



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



**Rapport de projet tutoré 2<sup>e</sup> année :  
Afficheur du régime moteur  
d'un kart électrique**

BRILLET Julien  
TAJUL ARIPIN Mohamad Izzul Imran  
2<sup>ème</sup> année 2010/2011  
Q1

Enseignant  
LEQUEU Thierry  
CHARLES Gliksohn

# Sommaire

Introduction.....	4
1.Présentation.....	5
1.1.Présentation du projet.....	5
1.2.Cahier des charges.....	5
1.3.Schémas fonctionnels.....	6
1.4.Planning.....	7
2.Étude théorique.....	7
2.1.Schéma électrique.....	8
2.2.Fonctionnement.....	8
2.3.Capteur ILS.....	10
2.4.La partie afficheur LEDs.....	11
2.5.Programmation.....	12
3.Réalisation et Test.....	13
3.2.Nomenclature.....	14
3.3.Tests anti-rebond.....	14
3.4.Problèmes rencontrés.....	15
Conclusion.....	17
Résumé.....	18
Index des illustrations.....	19
Bibliographie.....	20
Annexe 1.....	22
Annexe 2.....	23
Annexe 3.....	24
Annexe 4.....	28
Annexe 5.....	29
Annexe 6.....	30
Annexe 7.....	32
Annexe 8.....	38

## **Introduction**

Étant en 2<sup>e</sup> année au département GEII de Tours, nous avons eu la possibilité de réaliser un projet d'Étude et Réalisation allant du 4 Février 2011 jusqu'au 1 Avril 2011. Voici donc notre rapport de projet où nous allons tout d'abord vous présenter l'objectif de notre projet brièvement. Puis, nous aborderons les études théoriques et tests que nous avons effectués. Et enfin, nous traiterons de la réalisation du produit avec les problèmes rencontrés. Par manque de temps, le projet n'est pas fini, donc le programme et le typon n'ont pas passé de test final, en espérant qu'à l'oral, on pourra vous montrer un produit fini.

# 1. Présentation

## 1.1. Présentation du projet

Lors de ce quatrième semestre, nous devons réaliser un projet en Études et Réalisations. Mon binôme et moi-même avons décidé de choisir comme projet un afficheur de régime moteur à LEDs.

Cette afficheur permet d'indiquer le régime moteur actuel par rapport au régime max de façon visuelle, et simple sur un groupement de LEDs. Ce système est utilisé sur les voitures de course pour indiquer au pilote le bon moment pour changer de rapport de vitesse.

## 1.2. Cahier des charges

- alimentation : 12V (Batterie du kart)
- bornier pour l'alimentation
- Micro-contrôleur ATmega8535
- Capteur magnétique ILS<sup>1</sup>
- LED<sup>2</sup> : 8 à 12
- Environnement : extérieur
  - humidité : possible
  - température : ambiante : -10°C à 40°C
  - pression : atmosphérique : 1 bar
- Plage de vitesse (0 – 5000 tr/min) [moteur]

---

<sup>1</sup> ILS : Interrupteur à lame souple.

<sup>2</sup> LED : *Light Emitting Diode*. DEL : Diode Électro-luminescente

## 1.3. Schémas fonctionnels

### 1.3.1. Schéma fonctionnel de niveau 1

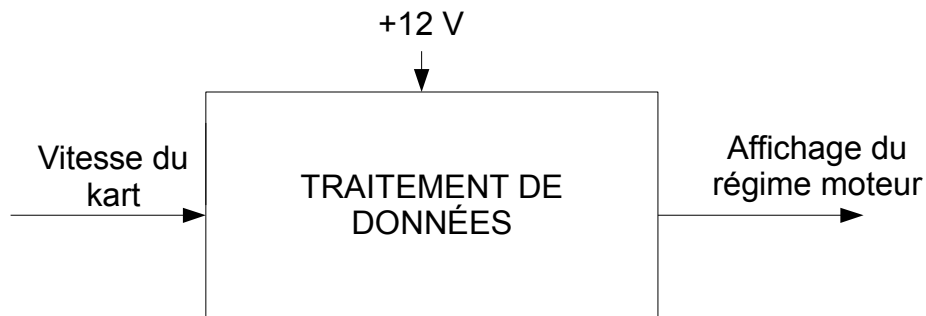


Illustration 1: Schéma fonctionnel de niveau 1

Le système utilise la vitesse de rotation des roues pour en déduire le régime actuel, et la comparer à la valeur maximum atteignable. La comparaison est ensuite affichée sur un bandeau de LEDs, permettant ainsi de montrer simplement le niveau du régime moteur.

### 1.3.2. Schéma fonctionnel de niveau 2

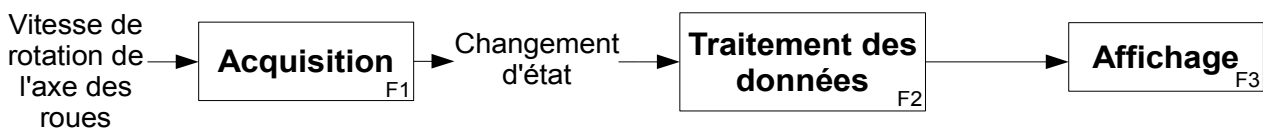


Illustration 2: Schéma fonctionnel de niveau 2

Description des fonctions principales :

#### F1 : Acquisition

Pour connaître le régime du moteur, on a fait une analogie entre la vitesse de rotation de l'axe des roues motrices et la vitesse de rotation du moteur. On détecte la vitesse de rotation de l'axe des roues grâce à un capteur ILS, et deux aimants fixés sur l'axe des roues<sup>3</sup>. Donc à chaque fois que l'axe fait un tour, le capteur est actionné deux fois.

<sup>3</sup> Pour mieux comprendre le principe, voir la partie ILS

## **F2 : Traitement des données**

C'est un microprocesseur qui effectue le traitement des données. Grâce au changement d'état du capteur ILS, on peut déterminer la vitesse de rotation de l'axe des roues. Étant donné que l'axe des roues est lié à l'axe du moteur, et qu'il existe qu'un seul lien entre les deux axes (pas de principe de boîte de vitesse), la vitesse de rotation des deux axes est donc proportionnels. Donc pour connaître la donnée recherchée, on compare le régime actuel du moteur et la valeur max atteignable.

## **F3: Affichage**

Elle est composée d'un groupement de 12 LEDs, et d'un écran LCD.

Les LEDs permet de représenter le niveau du régime moteur sur un ligne. Cela se trouve très pratique puisque les informations les plus importants y sont montrer : le régime moteur actuel, le régime max (fin de la ligne de LEDs).

L'écran LCD peut aller jusqu'à 16 caractères maximum par lignes. De plus il possède 4 lignes maximums (16\*4). Il permet d'afficher des données annexes qui servent dans la phase de test, mais des données utiles, servant dans l'utilisation finale, peuvent y être écrient si besoin est, comme par exemple le vitesse du kart.

## **1.4. Planning**

Pour voir le planning, voir annexe 1

Nous avons eu de graves retards dû aux tests du programme qui s'avère être non concluant à chaque fois. De plus, des problèmes imprévus sont survenus pas nos tests, ce qui ne nous facilitait pas la tâche sur l'exactitude du bon fonctionnement de la carte.

## **2. Étude théorique**

Nous débuterons par commenter chaque fonctionnement des montages puis nous détaillerons l'utilité des principaux composants.

## 2.1. Schéma électrique

Voir annexe 1 pour voir le schéma de la carte.

Le schéma comporte 3 parties :

Une alimentation, c'est une alimentation à découpage pour abaisser la tension.

Un micro-contrôleur, composé d'un ATMega.

Un afficheur LEDs, elle contient 12 LEDs.

## 2.2. Fonctionnement

### 2.2.1. L'alimentation

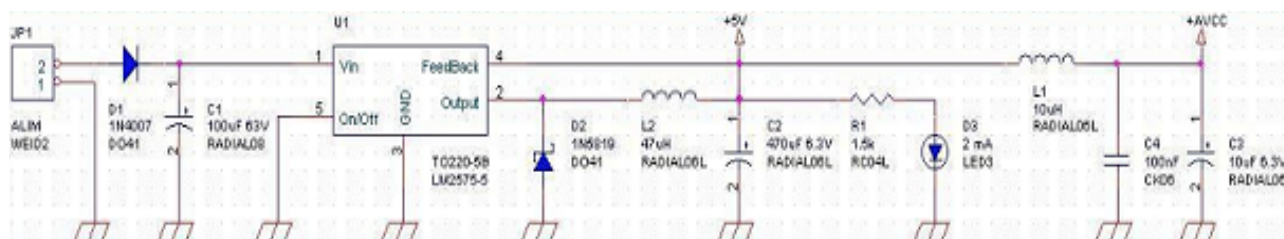


Illustration 3: Schéma de l'alimentation

Elle fournit de manière permanente l'énergie électrique à l'objet technique.

- Entrée : tension continue 24V.
- Sortie : tension continue 5V

L'alimentation est composée principalement d'un régulateur LM2575-5V permettant de stabiliser une tension à une valeur fixe, dans notre cas 5V. Elle est composée d'un ensemble de résistances, condensateurs, bobine, diode. C'est une alimentation à découpage.[1]

Enfin une led D3 de couleur rouge, est présente pour informer l'utilisateur si la carte est alimentée ou non.

Cette alimentation permet d'obtenir également une tension +AVCC, pour alimenter le micro contrôleur présent sur la carte. Dans notre cas, cette tension vaut +15 Volts.



## 2.2.2. Le micro-contrôleur ATmega8535

C'est un circuit intégré. Il possède deux mémoires:

- une mémoire vive<sup>4</sup> pour les données. C'est une mémoire informatique dans laquelle l'ordinateur stocke ces données. Elle est volatile, c'est à dire que lorsque l'alimentation est coupée, toutes les données sont perdues.

- une mémoire morte<sup>5</sup> pour les programmes. C'est une mémoire non volatile. La mémoire ne s'efface pas, même lorsqu'il n'y a pas d'alimentation. Ce type de mémoire ne peut pas être programmée et effacée par les utilisateurs.

La programmation se fait dans notre cas, par un connecteur. Les ports de d'entrés et sorties sont indispensables au composant. Ces périphériques permettent de traiter les données envoyées par le clavier, ainsi que de produire des sorties de types logiques pour commander l'écran LCD de la carte.[1]

Le montage ATmega comporte un quartz. C'est une horloge qui permet de le cadencer le micro-contrôleur, en oscillant à une fréquence stable, ici 16MHz. Il est essentiel pour le bon fonctionnement. Il est accompagné de deux condensateurs de 22pF. Ces condensateurs lui permettent d'osciller dans de bonne condition. Plus le micro-contrôleur fonctionne vite, plus il consomme. Cette horloge offre une vitesse d'exécution largement suffisante.

L'ATmega8535 comporte également un circuit Reset, qui possède deux fonctions. Premièrement, lors de la mise sous tension, il s'assure que la patte Reset du micro-contrôleur ne s'active pas tant que la tension d'alimentation n'a pas atteint un seuil défini et n'est pas stabilisée durant un certain temps de très courte durée.

Deuxièmement lors du fonctionnement, il s'assure que la tension d'alimentation ne passe pas sous un seuil défini. Si toutefois c'est le cas, le circuit intégré désactive la patte Reset et ne la réactive qu'avec les mêmes conditions de la mise sous tension.[1]

---

4 Appelée également **RAM** (*Random Access Memory*)

5 Appelée également **ROM** (*Read-Only Memory*)

### 2.3. Capteur ILS

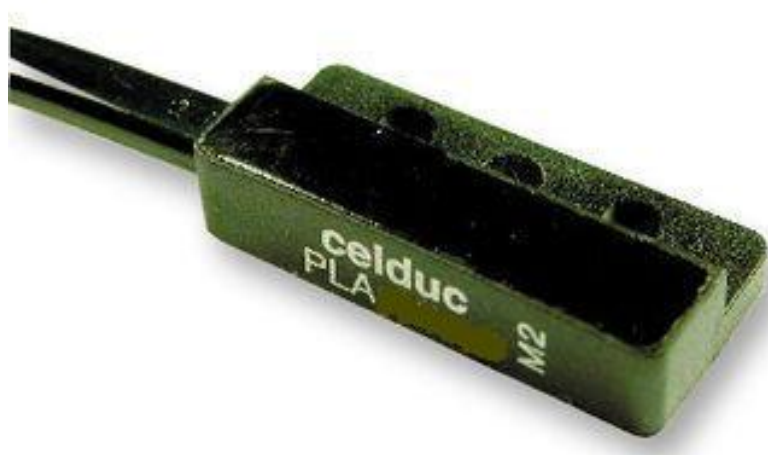


Illustration 4: Capteur ILS[2]

Nous avons choisi un capteur à lame souple puisqu'il permet de détecter des champs magnétiques à proximité, et ce dans tous les angles. De plus, ce choix est surtout dû au fait que toutes les autres solutions ne sont pas envisageables :

- La roue codeuse avec barrière lumineuse : solution fiable mais difficile à mettre en place sur le kart.
- Le roulement à bille intelligent : très fiable mais aussi très coûteux

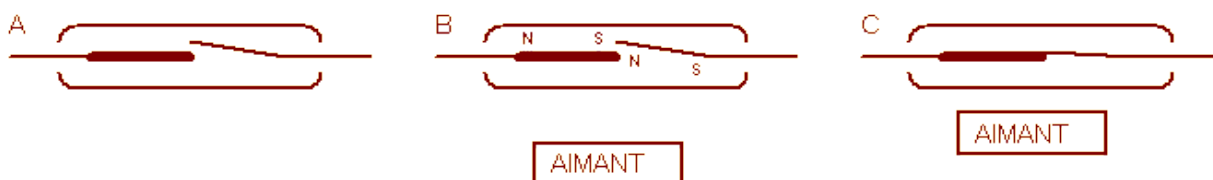


Illustration 5: Principe de fonctionnement d'un ILS[3]

- Lorsque l'aimant est éloigné de l'ILS, le contact de celui-ci est ouvert.
- Lorsque l'aimant, encore éloigné, s'approche de l'ampoule, les deux lames de l'ILS sont soumises au champ magnétique. Chaque lame se comporte alors comme un aimant. La force d'attraction entre les deux lames n'est pas encore suffisante pour que la lame souple se déforme.
- L'aimant est suffisamment proche de l'ILS. La lame souple se déforme et entre en contact avec la lame rigide. Le contact électrique est établi.

Le constructeur fournit la distance maximale qui doit exister entre l'aimant afin d'établir le contact. Un environnement métallique réduit considérablement cette distance (huisserie métallique par exemple). La distance d'établissement du contact (fermeture de l'issue) est toujours supérieure à celle de rupture (ouverture).

Il a une excellente tenue dans le temps. Par contre, le contact est sensible aux chocs et aux vibrations[2].

## 2.4. La partie afficheur LEDs

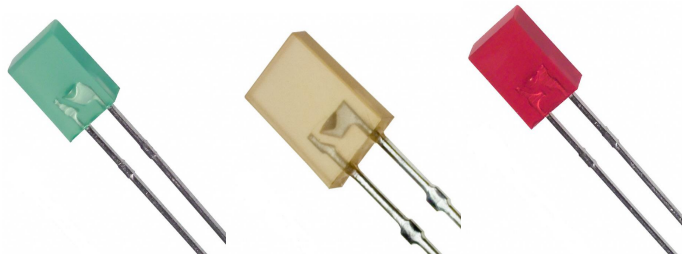


Illustration 6: Les LEDs : verte, orange, rouge[4]

Pour afficher le régime du moteur, 3 couleurs différentes sont utilisés pour indiquer les 3 étapes lorsque le moteur tourne, supposons :



Illustration 7: Bande de LEDs

- verte : bas régime ( bon moment pour rétrograder)
- orange : intermédiaire
- rouge : régime max (bon moment pour passer la vitesse supérieur)

Deux ULN2803A (voir annexe) sont utilisés pour augmenter la puissance émis par le micro-contrôleur pour alimenter les LEDs. Un ULN2803A est composé de 8 transistors en Darlington[6]. Un transistor en Darlington est une combinaison de deux transistors bipolaires<sup>6</sup> de même type.

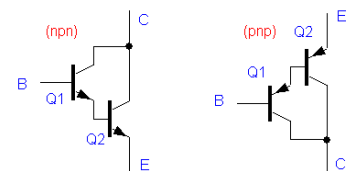


Illustration 8: Transistor de Darlington[5]

<sup>6</sup> Transistor utilisant 3 zones de polarité inversé (+ -). Il en existe 2 types : NPN(- + -) et PNP (+ - +)

## 2.5. Programmation

Pour calculer le régime moteur, on a programmé le microprocesseur à l'aide du logiciel AVR Studio. Cela permettra de vérifier le code, et de gérer l'affichage de l'écran LCD.

Principe de fonctionnement:

- Comptage du temps entre chaque tour :  
Pour chronométrer le temps que met l'axe à faire un tour, on utilise une interruption interne qui s'active qu'après un certain temps donné, dans notre cas 10  $\mu$ s.  
Ensuite, grâce à une interruption externe, à chaque tour on récupère le temps mis, on initialise la variable, et on calcule le régime moteur. Mais ici, l'axe aura 2 aimants, donc l'ILS s'activera deux fois par tour.
- Calculer le régime moteur : On s'appuie sur le temps récupéré pour calculer la vitesse en tr/s. Delà, on compare la vitesse actuel et la vitesse max, le résultat est en pourcentage. Suivant le pourcentage du régime, nous déterminons le nombre de LED à allumer.
- Affichage: Le microprocesseur écrit sur le LCD les résultats des calculs intermédiaires, et allume le nombre de LED demandé grâce au résultat final (nombre de LED).

Pour mieux comprendre le fonctionnement du programme voici l'ordinogramme:

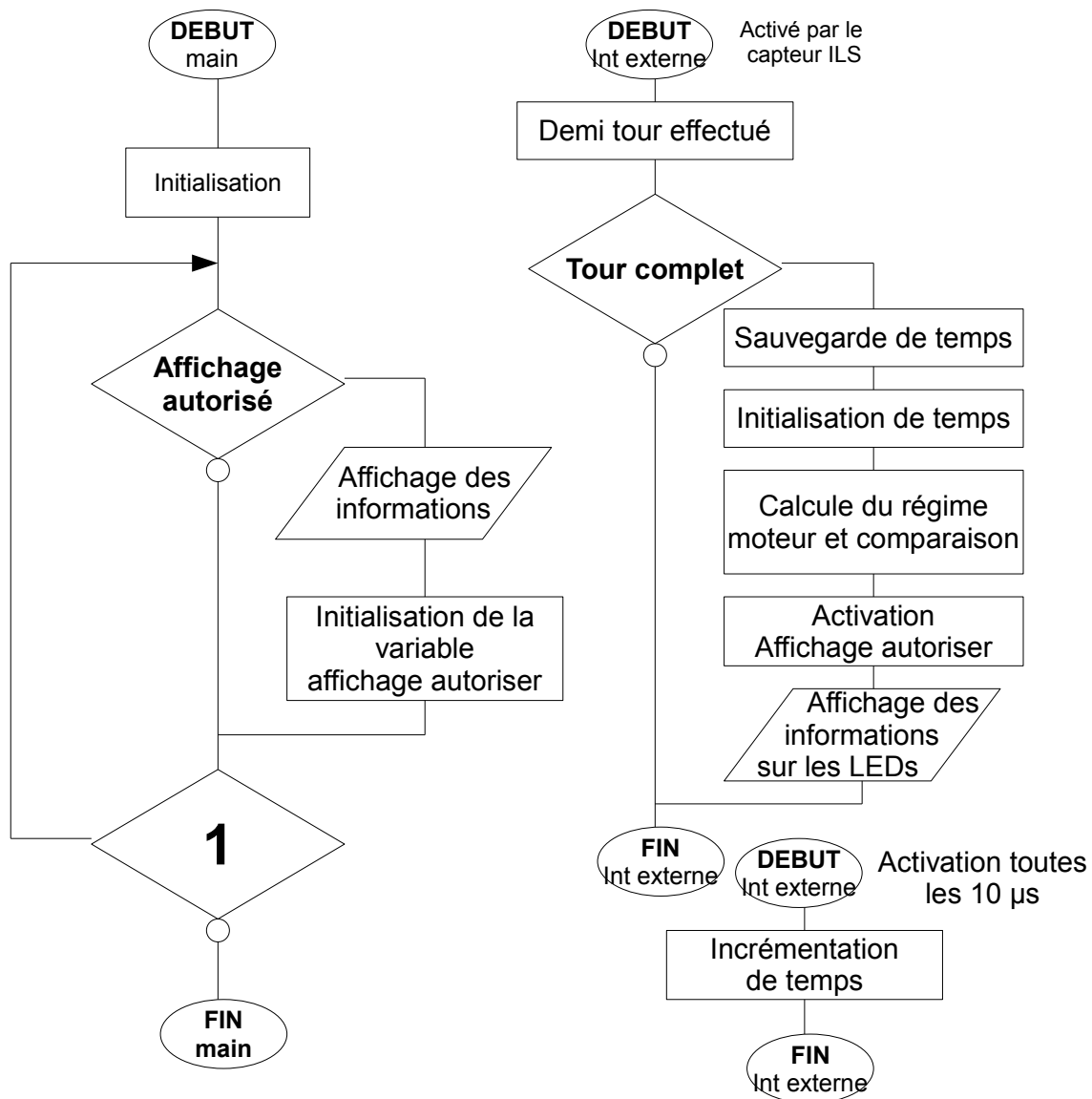


Illustration 9: Ordinoigramme du programme implanté dans le micro-contrôleur






















































### 3. Réalisation et Test

Dans cette partie, nous allons réaliser les différents typons grâce au logiciel OrCad présent dans les ordinateurs de la salle d'Étude et Réalisation. De plus nous allons effectuer plusieurs tests, pour vérifier le bon fonctionnement des montages.

#### 3.1.1. Typon

Voir annexe 3 pour le typon de la carte.

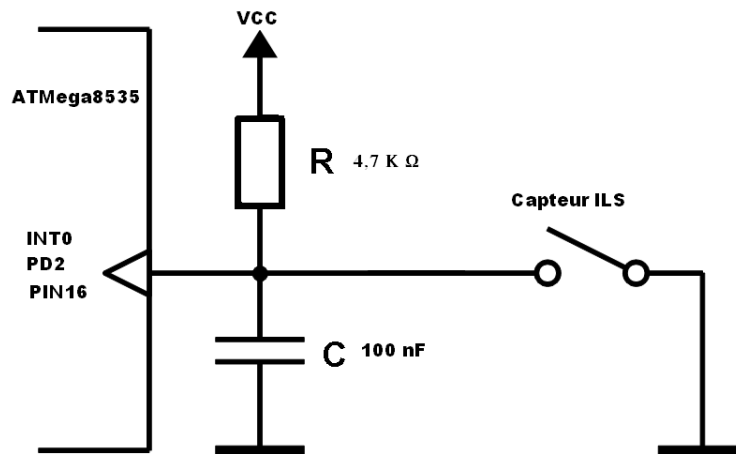
## 3.2. Nomenclature

1	S.,AFF1	LCD 16x4	29	 .JP2	CON ISP
2	 .C1	100uF 63V	30	 .JP8	HEADER 2
3	 .C2	470uF 6.3V	31	 .L1	10uH
4	 .C3	10uF 6.3V	32	 .L2	47uH
5	 .C4	100nF	33	 .P1	10K
6	 .C5	10uF 6.3V	34	 .Q1	16 MHz
7	 .C6	100nF	35	 .R1	1.5k
8	 .C7	22pF	36	 .R4	4.7k
9	 .C8	22pF	37	 .R5	1.5k
10	 .C9	100nF	38	 .R6	10k
11	 .C10	100nF	39	 .R7	150
12	 .D1	1N4007	40	 .R8	150
13	 .D2	1N5819	41	 .R9	150
14	 .D3	2 mA	42	 .R10	150
15	 .D4	3mm	43	 .R11	150
16	 .D5	LED	44	 .R12	150
17	 .D6	LED	45	 .R13	150
18	 .D7	LED	46	 .R14	150
19	 .D8	LED	47	 .R15	150
20	 .D9	LED	48	 .R16	150
21	 .D10	LED	49	 .R17	150
22	 .D11	LED	50	 .R18	150
23	 .D12	LED	51	 .J1	LM2575-5
24	 .D13	LED	52	 .J2	ATmega8535
25	 .D14	LED	53	 .J3	ULN2803
26	 .D15	LED	54	 .J4	ULN2803
27	 .D16	LED			
28	 .JP1	ALIM			

*Illustration 10: Nomenclature*

## 3.3. Tests anti-rebond

En observant sur un oscilloscope, Nous avons observé qu'il y avait de forts rebonds lorsque le moteur tourne avec une certaine vitesse. Pour le résoudre, un filtre RC (filtre anti-rebond) est ajouté, il s'agit d'ajouter un condensateur en parallèle avec le capteur.



*Illustration 11: Filtre RC (anti-rebond)*

Ne connaissant pas la tension des perturbations ni leurs fréquences, il était plus judicieux de tester les condensateurs en observant le résultat que de faire des calculs avec des données manquantes. On a testé 3 condensateurs différents : 1 nF, 10 nF, 100 nF.

Pour voir les oscillogrammes des test voir annexe 7.

A la fin, le condensateur le plus adapté c'est révélé être le 100 nF.

### **3.4. Problèmes rencontrés**

Lors de ce projet, nous avons rencontré des problèmes qui à ralenti l'avancement du projet. Ils sont apparus lors des tests de la programmation et des tests du capteur.

#### **3.4.1. Programmation**

D'abord, nous voulions calculer la nombres de tours et la pourcentage du régime moteur atteint. Pour cela, nous avons utilisé une bibliographie `<math.hpp>` , sauf que la programmation embarqué n'accepte pas cette bibliographie.

Ensuite, pour les afficher sur LCD, initialement nous avons utilisé par défaut la fonction « `lcd_puts( ) ;` » qui en fait n'accepte pas les variables du type réel ou entier. Alors, elle a été remplacé par la fonction « `sprintf( ) ;` » .

De plus, le micro-contrôleur redémarre tout seul en plein fonctionnement lorsque le capteur change d'état. Par l'expérience vécu par d'autres groupes, ce problème n'est pas un cas isolé. Il se pourrait que la carte ait des problèmes de C.E.M.<sup>7</sup>.

### **3.4.2. Capteur ILS**

Le débogage de la programmation n'apparaissait pas suffisant pour afficher correctement le régime moteur sur les LEDs. Nous nous sommes rendu compte qu'il y avait aussi des problèmes avec le capteur ILS produisant des rebonds. Comme la programmation calcule le régime moteur par rapport aux front descendants, les rebonds ont provoqués des calculs erronés. Voir le chapitre « Tests anti-rebond ».

---

<sup>7</sup> Compatibilité Électro-Magnétique



## Conclusion

Au début du projet, l'indicateur de changement de vitesse devait être utilisé en complément avec une simulation de course de voiture, mais vu les contraintes matériels et budgétaire, on a préféré se tourner vers quelque chose de plus concret, donc le kart de M. Lequeu. Et pour cela nous avons utilisé différents composants.

Nous avons choisi un micro-contrôleur ATmega8535 afin de programmer la carte pour effectuer le traitement des données récupérées par le capteur, et de les afficher sur le LCD et sur les LEDs. Ce qui a confirmé notre choix, a été le fait que l'établissement possède plusieurs documents, et schéma de ce micro-contrôleur, réalisé par M. Thierry Lequeu. Ensuite, on trouve que le capteur ILS n'est pas adapté à cette utilisation, puisqu'on le sollicite à des intervalles de temps très courts. Il a été conçu à la base pour détecter des fins de course de vérin, mais dans notre utilisation, le capteur a provoqué des fluctuations de signal, ce qui nous a fortement ralenti.

Enfin dans l'ordre général, mon binôme et moi avons apprécié cette expérience, ceci pour plusieurs raisons. D'abord, ce projet nous a permis de mettre en pratique nos connaissances et compétences apprises lors de notre formation. Il nous a permis, entre autres, de revoir certaines bases d'informatique embarquée vues au deuxième semestre. Puis, nous avons dû travailler en équipe de deux, supervisé par un professeur. Cette situation peut ressembler à celle rencontrée en entreprise. On aura acquis une certaine expérience qui sera utile lors du stage de fin d'année.

## Résumé

Pour ce projet tutoré du semestre 4, nous avons décidé de choisir une application avec un afficheur à LEDs, qui permet d'indiquer le régime moteur par rapport au maximum qu'il peut attendre. Cette étude met en évidence des bases d'informatique embarqués et de l'électronique.

Pour la réalisation de la carte, on a pris comme base la carte de test de M. Lequeu. De plus, on nous a fourni une petite plaquette avec un ban de 8 LEDs, et une maquette avec un petit moteur, pour commencer rapidement la programmation.

Au départ, le projet semblé simple et facile, mais il s'est révélé être plus compliqué que nous l'avions pensé. Le capteur ILS nous a dérangé au niveau de la programmation et nous a ralenti. De plus, la carte de test de M. Lequeu semblé être défaillant puisque quand on utilisait le programme, le micro-contrôleur redémarre de façon intempestive quand le capteur est actionné.

Enfin, pour conclure ce projet, il nous reste à tester le système dans les conditions réel de fonctionnement du kart, c'est-à-dire de monter l'afficheur sur le kart et vérifier le bon fonctionnement pendant un tour de test.

## Index des illustrations

Illustration 1: Schéma fonctionnel de niveau 1.....	6
Illustration 2: Schéma fonctionnel de niveau 2.....	6
Illustration 3: Schéma de l'alimentation.....	8
Illustration 4: Capteur ILS[2].....	10
Illustration 5: Principe de fonctionnement d'un ILS[3].....	10
Illustration 6: Les LEDs : verte, orange, rouge[4].....	11
Illustration 7: Bande de LEDs.....	11
Illustration 8: Transistor de Darlington[5].....	11
Illustration 9: Ordinogramme du programme implanté dans le micro-contrôleur.....	13
Illustration 10: Nomenclature.....	14
Illustration 11: Filtre RC (anti-rebond).....	15

Les illustrations 1, 2, 3, 7, 9, 10, 11 sont des réalisations personnelles

## Bibliographie

[1] : **Lequeu Thierry**, "*La documentation de Thierry LEQUEU sur OVH*", (consulté le 25 mars 2011), <<http://www.thierry-lequeu.fr/>>.

[2] : **celduc**, "*Capteur ILS*", (consulté le 25 mars 2011), <<http://www.celduc.com>>.

[3] : **Geea**, "*Principe ILS*", (consulté le 25 mars 2011), <<http://www.geea.org/spip.php?article210>>.

[4] : **digkey**, "*Led rectangulaire*", (consulté le 25 mars 2011), <<http://media.digkey.com/photos/Lumex%20Photos/SSL-LX2573GD,SSL-LX2573PGD.jpg>>.

[5] : **Hades Mech Northwestern**, "*Transistor de Darlington*", (consulté le 30 mars 2011), <[http://hades.mech.northwestern.edu/images/0/03/Darlington\\_pairs.gif](http://hades.mech.northwestern.edu/images/0/03/Darlington_pairs.gif)>.

[6] : **inconnu**, "*datasheet uln2803A*", (consulté le 25 mars 2011), <<http://www.st.com/stonline/books/pdf/docs/1536.pdf>>.

# **Annexe**

# Annexe 1

## Planning prévisionnel et réel

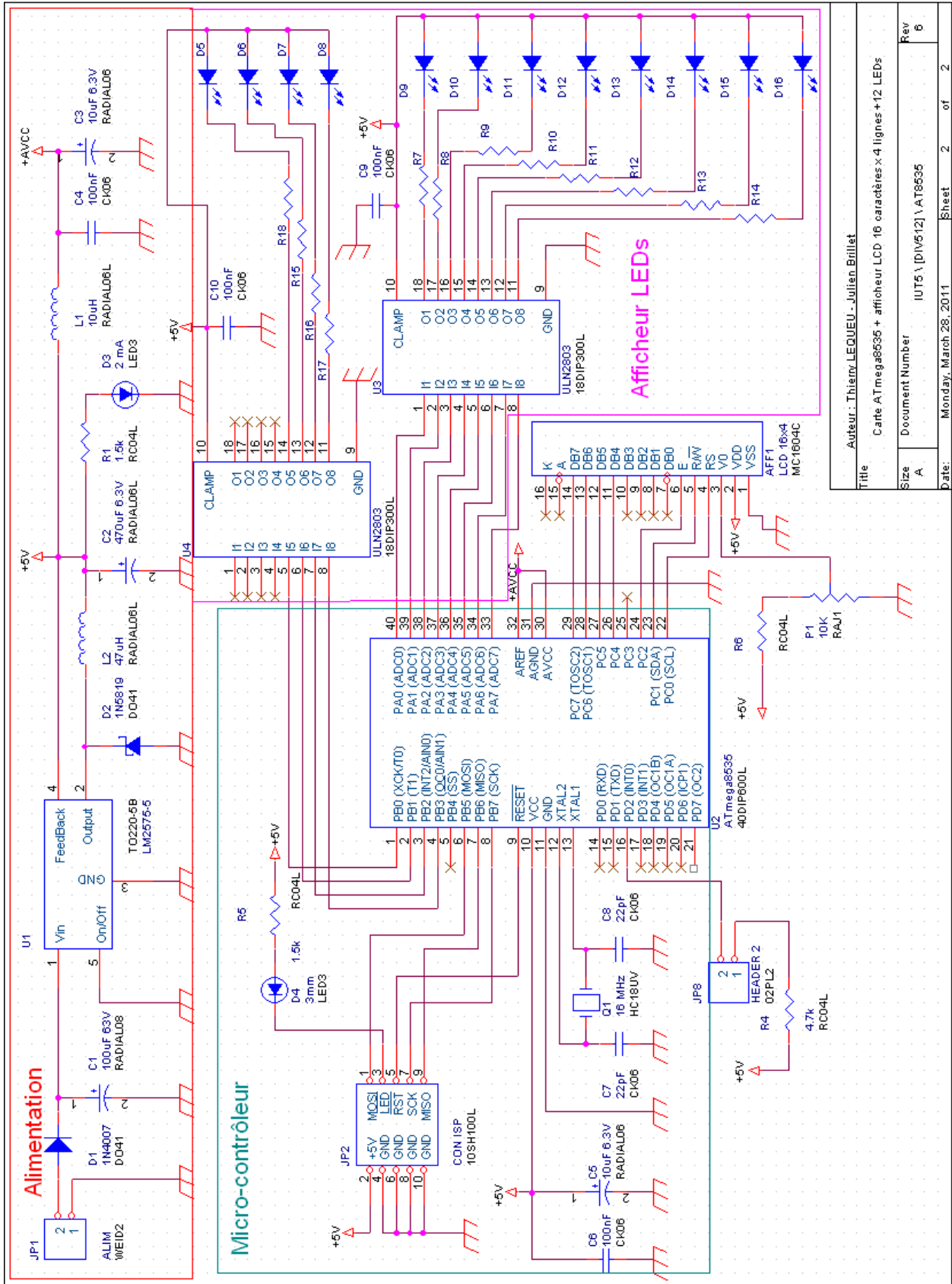
Semaine	5	6	7	8	9	10	11	12	13	14
Choix du projet & Découverte du projet	Prévision			Vacances	Vacances					
	Réel									
Élaboration du déroulement du projet	Prévision			Vacances	Vacances					
	Réel									
Achat des composants		Prévision		Vacances	Vacances					
Programmation		Prévision	Prévision	Vacances	Vacances	Prévision	Prévision	Prévision	Prévision	
		Réel	Réel			Réel	Réel	Réel	Réel	
Test		Prévision	Prévision	Vacances	Vacances	Prévision	Prévision	Prévision	Prévision	
		Réel	Réel			Réel	Réel	Réel	Réel	
Élaboration des schémas électroniques et des typons			Prévision	Vacances	Vacances	Prévision	Prévision			
						Réel	Réel	Réel	Réel	
Réalisation des cartes				Vacances	Vacances	Prévision	Prévision			
									Réel	
Résolution des problèmes éventuelles		Prévision	Prévision	Vacances	Vacances	Prévision	Prévision	Prévision	Prévision	
		Réel	Réel			Réel	Réel	Réel	Réel	
Synthèse du projet		Prévision	Prévision	Vacances	Vacances	Prévision	Prévision	Prévision		
		Réel	Réel			Réel	Réel	Réel		
Rendre le rapport				Vacances	Vacances				Prévision	
									Réel	
Oral				Vacances	Vacances					Prévision
										Réel

### Légende:

<b>Prévision</b>
<b>Réel</b>
<b>Vacances</b>

# Annexe 2

## Schémas électriques de la carte



## Annexe 3

### Programmation de 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 : afficheur régime moteur  
Version : 0.9  
Date : 11/02/2011  
Author : juju  
Company : d'orange  
Comments:  
béta 1 fonctionnement avec comptage du nombre de tours en 1s
```

```
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>
```

```
#define TEMPS_PAUSE 0  
#define LED 12
```

```
unsigned short Compt_implu=0;  
unsigned volatile short Affichage_LCD=0;  
unsigned volatile int Nb_led,temps;  
unsigned char Phrase[16];  
unsigned char Pour_cent='%';  
unsigned volatile int Regime_moteur;  
volatile float Nb_tours;
```

```
// External Interrupt 0 service routine// External Interrupt 0 service routine  
interrupt [EXT_INT0] void ext_int0_isr(void)  
{  
    /*if(temps>=TEMPS_PAUSE)*/ Compt_implu++;  
    if(Compt_implu>=2)  
    {  
        Affichage_LCD=1;  
  
        Nb_tours=(1/temps);
```



```

Regime_moteur = ((Nb_tours*60)/5000.0)*100.0 ; // calcule du régime moteur en pourcentage

Nb_led=(Regime_moteur/100.0)*LED;

switch(Nb_led)
{
    case 0:
PORTB=0x00;          PORTA=0x00;
                    break;

    case 1:
PORTB=0x80;          PORTA=0x00;
                    break;

    case 2:
PORTB=0xC0;          PORTA=0x00;
                    break;

    case 3:
PORTB=0xE0;          PORTA=0x00;
                    break;

    case 4:
PORTB=0xF0;          PORTA=0x00;
                    break;

    case 5:
PORTB=0xF0;          PORTA=0x80;
                    break;

    case 6:
PORTB=0xF0;          PORTA=0xC0;
                    break;

    case 7:
PORTB=0xF0;          PORTA=0xE0;
                    break;

    case 8:
PORTB=0xF0;          PORTA=0xF0;
                    break;

    case 9:
PORTB=0xF0;          PORTA=0xF8;
                    break;

    case 10:
PORTB=0xF0;          PORTA=0xFC;

```

```

        break;

        case 11:
            PORTB=0xF0;
            PORTA=0xFE;
            break;

        case 12:
            PORTB=0xF0;
            PORTA=0xFF;
            break;
    }
    temps=0;
    Compt_implu=0;
}
GIFR=0;
}

// Timer 0 output compare interrupt service routine
interrupt [TIM0_COMP] void timer0_comp_isr(void)
{
    temps++;
}

void main(void)
{
    // Input/Output Ports initialization
    // Port A initialization
    PORTA=0x00;
    DDRA=0xFF;
    // Port B initialization
    PORTB=0x00;
    DDRB=0xF0;
    // Port C initialization
    PORTC=0x00;
    DDRC=0xFF;
    // Port D initialization
    PORTD=0x00;
    DDRD=0x00;

    // Timer/Counter 0 initialization
    // Clock source: System Clock
    // Mode: Normal top=FFh
    TCCR0=0x02; //diviser la fréquence par 8 donc f=2MHz, T=0,5 µs
    OCR0=20; // 20 fronts donc F= 100 Khz, T=10 µs

    // External Interrupt(s) initialization
    // INT0: On
    // INT0 Mode: Falling Edge
    // INT1: Off
    // INT2: Off
    GICR|=0x40;
    MCUCR=0x02;
    MCUCSR=0x00;
    GIFR=0x40;

    // Timer(s)/Counter(s) Interrupt(s) initialization
    TIMSK=0x02;

```

```

// LCD module initialization
lcd_init(16);

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

/* switch to writing in Display RAM */
lcd_gotoxy(0,0);
lcd_putsf("Afficheur regime moteur");
delay_ms(1000);
lcd_clear();
lcd_gotoxy(4,1);
lcd_putsf("START !!");
delay_ms(1000);
Nb_tours=0;
Compt_implu=0;
Regime_moteur=0;
Nb_led=0;

while (1)
{

    if(Affichage_LCD)
    {
        lcd_clear();
        lcd_gotoxy(0,0); //
        sprintf(Phrase,"Nb tr/s: %d",Nb_tours/2); //
        lcd_puts(Phrase); //
        // affichage des information vitesse de rotation sur le LCD
        lcd_gotoxy(0,1); //
        sprintf(Phrase, "Nb tr/min: %d", Nb_tours*30); //
        lcd_puts(Phrase); //

        lcd_gotoxy(0,2);
        sprintf(Phrase,"Reg moteur: %d%c",Regime_moteur,Pour_cent); //affichage l'information régime
        // moteur sur le LCD
        lcd_puts(Phrase);

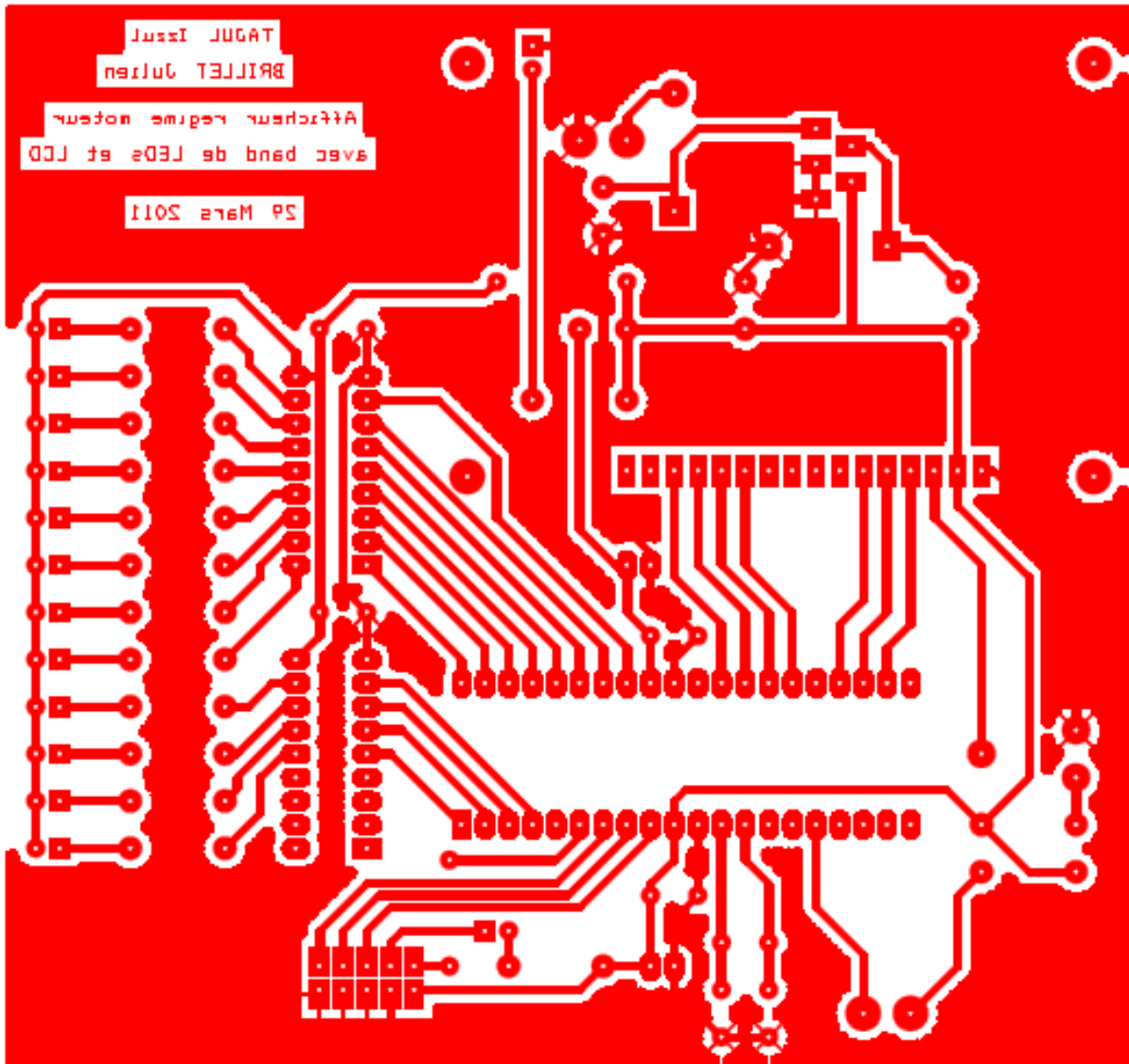
        lcd_gotoxy(0,3);
        sprintf(Phrase,"Nb LED: %d",Nb_led); //affichage l'information nombre de led sur le LCD
        lcd_puts(Phrase);

        Affichage_LCD=0;
    }
}
}

```

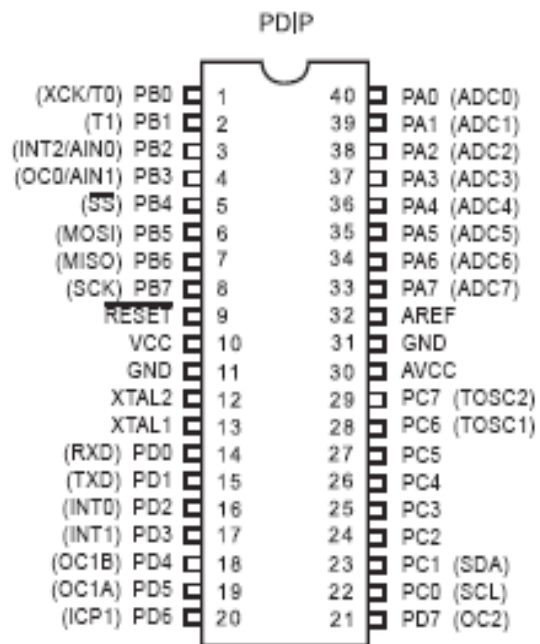
## Annexe 4

Typon de la cartes:



## Annexe 5

Configuration des broches de l'ATMega8535:

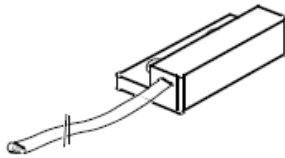


# Annexe 6

Datasheet de l'PLA13725

P/STD/PLA137 \* /D/22/01/2002

page 1 / 2 F/GB

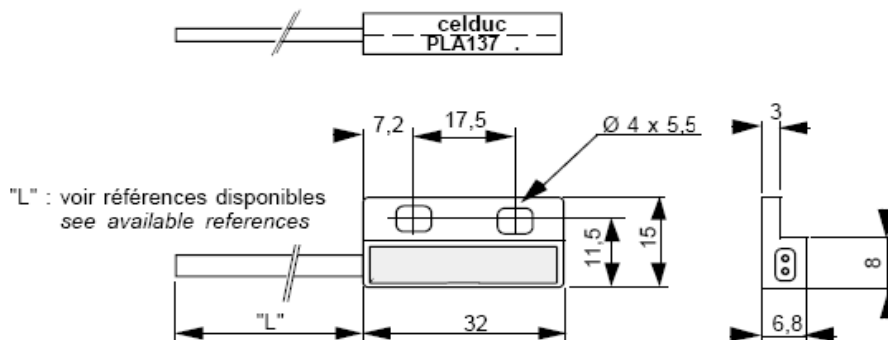


# PLA137 ..

## CAPTEUR DE PROXIMITE MAGNETIQUE - un contact travail

MAGNETIC PROXIMITY SENSOR - 1 normally open contact

### Dimensions / Outlines



### CARACTERISTIQUES ELECTRIQUES / ELECTRICAL DATA

<b>Puissance maxi commutable / Max. switching power</b>	<b>10 W / 12 VA</b>
<b>Tension maxi commutable / Max. switching voltage</b>	<b>250 Vdc</b>
<b>Courant maxi commutable / Max. switching current</b>	<b>400 mA</b>
<b>Courant de travail / Carrying current</b>	<b>1 A</b>
<b>Tension typique de tenue entre lames / Typical dielectric strength between contacts</b>	<b>250 Veff</b>
<b>Résistance de contact initiale ( hors cable ) / Initial contact resistance ( cable excepted )</b>	<b>≤ 150 mΩ</b>
<b>Résistance d'isolement ( min ) / Insulation resistance</b>	<b>10<sup>11</sup> Ω</b>
<b>Capacité / Capacitance</b>	<b>0,5 pF</b>
<b>Fréquence maxi d'utilisation / Max.frequency</b>	<b>200 Hz</b>
<b>Fréquence de résonance / Resonant frequency</b>	<b>2300 Hz</b>

Proud to serve you

**celduc**<sup>®</sup>  
r e l a i s

All technical characteristics are subject to change without previous notice.  
Caractéristiques sujettes à modification sans préavis.

**Temps de commutation**

à l'action (rebonds 0,6 ms inclus) / <i>Operate time (bounces 0,6 ms included)</i>	<b>1 ms</b>
au relachement / <i>Release time</i>	<b>0,1 ms</b>

**CARACTERISTIQUES PHYSIQUES / PHYSICAL DATA**

<b>Matériau utilisé pour le contact</b> / <i>Contact material</i>	<b>Rhodium</b>
<b>Température maxi d'utilisation</b> / <i>Max. operating temperature</i>	<b>- 40 à + 100°C</b>
<b>Température de stockage</b> / <i>Storage temperature</i>	<b>- 55 à + 100°C</b>
(température donnée hors câble / <i>given temperature, cable excepted</i> )	
<b>Chocs (11 ms)</b> / <i>Shocks (11 ms)</i>	<b>100 g</b>
<b>Vibrations (de 10 à 2000 Hz)</b> / <i>Vibrations</i>	<b>30 g</b>
<b>Durée de vie mécanique</b> / <i>Mechanical life expectancy</i>	<b>10<sup>9</sup></b>
<b>Indice de protection</b> (nettoyage par jet d'eau; usage industriel et ambiance poussiéreuse) <i>Protection level (Cleaning with jet of water, industrial use, dusty environment)</i>	<b>IP67</b>

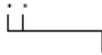
**AUTRES CARACTERISTIQUES / OTHERS**

<b>Plage de sensibilité utilisée</b> / <i>Used pull-in range</i>	<b>de 25 à 45 ATf</b>
Pour d'autres sensibilités, nous consulter / <i>For others, please contact us</i>	
<b>Cable méplat</b> / <i>Flattened electrical cable</i>	<b>5,6 mm x 3,4 mm - 2 x 0,5 mm<sup>2</sup> type H03VVF</b>
Pour d'autres longueurs ou types, nous consulter / <i>For other lengths or typical wires, please contact us</i>	

**Détection / Detection**

Commande par aimant / <i>Ordered by magnet</i>	<b>Type P6250000 ou / or P4200000 ou / or P4160000</b>
Nota : dimensions P625 et P420 identiques au PLA ... <i>Please note : P625, P420 and PLA ... own same dimensions</i>	
Pour d'autres types ou utilisations, nous consulter / <i>For others, please contact us</i>	

**Références disponibles / Available references**

PLA137 ..  
  
**Longueur (dm) / Length (dm)**  
 01 --> 1 dm de câble / *with cable 1 dm*  
 10 - 15 - 25 - 30 - 40 - 80



ISO 9001  
N° 1993/1106a

**celduc**<sup>®</sup>  
r e l a i s

www.celduc.com

Rue Ampère B.P. 4 42290 SORBIERS - FRANCE E-Mail : celduc-relais@celduc.com  
 Fax +33 (0) 4 77 53 85 51 Service Commercial France Tél. : +33 (0) 4 77 53 90 20  
 Sales Dept.For Europe Tel. : +33 (0) 4 77 53 90 21 Sales Dept. Asia : Tél. +33 (0) 4 77 53 90 19

# Annexe 7

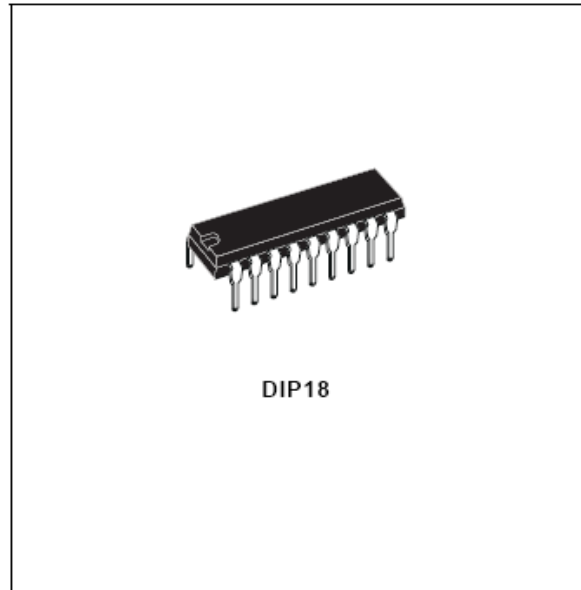
Datasheet de L'ULN2803



ULN2801A  
ULN2802A - ULN2803A  
ULN2804A - ULN2805A

## EIGHT DARLINGTON ARRAYS

- EIGHT DARLINGTONS WITH COMMON EMITTERS
- OUTPUT CURRENT TO 500 mA
- OUTPUT VOLTAGE TO 50 V
- INTEGRAL SUPPRESSION DIODES
- VERSIONS FOR ALL POPULAR LOGIC FAMILIES
- OUTPUT CAN BE PARALLELED
- INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY BOARD LAYOUT



DIP18

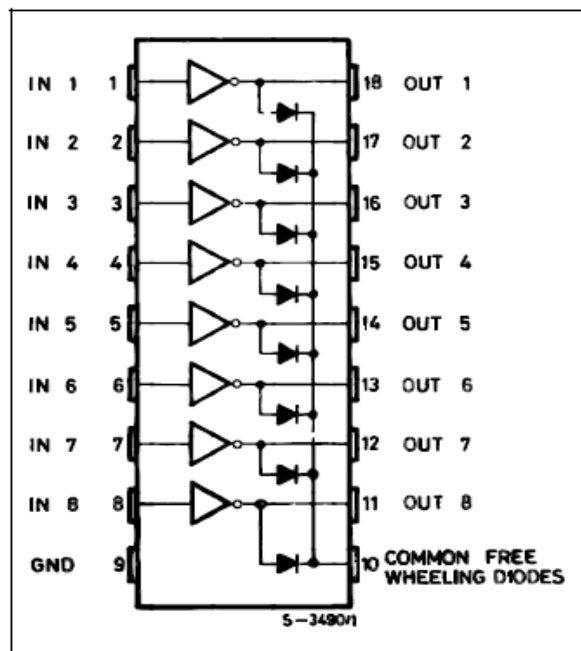
### DESCRIPTION

The ULN2801A-ULN2805A each contain eight darlington transistors with common emitters and integral suppression diodes for inductive loads. Each darlington features a peak load current rating of 600mA (500mA continuous) and can withstand at least 50V in the off state. Outputs may be paralleled for higher current capability.

Five versions are available to simplify interfacing to standard logic families : the ULN2801A is designed for general purpose applications with a current limit resistor ; the ULN2802A has a 10.5kΩ input resistor and zener for 14-25V PMOS ; the ULN2803A has a 2.7kΩ input resistor for 5V TTL and CMOS ; the ULN2804A has a 10.5kΩ input resistor for 6-15V CMOS and the ULN2805A is designed to sink a minimum of 350mA for standard and Schottky TTL where higher output current is required.

All types are supplied in a 18-lead plastic DIP with a copper lead from and feature the convenient input-opposite-output pinout to simplify board layout.

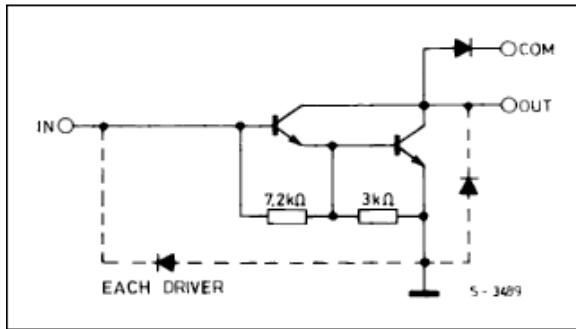
### PIN CONNECTION (top view)



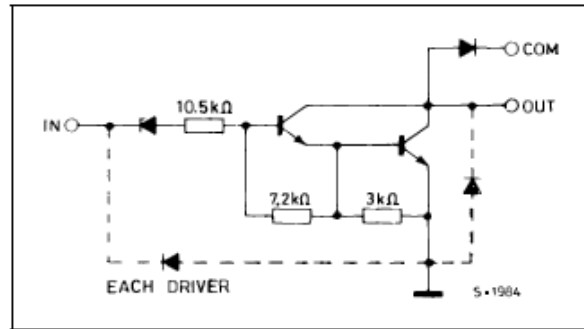


**SCHEMATIC DIAGRAM AND ORDER CODES**

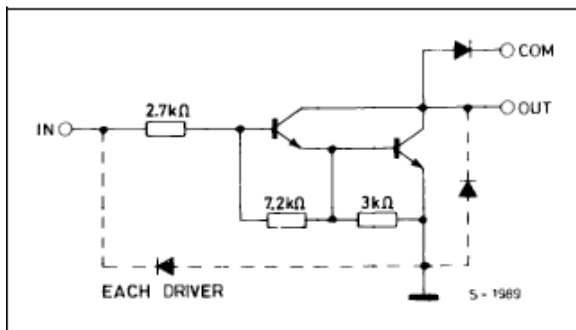
For ULN2801A (each driver for PMOS-CMOS)



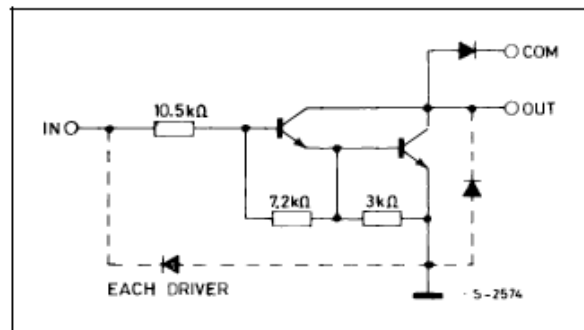
For ULN2802A (each driver for 14-15 V PMOS)



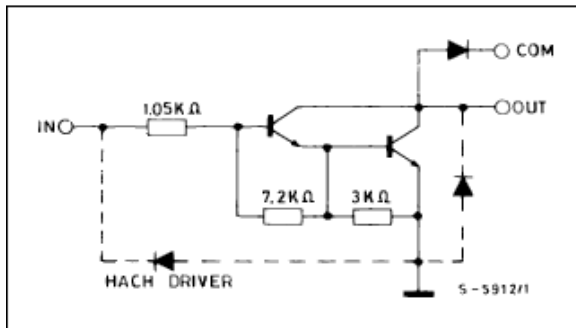
For ULN2803A (each driver for 5 V, TTL/CMOS)



For ULN2804A (each driver for 6-15 V CMOS/PMOS)



For ULN2805A (each driver for high out TTL)



**ABSOLUTE MAXIMUM RATINGS**

Symbol	Parameter	Value	Unit
$V_o$	Output Voltage	50	V
$V_i$	Input Voltage for ULN2802A, UL2803A, ULN2804A for ULN2805A	30 15	V
$I_c$	Continuous Collector Current	500	mA
$I_B$	Continuous Base Current	25	mA
$P_{tot}$	Power Dissipation (one Darlington pair) (total package)	1.0 2.25	W
$T_{amb}$	Operating Ambient Temperature Range	- 20 to 85	°C
$T_{stg}$	Storage Temperature Range	- 55 to 150	°C
$T_j$	Junction Temperature Range	- 20 to 150	°C

**THERMAL DATA**

Symbol	Parameter	Value	Unit
$R_{th\ j-amb}$	Thermal Resistance Junction-ambient Max.	55	°C/W

**ELECTRICAL CHARACTERISTICS** ( $T_{amb} = 25^\circ\text{C}$  unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit	Fig.
$I_{CEX}$	Output Leakage Current	$V_{CE} = 50\text{V}$ $T_{amb} = 70^\circ\text{C}$ , $V_{CE} = 50\text{V}$			50	$\mu\text{A}$	1a
		$T_{amb} = 70^\circ\text{C}$ for ULN2802A			100	$\mu\text{A}$	1a
		$V_{CE} = 50\text{V}$ , $V_i = 6\text{V}$ for ULN2804A			500	$\mu\text{A}$	1b
		$V_{CE} = 50\text{V}$ , $V_i = 1\text{V}$			500	$\mu\text{A}$	1b
$V_{CE(sat)}$	Collector-emitter Saturation Voltage	$I_c = 100\text{mA}$ , $I_B = 250\mu\text{A}$		0.9	1.1	V	2
		$I_c = 200\text{mA}$ , $I_B = 350\mu\text{A}$		1.1	1.3	V	
		$I_c = 350\text{mA}$ , $I_B = 500\mu\text{A}$		1.3	1.6	V	
$I_{i(on)}$	Input Current	for ULN2802A $V_i = 17\text{V}$		0.82	1.25	mA	3
		for ULN2803A $V_i = 3.85\text{V}$		0.93	1.35	mA	
		for ULN2804A $V_i = 5\text{V}$		0.35	0.5	mA	
		$V_i = 12\text{V}$		1	1.45	mA	
		for ULN2805A $V_i = 3\text{V}$		1.5	2.4	mA	
$I_{i(off)}$	Input Current	$T_{amb} = 70^\circ\text{C}$ , $I_c = 500\mu\text{A}$	50	65		$\mu\text{A}$	4
$V_{i(on)}$	Input Voltage	$V_{CE} = 2\text{V}$ for ULN2802A			13	V	5
		$I_c = 300\text{mA}$ for ULN2803A			2.4	V	
		$I_c = 200\text{mA}$			2.7	V	
		$I_c = 250\text{mA}$			3	V	
		$I_c = 300\text{mA}$ for ULN2804A			5	V	
		$I_c = 125\text{mA}$			6	V	
		$I_c = 200\text{mA}$			7	V	
		$I_c = 275\text{mA}$			8	V	
		$I_c = 350\text{mA}$ for ULN2805A			2.4	V	
		$I_c = 350\text{mA}$					
$h_{FE}$	DC Forward Current Gain	for ULN2801A $V_{CE} = 2\text{V}$ , $I_c = 350\text{mA}$	1000			-	2
$C_i$	Input Capacitance			15	25	pF	-
$t_{PLH}$	Turn-on Delay Time	$0.5 V_i$ to $0.5 V_o$		0.25	1	$\mu\text{s}$	-
$t_{PHL}$	Turn-off Delay Time	$0.5 V_i$ to $0.5 V_o$		0.25	1	$\mu\text{s}$	-
$I_R$	Clamp Diode Leakage Current	$V_R = 50\text{V}$			50	$\mu\text{A}$	6
		$T_{amb} = 70^\circ\text{C}$ , $V_R = 50\text{V}$			100	$\mu\text{A}$	6
$V_F$	Clamp Diode Forward Voltage	$I_F = 350\text{mA}$		1.7	2	V	7

TEST CIRCUITS

Figure 1a.

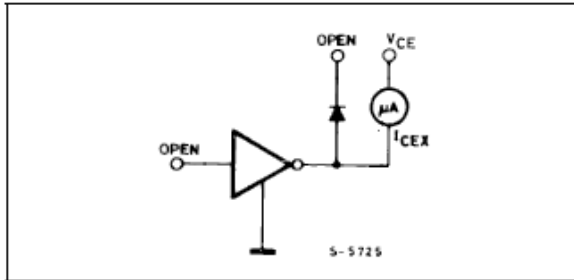


Figure 1b.

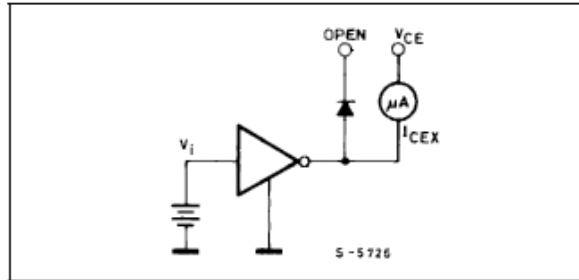


Figure 2.

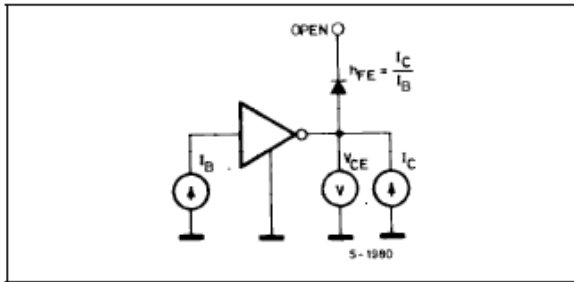


Figure 3.

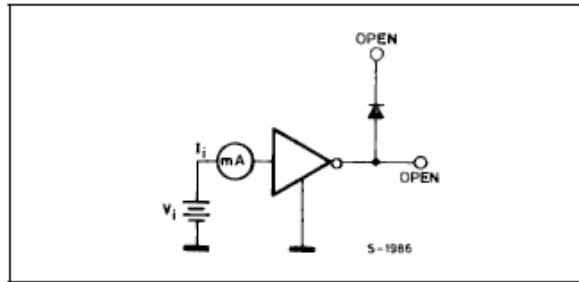


Figure 4.

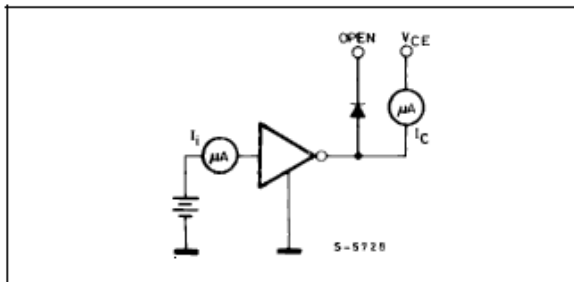


Figure 5.

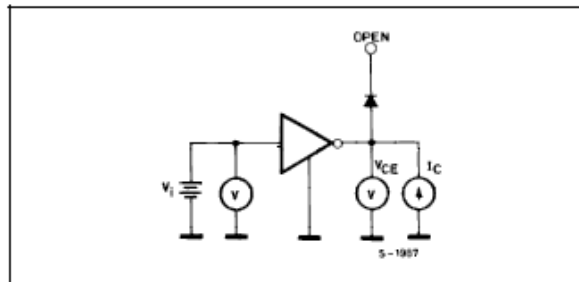


Figure 6.

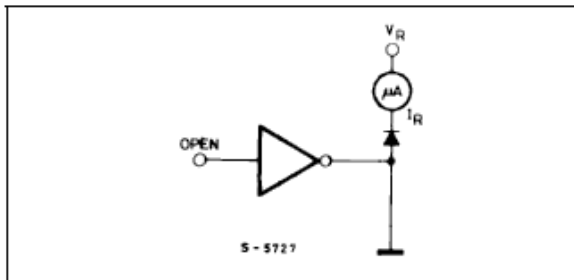
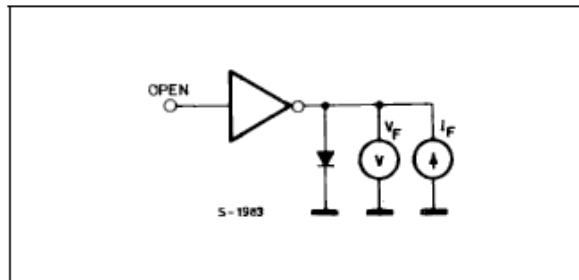
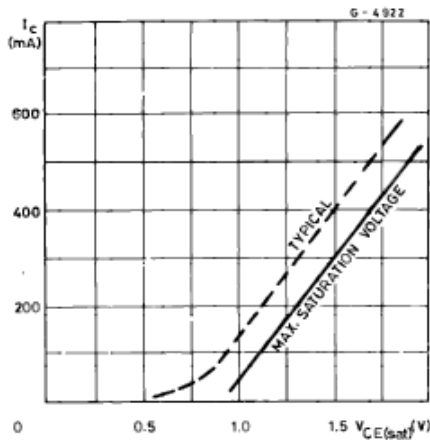


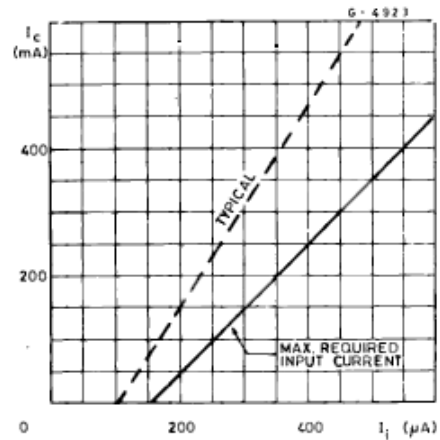
Figure 7.



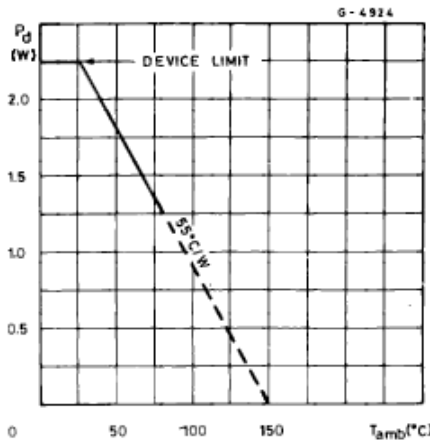
**Figure 8 :** Collector Current as a Function of Saturation Voltage.



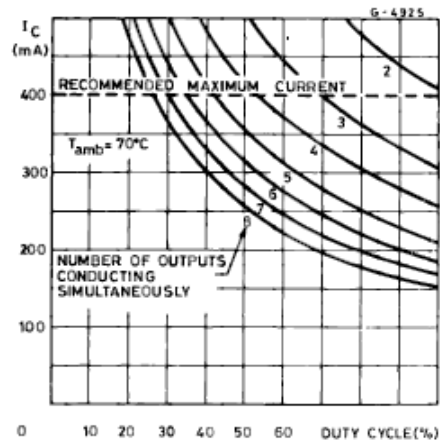
**Figure 9 :** Collector Current as a Function of Input Current.



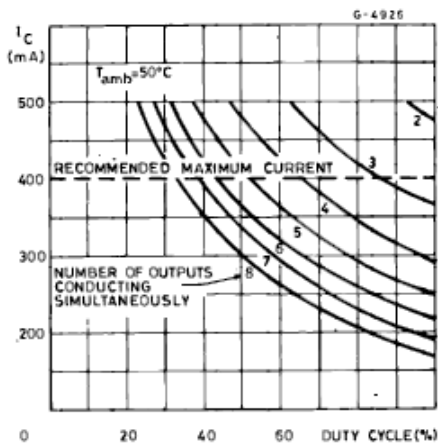
**Figure 10 :** Allowable Average Power Dissipation as a Function of Ambient Temperature.



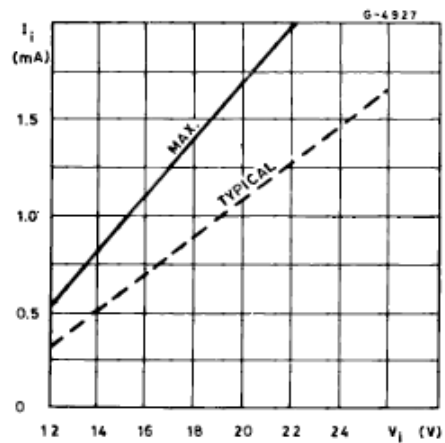
**Figure 11 :** Peak Collector Current as a Function of Duty Cycle.



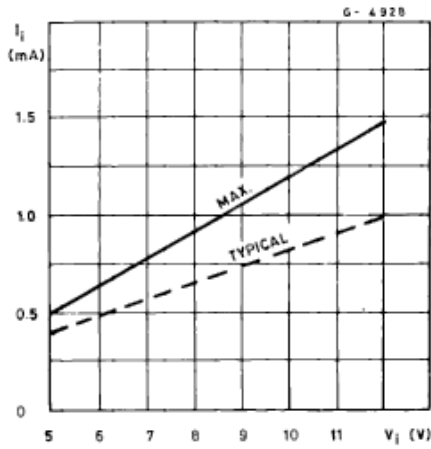
**Figure 12 :** Peak Collector Current as a Function of Duty.



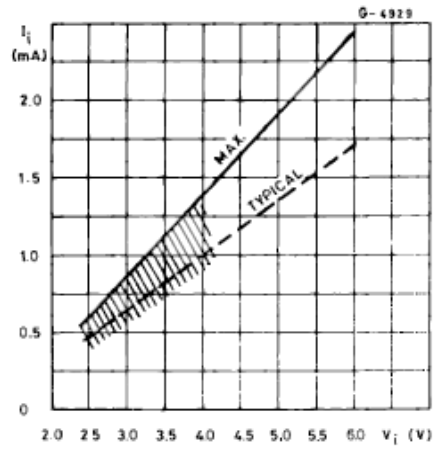
**Figure 13 :** Input Current as a Function of Input Voltage (for ULN2802A).



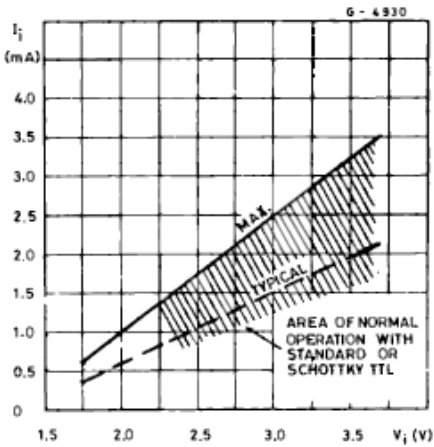
**Figure 14 :** Input Current as a Function of Input Voltage (for ULN2804A)



**Figure 15 :** Input Current as a Function of Input Voltage (for ULN2803A)



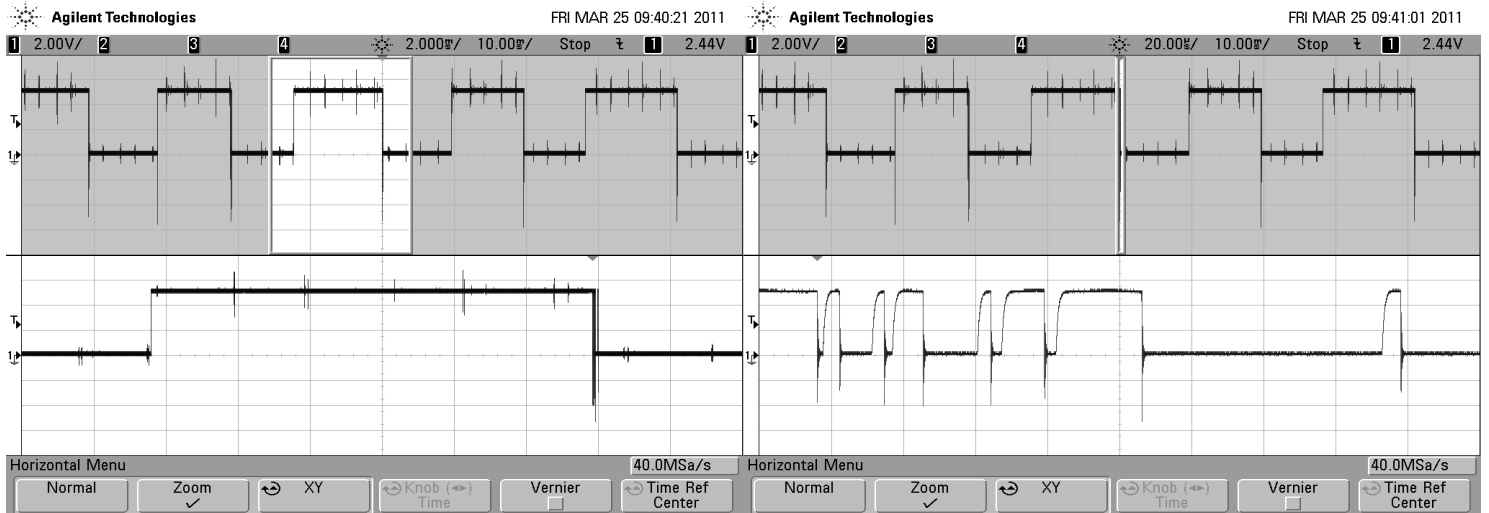
**Figure 16 :** Input Current as a Function of Input Voltage (for ULN2805A)



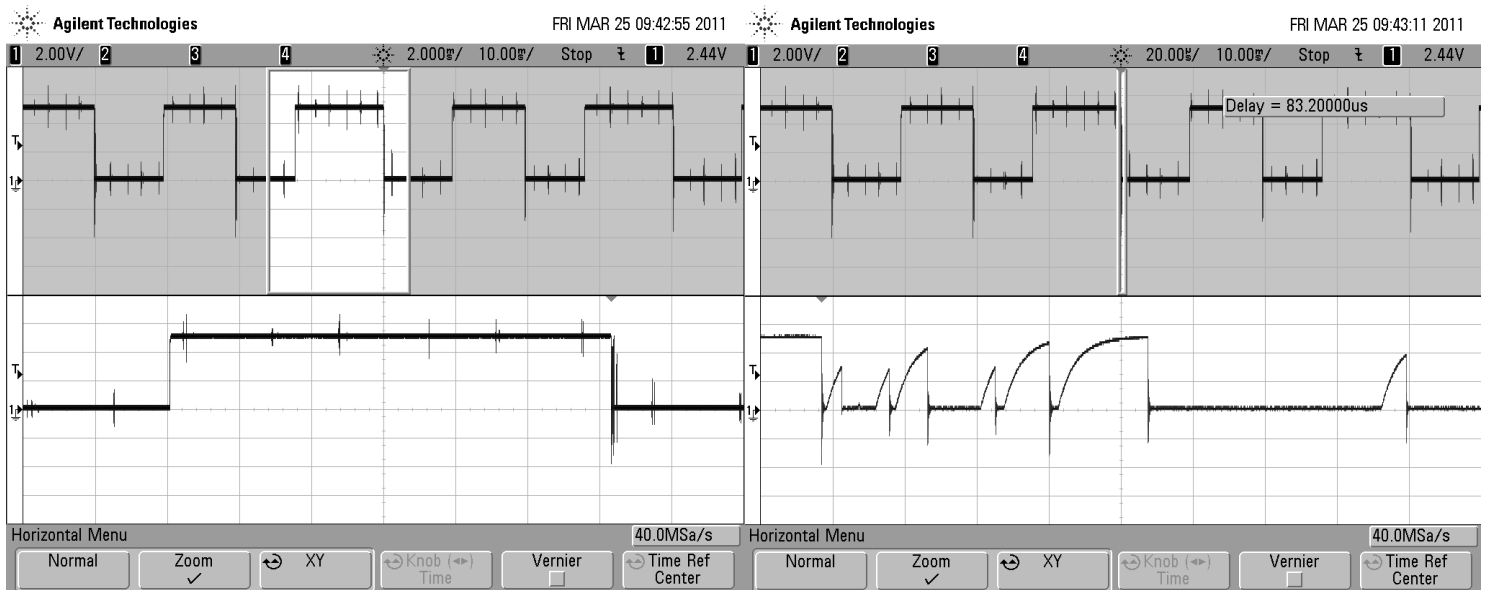
# Annexe 8

Oscillogrammes des tests :

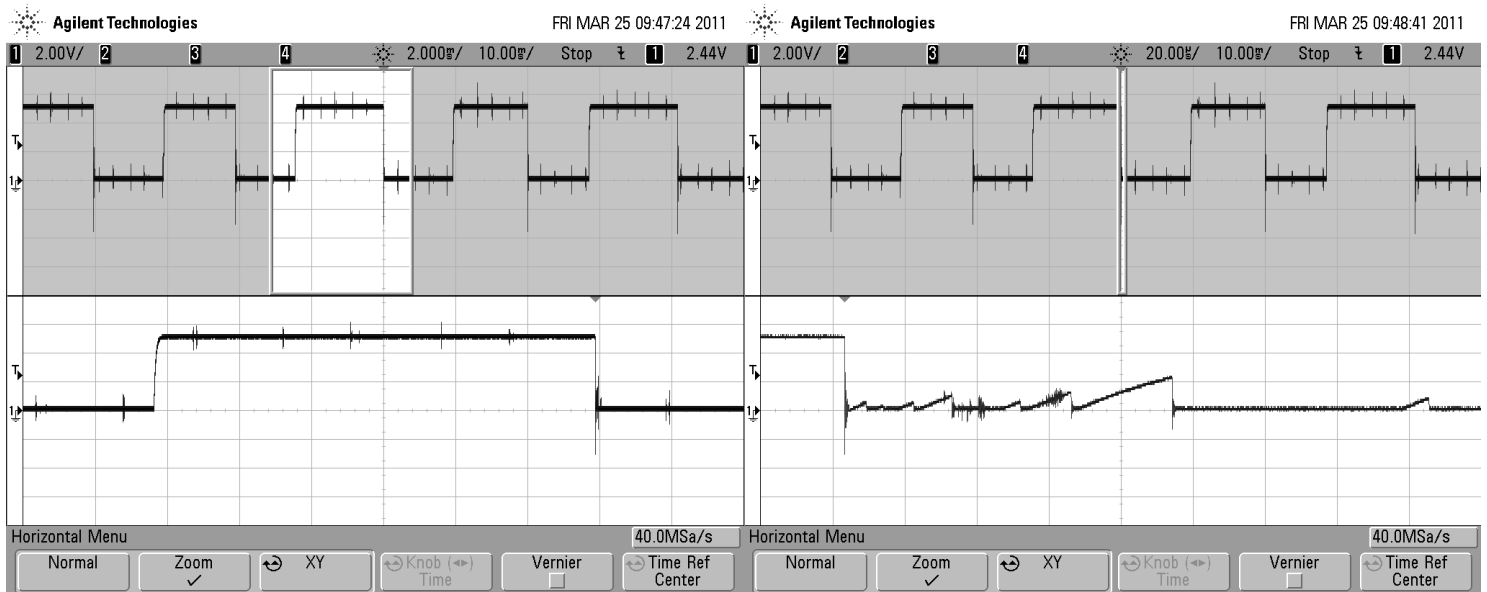
## Test sans condensateur



## Test avec un condensateur de 1 nF



## Test avec un condensateur de 10 nF



## Test avec un condensateur de 100 nF

