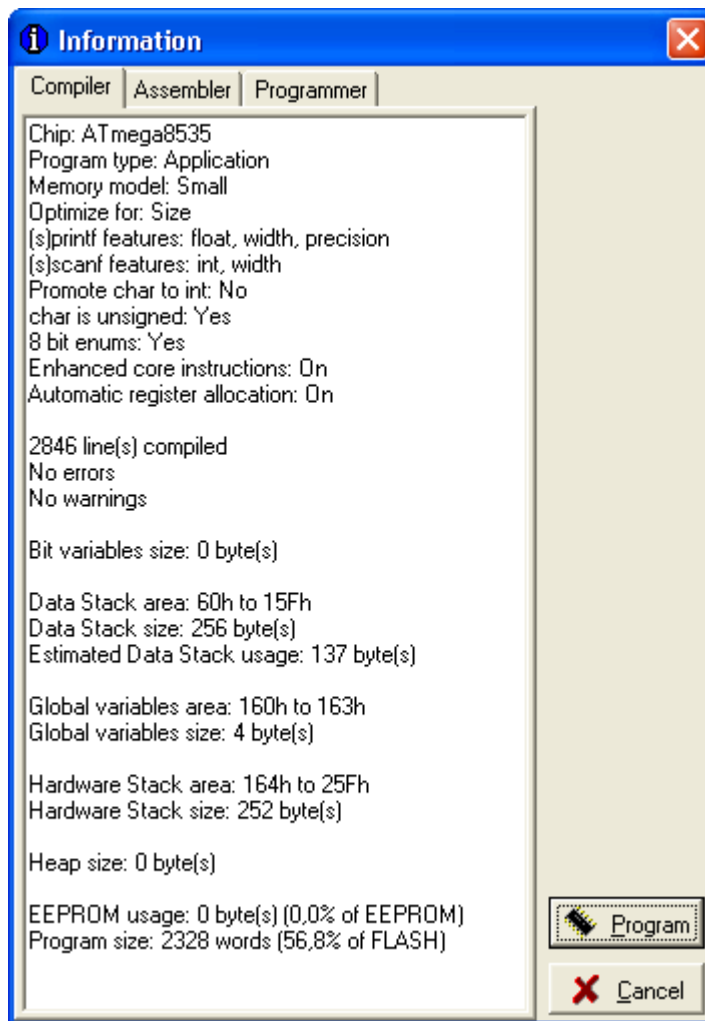
Il apparaît alors la fenêtre permettant de programmer. Les lignes de codes déjà écrites correspondent aux réglages que l'on vient de faire.

Toutefois, on ne peut pas encore envoyer le programme dans le micro-contrôleur.



*Illustration 44: Configuration du projet*

Il faut cliquer sur « Project » puis « Configure ». Il apparaît alors la fenêtre ci-dessus (Illustration 44). Il suffit de cocher les cases « Program the Chip » et « Execute User's Program » pour que l'on puisse envoyer le programme fait sur l'ATmega8535.



*Illustration 45: Compilation du programme*

Pour compiler le programme et l'envoyer sur l'ATmega8535, il suffit ensuite de cliquer sur « Project » et « Make ». La fenêtre ci-dessus (Illustration 45) apparaît alors et il suffit de lancer le programme.

## Conclusion

Nous pouvons conclure que l'étude et la réalisation d'un projet en partant juste d'une idée, d'une phrase et en quasi autonomie est une chose relativement nouvelle et donc relativement complexe car c'est un exercice totalement complet car il permet de mettre en œuvre à la fois les connaissances de la première année et celles de la deuxième. Toutefois nous pouvons constater qu'au jour d'aujourd'hui notre projet fonctionne totalement tant la partie électronique que la partie informatique.

La partie analogique du projet a nécessité l'étude des différentes sources d'énergies qui sont ici la batterie, la dynamo et le panneau solaire. Il a ensuite fallu mettre en place des hacheurs abaisseurs et élévateurs afin de traiter les tensions pour que celles-ci puissent être envoyées vers l'ATmega8535 qui ne supporte pas de tensions supérieures à 5 volts.

Dans la partie programmation, il a fallu tenir compte des sources d'énergies également. Une fois les tensions traitées par les hacheurs et envoyées sur l'ATmega8535, il a fallu écrire des lignes de codes faisant l'opération inverse des hacheurs afin d'afficher une tension image de la batterie, de la dynamo ou du panneau solaire sur l'écran LCD.

L'ensemble du projet aura coûté environ 35 euros pour la carte électronique ce qui est assez chère mais elle est composée de nombreux composants coûteux. Nous pensons qu'un tel dispositif pourra être commercialisé dans un avenir proche car nous voyons déjà apparaître sur le marché des vélos électriques.

En conclusion finale, ce projet a été mené à bien grâce à une équipe soudée et organisée.

## Résumé

Dans cette étude, nous avons abordé les trois grandes parties de notre projet, une présentation générale du système à étudier et à réaliser. Il s'agit d'un éclairage à LED pour vélo en autonomie. La partie électronique du projet consistait à étudier les différentes sources d'énergies ainsi que la réalisation du schéma électrique et du typon de la carte électronique. Puis nous avons terminés cette étude par l'aspect informatique du système qui consistait à étudier l'ensemble des fonctions que doit remplir l'ATmega8535 ainsi que découvrir le microcontrôleur utilisé. Dans cette partie nous détaillons également tout le programme informatique que nous avons créés.

Pour répondre au cahier des charges, nous avons élaboré une planning de réalisation et d'études à fin de pouvoir garder à l'esprit les différents objectifs à remplir. Nous avons réalisé une carte électronique qui n'a pas fonctionné du premier coup en revanche après modification cette carte fonctionne totalement. Nous avons également effectué une programmation complète de l'ATmega8535 pour que celui-ci fasse l'ensemble des fonctions qu'il doit réaliser. A ce jour notre projet est terminé et il est fonctionnel. De plus nous avons pu terminer ce projet avant la date limite par la bonne coopération de l'ensemble du binôme.

Pour terminer ce produit pourra être commercialisé dans un futur assez proche car dans l'ensemble ce projet à été bien exécuté néanmoins il peut recevoir des améliorations pour gagner au niveau de l'encombrement.

## Index des illustrations

Illustration 1: Schéma fonctionnel général.....	6
Illustration 2: Schéma fonctionnel du projet.....	6
Illustration 3: Planning prévisionnel et réel du projet.....	7
Illustration 4: Réalisation du test pour le panneau solaire.....	8
Illustration 5: Mesures lors du test du panneau solaire.....	9
Illustration 6: Caractéristique tension/courant du panneau solaire.....	9
Illustration 7: Panneau solaire ASI-F 2/12 utilisé pour le projet.....	10
Illustration 8: banc de test pour la dynamo.....	11
Illustration 9: Mesure de la tension au bornes de la dynamo.....	12
Illustration 10: Copie d'écran de l'oscilloscope.....	12
Illustration 11: Batterie utilisée pour ce projet.....	13
Illustration 12: Mesures du test pour la batterie.....	14
Illustration 13: Courbe exprimant la tension en fonction du temps dans la batterie.....	14
Illustration 14: Hacheur de type Boost (élevateur).....	15
Illustration 15: Pont redresseur double alternance.....	16
Illustration 16: Hacheur Buck (abaisseur).....	16
Illustration 17: Pont diviseur de tension.....	17
Illustration 18: Schéma explicatif du montage à réaliser.....	18
Illustration 19: Partie alimentation de la carte.....	19
Illustration 20: Partie 1 de la dynamo.....	21
Illustration 21: Partie 2 de la dynamo.....	21
Illustration 22: partie du panneau solaire.....	23
Illustration 23: Liste des composants, plus prix.....	25
Illustration 24: Typon de la carte.....	26
Illustration 25: Schéma global de l'ATmega 8535.....	27



Illustration 26: Définition des entrées et des sorties.....	28
Illustration 27: Déclaration des variables.....	29
Illustration 28: Initialisation de tout les ports.....	30
Illustration 29: Initialisation ADC et de l'écran LCD.....	31
Illustration 30: Explication du programme.....	31
Illustration 31: Incrémentation de « j ».....	32
Illustration 32: traitement des tensions.....	33
Illustration 33: Traitement des données.....	34
Illustration 34: Taille des variables.....	35
Illustration 35: Taille des variables.....	36
Illustration 36: Taille des variables bis.....	37
Illustration 37: suite des problèmes.....	38
Illustration 38: Démarrage de CodeVision.....	38
Illustration 39: Fenêtre CodeWizrd.....	39
Illustration 40: suite de l'explication du logiciel.....	40
Illustration 41: Onglet LCD.....	41
Illustration 42: configuration de l'ATmega.....	42
Illustration 43: Onglet ADC.....	43
Illustration 44: Configuration du projet.....	44
Illustration 45: Compilation du programme.....	45

## Bibliographie

[1]*La documentation de Thierry LEQUEU*. [En ligne]. Thierry Lequeu, mise à jour le mardi 1 novembre 2011 [consulté le 2 novembre 2011]. Disponible sur : [www.thierry-lequeu.fr](http://www.thierry-lequeu.fr)

[2]*RadioSpares* [En ligne]. RadioSpares, [consulté le 3 novembre 2011] Disponible sur : [www.radiospares-fr.rs-online.com](http://www.radiospares-fr.rs-online.com)

[3]*Farnell* [En ligne]. Farnell, [consulté le 13 octobre 2011]. Disponible sur : [fr.farnell.com](http://fr.farnell.com)

[4]*ATMEL Datasheet Atmega8535*. [En ligne] [consulté le 3 novembre 2011]. Disponible sur : [www.atmel.com](http://www.atmel.com)

[5]*LM2577 Datasheet*. [En ligne] [consulté le 3 novembre 2011]. Disponible sur [www.datasheetcatalog.com](http://www.datasheetcatalog.com)

[6]*LM2574 Datasheet*. [En ligne] [consulté le 3 novembre 2011]. Disponible sur [www.datasheetcatalog.com](http://www.datasheetcatalog.com)

# Annexes

Programme dans l'ATmega8535.....	52
Datasheet de l'ATmega8535.....	61
Datasheet de la carte de programmation.....	64
Datasheet du LM2574.....	67
Datasheet du LM2577.....	71

## Programme dans l'ATmega8535

/\*\*\*\*\*\*

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 : Panneau SOLaire  
Version :  
Date : 06/10/2011  
Author : V.Mollinier & T.Delaroche  
Company : IUT GEII Tours  
Comments: Groupe K4A avec Mr Thierry LEQUEU

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

// Entrées analogiques :
#define ADC_VREF_TYPE 0x00

#define MesBat 0      // Normalement entrée N°0
#define MesSol 1     // Normalement entrée N°1
#define MesDyn 2     // Normalement entrée N°2

// Entrées numériques :
#define BP1      PIND.0

// Sorties numériques :
#define PanToBat PORTA.3
#define Alim     PORTA.4
#define Buzzer  PORTA.5

// Read the AD conversion result
unsigned int 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 ADCW;
}

// Declare your global variables here

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

int j=0;

float kbat, kpan, kdyn;

unsigned int ibat, ipan, idyn;

unsigned char TensionBat[20], TensionPan[20], TensionDyn[20];

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=Out Func5=Out Func4=Out Func3=Out Func2=In Func1=In Func0=In
// State7=T State6=0 State5=0 State4=0 State3=0 State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x78;

// 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
// OCO 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
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
// Analog Comparator Output: Off
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: Free Running
ADMUX=ADC_VREF_TYPE;
ADCSRA=0x87;
SFIOR&=0x0F;

// LCD module initialization
lcd_init(16);
lcd_putsf("Bonjour");
lcd_gotoxy(0,0);      //x,y

//Mesurer la tension de la batterie au démarrage
ibat = read_adc(MesBat);
kbat = (float)ibat*2*5/1024;
delay_ms(2000);

while (1)
{

//TEST AFFICHAGE INCREMENTATION j
/* sprintf(TensionBat,"Bonjour - j = %d",j);
lcd_gotoxy(0,0);      //x,y
lcd_puts(TensionBat); */

ibat = read_adc(MesBat); //mesure de la tension batterie
idyn = read_adc(MesDyn); //mesure tension dynamo
ipan = read_adc(MesSol); //mesure tension panneau solaire

```

```

//TEST DU BP

/*if(PIND.0 == 1)
    {
        lcd_gotoxy(0,1);
        lcd_putsf("BP OFF");    //bouton relaché
        delay_ms(1000);
    }
else
    {
        lcd_gotoxy(0,1);
        lcd_putsf("BP ON ");    //bouton appuyé
        delay_ms(1000);
    } */

//Incrémentation de j lors d'un appuie sur BP
if(BP1==0)
    {
        j++;                    // j = j++ NE FAIT RIEN!!!!!!!
ATTENTION!!!!!!
        delay_ms(200);
    }
if(j>=4)
    j=1;

//MESURE TENSION BATTERIE
if(j==1)
    {
        lcd_clear();

```

```

        lcd_gotoxy(0,0);
        lcd_putsf("batterie");

        kbat = (float)ibat*2*5/1024;           //valeur de la tension
réelle, calcul forcé en float
        sprintf(TensionBat,"%4.3f",kbat);
        lcd_gotoxy(0,1);
        lcd_puts(TensionBat); //on affiche la valeur de k
        delay_ms(50);
    }

//MESURE TENSION PANNEAU
if(j==2)
{
    lcd_clear();
    lcd_gotoxy(0,0);
    lcd_putsf("panneau ");

    kpan = (float)ipan*24/1024;           //valeur de la tension réelle
    sprintf(TensionPan,"%4.3f",kpan);
    lcd_gotoxy(0,1);
    lcd_puts(TensionPan); //on affiche la valeur de k
    delay_ms(50);
}

//MESURE TENSION DYNAMO
if(j==3)
{
    lcd_clear();
    lcd_gotoxy(0,0);
    lcd_putsf("dynamo ");

```

```

        kdyn = (float)idyn*2*5/1024;          //valeur de la tension
réelle

        sprintf(TensionDyn,"%4.3f",kdyn);

        lcd_gotoxy(0,1);

        lcd_puts(TensionDyn); //on affiche la valeur de k

        delay_ms(50);

    }

if(kbat<=8.4)

    {

        PanToBat = 1;          //recharger batterie

        Buzzer = 1;          //buzzer en marche pendant 2s

        delay_ms(500);

        Buzzer = 0;          // éteindre buzzer

    }

else          //sinon ne rien faire

    {

        PanToBat = 0;

Buzzer = 0;

    }

if(kbat<=8)

    Alim = 0;

else

    Alim = 1;

};

}

```

---

## Features

- High-performance, Low-power AVR<sup>®</sup> 8-bit Microcontroller
- Advanced RISC Architecture
  - 130 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
  - 8K Bytes of In-System Self-Programmable Flash
    - Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits
  - In-System Programming by On-chip Boot Program
  - True Read-While-Write Operation
  - 512 Bytes EEPROM
    - Endurance: 100,000 Write/Erase Cycles
  - 512 Bytes Internal SRAM
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four PWM Channels
  - 8-channel, 10-bit ADC
    - 8 Single-ended Channels
    - 7 Differential Channels for TQFP Package Only
    - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x for TQFP Package Only
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
  - 32 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44-pad QFN/MLF
- Operating Voltages
  - 2.7 - 5.5V for ATmega8535L
  - 4.5 - 5.5V for ATmega8535
- Speed Grades
  - 0 - 8 MHz for ATmega8535L
  - 0 - 16 MHz for ATmega8535



---

8-bit **AVR<sup>®</sup>**  
Microcontroller  
with 8K Bytes  
In-System  
Programmable  
Flash

---

ATmega8535  
ATmega8535L

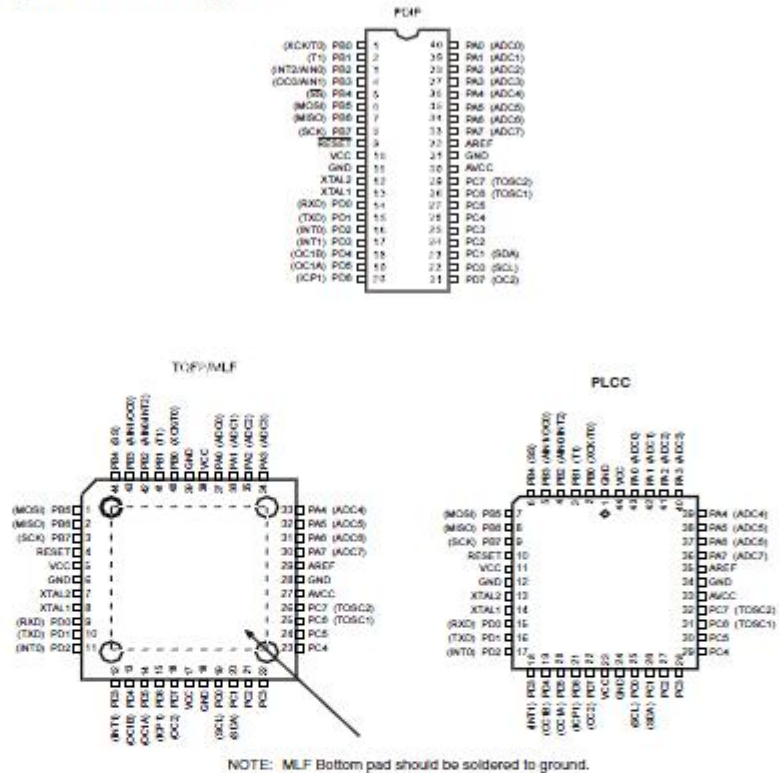
2502K-AVR-10/06





## Pin Configurations

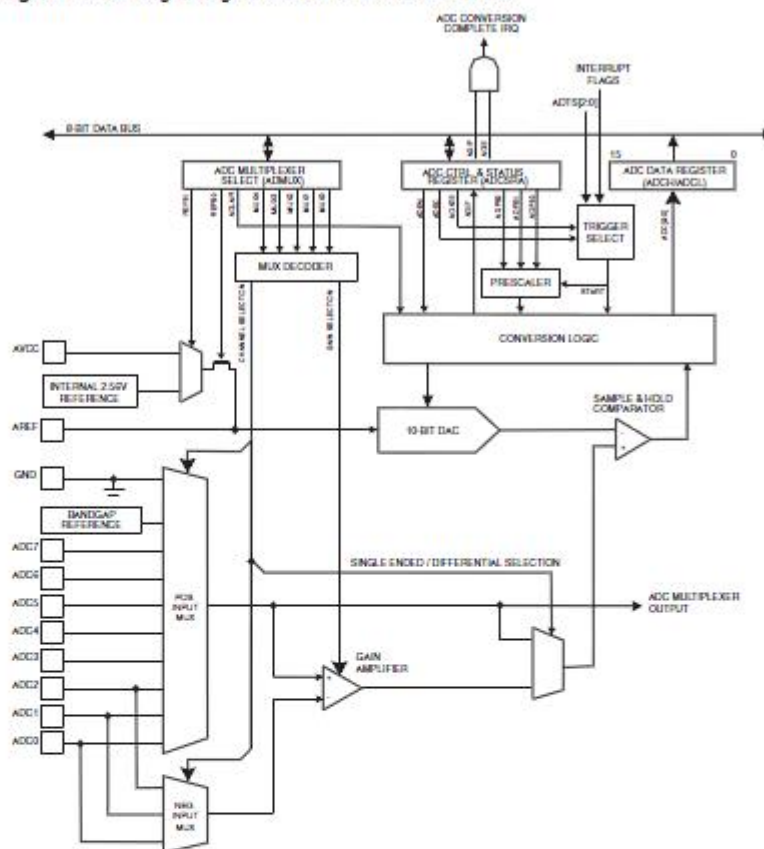
Figure 1. Pinout ATmega8535



## Disclaimer

Typical values contained in this data sheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

Figure 98. Analog-to-Digital Converter Block Schematic



**Operation**

The ADC converts an analog input voltage to a 10-bit digital value through successive approximation. The minimum value represents GND and the maximum value represents the voltage on the AREF pin minus 1 LSB. Optionally, AVCC or an internal 2.56V reference voltage may be connected to the AREF pin by writing to the REFSn bits in the ADMUX Register. The internal voltage reference may thus be decoupled by an external capacitor at the AREF pin to improve noise immunity.

The analog input channel and differential gain are selected by writing to the MUX bits in ADMUX. Any of the ADC input pins, as well as GND and a fixed bandgap voltage reference, can be selected as single ended inputs to the ADC. A selection of ADC input pins can be selected as positive and negative inputs to the differential gain amplifier.

If differential channels are selected, the differential gain stage amplifies the voltage difference between the selected input channel pair by the selected gain factor. This amplified value then becomes the analog input to the ADC. If single ended channels are used, the gain amplifier is bypassed altogether.

## Datasheet de la carte de programmation

Thierry LEQUEU – Octobre 2006 – [DATA076] – Fichier : PROJETS-IUT5.DOC

### 5.2 Désignation des composants

Tableau 5.1. Liste de composants (projets-iut5.xls / AT8535).

N°	Quantité	Référence	Désignation	Empreinte
1	1	AFF1	LCD 16x4	MC1604C
2	1	C1	100uF 63V	RADIAL08
3	1	C2	470uF 6.3V	RADIAL06L
4	2	C3,C5	10uF 6.3V	RADIAL06
5	2	C6,C4	100nF	CK06
6	1	C7	22pF	CK06
7	1	C8	22pF	CK06
8	1	D1	1N4007	DO41
9	1	D2	1N5819	DO41
10	1	D3	2 mA	LED3
11	1	D4	3mm	LED3
12	1	JP1	ALIM	WEID2
13	1	JP2	CON ISP	10SH100L
14	1	JP3	ANALOG	20SH100
15	1	JP4	n.c.	02PL1
16	1	JP5	HEADER 10X2	20SH100L
17	1	JP6	HEADER 10	10PL1
18	1	JP7	n.c.	05PL1
19	1	JP8	Port D	08PL1
20	1	JP9	BP1	02PL1
21	1	JP10	BP2	02PL1
22	1	JP11	BNC	02PL1
23	1	JP12	BNC	BNC1
24	1	L1	10uH	RADIAL06L
25	1	L2	47uH	RADIAL06L
26	1	P1	10K	RAJ1
27	1	Q1	16 MHz	HC18UV
28	2	R1,R5	1.5k	RC04L
29	1	R2	1k	RC04L
30	2	R3,R4	4.7k	RC04L
31	2	SW2,SW1	TOUCHE	REDROND
32	1	U1	LM2575-5	TO220-5B
33	1	U2	ATmega8535	40DIP600
34	4	VIS1,VIS2,VIS3,VIS4	VISSERIE	M3L



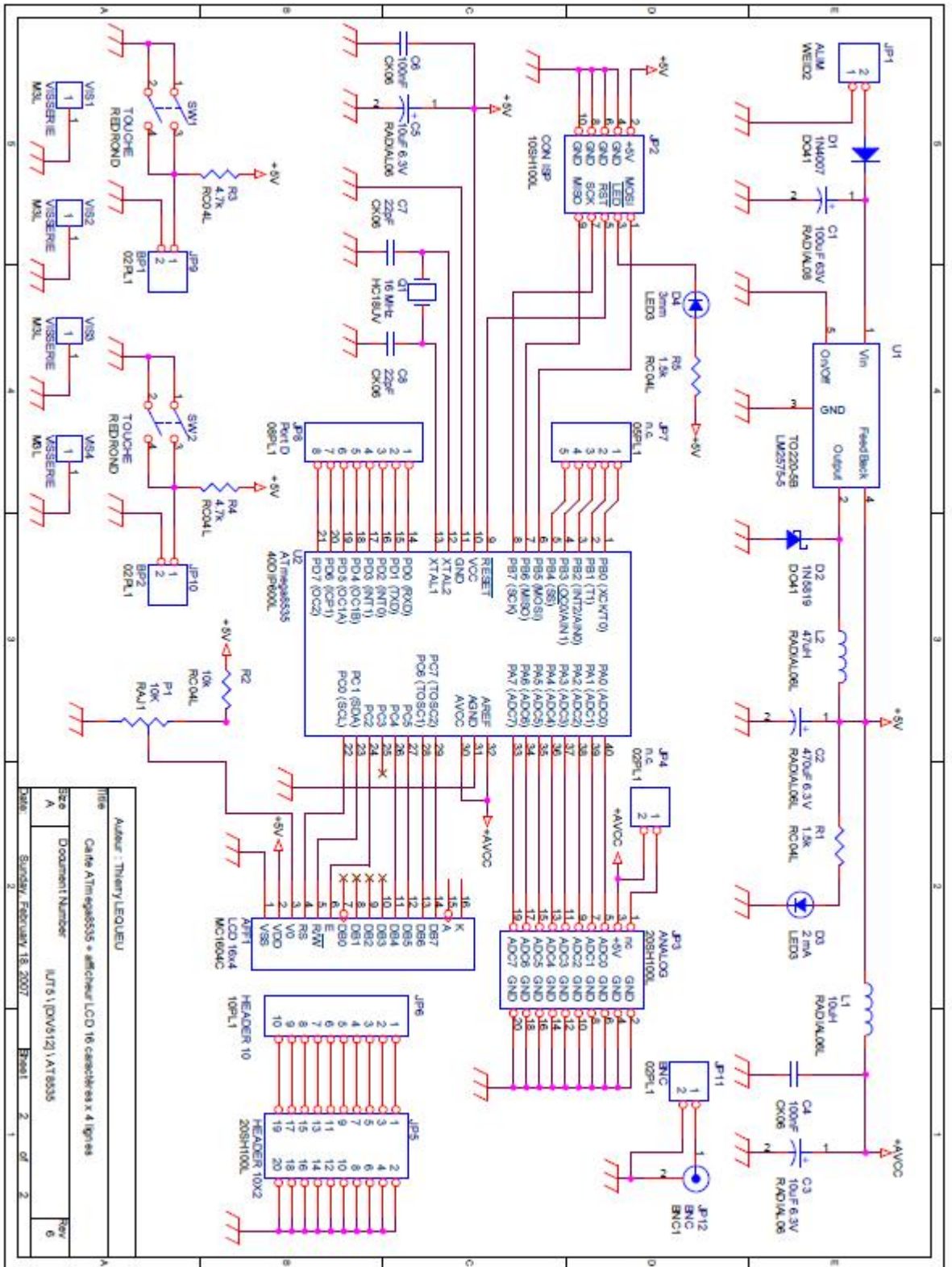
## Carte ATmega8535 + afficheur LCD 16 caractères x 4 lignes

Revised: Thursday, February 15, 2007

IUT5 \ [DIV512] \ AT8535 Revision: 5

Référence	Qté	Désignation	Fournisseur	Code Cde.	U.d.V.	Prix U.	Prix T.
AFF1	1	LCD 16x4 MC1604C-SYL	FARNELL	944-9019	1	23,28 €	23,280 €
C1	1	100uF 63V	Radiospares	315-0962	5	2,00 €	0,400 €
C2	1	470uF 6.3V	Radiospares	449-0845	5	1,21 €	0,242 €
C3,C5	2	10uF 6.3V	Radiospares	449-1006	5	0,85 €	0,340 €
C6,C4	2	100nF	IUT GEII		1	0,08 €	0,160 €
C7	1	22pF	IUT GEII		1	0,02 €	0,020 €
C8	1	22pF	IUT GEII		1	0,02 €	0,020 €
D1	1	1N4007	IUT GEII		1	0,02 €	0,020 €
D2	1	1N5819	Radiospares	544-4994	5	1,10 €	0,220 €
D3	1	LED verte 3mm 2mA	Radiospares	180-8451	10	3,32 €	0,332 €
D4	1	LED jaune 3mm 2mA	Radiospares	171-1228	10	3,40 €	0,340 €
JP1	1	Connecteur 2 points	Radiospares	131-8920	10	2,61 €	0,261 €
JP2	1	CON ISP	Radiospares		1		0,000 €
JP3	1	ANALOG	Radiospares		1		0,000 €
JP4	1	Support tulipe (2 plots)	Radiospares		1		0,000 €
JP5	1	HEADER 10X2	Radiospares		1		0,000 €
JP6	1	HEADER 10	Radiospares		1		0,000 €
JP7	1	Support tulipe (5 plots)	Radiospares		1		0,000 €
JP8	1	Support tulipe (8 plots)	Radiospares		1		0,000 €
JP9	1	Support tulipe (2 plots)	Radiospares		1		0,000 €
JP10	1	Support tulipe (2 plots)	Radiospares		1		0,000 €
JP11	1	BNC	Radiospares	477-090	1	3,42 €	3,420 €
JP12	1	Support tulipe (2 plots)	IUT GEII		1		0,000 €
L1	1	10uH	Radiospares	432-4388	10	12,93 €	1,293 €
L2	1	47uH	Radiospares	432-4394	10	12,93 €	1,293 €
P1	1	10K	IUT GEII		1	0,10 €	0,100 €
Q1	1	16 MHz	IUT GEII		1		0,000 €
R1,R5	2	1.5k	IUT GEII		1	0,01 €	0,020 €
R2	1	1k	IUT GEII		1	0,01 €	0,010 €
R3,R4	2	4.7k	IUT GEII		1	0,01 €	0,020 €
SW2,SW1	2	TOUCHE	CONRAD		1		0,000 €
U1	1	LM2575-5	Radiospares		1		0,000 €
U2	1	ATmega8535	FARNELL	917-1444	1	5,81 €	5,810 €
JP1	1	Visserie M3	IUT GEII		1	0,10 €	0,100 €
Divers	4	Connecteur HE10 20 broches femelle à sertir	FARNELL	109-9239	1	0,52 €	2,080 €
Divers	32	Circuit imprimé SF 70x45 mm	Radiospares	159-6091	600	13,99 €	0,746 €
Divers					1		0,000 €

TOTAL H.T. :	40,53 €
dont TVA : 19,60%	7,94 €
<b>TOTAL T.T.C. :</b>	<b>48,47 €</b>





Advance Information

**Easy Switcher™ 0.5 A Step-Down Voltage Regulator**

The LM2574 series of regulators are monolithic integrated circuits ideally suited for easy and convenient design of a step-down switching regulator (buck converter). All circuits of this series are capable of driving a 0.5 A load with excellent line and load regulation. These devices are available in fixed output voltages of 3.3 V, 5.0 V, 12 V, 15 V, and an adjustable output version.

These regulators were designed to minimize the number of external components to simplify the power supply design. Standard series of inductors optimized for use with the LM2574 are offered by several different inductor manufacturers.

Since the LM2574 converter is a switch-mode power supply, its efficiency is significantly higher in comparison with popular three-terminal linear regulators, especially with higher input voltages. In most cases, the power dissipated by the LM2574 regulator is so low, that the copper traces on the printed circuit board are normally the only heatsink needed and no additional heatsinking is required.

The LM2574 features include a guaranteed ±4% tolerance on output voltage within specified input voltages and output load conditions, and ±10% on the oscillator frequency (±2% over 0°C to +125°C). External shutdown is included, featuring 80 µA (typical) standby current. The output switch includes cycle-by-cycle current limiting, as well as thermal shutdown for full protection under fault conditions.

**Features**

- 3.3 V, 5.0 V, 12 V, 15 V, and Adjustable Output Versions
- Adjustable Version Output Voltage Range, 1.23 to 37 V ±4% max over Line and Load Conditions
- Guaranteed 0.5 A Output Current
- Wide Input Voltage Range: 4.75 to 40 V
- Requires Only 4 External Components
- 52 kHz Fixed Frequency Internal Oscillator
- TTL Shutdown Capability, Low Power Standby Mode
- High Efficiency
- Uses Readily Available Standard Inductors
- Thermal Shutdown and Current Limit Protection

**Applications**

- Simple and High-Efficiency Step-Down (Buck) Regulators
- Efficient Pre-regulator for Linear Regulators
- On-Card Switching Regulators
- Positive to Negative Converters (Buck-Boost)
- Negative Step-Up Converters
- Power Supply for Battery Chargers

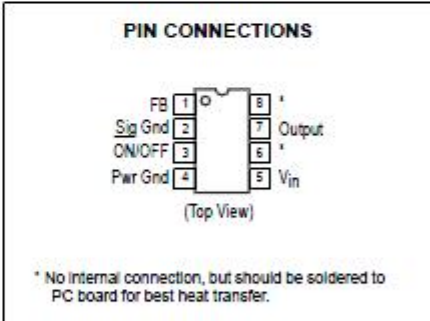
This document contains information on a new product. Specifications and information herein are subject to change without notice.

Order this document by LM2574/D

**LM2574**

**EASY SWITCHER™  
0.5 A STEP-DOWN  
VOLTAGE REGULATOR**

**SEMICONDUCTOR  
TECHNICAL DATA**



**DEVICE TYPE/NOMINAL OUTPUT VOLTAGE**

LM2574-3.3	3.3 V
LM2574-5	5.0 V
LM2574-12	12 V
LM2574-15	15 V
LM2574-ADJ	1.23 V to 37 V

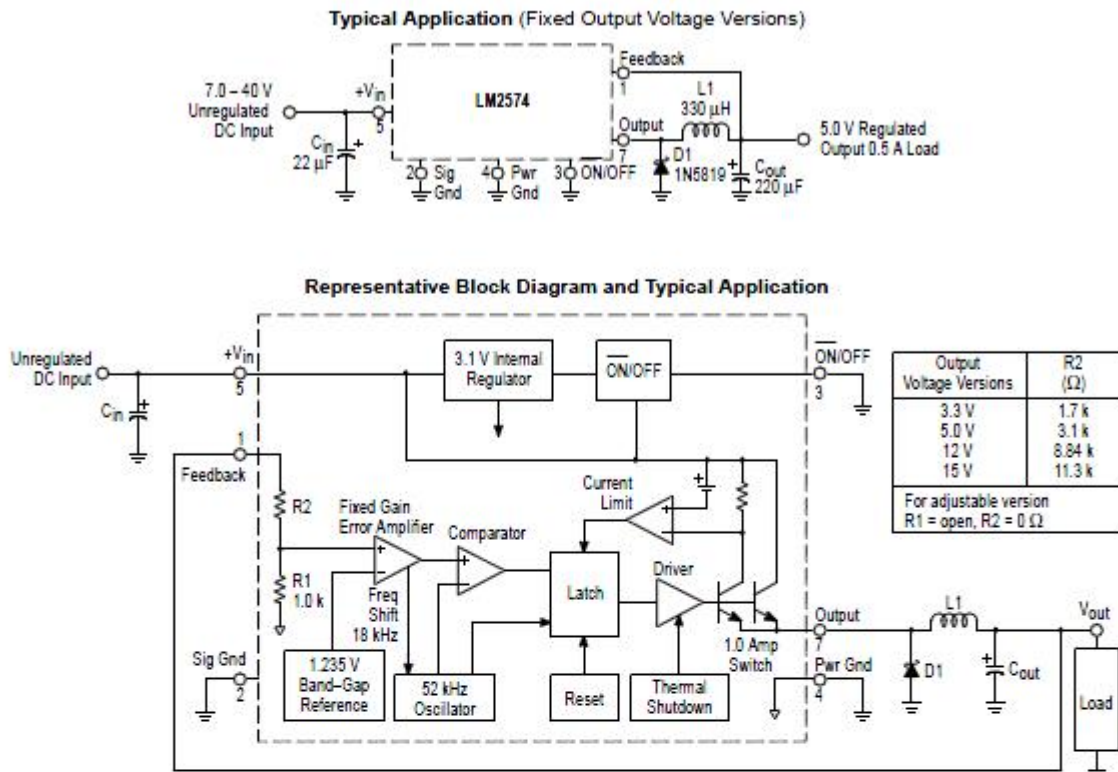
**ORDERING INFORMATION**

Device	Operating Temperature Range	Package
LM2574N-XX	T <sub>A</sub> = -40° to +125°C	DIP-8

XX = Voltage Option, i.e. 3.3, 5, 12, 15 V; and ADJ for Adjustable Output.

## LM2574

Figure 1. Block Diagram and Typical Application



**ABSOLUTE MAXIMUM RATINGS** (Absolute Maximum Ratings indicate limits beyond which damage to the device may occur).

Rating	Symbol	Value	Unit
Maximum Supply Voltage	$V_{in}$	45	V
ON/OFF Pin Input Voltage	-	$-0.3 \text{ V} \leq V \leq +V_{in}$	V
Output Voltage to Ground (Steady State)	-	-1.0	V
Power Dissipation	$P_D$	Internally Limited	W
Thermal Resistance, Junction-to-Ambient	$R_{\theta JA}$	100	$^{\circ}\text{C/W}$
Thermal Resistance, Junction-to-Case	$R_{\theta JC}$	5.0	$^{\circ}\text{C/W}$
Storage Temperature Range	$T_{stg}$	$-65^{\circ}\text{C}$ to $+150^{\circ}\text{C}$	$^{\circ}\text{C}$
Minimum ESD Rating (Human Body Model: C = 100 pF, R = 1.5 kΩ)	-	2.0	kV
Lead Temperature (Soldering, 10 seconds)	-	260	$^{\circ}\text{C}$
Maximum Junction Temperature	$T_J$	150	$^{\circ}\text{C}$

NOTE: ESD data available upon request.

## LM2574

**OPERATING RATINGS** (Operating Ratings indicate conditions for which the device is intended to be functional, but do not guarantee specific performance limits. For guaranteed specifications and test conditions, see the Electrical Characteristics).

Rating	Symbol	Value	Unit
Operating Junction Temperature Range	$T_J$	-40 to +125	$^{\circ}\text{C}$
Supply Voltage	$V_{IN}$	40	V

### SYSTEM PARAMETERS ([Note 1] Test Circuit Figure 16)

**ELECTRICAL CHARACTERISTICS** (Unless otherwise specified,  $V_{IN} = 12\text{ V}$  for the 3.3 V, 5.0 V, and Adjustable version,  $V_{IN} = 25\text{ V}$  for the 12 V version,  $V_{IN} = 30\text{ V}$  for the 15 V version,  $I_{LOAD} = 100\text{ mA}$ . For typical values  $T_J = 25^{\circ}\text{C}$ , for min/max values  $T_J$  is the operating junction temperature range that applies [Note 2], unless otherwise noted).

Characteristic	Symbol	Min	Typ	Max	Unit
<b>LM2574-3.3</b> ([Note 1] Test Circuit Figure 16)					
Output Voltage ( $V_{IN} = 12\text{ V}$ , $I_{LOAD} = 100\text{ mA}$ , $T_J = 25^{\circ}\text{C}$ )	$V_{OUT}$	3.234	3.3	3.366	V
Output Voltage ( $4.75\text{ V} \leq V_{IN} \leq 40\text{ V}$ , $0.1\text{ A} \leq I_{LOAD} \leq 0.5\text{ A}$ ) $T_J = 25^{\circ}\text{C}$ $T_J = -40\text{ to }+125^{\circ}\text{C}$	$V_{OUT}$	3.168 3.135	3.3 -	3.432 3.465	V
Efficiency ( $V_{IN} = 12\text{ V}$ , $I_{LOAD} = 0.5\text{ A}$ )	$\eta$	-	72	-	%
<b>LM2574-5</b> ([Note 1] Test Circuit Figure 16)					
Output Voltage ( $V_{IN} = 12\text{ V}$ , $I_{LOAD} = 100\text{ mA}$ , $T_J = 25^{\circ}\text{C}$ )	$V_{OUT}$	4.9	5.0	5.1	V
Output Voltage ( $7.0\text{ V} \leq V_{IN} \leq 40\text{ V}$ , $0.1\text{ A} \leq I_{LOAD} \leq 0.5\text{ A}$ ) $T_J = 25^{\circ}\text{C}$ $T_J = -40\text{ to }+125^{\circ}\text{C}$	$V_{OUT}$	4.8 4.75	5.0	5.2 5.25	V
Efficiency ( $V_{IN} = 12\text{ V}$ , $I_{LOAD} = 0.5\text{ A}$ )	$\eta$	-	77	-	%
<b>LM2574-12</b> ([Note 1] Test Circuit Figure 16)					
Output Voltage ( $V_{IN} = 25\text{ V}$ , $I_{LOAD} = 100\text{ mA}$ , $T_J = 25^{\circ}\text{C}$ )	$V_{OUT}$	11.76	12	12.24	V
Output Voltage ( $15\text{ V} \leq V_{IN} \leq 40\text{ V}$ , $0.1\text{ A} \leq I_{LOAD} \leq 0.5\text{ A}$ ) $T_J = 25^{\circ}\text{C}$ $T_J = -40\text{ to }+125^{\circ}\text{C}$	$V_{OUT}$	11.52 11.4	12 -	12.48 12.6	V
Efficiency ( $V_{IN} = 15\text{ V}$ , $I_{LOAD} = 0.5\text{ A}$ )	$\eta$	-	88	-	%
<b>LM2574-15</b> ([Note 1] Test Circuit Figure 16)					
Output Voltage ( $V_{IN} = 30\text{ V}$ , $I_{LOAD} = 100\text{ mA}$ , $T_J = 25^{\circ}\text{C}$ )	$V_{OUT}$	14.7	15	15.3	V
Output Voltage ( $18\text{ V} < V_{IN} < 40\text{ V}$ , $0.1\text{ A} < I_{LOAD} < 0.5\text{ A}$ ) $T_J = 25^{\circ}\text{C}$ $T_J = -40\text{ to }+125^{\circ}\text{C}$	$V_{OUT}$	14.4 14.25	15	15.6 15.75	V
Efficiency ( $V_{IN} = 18\text{ V}$ , $I_{LOAD} = 0.5\text{ A}$ )	$\eta$	-	88	-	%
<b>LM2574 ADJUSTABLE VERSION</b> ([Note 1] Test Circuit Figure 16)					
Feedback Voltage $V_{IN} = 12\text{ V}$ , $I_{LOAD} = 100\text{ mA}$ , $V_{OUT} = 5.0\text{ V}$ , $T_J = 25^{\circ}\text{C}$	$V_{FB}$	1.217	1.23	1.243	V
Feedback Voltage $7.0\text{ V} \leq V_{IN} \leq 40\text{ V}$ , $0.1\text{ A} \leq I_{LOAD} \leq 0.5\text{ A}$ , $V_{OUT} = 5.0\text{ V}$ $T_J = 25^{\circ}\text{C}$ $T_J = -40\text{ to }+125^{\circ}\text{C}$	$V_{FBT}$	1.193 1.18	1.23	1.267 1.28	V
Efficiency ( $V_{IN} = 12\text{ V}$ , $I_{LOAD} = 0.5\text{ A}$ , $V_{OUT} = 5.0\text{ V}$ )	$\eta$	-	77	-	%

NOTES: 1. External components such as the catch diode, inductor, input and output capacitors can affect the switching regulator system performance. When the LM2574 is used as shown in the Figure 16 test circuit, the system performance will be as shown in the system parameters section of the Electrical Characteristics.

2. Tested junction temperature range for the LM2574:  $T_{LOW} = -40^{\circ}\text{C}$   $T_{HIGH} = +125^{\circ}\text{C}$

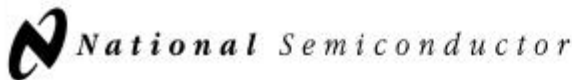
## LM2574

### SYSTEM PARAMETERS (Note 1) Test Circuit Figure 16)

**ELECTRICAL CHARACTERISTICS (continued)** (Unless otherwise specified,  $V_{in} = 12\text{ V}$  for the 3.3 V, 5.0 V, and Adjustable version,  $V_{in} = 25\text{ V}$  for the 12 V version,  $V_{in} = 30\text{ V}$  for the 15 V version.  $I_{Load} = 100\text{ mA}$ . For typical values  $T_J = 25^\circ\text{C}$ , for min/max values  $T_J$  is the operating junction temperature range that applies [Note 2], unless otherwise noted).

Characteristic	Symbol	Min	Typ	Max	Unit
<b>ALL OUTPUT VOLTAGE VERSIONS</b>					
Feedback Bias Current $V_{out} = 5.0\text{ V}$ (Adjustable Version Only) $T_J = 25^\circ\text{C}$ $T_J = -40\text{ to }+125^\circ\text{C}$	$I_b$	–	25	100	nA
Oscillator Frequency (Note 3) $T_J = 25^\circ\text{C}$ $T_J = 0\text{ to }+125^\circ\text{C}$ $T_J = -40\text{ to }+125^\circ\text{C}$	$f_O$	–	52	–	kHz
Saturation Voltage ( $I_{out} = 0.5\text{ A}$ , [Note 4]) $T_J = 25^\circ\text{C}$ $T_J = -40\text{ to }+125^\circ\text{C}$	$V_{sat}$	–	1.0	1.2	V
Max Duty Cycle ("on") [Note 5]	DC	93	98	–	%
Current Limit Peak Current (Notes 3 and 4) $T_J = 25^\circ\text{C}$ $T_J = -40\text{ to }+125^\circ\text{C}$	$I_{CL}$	0.7	1.0	1.6	A
Output Leakage Current (Notes 6 and 7), $T_J = 25^\circ\text{C}$ Output = 0 V Output = – 1.0 V	$I_L$	–	0.8	2.0	mA
Quiescent Current (Note 6) $T_J = 25^\circ\text{C}$ $T_J = -40\text{ to }+125^\circ\text{C}$	$I_Q$	–	5.0	9.0	mA
Standby Quiescent Current (ON/OFF Pin = 5.0 V ("off")) $T_J = 25^\circ\text{C}$ $T_J = -40\text{ to }+125^\circ\text{C}$	$I_{stby}$	–	60	200	$\mu\text{A}$
ON/OFF Pin Logic Input Level $V_{out} = 0\text{ V}$ $T_J = 25^\circ\text{C}$ $T_J = -40\text{ to }+125^\circ\text{C}$ Nominal Output Voltage $T_J = 25^\circ\text{C}$ $T_J = -40\text{ to }+125^\circ\text{C}$	$V_{IH}$  $V_{IL}$	2.2 2.4	1.4 –	– –	V
ON/OFF Pin Input Current ON/OFF Pin = 5.0 V ("off"), $T_J = 25^\circ\text{C}$ ON/OFF Pin = 0 V ("on"), $T_J = 25^\circ\text{C}$	$I_{IH}$ $I_{IL}$	– –	15 0	30 5.0	$\mu\text{A}$

- NOTES:**
- External components such as the catch diode, inductor, input and output capacitors can affect the switching regulator system performance. When the LM2574 is used as shown in the Figure 16 test circuit, the system performance will be as shown in the system parameters section of the Electrical Characteristics.
  - Tested junction temperature range for the LM2574:  $T_{low} = -40^\circ\text{C}$   $T_{high} = +125^\circ\text{C}$
  - The oscillator frequency reduces to approximately 18 kHz in the event of an output short or an overload which causes the regulated output voltage to drop approximately 40% from the nominal output voltage. This self protection feature lowers the average power dissipation of the IC by lowering the minimum duty cycle from 5% down to approximately 2%.
  - Output (Pin 2) sourcing current. No diode, inductor or capacitor connected to the output pin.
  - Feedback (Pin 4) removed from output and connected to 0 V.
  - Feedback (Pin 4) removed from output and connected to 12 V for the Adjustable, 3.3 V, and 5.0 V versions, and 25 V for the 12 V and 15 V versions, to force the output transistor OFF.
  - $V_{in} = 40\text{ V}$ .



June 1999

## LM1577/LM2577 Series SIMPLE SWITCHER® Step-Up Voltage Regulator

### General Description

The LM1577/LM2577 are monolithic integrated circuits that provide all of the power and control functions for step-up (boost), flyback, and forward converter switching regulators. The device is available in three different output voltage versions: 12V, 15V, and adjustable.

Requiring a minimum number of external components, these regulators are cost effective, and simple to use. Listed in this data sheet are a family of standard inductors and flyback transformers designed to work with these switching regulators.

Included on the chip is a 3.0A NPN switch and its associated protection circuitry, consisting of current and thermal limiting, and undervoltage lockout. Other features include a 52 kHz fixed-frequency oscillator that requires no external components, a soft start mode to reduce in-rush current during start-up, and current mode control for improved rejection of input voltage and output load transients.

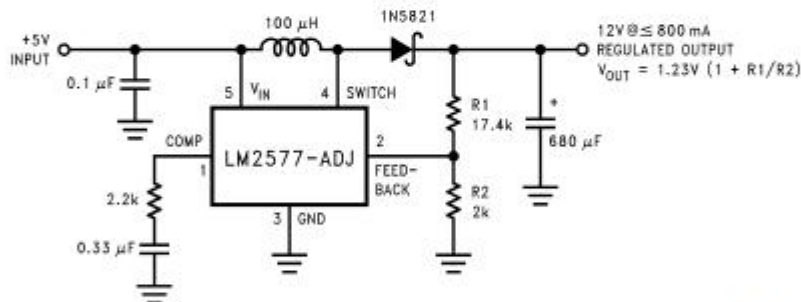
### Features

- Requires few external components
- NPN output switches 3.0A, can stand off 65V
- Wide input voltage range: 3.5V to 40V
- Current-mode operation for improved transient response, line regulation, and current limit
- 52 kHz internal oscillator
- Soft-start function reduces in-rush current during start-up
- Output switch protected by current limit, under-voltage lockout, and thermal shutdown

### Typical Applications

- Simple boost regulator
- Flyback and forward regulators
- Multiple-output regulator

### Typical Application



DS011468-1

Note: Pin numbers shown are for TO-220 (T) package.

### Ordering Information

Temperature Range	Package Type	Output Voltage			NSC Package Drawing	Package
		12V	15V	ADJ		
-40°C ≤ T <sub>A</sub> ≤ +125°C	24-Pin Surface Mount	LM2577M-12	LM2577M-15	LM2577M-ADJ	M24B	SO
	16-Pin Molded DIP	LM2577N-12	LM2577N-15	LM2577N-ADJ	N16A	N
	5-Lead Surface Mount	LM2577S-12	LM2577S-15	LM2577S-ADJ	TS5B	TO-263
	5-Straight Leads	LM2577T-12	LM2577T-15	LM2577T-ADJ	T05A	TO-220
	5-Bent Staggered Leads	LM2577T-12 Flow LB03	LM2577T-15 Flow LB03	LM2577T-ADJ Flow LB03	T05D	TO-220
-55°C ≤ T <sub>A</sub> ≤ +150°C	4-Pin TO-3	LM1577K-12/883	LM1577K-15/883	LM1577K-ADJ/883	K04A	TO-3

SIMPLE SWITCHER® is a registered trademark of National Semiconductor Corporation.

## Electrical Characteristics—LM1577-ADJ, LM2577-ADJ

Specifications with standard type face are for  $T_J = 25^\circ\text{C}$ , and those in bold type face apply over full Operating Temperature Range. Unless otherwise specified,  $V_{IN} = 5\text{V}$ ,  $V_{FEEDBACK} = V_{REF}$ , and  $I_{SWITCH} = 0$ .

Symbol	Parameter	Conditions	Typical	LM1577-ADJ Limit (Notes 3, 4)	LM2577-ADJ Limit (Note 5)	Units (Limits)
<b>SYSTEM PARAMETERS</b> Circuit of Figure 3 (Note 6)						
$V_{OUT}$	Output Voltage	$V_{IN} = 5\text{V to }10\text{V}$ $I_{LOAD} = 100\text{ mA to }800\text{ mA}$ (Note 3)	12.0	11.60/11.40 12.40/12.60	11.60/11.40 12.40/12.60	V V(min) V(max)
$\frac{\Delta V_{OUT}}{\Delta V_{IN}}$	Line Regulation	$V_{IN} = 3.5\text{V to }10\text{V}$ $I_{LOAD} = 300\text{ mA}$	20	50/100	50/100	mV mV(max)
$\frac{\Delta V_{OUT}}{\Delta I_{LOAD}}$	Load Regulation	$V_{IN} = 5\text{V}$ $I_{LOAD} = 100\text{ mA to }800\text{ mA}$	20	50/100	50/100	mV mV(max)
$\eta$	Efficiency	$V_{IN} = 5\text{V}$ , $I_{LOAD} = 800\text{ mA}$	80			%
<b>DEVICE PARAMETERS</b>						
$I_S$	Input Supply Current	$V_{FEEDBACK} = 1.5\text{V}$ (Switch Off)	7.5	10.0/14.0	10.0/14.0	mA mA(max)
		$I_{SWITCH} = 2.0\text{A}$ $V_{COMP} = 2.0\text{V}$ (Max Duty Cycle)	25	50/85	50/85	mA mA(max)
$V_{LUV}$	Input Supply Undervoltage Lockout	$I_{SWITCH} = 100\text{ mA}$	2.90	2.70/2.65 3.10/3.15	2.70/2.65 3.10/3.15	V V(min) V(max)
$f_O$	Oscillator Frequency	Measured at Switch Pin $I_{SWITCH} = 100\text{ mA}$	52	48/42 56/62	48/42 56/62	kHz kHz(min) kHz(max)
$V_{REF}$	Reference Voltage	Measured at Feedback Pin $V_{IN} = 3.5\text{V to }40\text{V}$ $V_{COMP} = 1.0\text{V}$	1.230	1.214/1.206 1.246/1.254	1.214/1.206 1.246/1.254	V V(min) V(max)
$\frac{\Delta V_{REF}}{\Delta V_{IN}}$	Reference Voltage Line Regulation	$V_{IN} = 3.5\text{V to }40\text{V}$	0.5			mV
$I_B$	Error Amp Input Bias Current	$V_{COMP} = 1.0\text{V}$	100	300/800	300/800	nA nA(max)
$G_M$	Error Amp Transconductance	$I_{COMP} = -30\text{ }\mu\text{A to }+30\text{ }\mu\text{A}$ $V_{COMP} = 1.0\text{V}$	3700	2400/1600 4800/5800	2400/1600 4800/5800	$\mu\text{mho}$ $\mu\text{mho(min)}$ $\mu\text{mho(max)}$
$A_{VOL}$	Error Amp Voltage Gain	$V_{COMP} = 1.1\text{V to }1.9\text{V}$ $R_{COMP} = 1.0\text{ M}\Omega$ (Note 7)	800	500/250	500/250	V/V V/V(min)
	Error Amplifier Output Swing	Upper Limit $V_{FEEDBACK} = 1.0\text{V}$ Lower Limit $V_{FEEDBACK} = 1.5\text{V}$	2.4	2.2/2.0	2.2/2.0	V V(min) V
			0.3	0.40/0.55	0.40/0.55	V V(max)
	Error Amp Output Current	$V_{FEEDBACK} = 1.0\text{V to }1.5\text{V}$ $V_{COMP} = 1.0\text{V}$	$\pm 200$	$\pm 130/\pm 90$ $\pm 300/\pm 400$	$\pm 130/\pm 90$ $\pm 300/\pm 400$	$\mu\text{A}$ $\mu\text{A(min)}$ $\mu\text{A(max)}$
$I_{SS}$	Soft Start Current	$V_{FEEDBACK} = 1.0\text{V}$ $V_{COMP} = 0\text{V}$	5.0	2.5/1.5 7.5/9.5	2.5/1.5 7.5/9.5	$\mu\text{A}$ $\mu\text{A(min)}$ $\mu\text{A(max)}$
D	Maximum Duty Cycle	$V_{COMP} = 1.5\text{V}$ $I_{SWITCH} = 100\text{ mA}$	95	93/90	93/90	% %(min)
$\frac{\Delta I_{SWITCH}}{\Delta V_{COMP}}$	Switch Transconductance		12.5			A/V



## Electrical Characteristics—LM1577-ADJ, LM2577-ADJ (Continued)

Specifications with standard type face are for  $T_J = 25^\circ\text{C}$ , and those in bold type face apply over full Operating Temperature Range. Unless otherwise specified,  $V_{IN} = 5\text{V}$ ,  $V_{FEEDBACK} = V_{REF}$ , and  $I_{SWITCH} = 0$ .

Symbol	Parameter	Conditions	Typical	LM1577-ADJ Limit (Notes 3, 4)	LM2577-ADJ Limit (Note 5)	Units (Limits)
<b>DEVICE PARAMETERS</b>						
$I_L$	Switch Leakage Current	$V_{SWITCH} = 65\text{V}$ $V_{FEEDBACK} = 1.5\text{V}$ (Switch Off)	10	300/600	300/600	$\mu\text{A}$ $\mu\text{A}(\text{max})$
$V_{BAT}$	Switch Saturation Voltage	$I_{SWITCH} = 2.0\text{A}$ $V_{COMP} = 2.0\text{V}$ (Max Duty Cycle)	0.5	0.7/0.9	0.7/0.9	V V(max)
	NPN Switch Current Limit	$V_{COMP} = 2.0\text{V}$	4.3	3.7/3.0 5.3/6.0	3.7/3.0 5.3/6.0	A A(min) A(max)
<b>THERMAL PARAMETERS (All Versions)</b>						
$\theta_{JA}$	Thermal Resistance	K Package, Junction to Ambient	35			°C/W
$\theta_{JC}$		K Package, Junction to Case	1.5			
$\theta_{JA}$		T Package, Junction to Ambient	65			
$\theta_{JC}$		T Package, Junction to Case	2			
$\theta_{JA}$		N Package, Junction to Ambient (Note 8)	85			
$\theta_{JA}$		M Package, Junction to Ambient (Note 8)	100			
$\theta_{JA}$		S Package, Junction to Ambient (Note 9)	37			

**Note 1:** Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating ratings indicate conditions the device is intended to be functional, but device parameter specifications may not be guaranteed under these conditions. For guaranteed specifications and test conditions, see the Electrical Characteristics.

**Note 2:** Due to timing considerations of the LM1577/LM2577 current limit circuit, output current cannot be internally limited when the LM1577/LM2577 is used as a step-up regulator. To prevent damage to the switch, its current must be externally limited to 8.0A. However, output current is internally limited when the LM1577/LM2577 is used as a flyback or forward converter regulator in accordance to the Application Hints.

**Note 3:** All limits guaranteed at room temperature (standard type face) and at temperature extremes (boldface type). All limits are used to calculate Outgoing Quality Level, and are 100% production tested.

**Note 4:** A military RETS electrical test specification is available on request. At the time of printing, the LM1577K-12/893, LM1577K-15/893, and LM1577K-ADJ/893 RETS specifications complied fully with the boldface limits in these columns. The LM1577K-12/893, LM1577K-15/893, and LM1577K-ADJ/893 may also be procured to Standard Military Drawing specifications.

**Note 5:** All limits guaranteed at room temperature (standard type face) and at temperature extremes (boldface type). All room temperature limits are 100% production tested. All limits at temperature extremes are guaranteed via correlation using standard Statistical Quality Control (SQC) methods.

**Note 6:** External components such as the diode, inductor, input and output capacitors can affect switching regulator performance. When the LM1577/LM2577 is used as shown in the Test Circuit, system performance will be as specified by the system parameters.

**Note 7:** A 1.0 M $\Omega$  resistor is connected to the compensation pin (which is the error amplifier's output) to ensure accuracy in measuring  $A_{VOL}$ . In actual applications, this pin's load resistance should be  $\geq 10\text{ M}\Omega$ , resulting in  $A_{VOL}$  that is typically twice the guaranteed minimum limit.

**Note 8:** Junction to ambient thermal resistance with approximately 1 square inch of pc board copper surrounding the leads. Additional copper area will lower thermal resistance further. See thermal model in "Switchers Made Simple" software.

**Note 9:** If the TO-263 package is used, the thermal resistance can be reduced by increasing the PC board copper area thermally connected to the package. Using 0.5 square inches of copper area,  $\theta_{JA}$  is 50°C/W; with 1 square inch of copper area,  $\theta_{JA}$  is 37°C/W; and with 1.8 or more square inches of copper area,  $\theta_{JA}$  is 32°C/W.