

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



Logo déformé

Projet d'Étude et Réalisation  
du semestre quatre

# Lu.Co.Cla.M

Oui  
LuCoClam suffirait ?  
*Lumière Commandée par Claquement de Main*



Amadou Tidiane FAYE  
Boubacar NDACK  
Groupe P1  
Promotion 2011/2013

Enseignants :  
Thierry LEQUEU  
Sofi RODIER

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



Projet d'Étude et Réalisation  
du semestre quatre

# Lu.Co.Cla.M

*Lumière Commandée par Claquement de Main*

Amadou Tidiane FAYE  
Boubacar NDACK  
Groupe P1  
Promotion 2011/2013

Enseignants :  
Thierry LEQUEU  
Sofi RODIER

## Sommaire

Introduction.....	4
1. Cahier des charges.....	5
1.1.Présentation du projet.....	5
1.2.Origine du projet.....	5
1.3.Contraintes et spécifications.....	5
2.Analyse technique du projet.....	6
2.1.Détecteur sonore.....	6
2.2.Partie amplificateur.....	7
2.3.Partie filtrage.....	10
2.4.Partie programmation.....	11
3.Suivi de projet.....	15
3.1.Problème rencontré.....	15
3.2.planning.....	15
Conclusion.....	16
Résumé.....	17
Bibliographie.....	19

## Index illustr

## Annexes ?

Aérer

Mettre en gras + espace TM1

Justifié le texte, tout le tps  
Utiliser index des illustr pour y faire  
référence dans dev [7]

# Introduction

En deuxième année du DUT Génie Électrique et Informatique Industrielle (GEII) à l'IUT de Tours, l'équipe enseignante a mis en place un projet d'étude et réalisation pour le semestre quatre. Les étudiants du GEII devront mener ce projet en binôme ou seul.

Ce travail a pour objectif de nous préparer à la rédaction de notre rapport de stage et à notre soutenance que nous effectuerons en fin de cycle. C'est de plus, l'occasion de nous familiariser avec la conduite de projets au travers de la mise en œuvre de certaines méthodes de travail et de certains outils de formalisation, directement inspirés des pratiques des entreprises industrielles.

Pour ce projet, nous avons voulu réaliser une Lumière Commandée par Claquement de Main, **Lu.Co.Cla.M**. Celui-ci serait constitué d'un capteur audio, d'un amplificateur audio mais aussi d'un filtre passe-bande qui permettraient d'isoler le claquement de main des autres sons captés et enfin d'une partie programmation.

On retrouvera dans ce dossier un cahier des charges, une analyse technique du projet, un planning des séances d'études et réalisations et les schémas fonctionnels utilisés pour l'élaboration de **Lu.Co.Cla.M**. Une partie, concernant les tests effectués et le suivi du projet avec les difficultés rencontrées, sera aussi mentionnée.



Présentée

Texte doit être justifié.

# 1. Cahier des charges

[Introduire cette partie](#)

## 1.1. Présentation du projet

Notre projet a pour but la réalisation d'un éclairage commandé par le claquement des mains. Nous avons décidé de l'intituler Lu.Co.Cla.M pour Lumière Commandée par Claquement de Main. Pour ce faire deux claquements de main provoque l'allumage de la lumière et un claquement correspond à l'extinction de l'éclairage.

## 1.2. Origine du projet

Le projet d'étude et de réalisation du semestre quatre devait s'appuyer en grande partie sur la programmation. Ainsi passionnés par la domotique et à tous ceux qui touchent à l'éclairage en général, l'idée de concevoir l'allumage d'une LED par claquement des mains nous est venue naturellement.

## 1.3. Contraintes et spécifications

Le projet d'étude et de réalisation du semestre quatre devait s'appuyer en grande partie sur la programmation. Or pour la réalisation de Lu.Co.Cla.M nous pouvions adopter une solution électronique ne nécessitant pas de programmation. La solution électronique serait plus simple à réaliser.

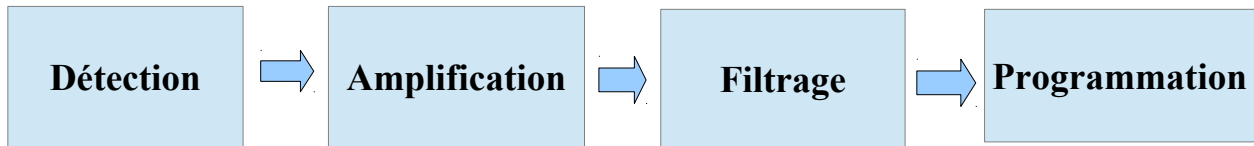
De plus n'ayant jamais utiliser le logiciel Code Vision AVR, nous avons dû rapidement maîtriser ce logiciel pour mener à bien ce projet.

[Faire un renvoi à 24 page 28](#)

*Alinéa ?*

## 2. Analyse technique du projet

Pour réaliser Lu.Co.Cla.M, nous avons divisé le travail en quatre parties. Il s'agit d'une partie détection du claquement de main, une partie amplification du signal, d'une partie filtrage et enfin d'une partie programmation.



### *←* 2.1. Détecteur sonore

*0/2*  
Pour détecter les claquements de mains, nous avons besoin d'un détecteur sonore. Nous avons décidé d'utiliser un microphone comme capteur sonore. Nous avons utilisé un microphone à Electret qui a un principe de fonctionnement strictement identique à celui du microphone électrostatique, accessible au niveau du magasin de l'IUT.

Le microphone à Electret est un microphone doté d'un composant nommé Electret assimilable à un condensateur. Il a la particularité d'être polarisée de façon permanente au moment de sa fabrication. En théorie elle est permanente, mais en pratique, la polarisation diminue au fil du temps, provoquant ainsi une baisse lente de la sensibilité du microphone.

Le composant Electret interne du microphone a une impédance de sortie très élevée, ainsi on ne peut pas y connecter directement une charge fortement capacitive ou d'impédance très faible.

Pour cette raison, la capsule à Electret a dans son boîtier, un étage électronique chargé d'abaisser cette impédance de sortie en une impédance de sortie plus faible et donc plus accessible.

On trouve sur le marché deux sortes de microphones à Electret :

- microphone à Electret à deux fils ;
- microphone à Electret à trois fils ;



Illustration 1: microphone à Electret à deux pattes



Illustration 2: Microphone à Electret à trois pattes

Dans notre projet nous avons utilisé un microphone à Electret à deux fils. Vous trouvez ci dessous le schéma fonctionnel de ce microphone.

### Illustr 3

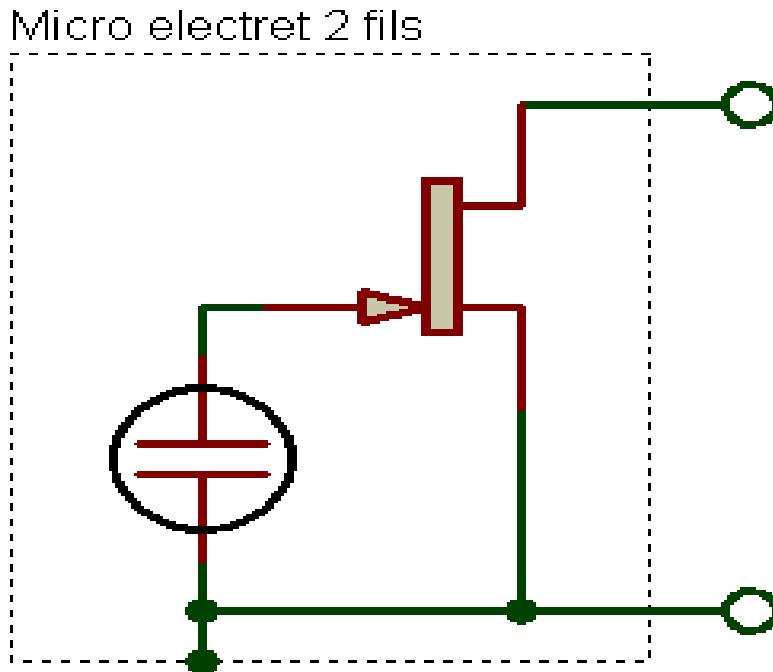


Illustration 3: Schéma interne micro à Electret

L'étage électronique qui accompagne le capteur se résume à un simple transistor FET qui normalement n'apporte aucune amplification dans la plupart du temps. Il nécessite toutefois une alimentation pour fonctionner. C'est pour cette raison qu'on nécessite l'apport d'une tension continue externe de 5Volts.

md  
md

## 2.2. Partie amplificateur

À la sortie du montage détection, nous plaçons un montage amplificateur pour amplifier le son capté. En effet le son capté par le microphone a une amplitude maximale d'environ 400mV d'où la nécessité de l'amplifier avant de le faire passer dans le filtre. Nous avons choisi de prendre un amplificateur de tension à base de transistor 2N222.

Le transistor 2N222 est un composant actif qui permet une amplification en tension. Il permet la réalisation de grandes fonctions en électronique telles que l'amplification et la commutation de signal ou des fonctions complexes réalisées grâce à des montages à base de transistors.

Il possède trois électrodes : Voir illustr 4

- B : la base
- C : le collecteur
- E : l'émetteur repéré par la flèche

Décaler

La flèche de l'émetteur (illustration 4) représente le sens réel de la circulation du courant dans le transistor. Le sens du courant dans un transistor est unidirectionnel. On distingue deux types de transistor NPN (courant est sortant de l'émetteur) et le PNP (courant est entrant dans l'émetteur).

Pour que le transistor se place au point de fonctionnement c'est-à-dire dans une zone où ses caractéristiques sont linéaires il faut faire une polarisation. Dans notre cas on a utilisé une polarisation par pont de base.

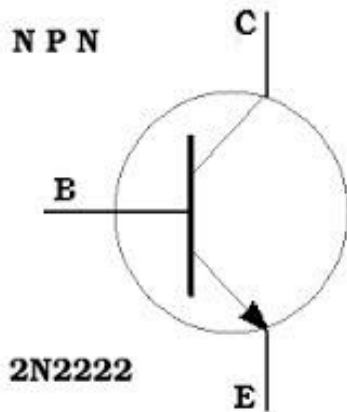


Illustration 4: transistor PNP



Illustration 5: transistor 2N222

Le schéma amplificateur réalisé est le suivant : Illustr 6

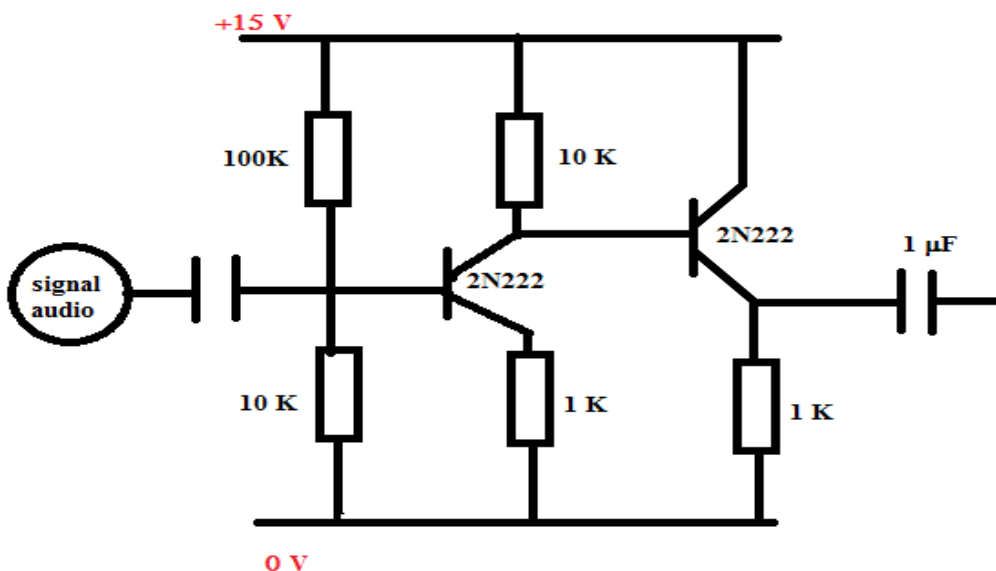


Illustration 6: montage amplificateur



~~Ceci~~ permet d'obtenir une amplification de 10 (fois 10) selon la formule suivante :

La formule  $A_v = \frac{\hat{U}_s}{\hat{U}_e} = \frac{\text{Amplitude de la tension de sortie}}{\text{Amplitude de la tension d'entrée}}$

Ainsi le signal à la sortie du microphone est alors amplifié et est maintenant de 4V. Ce qui est suffisant pour le faire passer au niveau du filtre passe-bande.

### 2.3. Partie filtrage

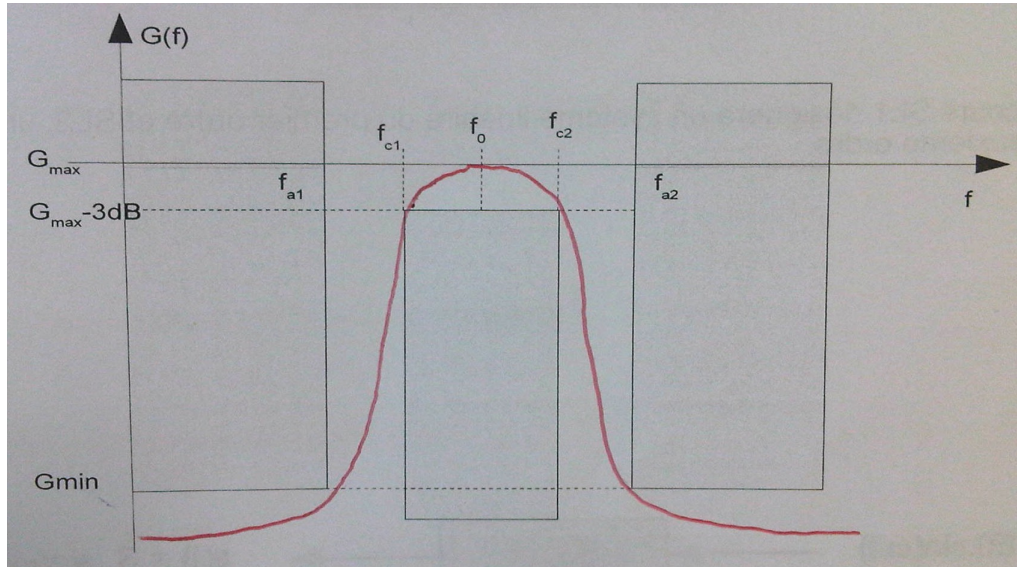


Illustration 7: description du filtre passe bande

Faire au moins 2 phrases !

Pour isoler le signal du claquement de main des autres signaux que pourrait capter le microphone nous mettons en place à la sortie du montage amplificateur un filtre passe bande. D'après nos recherches la fréquence d'un claquement de main tournerait autour de 1400Hz. Nous choisissons de prendre la bande de fréquences [1000;1800Hz] pour le filtre.

Le schéma du filtre passe-bande utilisé est le suivant :  
présenté illustr 8.

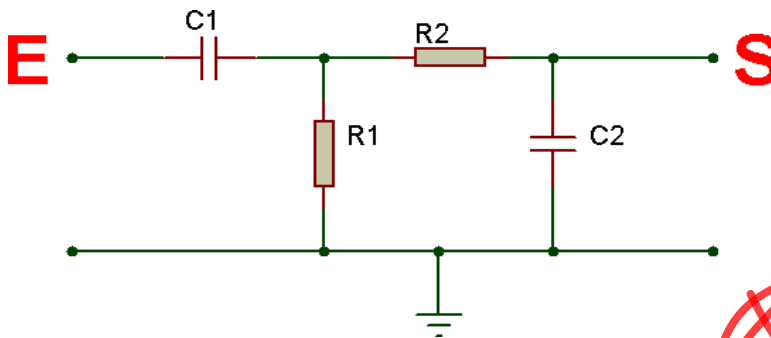


Illustration 8: filtre passe-bande

Avec :

R1=880 Ohms

C1=1μF

R2=800 Ohms

C2=100nF

Légende = autre style autre police, taille + petite...

À gauche de ce deuxième schéma, il y a un filtre passe-haut dont les composants ont été définis pour avoir une fréquence de coupure égale à 1000Hz. À droite, il y a un filtre passe-bas dont la fréquence de coupure  $F_c$  est de 1800Hz. En effet un filtre passe-bande résulte de l'association d'un filtre passe bas et d'un filtre passe haut .

Lequel où ?

$$F_c = \frac{1}{2\pi(RC)}$$

Ainsi nous arrivons à isoler les claquements de main des autres signaux sonores. Le signal filtré est ensuite injecté sur l'entrée analogique de l'ATMega.

## 2.4. Partie programmation

Pour qu'on puisse allumer la LED nous avons mis en place une partie programmation. La programmation se fait sur le logiciel Code Vision AVR.

Code Vision AVR est un compilateur conçu pour la famille des microcontrôleurs AVR d'ATMEL. Le compilateur C met en œuvre tous les éléments du langage ANSI C, comme le permet l'architecture AVR, avec quelques fonctionnalités supplémentaires pour profiter de la spécificité de l'architecture AVR et les besoins des systèmes embarqués. Ainsi grâce à Code Vision AVR nous pouvons programmer l'ATMega 8535.

RÉP

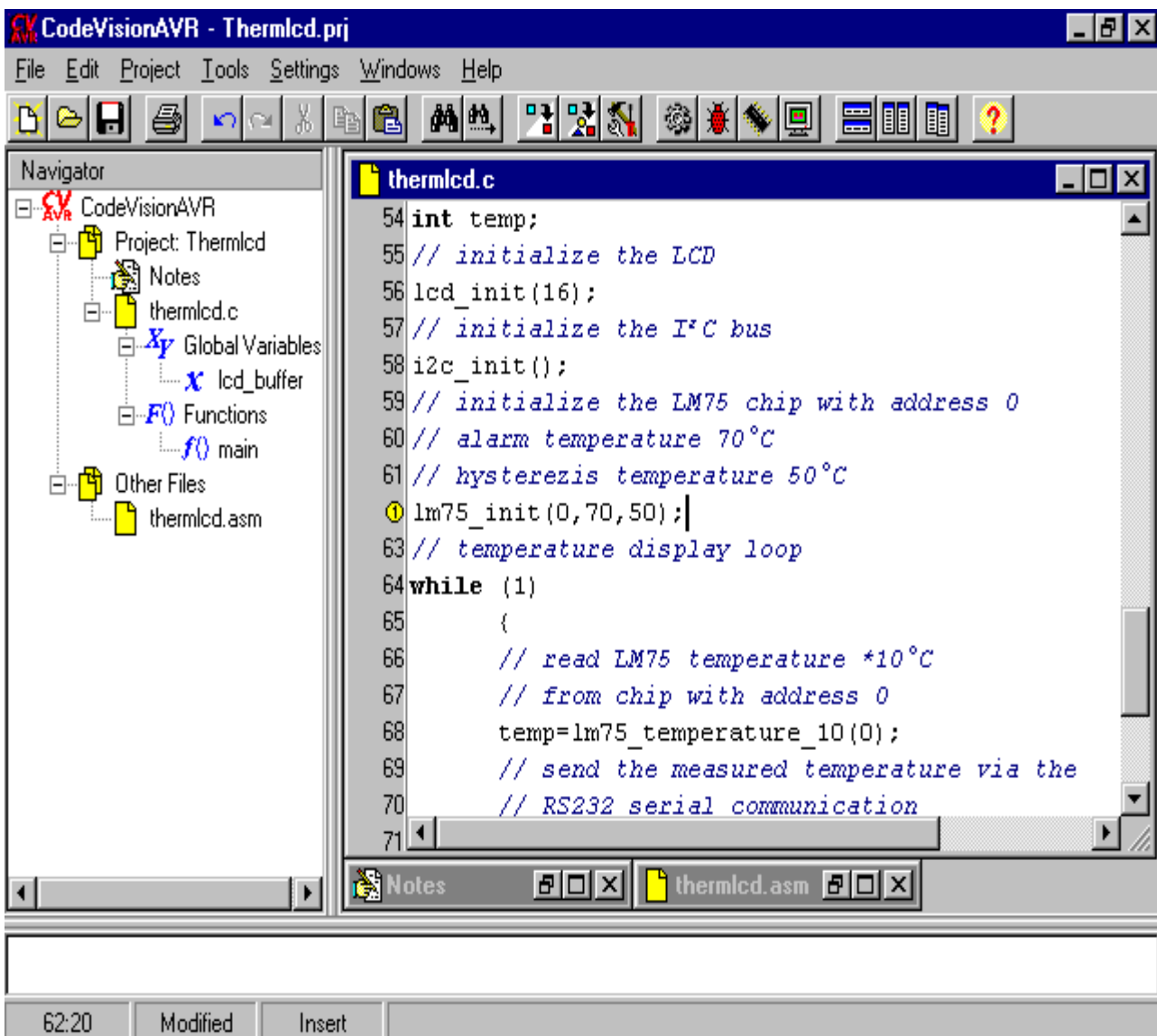


Illustration 9: interface de CodeVision AVR

L'ATMéga 8535 permet de programmer des systèmes embarqués. Ainsi elle est très utilisée dans la domotique.



Illustration 10: ATMéga 8535

Just 11

Le signal filtré est injecté sur l'entrée analogique JP1 (voir illustration ci-dessous). On définit ensuite une référence de tension. Dans notre cas nous avons pris 5V comme référence de tension.

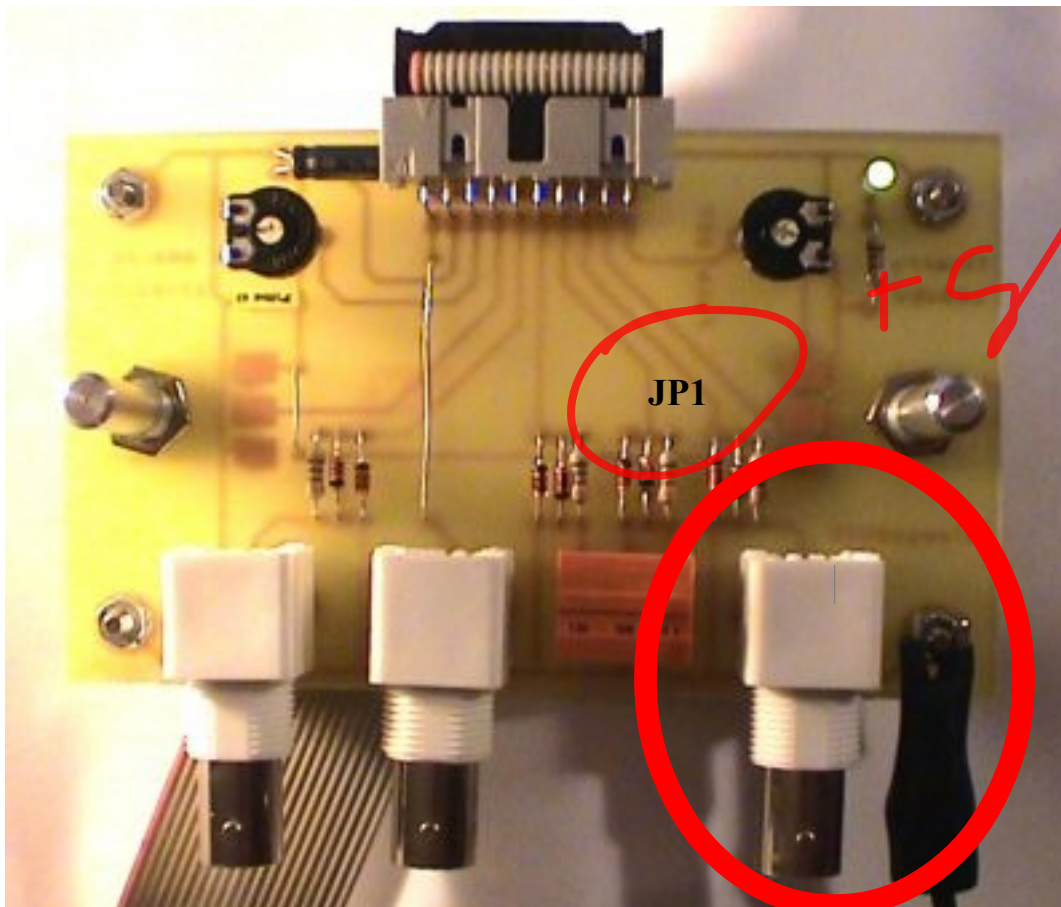
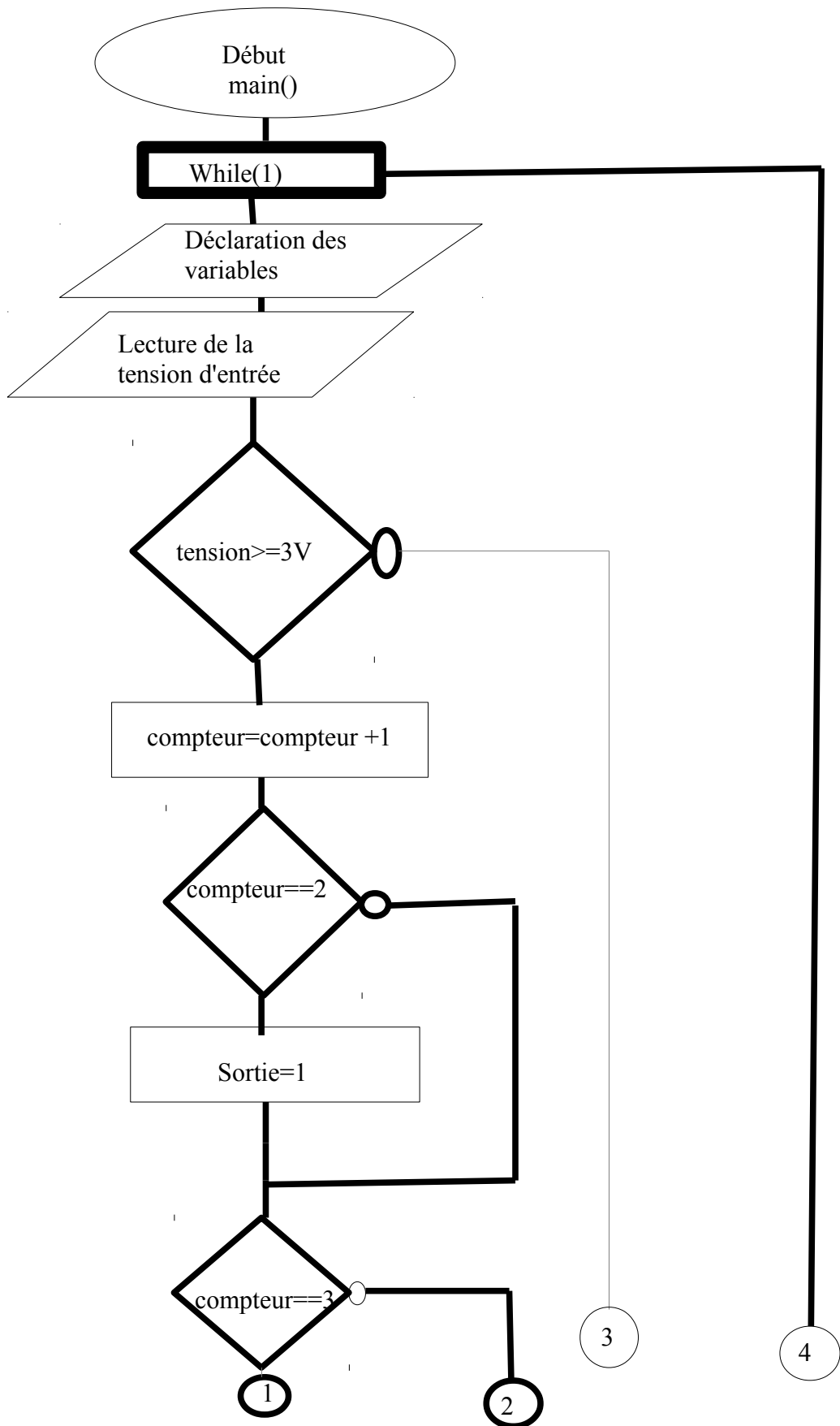


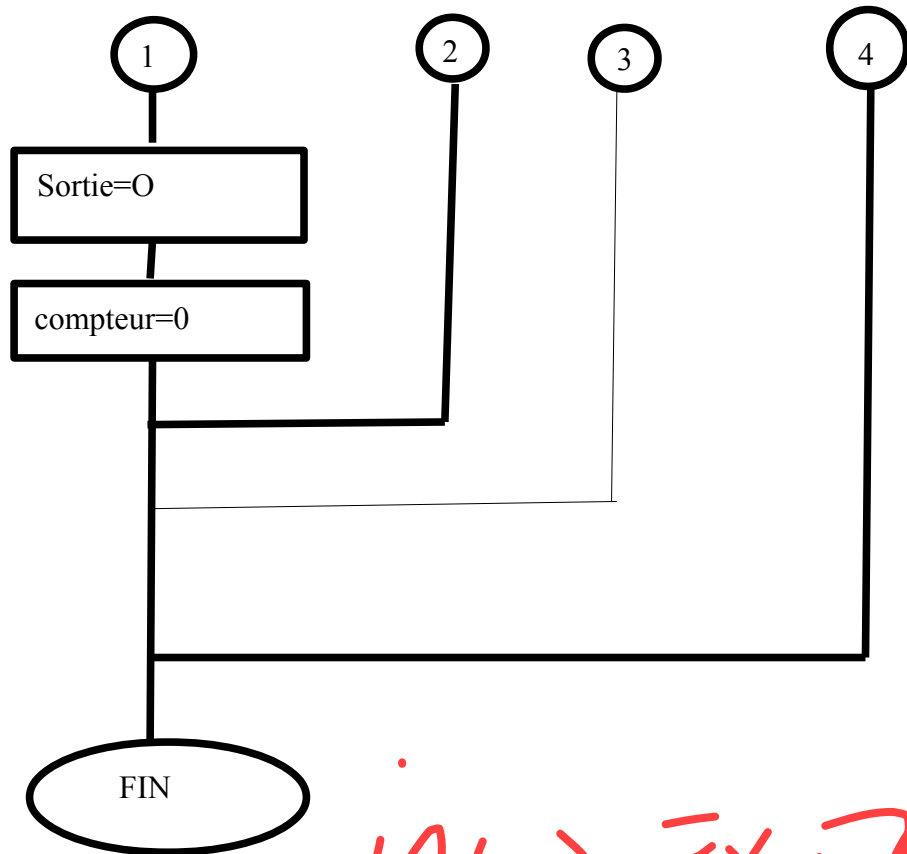
Illustration 11: Interface analogique

Nous avons choisi d'assigner au PORT A0 la valeur de la tension d'entrée. Pour la sortie, nous utilisons le Port B1 comme sortie numérique.

Le programme généré suit l'ordinogramme suivant.

*illustration*





*10V à l'EXER*

Un claquement de main génère une tension supérieure à 3V, ce qui incrémente le compteur. Au deuxième claquement de main, la valeur du compteur est de 2. Ceci permet de mettre à '1' la sortie et donc d'allumer la LED. Pour éteindre la LED, il faut claquer une troisième fois des mains.

Ainsi, au troisième claquement de main, le compteur vaut 3 et on éteint la LED. Le compteur est remis à zéro et le processus reprend. Vous retrouverez le programme complet en annexe 3.



### 3. Suivi de projet

Introduire cette partie

#### 3.1. Problème rencontré

Le principal problème rencontré dans la réalisation de ce projet a été l'extinction de la LED. Nous arrivions à allumer la LED avec deux claquements de main, par contre lorsqu'on claquer une troisième fois des mains, la LED ne s'éteint pas. Après plusieurs tests, nous avons pu détecter que le problème était dû à une temporisation « delay ». En effet, nous avons mis une temporisation d'une seconde après l'allumage. Ceci ne donnait pas le temps à l'utilisateur de claquer une troisième fois des mains et rentrer encore dans la boucle d'allumage. Ainsi en supprimant la temporisation, nous avons remédié à ce problème.

#### 3.2. ~~planning~~

Introduire

SEMAINE	5	6	7	8	9	10	11	12	12
Choix sujet									
Détection signal									
Amplification									
Filtrage									
Programmation									
Finalisation & Rédaction									

	Planning prévisionnel
	Planning réel
	Vacances scolaires

Commenter ces plannings.  
 Les placer avant le pb rencontré... Est-ce que cela a affecté votre pp ?  
 Avez-vous bien "mesuré" le temps, pour chaque étape ?

## Conclusion

La mise en place de ce projet d'étude et réalisation nous a permis d'être autonome sur les démarches et techniques à mettre en place. Nous étions menés à concevoir notre propre sujet.

Le choix que nous avons fait, celui de la réalisation de Lu.Co.Cla.M était plaisant. Cependant, nous n'étions pas conscient des éventuels problèmes que nous pouvions rencontrer. Face à cela, une mise en matière de toutes les notions que nous avons pu accumuler tout au long de cette scolarisation, nous a permis de trouver les réponses à nos problèmes.

Grâce aux recherches effectuées mais aussi aux différentes phases de tests de nos montages, nous avons pu comprendre le fonctionnement de nos maquettes, et donc élargir notre connaissance dans ces domaines.

Face aux difficultés rencontrées, nous étions menés à réagir différemment, notamment pour le problème de temps où nous devons redoubler d'efforts et prendre sur notre temps libre pour rattraper notre retard. Une expérience qui nous permet de prendre conscience qu'il est difficile de respecter les temps de réalisation si l'on n'élabore pas un bon planning.



## Résumé

Voici un projet d'étude et réalisation du semestre quatre lors d'une formation à un DUT GEII. Le but de ce projet est de réaliser un système permettant d'allumer une lampe avec nos claquements de mains. Deux coups pour l'allumer et le troisième coup pour l'éteindre et ainsi de suite.

La première chose à faire a été d'établir le cahier des charges. Le système est divisé en plusieurs parties, afin de mieux comprendre son fonctionnement.

Notre projet est composé de quatre parties : une partie pour la détection du signal, une partie d'amplification, une partie de filtrage et une partie programmation. Pour le détecteur sonore on utilise un microphone à Electret.

L'amplification a été réalisée avec des transistors associés à divers composants passifs.

Le filtrage est composé d'un filtre passe-bande. La valeur des composants a été calculée en fonction des fréquences de coupure que l'on souhaitait obtenir.

La programmation est réalisée grâce au logiciel Code Vision AVR et on applique ce programme sur un Atméga 8535.

Enfin, ce dossier est muni d'une partie spécifique sur le suivi du projet. Dans cette partie, il est question de l'avancement du projet, ainsi que des différents problèmes rencontrés lors de sa réalisation.

## **Index des illustrations**

Illustration 1: microphone a electret a deux patte .....	6
Illustration 2: Microphone a electret a trois pattes.....	6
Illustration 3: Schéma interne micro à electret.....	7
Illustration 4: transistor PNP.....	8
Illustration 5: transistor 2N222.....	8
Illustration 6: montage amplificateur.....	8
Illustration 7: description du filtre passe bande.....	10
Illustration 8: filtre passe-bande.....	10
Illustration 9: interface de CodeVision AVR.....	11
Illustration 10: ATMéga 8535.....	12
Illustration 11: Interface analogique.....	12

## Bibliographie

1. [www.diapovision.com/articles/micros\\_electret.htm](http://www.diapovision.com/articles/micros_electret.htm) consulte le 27/03/2013
2. <http://etronics.free.fr/dossiers/init/init10.htm> consulté le 27/03/2013
3. <http://fr.audiomicro.com/effets-sonores/de-l-39-homme/applaudir> consulté le 25/03/2013
4. [http://www.sonelec-musique.com/electronique\\_bases\\_transistor.html](http://www.sonelec-musique.com/electronique_bases_transistor.html) consulté le 25/03/2013
5. <http://www.thierry-lequeu.fr/data/DIV517.HTM>

*Normes à lire ?*

## Index des annexes

Annexe 1 : datasheet Ateméga 8535 .....	page21
Annexe 2 : carte de progration .....	page 22-24
Annexe 3 : page programme réaliser.....	page 25-32

Faire effort de mise en page

Annexes

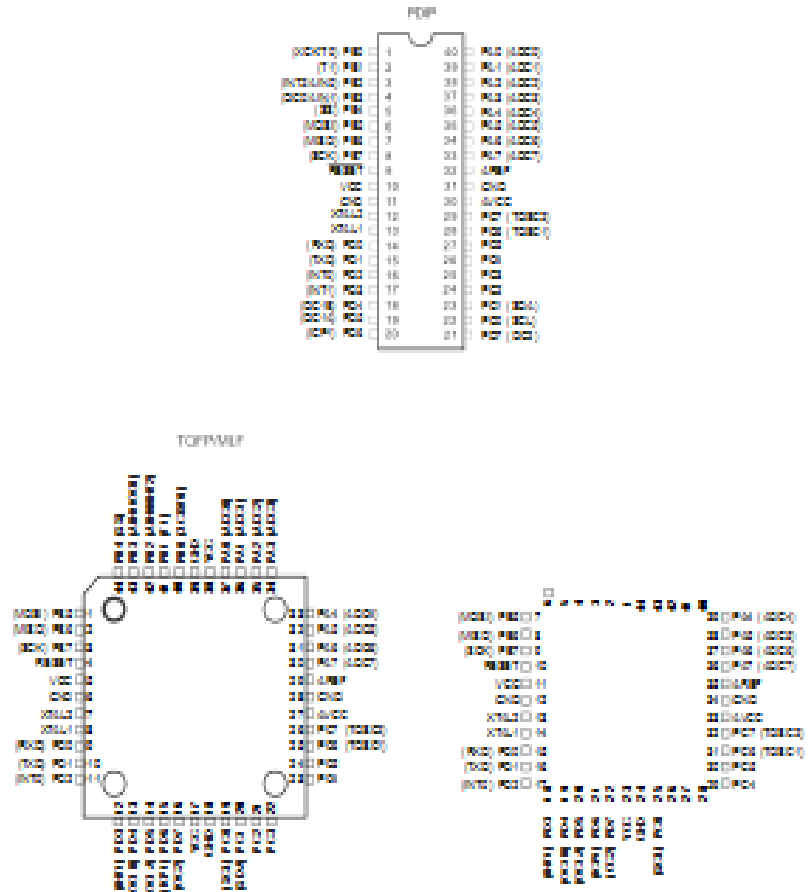
Sommaire des annexes

# TITRE ?



## 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.

## ATmega8535(L)

200002-AVR-08/03

### 5.5 Interface analogique de test

Projet : IUTS  
 Info : [DIV512]  
 Révision : 1 du 17 février 2007



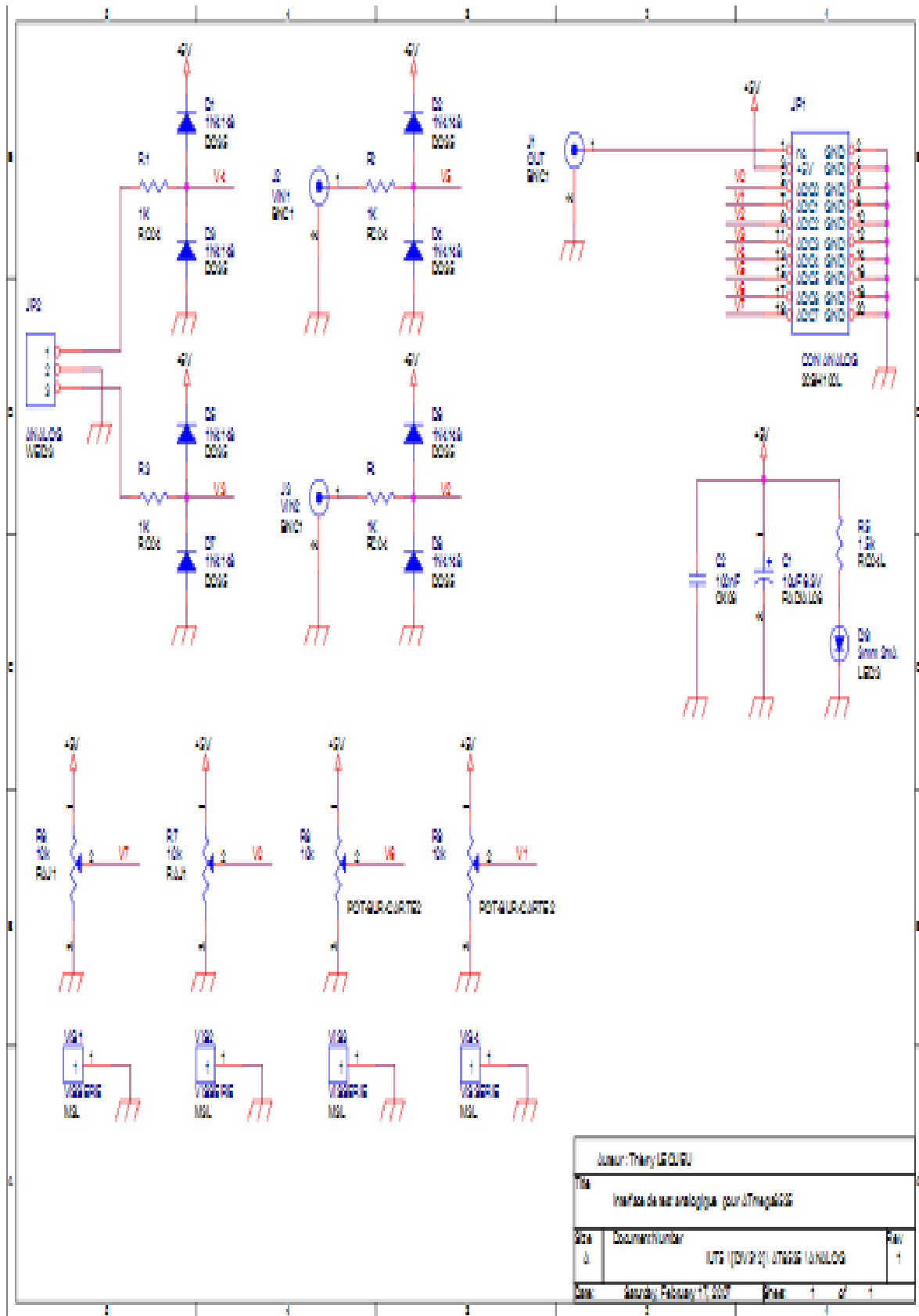
Figure 5.5 Interface analogique de test (images-maquettes [AT8333-ANALOG-12].jpg).

### 5.6 Désignation des composants

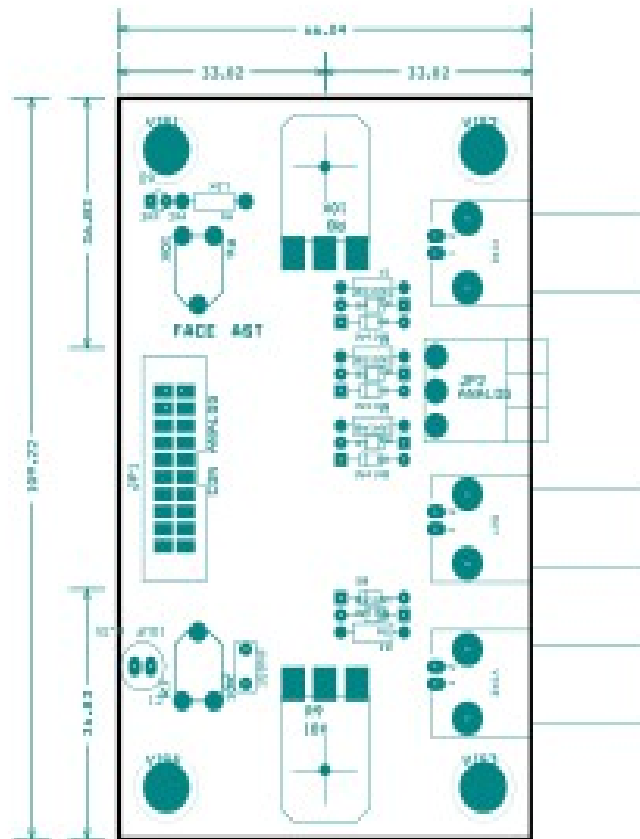
Tableau 5.3 Liste de composants (projets-iut5.xls / AT8333-ANALOG).

N°	Quantité	Référence	Désignation	Emprise
1	1	C1	100nF 6.3V	RADIAL08
2	1	C2	100nF	CK06
3	8	D1,D2,D3,D4,D5,D6,D7,D8	IN4148	DOC5
4	1	D9	3mm 2mA	LED8
5	1	JP1	CCN ANALOG	208 HD00L
6	1	JP2	ANALOG	WEID3
7	1	J1	OUT	BNC1
8	1	J2	VIN1	BNC1
9	1	J3	VIN2	BNC1
10	4	R1,R2,R3,R4	1K	RC04
11	1	R5	1.5k	RC04L
12	2	R7,R6	10k	RAJ1
13	2	R8,R9	10k	POT-SUR-CARTE2
14	4	VIS1,VIS2,VIS3,VIS4		

# T, tre analoge



# Tire d'axe





**Annexe 3**

TITRE

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

This program was produced by the  
CodeWizardAVR V2.03.4 Standard  
Automatic Program Generator  
© Copyright 1998-2008 Pavel Haiduc, HP InfoTech s.r.l.  
<http://www.hpinfotech.com>

Project :

Version :

Date : 13/03/2013

Author :

Company :

Comments:

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

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

```
#include <mega8535.h>
```

```
#include <delay.h>
```

```
#include <mega8535.h>
```

```
/* the LCD module is connected to PORTC */
```

# Tire Anne

```
#asm
    .equ __lcd_port=0x15
#endasm

/* Pour affichage */
#include <lcd.h>

#include <stdio.h>

// Declaration de variables
int compteur=0;
#define Sortie PORTB.1
#define entree PORTA.0
#define Tension 5
int valtension;

#define ADC_VREF_TYPE 0x00

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
```

# Titre annexe

```
return ADCW;  
}
```

```
// Declare your global variables here
```

```
void main(void)
```

```
{
```

```
// Declare your local variables here
```

```
unsigned char tampon[20];
```

```
// Input/Output Ports initialization
```

```
// Port A initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In  
Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTA=0x00;
```

```
DDRA=0x00;
```

```
// Port B initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In  
Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTB=0x00;
```

```
DDRB=0xFF;
```


```
// Port C initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In  
Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTC=0x00;
```

```
DDRC=0x00;
```



```
// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;

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

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

TCNT1H=0x00;

TCNT1L=0x00;

ICR1H=0x00;

ICR1L=0x00;

OCR1AH=0x00;

OCR1AL=0x00;

OCR1BH=0x00;

OCR1BL=0x00;

// Timer/Counter 2 initialization

// Clock source: System Clock

// Clock value: Timer 2 Stopped

// Mode: Normal top=FFh

// OC2 output: Disconnected

ASSR=0x00;

TCCR2=0x00;

TCNT2=0x00;

OCR2=0x00;

// External Interrupt(s) initialization

// INT0: Off

// INT1: Off

// INT2: Off

MCUCR=0x00;


MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization

TIMSK=0x00;

// Analog Comparator initialization





```

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


// ADC initialization
// ADC Clock frequency: 1000,000 kHz
// ADC Voltage Reference: AREF pin
// ADC High Speed Mode: Off
// ADC Auto Trigger Source: None
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;
SFIOA&=0xEF;
/* initialize the LCD for 2 lines & 16 columns */
lcd_init(16);
lcd_gotoxy(0,0);
lcd_putsf("bonjour!Claquez 2fois des mains  pour allumer ");

/* switch to writing in Display RAM */

while (1)
{
// Place your code here
// Place your code here
//PORTD =0x00;
//PORTD=0xFF;

valtension= read_adc(Tension);
sprintf(tampon,"%4d / %4d",valtension,compteur);

```



```
lcd_gotoxy(0,3);  
lcd_puts(tampon); //on affiche la valeur.
```

```
if(valtension>=600)  
{  
  
    compteur =compteur+1;  
    delay_ms(100);  
}
```

```
if(compteur==2)  
{  
    Sortie=1;  
    //cp_pllcd_init(16);  
    lcd_gotoxy(0,0);  
  
    lcd_putsf("Lumiere      allumee");
```

```
    sprintf(tampon,"%4d",valtension);  
    lcd_gotoxy(0,3);  
    lcd_puts(tampon); //on affiche la valeur.  
    //delay_ms(10);
```



```
}
```

```
if(compteur>=3)
```

```
{
```

```
Sortie=0;
```

```
lcd_gotoxy(0,0);
```

```
lcd_putsf("Lumiere eteinte ");
```

```
delay_ms(2000);
```

```
lcd_init(16);
```

```
lcd_gotoxy(0,0);
```

```
lcd_putsf("bonjour!Claquez 2fois des mains pour allumer ");
```

```
compteur =0;
```

```
delay_ms(2000);
```

```
}
```

```
}
```

```
}
```