

## Wattmètre secteur 220V/16A



Ala SAHMIM  
Mouna AIT EL MAJDOUB  
2ième – P2  
2008-2010

Enseignants  
M. Thierry LEQUEU  
Mme Véronique AUGER  
[<NOM CORRECTEUR 2>](#)

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



# Wattmètre secteur 220V/16A

Ala SAHMIM  
Mouna AIT EL MAJDOUB  
2ième – P2  
2008-2010

Enseignants  
M. Thierry LEQUEU  
Mme Véronique AUGER  
<NOM CORRECTEUR 2>

# Sommaire

Introduction.....	4
1. Cahier des charges .....	5
1.1. Présentation générale .....	5
1.2. Matériels utilisés .....	5
1.3. Schémas Fonctionnels.....	5
2. Schéma structurel et analyse technique .....	6
2.1. Atmega8535.....	6
2.2. Choix de l'afficheur.....	8
2.3. Logiciel utilisé.....	8
3. Analyse de fonctionnement .....	11
3.1. Description de méthode de calcul.....	11
3.2. la première expérience.....	11
3.3. Deuxième expérience .....	20
4. Le capteur de courant .....	23
5. Test et validation.....	25
6. Planning prévisionnel et réel .....	27
Conclusion.....	28
Résumé.....	29
Index des illustrations.....	30
Bibliographie.....	31

## Introduction

Dans le cadre de l'étude et réalisation du semestre quatre, nous avons réalisé un projet intitulé : Wattmètre secteur 220V/16A. Notre mission a consisté à mettre au point un programme qui permettrait l'affichage des valeurs efficaces du courant et de la tension ainsi que la puissance active (produit de ces deux grandeurs).

Afin de réussir notre projet, nous avons fixé le cahier des charges. Des schémas fonctionnels préciseront la démarche du projet permettant la programmation de Atmega8535. Des plannings prévisionnel puis réel ont été établis pour optimiser le temps de travail.

# 1. Cahier des charges

## 1.1. Présentation générale

Dans le cadre de notre projet tuteuré d'étude et réalisation, nous avons décidé de nous intéresser à la programmation d'un wattmètre triphasé.

Le but de notre projet consiste à mettre en place une programmation du micro-contrôleur ATmega8535 qui affichera les valeurs efficaces des tensions et courants sinusoïdaux puis, par la suite, par une simple opération mathématique prédéfinie, exprimera la puissance (voir figure 1).

D'après le cahier des charges, les contraintes imposées seront un microcontrôleur ATMEL Atmega8535 munis d'un CAN, un capteur de courant HASS 200-S, un afficheur LCD 16 segments et une alimentation alternative entre 0 et 5 Volts.

La carte électronique et la carte d'interface sont déjà fournies. Un logiciel imposé (CodeVisionAVR) sera utilisé pour la programmation du micro-contrôleur dans le déroulement du projet .

## 1.2. Matériels utilisés

Le Wattmètre est commandé par un microprocesseur Atmega8535. Ce composant est le micro-contrôleur ( circuit intégré ) que nous avons utilisé pour programmer l'affichage des différentes grandeurs à mesurer. Ces renseignements seront affichés grâce à un afficheur LCD 16 caractères / 4 lignes ( Farnell 944-9019 ).

Pour afficher les valeurs efficaces du courant et de la tension, nous avons utilisé un GBF<sup>1</sup> pour obtenir un signal sinusoïdal. Ce signal sera visualisé sur un oscilloscope avant d'être envoyé sur la carte d'interface analogique et par la suite programmer Atmega8535. Le port C de ce dernier sera connecté à l'afficheur LCD pour voir les grandeurs souhaitées.

## 1.3. Schémas fonctionnels

### 1.3.1. Le schéma fonctionnel de niveau 1



Figure 1: schéma fonctionnel de niveau 1

1 GBF : Générateur Basse Fréquence

### 1.3.2. le schéma fonctionnel de niveau 2

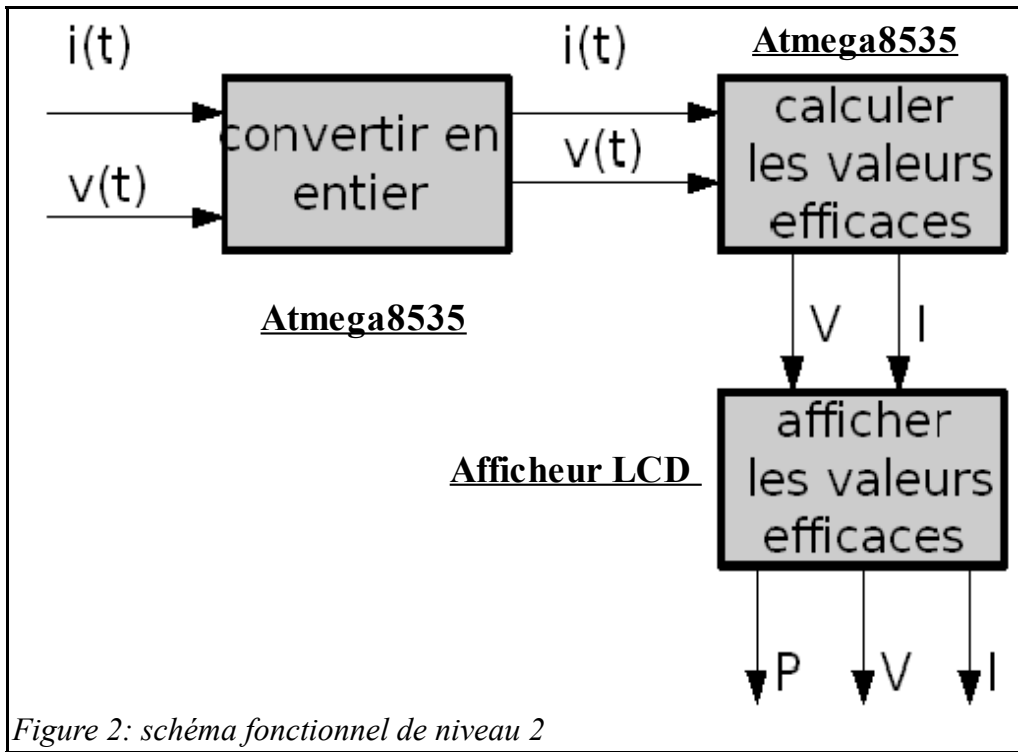


Figure 2: schéma fonctionnel de niveau 2

## 2. Schéma structurel et analyse technique

### 2.1. Atmega8535

Nous avons choisi ce microprocesseur Atmega8535, car il présente un nombre d'entrées/sorties qui s'adapte bien à notre dispositif technique.

Le Micro-contrôleur Atmega8535 présente de nombreuses fonctions. Il est composé de 4 ports A,B,C,D dont chacun comporte 8 Bits.

Chaque bit peut être pris en compte en tant qu'une entrée ou sortie numérique en fonction du choix de l'utilisateur.

Dans le cadre de notre projet « Wattmètre Triphasé » nous nous intéresserons uniquement aux ports A ,D et C .

Concernant le port A, les deux premiers bits PA0 et PA1, dont la tension sinusoïdale et le courant impulsionnel sont envoyés dans ces entrées, seront utiles.

Le choix de la voie se fait par le biais du multiplexeur afin d'effectuer les mesures en numérique.

Cette conversion est due à la présence d'un convertisseur analogique numérique CAN.

Remarque : Compte tenu de notre cahier des charges, la dynamique de tension doit être comprise entre 0V et 5V.

La variable de mesure « x » peut être comprise entre 0 et 1023 c'est à dire qu'elle peut atteindre 1024 valeurs (10 bits).

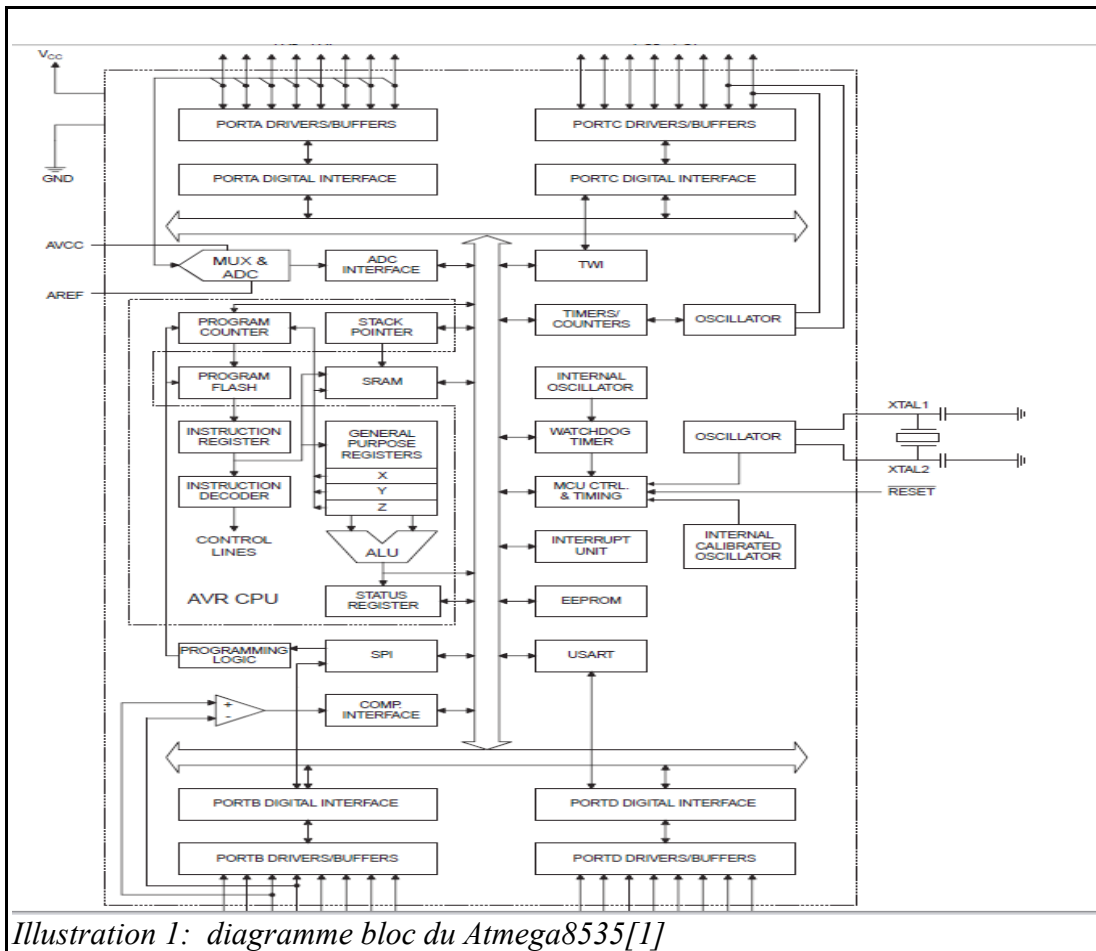
Prenant en considération ces différentes valeurs, on en déduit que la plus petite valeur de tension qui peut être mesurée (quantum) est de l'ordre de 4,88 mV.

Ceci signifie qu'à la moindre augmentation de 5mV au niveau de la tension , la valeur de « x » augmente de 1.

Sur le port D, nous avons connecté le bornier de programmation du microprocesseur afin d'envoyer notre programme.

Le port C permet de connecter l'afficheur. Ce choix est particulier puisque les différents bits peuvent être considérés à la fois comme des entrées et sorties

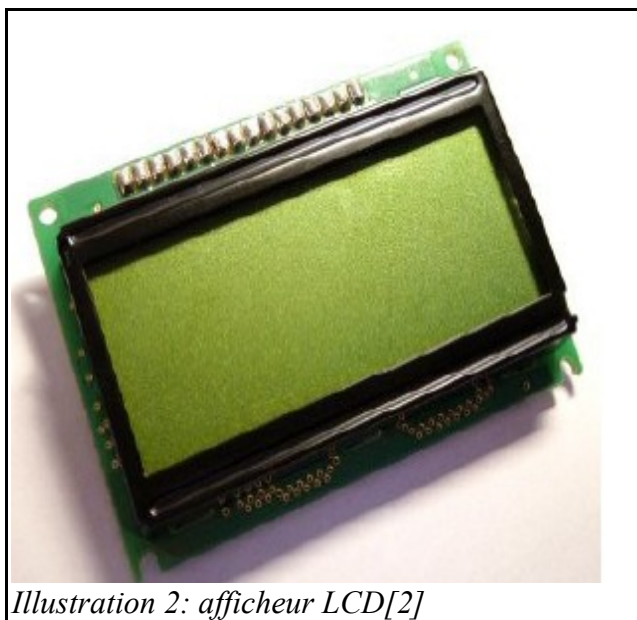
Voici le diagramme bloc de Atmega8535 :



## **2.2. Choix de l'afficheur**

Cet afficheur LCD 16\*4 Farnell 944-9019 a été utilisé pour sa capacité d'affichage de toutes les données demandées.

En effet, son utilisation a été imposée puisqu'il était déjà connecté sur la carte électronique.



*Illustration 2: afficheur LCD[2]*

## **2.3. Logiciel utilisé**

Durant le déroulement de notre projet, le micro-contrôleur Atmega8535 a été programmé via un nouveau logiciel, Code Vision AVR, simple d'utilisation mais nécessitant des connaissances en langage C .

Le type de Microprocesseur à utiliser ainsi que le port où sont affichées les données (où est connecté l'afficheur LCD) est renseigné avant toute programmation dans le logiciel. Ce dernier génère lui-même toutes les fonctions nécessaires au fonctionnement du microprocesseur telles que l'initialisation des ports.

Parmi les fonctions, qu'on a modifié, figurent dans les images suivantes :



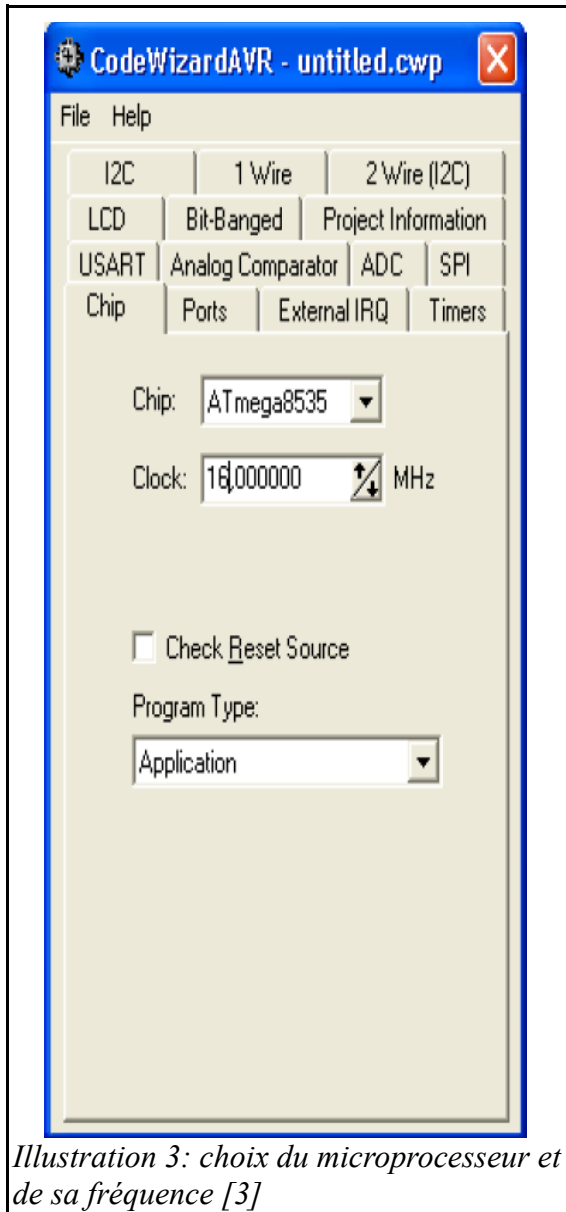


Illustration 3: choix du microprocesseur et de sa fréquence [3]

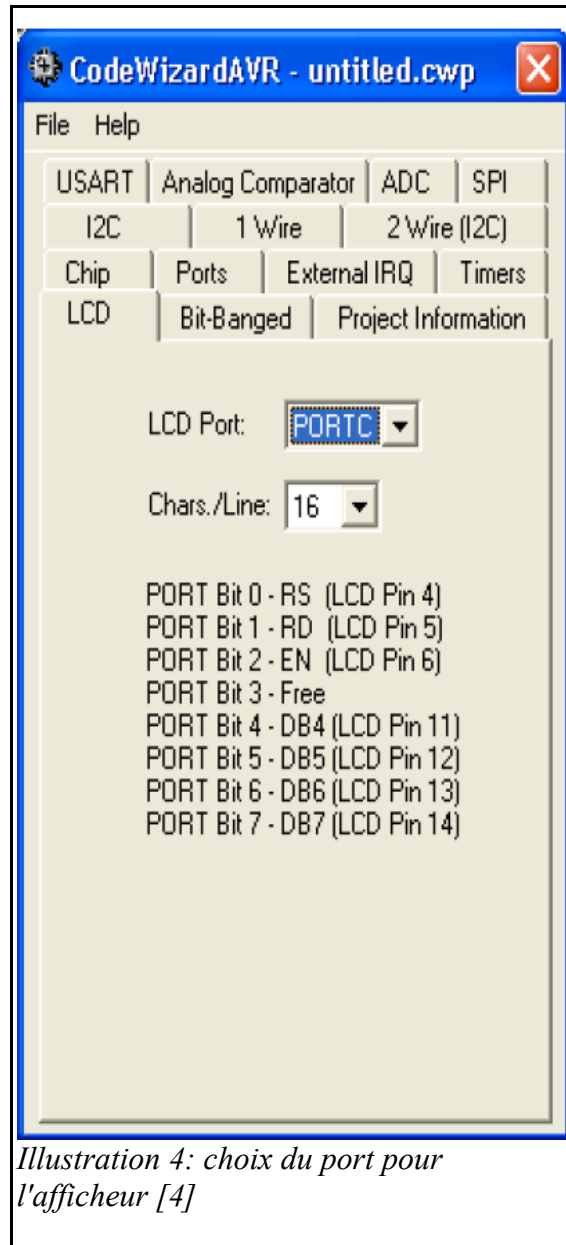


Illustration 4: choix du port pour l'afficheur [4]

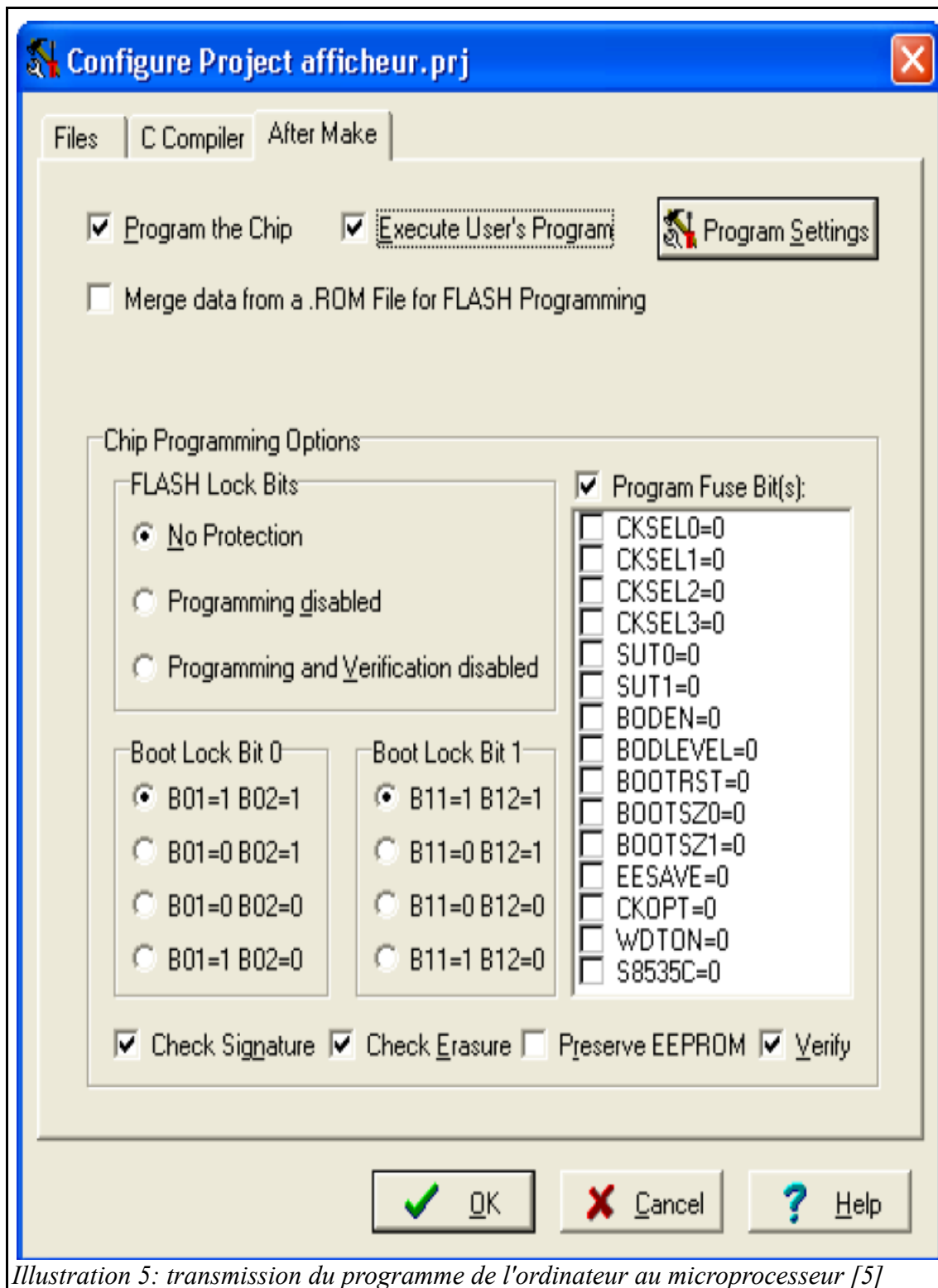


Illustration 5: transmission du programme de l'ordinateur au microprocesseur [5]

### 3. Analyse de fonctionnement

Notre projet consiste à calculer la valeur efficace de la tension sinusoïdale et du courant alternatif pour enfin arriver à calculer la puissance qui est le produit de ces deux grandeurs.

#### 3.1. Description de méthode de calcul

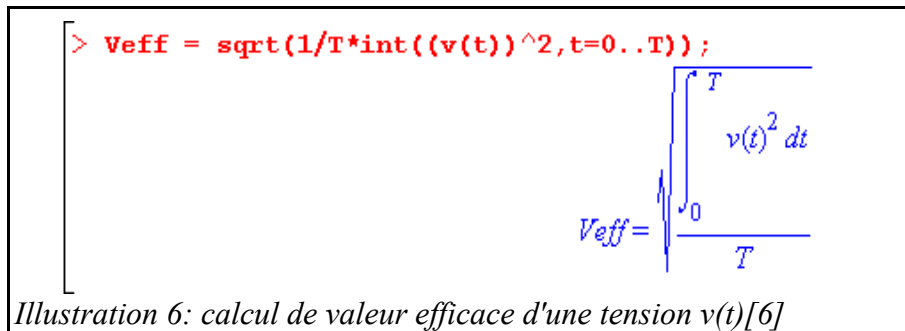


Illustration 6: calcul de valeur efficace d'une tension  $v(t)$ [6]

Pour calculer la valeur efficace d'une tension sinusoïdale, en général, on utilise la relation suivante :

Afin de réussir ce calcul, nous avons programmé l'Atmega8535 dans le but de le réaliser automatiquement. Sur ce micro-contrôleur, la relation ne peut pas être écrite directement, d'où la nécessité de réfléchir d'une manière logique. Un programme doit être envoyé au micro-contrôleur afin d'exécuter les calculs et de permettre le bon affichage.

Deux méthodes ont été expérimentées pour calculer la valeur efficace de la tension. Dans un premier temps, nous avons travaillé seulement sur une demi période alors que dans la deuxième expérience nous avons pris en compte toute la période du signal pour calculer sa valeur efficace.

#### 3.2. La première expérience

##### 3.2.1. Description

Lors des expériences effectuées dans les séances de TP E&R, nous avons été amenés à calculer le temps de conversion, à définir la durée d'échantillonnage (en choisissant une demi-période) et à fixer le nombre de points nécessaires pour effectuer nos calculs.

Le temps de conversion mis par l'Atmega8535 est le temps écoulé pour effectuer deux mesures successives. Celui-ci peut être défini par la manipulation suivante. Une sonde TSX2014 est placée sur l'entrée 0 du port D du Atmega8535 et la tension d'un potentiomètre de la carte d'interface analogique de test est modifiée jusqu'à obtenir la meilleure valeur. Cette dernière a été fixée à  $100\mu s$ . La partie programmation dépendra sensiblement de cette valeur.

◆ L'organigramme de la fonction «calcul de temps de conversion» :

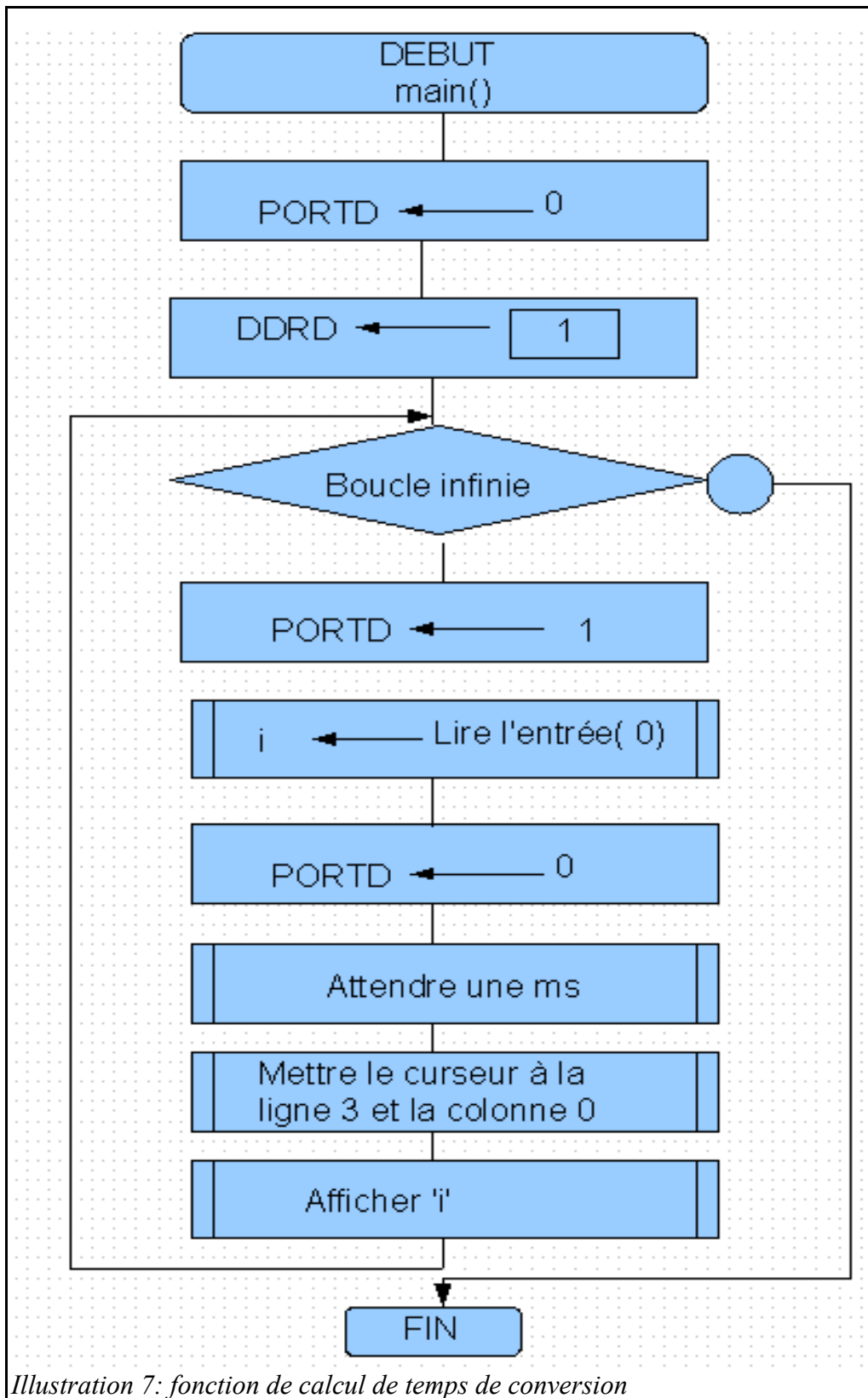


Illustration 7: fonction de calcul de temps de conversion

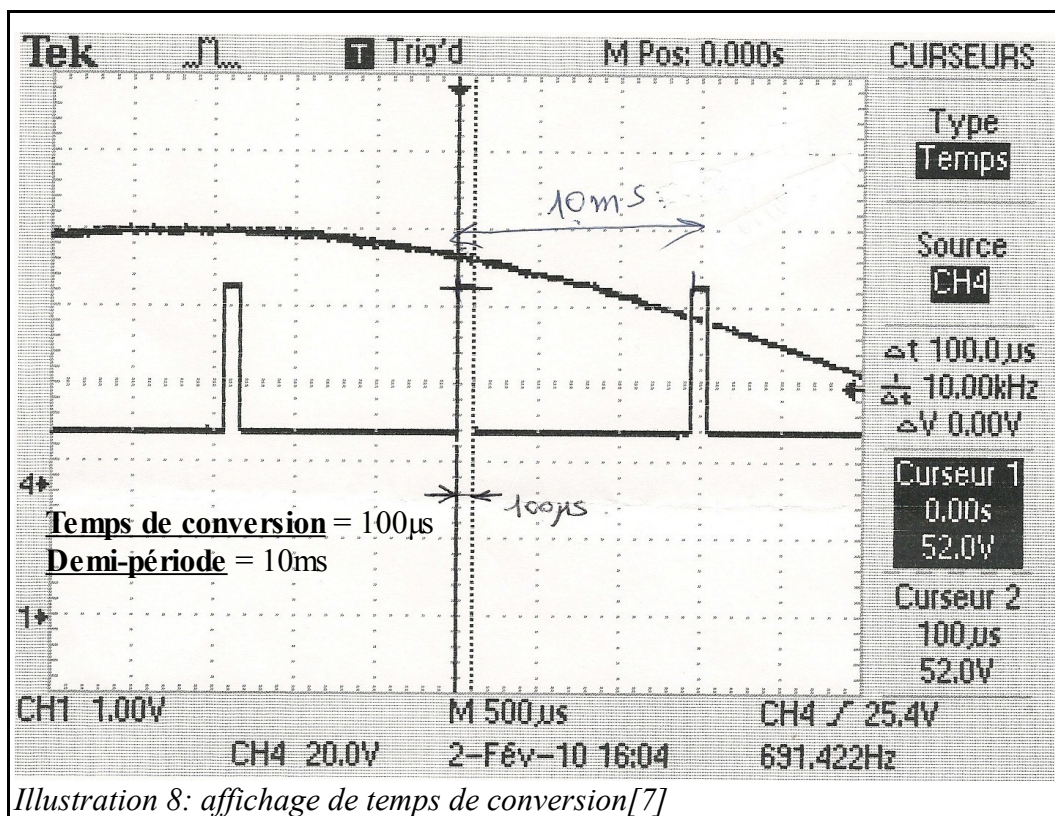
- ◆ Le programme pour calculer le temps conversion:

```

PORTD=0x00;
DDRD=0xFF; //port D en sortie .
while (1)
{
    PORTD=0xFF;
    i=read_adc(0);
    PORTD=0x00;
    delay_ms(1);
    sprintf(tampon,"%4d",i);
    lcd_gotoxy(3,0);
    lcd_puts(tampon);
};
}

```

nous avons obtenu cette courbe :



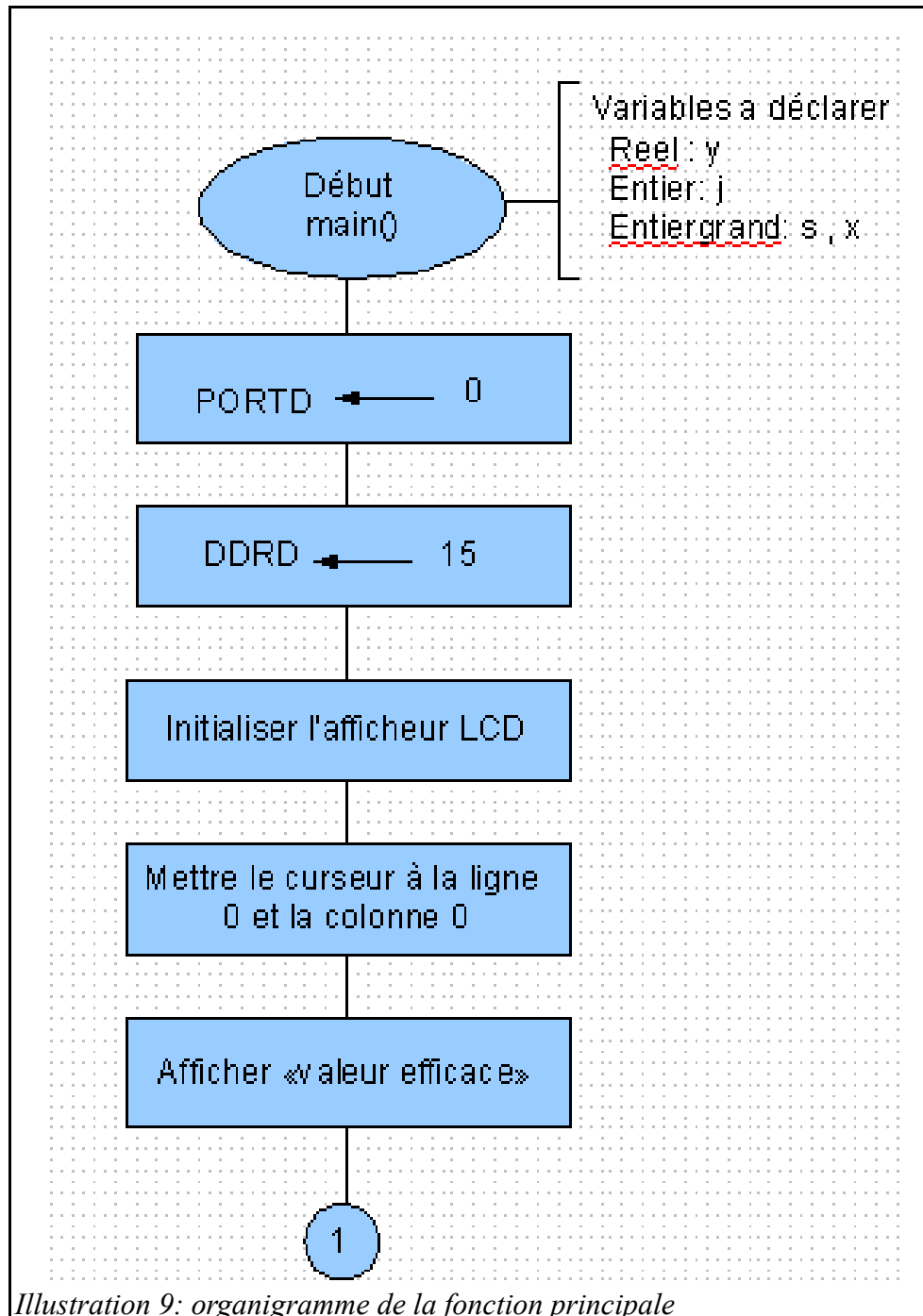
Le temps de conversion mesuré est de  $100\mu\text{s}$  auquel s'ajoute un temps de calcul de  $10\mu\text{s}$ . Le nombre de points de mesure possibles donnant les meilleures valeurs au moment de l'affichage sont calculées sur une demi période (10ms) par la formule suivante :

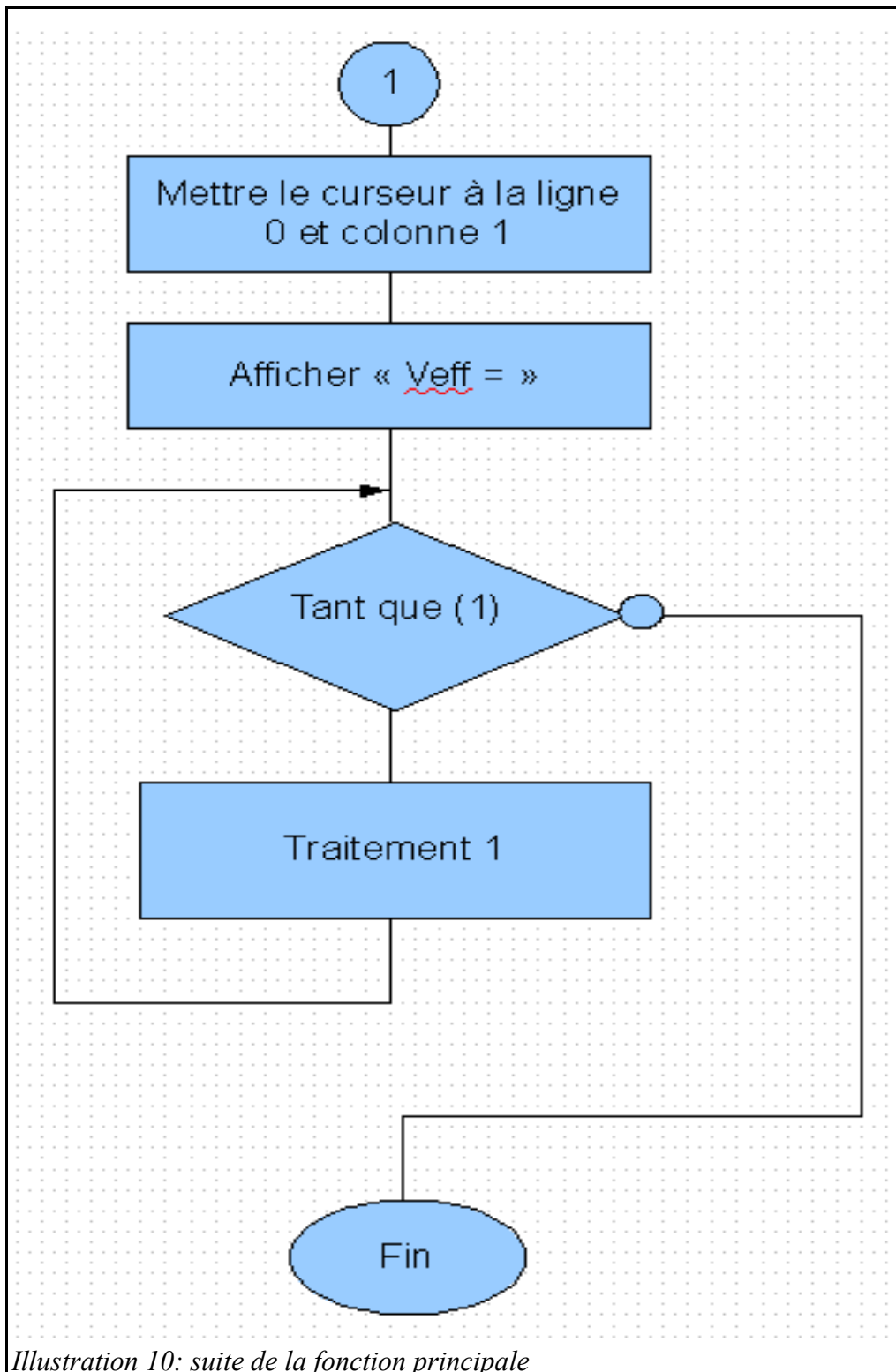
**Le nombre de points =  $\Delta T / 110\mu\text{s}$  = 90 points.**

Ces différents valeurs sont des valeurs expérimentales qui sont déduites à partir de l'oscilloscope.

### 3.2.1. Programmation

➤ Ordinogramme :





*Illustration 10: suite de la fonction principale*

- La déclaration des variables :

```
void main(void)
{
// Declare your local variables here
long int x ;
float y;
int j;
long int S ;

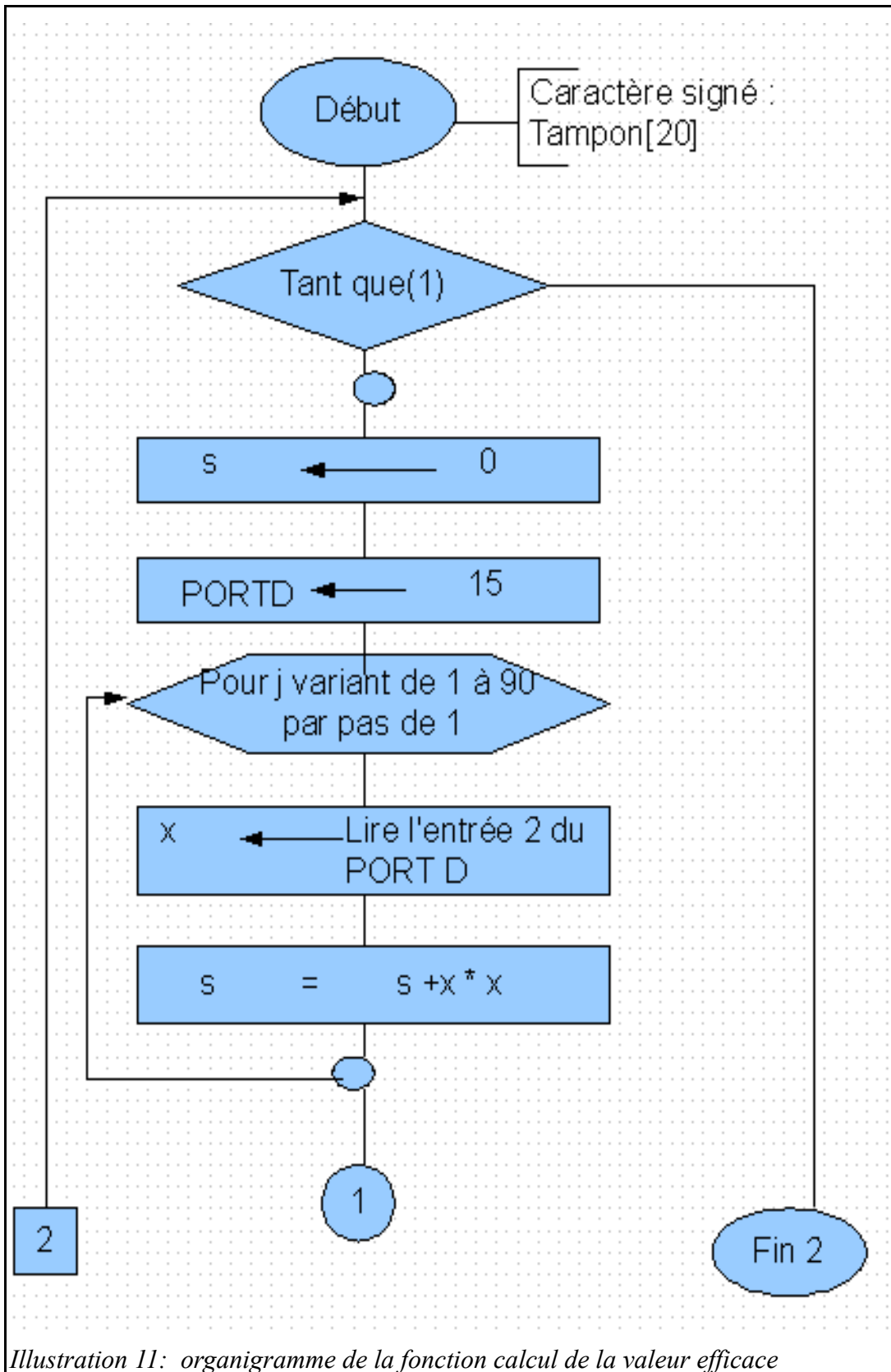
PORTD=0x00;
DDRD=0xFF; //port D en sortie .
```

- Le port D est en sortie

- Cette fonction permet d'écrire sur l'afficheur LCD

```
//LCD module initialization
lcd_init(16);
lcd_gotoxy(0,0);
lcd_putsf("valeur efficace"); //Pour écrire sur l'afficheur LCD
lcd_gotoxy(0,1);
lcd_putsf("Veff = ");
```





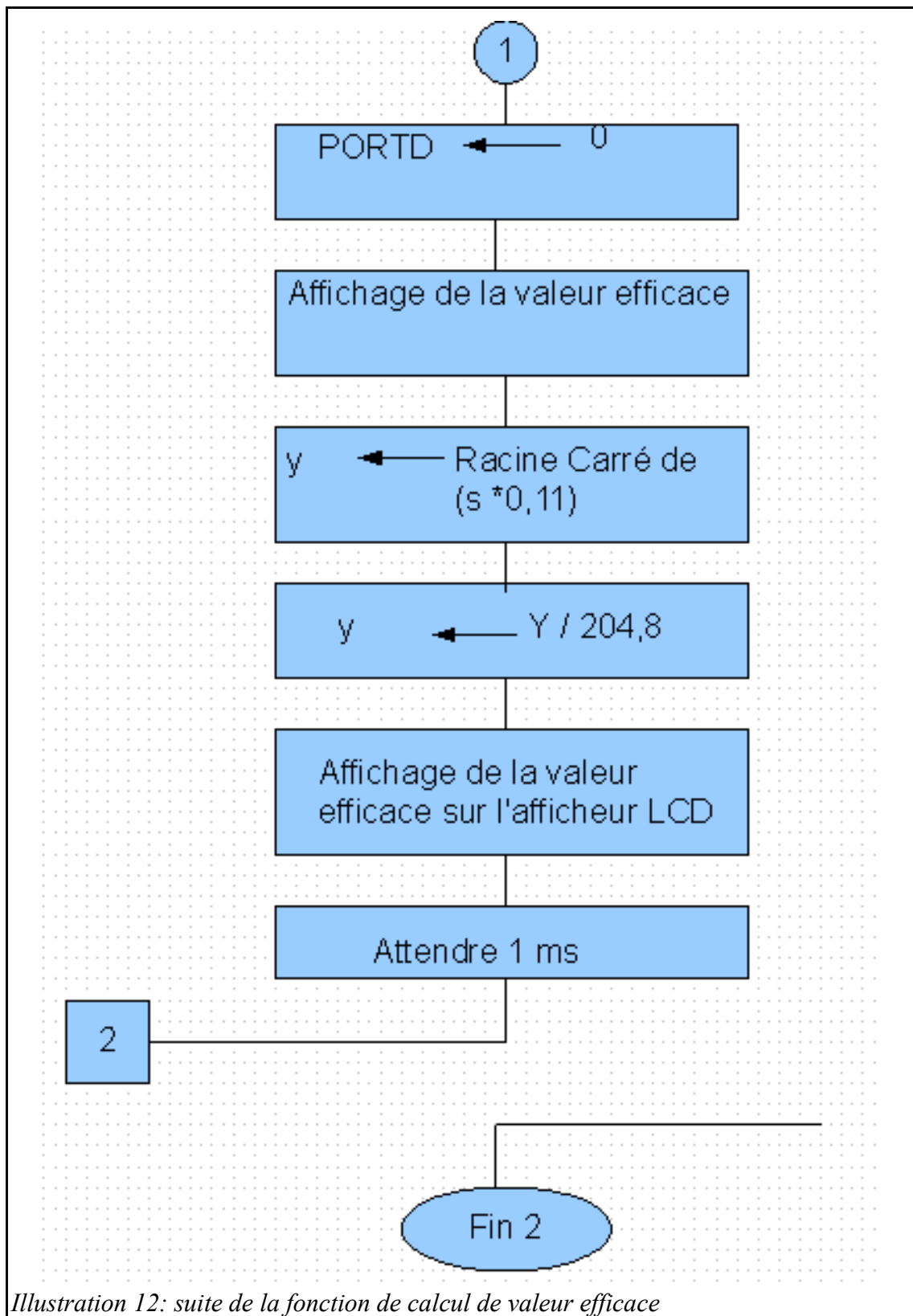


Illustration 12: suite de la fonction de calcul de valeur efficace

```

while (1)
{
    unsigned char tampon[20];
    S=0;
    PORTD=0xFF;           // calculer le temps de conversion et le temps d'affichage en mettant la sortie DO à 1
    for(j=1;j<=90
    {
        x=read_adc(2);    // converti l'entrée analogique en valeur numérique et le met à l'intérieur de tableau.
        // 2 est le numéro de l'entrée de la carte d'interface analogique
        S=S+x*x;         // l'opération qui permet le calcul des valeurs moyennes
    }
    PORTD=0x00;          // un créneau qui nous permet de calculer le temps de conversion
    sprintf(tampon, "%4d", x); // afficher la valeur efficace
    lcd_gotoxy(7,2);
    lcd_puts(tampon);    // afficher la valeur sur l'afficheur

    sprintf(tampon, "%6ld", S); // Pour afficher en long (int) il faut le "l" à ne pas oublier
    lcd_gotoxy(0,3);     // on se place à la 3ième ligne et à la 0 colonne.
    lcd_puts(tampon);

    y= sqrt(S*0.011)     // 110µs/10ms=0.011
    y = y/204.8;         // pour afficher la valeur en Volts
    sprintf(tampon, "%.2f V", y);
    lcd_gotoxy(7,1);
    lcd_puts(tampon);
    delay_ms(1000);

};
}

```

### 3.1. Deuxième expérience

#### 3.1.1. Description

L'expérience effectuée nous a permis de calculer le nouveau temps de conversion qui est égale à la période du signal qu'on va utiliser à la suite de notre expérience. Ce temps de conversion égale maintenant à 20ms ce qui correspond exactement à notre période <sup>2</sup>.

Nous avons obtenu la courbe suivante :

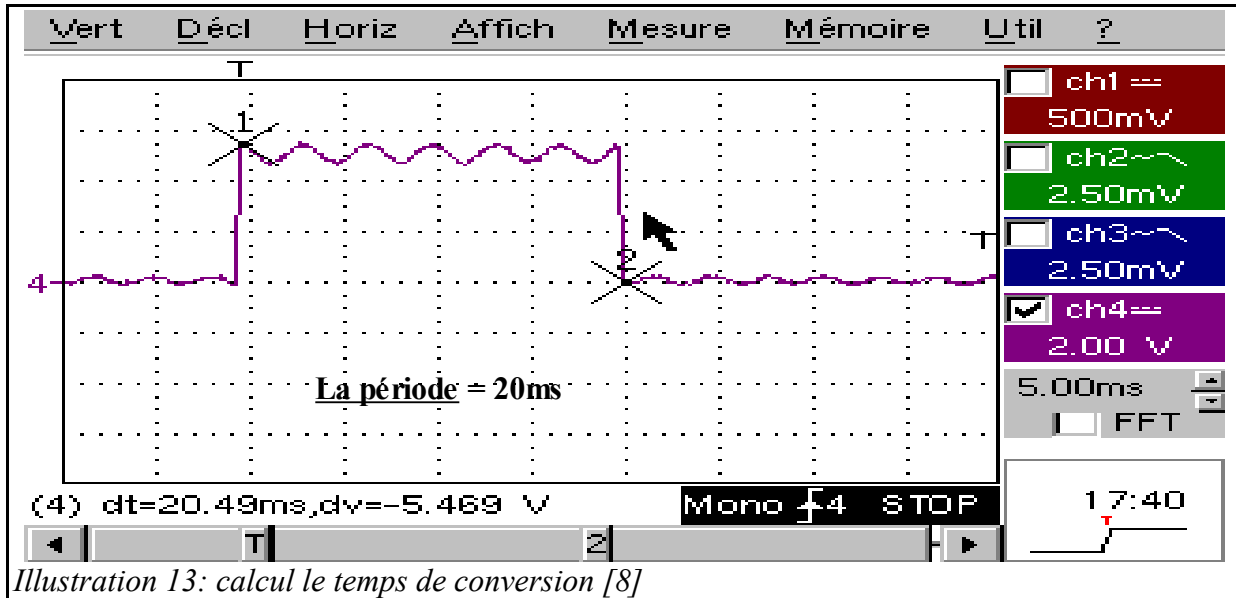


Illustration 13: calcul le temps de conversion [8]

Le signal utilisé est un signal sinusoïdale de fréquence  $F = 50\text{Hz}$  ( donc de période  $T = 20\text{ms}$  ) avec une composante continue et de valeur  $\text{Max } V_{\text{max}}^3 = 5,469\text{V}$ .

Le signal est le suivant :

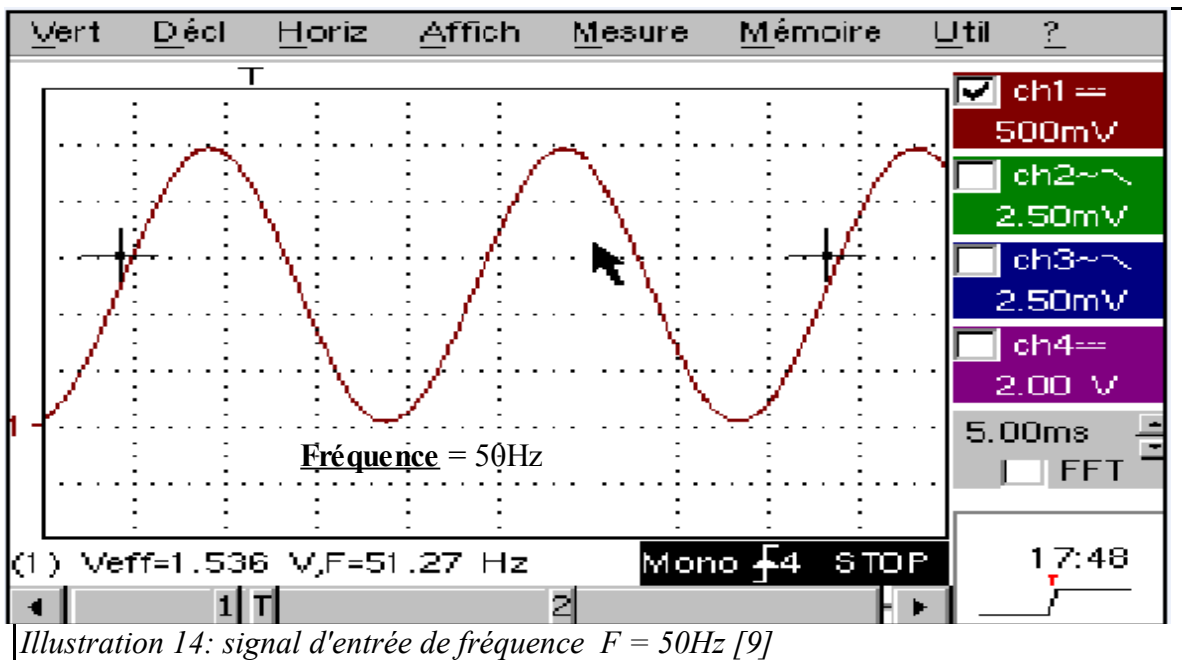


Illustration 14: signal d'entrée de fréquence  $F = 50\text{Hz}$  [9]

<sup>2</sup> La période  $T = 1/F = 1/50\text{Hz} = 20\text{ms}$

<sup>3</sup>  $V_{\text{max}}$  : c'est la valeur maximale de tension

A l'entrée de carte d'interface analogique, qui est connectée au port A de l'Atmega8535, nous avons envoyé le signal précédent sur l'entrée P2<sup>4</sup>.

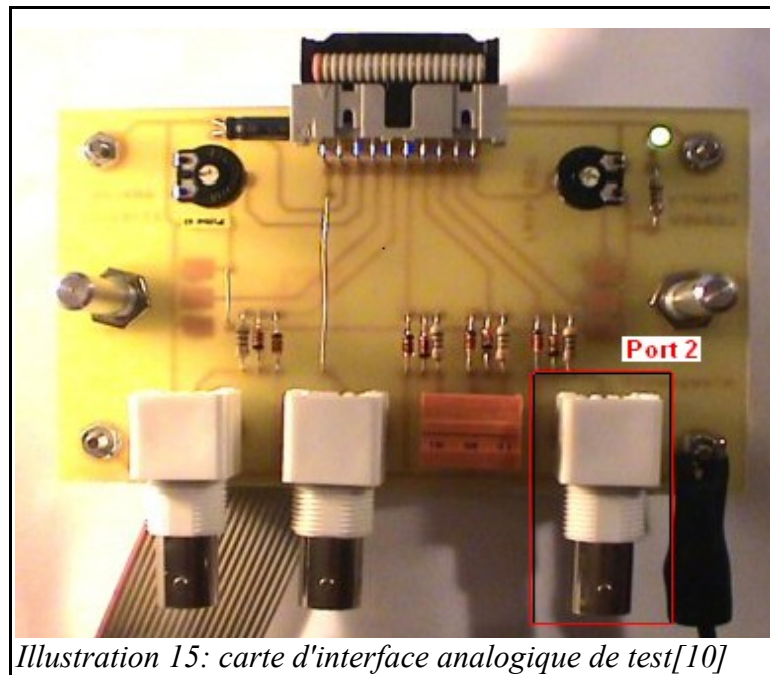


Illustration 15: carte d'interface analogique de test[10]

Après avoir envoyé le signal à cette carte d'interface reliée à la carte d'Atmega8535 via une liaison série, celle-ci est à son tour connectée à l'ordinateur afin de transférer le programme permettant le calcul de la valeur efficace.

### 3.1.2. Programmation

- La déclaration des variables :
- Le port D est en sortie

```
void main(void)
{
  // Declare your local variables here
  long int x ;
  float y;
  int j;
  long int S ;

  PORTD=0x00;
  DDRD=0xFF; //port D en sortie .
}
```

4 P2 : c'est l'entrée 2 de la carte d'interface analogique de test

- Cette fonction permet d'écrire sur l'afficheur LCD

```
// LCD module initialization
lcd_init(16);
lcd_gotoxy(0,0);
lcd_putsf("valeur efficace"); //Pour écrire sur l'afficheur LCD
lcd_gotoxy(0,1);
lcd_putsf("Veff = ");
```

```
while (1)
{
    unsigned char tampon[20];
    S=0;
    PORTD=0xFF; // calculer le temps de conversion et le temps d'affichage en mettant la sortie D0 à 1
    for(j=1;j<=170)
    {
        x=read_adc(2); // converti l'entrée analogique en valeur numérique et le met à l'intérieur de tableau.
        // 2 est le numéro de l'entrée de la carte d'interface analogique
        S=S+x*x; //l'opération qui permet le calcul des valeurs moyennes
    }
    PORTD=0x00; // un créneau qui nous permet de calculer le temps de conversion
    sprintf(tampon,"%4d",x); // afficher la valeur efficace
    lcd_gotoxy(7,2);
    lcd_puts(tampon); // afficher la valeur sur l'afficheur

    sprintf(tampon,"%6ld",S); //Pour afficher en long (int) il faut le "l" à ne pas oublier
    lcd_gotoxy(0,3); //on se place à la 3ième ligne et à la 0 colonne.
    lcd_puts(tampon);

    y= sqrt(S*0.0058) //117µs/20ms = 0.0058
    y = y/204.8; //pour afficher la valeur en Volts
    sprintf(tampon,"%0.2f V",y);
    lcd_gotoxy(7,1);
    lcd_puts(tampon);
    delay_ms(1000);
}
};
```

## 4. Le capteur de courant

Le capteur HASS200-S est un composant électronique qui permet de récupérer une tension image alternative du courant alternatif injecté qui est de l'ordre de 200A.

Afin de récupérer un courant alternatif, nous avons effectué une simple expérience. Elle consiste à enrouler un fil électrique long relié à un rétroprojecteur . 10 enroulements sont suffisants pour récupérer 20A en total c'est à dire 2A par enroulement.

Une fois, la tension alternative image est récupérée, nous appliquerons la même démarche utilisée dans la partie 3-(Analyse et fonctionnement), afin d'afficher la valeur efficace de la tension et par la suite déduire, celle du courant.



*Illustration 16: capteur de courant HASS 200-S [11]*

Suite à l'expérience réalisée, nous avons obtenu les courbes représentatives de la tension image du courant alternatif.

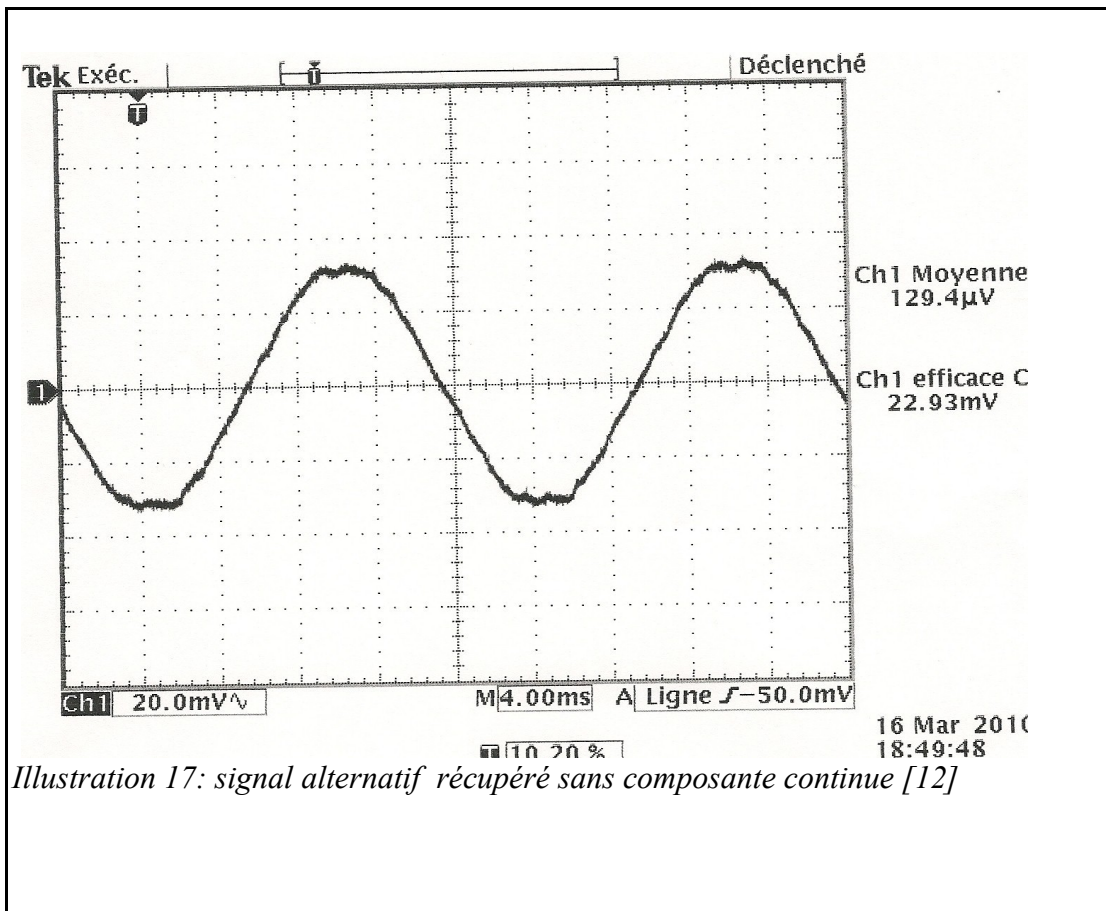


Illustration 17: signal alternatif récupéré sans composante continue [12]

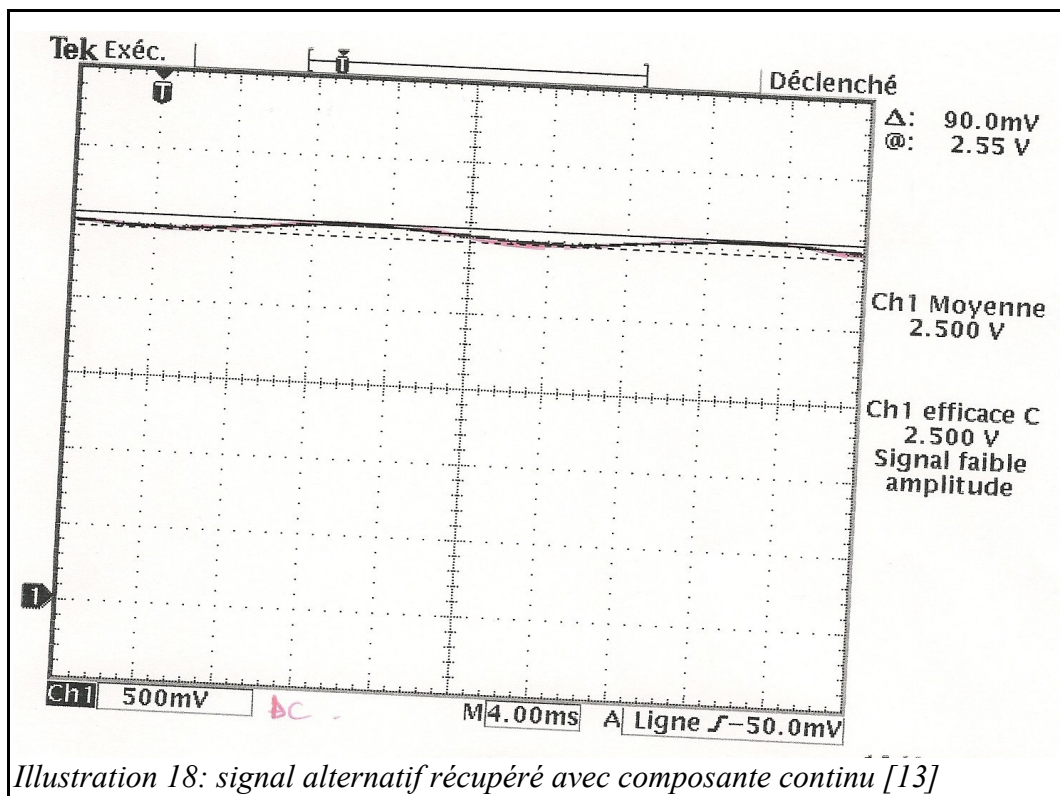


Illustration 18: signal alternatif récupéré avec composante continu [13]



## 5. Test et validation

Lors du déroulement de notre projet tuteuré, nous avons dû faire face à différents tests et essais afin de s'assurer de la validité de celui-ci.

A la première séance, nous avons effectué des tests préliminaires. Une fois le câblage réalisé, les différents ports ont pu être testés, plus précisément les ports A, D et C. Un programme de test relatif au calcul de la valeur moyenne d'un signal continu fourni a permis de vérifier l'état de l'afficheur LCD.

Cette étape achevée, nous nous sommes intéressés à l'étude du Micro-contrôleur, à la compréhension de son fonctionnement global ainsi qu'au repérage des différents ports qui nous seront utiles dans le cadre de notre projet .

A la troisième séance, ils nous a fallu effectuer une expérience afin de pouvoir décomposer notre signal en plusieurs points. Cette décomposition, doit être quasiment exacte et elle nécessite la détermination du temps d'affichage ainsi que le temps de calcul.

Ces différentes valeurs ont été calculées et citées dans la partie 3 (Analyse de fonctionnement). En effet, ces mesures sont prises en compte sur la base d'une demi-période du signal sinusoïdal envoyé ( $T = 10\text{ms}$ ).

Suite à la progression du projet et afin d'être plus précis dans les calculs de la tension efficace, le temps de calcul a été modifié puisque nous avons travaillé sur toute la période ( $T = 20\text{ms}$ ). Dans ce cadre les valeurs exactes que nous avons prises sont de  $117 \mu\text{s}$  (somme du temps conversion et de calcul).

A chaque modification effectuée, le programme est compilé à l'aide du logiciel Code Vision AVR afin de s'assurer de l'absence d'erreurs et de vérifier à l'aide d'un calcul manuel si la valeur efficace affichée sur 32 bits est juste.

Une fois que toutes nos valeurs mesurées sont vérifiées, une simple ligne de code a été ajoutée afin d'afficher la vraie valeur efficace. Elle consistera à diviser la valeur trouvée sur 32 bits de notre valeur efficace par 204,8 (204,8 correspondant au rapport entre la valeur maximale de notre variable de mesure « x » c'est-à-dire 1024 et la dynamique de tension qui est de l'ordre de 5 V).

Durant les tests effectués, nous avons rencontré un problème conséquent relatif à la compatibilité des variables, la difficulté se présentant lors de l'affichage de nos valeurs : la variable de mesure « x » avait été déclarée en tant que « Long int » c'est-à-dire un entier long qui peut être affiché sur 16 bits.

La multiplication de cette valeur par elle-même débordera nécessairement de 16 bits. Dans ce cas, l'affichage à travers la fonction « sprintf » doit être pris en compte. Il faut un « (%ld) » au lieu d'un « (%d) » pour pouvoir afficher le carré de notre variable « x » et par la suite notre valeur efficace sur 32 bits.

Concernant la partie « Capteur de courant », nous n'avons pas eu suffisamment le temps pour tester la partie programmation et afficher la valeur efficace du courant. Nous avons eu l'opportunité de réaliser l'expérience qui est, selon nous, la partie la plus intéressante puisque la partie programmation est identique à celle de l'affichage de la valeur efficace de la tension.



# Conclusion

## Conclusion Ala :

Dans le cadre de notre formation «Génie électrique et informatique industrielle », l'exercice de programmation d'un Micro-contrôleur a été très bénéfique. C'est un exercice intelligent, qui nous a permis de développer trois axes capitaux à la fois, pour une meilleure professionnalisation dans notre domaine :

D'abord au niveau théorique ; cet exercice a permis de valider, de maîtriser et de développer des connaissances théoriques, tant dans le domaine de l'informatique industrielle d'une manière générale et plus précisément au niveau des techniques de création, que de la mise en œuvre et le développement d'un projet au sein d'une entreprise. La programmation du micro-contrôleur nécessitait une bonne maîtrise des langages de programmation C et C++.

Au niveau pratique, la programmation de la carte ainsi que les différents tests nous ont mis face à plusieurs problèmes. Pour les limiter, il a fallu se montrer bien attentif à tout un ensemble de détails tels que la compatibilité des variables ou bien à l'utilisation des différentes fonctions comme la fonction d'affichage«printf».

Travailler en équipe ou encore en binôme est un moyen pédagogique qui reflétant la réalité du terrain et qui permet de développer plusieurs qualités aussi bien personnelles que professionnelles. Pour réaliser un travail comme le nôtre il fallait travailler en binôme. c'est une condition qui nous a permis de développer une intelligence relationnelle et communicative, une responsabilité dans le partage des tâches ainsi que dans l'établissement du planning du travail.

## Conclusion Mouna

Au cours de la réalisation de notre projet qui portait essentiellement sur la programmation, nous avons acquis des nouvelles connaissances sur l'utilisation du logiciel CodeVisionAVR.

L'idée de projet était de calculer la valeurs efficace des grandeurs alternatives, un calcul qui est purement mathématique et qu'il faut programmer. C'est la que réside toute la difficulté du projet.

Grâce à l'aide du professeur, nous avons pu avancer dans notre étude et mieux comprendre la démarche qu'il fallait suivre afin de réussir ce projet. La programmation était un élément essentiel dans notre projet, donc il fallait faire des programmes qui s'adaptent à notre situation.

Tout au long du projet, nous avons rencontré plusieurs problèmes au niveau de la programmation et le calcul de temps de conversion, mais le fait de refaire les programmes et de chercher la faute nous a permis de mieux comprendre voir même la maîtrise de la situation de notre projet étant donné que c'est la première fois que nous faisons ce type de projet.

Le travail d'équipe reste un élément essentiel pour réussir un projet compliqué, c'est un outil pédagogique qui reflète la réalité du terrain et qui permet de développer plusieurs qualités personnelles ainsi que professionnelles.

## Résumé

Au cours du 4<sup>ième</sup> semestre et dans le cadre E&R, nous avons réalisé un projet intitulé «Wattmètre secteur 200A/16V». Il consistera à calculer la valeur efficace de la tension et du courant. Le produit de ces deux grandeurs donnera lieu à la puissance active.

Afin de réussir d'afficher ces valeurs efficaces, nous avons utilisé le logiciel CodeVisionAVR, ainsi que la programmation pour arriver à faire ce calcul compliqué à réaliser. La visualisation se fera grâce à un afficheur LCD.

Tout au long du projet nous avons rencontré plusieurs difficultés. Le travail en binôme a facilité notre tâche. Nous avons validé une partie très importante de notre projet malgré les difficultés que nous avons pu rencontrer. Nous avons réussi à afficher la valeur efficace de la tension. Concernant le courant, nous avons eu une seule séance pour réaliser l'expérience et voir concrètement le rôle du HASS200-S.

L'affichage de la valeur efficace du courant est presque identique à celui de la tension. La seule différence est la conversion de courant en tension image via le capteur de courant HASS200/S.

Tenant compte de nos difficultés en programmation, nous avons réussi à réaliser la partie la plus importante du projet. Il nous manque seulement la partie de la puissance.

**205 mots**

## Index des illustrations

Illustration 1: diagramme bloc du Atmega8535[1].....	7
Illustration 2: afficheur LCD[2].....	8
Illustration 3: choix du microprocesseur et de sa fréquence [3].....	9
Illustration 4: choix du port pour l'afficheur [4].....	9
Illustration 5: transmission du programme de l'ordinateur au microprocesseur [5].....	10
Illustration 6: calcul de valeur efficace d'une tension $v(t)$ [6].....	11
Illustration 7: fonction de calcul de temps de conversion .....	12
Illustration 8: affichage de temps de conversion[7].....	13
Illustration 9: organigramme de la fonction principale .....	14
Illustration 10: suite de la fonction principale.....	15
Illustration 11: organigramme de la fonction calcul de la valeur efficace.....	17
Illustration 12: suite de la fonction de calcul de valeur efficace .....	18
Illustration 13: calcul le temps de conversion [8].....	20
Illustration 14: signal d'entrée de fréquence $F = 50\text{Hz}$ [9].....	20
Illustration 15: carte d'interface analogique de test[10].....	21
Illustration 16: capteur de courant HASS 200-S [11].....	23
Illustration 17: signal alternatif récupéré sans composante continue [12].....	24
Illustration 18: signal alternatif récupéré avec composante continu [13].....	24
Illustration 19: planning prévisionnel et réel.....	26

## Bibliographie

[1] : data-sheet du micro-processeur Atmega8535 / 20-03-2010

[3][4][5][11] : «<http://www.thierry-lequeu.fr>» / 20-03-2010

[6] : Le logiciel Maple 9.5 / 20-03-2010

[7][8][9][10][12][13]: Nos expériences réalisées au cours du projet / 8, 9, 10 et 11-03-2010

## Annexes